中图分类号：TP242.6 论文编号：1028703 18–SZ093
学科分类号：085210

硕士学位论文

室内机器人的单目视觉 SLAM 算法研究

研究生姓名 侯豆
专 业 类 别 工程硕士
专 业 名 称 兵器工程
指 导 教 师 范胜林 副教授

南京航空航天大学

研究生院 自动化学院

二〇一八年三月

Nanjing University of Aeronautics and Astronautics

The Graduate School

College of Nanjing University of Aeronautics and Astronautics

Research on Monocular SLAM of the indoor robot

A Thesis in

Arms Engineering

by

Hou Dou

Advised by

Prof. Fan Shenglin

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Engineering

March, 2018

承诺书

本人声明所呈交的博/硕士学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京航空航天大学或其他教育机构的学位或证书而使用过的材料。

本人授权南京航空航天大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本承诺书）

作者签名：侯亚

日　　期：2018.3.29

摘 要

随着科学技术的进步，人类对机器人智能化要求越来越高，因此如何实现移动机器人的高度智能化，已成为当前研究的热点。而移动机器人的同步定位与地图创建（Simulation Localization and Mapping，SLAM）是研究的关键之一，对于实现移动机器人的智能化有着极其重要的意义。SLAM 所要解决的基本问题就是实现移动机器人在未知环境中通过各种搭载传感器，在运动中建立环境模型和估计自身运动。而摄像机相对于其它传感器的容易安装、重量轻以及得到的信息量大等特点，在 SLAM 的研究中越来越受到欢迎。其中，单目 SLAM 又以其成本低、计算简单等特点引起越来越多研究人员的关注。

因为单目视觉 SLAM 计算量大，需要存储信息多，其发展受到了计算机计算能力和存储能力的限制，这使得单目视觉 SLAM 的发展时间较短，直至 2007 年，第一个纯单目 SLAM 系统才被开发出来。其理论和实践仍然存在着许多亟待攻克的问题。本文主要针对单目 SLAM 的前端部分进行研究，分析前端部分比较经典的算法及其一些缺陷，结合其它计算机视觉方面的知识，提出了相应的改进，从而设计一个比较适用于光线变化和具有较多运动物体场景的 SLAM 前端算法。

主要对于基于特征的单目 SLAM 算法中前端部分的特征点检测、匹配以及位姿估计部分进行研究和改进，主要分为三个部分：

第一，介绍了经典的 SURF 检测方法，因为其作为斑点检测所具有的对于角点的不敏感，提出加入角点检测的基于栅格的 SURF 算法，主要是对图片划分栅格，对图像中 SURF 检测的特征点较少的区域，进行角点检测，从而扩大特征点在场景中的分布范围。

第二，分析各种匹配算法，选用在各种光线变化下都可以有稳定匹配点的快速视网膜（Fast Retina Keypoint，FREAK）匹配算法。因为该匹配算法在光线变化上匹配正确率较低，所以研究了在简化的 FREAK 采样模型中加入线性插值的改进算法，具体的是在不影响算法在光线和分辨率变化下的匹配率，对其采样模型进行简化并对描述子进行线性插值，从而在光线变化和分辨率变化的场景中获得更高的匹配率。

第三，分析传统算法随机抽样一致性算法（Random Sampling Consensus, RANSAC）和相应的改进算法，1 点 RANSAC 和基于扩展卡尔曼滤波（Extended Kalman Filtering,EKF）1 点 RANSAC 算法，阐述了它们各自存在的缺陷。然后提出具有动态物体鲁棒性的 EKF 的 1 点 RANSAC 算法，该算法对已匹配的特征点进行静动态点的区分，加入对场景中运动物体的位姿估计，可以保证在动态物体占据图片较大位置时，可以通过计算场景中动态物体运动模型间接估计摄像机的位姿。实验证明，该前端算法对于室内场景（光线变化和移动物体较多的场景）

具有较好的鲁棒性。

关键词：机器人定位与建图；位姿估计；单目机器人；抽样一致性算法

ABSTRACT

With the progress of scientific technology, requirements on intelligence are much higher, and how to realize the highly intelligent of mobile robots become the hotspot of current research. Simultaneous localization and mapping (SLAM) of mobile robots are one of main research in the study. They are also extremely important for realizing the high intelligence of mobile robots. The basic problem SLAM has solved is that the mobile robot with a variety of sensors in the unknown environment can establish the environment model and estimate its own motion. Compared with other sensors, the camera is easy to install and light and can receive large amount of information, so the vision-based SLAM research is more and more attractive for SLAM researchers. In addition, monocular SLAM has attracted more and more researchers' attention due to its low cost and easy calculation.

