



汕 頭 大 學

SHANTOU UNIVERSITY

硕 士 学 位 论 文

题 目： 基于点线特征的视觉定位系统研究

英文题目： The Research of Visual location System based on
Point and Line features

姓 名 王宇鹏 学 号 51509010

所在学院 工学院 导师姓名 范 衡

专 业 电子与通信工程

入学日期 2015 年 9 月 答辩日期 2018 年 5 月

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其他机构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，均已在论文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

作者签名：_____ 日期：_____年____月____日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档，允许论文被查阅和借阅；学校可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、缩印或其他复制手段保存和汇编论文；学校可以向国家有关部门或机构送交论文并授权其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文，按照保密的有关规定和程序处理。

作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘 要

移动机器人的定位系统研究，是机器人自主导航技术中最基础也是最关键的技术之一。随着计算机视觉技术的发展，基于视觉传感器的视觉里程计技术取得了巨大的发展，成为了定位系统研究中重要的分支。点特征由于优秀的鲁棒性以及不同图像上的良好辨识度，受到了极为广泛的关注和应用。但是点特征却对环境有一定的依赖。在人造的结构化场景中，点特征的缺失使得依赖其设计的视觉里程计受到了极大的影响。合理利用场景中大量存在的线特征，会和点特征形成良好的互补，可以极大的拓展视觉里程计的应用场景。

本文通过从单目相机图像中获取点特征和线特征，提出了一种综合利用点线特征的半直接法视觉里程计算法。首先设计基于单目相机的视觉感知系统，从环境中获取需要的点特征和线特征。为矫正相机的畸变参数，获得空间坐标系和成像平面的对应关系，对相机进行标定，求取相机参数。

其次，构建视觉定位系统模型。本文提出的视觉里程计算法，首先通过直接法实现粗略但快速的定位。然后利用特征法，融合点特征和线特征获取当前帧和关键帧之间的精确位移。通过一种基于概率的深度滤波方法确定对空间中的点特征、线特征位置，用来实现位姿的精确估计。

最后，为了验证算法的可行性，在 TUM 数据集中采用多个序列进行实验，比较真实轨迹和估计轨迹之间的误差，验证了本文方法可以完成移动机器人的精确定位。同时为了验证算法在现实环境下的适用性，在走廊场景下通过和基于激光的里程计算法进行比较，结果表明，本文方法可以在走廊环境中准确定位机器人位置。

关键词：视觉定位；单目视觉；点线特征；半直接法视觉里程计

Abstract

The localization of mobile robot is one of the most basic and critical technology in autonomous navigation field. With the development of computer vision, visual odometry has made great progress and become an important branch of the localization research. Due to excellent robustness and good visibility on different images, point features has been widely used. However, point feature are more dependent on the environment. In man-made structured environment, the lack of point feature has a big influence on visual odometry which rely on point feature. Make use of the line features and point features together can expand the application scenario of the visual odometry.

In this thesis. We propose a semi-direct visual odometry based on point and line features extracted from the monocular camera. Firstly, a visual sensing system based on monocular camera is designed to extract point and line features in the environment. To rectify the aberration of camera and get the relationship between the space and image coordinate system, the camera was calibrated and the intrinsic parameters of the camera are obtained.

Secondly, a visual localization system is modeled. The first step of the visual odometry proposed in this paper is locate the approximate pose of the robot. It is achieved by the direct method. Then the feature-based method which use point and line features together is used to get the accurate pose between current frame and keyframe. A depth filter which based on probability is used in system to calculate the position of line and point in the space. This can be used to get the accurate pose.

Finally, in order to verify the feasibility of the proposed visual odometry, several sets of image sequences in TUM dataset captured by monocular camera are used. The error between the real trajectory and the estimated trajectory show that the visual odometry system proposed in the paper has a good localization performance. At the same time, the experiment in the corridor environment which compares the result between odometry based on laser and visual odometry proposed in the article was built. The experimental result demonstrate that our method can locate the pose of the robot

accurately in the corridor environment.

KeyWords: Visual localization; Mono Vision; point and line features; semi-direct visual odometry

目 录

摘 要	I
Abstract.....	II
第一章 绪 论	1
1.1 课题研究背景与意义	1
1.2 国内外研究现状	4
1.3 主要研究内容	6
1.4 章节安排	6
第二章 相机模型和线特征	8
2.1 相机模型	8
2.1.1 针孔相机模型	8
2.1.2 相机畸变参数	10
2.2 相机标定	12
2.3 图像中的线特征	13
2.3.1 线特征的提取	13
2.3.2 线特征的描述	15
2.3.3 3D 线段的参数化	16
2.3.4 3D 直线位置初始化	18
2.4 本章小结	19
第三章 视觉里程计算法理论	20
3.1 视觉里程计问题的建模	20
3.2 基于特征点法的视觉里程计	22
3.2.1 对极几何	23
3.2.2 点特征的重投影误差	24
3.2.3 线特征的重投影误差	26
3.3 基于直接法的视觉里程计	29
3.3.1 点特征的光度误差	30
3.3.2 线特征的光度误差分析	33
3.4 本章小结	33
第四章 基于点线特征的视觉里程计	34
4.1 系统框架	34
4.2 运动估计线程	35
4.2.1 初始化	35
4.2.2 稀疏图像匹配	36
4.2.3 特征匹配	37
4.2.4 位姿及结构优化	39
4.3 地图构建线程	40
4.3.1 特征的选取	41
4.3.2 深度滤波器	42
4.4 本章小结	44
第五章 实验与分析	45
5.1 仿真实验	45
5.2 走廊环境下点线视觉里程计的定位实验	55

5.2.1 实验平台硬件	55
5.2.2 试验结果与分析	57
5.3 本章小结	59
第六章 总结与展望	60
6.1 总结	60
6.2 展望	60
参考文献	62
致 谢	66

第一章 绪 论

1.1 课题研究背景与意义

机器人技术作为计算机科学、控制科学、微电子技术等众多学科交叉的领域，近年来随着这些学科的飞速发展而取得了巨大的进步，逐渐进入到人们的日常生活中。21 世纪世界各国对于机器人技术的发展投入了巨大的精力，将其作为高新技术、新兴产业竞争的制高点。发展机器人技术并将其作为人工智能领域的载体，对国家的经济和国防建设都会起到至关重要的作用。其中自主移动机器人作为智能移动平台，率先在外太空探索、工业、医疗、服务等行业获得了广泛应用，成为了机器人学术界的研究热点。特别是无人驾驶汽车概念的提出，更是使得移动机器人技术受到了学术界、工业界的极大关注。图 1-1 是特斯拉无人车驾驶系统。



图 1-1 特斯拉无人车系统

在自主移动机器人的研究中，一大关键点就是让移动机器人能安全的在环境中移动并到达指定位置。为了实现这一功能，就需要解决三个问题：1、我在哪里？2、周围的环境怎么样？3、我要去哪？第一个问题对应移动机器人领域的定位问题，第二个问题对应地图构建问题，第三个问题对应移动机器人的导航问题^[1]。移动机器人的将定位作为基础，才能实现后续的地图绘制、导航等功能。

在移动机器人的研究中，首先面临的问题就是要对机器人的运动进行检测。常用的传感器有两种：一种是对机器人自身的运动情况进行检测的传感器，常见

的有编码器、加速度计、惯性导航设备等；另一种是对机器人运动环境信息进行检测的外部传感器，常见的有激光雷达、超声波探测器、相机等等。不同的传感器由于结构和特点的不同，检测到的信息丰富程度不同，受环境因素影响程度也不同。对于编码器、加速度计等传感器可以比较方便的读取检测数据，但是检测精度较低，长时间运行的时候更容易产生积累误差。对于外部传感器，激光雷达是目前为止研究最深、应用范围最广的传感器。激光雷达可以对距离信息进行精确的测量，精度高而且量程大，是目前使用的最为可靠的外部传感器，在实际中有着最为广泛的使用。但是在获取高精度测量数据的同时，激光传感器也面临着体积功耗较大、传感器价格昂贵的问题。相比较激光传感器而言，相机是一种新兴的外部传感器。激光传感器一般提取的是二维的距离信息，而相机可以轻松获得多维度的信息，价格相对而言也更为低廉。使用相机作为传感器，可以获得包括几何外形、颜色、深度在内的多种信息，对移动机器人的移动环境进行更为丰富的描述。但是视觉传感器的缺点也很明显。激光传感器可以直接获取距离信息，而相机需要从图像中进行特征提取、特征匹配、位置计算等多种鲁棒的算法，才能获得需要的信息。相比而言，使用相机传感器会消耗更多的计算资源，造成较大的运行负担。



图 1-2 编码器、IMU 惯导设备、激光雷达和深度相机

在获取到传感器信息后，就需要通过各类算法来计算机器人的移动。当移动机器人在室外环境的时候，通过 GPS、北斗导航系统等卫星定位系统，可以方便的实现机器人的精确定位。但是处于室内环境时，由于卫星信号的衰弱，传统移

动机器人只能通过无线 AP^[2] (Wi-Fi Access Point, 无线接入点), 移动基站等设备提供外源的信号进行三角测量来获得相对的位置。这种方法定位精度较差, 即使在 AP 和基站集中的城市也往往只有几十米的精度, 一般的地方只有一两百米, 根本无法满足高精度定位的要求。因此, 移动机器人的无源定位研究就有重要的意义。

在实现移动机器人无源定位的时候, 最早提出的方法是航位推算法。这种方法依靠机器人内部的轮式里程计、惯导设备等各种传感器提供信息, 递归的预测机器人的位置。航位推算法可以独立实现简单的定位, 但却有着难以解决的缺陷: 轮式里程计只能针对轮式小车, 而且无法克服车轮移滑导致的计数或测量误差; 惯导设备通过积分来计算机器人的姿态并计算位姿信息, 但长时间的计算会导致出现“漂移”问题。

随着计算机视觉技术的发展, 人类可以通过摄像机和计算机识别目标, 并对其进行跟踪和测量。以摄像头为传感器来实现移动机器人定位功能的视觉里程计^[3]应运而生。视觉里程计 (VO, Visual Odometry) 通过摄像头获取图像序列, 运用计算机视觉的方法从图像中提取特征, 通过在不同帧间进行特征匹配来计算移动机器人的运动参数。视觉里程计相比较传统的里程计有明显的优势。它仅需要使用相机传感器, 无需场景和运动的先验信息; 可以在非结构化的环境或者非常规的平台上运行。而且由于图像包含丰富的环境信息, 可以为移动机器人的地图绘制、导航功能提供更充分的支持。

视觉里程计随着多年的发展, 已经有了较为成熟的发展。但是仍然有很多的制约因素限制了它的应用, 其中一个重要的局限就是对场景特征的过度依赖。现在主流的视觉里程计系统, 大部分都是在点特征的基础上进行的系统搭建。但是在纹理缺失的场景下, 无法提取足够的点特征往往使得系统的运行收到极大的干扰, 无法保证系统的精度。近年来获得较大发展的直接法视觉里程计, 虽然通过直接对像素进行处理避免了受到特征的限制, 但稠密或者半稠密的设计往往导致位姿跟踪需要进行大量的计算, 使用环境有一定的限制。而在人造环境中, 充斥着大量的结构化特征, 如线特征、面特征等, 但是大量白墙和地面的存在缺失的环境中点特征的提取成为一大难题。这种情况下, 使用具有良好精度和光照不变性的线特征, 就成了提升视觉里程计系统性能的一种有效方法。

1.2 国内外研究现状

将相机拍摄的图像作为输入来预测移动机器人轨迹，最初是在 Moravec 设计的斯坦福推车 (Stanford Cart) 上实现的。这是世界上第一架具有视觉能力并且可以自动导航的移动机器人。1980 年, Matthies 等人提出了视觉里程计的概念^[4], 并设计了特征提取、特征匹配与跟踪、运动估计的理论框架作。这一框架作为现代视觉里程计的雏形, 为大多数视觉里程计系统遵循。

视觉里程计最初的应用, 主要是为了满足 NASA 对于火星探测车的性能要求。火星探测车需要在崎岖的火星表面工作, 所以需要探测车能够测量在三维空间内六个自由度的运动。单纯的轮式里程计无法满足性能的需求。为了满足定位精度的要求, NASA 的 Yang 等人设计了“勇气号”、“机遇号”的视觉定位系统, 辅助火星探测车正常运行。火星车项目的顺利进行, 证明视觉定位系统在平坦路况下可以和轮式里程计有着同样良好的表现, 在复杂环境下视觉定位系统的表现则要远强于轮式里程计。视觉定位系统在火星探测这种复杂的任务中取得了成功的应用, 对视觉里程计的研究发展有着极大的推进作用。

2004 年, Nister 等人发表的关于路标的论文^[5], 设计了第一套实时的视觉里程计系统。他们在 NASA 火星车的视觉定位基础上做了扩展, 通过计算由于机器人移动而引起的图像变化, 增量的估计机器人的运动参数。而他们设计的单目视觉和立体视觉两类视觉里程计的实现途径和流程, 即特征提取、特征匹配、运动估计的系统流程, 为后来的视觉里程计研究奠定了基础。

2005 年、2007 年, 无人驾驶挑战赛在美国国防高级计算研究所的主持下进行。斯坦福大学设计制造了 Stanley 和 Junior 两辆无人车参赛, 综合使用了激光雷达、GPS、惯导设备、相机等多种传感器来实现无人车的定位导航, 并取得了一次冠军、一次亚军的成绩。

Tardif 等人在 2008 年提出了一种城市环境下长距离运行的车载视觉里程计系统^[29]。在其中实现了旋转矩阵的分解和平移向量的估计。旋转矩阵通过无穷远点进行估计, 平移向量则是从恢复的 3D 地图来求取, 通过 RANSAC 方法, 剔除了误差的点。

Badino 等人在 2013 年提出了一种整合多帧特征点信息的双目视觉里程计^[6]。不同于一般的视觉里程计主要通过相邻两帧图像上的连续特征点来估计运动, 他

们通过追踪完整路径上的全部特征点来实现位姿估计。在微量的运算量增加基础上，这种方法实现了定位精度的大幅提升。



图 1-3 Junior 无人车

慕尼黑工业大学计算机视觉组的Engel等人在2016年提出了属于稀疏直接法的视觉里程计DSO (Direct Sparse Odometry)^[7]。作为纯直接法的视觉里程计系统,DSO通过计算相邻帧间的光度误差来估计机器人的运动,同时采用了一套直接的概率模型来优化所有的参数。在运算速度上,DSO可以达到传统特征点法速度的五倍。

Georg Klein改进了基于关键帧优化的SLAM系统^[33],通过对点线特征的综合使用解决了高速运动下图像模糊导致的跟踪失败问题,并设计了利用线特征在三个视图下的匹配关系来估计相机运动。

Yan Lu在RGB-D相机上实现了点线综合特征的视觉里程计设计^[21]。利用RGB-D相机可以直接获取深度信息的特点,通过深度图对空间线段进行了最小二乘拟合,证明了点线特征综合使用对系统性能的提升。

Ruben实现了点、线特征在双目相机下的应用^[19]。分别计算点特征、线特征的重投影误差,根据二者的重投影误差比较来分配两种特征的使用权重。

国内对视觉里程计的研究起步晚于国外,但是随着近年来移动机器人领域的快速发展,越来越多的国内学者开始关注这一领域。浙江大学作为国内计算机视觉方面的先行者,在2007年的时候,浙江大学的吕强就设计了基于SIFT特征的单目视觉里程计^[8]。

国防科技大学较早开始进行移动机器人和无人车的研究,在视觉里程计研究

方向也处于国内领先地位。由国防科技大学自主研制的红旗 HQ3 无人车，在 2011 年 7 月首次完成了 286 公里的高速路全程无人驾驶实验，是我国在智能车辆研究方面的突破性进展。

孙作雷设计了基于 RANSAC 的奇异值剔除的单目视觉里程计^[30]，使用 RANSAC 算法提出首次计算的误匹配，通过提取地面数据来获得相机的平移尺度。

张毅等人通过改进 SURF 算法提高特征匹配的正确率和鲁棒性^[31]。但是特征点的提取和描述子计算消耗了较多的时间，在纹理信息比较缺乏的环境下，系统无法提取足够的特征，导致系统无法稳定运行。

1.3 主要研究内容

本文主要工作基于室内环境下的点特征、线特征综合应用的单目视觉里程计设计。针对结构化场景下线特征丰富、点特征缺失导致基于点特征的视觉里程计失效的问题，在 SVO 的工作基础上进行了扩展，综合使用点特征和线特征对相机位姿进行估计，实现对机器人位置的精确估计。主要内容如下：

(1) 研究基于点特征在单目视觉里程计中的应用，包括通过特征提取来实现的重投影方法和直接对像素点进行处理的直接法的误差分析。

(2) 研究线特征在单目视觉里程计中的应用，讨论线特征的提取方法，线段在空间中的描述，以及线特征的重投影误差的分析。

(3) 研究同时使用点线特征在纹理缺失环境下对移动机器人定位的改善效果。

1.4 章节安排

本论文的章节安排如下：

第一章：绪论。主要介绍视觉里程计问题的研究背景和意义，概述了视觉里程计的发展历史和现状，并概括本文的研究内容。

第二章：相机模型和线特征。首先对针孔相机模型进行了介绍，讲解视觉里程计系统需要用到的各坐标系间转换关系。然后介绍了如何对相机进行标定，获得各项需要的相机参数。最后介绍在点线视觉里程计中需要用到的线特征，从多个角度对线特征进行了描述。

第三章：视觉里程计算法理论。构建了视觉里程计问题的数学模型，并对基于特征提取的方法和对像素直接处理的直接法进行了详细的阐述。对点特征和线特征在两种不同的视觉里程计方法下的误差进行了分析。

第四章：基于点线特征的视觉里程计设计。结合第二章的背景知识和第三章对视觉里程计的理论描述，详细介绍点线综合利用下视觉里程计系统的实现方法，包括对相机位置的估计和优化，地图的构建方法。

第五章：实验与分析。通过仿真实验对本文提出的系统进行验证。

第六章：总结与展望。该部分对本文的主要工作进行总结，对之后可以优化和提高的方向做设想和展望。

第二章 相机模型和线特征

2.1 相机模型

视觉里程计使用相机作为传感器，图像特征在相机视角发生少量变化后会保持不变。视觉里程计根据对这些图像特征的位置、几何、亮度、色彩等信息进行相机位置估计。相机模型描述三维空间点映射到二维平面上的过程，称为投影；二维坐标点映射到三维空间的过程称为反投影。

2.1.1 针孔相机模型

常见的相机模型包括针孔相机模型、双目相机模型、RGB-D 相机模型等，本文使用的是针孔相机模型。针孔相机模型，就是一幅视图通过透视变换，将三维空间的点投影到图像平面。图 2-1 为相机针孔模型示意图。

