

分类号: S224.2

授予学位单位代码: 10434

研究生学号: 2016120023

山東農業大學

全日制硕士专业学位论文

基于 RealSense 的室内机器人 SLAM 方法设计与试验

Design and Experiment of SLAM Method for Indoor Robot

Based on RealSense

研 究 生 : 盖顺华

学 位 类 别 : 工程硕士

学 科 专 业 : 机械工程

学 院 : 机械与电子工程学院

指 导 教 师 : 闫银发 副教授 范永 高级工程师

2018 年 12 月 7 日

论 文 提 交 日 期:

论 文 答 辩 日 期:

学 位 授 予 日 期:

学 科 门 类:

答 辩 委 员 会 主 席:

关于学位论文原创性和使用授权的声明

本人所呈交的学位论文，是在导师指导下，独立进行科学研究所取得的成果。对在论文研究期间给予指导、帮助和做出重要贡献的个人或集体，均在文中明确说明。本声明的法律责任由本人承担。

本人完全了解山东农业大学有关保留和使用学位论文的规定，同意学校保留和按要求向国家有关部门或机构送交论文纸质本和电子版，允许论文被查阅和借阅。本人授权山东农业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、缩印或其他复制手段保存论文和汇编本学位论文。

保密论文在解密后应遵守此规定。

论文作者签名：_____

导 师 签 名：_____

日 期：_____

目 录

中 文 摘 要.....	I
Abstract.....	II
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 视觉 SLAM 概述.....	2
1.3 国内外研究现状.....	4
1.4 本文主要研究工作.....	6
2 视觉 SLAM 图像基础	8
2.1 针孔模型.....	8
2.2 相机坐标系.....	9
2.3 坐标变换.....	10
2.4 本章小结.....	12
3 RealSense 标定.....	13
3.1 RealSense 简介	13
3.2 深度原理.....	15
3.3 数据采集和预处理.....	17
3.4 RealSense 标定试验	19
3.5 本章小结.....	23
4 SLAM 数据关联	24
4.1 特征提取算法对比改进.....	25
4.1.1 特征提取算法原理分析.....	25
4.1.2 对比试验及改进方法.....	33
4.2 改进的 FLANN 分类树特征匹配方法.....	35
4.2.1 算法设计.....	36
4.2.2 算法实现及试验验证.....	38
4.3 改进后的位姿计算方法.....	40
4.4 基于词袋模型的闭环检测.....	43

4.4.1 算法原理分析	43
4.4.2 算法设计实现	44
4.5 本章小结	45
5 基于图优化的 SLAM 方法	47
5.1 构建图优化代价函数	47
5.2 求解最小二乘问题	48
5.3 稀疏性加速 SLAM 运算	49
5.4 构建占据网格地图	51
5.5 本章小结	54
6 系统设计与试验验证	55
6.1 SLAM 程序流程	55
6.2 试验设备	56
6.3 试验设计及试验结果	57
6.3.1 TUM 数据集试验	58
6.3.2 实时采集试验	60
6.4 结果讨论	63
6.5 本章小结	64
7 工作总结与展望	65
7.1 工作总结	65
7.2 创新点	66
7.3 展望	66
参考文献	67
致 谢	71
硕士期间发表论文及申请专利情况	72

中文摘要

本文以 RealSense R200 深度相机作为外部环境传感器，对未知环境中机器人的同时定位与地图构建（Simultaneous Localization and Mapping, SLAM）问题进行理论研究和试验验证，设计并实现了一种基于 RealSense 的 SLAM 方法。本研究得到了山东省重大科技创新工程项目“云基家庭服务机器人实用化关键技术研究及其产业化”（项目编号：2017CXGC0926）的资助。论文主要完成了以下工作：

（1）研究分析 RealSense 的成像原理和深度测量原理，通过搭建开发框架采集 RealSense 的数据，对深度图像采用双边滤波算法进行去噪和修复空洞，并对 RealSense 进行了标定试验，校正相机畸变并得到相机内参用于后续计算。

（2）分析了三种特征提取算法的原理，通过试验对比三种算法的性能，选择耗时最少的 ORB 特征提取算法，引入了图像区域分割的方法提高特征提取的均匀度。提出了改进后的 FLANN 分类树特征匹配算法，加快匹配速度并剔除绝大多数误匹配，并通过试验验证了算法的实时性。

（3）设计实现了改进后的位姿求解方法，首先通过 3D-2D 特征点投影求解 SVD 方式得到初始位姿，然后将初始位姿输入到 Point-to-Plane ICP 中，通过迭代最小化点到面的距离求解精准位姿，避免了常规透视 N 点投影法对正确特征匹配的依赖，相比常规迭代最近点方法速度更快。

（4）使用词袋模型方法实现闭环检测环节，设计了基于时间的闭环验证方法。实现了基于图优化理论的 SLAM 方法，构建出两种形式地图。

（5）设计实现 SLAM 程序，搭建机器人运动平台，通过采用 TUM 公开数据集和实时采集运行两种方式，对本文设计实现 SLAM 方法进行了试验验证。经验证，本文设计实现的 SLAM 方法能够较好的完成机器人的定位在室内环境下的定位和地图构建功能。

关键词：同时定位与地图构建；RealSense；机器人；数据关联；图优化

Design and Experiment of SLAM Method for Indoor Robot Based on RealSense

Abstract

As an external environment sensor, this paper conducts theoretical research and experimental verification on Simultaneous Localization and map construction (SLAM) of robots in unknown environments using RealSense R200 depth camera, and designs and implements a SLAM method based on RealSense. This research was funded by shandong province's major scientific and technological innovation project "research and industrialization of key technologies for the practical application of yunji family service robot" (project number: 2017CXGC0926). The thesis mainly completed the following work:

(1) Research and analyze the imaging principle and depth measurement principle of RealSense, collect RealSense data by constructing the development framework, use the bilateral filtering algorithm to denoise and repair the cavity in the depth image, and perform calibration test on RealSense to correct the camera distortion and obtain The camera internal parameters are used for subsequent calculations.

(2) The principle of three feature extraction algorithms is analyzed. By comparing the performance of the three algorithms, the ORB feature extraction algorithm with the least time is selected, and the image region segmentation method is introduced to improve the uniformity of feature extraction. The improved FLANN classification tree feature matching algorithm is proposed. The block matching speed is added and most mismatches are eliminated. The real-time performance of the algorithm is verified by experiments.

(3) The improved pose method is designed and implemented. Firstly, the initial pose is obtained by solving the SVD method by 3D-2D feature point projection, and then the initial pose is input into the Point-to-Plane ICP to minimize the point by iteration. The distance to the surface solves the precise pose, which avoids the dependence of the conventional perspective N-point projection method on the correct feature matching, which is faster than the conventional iterative method.

(4) The closed-loop detection process is implemented by using the word bag model method, and a time-based closed-loop verification method is designed. The SLAM method based on graph optimization theory is implemented, and two forms of maps are constructed.

(5) Design and implement the SLAM program, build the robot motion platform, and verify the design of the SLAM method by using TUM public data set and real-time acquisition and operation. It is verified that the SLAM method designed and implemented in this paper can exchange the positioning and map construction functions of the robot in the indoor environment.

Keywords: Simultaneous localization and mapping; RealSense; Robot; Data association; Graph optimization

1 绪论

1.1 研究背景及意义

近年来机器人领域的一些先进技术对社会发展和工业生产做出了非常大的贡献，如图 1-1 所示，机器人系统在自动化生产领域、军事领域、医疗领域以及家庭服务等领域得到了广泛应用，越来越多的机器人应用出现在日常生活中，一步步改善我们的生活环境。机器人产业的发展对国家现代化建设具有巨大的推动作用，世界上的工业大国都将机器人产业视为国家的战略发展目标（金周英，2009；吕铁，2015）。对于移动机器人来说，自主移动功能是需解决的核心问题，可靠的定位是机器人自主移动的基础，移动中导航的路径跟踪、路径规划等任务都需要机器人能够在移动过程中连续准确定位。然而，在未知环境下由于没有先验地图信息，机器人如何在未知环境中利用传感信息建立准确的环境地图，并同时利用创建的地图实现准确的自定位已成为移动机器人导航领域的核心研究问题，即同时定位与地图构建 (Simultaneous Localization and Mapping, SLAM) 问题。



(a) 工业机器人 Industrial robot



(b) 服务机器人 Service robot



(c) 医疗机器人 Medical robot



(d) 分拣机器人 Sorting robot

图 1-1 各类机器人

Fig.1-1 Various types of robots

SLAM 需要机器人依靠特定传感器在未知环境中估算自身运动并构建环境地图，常用传感器分为两类，一类是测量自身运动的传感器，如加速度计、陀螺仪、编码器

等，另一类是感知周围环境的传感器，如激光雷达、相机、声呐、超声波等，不同传感器获取信息的维度和丰富程度不同，SLAM 主要采用第二类传感器作为数据来源，其中激光雷达和相机应用较多。激光雷达能够直接测量环境中物体距离雷达距离，测量精度高，但价格昂贵，且体积较大；相比激光雷达，相机能够获取多维度信息且其价格相对低廉，因此成为研究 SLAM 问题优选的传感器，单目彩色相机能够获取物体的几何形状以及颜色信息，但不能直接获得物体到相机的距离，其需要通过多相机三角测量等方法间接求解物体到相机的距离。2014 年 Intel 推出了 RealSense 深度相机，可以获得带有物体距离信息的彩色图像，非常适用解决 SLAM 相关问题。本文通过对比市面几款主流深度相机，通过双目红外视差原理进行深度测量的 RealSense R200 相机，具有较好的测量精度和工作范围，并能有效降低光照影响。因此，本文选择 RealSense R200 作为室内移动机器人视觉 SLAM 传感器，设计实现了基于 RealSense 的视觉 SLAM 方法，为机器人室内定位与导航领域研究提供了有价值的参考。本研究得到了山东省重大科技创新工程项目“云基家庭服务机器人实用化关键技术研究及其产业化”（项目编号：2017CXGC0926）的资助。在研究课题期间，在优宝特智能机器人公司进行了试验。

1.2 视觉 SLAM 概述

机器人在未知环境中移动时，通过 SLAM 技术不断对自身进行定位并构建增量环境地图。其根据新观测到的环境信息与已构建的环境地图进行自定位，同时根据当前的定位结果来构建增量地图，其定位和构建地图是一个相互依存且相互影响的关系。在 SLAM 问题中，不能直接观测到机器人的位置信息。

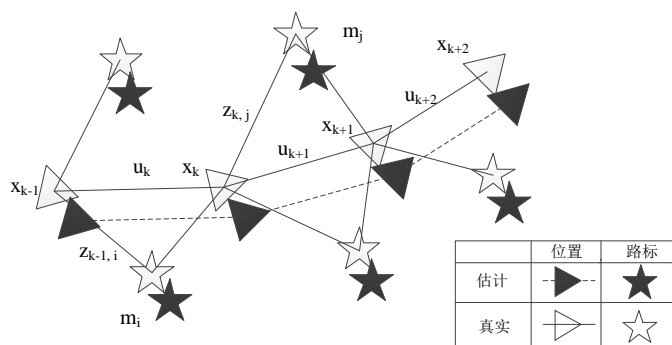


图 1-2 SLAM 方法状态图

Fig.1-2 SLAM method state diagram

如图 1-2 所示，机器人的位置状态需要利用观测的路标信息来确定，其中 x_k 表示 k 时刻机器人的位姿状态向量， m_i 表示第 i 个路标的位置状态向量， z_k 表示 k 时刻机器人的观测向量， u_k 表示机器人从 $k-1$ 时刻到 k 时刻的输入控制向量。SLAM 问题分解为定位和地图构建两个子问题，机器人利用里程计测量自身的运动状态，还需要根据观测外部环境信息，从而构建出自身所处未知环境中的地图。但是，由于里程计存在累计误差问题，随着时间增加，机器人的运动状态误差会不断变大，并且构建地图时依靠机器人的自定位，因此累计误差也会对地图构建产生不利影响；机器人自定位需要依靠构建出来的地图，地图中的观测误差也会影响机器人的自定位，所以 SLAM 的定位和地图构建是相互依赖的过程。因此，将机器人的位姿和路标都看作随机变量，把 SLAM 问题转化为两类随机变量的联合后验概率密度分布的最优估计问题。

SLAM 方法的发展经历两个阶段：基于滤波器的方法和基于图优化的方法。1987 年，Smith 等人首次提出 SLAM 方法，将 SLAM 构建为扩展卡尔曼滤波问题来求解机器人的位置和路标，通过数学建模把 SLAM 构造为一个运动方程和观测方程，以最小化两个方程中噪声的方式求解 SLAM 问题（Smith, 1987）。不过基于滤波器的 SLAM 方法存在线性化、更新效率低以及误差累计不可恢复等问题，无法应用到大规模环境 SLAM 中。Guivant 等人提出压缩扩展卡尔曼滤波的方法，解决了传统滤波器 SLAM 方法中的更新效率低下问题（Guivant, 2002）。随后专家学者们提出了基于图优化的 SLAM 方法，一般通过光束平差法的图优化理论迭代优化特征点的重投影误差，进而估算机器人位姿和路标。Strasdat 等人证明相同计算代价下，基于图优化的 SLAM 方法在精度方面远远超过基于滤波的 SLAM 方法，其可保证在大规模的环境中得到较好的位姿估算结果，并构建出可靠地图，因此基于图优化的 SLAM 方法现在已经成为主流方法（Strasdat, 2012）。

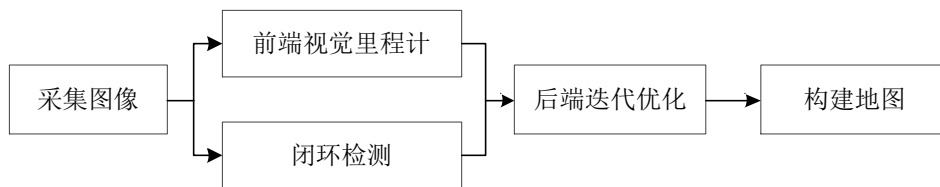


图 1-3 本文采用的视觉 SLAM 框架

Fig.1-3 This article uses the visual SLAM framework

基于图优化的视觉 SLAM 框架如图 1-3 所示，共分为五个环节：

（1）图像采集

通过相机采集图像作为 SLAM 系统的输入，相机种类一般有三类：单目、双目和深度相机，根据需求选择不同相机，本文采用 RealSense R200 深度相机同时采集环境彩色图像和深度图像。

（2）前端视觉里程计

根据相邻图像的信息计算机器人的运动位姿，为后端优化提供相机位姿初始值。本文采用基于特征的方法实现视觉里程计，首先提取图像特征，通过特征匹配建立数据关联，进而计算出机器人的变换位姿。

（3）闭环检测

闭环检测可以检测出自身之前是否到达过某个地点，闭环检测可以显著消除累计误差，并在视觉里程计追踪失败时，重定位机器人并校正运动轨迹和地图。

（4）后端迭代优化

为保证长时间运动轨迹和地图点的全局一致性和稳定性，通过图优化策略对全局相机位姿和地图空间点形成的约束进行优化，通过求解最小二乘法得到最优的相机位姿以及地图点。

（5）构建地图

SLAM 方法中地图具有两种含义，一种含义指的是能够实现机器人自定位的全局环境特征信息，机器人通过计算自身处在周围环境中的位姿来实现自定位；另一种含义是描述机器人所处环境细节的地图形式，本文根据全局优化后的相机位姿和地图点位置，首先构建出全局一致的点云地图，然后本文将点云地图转换为描述障碍物在空间中的占据信息的八叉树地图。

1.3 国内外研究现状

在 2003 年，牛津大学的 Davison 最早提出了视觉 SLAM 系统，并且构建了第一个在线实时的单目 SLAM 系统——MonoSLAM，该系统以扩展卡尔曼滤波为后端，追踪前端稀疏特征点，能够通过每秒 30 帧的视频流实现室内环境追踪并进行位姿估计（Davison, 2003），其运行效果如图 1-4 所示。2005 年，Sim 等人将粒子滤波法运用到 SLAM 研究中，将状态估计点使用粒子云的形式来表示，相机位姿和环境观测值都分别用粒子表示，根据更新当前时刻每个相机节点位姿和环境观测值来预测下一时刻

的相机姿态，粒子滤波算法处理速度比扩展卡尔曼滤波有了很大改善，其精度也有所提升（Sim，2005）。2007 年，Klein 等人提出了 PTAM 方法，构建了非线性优化为后端的实时 SLAM 系统，首次采用了位姿跟踪和构建地图双线程，能够完成实时的机器人定位和地图构建功能（Klein，2007）。



图 1-4 MonoSLAM 运行效果图

Fig.1-4 Operation effect diagram of MonoSLAM

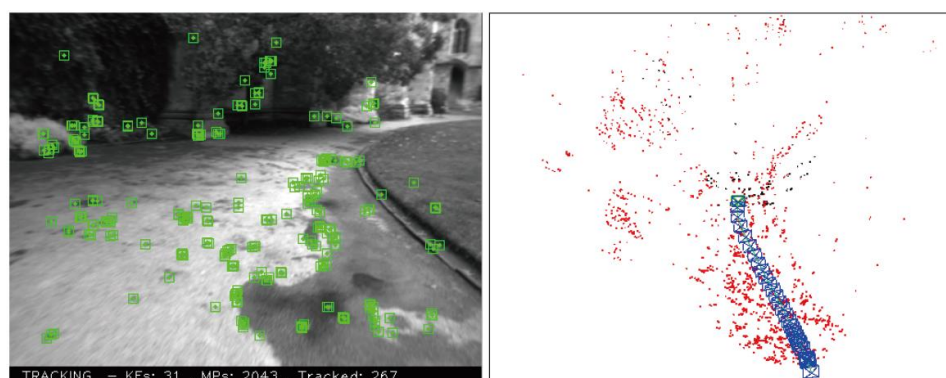


图 1-5 ORB-SLAM 运行效果图

Fig.1-5 Operation effect diagram of ORB-SLAM

2009 年，Vatani 等人利用智能车辆的单目相机采集环境图像，采用光流法获取前后时刻图像间的像素位移差，并计算车体的运动，实现了室外环境下连续序列帧 100m 的单目 SLAM（Vatani，2009）。2011 年，Geiger 等人使用双目相机实现了一系列图像序列帧的运动估计和三维地图构建，该算法通过卡尔曼滤波估计相机运动，建立了系数特征匹配器和高效的视觉测程法，且根据多视点连接方法生成了全局一致的点云地图，实现了定位功能以及三维重建功能（Geiger，2011）。2014 年，J.Engle 等人提出了 LSD-SLAM 改进算法，通过直接法追踪梯度变化明显的像素点来估计相机位姿，并构建半稠密地图，标志着单目直接法在 SLAM 中的成功应用（Engle，2014）。2015 年，Mur-Artal 等人在 PTAM 方法的基础上提出了 ORB-SLAM 方法，使用 Tracking、

Co-visibility Graph 和 Essential Graph 多线程结构完成 SLAM，实现了运动轨迹和地图的全局一致性（Mur-Artal, 2015），运行效果如图 1-5 所示，其多线程结构也被后续研究者普遍认同和采用，不过该方法只构建了稀疏地图，无法满足室内导航应用需求。

相对于国外研究，国内许多高校和科研机构也开展了机器人 SLAM 方面的研究，2004 年，浙江大学的徐则中在全协方差 SLAM 方案的基础上，通过重新构造系统状态向量，选用适当的观测向量得到一个基于线性系统模型的 SLAM 方案，并实现了室外道路环境下的同时定位与地图构建（徐则中，2004）。2007 年，天津大学的郭利进解决了大尺度未知环境下的数据关联问题，实现机器人在未知环境的自主导航并构建了自适应栅格地图（郭利进，2007）。2008 年，国防科技大学的陈伟提出的改进粒子滤波 SLAM 方法实现了月球车在非结构化环境中的定位与建图（陈伟，2008）。2012 年，东北大学的吴子章在采用 PCA-SIFT 特征结合全景相机实现改进 EKF 算法的 SLAM 方法（吴子章，2012）。2013 年，浙江大学章国锋等人对 PTAM 框架中的关键帧机制和双线程机制进行了改进，并提出了 RDSLAM 方法，实现了动态环境下地图构建（章国锋，2013）。2015 年，北京理工大学付梦印等人提出了一种实时的 SLAM 方法，得到机器人的 6D 位姿并构建环境的 3D 地图（付梦印，2015）。2016 年，清华大学研究人员采用光流传感器获取无人机的 6D 位姿，实现了无人机的室内定位和悬停，同时在 SLAM 中引入了深度学习技术，实现了高精确度的闭环检测功能（张连华，2016）。

1.4 本文主要研究工作

国内外专家学者提出许多 SLAM 方法，但也存在一些问题，因此本文在解决这些问题基础上设计实现基于 RealSense R200 深度相机的 SLAM 方法，论文的章节安排如下：

（1）第二章分析了视觉 SLAM 中运用到的相机模型及各坐标转换关系，为相机标定和 SLAM 中的数据关联提供图像基础理论支撑。

（2）第三章对 RealSense R200 参数进行简介，对比几种主流深度相机性能，分析深度相机的深度测量原理，使用 RealSense 进行数据采集和深度图像预处理，采用张正友标定原理标定 RealSense，校正相机畸变并得到相机内参用于位姿计算。

（3）第四章对 SIFT、SURF 和 ORB 特征提取理论进行分析，通过对比试验选取

适合本文 SLAM 方法的特征提取算法，采用区域分割算法改善特征提取均匀度；提出改进的 FLANN 分类树特征匹配算法，加速匹配并剔除匹配，试验验证了算法实时性；设计并实现了改进后的位姿计算方法；对数据关联中的闭环检测进行分析，设计实现了基于词袋模型的闭环检测方法，并采用了基于时间的闭环验证方法。

（4）第五章分析了基于图优化的 SLAM 方法，构建代价函数并通过求最小二乘问题优化全局机器人的位姿和路标点，分析图优化理论中的稀疏性问题；介绍了本文采用的地图构建方法。

（5）第六章对搭建的试验平台进行介绍，并设计实现完整的 SLAM 方法结构，通过 TUM 数据集和实时采集在线运行两种方式进行试验，试验验证了本文实现的 SLAM 方法的可行性。

（6）第七章总结了本文的工作内容，提出创新点，并对后续工作进行了展望。

2 视觉 SLAM 图像基础

视觉 SLAM 通过图像与真实世界的环境物体建立联系，其需要利用相机的内部参数，以及真实世界与图像的转换关系。相机成像是场景从世界坐标系投影到二维图像坐标系的过程，本章分析了视觉 SLAM 中运用到的相机模型及各坐标转换关系，为相机标定和 SLAM 中的数据关联提供图像基础理论支撑。