Because monocular SLAM has large amount of data need to be computed and be stored, its development has been limited by its computational and storage capabilities, which limits developments of the Monocular SLAM. Until 2007, the first pure monocular SLAM system was put forward. There are still many problems to be overcome in its theory and practice.This thesis mainly are based on the front-end part of the monocular SLAM. This thesis analyzes the front-end of visual SLAM of the classic algorithms and their disadvantages. Combined with knowledge of other computer vision, design a SLAM front-end algorithm that is more suitable for light changes and scenes with more moving objects This thesis mainly studies and improves the keypoints of detection and matching, and pose estimation of the front-end part of the feature-based SLAM algorithm, which is mainly divided into three parts:

Firstly, the classic SURF detection algorithm is introduced. Because of its insensitivity to corners, the algorithm is proposed that SURF based on grids is added with the corner detection. The algorithm divides image and detect the corners in the parts of image that has less keypoints, which expand the distribution of keypoints in the scene.

Secondly, various matching algorithms are analyzed and FREAK which is more stable numbers in the various scenes of lightness changing algorithm is chosen. It is proposed to the linearly interpolation of FREAK ,that simplify the sampling model and linearly interpolate the descriptors so as to obtain more matching keypoints in the scenes of lightness changing and resolution changing.

Thirdly, the RANSAC algorithm, the 1-point RANSAC algorithm and the 1-point RANSAC algorithm for EKF are introduced to and state their shortcomings. Then, the 1-point RANSAC for

EKF algorithm with good robustness in a dynamic scene. It is improved by distinguishing the static and dynamic regions for each image, where the keypoints matched are located. With the position and pose estimation of the moving objects in the scene, it can more accurately calculate the pose of camera in scene with moving objects.

The results show that this algorithm has better robustness to indoor scenes (scenes with many changes of lightness and moving objects).

Keywords: SLAM; Motion estimation; Monocular robot; RANSAC

目 录

图表清单

第一章 绪论

1.1 研究的意义和背景

1959 年第一台工业机器人诞生，标志着机器人历史的开始。这个时期的机器人只能通过所编写的程序完成一些简单重复性的工作，不具有对外界信息反馈的能力。直到 1972 年，斯坦福研究院(SRI)研制出的 Shakey 移动机器人，标志着移动机器人历史的开始。移动机器人作为一个综合系统，结合了环境感知、动态决策与规划、行为控制与执行等功能，其技术在这短短几十年中得到了飞速的发展，它的应用范围逐渐从传统的工业领域扩展到服务、救援、军事、海洋开发、宇宙探索等方面。移动机器人广泛的运用前景使得其成为目前科学技术发展最活跃的领域之一，得到世界各国的普遍关注。

最开始研制移动机器人的目的是让机器人代替人类从事危险、恶劣（如辐射、有毒）环境下作业和人所不及的（如宇宙空间、水下）环境作业，随着研究的深入和计算机技术的进步，人类不再满足对机器人的远程遥控，如何让移动机器人实现自主性成为了研究的热点。而其中如何在未知环境下让移动机器人事先预知所在环境的地图和定位信息是自主移动机器人研究的基础，由此诞生了移动机器人的同时定位与地图构建（Simultaneous Localization and Mapping，SLAM）理论。对 SLAM 的定义是机器人从未知环境中的未知地点出发，在移动过程中通过传感器不断观测环境实现定位和姿态估算，再根据定位的位置构建增量式地图，从而达到同时定位和导航的目的。

SLAM 算法最早是在 1988 年由 Randall Smith 和 Perter Cheseseman 提出[1]，通过设计轮式移动机器人的运动模型和观测模型，结合贝叶斯理论，从而实现移动机器人在未知环境的运动估计。Leoanrd 和 Durrant-Whyte[2] 提出移动机器人导航的三个最基本的关键问题，“我在哪？”，“我周围环境怎么样？”和“怎样才能到达目标地点？”，SLAM 试图要解决的就是第一个“我在哪儿？”（即定位）的问题。回答了这个问题，就可以实现对自身和周边环境的空间认知，然后在这个基础上进行路径规划和检测和躲避障碍物。所以，SLAM 是实现移动机器人自主性和智能化的基础和重要问题，其学术价值和应用价值决定了它是实现全自主移动机器人的关键技术。