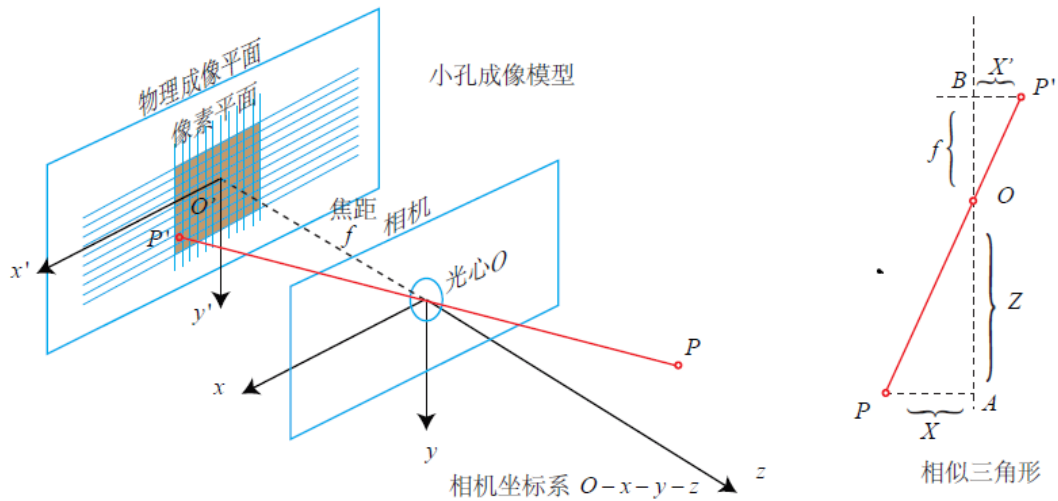


图 2-1 相机针孔模型

设 $O-x-y-z$ 为相机坐标系， z 轴指向相机前方， x 轴向右， y 轴向下。设 O 为摄像机的光心，即针孔相机模型中的针孔。三维空间点 P 经过针孔 O 投影后，在物理成像平面 $O'-x'-y'-z'$ 上获得成像点 P' 。设 P 的坐标为 $[X, Y, Z]^T$ ， P' 的坐

标为 $[X', Y', Z']^T$ ，并设置物理成像平面到小孔的距离为 f （焦距）。根据相似三角形原理，可得：

$$\frac{Z}{f} = -\frac{X'}{X} = -\frac{Y'}{Y} \quad (2.1)$$

其中负号表示所成的像是倒立的。我们将成像平面对称到相机前方，和三维空间点放在相机坐标系同一侧，可以将公式（2.1）中的负号去掉简化为：

$$\begin{aligned} X' &= f \frac{X}{Z} \\ Y' &= f \frac{Y}{Z} \end{aligned} \quad (2.2)$$

公式（2.2）描述了点 P 和它的像之间的空间关系。在相机中我们获得的是成像平面上的像素，设物理成像平面上固定着一个像素平面 $o-u-v$ ，通常设置像素平面的原点 o 在图像的左上角， u 轴向右与 x 轴平行， v 轴向下与 y 轴平行。 P' 的像素坐标为 $[u, v]^T$ 。像素坐标系和成像平面相差一个缩放和平移。设 u 轴上像素坐标缩放为 α 倍， v 轴上缩放为 β 倍，原点平移了 $[c_x, c_y]^T$ 。则 P' 的坐标和像素坐标 $[u, v]^T$ 关系为：

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \quad (2.3)$$

代入式（2.2）并将 αf 合并为 f_x ，将 βf 合并为 f_y ，得

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases} \quad (2.4)$$

其中， f 的单位是米， α, β 单位是像素/米，所以 f_x, f_y 的单位是像素。式（2.4）

可以改写为矩阵形式：

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq KP \quad (2.5)$$

通常，我们将中间的量组成的矩阵 K 称为相机的内参数矩阵。

在公式 (2.5) 中， P 对应空间点在相机坐标系下的坐标。因为相机是在三维空间运动，我们需要求解的坐标也应该是世界坐标系下坐标，记作 P_w 。通过从世界坐标系到相机坐标系的旋转矩阵 R 和平移向量 t ，相机的位姿可以表示为：

$$ZP_{uv} = Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K(RP_w + t) = KTP_w \quad (2.6)$$

像素坐标 P_{uv} 已知道的情况下，对应相机坐标系下空间点的位置不确定，但是由相机针孔模型可知点 P 位于光心 O 和像素点 P_{uv} 的射线上。如果我们可以测得其在 z 轴上的坐标，即深度 d 后，就能唯一确定点 P 的坐标，我们将通过像素点推测空间点的过程称为反投影：

$$P_w = dK^{-1}P_{uv} \quad (2.7)$$

2.1.2 相机畸变参数

当实际使用相机的时候，为了获得好的成像效果，通常在相机的前方会加透镜。但是光线在穿过透镜投影到成像平面的过程中，由于透镜的存在会对光线造成一定影响，导致光线在成像平面上的位置发生偏差。透镜对光线的影响主要有两方面^[28]：径向畸变和切向畸变。

透镜自身形状，会对光线的传播造成的影响，这种情况下引起的畸变成为径向畸变。针孔相机模型中，我们使用的都是理想情况，认为光线投影过程中保持一条直线。但是在实际拍摄的时候，透镜的存在会使得一条空间中的直线投影出曲线的情况。而且越靠近成像平面边缘，弯曲现象就越明显。径向畸变分为桶形畸变和枕形畸变两种，二者均会随着透镜中心和成像平面之间的距离增加而增长，可以用下列多项式函数来表示：

$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{cases} \quad (2.8)$$

其中 $[x, y]^T$ 是未纠正的点的坐标， $[x_{corrected}, y_{corrected}]^T$ 是纠正后的坐标。它们均为归一化平面上的点，而不是像素平面上的点。

除了透镜自身的原因外，机械组装过程中由于透镜无法与成像平面完全平行也会导致光线的位置变化，这种情况称为切向畸变。切向畸变可以通过下式来纠正：

$$\begin{cases} x_{corrected} = x + 2p_1xy + p_2(r^2 + 2x_2) \\ y_{corrected} = y + p_1(r^2 + 2y^2) + 2p_2xy \end{cases} \quad (2.9)$$



图 2-2 透镜导致的图像畸变

通过上述对径向畸变和切向畸变的描述，对于相机坐标系下空间点 $P(X,Y,Z)$ ，通过 5 个畸变参数就可以找到在成像平面上正确的投影位置：

- (1) 将空间点 $P(X,Y,Z)$ 投影到归一化图像平面，坐标为 $[x,y]^T$
- (2) 将归一化平面坐标纠正

$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x_2) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2y^2) + 2p_2xy \end{cases} \quad (2.10)$$

- (3) 纠正后的点通过相机内参矩阵投影到成像平面，求得空间点 P 正确的像素坐标：

$$\begin{cases} u = f_x x_{corrected} + c_x \\ v = f_y y_{corrected} + c_y \end{cases} \quad (2.11)$$

在视觉里程计实际使用过程中，对于去畸变处理，常见的处理方法是先对图像进行去畸变，得到去畸变的图像，然后讨论此图像上的点对于的空间坐标关系。

2.2 相机标定

相机标定的作用，其实就是复原相机模型中空间点的三维几何位置和二维成像平面上像素点之间的一一对应关系。相机标定的方法，可以根据辅助的标定物空间维数来分为两大类：三维标定和平面标定。常用的标定方法主要是二维标定方法，以张正友平面标定法^[27]为代表。

张正友标定法通过棋盘图来实现相机的标定，它的计算原理主要是通过空间点位置和成像平面的对应关系来计算相机内参矩阵的各项系数。由于针孔相机模型在前文已经有详细推导，这里就不在对其原理进行推导。

使用张正友标定法，要对一个已知尺寸的平面棋盘图，在三个以上的不同方位拍摄图像。拍摄过程中，保持棋盘图位置不变，改变相机位姿，相机位姿无需知道。求解参数的时候，假设相机内参矩阵不变，只有外参发生变化。

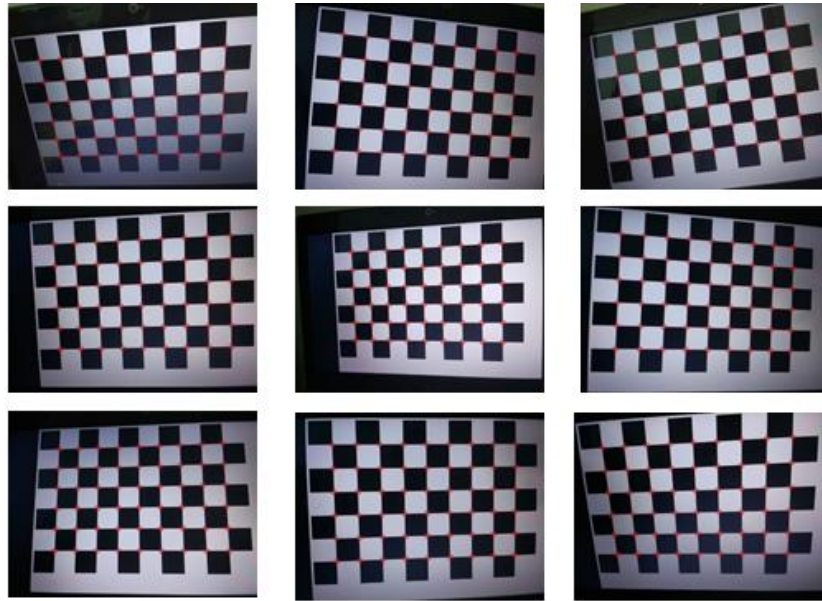


图 2-3 张正友标定法使用过程

通过张正友实验，我们测得所使用的单目相机参数 K 为

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 491.5822 & 0 & 325.4727 \\ 0 & 488.2297 & 228.9611 \\ 0 & 0 & 1.0 \end{bmatrix} \quad (2.12)$$

2.3 图像中的线特征

马尔的视觉计算理论认为,视觉是一个处理过程,要对外部世界的图像构成有效的符号描述。它的核心问题是在对图像结构理解的基础上,推导出外部世界的结构,最终实现对外部世界的理解。对于由二维图像推导三维物体结构的信息处理过程,马尔将其分为了三个阶段:图像初始简图(primal sketch)、2.5 维简图(2.5 dimensional sketch)和三维结构(3D model)^[9]。其中图像初始简图阶段是指由图像中的变化和结构获得适当的表象,通过边缘段、线、斑点、端点等基元的检测来描述图像的亮度变化、确定亮度变化的空间位置。因此对特征点、边缘等进行检测,是计算机视觉理论的基础。

视觉里程计主要的问题是根据图像来估计相机的运动。在解决这个问题的时候,首先需要从图像中选取有代表性的点,这些点在相机视角发生少量变化的时候会保持不变。在经典的 SLAM 模型中,这些点被称为路标。而在视觉里程计中,路标即是图像特征 (Feature),它们需要具备以下特征:

1. 可重复性 (Repeatability): 相同的区域可以在不同的图像中找到。
2. 可区别性 (Distinctiveness): 不同的区域有不同的描述。
3. 高效性 (Efficiency): 同一图像中,特征的数量应远小于像素的数量。
4. 本地性 (Locality): 特征仅与一小片图像有关。

图像信息中,可以作为特征的主要是角点、边缘和区块部分,对应构成点特征、线特征和面特征。点特征和线特征相比于面特征,在不同图像之间具有更高的辨识度,也有着更为广泛的应用。在不同场景中,点线特征的丰富程度是不一样的。线特征因其天然的光照和视角不变性,可以更加直观的用来描述环境;而点特征则具有更加广泛的分布。将这两类特征进行结合,可以有效的提高系统的精度和鲁棒性。点特征有着广泛的应用,在此不再赘述。这里主要介绍线特征的提取和描述。

2.3.1 线特征的提取

本文使用的 LSD (Line Segment Detector) 算法是 Rafeal 在 2010 年设计提出的一种线段检测算法^[10]。它能在线性的时间内得出亚像素级精度的检测结果。LSD

的核心是像素合并与误差控制。通过检测图像中梯度变化较大的像素点集，将梯度方向相近的像素点进行合并。算法首先计算每个像素的梯度方向，所有像素的梯度构成梯度场。在梯度场中方向相似并且在容忍度 τ 内的像素点构成线段支持域，具体图 2-4 所示。

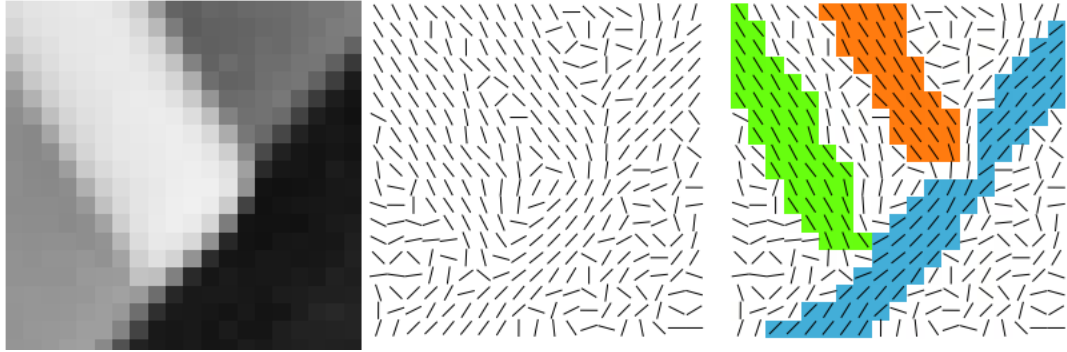


图 2-4 线段支持域

每个线段支持域都是一个直线分隔的备选区域，通过一个矩形来描述。线段支持域的主轴用来表示矩形的主方向，矩形需要覆盖整个区域。当矩形内某像素的梯度方向和矩形的主方向在阈值范围内保持一致，该像素点成为支持点。通过矩形内支持点所占的比例来判定该线段支持域是否为一条直线。

基于 LSD 的线特征提取算法流程如下：

输入：灰度图

输出：一系列的直线分隔结果

Step1: $s=0.8$ 的尺度对输入图像进行高斯核采样。

Step2: 计算每个像素点的梯度方向，包括水平和垂直方向的梯度。

Step3: 根据梯度值对所有点进行伪排序，建立状态列表，所有点设为 UNUSED。

Step4: 将梯度值小于 ρ 的点状态表中相应位置设置为 UNUSED。

Step5: 取出列表中梯度最大的点作为种子点 (Seed)，设置状态为 USED

do:

a.以 Seed 为起点，搜索附近 UNUSED 切方向在阈值 $[-\tau, \tau]$ 范围内的点，状态改为 USED.

b.生成包含所有满足点的矩形 R。判断支持点密度是否满足阈值。如不满足，将 R 分隔为多个矩形框，直到满足条件。

- c. 计算拟合矩形精度误差 NFA。改变 R 使得 NFA 的值变小直至小于阈值，将 R 添加入输出列表
- d. 计算 NFA
- e. 改变 R 使 NFA 的值更小直到低于阈值，R 加入输出列表

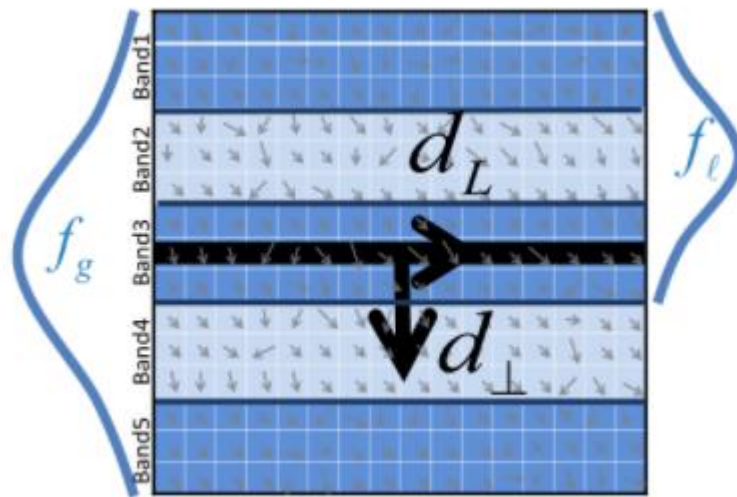
step6: 继续回到第五步，从链表中找到下一个种子点，在剩下的图像中进行区域扩散，知道遍历全图，得到所有检测的直线。

2.3.2 线特征的描述

特征匹配是视觉里程计中极为关键的一环。通过图像与图像或者图像与地图中间描述子进行精确匹配，可以为后续的位姿估计、优化等环节节省大量的资源。线特征的匹配相对于点特征的匹配来说，有更多的困难：包括线段位置的误差、线段的断裂、特殊场景下辨识度较低等等。Lilian Zhang 提出了 LBD 描述子^[11]，引入了全局和局部的高斯权重系数，匹配效果和速度都能满足需求。

在提取出线特征后，LBD 描述子从线段提取域开始计算。该支持域被划分为一条组带 $\{B_1; B_2; \dots; B_m\}$ ，每个条带都是线段支持域的子区域并且互相之间平行。

定义线段的方向 d_L 和正交方向 $d_{\perp}$ ，如图 2-5 所示。



2-5 LBD 特征描述中的线段支持域

每一个条带 B_j 可以计算出对应的特征向量 BD_j 。将所有条带的特征向量合并，可以形成 LBD 描述子：

$$LBD = (BD_1^T, BD_2^T, \dots, BD_m^T)^T \quad (2.13)$$

条带 B_j 通过最近的两相邻行的条带 B_{j-1} 、 B_{j+1} 的局部梯度 $g' = (g'_{d_L}, g'_{d_\perp})$ 来计算。对于第 k 行，有：

$$\begin{aligned} v1_j^k &= \lambda \sum_{g'_{d_\perp} > 0} g'_{d_\perp}, v2_j^k = \lambda \sum_{g'_{d_\perp} < 0} g'_{d_\perp}, \\ v3_j^k &= \lambda \sum_{g'_{d_L} > 0} g'_{d_L}, v4_j^k = \lambda \sum_{g'_{d_L} < 0} g'_{d_L} \end{aligned} \quad (2.14)$$

其中 $\lambda = f_g(k) f_l(k)$ 为高斯系数。将结果妨碍一起，可以算出特征向量 BD_j 对应的特征描述矩阵：

$$BDM_j = \begin{pmatrix} v1_j^1 & v1_j^2 & \dots & v1_j^n \\ v2_j^1 & v2_j^2 & \dots & v2_j^n \\ v3_j^1 & v3_j^2 & \dots & v3_j^n \\ v4_j^1 & v4_j^2 & \dots & v4_j^n \end{pmatrix} \in \mathbb{R}^{4 \times n} \quad (2.15)$$

其中 n 是 B_j 中行的个数 $n = \begin{cases} 2w, j=1 \text{ or } m \\ 3w, \text{else} \end{cases}$ 。 BD_j 由 BDM_j 矩阵的均值向量 M_j 和

方差向量 S_j 求得：

$$LBD = (M_1^T, S_1^T, M_2^T, S_2^T, \dots, M_m^T, S_m^T) \in \mathbb{R}^{8m} \quad (2.16)$$

2.3.3 3D 线段的参数化

通过线特征对相机位置进行非线性优化的时候，需要对 3D 线段进行参数化，并设定观测方程。采用正确的模型对 3D 线段进行描述，对降低系统的运算复杂度会有积极的作用。在本文的系统里，采用了普吕克坐标^[12]的方法对 3D 线段进行参数化。