2.1 针孔模型

相机的成像模型基于生物视觉模型，通过机器视觉能够模拟生物视觉观察目标并获取信息（程一伦，2017）。相机成像的几何模型中最常用的是针孔模型，描述了一束光线通过针孔之后，在针孔背面投影成像的关系。

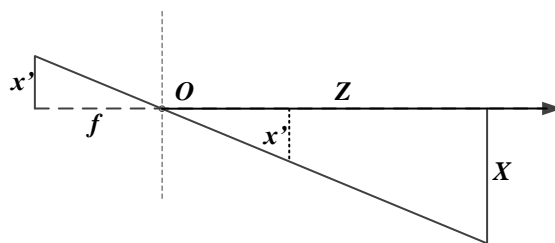


图 2-1 针孔模型示意图

Fig.2-1 Pinhole model diagram

针孔模型如图 2-1 所示，通过针孔 O 将三维世界中长度为 X 物体投影到成像平面，其模型根据三角形相似原理可知：

$$-\frac{X}{x'} = \frac{Z}{f}, \quad \text{同理可得,} \quad -\frac{Y}{y'} = \frac{Z}{f} \quad (2-1)$$

其中负号表示物体成像为倒立，将负号去掉简化公式：

$$\frac{X}{x'} = \frac{Y}{y'} = \frac{Z}{f} \quad (2-2)$$

针孔模型中物体距离相机光心的距离与相机焦距的比例，可用来表示物体在三维世界中真实尺寸与物体在成像平面上投影的比例，即建立了三维世界与二维投影世界的比例关系。整理可得，

$$x' = f \frac{X}{Z}, \quad y' = f \frac{Y}{Z} \quad (2-3)$$

2.2 相机坐标系

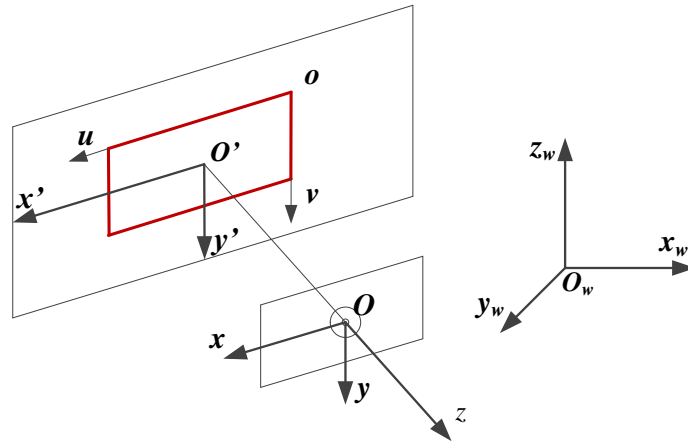


图 2-2 坐标系转换图

Fig.2-2 Coordinate transformation diagram

在视觉 SLAM 中主要用到四种坐标系：世界坐标系、相机坐标系、物理成像坐标系、图像像素坐标系。

(1) 世界坐标系

为描述客观物体与相机的位置关系，需要建立一种参考坐标系，即为图 2-2 中的 $O_w-x_w-y_w-z_w$ 坐标系，称为世界坐标系，其可以通过旋转矩阵和平移矩阵转换成相机坐标系，在 SLAM 系统中相机位置和路标位置统一由世界坐标系描述，其单位属于物理单位。

(2) 相机坐标系

根据针孔模型成像的原理，定义相机坐标系 $O-x-y-z$ 为以相机光心为坐标原点、 z 轴与光轴重合、 $O-x-y$ 平面与成像平面平行的直角坐标系，其中 OO' 为相机焦距。

(3) 物理成像坐标系

三维世界物体透过相机光心映射到物理成像平面，物理成像平面与相机光轴相交点作为物理成像坐标系 $O'-x'-y'$ 的坐标原点，物理成像坐标系的 x 轴、 y 轴分别与相机坐标系的 x 轴与 y 轴平行且方向一致。

(4) 图像像素坐标系

物理成像平面的图像在计算机中的存储单位为数字图像，其为 $M \times N$ 数组，每行每列组成元素为像素。图像像素坐标系 $o-u-v$ 以图像的右上角为坐标原点， x 轴、 y 轴

分别于物理成像坐标系平行且方向一致。

2.3 坐标变换

(1) 世界坐标系与相机坐标系

世界坐标系对三维世界中物体进行客观描述，当相机对物体进行拍摄时，仅在相机坐标系中描述了相机与物体的相对位置，但要对 SLAM 系统中的所有路标和相机位置进行描述，必须统一到世界坐标系中。世界坐标系到相机坐标系的变换是刚性变换，表示相机坐标系相对于世界坐标系的位姿。假设某空间点 P 在世界坐标系中的坐标为 (X_w, Y_w, Z_w) ，在相机坐标系中的坐标为 (X_c, Y_c, Z_c) ，如式 2-4 所示，其变换关系可表示为：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t \quad (2-4)$$

其中， R 为 3×3 旋转矩阵， t 为 3×1 的平移矢量：

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (2-5)$$

将空间点 P 的在世界坐标系和相机坐标系中表示为齐次方程坐标，其变换关系可以用公式 2-6 来表示：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = T_{4 \times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2-6)$$

其中，矩阵 T 为外参矩阵，其值会随着相机运动发生改变，在 SLAM 系统中，矩阵 T 是估算相机运动轨迹的目标。

(2) 相机坐标系与物理成像坐标系

相机坐标系到物理成像坐标系的转换就是针孔模型成像变换，与式 2-3 基本一致，这里表示为它的齐次坐标形式，如式 2-7 所示：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \frac{f}{Z_c} \begin{bmatrix} X_c \\ Y_c \\ \frac{Z_c}{f} \end{bmatrix} = f \begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ \frac{1}{f} \end{bmatrix} \quad (2-7)$$

(3) 物理成像坐标系与图像像素坐标系

图像像素坐标系与物理成像坐标系之间变换需要将物理成像坐标缩放，且将坐标原点平移。将物理成像坐标的 x 轴缩放 α 倍形成图像像素坐标系的 u 轴，同理 y 轴缩放 β 倍形成 v 轴，坐标原点平移 $[c_x, c_y]^T$ ，最终变换成图像像素坐标系。其齐次坐标如式 2-8 所示：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (2-8)$$

(4) 相机坐标系与图像像素坐标系

从相机坐标系变换到图像像素坐标系，即从三维转换成二维数字图像的过程，由式 2-7 和式 2-8 可以得到：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = f \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{X_c}{Z_c} \\ \frac{Y_c}{Z_c} \\ \frac{1}{f} \end{bmatrix} \quad (2-9)$$

式 2-9 中 f 乘上 α 合并成 f_x ， f 乘上 β 合并成 f_y ，式 2-9 可化为：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \frac{1}{Z_c} K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2-10)$$

式 2-10 中矩阵 K 即为相机的内参矩阵，由标定得出。

(5) 世界坐标系与图像像素坐标系

根据上述变换关系，可知世界坐标系到图像像素坐标系的变换为，其中， s 为尺度因子（这里等同于相机坐标系下的 Z_c ）。

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2-11)$$

在 SLAM 的前端视觉里程计中，根据图像坐标系与世界坐标系的转换关系，求解相机位姿时即求解式 2-11 中的旋转矩阵 R 、平移向量 t 。

2.4 本章小结

对视觉 SLAM 方法运用到的图像基础进行介绍，对 RealSense 相机的彩色相机的成像模型进行分析，对 SLAM 方法中计算运用到的相机坐标系及它们的转换关系进行了阐述，为 RealSense 标定工作和 SLAM 数据关联中计算机器人位姿提供理论基础。

3 RealSense 标定

对 RealSense R200 参数进行简介, 对比几种主流深度相机性能, 分析深度相机的深度测量原理, 在 Ubuntu 平台采集 RealSense 的图像数据, 结合第二章介绍的相机模型和坐标转换关系, 以及张正友标定原理, 对 RealSense R200 相机进行标定试验, 对 RealSense 相机中的彩色相机进行了畸变校正, 并取得彩色相机内部参数, 对两个红外相机进行标定取得变换矩阵, 配准相机的彩色图像和深度图像, 并采用基于时间戳的方法对齐彩色图像和深度图像。

3.1 RealSense 简介

RealSense R200 是 Intel 公司在 2014 年在 CES 展览会上推出的一款智能深度相机, 图 3-1 描述了 RealSense 的外观。相比于同类产品, RealSense 拥有应用距离更远的深度传感器和 1080P 更高分辨率色彩传感器。深度相机与传统彩色相机相比, 大多数彩色相机将真实三维世界投影到二维平面, 仅能获取 RGB (红、绿、蓝) 彩色信息, 因而缺失了真实世界的深度信息。RealSense R200 深度相机, 能够同时输出彩色视频流和深度视频流, 将三维世界信息立体呈现。RealSense R200 主要适用于远距离场景 (室内 0.6~3.5 米, 室外 3~15 米), 深度图像最高分辨率为 640×480 , 色彩分辨率最高 1080P, 水平视角 59° , 垂直视角 46° 。



图 3-1 RealSense R200 外观图

Fig.3-1 Appearance of RealSense R200

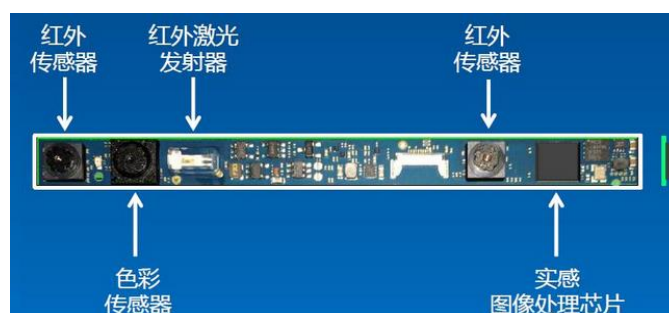


图 3-2 RealSense R200 内部构造图

Fig.3-2 Internal structure diagram of RealSense R200

如图 3-2 所示, RealSense R200 包含内置的左右两个红外传感器、色彩传感器, 红外激光发射器, 实感图像处理芯片等主要部件。其利用双目视差原理进行两路红外传感器接收, 实时计算出每个像素点的深度信息, 从而提高远距离测距的准确性, 色彩传感器获得正常的高质量 RGB 信息。

为满足室内机器人完成同时定位与地图构建的需求, 需要选择最适合运动机器人的深度相机, 表 3-1 分别从深度探测距离、深度帧率、深度分辨率、彩色分辨率、视场范围、外形尺寸、USB 接口、有无外接电源及价格等方面详细对比了 Kinect V1、Kinect V2、Xtion、RealSense 这几种 RGB-D 相机的性能。

表 3-1 不同型号深度相机数据对比表

Tab.3-1 Different model depth camera data comparison table

指标	Kinect V1	Kinect V2	Xtion	RealSense R200
深度探测距离/m	0.8~4.0	0.5~4.5	0.8~3.5	0.6~3.5(室内)
深度帧率/fps	30	30	30	60
深度分辨率(最高)	320×240	512×424	640×480	640×480
彩色分辨率	640×480	1920×1080	320×240	1920×1080
视场范围(V×H)/°	43×57	60×70	43×57	46×59
外形尺寸/mm	282×63×70	249×66×67	180×35×500	130×20×7
接口	USB2.0	USB3.0	USB2.0	USB3.0
外接电源	有	有	无	无
价格/RMB	1000	1500	1500	1250

综合多款相机的参数, 从深度探测距离上, Kinect V2 与 RealSense R200 的近景探测距离最近, 分别为 0.5m、0.6m, 远景探测距离(仅室内)最远的是 Kinect V2 和 Kinect V1; 深度帧率最多的是 RealSense R200, 最高帧率达 60fps, 其他相机均为 30fps, 相同时间内得到的场景信息更多, 能满足更高速的运动需求; 在深度分辨率上占据优势的是 Xtion 和 RealSense R200, 其能够提供更加精细的图像深度信息; 彩色分辨率方面, Kinect V2 和 RealSense R200 均能提供 1080P 的超清画质, 二维信息更丰富, 特征更加明显; Kinect V2 和 RealSense R200 的视场探测范围最大, 提供更好的视野; RealSense R200 的外形尺寸最小, 仅为其他相机的十几分之一大小, 便于装载到机器人; 支持的 USB 接口方面, Kinect V1 和 Xtion 支持 USB2.0, 而 Kinect V2

和 RealSense R200 支持 USB3.0，能提供更快速的高清视频传输性能；Xtion、RealSense R200 均无需外接电源，使用更加灵活。综上所述，RealSense R200 各方面性能较优，且价格低廉，因此选择它作为本文研究室内机器人 SLAM 的深度视觉传感器。

3.2 深度原理

深度相机测量深度常用方法主要有：光编码法、飞行时间法、双目视差法。首先简单介绍光编码法和飞行时间法的基本原理，然后重点分析 RealSense R200 的深度测量原理。

（1）光编码法

如 Kinect V1 这类深度相机探测深度采用的是光编码技术，光编码技术是一种特殊的结构光测量技术，它与传统的结构光方法不同，光编码技术使用红外投影仪投射的光源并不是周期性变化的二维图像编码，而是具有三维纵深的激光散斑（laser speckle），激光散斑具有高度随机性，且会随着距离不同改变图案，即空间中任意两处的散斑图案都不同（李智，2016）。基于激光散斑的高度随机特征，采用空间标记方法，将空间分割为若干等距参考平面，记录不同平面的激光散斑和其对应的深度距离，便可以得到整个空间散斑和空间物体距离的对应关系。当物体处于已标定距离的空间中，通过 CMOS 感光传感器获取物体表面的激光散斑图案，从而得出物体的深度距离信息（龙超，2016）。

（2）飞行时间法

飞行时间法，其工作原理是深度相机连续发射光脉冲到被观测目标场景，光脉冲经过物体反射后被接收器接收，通过计算光脉冲发射和反射接收的往返（飞行）时间来计算深度相机到被观测场景的距离，从而获得被观测场景深度信息。

（3）双目视差法

双目视差测距原理是通过模仿人眼的立体视觉，利用双摄像机对同一场景下物体进行拍摄得到不同视角下两幅图像，根据图像之间像素的匹配关系计算像素偏移量，已知两摄像机相对位置及内部参数，通过三角测量原理计算出物体的三维信息。传统双目视觉技术很大程度上依赖于环境中的纹理，在纹理稀少或无纹理环境中效果不佳。RealSense R200 深度相机采用的是主动式的近红外双目视觉技术，它采用一个红外激

光发射器投射近红外光来增加红外纹理，两个红外传感器采集两张近红外图像，实感图像处理芯片通过对两幅近红外图像进行图像匹配计算视差图，得到整个场景的三维深度信息，进而生成与彩色图像对应的深度图像。

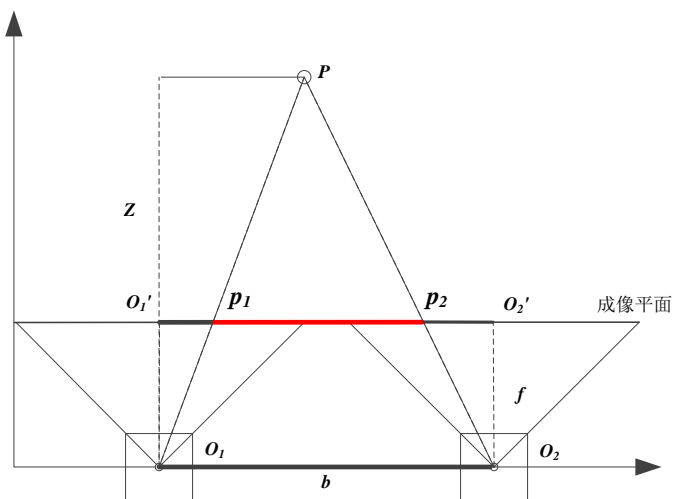


图 3-3 双目视差原理图

Fig.3-3 Schematic diagram of binocular parallax

双目视差法中双目图像匹配的像素一定在图像中的同一行中，即需满足极线约束（也称为对极几何），但多数双目相机采集到的双目图像对因相机实际位置的偏差无法满足极线约束，其同一物体对应的像素点在两幅图像中会存在垂直视差，因此在运用双目视差法求解拍摄场景下物体三维信息之前，需对双目图像进行极线校正（张学贺，2016）。双目视差法的原理如图 3-3 所示，两相机内参相同， f 为相机焦距， O_1 、 O_2 为两个相机的光学中心，连线 O_1O_2 表示基线，长度为 b ， $O_1'O_2'$ 所在平面为成像平面， P 为特征空间点， p_1 、 p_2 为两个相机在成像平面的成像点，空间点 P 到基线的距离为深度 Z 。

设左相机成像平面坐标系下 P_1 点的横坐标为 $O_1'p_1 = x_1$ ，右相机成像平面坐标系下 P_2 点的横坐标为 $O_2'p_2 = -x_2$ ，根据三角形相似原理，可得：

$$\frac{b - x_1 + x_2}{b} = \frac{Z - f}{Z} \quad (3-1)$$

整理为：

$$Z = \frac{fb}{d}, \quad d = x_1 - x_2 \quad (3-2)$$

式 3-2 中 d 为左右成像点的横坐标之差，称为视差（Disparity），焦距 f 基线 b 是双目相机的内部参数。式中可以看出，视差越小，空间点深度越大。当视差达到最小值（整数）时，双目相机的测量深度会取得一个最大值，其值由 fb 确定。焦距 f 值固定，基线 b 越长，双目相机能够测量的距离就越远。

RealSense R200 相机采用编码散斑辅助的双目视差法测量深度，在纹理较少情况下依然能够保持较好的精度，相比起传统双目彩色相机具有很大的优势；相比单纯光编码技术的深度相机，其对光照的抗干扰能力更强。

3.3 数据采集和预处理

（1）开发环境搭建

本文研究都是在 Ubuntu 系统上进行，在 ROS（Robot Operating System，机器人操作系统）框架下使用 RealSense R200 深度相机采集数据，其开发环境搭建步骤如图 3-4 所示。

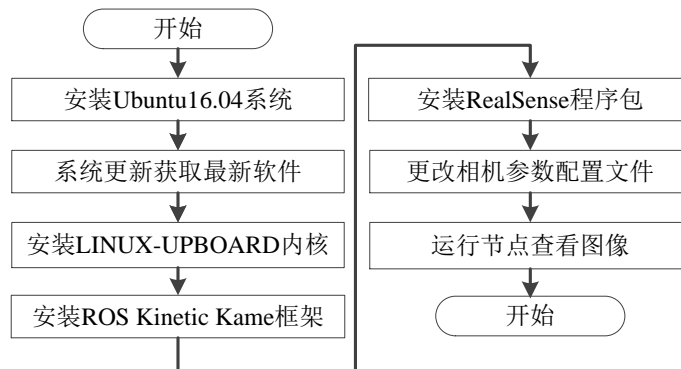


图 3-4 开发环境搭建流程图

Fig.3-4 Development environment construction flow chart

（2）使用 Rviz 查看数据

在计算科学和计算机图形学中，人们对真实领域中的信息进行可视化存在强烈的要求，并且提出了许多可用于可视化的数据结构和算法来实现这个目标。Rviz 就是基于这个目标的一个 3D 可视化工具，它工作在 ROS 框架中，基本上关于机器人的所有数据都可以在 Rviz 上进行可视化展示出来（Kam，2015）。使用 Rviz 可以对视频流进行实时查看，如图 3-5 所示，步骤分为两步。

第一步，运行节点启动 RealSense R200 深度相机，将相机的彩色视频流和深度视频流定义为消息（Message），通过发布主题（Topic）把消息广播到 ROS 网络中。

第二步，启动 Rviz 通过订阅主题，接收并解耦相机节点发布的消息，通过视频的形式显示，图 3-6 是使用 RealSense 在实验室采集的数据。

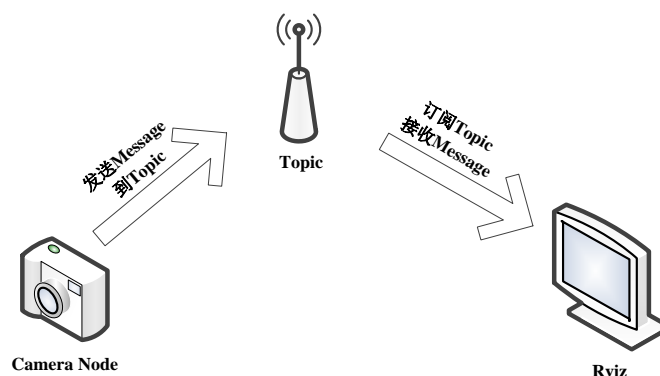


图 3-5 ROS 网络传输示意图

Fig.3-5 ROS network transmission diagram

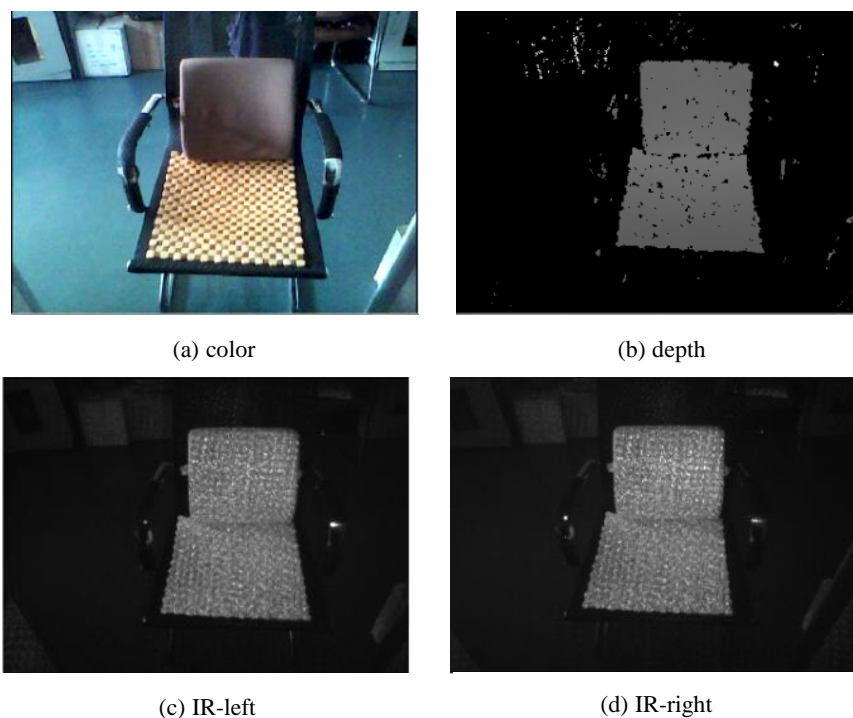


图 3-6 采集图像

Fig.3-6 Collecting images

（3）基于时间戳的图像配准

RealSense R200 在出厂时已经做好彩色图像和深度图像的在空间上的配准工作，对于同一时刻采集、相同分辨率的彩色图像和深度图像来说，每个坐标点的颜色亮度

和深度值是一一对应的，无需在使用相机的过程中去繁琐校正两者的配准。不过在数据传输的过程中会出现时间上的不同步现象，经分析是因为图像的格式差异和图像文件大小不同，会导致传输速率不同，这给 SLAM 方法中同时利用彩色图像和深度图像信息带来了困难。