假设直线上存在任意两点 $\{x_1, x_2\} \in \mathbb{R}^3$ ，齐次坐标分别表示为

$A_1 = (x_1, y_1, z_1, w_1)$ 、 $B_2 = (x_2, y_2, z_2, w_2)$ 。定义该直线的普吕克坐标为一个六维

向量 $L = \begin{bmatrix} n \\ v \end{bmatrix}$ ，如图 2-6 所示

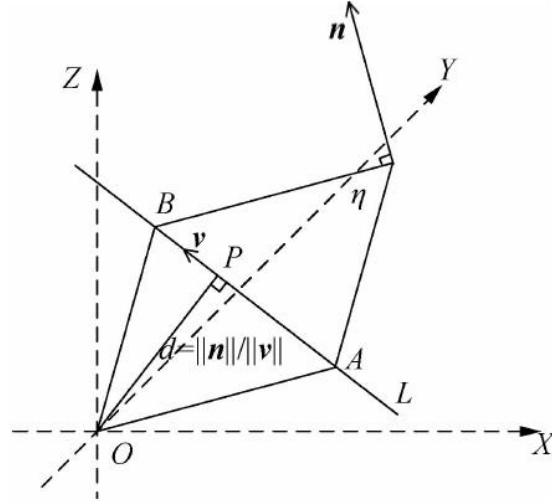


图 2-6 线段的普吕克坐标

其中，世界坐标系原点为 O ； v 是直线的方向向量，方向由 A 指向 B ； n 为该 3D 直线和坐标系原点所构平面的矩向量，方向垂直于该平面，即 $n^T v = 0$ ，幅值为 OAB 三角形面积的二倍。可以用下式进行计算：

$$\begin{cases} v = \overline{OB} - \overline{OA} \\ n = \overline{OA} \times \overline{OB} = \overline{OA} \times v = \overline{OB} \times v \end{cases} \quad (2.17)$$

对于三维直线的普吕克坐标，满足如下性质：

- (1) 满足普吕克约束 $v \perp n$ ；
- (2) 直线到空间原点的距离为 $d = \frac{\|n\|}{\|v\|}$ 。

与空间点坐标类似，我们设 L_c 为相机坐标系下直线的空间坐标， L_w 为世界坐标系下直线的空间坐标，二者之间的坐标转换关系为：

$$L_w = H L_c \triangleq \begin{bmatrix} R & [t]_{\times} R \\ 0 & R \end{bmatrix} L_c \quad (2.18)$$

$$L_c = H^{-1} L_w \sim \begin{bmatrix} R^T & -R^T [t]_{\times} \\ 0 & R^T \end{bmatrix} L_w \quad (2.19)$$

式中， R 为相机位姿中的旋转矩阵，表示相机坐标系方向； t 为相机光心的世界坐标系坐标； H 为变换矩阵。

将 3D 直线 L_c 投影到相机成像平面得到直线 l 。经过投影变换，直线的普吕克

坐标中仅 n_c 是可测的, v_c 中包含了直线的深度和方向信息, 均不可知。投影关系表示如下:

$$l = \kappa R^T (n_w - t \times v_w) = \kappa n_c$$

$$\kappa = \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & f_u c_v & f_u f_v \end{bmatrix} \quad (2.20)$$

由 (2.17) 可知, 经过投影变换, 空间直线的普吕克坐标中仅仅 n_c 可以被观测, 而 v_c 包含直线的深度和方向信息, 无法被观测到。

2.3.4 3D 直线位置初始化

对于单目视觉里程计系统, 由于传感器的限制, 无法从一副图像中获得特征的完整位置信息。但对于成像平面上被检测到的线特征, 系统会首先对其位置进行初始化, 这里我们采用 Sola^[22] 提出的基于非延迟的初始化方法, 步骤如下:

Step1: 从图像中检测线特征 $z = (p_1, p_2)$, p_1 、 p_2 表示线特征两个端点在成向平面上的像素坐标 $p_1(p_{1u}, p_{1v})$ 、 $p_2(p_{2u}, p_{2v})$, 并计算线段在图像坐标系下的普吕克坐标 $l = \{l_1, l_2, l_3\}$, 记为

$$l = \underline{p_1} \times \underline{p_2} \quad (2.21)$$

其中 $\underline{p_1}$ 、 $\underline{p_2}$ 是两个端点的齐次坐标。

Step2: 对 l 反投影变换, 计算其在相机坐标系下的普吕克坐标 L_c 。在上一小节提到, 直线的普吕克坐标中仅 n_c 是可测的, 所以表示为:

$$n_c = K^{-1}l \quad (2.22)$$

Step3: 将 n_c 单位化, 记作 $\hat{n}_c = \frac{n_c}{\|n_c\|}$ 。由直线投影关系可知, 3D 空间直线 L 必定处于相机光心和投影线段 l 所构成的平面上。构造位于该平面上的正交基 $\varepsilon = \{\xi_1, \xi_2\}$, 则:

$$\mathbf{v}_c = \beta_1 \xi_1 + \beta_2 \xi_2 \quad \beta_1, \beta_2 \in R \quad (2.23)$$

$$\xi_1 = \frac{[\hat{n}_y \quad \hat{n}_x \quad 0]}{\sqrt{(\hat{n}_x)^2 + (\hat{n}_y)^2}}, \quad \xi_2 = \frac{\hat{n}_c}{\|\hat{n}_c\|} \times \xi_1 \quad (2.24)$$

式中, β_1 、 β_2 对应 ε 的基向量方向。对于任意的 3D 直线, 都有可以找到相应的参数 β_1 、 β_2 与之对应, 且都满足普吕克约束 $\mathbf{v} \perp \mathbf{n}$ 。将式 (2.12)、(2.23)、(2.24)

代入 $d = \frac{\|\hat{n}_c\|}{v_c} = \frac{1}{\|\beta\|}$, 这样可以求出普吕克坐标 L_c 的方向向量 $\mathbf{v}_c$ 。

Step4: 根据式 (2.15), 将成像平面限度的坐标由相机坐标系 L_c 转换到世界坐标系 L_w 。

2.4 本章小结

本章介绍了视觉里程计系统需要使用的基础知识。首先介绍了针孔相机模型, 对视觉里程计系统的需要涉及的坐标转换关系进行了描述。为了保证空间点可以精确投影到相机成像平面, 通过相机标定引入了反映相机实际状况的相机内参。最后介绍了图像中的线特征和世界坐标系下的空间直线, 从线特征的提取、匹配、空间直线的参数化方法和观测模型、空间直线的初始化设定角度, 对视觉里程计系统需要用到的线特征进行了介绍。

第三章 视觉里程计算法理论

3.1 视觉里程计问题的建模

视觉里程计通过相邻帧图像估计机器人的运动，并恢复场景的空间结构。因为相机通常安装在移动机器人上固定位置，所以对机器人运动的估计即为对相机位姿的估计。设相机图像序列为 $I_{0:n} = \{I_0, I_1, \dots, I_n\}$ 。相邻两帧之间，相机位姿通过一次旋转 $R_{k,k-1}$ 和平移变化 $t_{k,k-1}$ 进行变换。设相邻两帧 I_{k-1} 、 I_k 之间的旋转矩阵为 $R_{k,k-1}$ ，平移向量 $t_{k,k-1}$ 构成变换矩阵：

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} \quad (3.1)$$

$$C_k = C_{k-1} T_{k,k-1}$$

其中 C_k 、 C_{k-1} 对应不同时刻的相机位姿。因此，视觉里程计问题就变为了从图像中获取旋转矩阵和平移向量。

机器人的定位模型如图 3-1 所示：

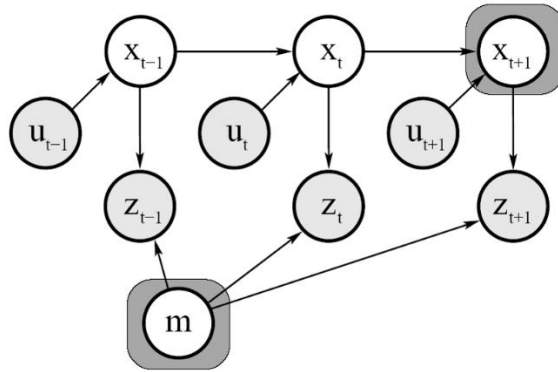


图 3-1 机器人定位模型

其中 x_t 表示机器人在不同时刻的位姿， u_t 表示在 t 时刻机器人收到的控制信息， z_t 表示运动到 x_t 是机器人产生的观测数据， m 表示环境中的路标点。这一模型可以用式 3.1 表示：

$$\begin{cases} x_t = f(x_{t-1}, u_t) + w_t \\ z_{t,j} = h(m_j, x_t) + v_{t,j} \end{cases} \quad k=1, \dots, N, j=1, \dots, M \quad (3.2)$$

两个式子分别机器人的运动方程和观测方程。需要注意的是，在观测方程中，只有当 x_t 观测到 m_j 的时候，才会产生观测数据。对于视觉里程计来说，由于只有相机传感器来获取信息，而没有测量运动的装置。这样导致了我們只能获得观测方程，而没有运动方程。这样整个优化问题就只由多个观测方程组成。

对于相机投影模型，我们已经在第二章的针孔相机模型中做过详细的推导，用下方的流程图表示：

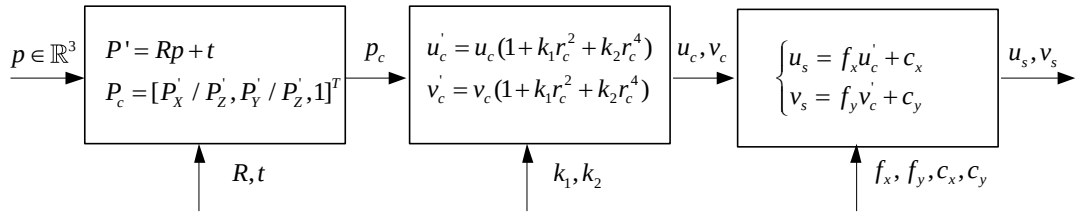


图 3-2 投影模型计算流程图

中间的畸变模块中 $r_c^2 = u_c^2 + v_c^2$ 。

对于观测方程，我们可以将其抽象的记为：

$$z = h(x, y) \quad (3.3)$$

这里的 x 指相机的位置，即外参 R, t ，对应的李代数为 ξ 。路标点即这里的三维空间点 p ，而观测数据即为像素坐标 $z \triangleq [u_s \quad v_s]^T$ 。从最小二乘角度，可以将观测的误差写为：

$$e = z - h(\xi, p) \quad (3.4)$$

同时考虑其他时刻的测量，设 z_{ij} 为在位姿 ξ_i 处观测路标 p_j 产生的数据，则整体的代价函数为：

$$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \|e_{ij}\|^2 = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \|z_{ij} - h(\xi_i, p_j)\|^2 \quad (3.5)$$

对这个最小二乘问题进行求解，即可对相机位姿和路标进行调整。

3.2 基于特征点法的视觉里程计

视觉里程计的实现，根据是否需要提取特征，可以分为特征点法的 VO 和直接法的 VO。其中基于特征点法的视觉里程计，因为特征点的稳定性和对于光照、运动物体的不敏感性，长期以来都是视觉里程计的主流方案。基于特征点法的视觉里程计，其流程如图 3-3 所示：

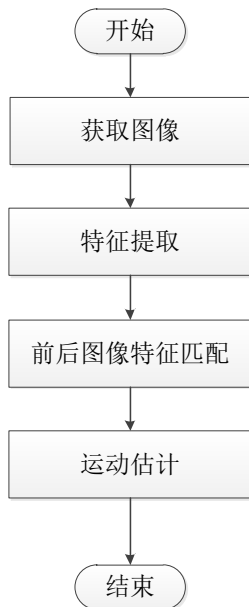


图 3-3 基于特征点法的视觉里程计算法流程

对于特征点法视觉里程计来说，其运动估计依靠不同帧之间特征点的对应关系。三维空间中的坐标点可以在不同的图像帧中被观察到，通过对不同图像帧进行特征匹配或跟踪，可通过重投影方法^[12]求得相机位置。

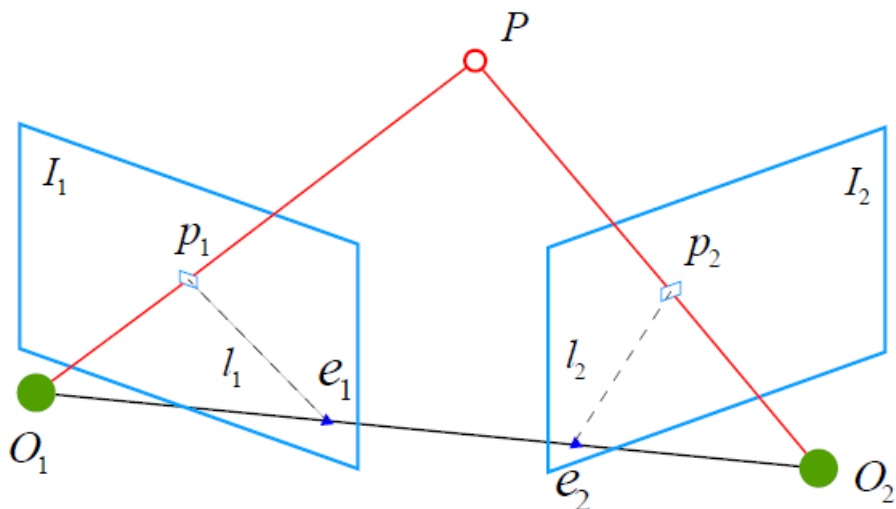


图 3-4 对极几何约束

图 3-4 对极几何约束^[13]描述了两个图像中匹配点的几何关系。两个相机中心设为 O_1 、 O_2 。图像 I_1 中的特征点 p_1 ，在图像 I_2 中有对应的特征点 p_2 ，他们是空间点 P 在两个成像平面上的投影。 O_1 、 O_2 、 P 确定一个平面，称为极平面 (Epipolar plane)。 O_1O_2 连线和向平面 I_1I_2 的角点分别为 e_1 、 e_2 ，称为极点 (Epiholes)， O_1O_2 称为基线 (Baseline)，极平面和两个向平面之间的交线 l_1 、 l_2 为极线 (Epipolar line)。

对于相机的运动估计，可以有线性方法和非线性方法两种求解途径。常见的线性方法包括基于对极约束的方法、PnP(Perspective-n-Point)^[14]和迭代最近点 (Iterative Closest Point, ICP)^[15]求解。通过线性方法求解视觉里程计问题，往往需要先求解相机位置，再求空间点位置。而使用非线性优化的方法，则可以将相机位置和空间点位置都作为优化变量来求解。因为我们使用了单目相机作为传感器，所以在这重点介绍基于对极约束的方法。

3.2.1 对极几何

使用单目相机作为传感器，我们只能获得像素平面上的像素坐标。设图 3-4 中空间点 P 的坐标为 $[X, Y, Z]^T$ 。由第二章针孔相机模型，可以求出两帧图像上像素点 p_1 、 p_2 的像素位置：

$$p_1 = KP, \quad p_2 = K(RP + t) \quad (3.6)$$

K 为相机内参矩阵。

现在，取：

$$x_1 = K^{-1}p_1, \quad x_2 = K^{-1}p_2 \quad (3.7)$$

x_1 、 x_2 是像素点归一化平面坐标。将 3.6 带入 3.5 可得：

$$x_2 = Rx_1 + t \quad (3.8)$$

对等式两边同时左乘 $x_2^T t^\wedge$ ，可以化简为：

$$x_2^T t \times Rx_1 = 0 \quad (3.9)$$

定义 $E = t \times R$ 为本质矩阵 (Essential Matrix)。对 E 进行奇异值分解，就可以获

得相机位移矩阵。

3.2.2 点特征的重投影误差

在对相机位姿进行求解的时候，非线性方法相较于线性方法有更高的处理效率，以及在不同传感器条件下的通用性，已经成为了视觉里程计位置估计的主流方法。在视觉里程计实际运行时，通常的做法是先用对极几何、PnP 解算等线性方法估计相机位姿，将线性方法作为初始值，通过构建最小化重投影误差（Reprojection error）对其进行调整。

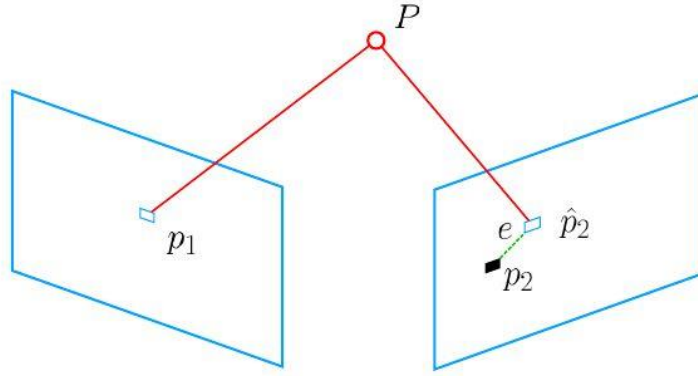


图 3-5 重投影误差示意图

对于空间点 P 及其投影点 p_1 、 p_2 ，需要计算相机的外参 T ，也就是相机的位姿 R ， t ，他的李代数表示为 ξ 。设空间点坐标为 $P_i = [X_i, Y_i, Z_i]^T$ ，对应投影的像素坐标 $p_i = [u_i, v_i]^T$ 。二者关系如下所示：

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = K \exp(\xi^\wedge) \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (3.10)$$

矩阵形式为：

$$s_i p_i = K \exp(\xi^\wedge) P_i \quad (3.11)$$

但是由于相机位姿不确定以及观测点噪声的影响，使得像素坐标 p_2 （观测到的投影位置）和空间点按照估计位姿投影得到的位置 $\hat{p}_2$ 存在误差 e ，如图 3-5 所

示，这个误差称为重投影误差。将其构建最小二乘问题，通过寻找最好的相机位姿，使重投影误差误差最小化：

$$\xi^* = \arg \min_{\xi} \frac{1}{2} \sum_{i=1}^n \left\| u_i - \frac{1}{s_i} K \exp(\xi^{\wedge}) P_i \right\|_2^2 \quad (3.12)$$

在通过高斯牛顿法^[16]等方法求解该问题之前，我们需要对重投影误差线性化处理：

$$e(x + \Delta x) \approx e(x) + J \Delta x \quad (3.13)$$

对于定位问题来说， e 是像素位置的误差，为 2 维向量； x 是相机位姿，为 6 维向量。所以 J 会是一个 2×6 的矩阵。记相机坐标系下空间点坐标为 P' ，

$P' = (\exp(\xi^{\wedge}) P)_{1:3} = [X' \ Y' \ Z']^T$ 。由第二章针孔相机模型可以，相机投影点坐标 P 坐标为 (u, v) ：

$$\begin{cases} u = f_x \frac{X'}{Z'} + c_x \\ v = f_y \frac{Y'}{Z'} + c_y \end{cases} \quad (3.14)$$

我们对 $\xi^{\wedge}$ 左乘扰动量 $\Delta \xi$ ，计算 e 的变化关于扰动量的导数：

$$\frac{\partial e}{\partial \delta \xi} = \lim_{\delta \xi \rightarrow 0} \frac{e(\delta \xi \oplus \xi)}{\delta \xi} = \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial \delta \xi} \quad (3.15)$$

其中 $\oplus$ 表示李代数上的左乘扰动。3.14 中第一项为误差关于投影点的导数，可由 3.13 得出：

$$\frac{\partial e}{\partial P'} = - \begin{bmatrix} \frac{\partial u}{\partial X'} & \frac{\partial u}{\partial Y'} & \frac{\partial u}{\partial Z'} \\ \frac{\partial v}{\partial X'} & \frac{\partial v}{\partial Y'} & \frac{\partial v}{\partial Z'} \end{bmatrix} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \quad (3.16)$$

式 3.14 中后一项作为变换后的点关于李代数的导数，为：

$$\frac{\partial P'}{\partial \delta \xi} = [I, -P'^{\wedge}] \quad (3.17)$$

将(3.16)和(3.17)相乘，可以得出 2×6 的雅可比矩阵：

$$\frac{\partial e}{\partial \delta \xi} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X^2}{Z'^2} & -\frac{f_x Y'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_x - \frac{f_y Y'^2}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & -\frac{f_y X'}{Z'^2} \end{bmatrix} \quad (3.18)$$

式 3.17 即为描述空间点的重投影误差关于相机位置李代数的一阶变化关系的雅可比矩阵^{[24],[25]}。它在优化过程中提供了梯度方向，指导优化的迭代。

3.2.3 线特征的重投影误差

常见的三维空间直线观测模型有多种方式，常见的包括 3D-3D 的方式和 3D-2D 的方式。3D-3D 方法主要计算的是互相匹配的空间直线之间的误差；3D-2D 方式则计算的是三维空间直线和与其匹配的投影直线的误差。相比较而言，3D-2D 的方式有着更高的精度^[23]。

与点特征的处理类似，3D-2D 的方法也是首先将世界坐标系下直线的普吕克坐标 L_w 变换为相机坐标系下坐标 L_c ，然后根据相机模型计算 L_c 在成像平面上的投影 l' ，最后匹配线段 z 的端点 x_s 、 x_e 到投影直线 l' 的距离^[26]。但是与特征点不同的是，线特征的重投影误差不再是简单的投影预测值和测量值之差，而是将误差模型定义 $e = [x_s^T l', x_e^T l']$ ，其中 x_s 、 x_e 是匹配线段 z 的两个端点。误差模型可以通过下式表示：

$$e = d(z, l') = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} x_s^T \cdot l' / \sqrt{l_1^2 + l_2^2} \\ x_e^T \cdot l' / \sqrt{l_1^2 + l_2^2} \end{bmatrix} \quad (3.19)$$

式中， $l' = \{l_1, l_2, l_3\}^T$ ，可以通过第二章式 (2.20) 计算。在实际测量过程中，受孔径问题以及图像中直线端点的不确定性影响，只有匹配点线段 z 垂直投影投影直线这部分才能提供有效信息，如下图 3-6 所示。

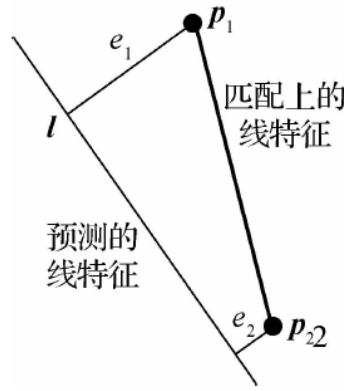


图 3-6 空间直线的误差定义

通过非线性方法对线特征的误差函数求解的时候,通常都是在估计点的附近,对误差函数线性展开,求解误差函数关于状态变量的雅可比矩阵^[34]。对于空间直线的重投影误差 $e = d(z, l')$, 设 z 的两个端点 $x_s = [u_1 \ v_1 \ 1]^T$ 、 $x_e = [u_2 \ v_2 \ 1]^T$, 误差对投影直线 $l' = [l_1 \ l_2 \ l_3]$ 的雅可比矩阵为

$$\frac{\partial e}{\partial l'} = \begin{bmatrix} \frac{u_1 l_2^2 - l_1 l_2 v_1 - l_1 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{v_1 l_1^2 - l_1 l_2 u_1 - l_2 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{1}{\sqrt{l_1^2 + l_2^2}} \\ \frac{u_2 l_2^2 - l_1 l_2 v_2 - l_1 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{v_2 l_1^2 - l_1 l_2 u_2 - l_2 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{1}{\sqrt{l_1^2 + l_2^2}} \end{bmatrix} \quad (3.20)$$

其中, $e_1 = x_s^T l'$, $e_2 = x_e^T l'$, $l_n = \sqrt{l_1^2 + l_2^2}$ 。

$$\frac{u_1 l_2^2 - l_1 l_2 v_1 - l_1 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} = \frac{(u_1 l_2^2 + u_1 l_1^2) - (u_1 l_1^2 + l_1 l_2 v_1 + l_1 l_3)}{(l_1^2 + l_2^2)^{\frac{3}{2}}} = \frac{1}{l_n} \left(u_1 - \frac{l_1 e_1}{l_n^2} \right) \quad (3.21)$$

所以

$$\frac{\partial e}{\partial l'} = \frac{1}{l_n} \begin{bmatrix} u_1 - \frac{l_1 e_1}{l_n^2} & v_1 - \frac{l_2 e_1}{l_n^2} & 1 \\ u_2 - \frac{l_1 e_2}{l_n^2} & v_2 - \frac{l_2 e_2}{l_n^2} & 1 \end{bmatrix}_{2 \times 3} \quad (3.22)$$

对于相机坐标系下直线的投影方程 $l' = \kappa n_c$, 投影直线 l' 对普吕克坐标 L_c 的雅可比矩阵为 $\frac{\partial l'}{\partial L_c} = \frac{\partial \kappa n_c}{\partial L_c} = [\kappa \ 0]_{3 \times 6}$ 。对于坐标转换 $L_c = H_{cw} L_w$, 分别对位置增量 δ_ξ

和空间直线正交表示增量 δ_θ 求雅可比矩阵, 令 L_w 的正交表示为

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix}, \quad W = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix}.$$

$$\frac{\partial L_c}{\partial L_w} = \frac{\partial H_{cw} L_w}{\partial L_w} = H_{cw} \quad (3.23)$$

$$\frac{\partial L_w}{\partial \delta_\theta} = \begin{bmatrix} -[w_1 u_1]_\times & -w_2 u_1 \\ -[w_2 u_2]_\times & -w_1 u_2 \end{bmatrix}_{6 \times 4} \quad (3.24)$$

对于 $\frac{\partial L_c}{\partial \delta_\xi}$ ，将其分为对位移部分 δ_p 和旋转部分 δ_θ 分别求解。首先令旋转部分

δ_θ 为 0， $\delta_\xi = \begin{bmatrix} 0 \\ \delta_p \end{bmatrix}$ ，先求更新后的空间直线 L_c^*

$$\begin{aligned} T^* &= \exp(\delta_\xi^\wedge) T_{cw} \approx \begin{bmatrix} I & \delta_p \\ 0^T & 1 \end{bmatrix} T_{cw} \\ R^* &= R_{cw}, \quad t^* = \delta_p + t_{cw} \\ H_{cw}^* &= \begin{bmatrix} R_{cw} & [\delta_p + t_{cw}]_\times R_{cw} \\ 0 & R_{cw} \end{bmatrix} \\ L_c^* &= H_{cw}^* L_w = \begin{bmatrix} R_{cw} n_w + [\delta_p + t_{cw}]_\times R_{cw} v_w \\ R_{cw} v_w \end{bmatrix} \end{aligned} \quad (3.25)$$

再对 δ_p 求偏导，移除不相关项：

$$\begin{aligned} \frac{\partial L_c^*}{\partial \delta_p} &= \begin{bmatrix} \frac{[\delta_p + t_{cw}]_\times R_{cw} v_w}{\partial \delta_p} \\ 0 \end{bmatrix} = \begin{bmatrix} -\frac{[R_{cw} v_w]_\times \delta_p}{\partial \delta_p} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -[R_{cw} v_w]_\times \\ 0 \end{bmatrix}_{6 \times 3} \end{aligned} \quad (3.26)$$

其次对于旋转 δ_θ ，令位移部分 $\delta_p = 0$ ，则 $\delta_\xi = \begin{bmatrix} \delta_\theta \\ 0 \end{bmatrix}$ 。同样先求出更新后的空间直线

L_c^* ，这里去掉了 R 和 t 的下标，让式子简明一些：

$$\begin{aligned}
T^* &= \exp(\delta_\xi^\wedge) T_{cw} \approx \left(I + \begin{bmatrix} [\delta_\phi]_\times & 0 \\ 0^T & 1 \end{bmatrix} \right) T_{cw} \\
R^* &= (I + [\delta_\phi]_\times) R_{cw}, \quad t^* = (I + [\delta_\phi]_\times) t_{cw} \\
H_{cw}^* &= \begin{bmatrix} (I + [\delta_\phi]_\times) R & (I + [\delta_\phi]_\times) [t]_\times R \\ 0 & (I + [\delta_\phi]_\times) R \end{bmatrix} \\
L_c^* &= H_{cw}^* L_w = \begin{bmatrix} (I + [\delta_\phi]_\times) R n + (I + [\delta_\phi]_\times) [t]_\times R v \\ (I + [\delta_\phi]_\times) R v \end{bmatrix}
\end{aligned} \tag{3.27}$$

再对 δ_ϕ 求偏导，移除不相关的项可得

$$\begin{aligned}
\frac{\partial L_c^*}{\partial \delta_\phi} &= \begin{bmatrix} \frac{\partial [\delta_\phi]_\times R n}{\partial \delta_\phi} + \frac{\partial [\delta_\phi]_\times [t]_\times R v}{\partial \delta_\phi} \\ \frac{\partial [\delta_\phi]_\times [t]_\times R v}{\partial \delta_\phi} \end{bmatrix} = \begin{bmatrix} -\frac{[R n]_\times \delta_\phi}{\partial \delta_\phi} - \frac{\partial [t]_\times R v}{\partial \delta_\phi} \delta_\phi \\ -\frac{[R v]_\times \delta_\phi}{\partial \delta_\phi} \end{bmatrix} \\
&= \begin{bmatrix} -[R n]_\times - [t]_\times R v \\ -[R v]_\times \end{bmatrix}_{6 \times 3}
\end{aligned} \tag{3.28}$$

因此

$$\frac{\partial L_c}{\partial \delta_\xi} = \frac{\partial L_c^*}{\partial \delta_\xi} = \begin{bmatrix} -[R_{cw} n_w]_\times - [t_{cw}]_\times R_{cw} v_w & -[R_{cw} v_w]_\times \\ -[R_{cw} v_w]_\times & 0 \end{bmatrix}_{6 \times 6} \tag{3.29}$$

最后，通过链式法则求得重投影误差 e_l 关于位姿和空间点的雅可比矩阵^[26]

$$Jl_\xi = \frac{\partial e_l}{\partial \delta_\xi} = \frac{\partial e_l}{\partial l'} \frac{\partial l'}{\partial L_c} \frac{\partial L_c}{\partial \delta_\xi} \tag{3.30}$$

$$Jl_\theta = \frac{\partial e_l}{\partial \delta_\theta} = \frac{\partial e_l}{\partial l'} \frac{\partial l'}{\partial L_c} \frac{\partial L_c}{\partial L_w} \frac{\partial L_w}{\partial \delta_\theta} \tag{3.31}$$

3.3 基于直接法的视觉里程计

虽然特征点法在视觉里程计的研究中占了主导地位，但其固有的缺点也制约它的发展。特征点法的实现建立在特征点的提取上，但是特征的提取及其描述子的计算会耗费大量的时间，尤其是在实时系统上，有一大半的时间都消耗在了特征的计算上，极大的限制了系统的效率。由于依靠特征点的匹配来估计相机运动，

特征点法对于图像上特征点外的信息都进行了忽略。这对图像信息是极大的浪费。而且最为关键的是，并非所有的图像都能提取足够的特征点。一旦出现特征点数量不足，就会使得基于特征点法的视觉里程计失效。

直接法视觉里程计是直接利用像素点灰度信息来构图与定位。直接法的视觉里程计一定程度上克服了特征点法的视觉里程计的缺点，它能使用图像上所有像素点信息。正因为如此，在特征缺失的环境下，直接法视觉里程计仍然可以有很高的精度和鲁棒性。

3.3.1 点特征的光度误差

设空间点 P 的世界坐标为 $[X, Y, Z]$ ，他在相邻两个时刻的相机上成像，记其非齐次像素坐标为 p_1 、 p_2 。设第一帧相机的旋转、平移为 I 、 0 。第二个相机相对第一个发生平移和旋转。记第二个相机的外参为 R 、 t （李代数为 ξ ）。两个相机具有相同的内参 K 。完整的投影方程为：

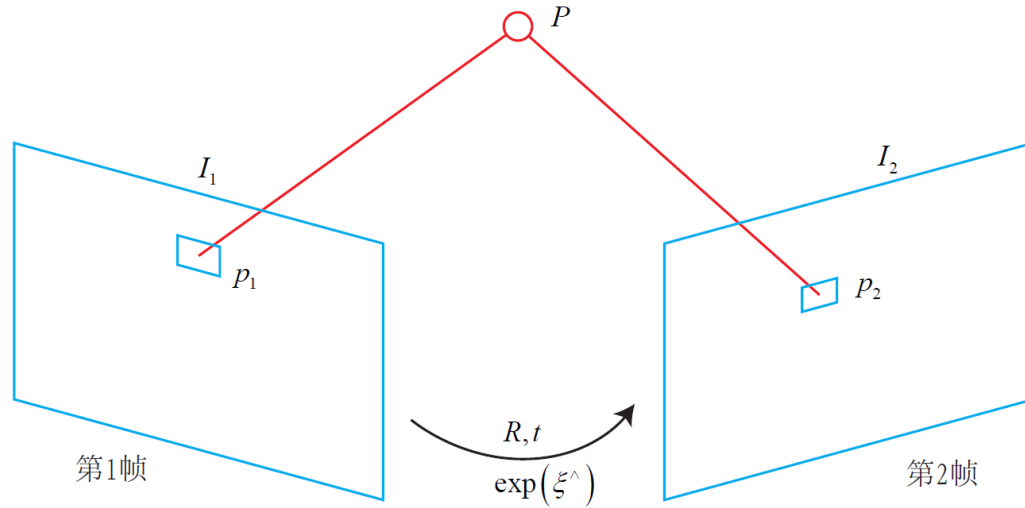


图 3-7 直接法示意图

$$p_1 = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_1 = \frac{1}{Z_1} KP$$

$$p_2 = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_2 = \frac{1}{Z_2} K(RP + t) = \frac{1}{Z_2} K(\exp(\xi^\wedge)P)_{1:3} \quad (3.32)$$

其中 Z_1 、 Z_2 分别表示两个相机坐标系下的深度。

在特征点法的视觉里程计中，由于对描述子进行匹配，我们可以得到 p_1 、 p_2 的对应的像素位置，在此基础上进行重投影误差的计算。但是在直接法的实现过程中，并未对特征点进行提取，也就无法获得匹配点的像素坐标。这里，直接法基于灰度不变假设，通过当前相机的位姿来寻找像素点 p_1 的匹配点 p_2 。灰度不变假设认为，同一个空间点的像素灰度，在各个图像中是固定不变的。如果相机位姿不够好，那么 p_2 和 p_1 之间就会有明显的光度误差（Photometric Error）。为了减小光度误差，直接法对相机位姿进行优化来寻找更相似的匹配。设两个像素点的光度误差为：

$$e = I_1(p_1) - I_2(p_2) \quad (3.33)$$

实际处理过程中，可以同时获得 N 个空间点，相机的位置估计问题相应变化为：

$$\min_{\xi} J(\xi) = \sum_{i=1}^n e_i^T e_i, \quad e_i = I_1(p_{1,i}) - I_2(p_{2,i}) \quad (3.34)$$

求解这个问题的过程中，我们需要优化相机的位姿 ξ ，使得误差 e 最小。为了分析导数关系，我们使用李代数扰动模型。给 $\exp(\xi)$ 左乘一个扰动 $\exp(\delta\xi)$ ，得：

$$\begin{aligned} e(\xi \oplus \delta\xi) &= I_1\left(\frac{1}{Z_1} KP\right) - I_2\left(\frac{1}{Z_2} K \exp(\delta\xi^\wedge) \exp(\xi^\wedge) P\right) \\ &\approx I_1\left(\frac{1}{Z_1} KP\right) - I_2\left(\frac{1}{Z_2} K(1 + \delta\xi^\wedge) \exp(\xi^\wedge) P\right) \\ &= I_1\left(\frac{1}{Z_1} KP\right) - I_2\left(\frac{1}{Z_2} K \exp(\xi^\wedge) P + \frac{1}{Z_2} K \delta\xi^\wedge \exp(\xi^\wedge) P\right) \end{aligned} \quad (3.35)$$

记：

$$\begin{aligned}
q &= \delta \xi^\wedge \exp(\xi^\wedge) P \\
u &= \frac{1}{Z_2} Kq
\end{aligned} \tag{3.36}$$

这里 q 为扰动分量在第二个相机坐标系下的坐标。 u 为像素坐标。对式 (3.22) 进行泰勒展开, 有

$$\begin{aligned}
e(\xi \oplus \delta \xi) &= I_1 \left(\frac{1}{Z_1} KP \right) - I_2 \left(\frac{1}{Z_2} K \exp(\xi^\wedge) P + u \right) \\
&\approx I_1 \left(\frac{1}{Z_1} KP \right) - I_2 \left(\frac{1}{Z_2} K \exp(\xi^\wedge) P \right) - \frac{\partial I_2}{\partial u} \frac{\partial u}{\partial q} \frac{\partial q}{\partial \delta \xi} \delta \xi \\
&= e(\xi) - \frac{\partial I_2}{\partial u} \frac{\partial u}{\partial q} \frac{\partial q}{\partial \delta \xi} \delta \xi
\end{aligned} \tag{3.37}$$

式 (3.23) 中, $\frac{\partial I_2}{\partial u}$ 为 u 处的像素梯度, $\frac{\partial u}{\partial q}$ 是投影方程关于相机坐标系下的

三维点的导数, $\frac{\partial q}{\partial \delta \xi}$ 则为变换后的三维点对应的导数:

$$\begin{aligned}
\frac{\partial u}{\partial q} &= \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \end{bmatrix} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} \end{bmatrix} \\
\frac{\partial q}{\partial \delta \xi} &= [I, -q^\wedge]
\end{aligned} \tag{3.38}$$

由于 $\frac{\partial u}{\partial q}$ 、 $\frac{\partial q}{\partial \delta \xi}$ 只于三维空间点 P 有关, 而与投影平面无关, 我们常把它合并

在一起:

$$\frac{\partial u}{\partial \delta \xi} = - \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} & -\frac{f_x XY}{Z^2} & f_x + \frac{f_x X^2}{Z^2} & -\frac{f_x Y}{Z^2} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} & -f_y - \frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X'}{Z^2} \end{bmatrix} \tag{3.39}$$