为解决这一问题，本文采用了基于时间戳的图像配准方法，如式 3-3 所示， T_c 和 T_d 分别表示彩色图像和深度图像的采集时间，处理彩色图像和深度图像时，只有当彩色图像和深度图像的时间戳差值的绝对值小于阈值 θ 时，认为彩色图像和深度图像是同一时间采集，这个时候彩色图像的像素点和深度图像像素点是对齐的。

$$f(t) = \begin{cases} 1 & |T_c - T_d| < \theta \\ 0 & \text{else} \end{cases} \quad (3-3)$$

(4) 深度图像预处理

针对原始深度图像出现的噪声、空洞问题，分析加权均值滤波、中值滤波、高斯滤波和双边滤波等方法对深度图像的去噪和修复作用。加权均值滤波方法通过对像素点一定区域内的深度值进行加权求均值，会导致图像整体像素的深度值发生变化，且会平滑边缘部分的深度值，导致边缘信息的丢失；中值滤波通过对像素点一定区域内的深度值去中值来代替当前像素点，能够去除小的噪点，对大的空洞无效，但会平滑边缘；高斯滤波的邻域深度值的权值随距离增加而递减，对颗粒噪声平滑效果明显，也会平滑掉边缘深度值；双边滤波方法考虑了深度值之间的相似度，是对高斯滤波方法的改进，可以避免边缘部分深度值的缺失。因此选用双边滤波方法来对原始深度图像进行预处理。

3.4 RealSense 标定试验

相机标定是计算机视觉的首要解决问题，相机标定结果的准确度对图像像素点和三维空间点之间的映射转换精度影响至关重要。标定目的是得到高精度的相机内部参数：相机焦距 $[f_x, f_y]$ ，主像点坐标 $[c_x, c_y]$ ，以及实际使用时因设计、制造及装配误差引起的相机畸变参数。在 SLAM 系统中，对路标三维特征点以及相机运动轨迹的求解都会用到标定出的相机内部参数。

本文采用张正友相机标定法（Zhang, 2000）对 RealSense R200 相机进行标定，因棋盘格易于提取角点，使用棋盘格标定板作为标定模板，棋盘格尺寸为

180mm×140mm，其方形格边长为 20mm。张正友标定方法的原理是采集多幅不同、不同距离的图像，然后提取图像中的棋盘格角点，根据图像角点间的对应关系求解每幅图像的单应性矩阵，从而计算出相机的内外参数。张正友标定方法基本原理如式 3-4 所示：

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3-4)$$

式 3-4 中 $[u, v, 1]$ 是图像像素点齐次坐标， s 为尺度因子， K 为相机内参矩阵， $[r_1, r_2, r_3, t]$ 是相机外参旋转矩阵和平移向量，世界空间点的齐次坐标为 $[X, Y, Z, 1]$ 。假定标定模板所在平面为世界坐标系的 $Z = 0$ 平面上，将 $[r_1, r_2, r_3, t]$ 简化为 $[r_1, r_2, t]$ ， $[X, Y, 0, 1]$ 简化为 $[X, Y, 1]$ 。可以表示为：

$$sm = HM \quad (3-5)$$

式 3-5 中 $m = [u, v, 1]^T$ ， $M = [X, Y, 1]^T$ ，单应矩阵矩阵 H 如式 3-6 所示：

$$H = [h_1 \ h_2 \ h_3] = \alpha K [r_1 \ r_2 \ t] \quad (3-6)$$

式 3-6 中 $\alpha = 1/s$ ，因旋转矩阵具有正交性，所以 $r_1^T r_2 = 0$ ， $\|r_1\| = \|r_2\| = 1$ ，每幅图像可以得到：

$$h_1^T K^{-T} K^{-1} h_1 = 0 \quad (3-7)$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2 \quad (3-8)$$

当采集图像数量大于 3 张后，通过式 3-7 和式 3-8 就可以求解出线性唯一解，得到相机焦距及坐标主像点。实际使用时相机成像过程中会出现畸变，因此需要求解相机的畸变参数，相机畸变以径向畸变为主（刘艳，2014），所以本文求解一阶和二阶径向畸变。求解原理如式 3-9 所示：

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2) \\ (v-v_0)(x^2+y^2) & (u-u_0)(x^2+y^2) \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} u_s - u \\ u_s - v \end{bmatrix} \quad (3-9)$$

式 3-9 中 (u, v) 为理想像素坐标， (x, y) 为理想物理成像平面坐标， (u_s, v_s) 为发生畸变后实际的像素坐标， (k_1, k_2) 为 2 阶径向畸变参数（王勇，2018）。式 3-9 与式 2-8、

2-10 联立求解，利用最小二乘法理论，当误差引起的代价函数达到最小值时，得到最优畸变参数。

根据上述原理，在学实楼 203 实验室进行了标定试验。标定设备为计算机、RealSense R200、平直木板、棋盘格打印纸。标定步骤为：

- (1) 根据上述原理通过 Python 及 OpenCV 库来实现标定程序的编写；
- (2) 为保证棋盘格精度，通过 CAD 绘制棋盘格图像并打印，贴于平板上；
- (3) 启动相机标定程序采集标定图像，调整棋盘格采集不同角度、不同距离标定图像，目的是得到不同位姿的棋盘格图像，建立单应性矩阵；
- (4) 去除采集的标定图像中产生运动模糊和过度重复的图像，对选定的标定图像提取角点；
- (5) 为提高标定精度，降低标定偏差，对每一幅标定图像进一步提取亚像素角点，构建单应性矩阵计算理想状态下的相机内参和畸变参数，及相机外参。
- (6) 采用最小二乘法估计实际情况存在的径向畸变参数，并优化估计精度。
- (7) 对标定结果进行评价，通过求解出的相机内外参数对空间的三维点进行重投影，计算出重投影偏差，偏差越小时标定结果越好，最后取各投影偏差的均值。

根据上述步骤分别对 RealSense 内部的彩色相机和两个红外相机进行标定，标定彩色相机和红外相机的过程是独立的，因此图 2-9 和图 2-10 中的彩色图像和红外图像并不对应。标定左右两个红外相机是同时标定的，因此图 2-10 中所示图像是两两一对同时采集的左视和右视图像。

首先对彩色单目相机进行标定，一共采集了 58 幅不同距离及不同旋转角度的图像，图 3-7 中图像是标定试验过程中其中 16 幅图像，根据上述标定步骤得到的相机标定结果如表 3-2 所示。

表 3-2 彩色相机的内部参数

Tab.3-2 Color camera internal parameters

标定参数	结果
焦距(f_x, f_y)	[618.195268, 621.185850]
主像点(c_x, c_y)	[340.920665, 238.603545]
径向畸变(k_1, k_2, k_3)	[0.099860, 0.406082, 0.004511]

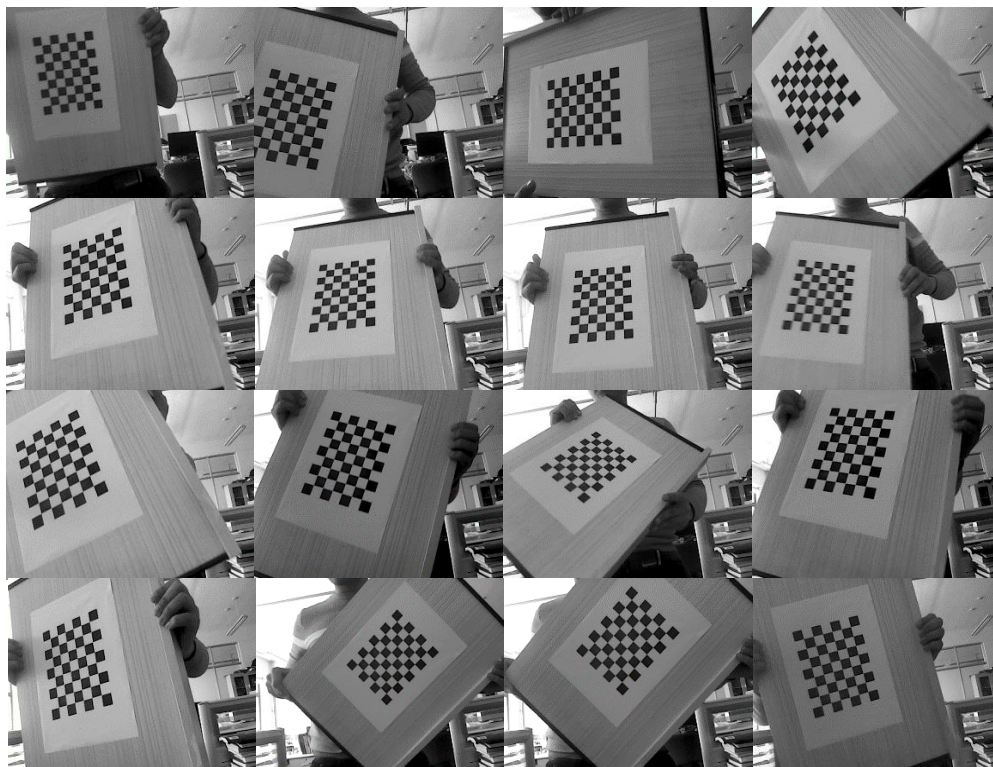


图 3-7 标定的彩色图像

Fig.3-7 Calibrated color image

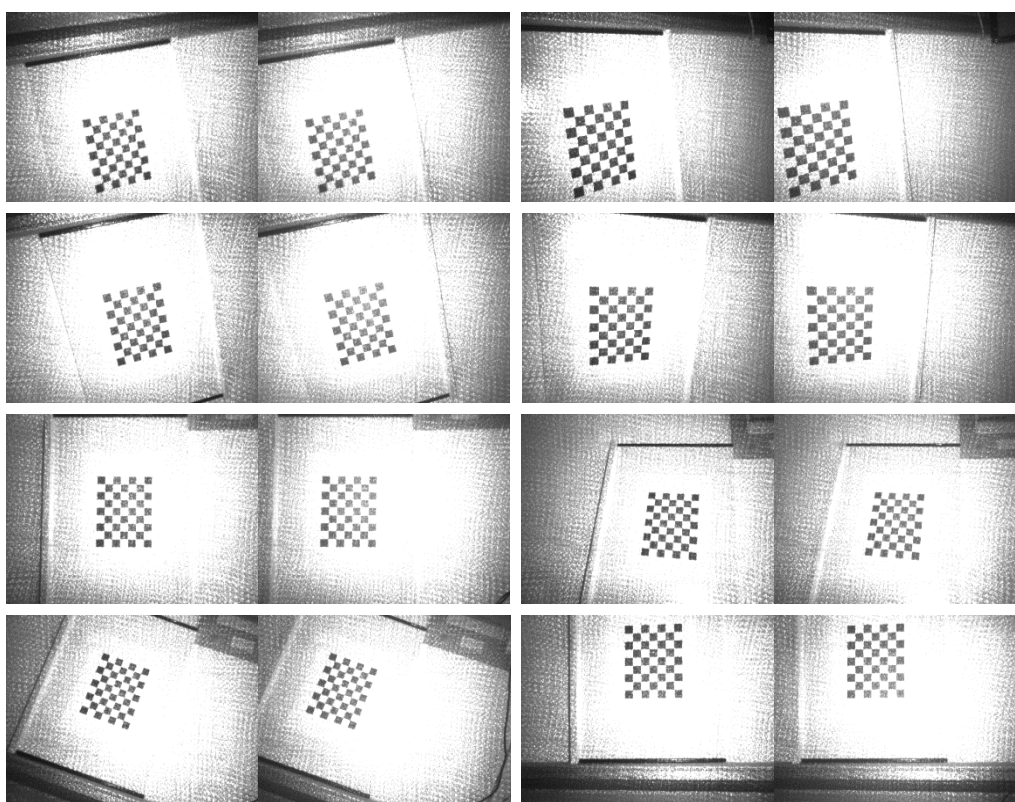


图 3-8 标定的双目红外图像

Fig.3-8 Calibration of two camera infrared images

然后对 RealSense R200 相机的双目红外相机标定，如图 3-8 所示，这是其中 8 对红外标定左右图像，通过标定分别得到两个相机的内参和畸变参数，以及右相机到左相机的旋转矩阵和平移向量，其结果如表 3-3 所示。

表 3-3 双目红外相机的内部参数

Tab.3-3 Internal parameters of two infrared cameras

标定参数	左红外相机	右红外相机
焦距(f_x, f_y)	[584.514020, 582.453866]	[585.855589, 584.756791]
主像点(c_x, c_y)	[311.827440, 243.321308]	[306.124328, 253.199401]
径向畸变($k1, k2, k3$)	[0.003096, 0.040565, -0.001839]	[-0.009004, 0.077531, 0.006435]
平移向量	[-0.069286, -0.000189, 0.001644]	
旋转矩阵	$\begin{bmatrix} 0.999964 & -0.001051 & 0.008383 \\ 0.001202 & 0.999838 & -0.0179838 \\ -0.008364 & 0.017993 & 0.9998031 \end{bmatrix}$	

通过三维点重投影的误差评价方法对标定结果进行评价，标定误差小于 0.1 个像素，精度较好满足后续使用要求。最后将彩色相机和双目红外相机的标定结果保存成 yaml 格式文件，输入到相机的配置文件中，由相机内部图像处理芯片完成深度图像的配准矫正。标定相机的目的除了矫正相机外，主要目的是为了得到 RealSense R200 中彩色相机的内部参数，用来完成图像坐标系与世界坐标系之间的坐标变换，将图像中反映的几何关系转换到世界坐标系中。

3.5 本章小结

介绍了 RealSense R200 的结构和性能参数，并对几种主流深度相机性能进行了对比，研究分析三种深度测量原理，搭建开发环境采集图像数据并分析深度图像的预处理方法，提高深度图像质量，根据相机标定原理，完成对 RealSense R200 彩色相机和双目红外相机的标定试验，通过计算重投影误差得出标定误差在 0.1 个像素之内，达到使用要求，通过畸变校正提高了 RealSense R200 的彩色图像采集质量，并得到彩色相机的内部参数用于 SLAM 方法的机器人位姿计算。

4 SLAM 数据关联

SLAM 方法中的数据关联，目的是实现机器人对自身的追踪定位。数据关联分为两部分，一部分是前端视觉里程计，称为顺序数据关联，通过连续时间里追踪图像特征点完成；另一部分是闭环检测环节，称为环形数据关联，能够检测机器人是否到达已去过的场景，当检测成功时能够消除累计误差，并且当前端视觉里程计追踪特征点失败时，闭环检测通过检测最新图像与已有关键帧图像的相似性来完成机器人的重新定位。

相机标定的 RealSense 相机的内部参数，可用于数据关联中的机器人运动位姿计算。数据关联的基础是图像的相似度或共视关系，通过图像间匹配特征点来完成数据关联。本章对 SLAM 方法中的数据关联部分进行研究设计，对其中相应算法出现的问题引入解决方案并做出改进。

首先，本章对 SLAM 方法中的前端视觉里程计进行算法设计与实现。为保证数据关联的准确性，需要选取合适的图像特征提取算法，特征点应具有光照不变性、旋转不变性、尺度不变性；采用基于特征匹配的方法进行数据关联，然后计算出机器人不同时刻下的位姿，也就得到机器人的连续运动轨迹，实现机器人在周围环境中的定位。接下来对 SLAM 方法中的闭环检测进行研究与实现，闭环检测依赖于图像相似度的检测，在本章研究了基于词袋模型的闭环检测方法，实现了消除累计误差和重定位目的。

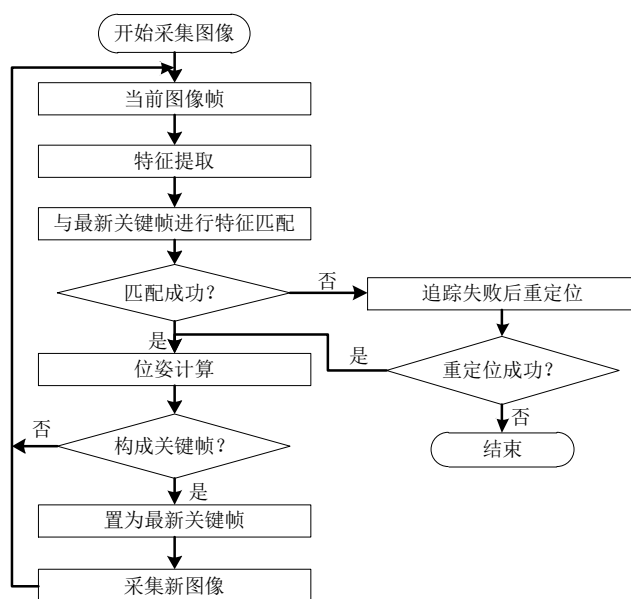


图 4-1 视觉里程计算法流程图

Fig.4-1 Flow chart of vision odometry algorithm

本文研究视觉里程计的算法流程如图 4-1 所示，首先对当前图像帧进行特征提取，当前图像帧 I 与最新关键帧 I_k 存在共视关系时，进行特征匹配（没有关键帧时第一幅特征点丰富的图像作为关键帧），匹配成功时根据计算出机器人的变换位姿；当新采集的图像帧 I 满足成为关键帧的条件时，将其置为最新关键帧 I_k ；当前图像帧 I 不断与最新关键帧 I_k 进行特征匹配并计算位姿，这样便计算出了机器人连续的运动轨迹；同时不断将关键帧中的特征点构造 3D 特征点变换到世界坐标系中，这样就构成了稀疏的特征地图点。

4.1 特征提取算法对比改进

特征点（也称为角点）为图像周围亮度变化剧烈的点或者图像边界曲线上具有足够大曲率的点。特征点大量存在于图像中，在不同的尺度空间以及不同的空间位置，相同的点特征都能够被准确提取检测到。成熟的特征点检测方法在尺度变化、旋转变换、平移透视及光照变化情况下能够保持一致性。其广泛应用于运动检测、视频跟踪、图像匹配、目标识别和三维建模等领域中，其中性能较优的特征提取算法有 SIFT、SURF、ORB 等，本章在研究分析三种特征提取算法的原理基础上，通过特征提取试验对比了三种算法的性能，选择适合 SLAM 实时性运行的特征提取算法，并引入了图像区域分割的方法改进特征提取的均匀度，减少了因特征点集中引起的误匹配及匹配失败的概率。

4.1.1 特征提取算法原理分析

（1）SIFT 特征

1999 年 David 提出 SIFT（Scale Invariant Feature Transform，尺度不变特征变换）特征提取算法，并于 2004 年将 SIFT 完善总结（David，2004）。其广泛应用于物体辨别、机器人地图感知与导航、图像检索匹配等。SIFT 特征是图像的局部特征，其对尺度缩放、旋转角度、视角变化及亮度变化保持不变性。

1) 构建尺度空间

首先对原始图像进行平滑后进行降采样，得到多分辨率的图像金字塔，在图像金字塔的基础上进行高斯卷积生成不同尺度空间，将图像的尺度空间表示成函数 $L(x, y, \sigma)$ ，它是由一个变尺度的高斯函数 $G(x, y, \sigma)$ 与图像 $I(x, y)$ 卷积产生的，表示为：

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4-1)$$

$G(x, y, \sigma)$ 是尺度可变高斯函数：

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (4-2)$$

如图 4-2 所示，构建尺度空间的目的是为了检测出在不同尺度下都存在的特征点，采用高斯差分金字塔的形式来得出，可表示为：

$$\begin{aligned} D(x, y, \sigma) &= [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (4-3)$$

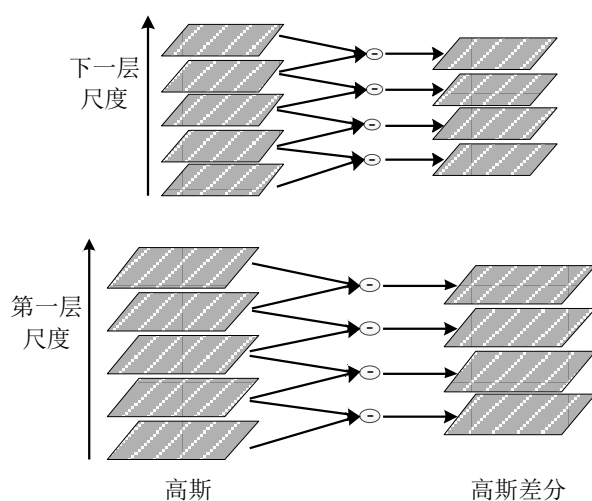


图 4-2 差分高斯空间

Fig.4-2 Differential Gaussian space

2) DOG 空间极值检测

为检测尺度空间的极值点，每个像素点要和同一尺度空间及上下两个相邻尺度空间的所有相邻点进行比较，当其大于（或小于）所有相邻点时，该点就是极值点。剔除低对比度的极值点及不稳定的边缘响应点，得到在不同尺度下都存在的特征点。

3) 求取特征点主方向

为实现图像旋转不变性，给特征点的方向进行赋值，利用特征点邻域像素的梯度分布特性确定其方向参数，通过图像的梯度直方图求取特征点局部结构的稳定方向。

4) 生成特征描述

对于每个特征点可以得到三个信息 (x, y, σ, θ) ，即位置、尺度和方向。为了保证特征矢量的旋转不变性，以特征点为中心，在附近邻域内将坐标轴旋转 θ 角度，即将坐标轴旋转为特征点的主方向。旋转后邻域内像素新坐标为：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4-4)$$