由此, 我们推导出光度误差相对于李代数的雅可比矩阵:

$$J = - \frac{\partial I_2}{\partial u} \frac{\partial u}{\partial \delta \xi} \tag{3.40}$$

在获得雅可比矩阵的基础上, 采用高斯牛顿法计算增量, 迭代求导, 实现相机的位置估计。

3.3.2 线特征的光度误差分析

对于线特征的光度误差，和点特征类似。这里，我们通过下式定义图像上的线特征 L ，它的端点 p 、 q 深度值在 $k-1$ 时刻已经获知：

$$L = \{p, q, w_n \mid p, q \in L_{k-1} \wedge \pi(\exp(\xi^\wedge) \bullet \pi^{-1}(p, d_p)) \in t_k \wedge \pi(\exp(\xi^\wedge) \bullet \pi^{-1}(q, d_q)) \in t_k\} \quad (3.41)$$

其中， p 、 q 分别表示线特征两个端点， w_n 表示对线段 L 进行 n 等分， d_p 、 d_q 表示点 p 、 q 的深度。

对于线段 δI_l 的光度误差，就可以通过线段上两端段之间等分点的光度误差来计算：

$$\delta I_l(\xi) = \frac{1}{N_l} \sum_{n=0}^{N_l} |I_k(\pi(\exp(\xi^\wedge) \bullet w_n)) - I_{k-1}(w_n)| \quad (3.42)$$

通过这种方法，我们将线段的光度误差通过点的形式进行了求解。

3.4 本章小结

本章介绍了视觉里程计的数学模型，探讨了特征点法的视觉里程计和直接法的视觉里程计两种不同方法的特点，描述了两种视觉里程计设计方法的算法思想和流程进行了介绍，并对点、线特征在两种方法中的非线性优化问题进行了一定的推导。

第四章 基于点线特征的视觉里程计

4.1 系统框架

本文设计的基于点线特征的视觉里程计，是依靠对三维空间中的点特征和线特征的跟踪处理来实现相机的位姿估计。该系统是在半直接法视觉里程计 SV0 (Semi-direct visual odometry)^[17]的基础上进行的扩展。SV0 的特点是混合的使用了直接法和特征点法。它提取了图像上的某些关键点，但是并未通过描述子匹配来进行跟踪，而是通过关键点附近的像素信息估计相机的运动。因为不需要计算关键点的描述子，也不需要像稠密直接法那样处理所有的像素信息，所以 SV0 可以实现较快的运行速度。

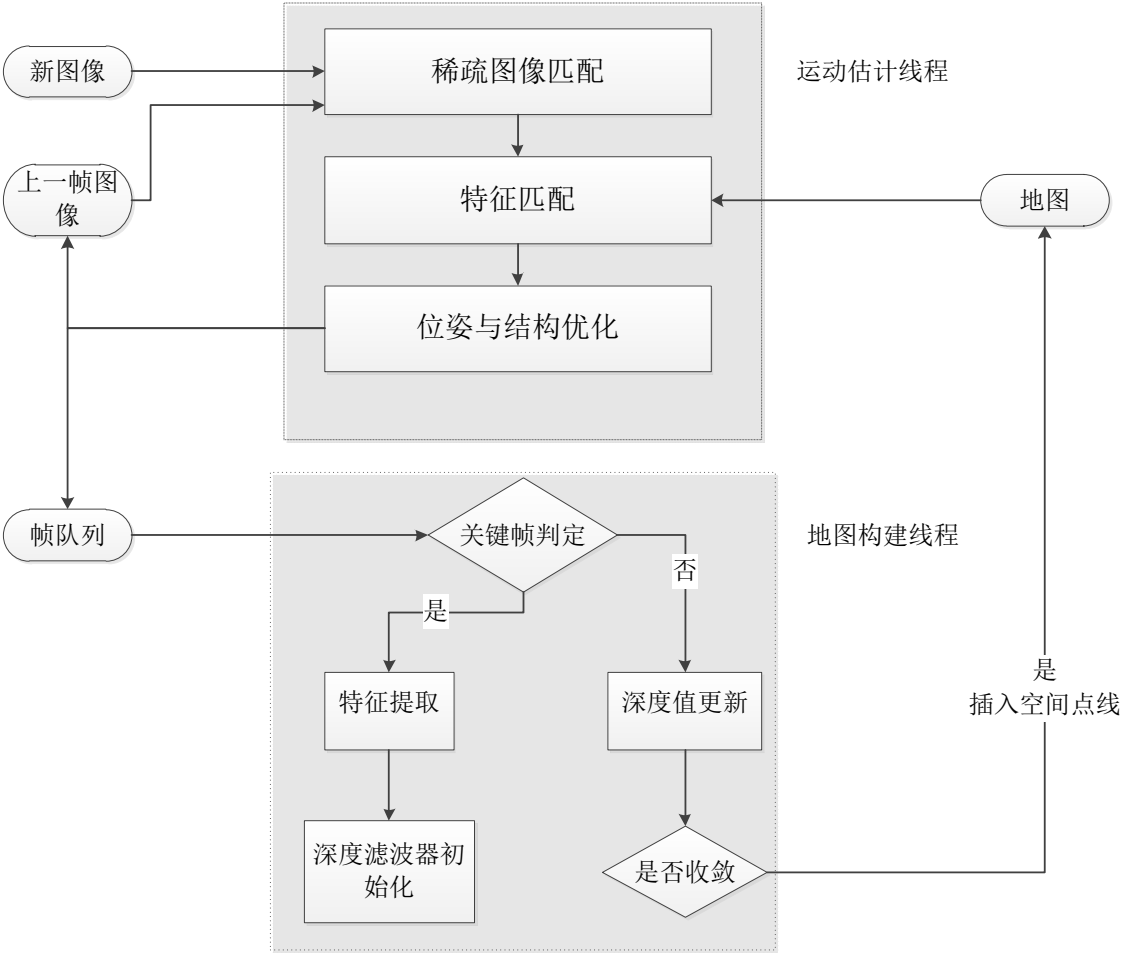


图 4.1 基于点线特征的视觉里程计

本系统是在 SV0 的基础上进行的扩展，沿用了相同的主体结构。系统主要包含两个并行的线程：运动估计线程和地图构建线程。每个线程的主要内容如下：

- 1) 运动估计线程:输入为单目相机获得的图像。运动估计主要包括三个阶段,首先通过相机获取相邻两帧图像来估计相机位姿,然后通过地图模型中存储的三维空间点对之前获得位姿进一步优化并得到空间点在当前帧的预测位置,最后通过重投影计算相机的位姿。
- 2) 地图构建线程:主要功能是计算特征点的深度。输入为经过运动估计线程优化的图像队列。首先判断图像是否为关键帧。如果是关键帧的话,提取特征点作为种子点放入队列;如果不是关键帧的话,就用该图像来个更新种子点的分布。但种子点深度分布收敛,就将该点添加到地图中,供运动估计线程使用。

下面将对两个线程进行详细的描述。

4.2 运动估计线程

运动估计线程,主要执行相机位姿估计功能。它会根据相机获取的相邻时刻图像中重复出现的信息,来估计相机的相对运动。通过积累的位姿变化,计算出相机在三维空间中的移动轨迹。

4.2.1 初始化

读取单目相机获得的第一帧图像,获得其金字塔图像,对这一帧图像进行处理。首先检测该帧图像中 FAST 特征点^[17]和边缘特征的数量。如果该帧图像中可以提取到超过 50 个特征点,就将其设置为第一个关键帧。

在获得第一帧图像后,继续处理下一帧图像,用于和第一帧图像实现三角初始化。通过计算两帧之间相机运动的基础矩阵或者单应矩阵,分解得到相应的 R , t 。这里有一个问题,对于求得的 t 乘以任意非零常数,该分解依旧成立。这就是制约单目视觉系统的尺度不确定问题,需要对尺度进行固定。使用单目相机的时候,需要对其进行一定的平移才可以初始化。如果这个平移过小或过大的话,都会导致单目视觉里程计初始化失败。

在成功初始化之后,将两帧图像都作为关键帧送入到深度滤波器中。深度滤波器在极线上搜索种子点的匹配点并对种子点进行更新,直到其收敛出新的候选

地图点。关于深度滤波器将在之后的章节详细讲解。

4.2.2 稀疏图像匹配

在对单目视觉里程计成功初始化之后，随着相机的移动，图像逐渐输入到运动估计线程。稀疏图像匹配使用直接法，通过最小化图像块光度误差的方法，对相邻两帧图像进行跟踪来粗略估计当前时刻的位置。如图 3-7 所示：其中红色的 $T_{k,k-1}$ 为优化变量。

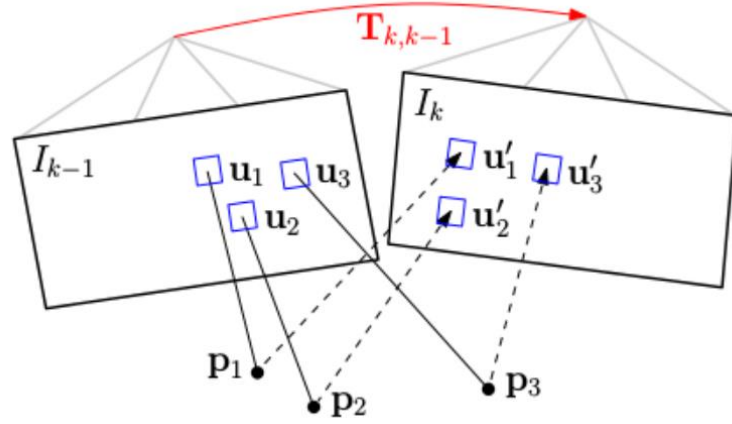


图 4.2 系数图像匹配示意图

稀疏图像匹配的流程如下：

Step1: 假设相邻帧之间位姿 $T_{k,k-1}$ 已知，一般以 $k-1$ 时刻的相机位姿作为初始值。通过 $k-1$ 时刻之前的特征检测和深度估计，已经提前获得图像 I_{k-1} 中的关键点位置和深度信息。

Step2: 知道 I_{k-1} 中某关键点的位置 (u_{k-1}, v_{k-1}) 和深度信息 d_{k-1} 后，我们可以通过反投影的方式在 I_{k-1} 的相机坐标系内获得其空间坐标。要将该空间点投影到 I_k 上，需要通过位姿转换矩阵 $T_{k,k-1}$ 来计算在该空间点在 k 时刻相机坐标系下坐标，然后投影到 I_k 的图像平面 (u_k, v_k) ，完成投影过程。这里的点既可以是单独的像素点，也可以是线特征上的均分点。

Step3: 通过第三章直接法里的描述，我们知道对于空间中的同一个点，在相邻的两帧图像上亮度值是保持不变的。但因为我们将上一时刻的相机位姿作为了

$T_{k,k-1}$ 的值，导致投影的时候产生了误差，光度误差由此产生。通过优化相机的位姿，使得光度误差最小，可以获得优化后的 $T_{k,k-1}$ 。

$$T_{k,k-1} = \arg \min_{T_{k,k-1}} \frac{1}{2} \left\{ \sum_{i \in P} \left\| \delta I_p(T_{k,k-1}, x_i) \right\|_2^2 + \sum_{j \in L} \left\| \delta I_l(T_{k,k-1}, l_j) \right\|_2^2 \right\} \quad (4.1)$$

其中

$$\begin{aligned} \delta I_p(T, x_i) &= I_k(\pi(T \cdot \pi^{-1}(x_i, d_{x_i}))) - I_{k-1}(x_i) \\ \delta I_l(T, l_j) &= \frac{1}{N_l} \sum_{n=0}^{N_l} |I_k(\pi(T \cdot w_n)) - I_{k-1}(w_n)| \end{aligned} \quad (4.2)$$

通过高斯牛顿迭代法，可以求解该问题。在第三章中已有详细推导，在此不再赘述。

通过稀疏图像匹配，我们可以大致估计出相机位置。

4.2.3 特征匹配

虽然通过求解光度误差的方法获得了相机的大致位姿，但是由于稀疏图像匹配是递进式推导相机的位姿，受噪声等种种因素的干扰，误差会逐渐累积并导致出现漂移。所以我们不能通过这种方法来精确求得 k 时刻的相机位姿，通过稀疏图像匹配获得的投影点在 I_k 中的位置也无法与实际位置真正重合，如图 4-3 所示：

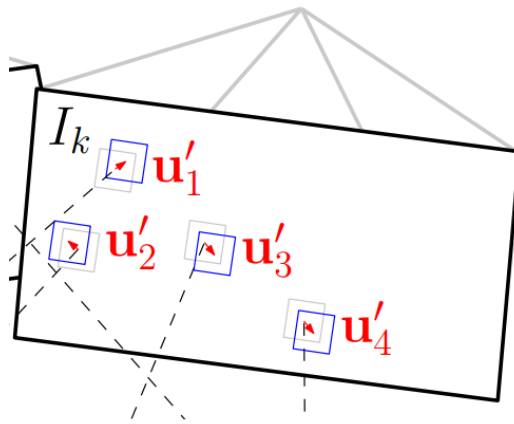
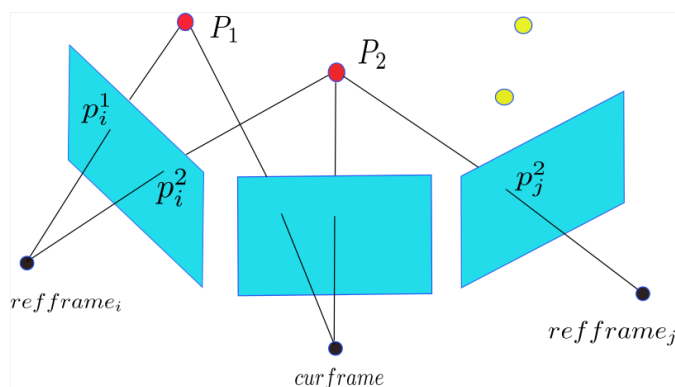


图 4-3 点特征的重投影误差

图中灰色图像块表示真实位置，蓝色为预测位置。二者之间存在偏差。所以在运动估计线程，我们加入了特征匹配来进一步优化。

系统中的地图保存了相机轨迹上所有深度收敛的空间点、线。相机获取的任意一帧图像，都可能包含有一定数量的地图点。这些地图点不仅包含在其所在的关键帧上，也可以在其他帧被观测到，如图 4-4 所示。在处理当前帧的时候，系统会从相近的关键帧中选择共视的地图点



4-4 特征点与地图点的特征匹配

在处理当前帧的时候，系统会从地图中选择相近的关键帧。前提是，这些关键帧上有地图点可以被投影到当前帧。同时，地图中保存的尚未收敛的候选点也要投影到当前帧上。投影到时候依据关键帧和当前帧之间的距离，距离最近的关键帧先投影。对关键帧上的特征点进行遍历，通过特征点找到与其对应的地图点，再将它们一一投影过来。每个地图点只会被投影一次。投影过程如 4-5 所示。这样在当前帧中就可以获得多个投影点。在实际处理的过程中，系统并不会对所有的投影点都进行计算。通过对当前帧进行分块，系统会选择每个区域中质量最好的特征点来进行后续的处理。

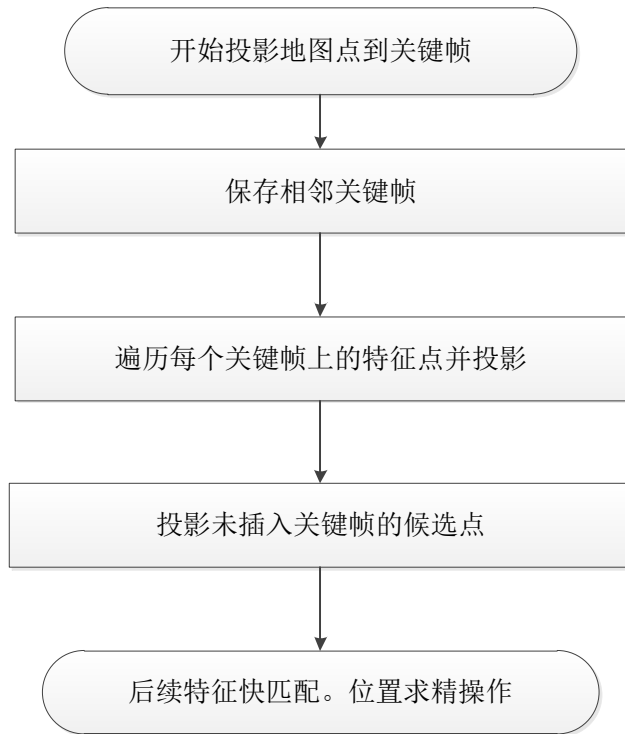


图 4-5 特征匹配流程图

通过稀疏图像匹配处理，我们已经可以获得当前帧的位姿。将这些地图点投影到当前帧，投影位置和真实位置之间会存在一定的偏差。基于灰度不变假设，系统重新构建了一个残差，对特征投影位置进行优化：

$$u_i' = \arg \min_{u_i} \frac{1}{2} \sum_{i \in R} \|I(u_i') - A_i \cdot I_r(u_i)\|_2^2 \quad (4.3)$$

这里，系统不再是对相机位姿进行优化，而是优化的像素位置。公式 4.3 中， $I(u_i')$ 表示当前图像上的像素亮度， $I_r(u_i)$ 则是由投影的空间点或线追溯到其所在关键帧上的像素灰度值。因为两帧图像之间有一定的空间距离，所以在计算偏差的时候，加入了仿射变换^[18]。

通过特征匹配获得的特征点预测位姿，对比经由稀疏图像匹配中相机位姿预测的位置更加的准确。利用这里的出的结果也能进一步去优化相机位姿和特征点、线的三维空间坐标。

4.2.4 位姿及结构优化

在稀疏图像匹配中，系统通过光度误差来优化。通过特征匹配，系统获得了优化后的特征位置，它和最初的预测值存在偏差。所以对于当前帧上所有新的特

征点 p_i 和特征线段 l_j ，如果存在地图点 P 和线段 L 与其对应，通过优化当前帧的相机位置 $T_{k,w}$ ，使得地图点、线段的投影位置和最佳匹配位置的残差最小。

$$T_{k,w} = \arg \min_{T_{k,w}} \frac{1}{2} \left(\sum_{i \in X_c} \rho_p \|ep_i\|_{\Sigma_p}^2 + \sum_{j \in X_c} \rho_l \|el_j\|_{\Sigma_L}^2 \right) \quad (4.4)$$

这里，计算的不再是光度误差，而是变为了像素重投影后的位置偏差，优化变量则是相机位姿。 X_c 则表示点特征和线特征是在相邻帧之间进行的匹配。

所以，运动估计线程的步骤可以总结为：

Step1: 帧间位姿估计；

Step2: 遍历所有地图点，计算在当前图像中位置。由于当前帧有粗略的位置估计，这个投影位置和真实位置间有少量的误差；

Step3: 对每个成功投影的地图点，比较这些点的初始观测图像与当前帧的图像。通过计算光度误差，求取更加精确的投影位置；

Step4: 根据上一步中投影位置，进行位置与地图点优化。

4.3 地图构建线程

地图构建线程的主要任务是完成特征点的深度估计。在对三维空间进行拍摄的时候，单目相机无法通过一帧图像来获得空间中点的深度信息。通用的解决办法是对相机进行平移通过空间点在两帧图像上的共视关系来实现三角化（如图 4-6 所示）来估计这些点的深度值。

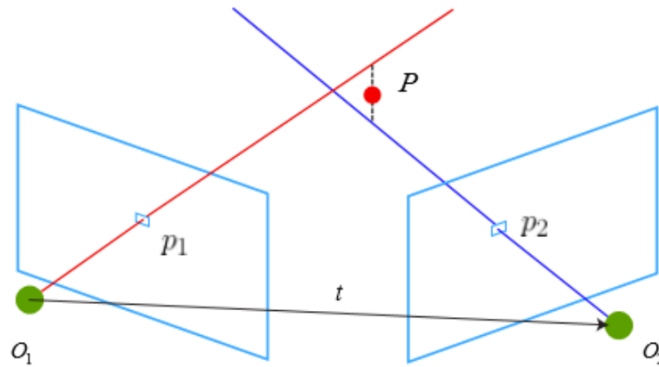


图 4-6 三角化获得空间点深度

在地图构建线程中，将尚未确定准确深度信息的点称为候选点，深度值已经

收敛的则称为地图点。所有的候选点和地图点都从系统判定的关键帧上选取。所以当图像序列输入地图构建线程的时候，首先对其进行是否为关键帧的辨别。

在相机读取到一帧新的图像时，系统会计算着该帧图像和相邻关键帧之间的平移量。如果超过当前场景深度平均值的 12%，那么这一帧图像就会被选为关键帧插入到地图中。对于关键帧，系统会检测上边的特征点和线段，将它们作为深度估计的种子。这些种子会不断融合新的关键帧来进行深度估计。系统的地图模型保存的就是提取出来的关键帧以及还未插入地图的关键帧上已经收敛的空间点、线坐标。

4.3.1 特征的选取

对于关键帧，地图构建线程会从中提取需要的特征点、线。这里我们以点特征为例，来介绍特征的选取。

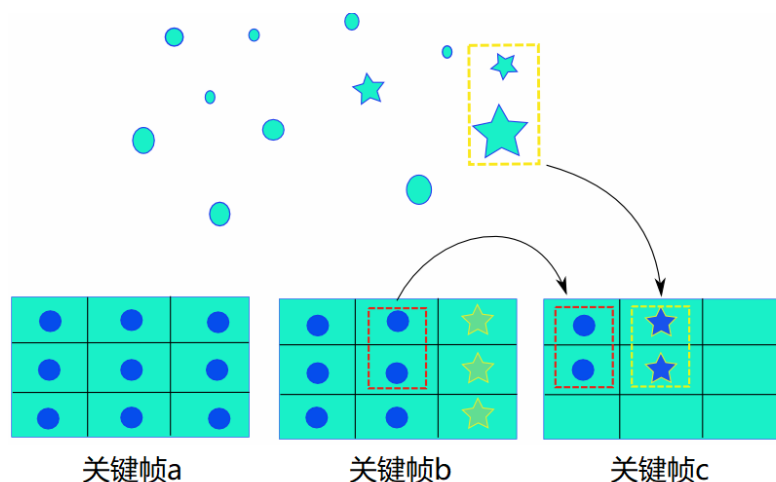


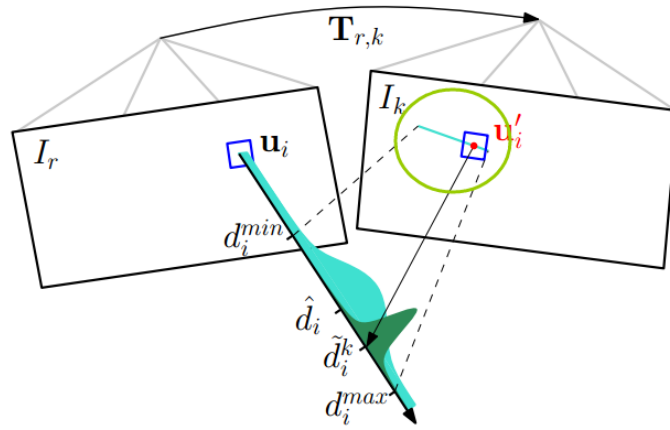
图 4-7 候选点选择

如图 4-7 所示，有三个关键帧 a、b、c。对于关键帧 a 来说，每个格子中都可以检测到圆形特征点，并且深度值都已经收敛，作为地图点保存了下来。相机运动提取到关键帧 b，关键帧 a 中的地图点只有 6 个可以在 b 中被观测到。此时系统会在没有公司特征点的格子内检测新的特征点，用星形表示，记为候选点。在关键帧 c 被检测到时，系统将星形特征点添加到它们第一次被检测出的关键帧 b 中。在这个过程中，并不是所有的候选点都可以添加到关键帧上。如果一个候选点在后来的关键帧上无法被检测到，就会被保存在候选队列中。如果多次投影失败，则该候选点会被删除。

4.3.2 深度滤波器

在关键帧被确定并提取特征之后，系统就会对关键帧上深度信息尚未收敛点特征点、线通过深度滤波器的方式进行深度估计，初始值为该关键帧的平均深度，并且赋予极大的不确定性。

如前文所说，在使用单目相机作为传感器的时候，最基本的深度估计方法就是三角化方法，它主要依靠两帧图像上的匹配点来计算空间点深度。如果有多帧图像的话，就相当于多次测量了该点的深度值。然后通过贝叶斯估计的方法，缩小测量值的不确定性，如图 4-8 所示。



4-8 空间点深度概率分布^[20]

在图 4-8 所示，对于参考帧 I_r 中的像素点 u_i ，如果只从参考帧的角度来说，其深度值是不确定的，我们假设深度值在 $[d_{\min}, d_{\max}]$ 之间。而在当前帧 I_k 中，系统通过极线搜索的方式来获得匹配的像素点。根据两个端点深度值 $d_{\min}$ 和 $d_{\max}$ ，计算该特征在当前帧 I_k 对应的像素点在蓝色线段上，称为极线段位置。之后就可以进行特征搜索匹配。如果极线段长度小于两个像素，则认为两个像素点匹配，可以直接采用光流法来较为准确的预测相机位姿。如果极线段长度较长的话，则需要通过块匹配的方式，来寻找和参考帧最为匹配的特征块。本文的系统使用了归一化互相关（Normalized Cross Correlation, NCC）的方法来实现块匹配。最后通过特征那个匹配的方式，求得像素对应的特征点位置。

通过上述方法获得一个特征点的深度估计值后，系统用贝叶斯方法对深度值

进行估计。深度滤波器采用了一种高斯-均匀的滤波器^[20]，模型如下：

$$p(\tilde{d}_i^k | d_i, \rho_i) = \rho_i N(\tilde{d}_i^k | d_i, \tau_i^2) + (1 - \rho_i) U(\tilde{d}_i^k | d_i^{\min}, d_i^{\max}) \quad (4.5)$$

其中 $\tilde{d}_i^k$ 表示观测到的深度值， d_i 表示特征点的实际深度， ρ_i 表示特征点为内点的概率， τ_i 表示一个像素单位对应的空间深度， $d_i^{\min}$ 、 $d_i^{\max}$ 表示深度值区间最小、最大值。通过该模型，系统实现了地图点的深度估计。

对于点特征的深度估计，在 SV0^[32] 中已经有了详细的描述，这里主要分析线特征的深度估计。

在针孔相机模型下，通常认为空间直线经过投影变换到成像平面后，依旧会使一条直线。通过成像平面上 l 的投影直线 l 来恢复空间直线 L 的时候，最普遍的方法是通过线段端点的深度估计来实现。这种方法在实际使用的时候，可能会有较大的误差。这是因为对于二维平面上的线段 l 来说，并不一定能找到一条三维空间线段 L 与其完全匹配。仅仅对线段的两个端点进行深度估计，并不能保证两条直线完全匹配。对于一条二维直线 l 来说，需要对线段上更多的点进行处理。

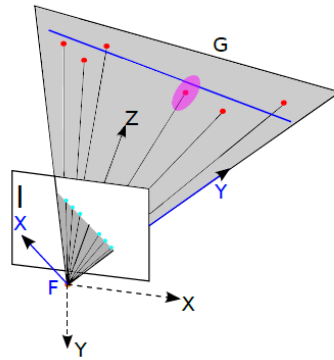


图 4-9 空间直线正则化

这里我们定义一个新的空间坐标系 F ，这个坐标系原点为相机光心， X 、 Y 轴都在直线 l 对应的极平面上。对于 l 以一个像素距离为单位均匀采样，获得一组深度信息未知的点。通过之前的分析，我们已经可以获得单独点的空间分布概率。这里，我们将这些采样点反投影到三维空间，如图 4-9 所示。图中的紫色椭圆，表示采样点的位置不确定性。

通过 RANSAC 方法剔除 outlier 点，可以获得了一条三维空间中的最佳匹配直线 L 。这里主要通过马氏距离 (Mahalanobis distance) 来对内点、外点进行判断^[33]：

$$dmah = \min_{q' \in l'} (p' - q')^T \sum_{p'}^{-1} (p' - q')$$

其中 p' 表示采样点在 F 上的反投影坐标， q' 则为采样点。

经过 RANSAC 方法，获得了一组内点 p'_i ， $i = 0, 1, \dots, n$ 。如果 p'_i 在采样点中占得比例低于预设的阈值，系统中用的是 0.6，则认为这条直线的三维空间位置无法进行估计，将其从深度滤波器中删除。如果比例高于阈值，则 F 坐标系下对应的空间直线 L^* 可被求出：

$$L^* = \min_L \sum_i \delta(p'_i)^T \sum_{p'_i}^{-1} \delta(p'_i)$$

$\delta(p'_i)$ 表示内点 p'_i 到直线 L 的距离。

将最为匹配的直线 L^* 重新投影到成像平面 I ，确定其在成像平面上对应的像素点及其对应的深度信息。通过这种方法，系统对地图中的线特征进行处理。

4.4 本章小结

本章介绍了基于点线综合特征的视觉里程计方法的原理，分别对系统的两个主要线程进行了阐述，包括系统的初始化，点线特征在相邻帧和地图上的匹配方法、地图中点线的深度估计等步骤，描述了系统的整个流程。参照此流程可以复现此单目视觉里程计系统。

第五章 实验与分析

通过对基于特征点法的视觉里程计和基于直接法的视觉里程计算法进行研究后,本文在第四章提出了基于点线特征融合的视觉里程计算法。该算法不仅可以在环境信息丰富的场景应用,而且在点特征缺失、线特征丰富的结构化环境中有着良好的定位性能。本章将通过实验,对该系统性能进行分析。

5.1 仿真实验

本文使用的系统是在 SVO 的基础上实现的,在原有的系统基础上添加了对线特征的提取与分析,以及对移动机器人位姿的非线性优化。仿真实验主要在德国慕尼黑工业大学提供的 TUM 数据集^[35]上进行测试,对该系统的精度、效率等进行分析。TUM 数据集作为 SLAM 领域国际通用的数据集之一,以机器人救援为背景提供了大量的真实图像数据,并提供了与数据集相对应的动捕数据作为系统的真值,为系统的精确度分析提供了基础。数据集附带的系统性能评估程序可以全面的对 SLAM、VO 系统进行评测,具有很高的权威性。本实验选取了其中 7 个场景进行实验,如下图所示。这些场景的共同特征是环境中包含了比较丰富的线特征,便于基于点线特征的系统进行测试。



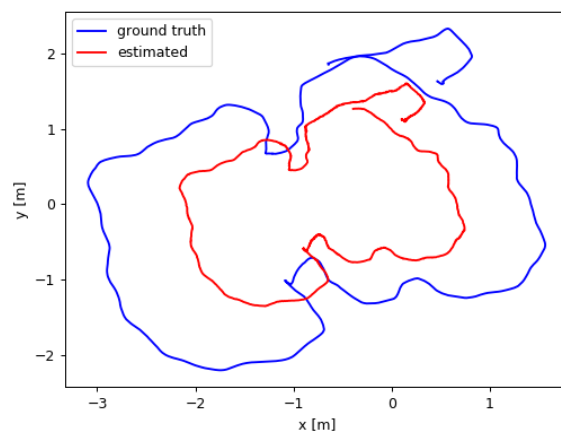
图 5-1 选取的 TUM 数据集

这里通过对 freiburg3_long_office_household 数据集的测试实验为例来对所设计的视觉里程计系统进行分析。

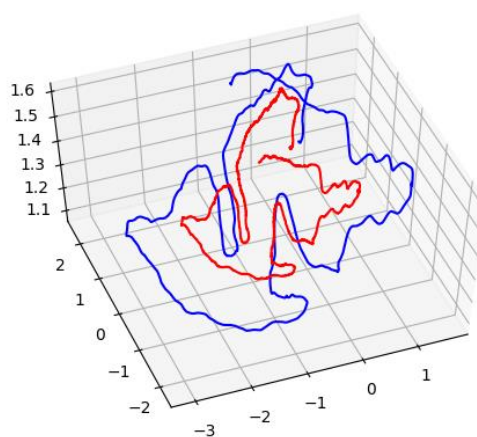
fr3 数据集在办公室场景内进行的图像采集,机器人围绕场景中的办公桌行进了一个环形的轨迹。场景内有着丰富的点线特征,可以满足我们设计的系统对两种不同特征的要求。fr3 数据集的完整轨迹长度为 20.228m,总共耗时 90.1s。



图 5-2 Freiburg3_long_office_household 数据集



(a) x-y 平面轨迹对比



(b) 三维空间中轨迹对比

图 5-3 fr3 数据集中估计轨迹与真实轨迹对比

图5-3是本文设计的点线视觉里程计在 fr3 数据集下估计的相机轨迹和真实轨迹的对比图。图中蓝色轨迹是数据集提供的基于动捕系统采集的真实轨迹数据，红色轨迹为本文设计的点线视觉里程计估计的程度轨迹值。图（a）是 x-y 平面上的二维运动轨迹，图（b）是三维空间中两条轨迹的对比情况。真实轨迹数据的单位是 m。在对轨迹进行估计的过程中，系统总共提取了 2583 帧关键帧。从图 5-3 可以发现，算法估计的轨迹相对真实轨迹有一定比例的缩放，这是因为本文设计的视觉里程计系统为单目视觉里程计。由于单目系统的尺度不确定性，单目系统测得的距离和真实值之间存在缩放关系。单目视觉里程计系统在启动的时候，需要通过初始化来确定尺度，尺度的大小由初始位移来决定。

由图像 5-3 我们可以看出，系统估计的运动轨迹在前半程和真实轨迹有良好的匹配，基本上是对真实路径的还原。但是随着运行时间段增长，相机运动到轨迹末端的时候，数据集提供的真实路径在 x-y 平面上有明显的闭合，但是我们设计的系统估计的数据出现了一定程度的偏差，未能完全闭合。出现偏差的原因是由于系统的累积误差。对于视觉里程计系统来说，因为没有像完整的 SLAM 系统那样有对全局进行优化的后端，所以系统的累积误差难以完全消除。

由于单目视觉里程计系统测得的轨迹相对真实轨迹有一定的缩放，我们采用相对位置误差^[36]（Relative Pose Error, RPE）作为评估指标，来对视觉里程计系统的精度进行分析。

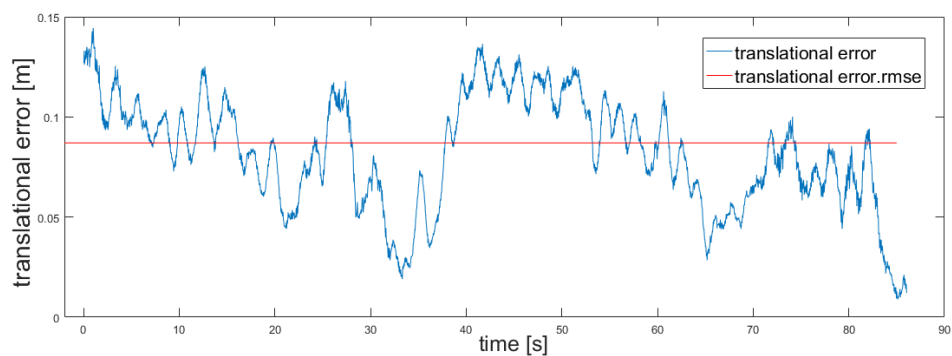
记视觉里程计估计的相机位姿 P_i 和真实轨迹的相机位姿 Q_i 均为 4×4 变换矩阵，固定窗口 Δt ，本次测量取 1。对于一组具有 n 对匹配点的轨迹序列，需要对 $m = n - \Delta t + 1$ 段位姿匹配点进行相对位姿误差的计算，并对这些匹配段求取均方误差 RMSE(Root mean Squared Error)。每对位姿匹配点的 RPE 为：

$$E_i = (Q_i^{-1} Q_{i+\Delta t})^{-1} (P_i^{-1} P_{i+\Delta t})^{-1} \quad (5.1)$$

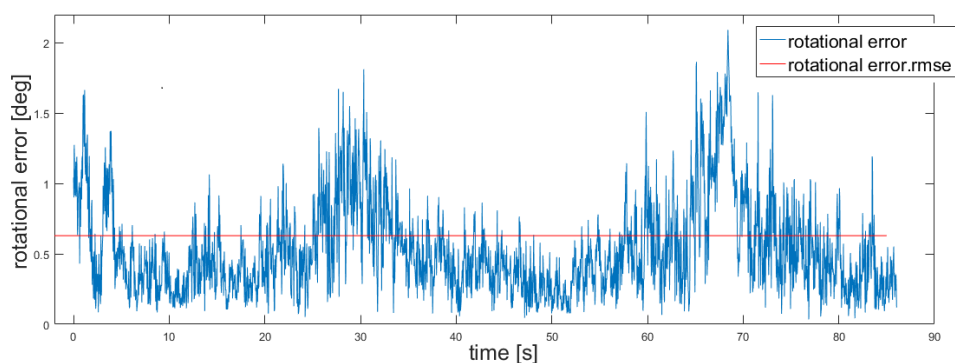
整个轨迹的 RMSE 为：

$$\begin{aligned} RMSE_{trans}(E_{i:n}, \Delta) &= \left(\frac{1}{m} \sum_{i=1}^m \|trans(E_i)\| \right)^{1/2} \\ RMSE_{rot}(E_{i:n}, \Delta) &= \left(\frac{1}{m} \sum_{i=1}^m \|rot(E_i)\| \right)^{1/2} \end{aligned} \quad (5.2)$$