旋转后以主方向为中心取 8×8 的窗口。如图 4-3 所示，左图的中央为当前特征点的位置，每个小格代表为关键点邻域的一个像素，求取每个像素的梯度幅值与梯度方向，然后利用高斯窗口对其进行加权运算。最后在每个 4×4 的小块上绘制 8 个方向的梯度直方图，计算每个梯度方向的累加值，即可形成一个种子点，对每个特征点使用 4×4 共 16 个种子点来描述，这样一个特征点就可以产生 128 维的 SIFT 特征向量。邻域方向性信息联合增强了算法的抗噪能力，保持了良好的尺度和旋转不变性。

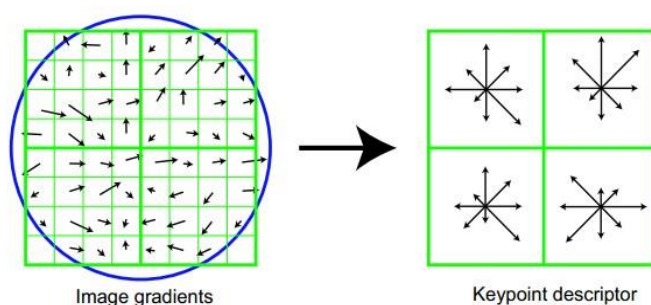


图 4-3 特征描述邻域

Fig.4-3 Feature Description Neighborhood

图 4-4 是在山东农业大学学实楼 203 实验室利用 RealSense 采集的一幅彩色图像，通过算法提取图像中的 SIFT 特征点，这里是特征点数量阈值设为 500 时的效果图。

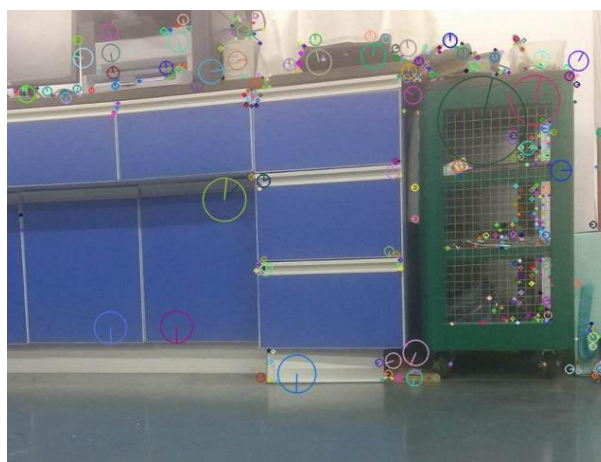


图 4-4 SIFT 特征提取效果图

Fig.4-4 SIFT feature extraction effect diagram

(2) SURF 特征

2006 年，Bay 等人提出了 SURF 特征提取算法（Speeded Up Robust Feature，加速鲁棒特征）（Bay，2006），优化了 SIFT 特征提取算法，SURF 采用统计特征点 Haar

小波特征的方式代替梯度直方图，利用积分图的技巧卷积图像进行快速计算，缩短特征提取的时间。

1) Hessian 矩阵构建

首先利用积分图的方式实现对图像矩形面积的快速计算，积分图 $I_{\Sigma(x, y)}$ 表示 (x, y) 处到原点区域内所有像素和。图像中所有位置的积分图像计算出得值，那么图像中任意矩形区域内像素点求和 Σ 运算只需三步加减运算就可以得到：

$$I_{\Sigma(x, y)} = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x, y) \quad (4-5)$$

如图 4-5 所示，已计算出 A 、 B 、 C 和 D 点的积分图，矩形的面积为：

$$\Sigma = A + D - B - C \quad (4-6)$$

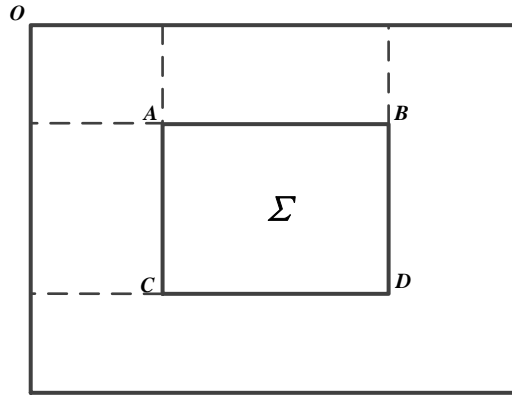


图 4-5 积分图计算方式示意图

Fig.4-5 Integral diagram calculation method

计算出积分图后，通过构建 Hessian 矩阵检测特征点，Hessian 矩阵在图像 I 中的 (x, y) 处、尺度为 σ 的定义：

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \quad (4-7)$$

式 4-7 中 $L_{xx}(x, y, \sigma)$ 等是高斯二阶偏导数对图像 I 在 (x, y) 处的卷积：

$$L_{xx}(x, y, \sigma) = \frac{\partial^2 g(\eta)}{\partial x^2} * I(x, y, \sigma) \quad (4-8)$$

式 4-8 中， $g(\eta)$ 为高斯核函数， η 为高斯方差，通过式 4-7 与 4-8 可以为每个像素计算出不同尺度下的 Hessian 矩阵值，计算行列式 $\det(H(x, y, \sigma))$ 来判别像素是否为极值点，这里用盒子滤波器（box filter）近似代替高斯二阶偏导数与图像 I 做卷积，

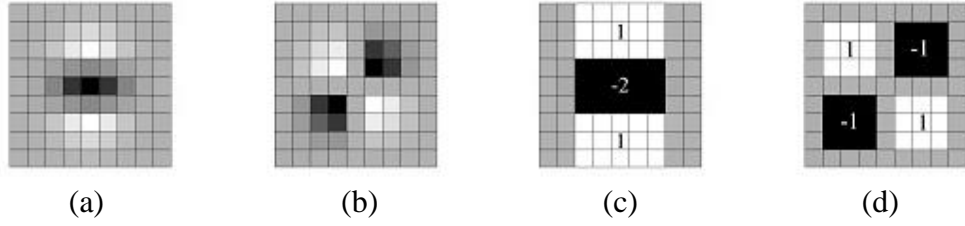


图 4-6 高斯模板

Fig.4-6 Gaussian template

如图 4-6 所示，用近似模板(c)(d)代替高斯模板(a)(b)，即用 D_{xx} , D_{yy} , D_{xy} 来近似代替 L_{xx} , L_{yy} , L_{xy} ，其行列式判别式近似为：

$$\det(H(x, y, \sigma)) = D_{xx}D_{yy} - (\lambda D_{xy})^2 \quad (4-9)$$

2) 构建尺度空间

DOG 图像金字塔中构建多层 octave, 每层 octave 都是由上一层 octave 图像高斯平滑后降采样得到，同一 octave 层中有图像尺寸相同、尺度不同多张图像。因此允许 octave 中多层图像同时被处理，提高算法的效率。

3) 非极大值抑制定位特征点

对图像金字塔中响应图像进行极值点搜索，检测过程中使用与该尺度层图像解析度相对应大小的滤波器进行检测，以 3×3 滤波器为例，检测特征点与自身尺度层及上下相邻尺度层的 26 个点比较，若特征检测点的值大于其周围像素则确定该点为局部区域特征点。

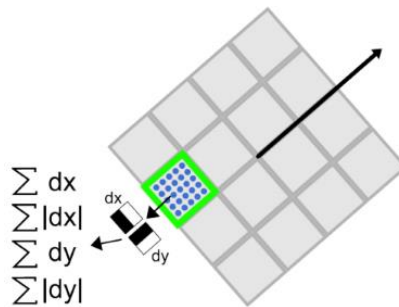


图 4-7 SURF 描述子

Fig.4-7 SURF descriptor

4) 特征点描述

如图 4-7 所示，为保证旋转不变性，采用统计特征点邻域内 Haar 小波特征的方法确定特征点主方向。将坐标轴旋转到特征点主方向，选取以特征点为中心边长为 20σ 的正方形区域，该区域主方向为特征点方向，将区域分为 16 个子区域，每个区域统计

25 个空间归一化采样点的水平方向和垂直方向的 Haar 小波特征，得到每个区域内四个特征系数，即水平方向值之和、垂直方向值之和、水平方向绝对值之和、垂直方向绝对值之和。每个特征点有 $16 \times 4 = 64$ 维向量，相比 SIFT 特征提取算法，SURF 算法描述子向量维度减少一半，在匹配过程中会极大加快匹配速度。SURF 算法匹配时计算两向量内积最大值作为匹配点。

如图 4-8 所示，采用算法对图像提取 SURF 特征点，这是将 Hessian 矩阵行列式响应值的阈值设为 500 时的提取效果。

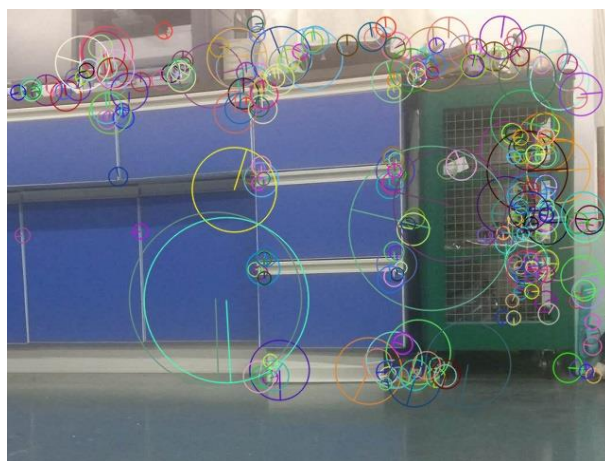


图 4-8 SURF 特征提取效果图

Fig.4-8 SURF feature extraction effect diagram

(3) ORB 特征

2011 年 Ethan 等人提出 ORB (Oriented FAST and Rotated BRIEF) 特征提取算法 (Ethan, 2011)，ORB 算法分为两部分，分别是改进的 FAST (Features from Accelerated Segment Test) 算法和改进的 BRIEF (Binary Robust Independent Elementary Features) 特征描述算法。

1) FAST 特征检测

FAST 算法定义基于特征点周围的图像灰度值，如果候选点周围领域内有足够多的像素点与该候选点的灰度差绝对值大于一个阈值，则认为该候选点为特征点。如图 4-9 所示，从图像中选取一点 P，以 P 为圆心画一个半径为 3 像素的圆，圆周上如果有 N 个像素点的灰度值与 P 点灰度值之差绝对值大于阈值，则认为 P 为特征点。

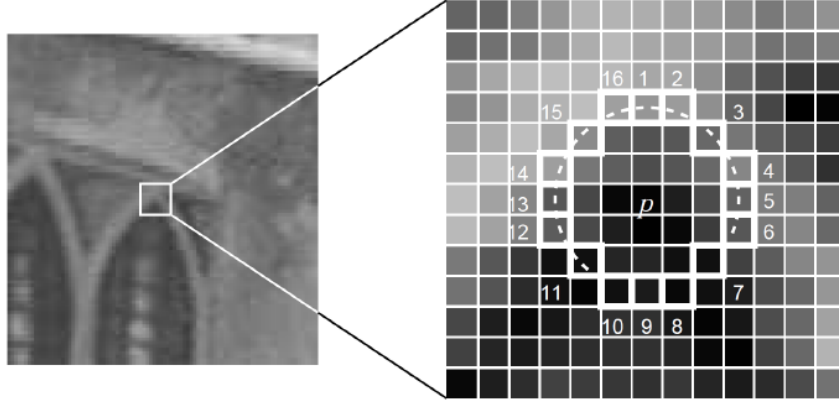


图 4-9 FAST 特征检测

Fig.4-9 FAST feature detection

对图像进行非极大值抑制，去除图像局部较密集特征点，计算 P 点邻域中所有特征点的 FAST 得分值（圆周上 16 点与 P 点灰度差值的绝对值总和），邻域中若有多个特征点，判断每个特征点的 FAST 得分值，若 P 点是邻域所有特征点中 FAST 最大值，则保留；否则，抑制。若邻域中只有一个特征点时，保留 P 点，FAST 得分值公式为：

$$N = \sum_{x \in \text{circle}(p)} |I(x) - I(p)| > \varepsilon_d \quad (4-10)$$

引入对 FAST 特征提取算法改进，假设在图像中提取 n 个特征点，通过降低判定 FAST 特征点的阈值，使 FAST 算法检测到特征点数大于 n 个，然后计算每个特征点的 Harris 响应值，按响应值大小进行排序，取前 n 个 Harris 响应值大的点作为 FAST 特征点。构造图像金字塔，原图像按比例因子缩放，多幅不同比例图像提取的特征点总和作为所有 FAST 特征点，解决尺度不变性问题。

2) BRIEF 特征描述

BRIEF 是一种二进制编码的描述子，为使 BRIEF 描述子具有旋转不变性，计算出特征点的主方向。这里引入灰度质心法，通过计算矩得到特征点半径为 r 的圆形邻域内的灰度质心位置，将特征点到质心位置的向量定义为该特征点的主方向。通过式 4-11 定义特征点邻域的矩为：

$$m_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad (4-11)$$

式 4-11 中 $i, j \in \{0, 1\}$ ， $I(x, y)$ 表示邻域内像素点在 (x, y) 处的灰度值，且 $x, y \in (-r, r)$ ， r 为邻域半径。通过特征点邻域矩的计算，得到该邻域质心为：

$$C = \left(\frac{m10}{m00}, \frac{m01}{m00} \right) \quad (4-12)$$

式 4-12 中 $m10 = \sum_x \sum_y xI(x, y)$, $m01 = \sum_x \sum_y yI(x, y)$, $m00 = \sum_x \sum_y I(x, y)$ 。由此构造特征点到邻域质心的向量 \vec{PC} ，特征点的主方向表示为：

$$\theta = \text{atan2}(m01, m10) \quad (4-13)$$

每个特征点通过灰度质心法得到的特征点主方向作为其邻域的 x 轴主方向，进行特征点描述，由此解决旋转不变性问题。

利用积分图像解决噪声敏感问题，以特征点为中心的 $S \times S$ （一般为 31×31 ）的窗口中，产生一对随机点后，以随机点为中心取 5×5 的子窗口，然后比较两个子窗口内像素灰度值之和的大小进行二进制编码，编码如式 4-15：

$$\tau(p; x, y) = \begin{cases} 1 & \text{if } \sum I(x) > \sum I(y) \\ 0 & \text{otherwise} \end{cases} \quad (4-14)$$

式 4-14 中 x, y 分别代表两个随机点， $\sum I(x)$ 、 $\sum I(y)$ 分别为两个区域内所有像素灰度值之和。一个 BRIEF 描述子由若干个 τ 测试组成的二进制码串，如式 4-15 所示，按照一定模式构造随机点对，可得到 BRIEF 特征描述符：

$$f_n(P) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(P; x_i, y_i) \quad (4-15)$$

式 4-15 中 n 表示构造随机点对的数目，即 BRIEF 描述符二进制码串的长度。对一个特征点进行描述，其 BRIEF 描述符长度为 n ，该描述符由特征点周围 n 组点（共 $2n$ 个点）对通过 n 次 τ 测试组成，将 $2n$ 个点组成 $2 \times 2n$ 矩阵 S ：

$$S = \begin{pmatrix} x_1, x_2, \dots, x_{2n} \\ y_1, y_2, \dots, y_{2n} \end{pmatrix} \quad (4-16)$$

改进的 BRIEF 描述符的特征点主方向 θ 与其对应的旋转矩阵 R_θ 构建校正矩阵 S_θ ，使 BRIEF 描述符具有旋转不变性，校正矩阵公式为：

$$S_\theta = R_\theta \times S \quad (4-17)$$

式中，

$$R_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (4-18)$$

校正后特征点的 BRIEF 描述符称为 steered-BRIEF，用公式表示为：

$$g_n(P, \theta) := f_n(P) | (x_i, y_i) \in S_\theta \quad (4-19)$$

对于图像中的每一个特征点，都得到一个 256 维的二进制编码，ORB 特征匹配利用汉明距离进行判别。图 4-10 是 ORB 特征点提取数量阈值设为 500 时的提取效果。



图 4-10 ORB 特征提取效果图

Fig.4-10 ORB feature extraction effect diagram

4.1.2 对比试验及改进方法

本文在学实楼 203 实验室的服务器（CPU i5-6500 @3.20GHz，运行内存 8GB）上对 SIFT、SURF、ORB 特征点提取算法的提取进行了试验对比。因 SLAM 系统的实时性需求，在保证提取精度的同时，主要在特征点提取速度方面进行衡量。

特征提取速度对比试验的步骤为：

- （1）准备一定数量的图像备用；
- （2）编写特征提取程序，在程序中设置不同的特征点提取个数；
- （3）在程序中添加计时语句，统计每张图像使用不同特征点提取算法和不同数量时的用时，最后求取平均值作为对比值，降低对比试验中的偶然性；
- （4）对图像中的特征点分布情况进行分析。

特征提取速度对比结果如表 4-1 及图 4-11 所示，表中数据是测量共 798 张图像后得出实验数据。通过数据对比，当提取相同数量特征点时，SURF 特征提取算法的速度比 SIFT 快一倍以上，ORB 特征提取速度是 SIFT 的 10 倍左右，当提取特征点数量设为 1000 个时，ORB 速度是 SIFT 的 17 倍。视觉 SLAM 方法中，在不使用 GPU 加速的情况下，经过试验对比发现只有 ORB 特征提取算法能够在 CPU 平台做到实时运

算，因此，本文选择 ORB 特征作为视觉 SLAM 系统的特征提取算法。

表 4-1 SIFT、SURF、ORB 特征提取速度

Tab.4-1 SIFT, SURF, ORB Feature Extraction Speed

提取特征点数量	300	500	1000
SIFT	0.740480	0.773738	0.880562
SURF	0.482301	0.410879	0.328760
ORB	0.061128	0.056618	0.064883

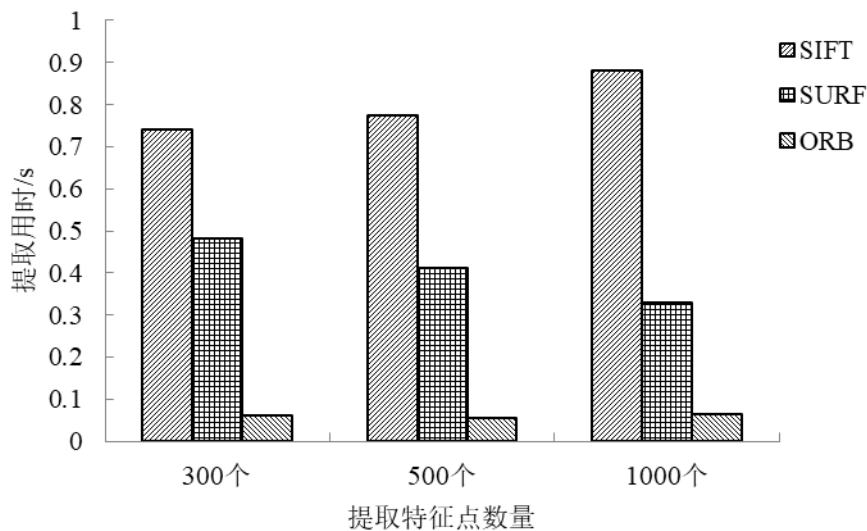


图 4-11 SIFT、SURF 和 ORB 特征提取速度

Fig.4-11 SIFT, SURF, ORB Feature Extraction Speed

通过观察多幅特征点提取效果图发现，三种特征提取算法在边缘及角点明显处能够很好的提取出特征点。图像上特征点分布的整体情况上可以看出，SIFT 和 SURF 特征点分布较均匀，ORB 特征点比较集中在物体边缘处，在图像中有很区域没有提取出特征点，即使将 ORB 特征点提取数量设置较大时，其特征点依然集中在纹理丰富区域，在其他纹理不丰富的区域基本提取不到特征点，这种分布不均在有噪声影响下或有遮挡情况发生时，会对特征匹配产生极大干扰，导致误匹配甚至匹配不到特征点。

本文为解决特征点分布不均问题，采用了图像区域分割的方法进行改善。将图像分为 $M \times N$ 个等分子区域，分别在子区域图像提取特征点，ORB 特征提取算法调整 FAST 阈值使得每个子区域都能够检测提取到最少数量的特征点，这样就保证了图像每个子区域都能够取到特征点。特征点比较集中的子区域提取 Harris 响应值较大的点，保证特征点分布的均匀性，在有噪声影响下或有遮挡情况发生时，减少对 SLAM 方法

中特征匹配的误匹配、匹配不到特征点等情况。从图 4-12 中看出，把图像通过区域分割（3×3）的方法其特征提取效果更加均匀。

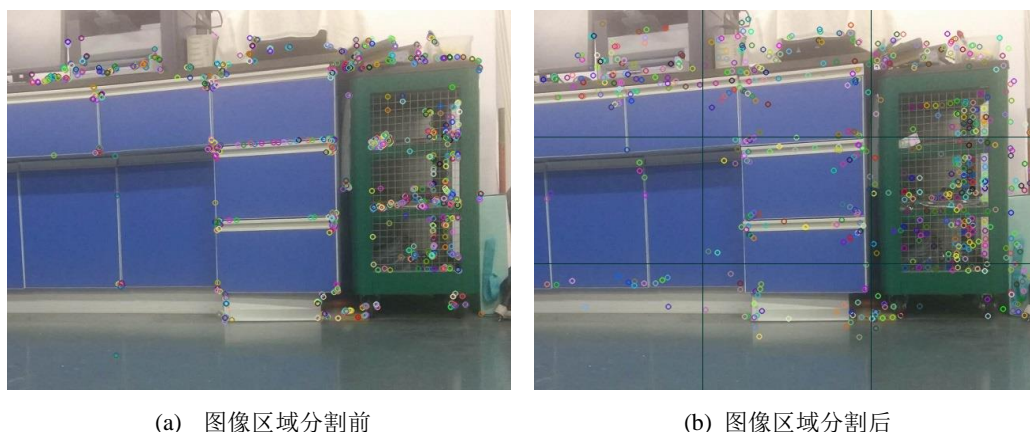


图 4-12 图像区域分割前后对比效果图

Fig.4-12 Image contrast before and after segmentation

4.2 改进的 FLANN 分类树特征匹配方法

完成对图像进行特征点提取和描述，进行图像之间特征匹配工作。ORB 特征的描述子是二进制码串，通过计算描述子的汉明距离来完成特征匹配。机器人运动过程中通过 RealSense 采集数据，然后特征匹配进行数据关联，特征点的是否能够正确匹配影响机器人位姿计算，所以特征匹配的正确率很大程度上能够决定 SLAM 系统的定位精度，当特征点错误匹配时，会给 SLAM 系统引入很大的噪声，严重降低定位精度和运算速度，同时特征匹配的速度影响了 SLAM 算法运行的实时性。



图 4-13 待匹配图像

Fig.4-13 Images to be matched

如图 4-13 所示，这是使用 RealSense 通过微小平移移动采集到的两幅图像，两幅图像有大面积共视区域，都包含纹理丰富区域和纹理稀少区域，特征匹配。本文对特征匹配的原理进行分析的同时，通过这两幅待匹配图像试验验证改进后的特征匹配算

法效果。本文设计实现改进后的 FLANN 分类树特征匹配方法，降低误匹配并加速匹配速度，为位姿计算提供正确的特征匹配关系，

4.2.1 算法设计

特征匹配时，将第一幅图像其中一个特征点 x ，与第二幅图像中的所有特征点进行对比，计算出特征点描述子距离，排序选出最短距离，这种匹配方式称为暴力匹配（Brute-Force Matcher）。

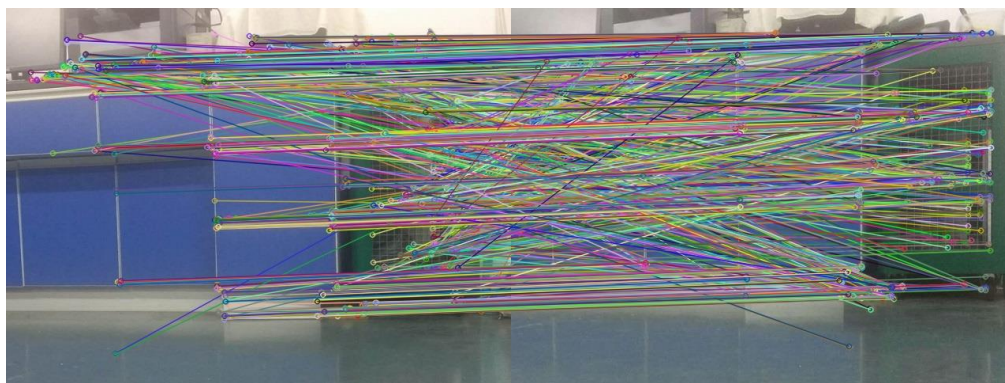


图 4-14 暴力匹配结果

Fig.4-14 Brute force match results

如图 4-14 所示，这是只使用暴力匹配方法进行匹配的效果图，图像中占多数的水平线表示正确匹配，斜线表示误匹配。在实际应用当中，暴力匹配会出现两个不可避免的问题：其一是在匹配过程中，第二幅图像中的一个特征点可能是第一幅图像中多个特征点的最优匹配，即因局部特征提取导致的一对多误匹配问题，其会导致图像之间的误匹配；第二个问题是暴力匹配本质上是穷尽匹配，时间复杂度为 $O(N^2)$ ，匹配速度慢，在特征点稀少的情况下可以取得不错的匹配效果，一旦在纹理复杂、特征点非常多情况时，其匹配速度是非常缓慢的。SLAM 系统是一个实时系统，对特征匹配的运行效率提出很高的要求，因此需要采用更高效特征匹配方法。

FLANN（Fast Library for Approximate Nearest Neighborsm，快速近似最近邻搜索）方法可以较好解决暴力匹配速度慢的问题。FLANN 是一种快速的高维空间近似最近邻搜索算法，根据数据维度的大小采用多重随机 K-Means 树的数据结构来对数据分类，当特征匹配时，通过分层遍历计算特征点与树节点的最小汉明距离，即可快速搜索匹配特征点。当数据量较大时，其匹配效率远远高于暴力匹配。但是，只采用 FLANN 方法来实现特征匹配方式只能够加快匹配速度，对暴力匹配中大量存在的误匹配没有

剔除，因此本文在 FLANN 特征匹配基础上进行了改进，引入了相互一致性检验方法和 RANSCA 方法逐步剔除误匹配。

首先，为解决一对多误匹配问题引入相互一致性检验的改进方法，方法的思想是只有当两幅图像中特征点之间都是彼此最优匹配时，才认为两个特征点是正确匹配。具体来说，第一幅图像中特征点 x 在第二幅图像中最优匹配为特征点 y ，只有当特征点 y 在第一幅图像中的最优匹配是特征点 x 时，才认为是特征点 x 与特征点 y 是正确匹配。满足相互一致性检验的匹配保留，不满足的匹配剔除掉。使用这一方法可以很大程度上减少误匹配，匹配效果图如图 4-15 所示。

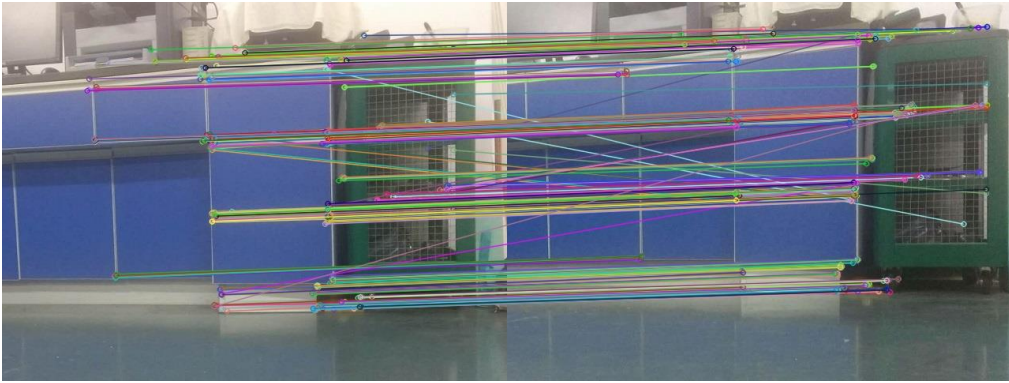


图 4-15 相互一致性检验后的匹配结果

Fig.4-15 Matching results after checking mutual consistency

使用在 FLANN 基础上增加相互一致性检验方法在加速匹配过程的同时，可以去除大部分的误匹配，但仍会有较少的难以剔除离散误匹配存在，会给位姿计算引入噪声。因此在上述改进方法基础上后，继续引入 RANSAC (Random Sample Consensus, 随机采样一致性) 方法来去除难以剔除的错误离散匹配。

首先，确定初始运动模型，随机从完成相互一致性检验的匹配对中抽取 4 对匹配点，求解 3×3 变换单应矩阵 H 。

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \quad \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} \quad (4-20)$$

单应矩阵 H 有 8 自由度，矩阵 H 为估计的运动模型 S ， $(x_m, y_m, 1)$ 和 $(x_n, y_n, 1)$ 是匹配点对的图像像素齐次坐标。计算运动模型 S 与所有匹配点的投影误差 θ ，计算公式为：

$$\theta = \left(x_n - \frac{h_{11}x_m + h_{12}y_m + h_{13}}{h_{31}x_m + h_{32}y_m + 1} \right)^2 + \left(y_n - \frac{h_{21}x_m + h_{22}y_m + h_{23}}{h_{31}x_m + h_{32}y_m + 1} \right)^2 \quad (4-21)$$

若某一匹配点对与运动模型 S 计算得出投影误差 θ 小于设定阈值 T ，则该匹配点对为“内点”，将初次所有“内点”作为最优内点集 Q ，然后从当前最优内点集随机抽取 4 对匹配点计算最优模型 S 和投影误差 θ ，若新的内点集的内点个数多于最优内点集 Q ，则新的内点集更新为最优内点集，不断迭代上述过程，直到迭代次数达到最大值。最终少量的离散点会因不符合通过大多数正确匹配点计算出来的运动模型而被剔除。

图 4-16 为继续使用 RANSAC 方法剔除离散点后的匹配效果图，改进后的 FLANN 分类树特征匹配算法已经能够去除绝大多数的误匹配。

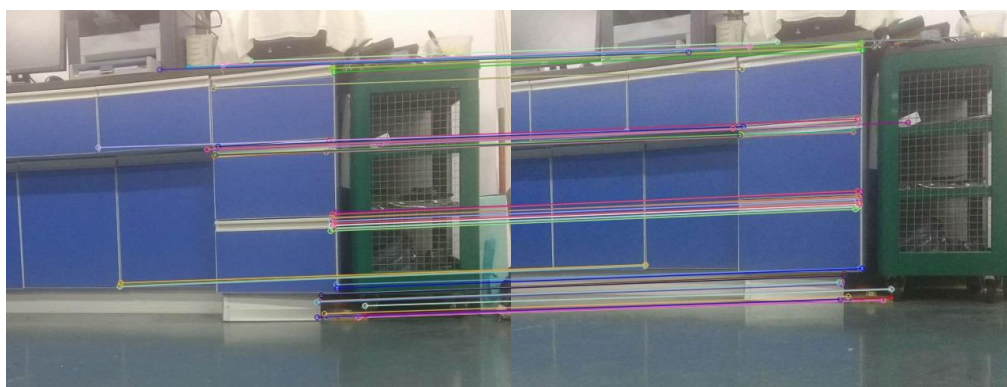


图 4-16 RANSAC 剔除误匹配后的匹配结果

Fig.4-16 RANSAC matching results after excluding mismatching

4.2.2 算法实现及试验验证

改进后的 FLANN 分类树特征匹配算法实现步骤为：

(1) 首先对已有图像所有特征通过 K-means 聚类分成 K 类，得到第一层特征树节点，然后再将第一层每个树节点中的特征点聚类分成 K 类，依次递推，最后得到叶子节点，每个叶子都是一个特征点描述子，建立图像特征分类树 Tree-Total；

(2) 对新采集的图像与已有关键帧图像进行特征匹配时，取当前图像的特征点 x ，计算与图像特征分类树中第一层 K 个聚类中心的汉明距离，判断最短距离选择当前聚类中心，并沿着当前聚类中心继续计算特征点 x 在下一层的最短距离的 K 节点，知道找到叶子节点；

(3) 完成当前图像中所有特征点的 FLANN 匹配后，得到对应的匹配点集 Q ，将当前图像中的特征点建立为图像特征分类树 Tree-Current，匹配点集 Q 在分类树 Tree-Current 上通过与步骤 (2) 相同方法完成 FLANN 匹配，得到匹配点集 P ；

(4) 匹配点集 Q 和 P 通过相互一致性判别剔除一对多的误匹配, 得到匹配点集 Q' 和 P' ;

(5) 在匹配点集 Q' 和 P' 中随机选取匹配点对, 完成 RANSAC 方法检验, 剔除错误离散的匹配。

综上所述, 改进后的 FLANN 分类树特征匹配方法的实现起来并不繁琐, 为了验证改进后特征匹配方法的优势, 本文设计了验证试验, 在算法原理设计部分展示了其剔除误匹配的效果, 下面将针对特征匹配速度方面的提升做验证试验结果如表 4-2 和图 4-17 所示。

表 4-2 特征匹配速度对比

Tab.4-2 Feature matching speed comparison

算法用时/s	图像特征点数量/个				
	157	339	508	872	1333
暴力匹配	0.037483	0.051636	0.065473	0.115884	0.159846
FLANN 方法	0.027615	0.028057	0.030864	0.0339417	0.037029
改进后算法	0.029473	0.031849	0.033916	0.0361263	0.041683

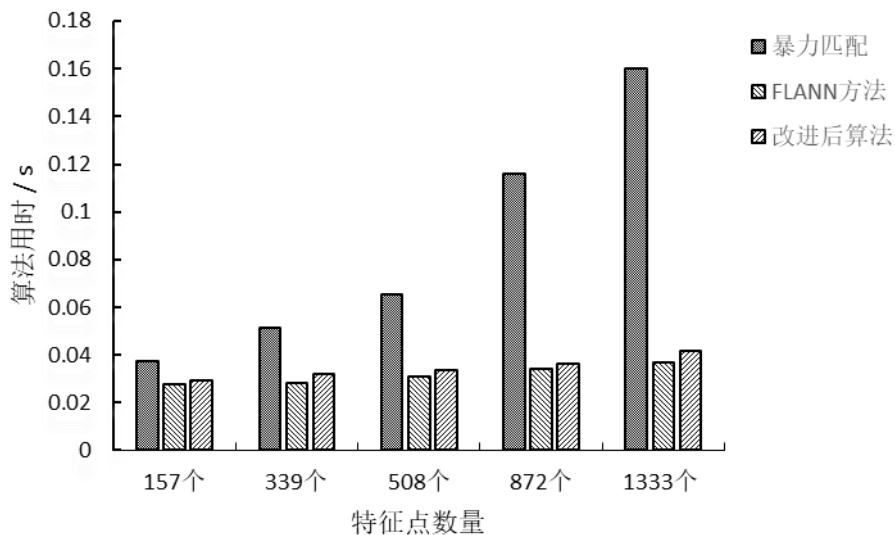


图 4-17 三种匹配算法速度对比

Fig. 4-17 Speed comparison of three matching algorithms

改进的 FLANN 分类树特征匹配算法时间复杂度为 $O(N\log N)$, 处理大量图像特征点匹配时, 匹配速度远远少于原始的暴力匹配方法, 与 FLANN 算法用时相近。在验证试验中采用不同数量特征点图像进行特征匹配试验, 通过在算法程序中加入计时语

句来统计用时。通过对比暴力匹配、FLANN、改进后 FLANN 分类树特征匹配算法的所用时间，可以在表 4-2 和图 4-17 中观察得出结论，随特征点数量的增加，暴力匹配方法所用时间剧增，而 FLANN 方法和改进后的 FLANN 分类树算法所用时间虽然也增加，但其增势缓慢，因为改进后的算法结合了 FLANN 和相互一致性检验，改进后的 FLANN 分类树算法用时比 FLANN 多一些；当特征点数量较少时，三种算法用时接近，特征点数量越多，它们的用时对比越明显，图像特征数量为 1333 个时，改进后 FLANN 分类树算法用时是暴力匹配算法是 1/5 左右，只比 FLANN 算法多花费 0.004654 s，显然，改进后 FLANN 分类树算法耗时与 FLANN 方法相当。经试验验证，本文改进后的 FLANN 分类树特征匹配方法能够在去除绝大多数误匹配的情况下，依然保持 FLANN 方法的匹配速度优势。

4.3 改进后的位姿计算方法

SLAM 系统中前端视觉里程计完成的任务为追踪机器人的在连续时间里的运动轨迹，追踪运动轨迹的计算过程分解为求解两个相机坐标系的刚性位姿变换。特征匹配之后利用匹配的点对来计算机器人在环境中的位姿。由于采用 RealSense 相机，在彩色图像特征匹配完成后，通过深度图像直接取得每个特征点的深度值，因此可以构建相机坐标系下的 3D 特征点坐标 (x, y, d) ，建立 3D-2D 特征点对的匹配关系。计算位姿时，常规的透视 N 点投影法（Perspective-n-Point, PnP）比较依赖正确的特征匹配，一旦出现误匹配，会对位姿计算产生很大影响；通过点云特征匹配和迭代最近点（Iterative Closest Point, ICP）求解位姿时，点云三维特征点提取算法的时间复杂度较高，迭代最近点算法依赖好的初始值，容易陷入局部最小值，收敛速度较慢。

鉴于常规方法中存在的问题，本文计算位姿的方法分为粗细结合的方法，首先通过 3D-2D 特征点投影求解 SVD 方式得到初始位姿，然后将初始位姿输入到 Point-to-Plane ICP 中，通过迭代最小化点到面的距离求解精准位姿。

首先，通过多对 3D-2D 匹配点求解初始位姿。如图 4-18 所示，对于任意的 3D 空间点 P ，其投影到两幅图像中的成像点 p_1 、 p_2 ，理想状态下从每个相机光心发射出来经过成像点 p_1 、 p_2 的光线都将交于空间点 P ，然而实际情况中因为噪声的存在，每条光线几乎不可能交于一点（Lepetit, 2009）。

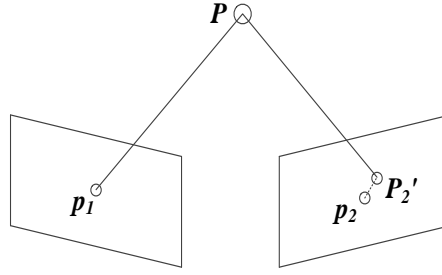


图 4-18 重投影误差

Fig.4-18 Reprojection error

设世界坐标系中 n 个 3D 空间点 P_i^w , 4 个非共面虚拟控制点 c_i^w ($i=1, 2, 3, 4$), 则存在关系:

$$P_i^w = \sum_{j=1}^4 a_{ij} c_j^w, \text{ 并且 } \sum_{j=1}^4 a_{ij} = 1 \quad (4-22)$$

在相机坐标系下点 P_i^c 与虚拟控制点的关系同样成立:

$$P_i^c = \sum_{j=1}^4 a_{ij} c_j^c, \text{ 并且 } \sum_{j=1}^4 a_{ij} = 1 \quad (4-23)$$

因相机内部参数矩阵 K 为固定值, 将其设为单位矩阵:

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \sum_{j=1}^4 a_{ij} \begin{bmatrix} X_j^c \\ Y_j^c \\ Z_j^c \end{bmatrix} \quad (4-24)$$

式中,

$$s = \sum_{j=1}^4 a_{ij} Z_j^c \quad (4-25)$$

将式 4-25 代入 4-24 可得:

$$\sum_{j=1}^4 (a_{ij} X_j^c - a_{ij} u_i Z_j^c) = 0, \sum_{j=1}^4 (a_{ij} Y_j^c - a_{ij} v_i Z_j^c) = 0 \quad (4-26)$$

将 4-26 表示为矩阵形式:

$$Mx = 0 \quad (4-27)$$

式 4-27 中 x 向量为 4 个相机坐标系下的虚拟控制点坐标, 一般通过求解 $M^T M$ 矩阵的方式求解出虚拟控制点坐标, 矩阵 M 展开为:

$$M = \begin{pmatrix} a_{11} & 0 & -a_{11}u_1 & a_{12} & 0 & -a_{12}u_1 & a_{13} & 0 & -a_{13}u_1 & a_{14} & 0 & -a_{14}u_1 \\ 0 & a_{11} & -a_{11}v_1 & 0 & a_{12} & -a_{12}v_1 & 0 & a_{13} & -a_{13}v_1 & 0 & a_{14} & -a_{14}v_1 \\ M & M & M & M & M & M & M & M & M & M & M & M \\ a_{n1} & 0 & -a_{n1}u_n & 0 & a_{n2} & -a_{n2}u_n & a_{n3} & 0 & -a_{n3}u_n & a_{n4} & 0 & -a_{n4}v_n \\ 0 & a_{n1} & -a_{n1}v_n & a_{n2} & 0 & -a_{n2}v_n & 0 & a_{n3} & -a_{n3}v_n & 0 & a_{n4} & -a_{n4}v_n \end{pmatrix} \quad (4-28)$$

M 矩阵是 $2n \times 12$ 矩阵, 解出 x 后通过式 4-24 求解出相机坐标系下空间点 P_i^c 的 3D 坐标。通过相机内部参数 K 将 2D 的图像像素坐标转换成 3D 空间点坐标, 然后计算出世界坐标系下 3D 空间点到相机坐标系下 3D 空间点的变换矩阵, 即求出相机的初始变换位姿 T_0 。

初始位姿 T_0 输入到 Point-to-Plane ICP 方法作为初始值来迭代。若初始位姿 T_0 接近真实位姿, 迭代过程中可以大幅度减少 ICP 的迭代次数, 加速位姿估计; 若初始位姿 T_0 误差较大时, Point-to-Plane 也能够依靠最小化点到面的距离来求解位姿。Point-to-Plane ICP 方法与普通 ICP 方法不同之处在于误差度量方式由点到面代替了点点到点, 其方法更稳定并且收敛速度更快。

Point-to-Plane ICP 求解点云的刚性变换, 根据已完成配对的两帧图像及深度图像, 构建存在共视关系的点云集 P_{raw} 和 P'_{raw} , 原始点云集非常稠密, 采用法线空间降采样方法 (阈值设为 0.1) 得到十分之一数量的点云子集 P 和 P' , 目的是加速后续算法运算速度。

去除 P 和 P' 中的无效数据点, 计算点云子集 P 和 P' 中每个点对应的法向量。对于点云中的一点 p , 取一定半径邻域的点进行平面拟合 (由于拟合曲面计算度太大, 这里半径区域很小所以用平面代替), 在拟合平面基础上求对应点的法向量。由此得到 P 和 P' 中每一个点的法向量。

Point-to-Plane ICP 方法的迭代步骤为:

(1) 将 P 和 P' 的点和对应法向量分别作为源数据和目标数据, 将通过求解 SVD 方式得到的位姿作为初始值输入, 通过 Point-to-Plane ICP 方法求解精确位姿, P 和 P' 的点表示为:

$$P = \{p_1, p_2, p_3, K, p_n\}, P' = \{p_1', p_2', p_3', K, p_m'\} \quad (4-29)$$

(2) 通过初始位姿 T_0 将点集 P 旋转平移到点集 P' 所在位置;

(3) 对 P 中每一点 p_i 基于法向量和最小欧氏距离找到对应的点对 q_j , 通过对应

点对求解出新的变换矩阵 T , S_j 是 P' 在 q_j 处的切平面, 设置误差度量函数为:

$$e = \sum_{i=1}^N \|Tp_i - q\|^2 \quad \text{with } q = \underset{q \in S_j}{\arg \min} \|Tp_i - q\| \quad (4-30)$$

(4) 不断将 P 通过 (3) 求解出变换矩阵 T 变换到 P' 所在位置, 并计算误差度量函数, 直到达到最大迭代次数或误差度量函数小于一定阈值后停止迭代, 最后得到最优的变换位姿 T 。

求解步骤中为使误差度量函数不断减小, 引入列文伯格-马夸尔特方法的非线性优化方法, 使得梯度函数不断下降, 变换位姿逼近最优解。

4.4 基于词袋模型的闭环检测

闭环检测 (Loop Closure Detection), 也称为回环检测, 是机器人检测当前观测到的场景是否在曾经出现过, 正确的闭环检测可以极其显著的消除累计误差, 纠正运动轨迹和地图。若机器人的运动中没有检测出闭环, 其累计误差会对后续帧的位姿估计带来很大影响, 其估计的轨迹和构建的地图都很不稳定。当机器人运动中产生一些局部闭环或是大闭环时, 形成闭环的几个关键帧之间形成了约束, 能够将因误差累计造成的运动轨迹偏移修正到正确的位置; 同时, 当顺序数据关联部分追踪丢失后, 闭环检测模块能够将当前观测的场景与已观测场景建立数据关联, 实现重定位的功能。