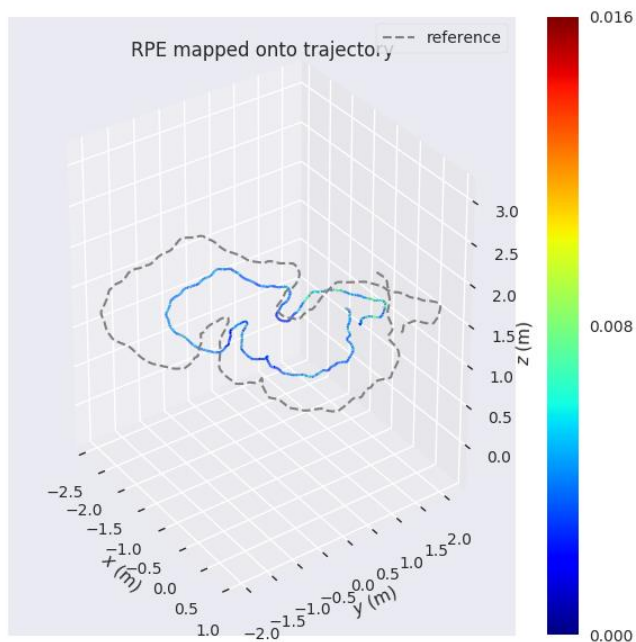
式（5.2）分别求取了平移变换的相对位姿误差和旋转变换的相对位姿误差。



(a) 平移误差



(b) 旋转误差

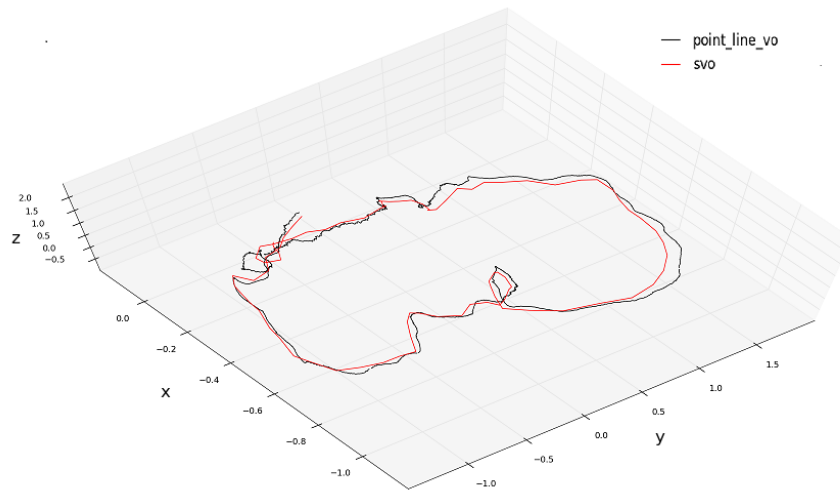


(c) 预测轨迹与真实轨迹对比

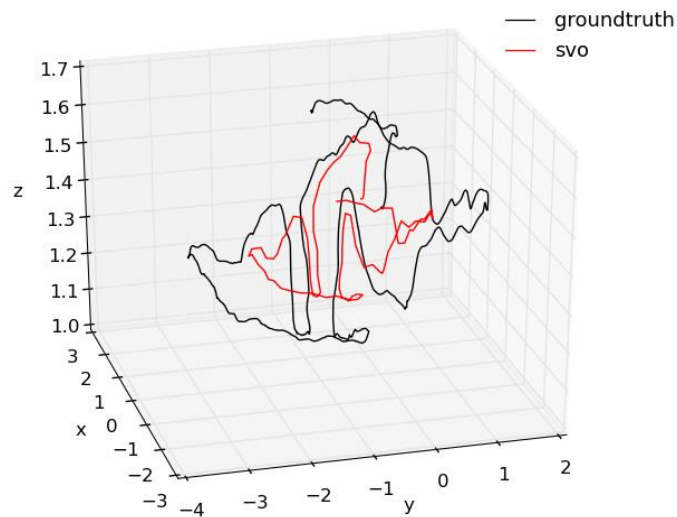
图 5-3 fr3 数据集下基于点线特征的视觉里程计估计轨迹相对误差

图 5-3 为本文设计的视觉里程计在 fr3 数据集进行测试的误差分布情况。图(a)

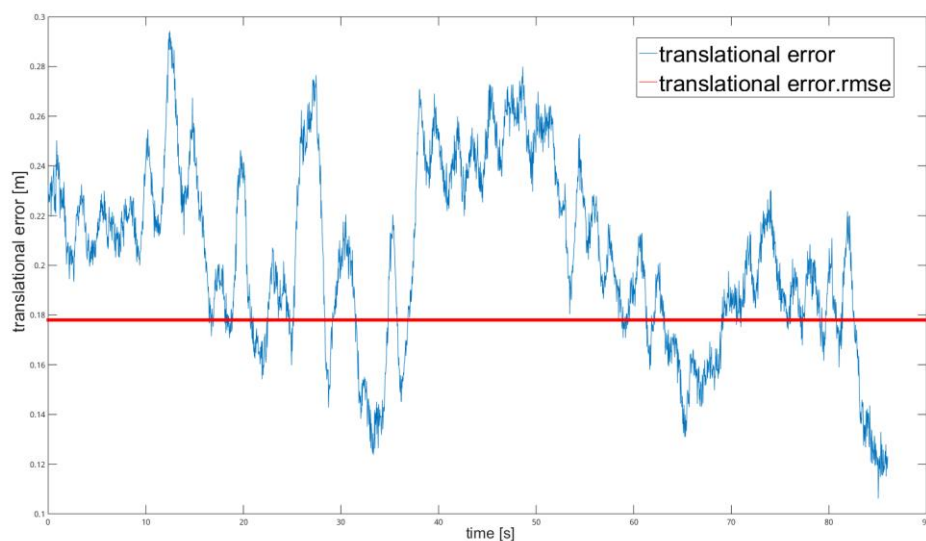
为轨迹的平移误差，图（b）为旋转误差，图（c）这是估计轨迹和真实轨迹的对比，通过不同的颜色表示该位置相对误差的大小。通过图（c）我们可以看出，估计值和真实值拟合基本正确，平移误差最大为 0.144 m，旋转误差最大为 2.09 deg，基本可以满足定位精度需求。



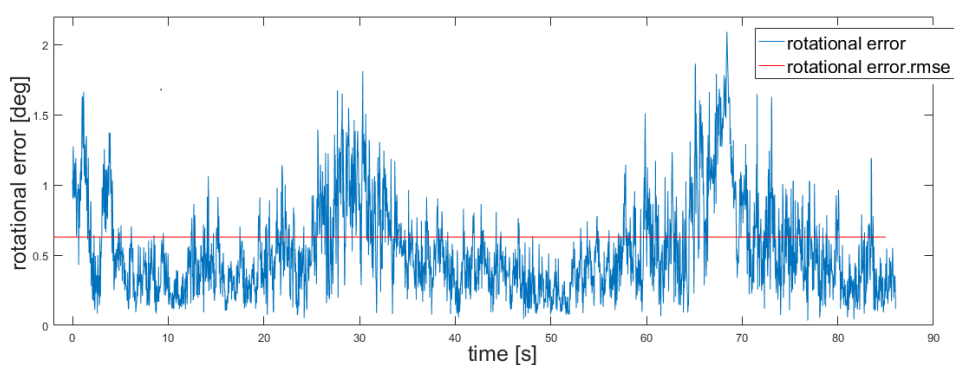
(a)点线视觉里程计和 SVO 的三维轨迹对比



(b) SVO 轨迹与标准轨迹对比



(c) svo 在 fr3 数据集下平移误差分析



(d) svo 在 fr3 数据集下旋转误差分析

图 5-4 基于点线特征的视觉里程计和 svo 在 fr3 数据集下对比

图 5-4 是基于点线特征的视觉里程计和基于点特征的视觉里程计在 fr3 数据集下的轨迹对比和分析。图 (a) 是 fr3 数据集下两种视觉里程计的三维轨迹对比图。其中黑色轨迹是基于点线特征的视觉里程计轨迹，红色的是基于点特征的 svo 轨迹；图 (b) 是 svo 所测机器人运动的红色轨迹和真实路径的黑色轨迹对比图。从这两个图我们可以看出两种方法都可以实现机器人的轨迹跟踪，并且基于点线特征的轨迹跟踪效果较好。图 (c) 和图 (d) 则是对 svo 算法所绘轨迹的误差分析。通过和图 5-3 对比，我们可以直观地看出，线特征的使用可以明显的提高系统的精度。表 5.1 为两种不同的视觉里程计方法在 fr3 下精度对比结果。

表 5.1 基于点特征和点线特征的视觉里程计在 fr3 数据集下精度对比

	SVO	点线特征的视觉里程计
Translational_error.rmse(m/s)	0.1794	0.0824
Rotational_error.rmse(deg/s)	1.1132	0.6292

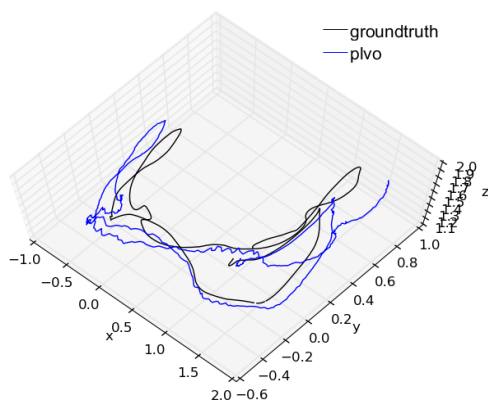
视觉里程计系统的运行耗时也是衡量系统性能的一个指标。相比较点特征而言，线特征的提取、匹配和位置估计都要使用更多的运算资源，势必会对系统的运行速度造成一定的降低。表 5.2 是运行 fr3 数据集情况下，两种不同原理的视觉定位系统在各环节的耗时对比。

表 5.2 系统平均耗时

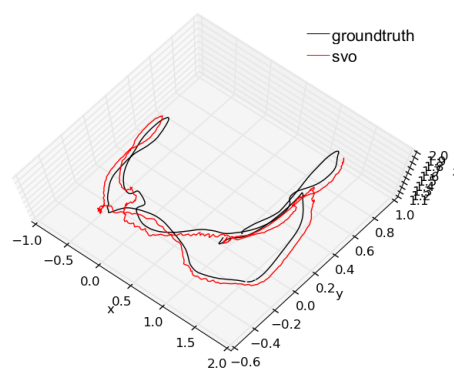
	点线视觉里程计	SVO
稀疏图像匹配	8.43ms	2.86ms
特征匹配	9.34ms	4.13ms
位姿与结构优化	1.67ms	0.35ms
运动估计	19.94ms	8.6ms

在时间效率上，对于 640x480 的图像，系数匹配环节需要 8.43ms，特征匹配 9.34ms，位置与结构优化 1.67ms，运动估计 19.94ms。因为线特征的加入，系统运行速度相比单纯使用点特征的时候，会有一定程度的减慢。但是在运动估计线程仍然可以达到约 40 帧的速度，可以在实时条件下运行。

下面是在其他几个测试序列中运行结果的对比图。在这里，我们都通过 TUM 数据值提供的标准评估工具，将系统估计的运动轨迹和真实轨迹做了一个对比。

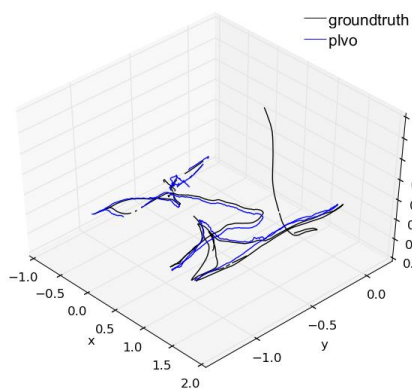


(a) 点线视觉里程计轨迹与标准轨迹对比

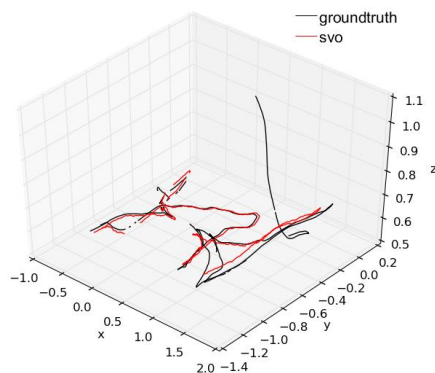


(b) svo 轨迹与标准轨迹对比

fr1_desk 数据集下点线视觉里程计和 svo 结果对比

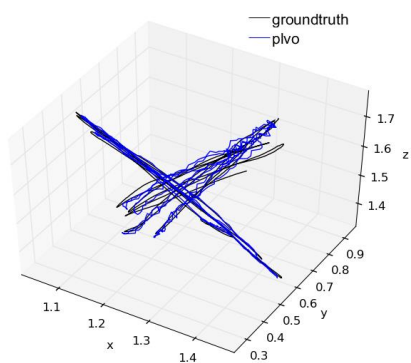


(a) 点线视觉里程计轨迹与标准轨迹对比

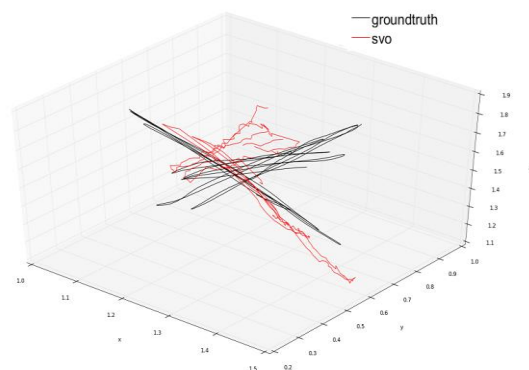


(b) svo 轨迹与标准轨迹对比

fr1_floor 数据集下点线视觉里程计和 svo 结果对比

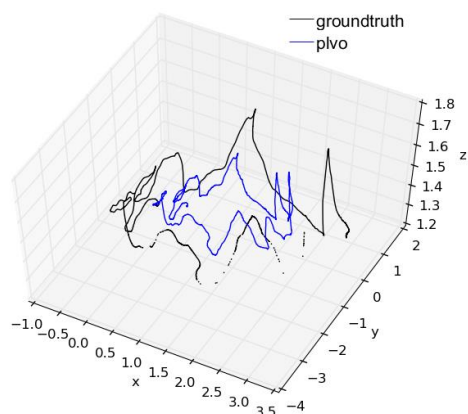


(a) 点线视觉里程计轨迹与标准轨迹对比

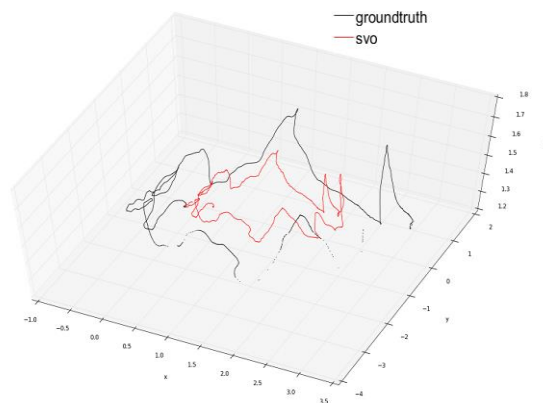


(b) svo 轨迹与标准轨迹对比

fr1_xyz 数据集下点线视觉里程计和 svo 结果对比

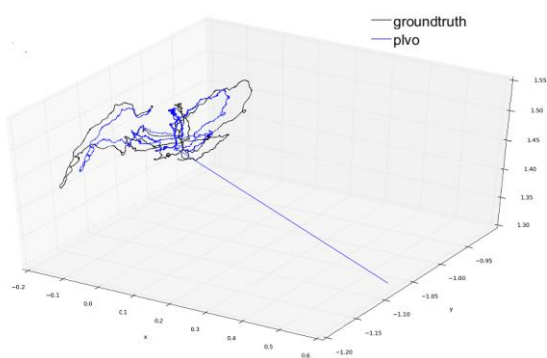


(a) 点线视觉里程计轨迹与标准轨迹对比

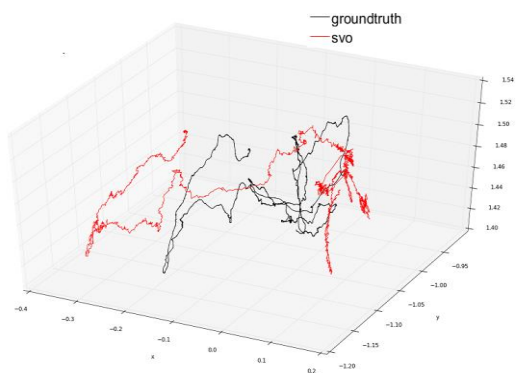


(b) svo 轨迹与标准轨迹对比

fr2_desk 数据集下点线视觉里程计和 svo 结果对比

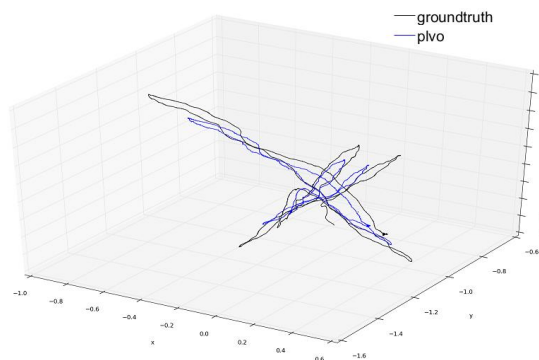


(a) 点线视觉里程计轨迹与标准轨迹对比

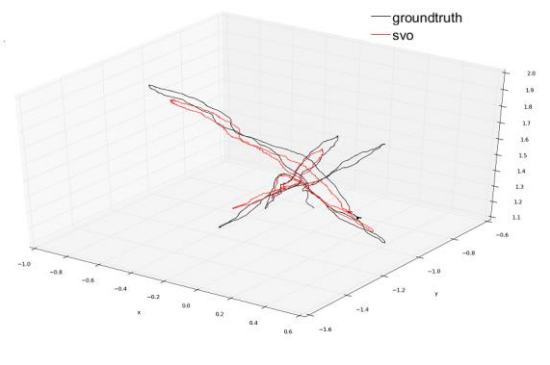


(b) svo 轨迹与标准轨迹对比

fr2_rpy 数据集下点线视觉里程计和 svo 结果对比



(a) 点线视觉里程计轨迹与标准轨迹对比



(b) svo 轨迹与标准轨迹对比

fr2_xyz 数据集下点线视觉里程计和 svo 结果对比

图 5-5 不同数据集下系统运行轨迹记录以及和 svo 的对比

图 5-5 是基于点线特征的视觉里程计系统和基于点特征的 svo 系统在不同数据集下的对比。通过系统在这些数据集中的运行情况，可以看出大部分数据集中，

估计的轨迹值和真实的路径是可以匹配的，但是随着系统运行距离的增长，系统估计的轨迹都有一定程度上的漂移。而在 fr2_rpy 数据集中，可以看到估计的相机位姿在轨迹后半段丢失，并且未能找回。这是由于在系统运行时，未能从图像中采集到足够的线特征，造成了位置估计线程的失效，并且后续的重定位也未能重新确定相机位置。所以本文设计的基于点线的视觉里程计，并不能满足所有的场景。在一些点线特征缺失的情况下，系统有跟踪失败的可能性。表 5.3 是通过测评工具对这些数据集下系统性能的评估。