4.4.1 算法原理分析

如图 4-19 所示, 图像表示机器人运动轨迹的拓扑图, 最左侧图中没有闭环检测环节, 位姿估计只有单个帧之间的约束, 其结构是非常单一的, 当其中一个帧间估计出现较大误差, 其后面的整个轨迹都会受到影响, 长时间的累计误差会导致轨迹发生很大的偏离; 从中间图中可以看出, 当 SLAM 方法中加入闭环检测环节时, 第 7 帧除了与第 6 帧匹配完成顺序数据关联, 还检测出与第 1 帧、第 2 帧图像发生环状数据关联, 形成闭环约束, 因此能够通过检测闭环来纠正累计偏差优化整个运动轨迹和地图。

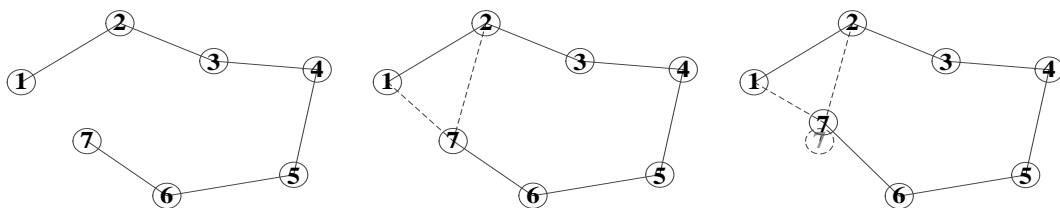


图 4-19 闭环约束

Fig.4-19 Closed-loop constraints

闭环检测的实质是对观测数据相似性的检测，在视觉 SLAM 系统中，有几种做闭环检测的方法：最原始的方法是做暴力检测，检测当前图像和所有的历史数据之间是否能够正确匹配，但是当历史数据数量太大时其检测效率非常低；在历史数据中随机抽取 K 帧与当前数据做相似性比较，这种方法可以在很小的场景中运用，但是当历史数据的数量增多时，其能够检测到闭环的几率也随之减少；基于词袋模型（Bag-of-Words, BoW）的方法，历史数据（图像）中的视觉特征（这里采用 ORB 特征）通过 K-means 聚类的方法建立向量化的字典，当观测到新的数据，对新数据同样提取特征点进行向量化，在词典中逐层查找，通过比较向量之间相似性进行判断。

4.4.2 算法设计实现

本文采用基于词袋模型的闭环检测方法。算法流程设计如图 4-20 所示，闭环检测之前需要提前建立特征词典，当检测新图像帧时，从词典中查找相似度大于阈值的关键帧，并通过闭环验证确定闭环，确定闭环帧之后，通过闭环帧与当前帧之间的位姿关系，以闭环帧位姿为基准纠正当前帧及与当前帧有共视关系的关键帧位姿；或是通过闭环检测来重定位，重定位成功后通过闭环帧与当前帧位姿关系，得到当前相机位姿。

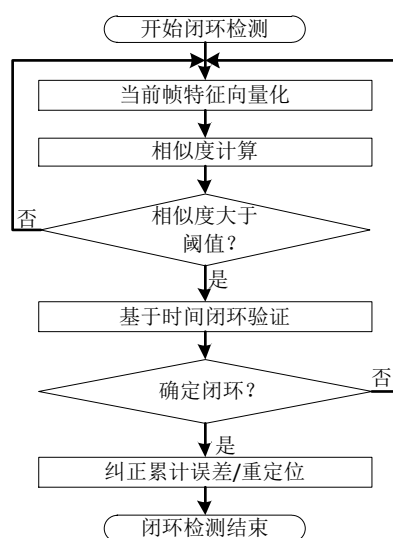


图 4-20 闭环检测环节的算法设计流程

Fig.4-20 Closed-loop detection algorithm design process

将已观测图像数据通过 K-means 的方法进行聚类，并且将构建的词典以 K 叉树的方式进行分层处理，方便快速查找“单词”。K-means 聚类的步骤为：

- (1) 将历史数据提取 N 个特征点，随机选取 K 个中心点；

(2) 对 N 个特征点分别计算其与 K 个中心点的距离，离得最近的归为一类；

(3) 重新计算 K 个类的中心点，即类的质心，如果中心点的变化很小即表示算法收敛，退出；否则返回第二步迭代。

通过 K-means 聚类方法构建 K 叉树词典结构：

(1) 对 N 个特征点通过上述 K-means 方法聚类成类，得到 K 叉树的第一层；

(2) 对第一层的每个节点通过 K-means 方法聚类成 K 类，然后递推每层节点进行 K-means 聚类；

(3) 最后得到 K 叉树的叶子层，在每个叶子层构建单词。

将所有历史关键帧图像中的特征建立为词典，本文首先在线下通过采集的图像进行训练，当运行 SLAM 程序时，首先载入训练好的词典，然后在 SLAM 运行时，不断将关键帧图像进行特征向量化，不断的加入到词典中。当对新的图像帧最闭环检测时，不断在词典中查询与新图像帧的相似度，满足闭环验证条件表示检测到闭环。

计算当前帧和临近 N 帧关键帧的相似度分值，取其最小值作为先验相似度，并且设定一个阈值与先验相似度相乘得到一个评分标准 $score$ ，闭环检测时，检测到相似度评分大于 $score$ 的候选闭环帧。基于词袋模型的图像特征向量化只依赖于特征的数量，对特征的位置和顺序都没有约束，因此相似场景可能会出现错误的闭环，因此需要一定的验证机制。当前图像帧和临近关键帧之间因为相隔很近，所以它们之间相似度很高，若只考虑相似度评分来做闭环检测很大可能是检测出临近关键帧，造成错误的闭环，因此需要在计算出相似度后做相应验证处理。本文采用基于时间的闭环验证方法，在时间上，不与临近的前 K 帧（本文设定为 $K = 6$ ）关键帧做闭环检测，并且在完成一次闭环检测后的连续 L 帧图像不做闭环检测，保证闭环检测运行的间歇，节省系统开销。当检测到单次的闭环检测不认为是正确的闭环，一段时间中持续检测到时才认为构成正确的闭环。闭环后通过世界坐标系中地图点重投影到当前帧的方式计算闭环帧和当前帧的位姿关系，以闭环帧位姿为基准纠正当前帧的位姿，并纠正和当前帧有共视关系的关键帧位姿。

4.5 本章小结

分析了 SIFT、SURF、ORB 特征提取算法原理，通过特征提取试验对比三种算法的性能，经试验验证得知 ORB 特征提取效率最高，能够满足 SLAM 方法的实时性要

求，SIFT 和 SURF 特征分布较均匀，ORB 特征分布较集中，为了改进 ORB 特征分布不均匀问题，采用了图像区域分割的方法加以改进，改进后效果良好；采用改进的 FLANN 分类树特征匹配方法，通过试验验证改进后的算法在匹配速度和剔除误匹配两方面性能优异；采用改进的位姿计算方法，通过 3D-2D 特征点投影求解 SVD 方式得到初始位姿，将初始位姿输入到 Point-to-Plane ICP 中，通过迭代最小化点到面的距离求解精准位姿；对词袋模型算法原理进行分析，设计实现了闭环检测环节，最后采用了基于时间的闭环验证方法做闭环验证。

5 基于图优化的 SLAM 方法

SLAM 的数据关联实现了机器人进行定位追踪，但仅通过数据关联并不能得到精确的定位结果，定位精度会随着时间推移越来越偏离真实运动轨迹。为保证长时间运动轨迹和路标的全局一致性和稳定性，需要对全局机器人位姿和路标点形成的约束进行优化。

从 SLAM 的发展历程中，有基于滤波的 SLAM 方法（比如 EKF-SLAM 以及 PF-SLAM），这种方法采用概率的方法来表示观测信息的不确定性，其认为观测数据中的路标信息是各自独立的，长时间运行后路标的数量太多会导致计算机的运行资源不足，并且这种方法是将非线性问题近似为线性来处理，本身对 SLAM 问题的建模就导致误差的产生，因此基于滤波的视觉 SLAM 方法不适用于大规模场景。本文采用了基于图优化的方式对 SLAM 方法进行建模优化，在图优化中基于前端顺序数据关联，以及闭环检测中的环形数据关联，都是根据观测数据来建立关于图的“节点”之间的约束（王忠立，2015），图优化理论是基于观测数据不确定性的最大似然估计，常用方式是通过构建非线性二乘法问题求解。

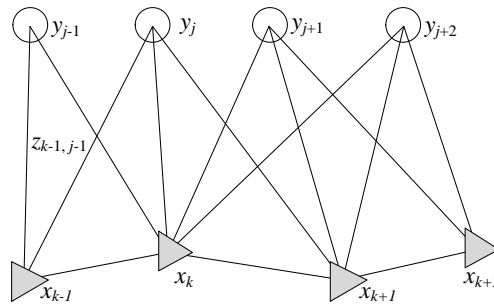


图 5-1 图优化结构

Fig.5-1 Graph optimization structure

5.1 构建图优化代价函数

在图 5-1 中，圆形表示路标点，三角形表示机器人位姿，三角形与圆形之间连线表示观测，三角间连线表示相机运动轨迹。假设 $t = 0$ 到 $t = N$ 的时间内，位姿从 x_0 到 x_N ，路标有 y_1, y_2, \dots, y_M 。那么 SLAM 系统中的运动方程和观测方程表示为：

$$\begin{cases} x_k = f(x_{k-1}, u_k) + w_k, & k = 0, 1, \dots, N, \\ z_{k,j} = h(y_j, x_k) + v_{k,j}, & j = 1, 2, \dots, M \end{cases} \quad (5-1)$$

其中, $f(x_{k-1}, u_k)$ 表示 x_k 时刻位姿与 x_{k-1} 时刻位姿的关系, u_k 表示运动数据, w_k 表示运动误差, $h(y_j, x_k)$ 表示 k 时刻相机位姿观测到路标 y_j 建立的观测数据, $v_{k,j}$ 表示观测误差。世界坐标系中的 3D 路标点 $P_j = [X, Y, Z]$ 到相机图像像素点 $z_j = [u, v]$ 的观测关系可以表示为 (这里忽略齐次坐标和非齐次坐标的差异) :

$$z_{k,j} = [u, v]^T = K(R_k P_j + t_k) + v_{k,j} \quad (5-2)$$

在没有其他运动传感器的视觉 SLAM 系统中, 图优化的优化目标主要为观测方程中的误差, 将相机的位姿 R, t 表示为李代数的形式 ξ , 联立式 5-1 和式 5-2, 因此 k 时刻相机位姿与路标点的观测误差表示为:

$$e = z - h(\xi_k, P_j) \quad (5-3)$$

将观测误差函数构建成最小二乘问题求解, 整个 SLAM 系统中的相机位姿和路标点作为优化对象, 其代价函数表示为:

$$\frac{1}{2} \sum_{k=0}^M \sum_{j=1}^N \|e_{k,j}\|^2 = \frac{1}{2} \sum_{k=0}^M \sum_{j=1}^N \|z_{k,j} - h(\xi_k, P_j)\|^2 \quad (5-4)$$

求解最小代价函数即对 SLAM 系统中的所有相机位姿和路标点的位置做出调整, 其本质是求解最小二乘问题。

5.2 求解最小二乘问题

求解最小二乘问题一般通过迭代的方式, 从一个初始值出发, 通过不断寻找下降方向 ∇x 来找到目标函数的最优解, 即不断调整增量更新当前的优化变量, 使代价函数不断下降。因此求解最小二乘问题就是一个求解最小增量的问题, 当增量极小, 代价函数无法下降时算法收敛, 此时取得最优解。求解最小二乘问题步骤步骤为可以概括为:

- (1) 给定一个初始值, 这里是初始相机位姿和路标点位置;
- (2) 对于第 n 次迭代, 寻找增量 Δx , 使得 $\|f(x_k + \Delta x_k)\|^2$ 达到最小;
- (3) 若增量 Δx 足够小, 则停止迭代, 否则令 $x_{k+1} = x_k + \Delta x$, 重复第二步。

本文采用列文伯格-马夸尔特方法方法求解最小二乘问题。1977 年由 More 首次提出列文伯格-马夸尔特方法 (Levenberg-Marquardt), 也称为阻尼牛顿法 (More, 1977)。其在高斯牛顿法的基础上给 Δx 增加了信赖区域, 解决了高斯牛顿法中因增量 Δx 较大,

函数本身近似不准确的问题。信赖区域即通过一个评价标准来不断调整增量 Δx 的大小，通过求解实际下降的值和模型预估下降值的比值来对增量 Δx 做调整：

$$\rho = \frac{f(x+\Delta x) - f(x)}{J(x)\Delta x} \quad (5-5)$$

式 5-5 中 ρ 值越接近 1，说明近似结果越好，如果 ρ 太小，近似减小的值远远多于实际较小的值，近似结果差， ρ 的值比较大，则实际下降的比模型预估下降值大，此时应放大近似范围。因此列文伯格-马夸尔特方法计算步骤为：

- (1) 给定一个初始值 x_0 ，以及初始半径 γ ；
- (2) 对于第 k 次迭代，计算， $\min_{x_k} \frac{1}{2} \|f(x_k) + J(x_k)\Delta x_k\|^2$ ， $\|D\Delta x_k\|^2 \leq \gamma$ ；
- (3) 计算 ρ 的值，若 $\rho > 0.75$ 则 $\gamma = 2\gamma$ ，若 $\rho < 0.25$ 则 $\gamma = 0.5\gamma$ ；
- (4) 如果 ρ 的值大于某个阈值，则认为近似效果好，令 $x_{k+1} = x_k + \Delta x$ ；
- (5) 判断算法收敛则退出，不收敛返回第 2 步。

其中， D 为单位矩阵，是系统引入的约束，列文伯格-马夸尔特增量方程为：

$$H\Delta x = g, \quad (J(x)^T J(x) + \lambda D^T D)\Delta x = g \quad (5-6)$$

一般 D 为单位矩阵，因此其简化形式为：

$$(J(x)^T J(x) + \lambda I)\Delta x = g \quad (5-7)$$

当 λ 值较小时， $H = J(x)^T J(x)$ 起的作用较大，其模型接近于高斯牛顿， λ 值较大时， λI 起的作用比较大，说明 x 附近二阶近似不够好。这种求解方式可在一定程度上体面 H 矩阵的非奇异和病态问题，提供更加鲁棒的增量 Δx 。

5.3 稀疏性加速 SLAM 运算

全局优化是将所有的相机位姿和路标点放在一起优化，对其优化求解的增量方程的维度是非常大的，若直接对矩阵 H 进行求逆来计算增量方程，其复杂度为 $O(N^3)$ ，非常耗费计算机运行资源，在 SLAM 方法中无法达到实时求解，利用矩阵 H 的稀疏性，可以加速增量方程的计算速度。优化变量中把相机位姿放在一起考虑，路标点放在一起，分别有：

$$x_c = [\xi_1, \xi_2, K, \xi_m], \quad x_p = [p_1, p_2, K, p_n] \quad (5-8)$$

把优化问题的方程有如下表示：

$$\frac{1}{2} \|f(x + \Delta x)\|^2 = \frac{1}{2} \|f(x) + F\Delta x + E\Delta x\|^2 \quad (5-9)$$

这里的 F 和 E 分别是关于相机位姿和路标点的雅克比矩阵，其代表了整体目标函数对整体优化变量的导数，其维数是非常大的，其内部都是由每个相机位姿或空间点的误差项的导数 E_i 和 F_i 构成，使用列文伯格-马夸尔特方法求解优化问题，其增量方程表示为：

$$H\Delta x = g, \quad H = J^T J \quad (5-10)$$

这里的 $J(x)$ 是由 F 和 E 构成，其形式表示为 $J=[F \ E]$ ，因此 H 表示为：

$$H = J^T J + \lambda I = \begin{bmatrix} F^T F & F^T E \\ E^T F & E^T E \end{bmatrix} + \lambda I \quad (5-11)$$

在整体的代价函数中，其中一个误差项 e_{ij} 描述了位姿 ξ_i 看到路标点 P_j ，因此每个误差项都是只描述了第 i 个相机位姿和第 j 个路标点，因此矩阵 H 中每一部分，都是对应相机位姿和此时位姿观察到路标点误差项对优化变量的导数，当响应位姿观察不到其他路标点，其对应的矩阵 H 中的导数部分为零，这就构成了 H 矩阵的稀疏性，其表示为：

$$J_{ij}(x) = \begin{pmatrix} 0_{2 \times 6}, L, 0_{2 \times 6}, \frac{\partial e_{ij}}{\partial \xi_i}, 0_{2 \times 6}, L, 0_{2 \times 3}, \frac{\partial e_{ij}}{\partial P_j}, 0_{2 \times 3}, L, 0_{2 \times 3} \end{pmatrix} \quad (5-12)$$

式 5-12 中除了误差项对相机位姿的偏导数 $\frac{\partial e_{ij}}{\partial \xi_i}$ 和误差项对路标点的偏导数 $\frac{\partial e_{ij}}{\partial P_j}$ 为非零

外，其他部分都为零。对于整体矩阵 H 来说，其形式 $J^T J$ 也是稀疏的形式，

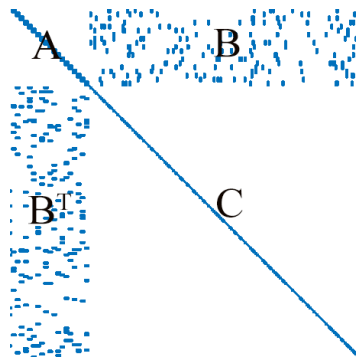
$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta x_p \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \quad (5-13)$$


图 5-2 Hessian 矩阵区域划分

Fig.5-2 Hessian matrix area division

如图 5-2 所示，其可以分为 4 个区域矩阵块。其中矩阵块 A 为对角块矩阵，只与相机位姿相关，每个对角块的维度与相机位姿的维度相同，对角块个数与相机位姿个数相同；每个块矩阵块 B 与 B^T 包含了相机位姿能够观察到的路标点关系；矩阵块 C 是只与路标点相关的对角块，其中每个对角块为 3×3 的矩阵，因某一个相机位姿能够观察到几百个路标点，因此矩阵块 C 中对角块数量远远多于矩阵块 A 。

对角块矩阵求逆的比一般矩阵求逆容易许多，因为只需对对角线矩阵块分别求逆即可。借助对角矩阵特性，这里对上述方程进行高斯消元，这个过程一般称为边缘化求解，

$$\begin{bmatrix} I & -BC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} = \begin{bmatrix} I & -BC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \quad (5-14)$$

可以得到:

$$\begin{bmatrix} A - BC^{-1}B^T & 0 \\ B^T & C \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta x_c \end{bmatrix} = \begin{bmatrix} a - BC^{-1}b \\ b \end{bmatrix} \quad (5-15)$$

因此可以得到只与 Δx_c 相关的增量方程:

$$\begin{bmatrix} A - BC^{-1}B^T \end{bmatrix} \Delta x_c = a - BC^{-1}b \quad (5-16)$$

此线性方程维度与矩阵块 A 相同，且 C 为对角矩阵，所以 C^{-1} 非常容易解出，因此可以快速求出 Δx_c ，然后将 Δx_c 带入式 5-15 求解出 Δx_p 。

以上就是通过 *Hessian* 矩阵稀疏性来加速求解增量方程的方法，其可以使 SLAM 系统在运行较大规模相机位姿和路标点时，仍能进行实时运算，*Hessian* 矩阵的稀疏性是图优化框架 SLAM 方法能够实时运行的关键。

5.4 构建占据网格地图

SLAM 的两大任务分别是实现机器人定位功能和地图构建功能。前面章节介绍了 SLAM 系统中数据关联及全局优化问题，实现机器人在环境中的精确追踪定位，并优化了稀疏形式的路标点，这里的路标点就是地图形式的一种，其为图像特征点在世界坐标系中的 3D 位置。SLAM 系统作为一种底层技术，其为不同的上层应用提供不同意义的地图，地图一般用于定位、导航、避障、重建以及交互等功能，其中稀疏点的

路标地图只能用于实现定位功能，若想要实现其他更复杂功能，需要对地图形式做出改善。

为实现机器人地图构建功能，本文首先构建稠密点云地图（Point Cloud Map），然后将点云地图转换为占据网格地图（Occupancy Map）。构建点云地图通过关键帧图像及其深度图像来实现，通过构造每帧图像的点云地图，然后进行点云拼接，拼接完成后进行降采样和体素滤波。不过点云地图的形式是由离散点构成，其为比较初级的地图，不能够直接用来做导航和避障，并且点云地图的规模通常非常大，其中蕴含的许多信息是导航和避障等功能运用不到的，比如阴影、褶皱等细节，因此需将其转化为更加灵活、规模小、能够随时更新的地图形式。本文采用八叉树地图（Octomap），这是一种占据网格地图，它是在 2013 年由 Armin Hornung 等人提出的（Armin, 2013），其能够在点云地图基础上进行转换，且能节省大量的存储空间。

八叉树地图是一种通过树结构存储地图占据信息的地图。其思想是将点云地图通过八叉树的结构进行分割划分，如图 5-3 所示，首先构建包含所有点云地图的立方体根节点，然后开始八叉树的递归分割，每一次分割都会使点云数据细分到 8 个子立方体中，也就是分为 8 个子节点，这样不断的递归分割，直到达到所需的分辨率为止。八叉树地图中并不是每个节点都需要分割成更精细的子节点，当某个节点的所有子节点都被占据或者是不被占据时，是没必要分割展开这个节点的。

八叉树地图能够根据导航避障需求调节精细程度，如图 5-4 所示，这是 4 种分辨率级别下的八叉树地图。八叉树地图中每个节点表示空间占据信息，但其表示信息并不是简单的 0 或 1，因为噪音的影响，某一节点代表的占据信息可能上一时刻为占据，下一时刻没有被占据，因此八叉树地图使用概率的形式来表达某一节点是否被占据。