表 5.3 不同数据集下系统性能指标

	点线视觉里程计		SVO	
	Trans_error.rmse (m/s)	Rot_error.rmse (deg/s)	Trans_error.rmse (m/s)	Rot_error.rmse (deg/s)
fr1_desk	0.1626	4.3206	0.0623	2.1243
fr1_floor	0.0259	1.4980	0.4112	21.0040
fr1_rpy	x	x	0.14234	1.3234
fr1_xyz	0.0261	1.5191	0.1780	11.8003
fr2_desk	0.0909	0.6044	0.0908	0.9912
fr2_rpy	0.0705	2.0118	0.0155	0.6424
fr2_xyz	0.0179	0.3925	0.0213	1.1132

通过表 5.3 可以看出，本文设计的系统在大部分场景下性能优于或者持平基于点特征的 SVO 系统，但是在一些特定场景下可能出现位姿估计失败的情况，如 fr1_rpy 数据集。位姿估计失败的原因大部分是无法从场景中提取足够的线特征，以及线特征的匹配错误导致的。如果由于相机运动过快造成图像模糊、遮挡等情况，或者由于图像噪声过多导致系统出现了误匹配，这些情况在设计系统的时候没有进行处理。而在 fr2_rpy 数据集中，前半程基于点线特征的视觉里程计还可以稳定的估计轨迹，但是由于运行过程中特征点缺失导致位姿估计失效，误差急剧上升，最终定位效果远不如 SVO 系统。在较长轨迹上运行本系统的时候，累积误差的消除也是需要考虑的地方。这些情况的发生为后续系统性能的提升提供了一定的参考。

5.2 走廊环境下点线视觉里程计的定位实验

为了验证本文方法在实际场景下的有效性,我们以 Turtlebot2 作为移动平台,分别搭载 RPLidar 激光雷达和罗技 C270 摄像头,在实际的走廊场景下进行了实验。

5.2.1 实验平台硬件

5.2.1.1 Turtlebot2 机器人

Turtlebot^[37] 机器人是美国 Willow Garage 公司在 2010 年开发的移动平台。因其价格较低且有完善的开源支持,使得 Turtlebot 一推出就在移动机器人开发者中受到了广泛的欢迎。到目前为止, Turtlebot 已经发布了三代产品,这里我们采用的是第二代 Turtlebot2,如图 5-5 所示。Turtlebot2 机器人有软件部分和硬件部分组成。在软件方面, Turtlebot2 主要通过机器人操作系统 (Robot Operating System, ROS)^[38] 实现对移动机器人的控制。



图 5-6 Turtlebot2 机器人

硬件方面, Turtlebot2 机器人通过 Kobuki 底座来实现运动功能。Yujin Kobuki 控制底座如图 5-6 所示,是 Yujin 公司为 Turtlebot 机器人专门开发的控制底座,内置 32 个传感器,2 个驱动轮,110 度/秒单轴陀螺仪,一个预排程序的行为编码器,一个可以扩大的输入输出端口和一个后挡板^[39]。Kobuki 拥有基于 ROS 操作系统的完整开发包,实现了对移动平台的底层控制,用户可以直接对上层系统进行

设计，极大的提高了机器人系统的开发便捷性。



图 5-7 Kobuki 控制底座

5.2.1.2 外部传感器

在对比实验中，分别采用了 Rplidar 激光雷达和罗技 C270 网络摄像头作为外部传感器。激光雷达是最早应用在移动机器人研究中的外部传感器，拥有解析度高，测距精度高、不受光线影响等诸多优点。但是激光雷达高昂的价格也限制了应用。RPLidar 相对工业级激光雷达使用了相对普通的部件，通过算法来弥补性能的差异。在室内场景下，可以满足定位精度的要求，所以我们选择了 RPLidar 来做对比试验



图 5-8 RPLidar 激光雷达

视觉传感器方面，我们选用了罗技 C270 网络摄像头。C270 作为 USB 摄像头，可以在 Ubuntu 系统下直接使用，无需复杂的驱动设置。C270 使用 300 万像素的

单目摄像头，输出 640x480 分辨率的图像，可以满足视觉里程计算法的需求。



图 5-9 罗技 C270 摄像头

5.2.2 试验结果与分析

我们以 Turtlebot2 机器人为载体，搭载 RPLidar 和罗技 C270 摄像头，在一个 30m x 2.5m 的走廊场景下进行实验。实验场地如图 5-9 所示。走廊两侧为棕色木门和宣传板，并伴有较多的白色墙壁。木门和宣传板可以提供一定点特征与线特征，但是白墙会对视觉算法造成极大的干扰。本次实验主要分为两部分：使用激光传感器，用自适应蒙特卡罗定位（Adaptive Monte Carlo Localization, AMCL）^[40] 获取轨迹；使用视觉传感器，基于点线视觉里程计的估计相机运行轨迹。通过两种方法所得轨迹的对比，来验证本文方法的有效性。



(a) 走廊整体场景



(b) 走廊一侧棕色木门



(c) 走廊一侧白色墙壁



(d) 走廊尽头电梯

图 5-10 走廊场景

图 5-10 是实验场景的二维空间展示。在实验中，Turtlebot2 机器人从点 A 沿着蓝色轨迹运动点 B。因为两个传感器均固定在地面移动平台上。所以我们只考虑 X-Y 轴上的轨迹运动，忽略 Z 轴的变化。

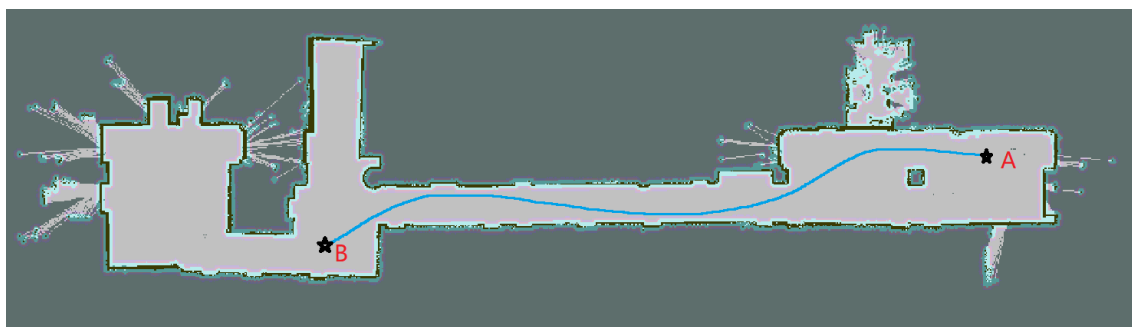
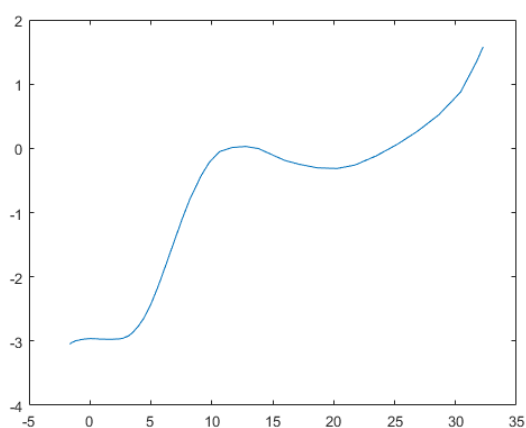
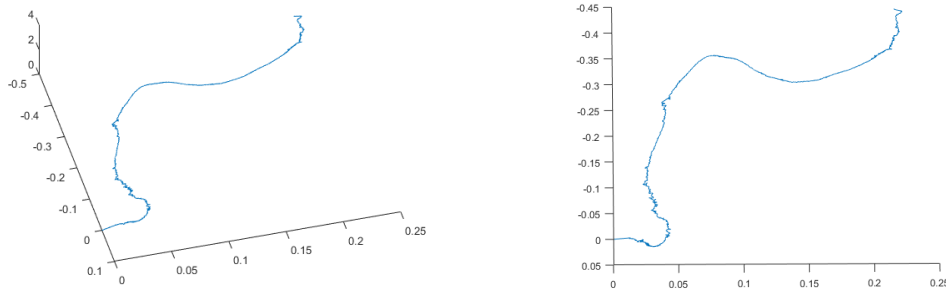


图 5-11 走廊场景



(a) 基于 amcl 算法的轨迹



(b) 基于视觉里程计的三维轨迹

(c) 基于视觉里程计的 X-Y 平面轨迹

图 5-12 两种方法测量轨迹对比

图 5-11 是通过两种不同算法测得的轨迹对比图。图 (a) 是通过激光雷达, 基于 amcl 算法测得的机器人运动轨迹; 图 (b) 是使用本文的视觉里程计测得的机器人三维空间运动轨迹; 图 (c) 是视觉里程计所测轨迹在 X-Y 平面上的投影。需要说明的是, 激光雷达测得的数据忽略了 z 轴的变化, 所以在图 (a) 中我们通过二维图像来描述结果。但对于视觉里程计来说, 其测得的轨迹是三维轨迹, 该运动轨迹以相机初始位姿为原始坐标, 与激光雷达的坐标系之间存在一个旋转。对于视觉里程计算法, 通过 2497 帧关键帧来绘制运动轨迹, 而激光雷达是通过 50 组空间点来绘制的运动轨迹。相比较激光算法, 基于点线的视觉里程计可以绘制更为详细的轨迹。这里我们对二者轨迹只进行定性的分析, 因为单目视觉里程计的尺度问题, 二者轨迹存在较大比例的缩放。通过二者的对比, 我们可以发现, 基于点线特征的视觉里程计轨迹在成比例放大后, 与激光算法所获得的轨迹基本相符, 说明本文设计的视觉里程计基本可以满足定位的需求。二者轨迹的偏差, 主要还是因为实验场景中大量的白墙, 使得视觉里程计算法在运行时提取的特征数量受到限制, 定位精度收到影响。而激光传感器这不会受到实验环境的影响。

5.3 本章小结

本章对基于点线特征的视觉定位系统在 TUM 数据集上的测试, 通过计算相对位置误差对整个系统进行了分析, 证明了基于点线特征设计的系统的精确度和实效性; 通过走廊实验证明了基于点线特征的视觉里程计算法在实际场景中的实用性。在结构化的场景中, 同时处理点特征和线特征可以提高系统的准确定。但是线特征的引入也会降低系统的稳定性, 在特定场景下会导致系统的失效。

第六章 总结与展望

6.1 总结

本文首先对视觉里程计系统进行了综述，对视觉里程计问题的提出和发展进行了介绍。针对结构化场景中点特征数量较少，线特征大量出现，基于点特征的视觉里程计不能很好的完成位姿估计任务，设计了基于点线特征的视觉里程计定位系统。本论文的主要工作总结为：

1. 介绍了视觉定位系统中使用的传感器：单目相机，研究其工作原理以获得针孔相机模型；介绍了相机参数的标定方法，并对系统中用到的线特征的提取、匹配，线特征的表达进行了描述。
2. 构建了视觉定位系统的数学模型，并对特征点法和直接法的算法流程和核心思想进行了推导。
3. 针对基于点特征的视觉定位系统在结构化场景中不能稳定工作的问题，提出了一种基于点、线特征综合使用的视觉里程计方法，综合运用特征点法和直接法的优点，通过多重优化的方法对相机位姿精确估计。
4. 利用 TUM 的数据集对本文搭建的基于点线特征的视觉定位系统进行评估，通过实验证明该视觉定位系统有足够的定位精度，在结构化的场景中系统的精确度和实效性都可以获得保证。

6.2 展望

本文提出的方法虽然在一定程度上可以解决点特征缺失时系统的定位问题，但是仍有不足和可以改进的地方：

1. 优化对线特征的处理。在线特征的提取阶段，遮挡、模糊等多种情况都可能造成检测线段的断裂；空间中干扰项太多可能造成线特征的错误检测与匹配。可以尝试使用结构化直线的方式来替代现有的空间直线，对系统的性能进行改进。
2. 添加闭环检测环节。通过闭环检测实现完整路径上的系统优化，可以有效避免本文系统中累积误差造成的轨迹偏移问题。

3. 结合惯性测量单元提高系统精度。可以通过非线性优化的方式，综合处理相机传感器获得的环境信息和惯导设备提取的自身运动参数，提高系统的鲁棒性。
4. 可以考虑通过深度学习的方式，实现特征的匹配。作为人工智能领域的前沿方向，将深度学习和视觉里程计融合，会是将来发展的趋势。

参考文献

- [1] H. Durrant-Whyte. Where am I? A tutorial on mobile vehicle localization[J]. Industrial Robot: An International Journal, 1994, 21(2), 11-16.
- [2] Biswas J, Velose M. WiFi localization and navigation for autonomous indoor mobile robots[J].2010
- [3] Scaramuzza D, Fraundorfer F. Visual Odometry[j]. IEEE Robotics& Automation Magazine. 2011, 18(4): 80-92.
- [4] Matthies L, Shafer S. Error modeling in stereo navigation[M].Autonomous robot vehicles. Springer-Verlag New York, Inc. 1990:239-248.
- [5] Nister, D, O. Naroditsky, and J. Bergen. "Visual odometry." Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on IEEE, 2004:I-652-I-659 Vol.1.
- [6] Badino H, Yamamoto A, Kanade T. Visual Odometry by Multi-frame Feature Integration[C]. IEEE International Conference on Computer Vision Workshops. IEEE Computer Society, 2013:222-229.
- [7] Engel J, Koltun V, Cremers D. Direct Sparse Odometry.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99):1-1.
- [8] 吕强, 周文晖, 刘济林. 基于 SIFT 特征提取的单目视觉里程计在导航系统中的实现[J]. 传感技术学报, 2007, 20(5):1148-1152.
- [9] Dava Marr. "Vision: A computational investigation into the human representation and processing of visual information", W.H. Freeman and Company,1982.
- [10] 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.
- [11] 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.
- [12] Richard Hartley and Andrew Zisserman. Multiple View Geometry in computer vision. Cambridge University Press,2003.

- [13] Grejner-Brzezinska D A, Adviser A, Yilmaz R R, et al. Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-based Georeferencing of Aerial Images[J]. 2011.
- [14] Penate-Sanchez A, Andrade-Cetto J, Moreno-Noguer F. Exhaustive linearization for robust camera pose and focal length estimation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(10):2387.
- [15] Rusinkiewicz S, Levoy M. Efficient variants of the ICP algorithm[C].International Conference on 3-D Digital Imaging and Modeling, 2001. Proceedings. IEEE, 2002:145-152.
- [16] Björck, Åke. Numerical methods for least squares problems[J]. 1996.
- [17] Rosten E, Drummond T. Machine learning for high-speed corner detection[C].European Conference on Computer Vision. Springer-Verlag, 2006:430-443.
- [18] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces[C]. IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2007:1-10.
- [19] Gomez-Ojeda R, Gonzalez-Jimenez J. Robust stereo visual odometry through a probabilistic combination of points and line segments[C].IEEE International Conference on Robotics and Automation. IEEE, 2016:2521-2526.
- [20] George Vogiatzis, Carlos Hernández. Video-based, real-time multi-view stereo [J]. Image & Vision Computing, 2011, 29(7):434-441.
- [21] Lu Y, Song D. Robust RGB-D Odometry Using Point and Line Features[C]. IEEE International Conference on Computer Vision. IEEE Computer Society, 2015:3934-3942.
- [22] Solà J, Vidal-Calleja T, Devy M. Undelayed initialization of line segments in monocular SLAM[J]. IEEE, 2009:1553 - 1558.
- [23] Nister, D; Naroditsky, O; Bergen, J. Visual Odometry. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), Washington, DC, USA, 27 June–2 July 2004. Volume 1; pp. 652–659.
- [24] Barfoot T D. STATE ESTIMATION FOR ROBOTICS[M].2017.

- [25] Blanco J L. A tutorial on SE(3) transformation parameterizations and on-manifold optimization[J].University of Malaga, Tech. Rep,2010(3).
- [26] 谢晓佳, 基于点线综合特征的双目视觉 SLAM 方法[D].浙江大学硕士学位论文, 2017。
- [27] Zhang Z. A flexible new technique for camera calibration[J]. Tpami, 2000, 22(11):1330-1334.
- [28] 高翔。视觉 SLAM 十四讲。电子工业出版社, 2017。
- [29] Tardif J, Pavlidis Y. and Daniilidis K. Monocular visual odometry in urban environments using an omnidirectional camera[C]. In Proc.IEEE RSJ Int. Conf. Intelligent Robots and Systems, 2008, pp. 2531-2538.
- [30] 孙作雷, 黄加明, 张波。基于 RANSAC 的奇异值剔除的单目视觉里程计[J].上海海事大学学报.2016(12):87-91.
- [31] 张毅, 童学容, 罗元。一种改进 SURF 算法的单目视觉里程计[J].重庆邮电大学学报(自然科学版), 2014, 26(3):390-396.
- [32] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C].IEEE International Conference on Robotics and Automation. IEEE, 2014:15-22.
- [33] Klein G, Murray D. Improving the Agility of Keyframe-Based SLAM[M]// Computer Vision – ECCV 2008. Springer Berlin Heidelberg, 2008:802-815.
- [34] Zuo X, Xie X, Liu Y, et al. Robust visual SLAM with point and line features[C].IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2017:1775-1782.
- [35] 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. IEEE,2013:573-580
- [36] Endres F, Hess J, Engelhard N, et al. An evaluation of the RGB-D SLAM system[C].IEEE International Conference on Robotics and Automation. IEEE, 2012:1691-1696.
- [37] <http://www.turtlebot.com>
- [38] <http://www.ros.org>

-
- [39] 赵雷, 基于二维地图蒙特卡罗定位的三维室内地图构建[D].汕头大学硕士学位论文, 2017.
- [40] Dellaert F, Fox D, Burgard W, et al. Monte Carlo localization for mobile robots[C]. IEEE International Conference on Robotics and Automation, 1999. Proceedings. IEEE, 2002:1322-1328 vol.2

致 谢

光阴如梭，时光荏苒。眨眼之间，我已经迎来了汕头大学研究生生活的终点。

首先衷心的感谢我的导师范衡老师。范衡老师为人亲和，思维活跃，视野开阔。在范衡老师的引领下，我走进了学术的殿堂，学习到了更多的专业知识。在研究生期间，为我指明了研究的方向，提供了良好的科研条件，并给予了我充分的自由让我在喜欢的领域钻研。范衡老师用创新和敏捷的思想指导我攻克难题，用踏实严谨的科研态度激励我前进。在此谨致以我最深切的敬意。

齐次，要感谢实验室的研究小组成员，是大家的互相帮助互相鼓励让我的学习不断前进。在每周的小组讨论，大家让我学习到了不同领域的知识，拓宽了我的思路，也促进了和大家的交流。感谢赵雷师兄、红辉师兄、肖杨师兄、贵杰师兄的在我学习过程中提供的无私帮助和指引。感谢李冲、姚利、方毅、陈燊、中兴、继彪的陪伴和帮助使得我们共同成长。感谢汕头大学为我提供的舒适的生活环境和方便的科研条件，庆幸可以在这里和大家共同度过三年的美好时光

最后，感谢在身后默默支持我的家人。是家人的支持和鼓励给了我继续学习下去的勇气。每当遇到困难感觉无助的时候，是家人的陪伴让我坚强。你们是我坚强的后盾和温暖的港湾。

我怀揣着深切的敬意，感谢那些帮助过、鼓励过、支持过我的人，谢谢你们的陪伴让我成长。

王宇鹏

2018年5月 汕头大学