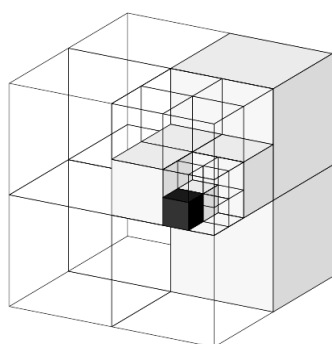


图 5-3 八叉树地图示意图

Fig.5-3 Diagram of octomap

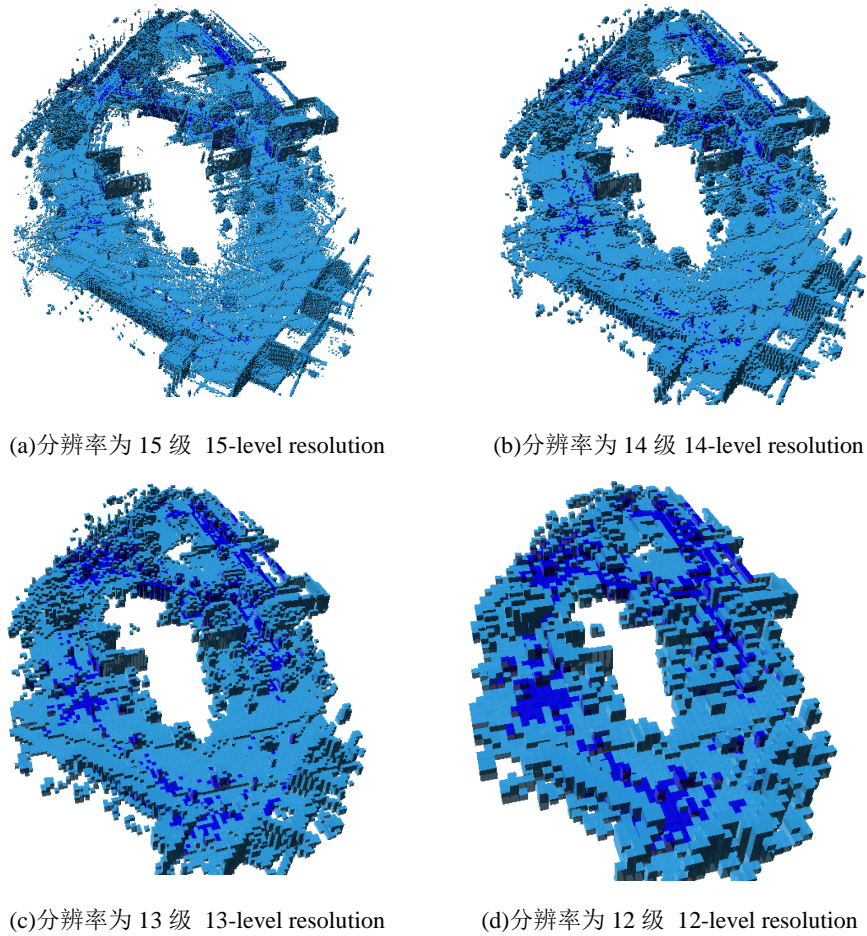


图 5-4 不同分辨率的八叉树地图

Fig.5-4 Octomap with different resolutions

通过不断的观测某一节点是否被占据，来得到该节点的占据信息，通过此方式可以动态的对空间中的障碍物进行建模，一般占据信息使用概率对数值（Log-odds）的形式来描述：

$$p_i = \log it(x) = \log\left(\frac{x}{1-x}\right) \quad (5-17)$$

式 5-17 中 $p_i \in R$ ， x 为 0~1 的概率，节当 p_i 从 $-\infty$ 变到 $+\infty$ ，概率 x 随之从 0 变到 1，节点占据信息初始值为 0.5，其概率对数值 0，当观测到该节点被占据时，概率对数值增加一个值，没有观测到被占据时减少一个值。当查询该节点是否被占据 ($x > 0.5$) 时，将式 5-17 逆转换得到占据概率信息。更新某一个节点的占据概率，设某节点为 n ，观测数据为 z ，从开始到 t 时刻该节点的概率对数值为 $L(n | Z_{1:t})$ ，则 $t+1$ 时刻的概率对数值表示为：

$$L(n | z_{1:t+1}) = L(n | z_{1:t}) + L(n | z_t) \quad (5-18)$$

利用上述更新八叉树地图中节点占据概率的方法，通过 RealSense R200 相机不断观测每个像素的深度，可以得知光心到每个像素对应空间点的线段上没有物体，利用这个特性可以不断对观测到的节点进行更新，当节点占据概率达到上限阈值或者下限阈值时，认为这个节点是稳定的被占据或没被占据。当一个节点中的 8 个子节点同时稳定的被占据或没被占据时，这个节点不必展开。八叉树地图中有大量连续物体或空白区域，因此有大量的节点不必都分割到最深的子节点，这种特性压缩了大量存储空间。

本文 SLAM 方法进行地图构建的步骤为：

- (1) 首先构建单幅关键帧图像的稠密点云，对点云进行体素滤波降采样，减少点云数量加速点云拼接速度，同时体素滤波能够保持点云的形状；
- (2) 根据关键帧图像在世界坐标系下的位置，将多组点云变换到世界坐标系下进行拼接；
- (3) 使用统计滤波器对拼接后点云进行过滤，去除离散点；
- (4) 将滤波后的点云转换为八叉树地图。

5.5 本章小结

根据 SLAM 数据关联得到的机器人位姿和路标点约束，构建基于图优化的代价函数，通过迭代计算最小二乘误差函数，对全局优化问题进行求解，对其中运用到的 *Hessian* 矩阵稀疏性原理进行了分析，其对增量方程的加速求解是 SLAM 系统中全局优化能够实时运行的关键。分析不同地图形式的作用及八叉树地图的构建原理，在全局优化后的机器人位姿基础上拼接点云地图，并利用体素滤波器对点云降采样加速拼接过程，使用统计滤波器过滤掉离散点，最后转换为八叉树地图形式，八叉树地图具有节省存储空间、带有占据信息等优点。

6 系统设计与试验验证

前面章节对 RealSense 进行了标定试验，对特征提取算法和特征匹配算法进行了对比试验验证。本章分别通过 TUM 数据集和实时采集图像运行两种方式验证本文设计的 SLAM 方法实用性，并绘制出机器人运行轨迹并构建点云地图和占据网格地图。通过对比数据集中的真实轨迹来计算本文 SLAM 方法的平均误差，评估本文方法的定位精度，并将地图与真实场景进行对比，评估其对环境建模的细节以及空间占据情况。通过实时采集运行试验来验证本文 SLAM 方法的实用性，评估其是否能够实时估计出相机位姿，以及能否实时构建地图。本章首先本文所设计 SLAM 方法的总流程，然后介绍了试验所用的硬件设备及数据传输流程，最后进行机器人定位于地图构建试验，并对实验结果进行分析讨论。

6.1 SLAM 程序流程

基于前几章内容，本文设计的 SLAM 方法的程序流程概括为：

(1) 对 RealSense R200 相机标定，获取相机内部参数，存储进 yaml 格式文件中，供程序调用，并校正相机畸变；

(2) 根据已有图像来训练 BOW 字典，为闭环检测环节提供训练好的字典词库；

(3) 设置 RealSense R200 的采集频率和图像分辨率，通过相机实时采集彩色图像和深度图像；

(4) 将采集的图像预处理后，提取 ORB 特征，生成特征描述子，利用改进后的 FLANN 分类树特征匹配方法对当前图像特征与最新关键帧图像进行匹配，通过使用 3D-2D 特征投影解 SVD 求解初始位姿，然后将初始位姿带入到 Point-to-Plane ICP 中迭代最小化点到平面距离，求解出精确位姿；追踪失败情况发生时，通过闭环检测进行重定位；

(5) 同时对当前图像进行闭环检测，检测是否出现闭环，若产生闭环，以闭环帧为基准纠正当前关键帧偏移误差，并纠正与当前关键帧存在共视关系的其他关键帧位姿及其地图点位置；

(6) 在全局优化环节，把当前所有的相机位姿和地图路标点放在一起构造代价函数，优化最小重投影误差，求得最优化后的相机位姿轨迹和地图点位姿；

(7) 当全局优化完成时, 根据当前所有关键帧在全局地图中的位姿, 以及深度关键帧对应深度图像, 构建稠密点云地图 (不显示, 若显示耗费内存过大), 之后将点云转化为八叉树地图, 构建全局占据网格地图。

如图 6-1 所示, 整个算法设计总结构图可以表示为:

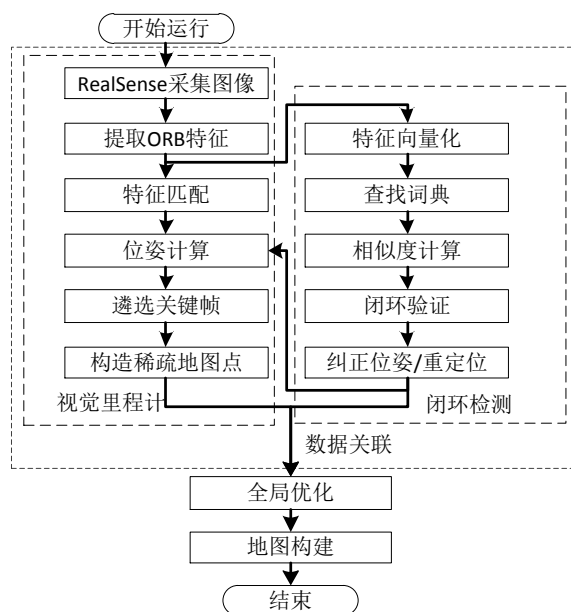


图 6-1 SLAM 方法设计总结构图

Fig.6-1 SLAM method design total structure diagram

6.2 试验设备

(1) 硬件设备

硬件设备包括 UP board (搭载 X5-Z8350 处理器, 4GB RAM 内存, 32GB ROM 存储空间)、服务器 (CPU i5-6500 @3.20GHz, 8GB RAM 内存)、RealSense R200、路由器、1000M 无线网卡和机器人运动平台等。

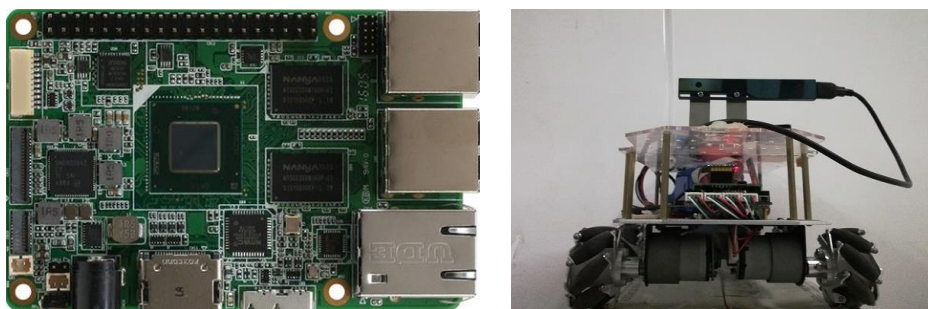


图 6-2 Up board 和机器人运动平台

Fig.6-2 Up board and robotic motion platform

图 6-2 是部分试验设备的实物图。机器人运动平台由 STM32、4 个步进电机控制器、4 个步进电机和车架组成，STM32 作为控制单元驱动 4 个步进电机控制器来控制运动平台的前进、后退和转向。

（2）数据传输流程

服务器和 Up board 都在 Ubuntu16.04 系统环境下的 ROS 框架下进行试验。数据传输流程如图 6-3 中所示，RealSense R200 安装在 Up board 上，Up board 直接读取摄像机彩色视频流和深度视频流。通过无线路由构建连接 Up board 和服务器的 ROS 网络，在 UP board 上把 RealSense R200 采集的图像通过发布话题方式在 ROS 网络中广播，服务器通过订阅解耦的方式接收图像，SLAM 程序在服务器上运行。这种分布式的方法能够将 SLAM 程序的数据读取任务和数据处理任务分开运行。

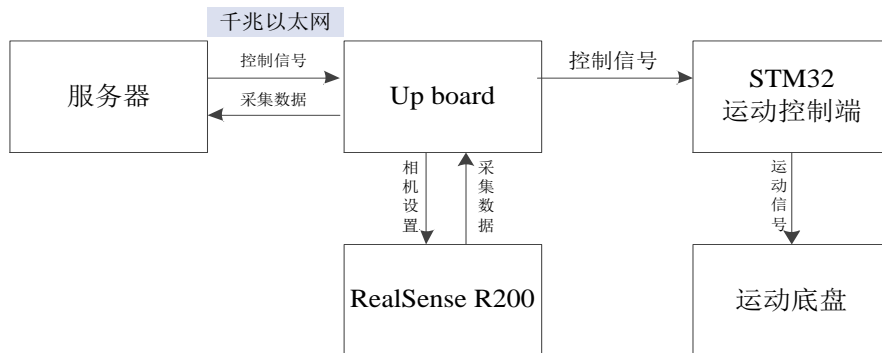


图 6-3 数据传输流程

Fig.6-3 Data transfer process

通过在服务器端操作键盘控制机器人运动平台在室内环境下移动，服务器将运动控制信号通过发布话题到 ROS 网络，Up board 订阅此话题解耦控制信号，然后 Up board 将运行控制信号通过串口发送给 STM32，最终通过 STM32 控制机器人运动平台移动。

6.3 试验设计及试验结果

（1）试验目的

设计试验验证本文设计 SLAM 方法的定位精度，验证算法能否正确的构建出描述周围环境信息的点云地图和八叉树地图。

（2）试验方法

分别通过 TUM（Technical University of Munich）数据集和实时采集运行两种方式进行试验验证，公开数据集带有相机运动的真实轨迹，使用本文 SLAM 方法估计出机

机器人的运动轨迹，通过计算真实轨迹和估计轨迹平均误差的方式评判本文 SLAM 方法的定位精度，对构建的点云地图和八叉树地图进行分析，判断是否能够真实反映相机周围的环境信息。实时采集试验则通过测量机器人真实运动距离与估计的运动距离计算误差信息。

6.3.1 TUM 数据集试验

分别用 freiburg1_xyz 和 freiburg2_pioneer_360 两种场景下的数据集图片来运行本文设计的 SLAM 程序，数据集包括彩色图像和深度图像，图像尺寸为 640×480，其中 TUM 数据集的信息如表 6-1 所示。

表 6-1 数据集介绍

Tab.6-1 Data set introduction

数据集系列	总帧数	时长	范围
freiburg1_xyz	798	30.09s	0.46m x 0.70m x 0.44m
freiburg2_pioneer_360	1225	72.75s	4.24m x 4.38m x 0.06m

本文设计实现的 SLAM 方法在 freiburg1_xyz 数据集上进行验证，如图 6-4、6-5 所示，这是通过本文设计实现 SLAM 方法估计出的机器人运动轨迹，点云地图及占据网格地图。从试验的结果来看，本文方法对 freiburg1_xyz 这种小范围内的机器人移动轨迹估计效果较好，与真实轨迹的平均误差仅为 0.01m，定位精度良好，本文实现的 SLAM 方法构造出的点云地图非常规整，细节非常精细，没有出现错乱拼接的现象，八叉树地图能够直观的描述出空间中的占据情况。

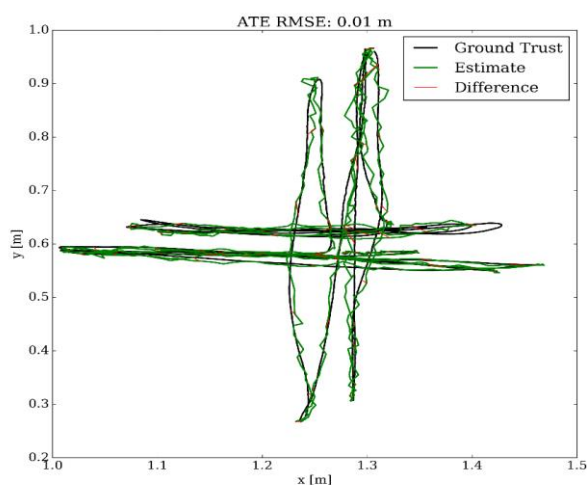


图 6-4 freiburg1_xyz 运动轨迹

Fig.6-4 Freiburg1_xyz motion track

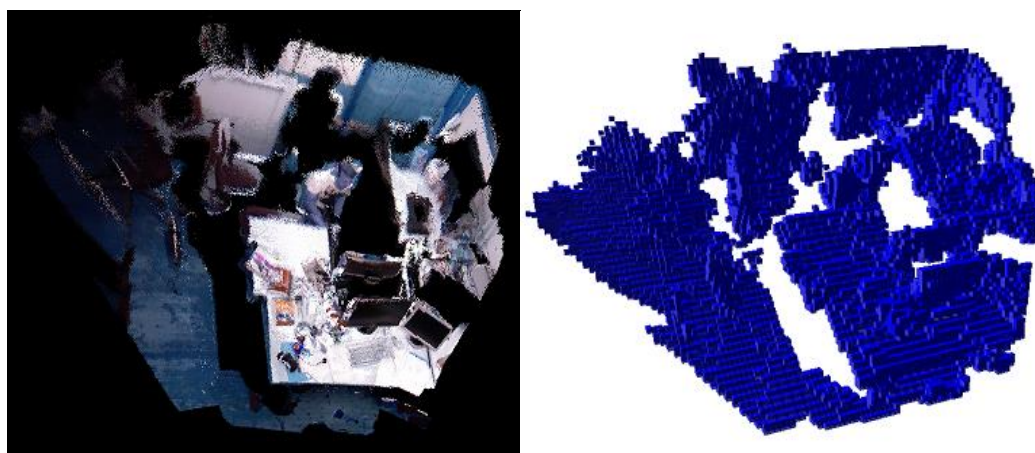


图 6-5 freiburg1_xyz 点云地图和八叉树地图

Fig.6-5 Freiburg1_xyz point cloud map and otcomap

本文设计实现的SLAM方法在freiburg2_pioneer_360数据集上进行验证,如图6-6、6-7所示,这是通过本文实现的SLAM方法估计出的机器人运动轨迹、点云地图及占据网格地图。本文方法对freiburg2_pioneer_360这种较大范围场景,能较好的估计出大致轨迹,从定位精度评估结果来看,根据SLAM方法估计的运动轨迹偏差相比起小场景来说较大,与真实轨迹的平均误差为0.24m,构造的点云地图在中心部位区域拼接良好,四周因采集频率少、传感器测量范围限制导致点云出现缺失现象,八叉树地图能够较好的描述出空间中的占据情况。

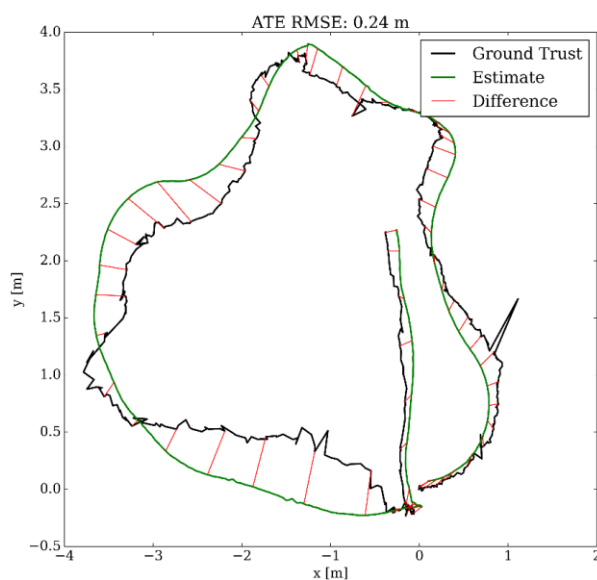


图 6-6 Freiburg2_pioneer_360 运动轨迹

Fig.6-6 Freiburg2_pioneer_360 motion track

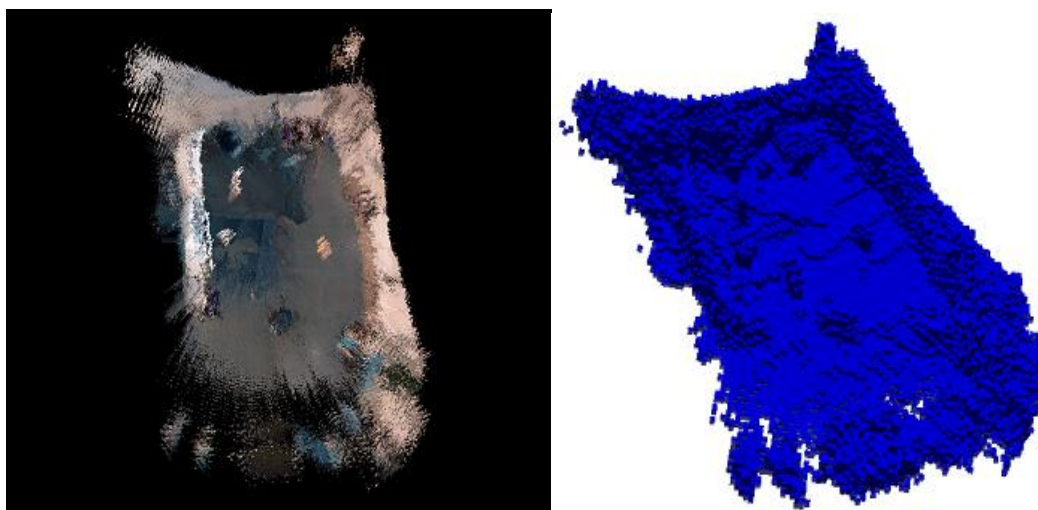


图 6-7 Freiburg2_pioneer_360 点云地图和八叉树地图

Fig.6-7 Freiburg2_pioneer_360 point cloud map and otcomap

6.3.2 实时采集试验

在山东农业大学学实楼 203 实验室内进行了实时采集运行试验。表 6-2 数据表示实验室中的两处不同试验场景下的机器人运动时长和范围。

表 6-2 实时数据采集场景

Tab.6-2 Real-time data collection scenario

实时运动场景	时长	范围
场景 A 处	65.08s	2.40m×4.15m×0.02m
场景 B 处	83.67s	3.53m×2.31m×0.02m

将 Up board 和服务器设置处于同一 ROS 网络中，将 RealSense R200 深度相机的相机模式设为 manual（手动模式），然后将彩色视频流的分辨率设为 640×480，fps 设为 20，深度视频流的分辨率设为 640×480，fps 设为 20。图 6-8 是场景 A 处实时试验过程，在实验室场景 A 处运行本文实现的 SLAM 程序，计算出的相机位姿轨迹、点云地图、占据网格地图如图 6-9、6-10 所示。



图 6-8 场景 A 处的试验过程

Fig.6-8 Test procedure at scene A

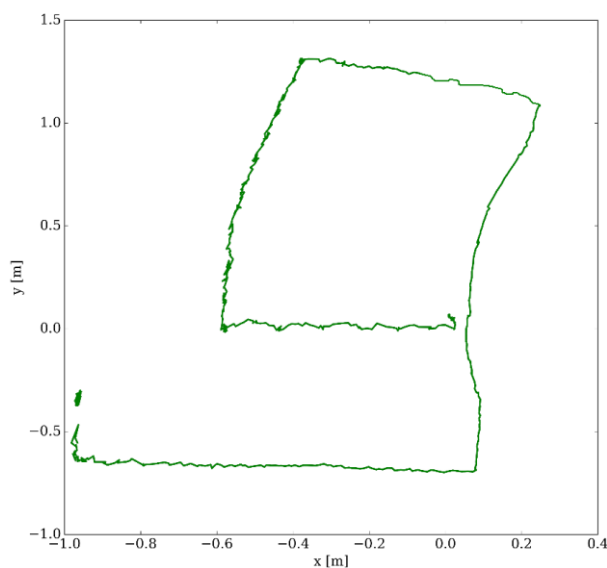


图 6-9 实时运行时运动轨迹

Fig.6-9 Real-time runtime trajectories

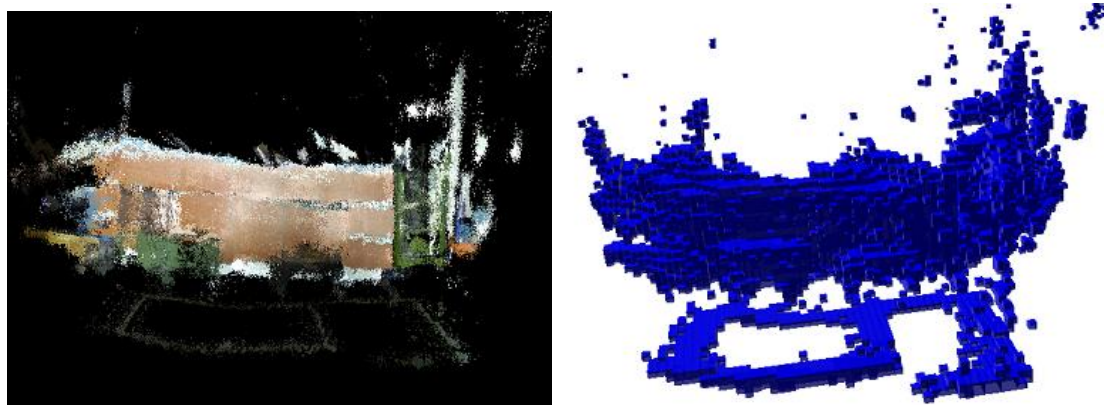


图 6-10 实时运行时构建点云地图和八叉树地图

Fig.6-10 Building point cloud maps and octomap in real-time runtime

通过在地面规划好机器人的运动轨迹后，使用钢尺测量机器人真实轨迹和估计出的轨迹，计算出实时采集时本文 SLAM 方法估算的轨迹平均误差。从运行结果来看，本文 SLAM 方法能够实时估计出机器人运动轨迹并构建地图。通过钢尺测量真实运动轨迹，与估计出的轨迹想比较，其定位误差为 0.13m，从构建的点云地图来看，其拼接效果良好，能反映周围环境细节，从构建的八叉树地图来看，其能真实反映三维空间中障碍物信息。

图 6-11 是场景 B 处实时试验过程，在实验室场景 B 处运行本文实现的 SLAM 程序，通过本文 SLAM 方法计算出的相机位姿轨迹、点云地图、占据网格地图如图 6-12、6-13 所示。

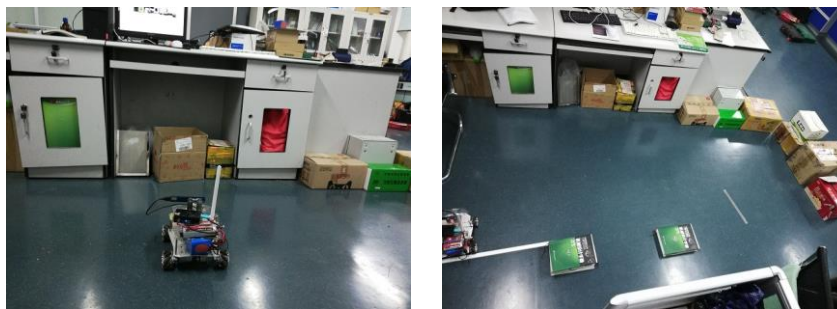


图 6-11 场景 B 处的试验过程

Fig.6-11 Test procedure at scene B

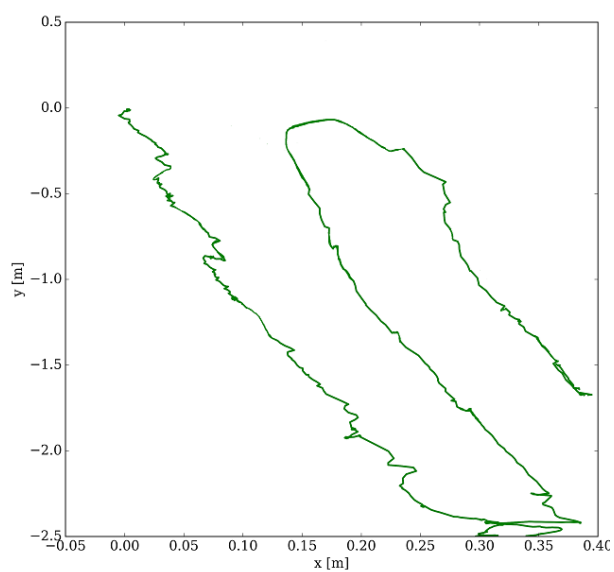


图 6-12 实时运行时运动轨迹

Fig.6-12 Real-time runtime trajectories



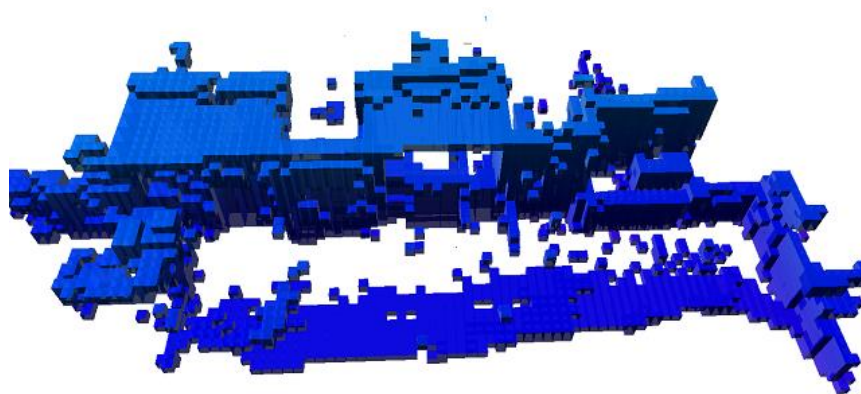


图 6-13 实时运行时构建点云地图和八叉树地图

Fig.6-13 Building point cloud maps and octomap in real-time runtime

在实验室场景 B 处控制机器人运动并运行本文实现的 SLAM 程序。从运行结果来看, 本文 SLAM 方法能够实时估计出机器人运动轨迹并构建地图。通过钢尺测量真实运动轨迹, 与估计出的轨迹想比较, 其定位误差为 0.16m, 从构建的点云地图来看, 其拼接效果良好, 能反映周围环境细节, 从构建的八叉树地图来看, 其能真实反映三维空间中障碍物信息。

6.4 结果讨论

通过 TUM 公开数据集对本文研究算法进行验证, freiburg1_xyz 数据集采集的范围较小, 并且采集路径有很多重复, 因此能够不断的检测到闭环, 机器人位姿和路标空间点构成的约束非常多, 因此优化过后的机器人位姿和真实轨迹比较, 误差很小。本文实现的 SLAM 方法对相机位姿的估计精度非常高, 构建的点云地图的拼接基础是好的机器人定位精度。在机器人位姿估计良好情况下, 点云地图拼接效果也非常好, 细节丰富。评价点云地图和八叉树地图, 其能够真实反映空间中障碍物的占据情况。

freiburg2_pioneer_360 数据集采集的范围远远大于 freiburg1_xyz 的范围, 且其采集过程中没有重复的路径。本文 SLAM 方法运行时, 没有足够的闭环来消除累计误差, 同时因为机器人运动范围较大, 所以本文 SLAM 方法对该数据集的相机位姿估计平均误差较大。从构建的点云地图分析, 由于其中间区域观测次数较多, 因此中间区域细节较丰富, 四周区域因相机的探测距离有限, 因此点云有离散情况, 点云地图和八叉树地图, 能真实反映空间占据情况。

实时采集数据运行本文 SLAM 方法, 机器人平台的运动底盘是麦克纳姆轮, 可以

任意前行或横移，本文在试验时，机器人运动平台的移动多为平移移动，旋转较少。因此，采集的图像主要集中在实验室一侧。通过在实验室场景 A 处及场景 B 处运行本文实现的 SLAM 程序，对相机运动进行估计，其轨迹较为连贯，因为相机视角能够观测到地面，所以点云地图中也反映了运动轨迹，通过对机器人真实运动轨迹做测量，对比估计的轨迹，其误差较小；对照实验室真实场景，能够证明本文所实现的 SLAM 方法地图构建能力较好，构建出的点云地图和八叉树地图能够清晰反映实验室中障碍物的占据信息。

综合来说，本文设计实现的 SLAM 方法能够较好的估计出机器人的运动轨迹，即能实现对机器人的定位功能，并且能实时对环境地图进行三维建模。

6.5 本章小结

本章详细介绍了本文 SLAM 方法的总设计流程，并对实验设备和数据传输路径进行介绍，最后分别通过公开数据集 TUM 和实时采集运行两种方式对本文设计的 SLAM 方法进行试验验证，验证得出本文设计实现的 SLAM 方法能够较好的实现机器人在室内环境下的定位和地图构建。

7 工作总结与展望

同时定位与地图构建 (SLAM) 的研究是移动机器人能够实现自主移动的关键, 本文设计并实现了基于 RealSense 相机的 SLAM 方法, 实现了在室内环境下的定位功能, 并成功对室内环境进行地图构建。通过两种试验方法验证了本文 SLAM 方法的较精确的定位精度及地图构建能力。公开数据集定位精度验证结果: 在小场景下本文 SLAM 方法定位精度为 0.01m, 大场景下定位精度为 0.24m; 实时采集试验定位精度验证结果: 两场景下定位精度分别为 0.13m、0.16m。本文 SLAM 方法构建出的点云地图和八叉树地图能较好的反应机器人所处环境的空间物体占据情况。

7.1 工作总结

主要完成的研究工作总结如下:

(1) 完成 RealSense 标定和数据配准工作。对深度相机进行选型, 选出最适合用来研究 SLAM 的深度相机, 进行开发环境搭建和数据读取, 完成对相机的标定工作, 分别获得彩色相机和双目红外相机的内部参数, 准确的内部参数是后续工作的前提, 并采用了基于时间戳配准的图像对齐方法对齐彩色图像和深度图像, 并选取双边滤波算法对深度图像进行预处理。

(2) 完成特征提取对比试验并改进均匀度。通过试验对比选择速度最快的 ORB 特征提取算法, 保证算法的实时性, 并采用了图像区域分割的方法改善 ORB 特征点集中问题;

(3) 设计并实现了改进的 FLANN 分类树特征匹配算法。在加速匹配过程的同时, 可以去除绝大多数误匹配, 通过设计试验验证了改进后算法的优异性能。

(4) 设计并实现了改进后的位姿计算方法。使用 3D-2D 特征点投影求解 SVD 方式得到初始位姿, 并将初始位姿输入到 Point-to-Plane ICP 中, 通过迭代最小化点到面的距离求解精准位姿。在最后的试验中通过对比估计轨迹和真实轨迹的平均误差, 验证了算法能够较精准计算出机器人的位姿。

(5) 使用词袋模型的方法实现闭环检测环节, 并设计了基于时间的闭环验证方法。实现了基于图优化理论的 SLAM 方法, 构建出两种形式地图。

(6) 设计实现完整 SLAM 程序, 搭建机器人运动平台, 通过采用 TUM 公开数据集和实时采集运行两种方式, 对本文设计实现 SLAM 方法进行了试验验证。经验证,

本文设计实现的 SLAM 方法能够较好的完成机器人的定位在室内环境下的定位和地图构建功能。

7.2 创新点

本文的创新点在于：

(1) 设计实现了改进的 FLANN 分类树特征匹配方法，加速特征匹配，并能有效剔除绝大部分误匹配，相比常规 FLANN 匹配提高了误匹配的剔除能力。

(2) 设计实现了改进的位姿求解方法，首先通过 3D-2D 特征点投影求解 SVD 方式得到初始位姿，然后将初始位姿输入到 Point-to-Plane ICP 中，通过迭代最小化点到面的距离求解精准位姿，避免了常规透视 N 点投影法对正确特征匹配的过多依赖，相比常规迭代最近点方法速度更快。

7.3 展望

(1) 本文对小场景下的机器人运动能够较好的估计估计以及构建地图，较大场景时误差较大，因此在算法精度上需要进一步优化。

(2) 本文方法目前适用于静态场景，后续将扩展到动态场景。

(3) 地图构建完成后将进行路径规划方面工作。

参考文献

- 陈伟. 单目视觉移动机器人的定位与建图研究[D].国防科学技术大学,2008.
- 程一伦. 基于双目视觉的智能车道路环境识别研究[D]. 吉林大学, 2017.
- 董春侠, 司占军, 刘建东. 基于 Intel RealSense 的胶印机感知展示系统设计[J]. 包装工程, 2017, 38(11): 204-208.
- 段华旭. 无人车基于双目视觉室外即时定位与地图构建[D]. 大连理工大学, 2015.
- 付梦印, 吕宪伟, 刘彤, 杨毅, 李星河, 李玉. 基于 RGB-D 数据的实时 SLAM 算法[J]. 机器人, 2015, 37(06): 683-692.
- 郭利进. 大尺度环境下移动机器人同时定位与地图创建研究[D].天津大学,2007.
- 金周英, 白英. 我国机器人发展的政策研究报告[J]. 机器人技术与应用, 2009(02): 1-7.
- 刘建东. 基于 Intel RealSense 技术的实感游戏的研究与实现[D]. 天津科技大学, 2016.
- 刘艳, 李腾飞. 对张正友相机标定法的改进研究[J]. 光学技术, 2014, 40(06): 565-570.
- 李智. 基于 Kinect 的同时定位与地图构建的研究[D].北京交通大学,2016.
- 龙超. 基于 Kinect 和视觉词典的三维 SLAM 算法研究[D]. 浙江大学, 2016.
- 吕铁, 韩娜. 智能制造: 全球趋势与中国战略[J]. 人民论坛·学术前沿, 2015(11): 6-17.
- 庾宇翔. 基于双目视觉的移动焊接机器人障碍物三维定位方法[D]. 上海交通大学, 2011.
- 王雪姣. 基于深度神经网络的 ToF 深度图像去噪研究[D]. 浙江大学, 2016.
- 王忠立, 赵杰, 蔡鹤皋. 大规模环境下基于图优化 SLAM 的后端优化方法[J]. 哈尔滨工业大学学报, 2015, 47(07): 20-25.
- 王勇, 陈国良, 李晓园, 田鹏, 周雨. 一种相机标定辅助的单目视觉室内定位方法[J]. 测绘通报, 2018(02): 35-40.
- 吴子章. 基于全景视觉的机器人 SLAM 方法研究[D].东北大学,2012.
- 徐伟诚. 基于 RealSense~(TM)的上肢辅助复健系统的设计与实现[D]. 北京工业大学, 2016.
- 徐则中. 移动机器人的同时定位和地图构建[D].浙江大学,2004.
- 余俊. 基于双目视觉的机器人目标检测与控制研究[D]. 北京交通大学, 2011.
- 杨森, 吴福朝. 摄像机位姿的加权线性算法[J]. 软件学报, 2011,22(10): 2476-2487.

- 阳鹏程. 基于双目结构光的三维测量技术研究[D]. 浙江大学, 2014.
- 伊星星. 基于 RGB-D 的移动机器人实时定位与建图研究[D]. 哈尔滨工业大学, 2017.
- 曾少青. 基于飞行时间法原理的三维成像系统设计[D]. 湘潭大学, 2017.
- 张垒, 蒋乐天. 基于 RealSense 的三维物体识别算法研究[J]. 信息技术, 2017(10): 78-83.
- 张连华, 王京, 石宗英, 钟宜生. 基于光流的四旋翼直升机鲁棒自主着陆控制[J]. 控制理论与应用, 2016,33(11): 1492-1500.
- 张学贺. 基于双目视觉的六足机器人环境地图构建及运动规划研究[D]. 哈尔滨工业大学, 2016.
- 朱新新. 基于 RealSense 深度信息的枝上柑橘果实近景识别研究[D]. 江苏大学, 2017.
- Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF)[J]. Computer vision and image understanding, 2008, 110(3): 346-359.
- Chen Y, Medioni G. Object modeling by registration of multiple range images[C]. 1991.
- Davison A J. Real-time simultaneous localization and mapping with a camera[C]. Computer Vision, Proceedings Ninth IEEE International Conference on 2003,1403-1410.
- Davison A J, Reid I D, Molton N D, et al. MonoSLAM: real-time single camera SLAM.[J]. IEEE Trans Pattern Anal Mach Intell, 2007, 29(6):1052-1067.
- Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[J]. 2014, 8690:834-849.
- Geiger A, Ziegler J, Stiller C. StereoScan: Dense 3d reconstruction in real-time[C]// Intelligent Vehicles Symposium. IEEE, 2011:963-968.
- Guivant J, Nebot E. Improving computational and memory requirements of simultaneous localization and map building algorithms[C]. IEEE International Conference on Robotics and Automation. Proceedings. ICRA. 2002:2731-2736.
- Hornung A, Kai M W, Bennewitz M, et al. OctoMap: an efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous Robots, 2013, 34(3): 189-206.

- 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.
- Kam H R, Lee S H, Park T, et al. RViz: a toolkit for real domain data visualization[J]. Telecommunication Systems, 2015,60(2): 337-345.
- Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- Lepetit V, Moreno-Noguer F, Fua P. EPnP: An Accurate $O(n)$ Solution to the PnP Problem[J]. International Journal of Computer Vision, 2009, 81(2): 155-166.
- Mur-Artal R, Montiel J M M, Tardós J D. Orb-slam: a versatile and accurate monocular slam system[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.
- Mur-Artal R, Tardós J D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.
- More J J. The Levenberg-Marquardt algorithm: Implementation and theory[J]. Lecture Notes in Mathematics, 1978, 630: 105-116.
- Nourani-Vatani N, Roberts J, Srinivasan M V. Practical visual odometry for car-like vehicles[C]// IEEE International Conference on Robotics and Automation. IEEE Press, 2009: 1411-1417.
- Sim R, Elinas P, Griffin M. Vision-based SLAM using the Rao-Blackwellised particle filter[J]. Ijcai Workshop on Reasoning with Uncertainty in Robotics, 2005, 9(4): 500-509.
- Smith R, Cheeseman P. On the representation and estimation of spatial uncertainty [J]. The International Journal of Robotics Research, 1987, 5(4) : 56-68.
- Strasdat H, Montiel J M M, Davison A J. Visual SLAM:why filter [J]. Image and Vision Computing, 2012, 30(2): 65-77.

- Tan W, Liu H, Dong Z, et al. Robust monocular SLAM in dynamic environments[C]// IEEE International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2013:209-218.
- Thrun S, Koller D, Ghahramani Z, et al. Simultaneous Mapping and Localization with Sparse Extended Information Filters: Theory and Initial Results[M]. Algorithmic Foundations of Robotics V. Springer Berlin Heidelberg, 2004:363-380.
- Zhang Zhengyou. A flexible new technique for camera calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22 (11): 1330-1334.

致 谢

两年多的研究生生活即将结束，这两年多是我人生中的一个重要阶段，在这期间我学习到许多思考问题和解决问题的方法，让我获益良多，不管是现在还是将来这些都是我的宝贵财富。我非常感激老师和同学们对我的帮助，让我克服了许多学习和生活中的困难。

首先感谢我的导师闫银发，在我攻读研究生阶段，闫老师严谨的科研态度、求实的学术精神深深的影响并激励着我。在学习过程中，每当我遇到难点，闫老师都耐心给我指导。生活中，闫老师也对我关怀备至，感谢闫老师在我就业时对我的帮助和支持，能够师从闫老师，我感到非常荣幸。还要感谢李法德、刘莫尘、宋占华、刘平老师，几位老师对我谆谆教导，为我能顺利完成毕业课题提供了极大帮助，谢谢你们。

我还要感谢我的师兄王少刚对我学习工作的指导和帮助，在专业学习方面非常感谢王师兄引领我入门，感谢王师兄对我的学习生活中的监督和爱护，感谢韩守强、许荣浩、李承、王广瑞在生活和学习上的帮助和陪伴，和你们意气相投，相处起来非常愉快。

特别感谢我的家人和我的女朋友宗璐，谢谢你们在生活中对我的关怀和支持，你们是我攻读研究生期间迷茫时的明灯，谢谢你们的理解与照顾，你们一直是我奋斗的动力，以后我会加倍努力。

最后，我衷心的感谢参与审阅、评议本论文的各位老师，还有感谢参与我答辩的各位老师，谢谢你们！

硕士期间发表论文及申请专利情况

- [1] Yinfa Yan, Shunhua Gai , et al. Design and Experiment of Visual SLAM Method Based on RealSense for Mobile Robot[C].Xi'an, CAC,2018.
- [2] 闫银发,盖顺华等. 基于北斗的农业机械立体视觉辅助导航系统设计[J].农业工程,2017,7(05):36-39+50.
- [3] 闫银发, 盖顺华等.一种槽轮式变量排肥掺混装置的控制系統[P].发明及实用新型, 201720052155.9/201710031508.1.
- [4] 刘平, 盖顺华等. 一种多旋翼无人机实时测距和视觉避障系統[P].实用新型, 201621129232.8.
- [5] 闫银发, 韩守强, 刘莫尘, 刘双喜, 李法德, 宋占华, 盖顺华等. 一种电动拖拉机无线充电技术[P]. 实用新型, 201721551204.X.
- [6] 闫银发, 韩守强, 刘双喜, 刘莫尘, 李法德, 宋占华, 盖顺华等. 一种电动拖拉机无线充电智能监控系统[P]. 实用新型, 201721545094.6.
- [7] 闫银发, 韩守强, 宋占华, 李法德, 王金星, 刘双喜, 盖顺华等. 基于测土配方的智能变比配肥撒肥机[P]. 201621446110.1.
- [8] 农田作业机械直线行驶导航软件. 软件著作权. 2016SR161083.
- [9] 农田作业机械直线行驶偏转角度测量软件. 软件著作权.2016SR160995.