SLAM 常用的外部传感器有激光传感器、超声传感器以及视觉传感器。相较于激光传感器和超声传感器，视觉传感器具有成本低，重量轻，容易安装，拍下来的图像含有丰富的信息，特征区分度高等优点，另外随着计算机硬件能力的提升，在小型 PC、嵌入式设备乃至移动设备上运行实时视觉 SLAM 已成为可能。视觉 SLAM 是指相机作为唯一外部传感器的 SLAM，根

据采用的视觉传感器的不同，可将其分为三类：单目视觉 SLAM、立体视觉 SLAM 以及 RGB-D SLAM。RGB-D SLAM 可以同时获得彩色图像和深度图像这两种图像，然后通过深度图像获取深度信息，同时获得稠密地图，但是因其成本高、体积大、探测距离短，应用环境有限。现今使用的立体视觉 SLAM 主要以双目视觉为主。双目视觉 SLAM 是利用外极线几何约束的原理去匹配左右两个相机的特征，从而能够在当前帧速率的条件下直接提取完整的特征数据，可以直接解决系统地图初始化问题和对深度的估计问[3]。但是双目视觉 SLAM 系统设计复杂，成本高且视角范围受限，只能对一定尺度进行可靠测量，缺乏灵活度。与双目 SLAM 相反，单目 SLAM 只使用一个视觉传感器，这使得其应用更加灵活、简单且成本低。但是，又因单目 SLAM 只使用一个传感器，同时刻只能获取一张图片，这样只能靠对相邻两帧图像进行对比来获得方向信息和深度信息，这使得很难对摄像机进行初始定位和恢复场景深度，具有尺度不确定性。现如今的单目 SLAM 主要通过对相邻两帧图像进行匹配，通过匹配点计算摄像头位姿变换，然后通过对两幅图进行三角测距来得到深度信息，最后进行迭代来实现同时定位与建图。但是因单目视觉的尺度不确定性，三角计算测量的深度信息仍然存在局限性。虽然单目 SLAM 目前仍受到理论和技术的限制，但其显著的优势仍引起了广泛的研究。

1.2 视觉 SLAM 相关国内外研究概述

1.2.1 国外研究概述

自 SLAM 概念的提出到如今，SLAM 系统使用的传感器不断拓展，同时随着 21 世纪计算机技术的发展，SLAM 的运用不在局限于几种传统的传感器。视觉传感器以它便宜方便等优势成为了研究的热点。

早在 19 世纪 80 年代末，Moravec 就已经单独用视觉传感器来估计移动运载工具的运动状态，他不仅介绍了运动估计的流程（主要功能模块现如今仍被使用），还提出了最早的图像角点检测算法 Moravec corner detector。Matthies[4]和 Shafer[5]结合 Maravec 的方法，使用双目视觉系统来检测和跟踪特征角点。他们使用特征地图中的特征错误协方差矩阵来替代 Moravec 方法中表示系统不确定性的一个常量表达式，实验证明该方法可以将移动小车的轨迹误差控制在 2% 的范围以内，比 Moravec 的方法更优。在“行星漫步者”机器人平台上，Lacroix 等人[6]通过一种新的方法，即分析候选点的相关函数取值来选取特征点，然后对左右两个摄像头选取的特征点进行立体匹配，从而得出特征点的深度信息和机器人的位置。Nister[7]等人提出用随机抽样一致性（Random Sampling Consensus, RANSAC）[8]来剔除错误的匹配点，从而增强 SLAM 数据关联的稳定性和系统的精度。

对比双目 SLAM 的发展，因单目 SLAM 理论和硬件设施的限制，发展较慢，且在实践方面有很大不足。直至 2007 年第一个纯视觉的单目 SLAM 系统 MonoSLAM[9] (Monocular

Simultaneously Localization and Mapping，单目实时跟踪与地图生成)才由 Davison 等开发出来，这是一种实时摄像机跟踪系统，可以同时进行摄像机运动跟踪和未知场景的地图构建。在统一的线性系统中，MonoSLAM 系统使用扩展卡尔曼滤波求解摄像机运动参数和三维点云的坐标。因为每一时刻摄像机方位和每个三维点的位置都会存在概率偏差，所以使用椭球来表示三维点，椭球中心表示估计值，体积表示不确定度，投影至二维图片中为一个椭圆形点。MonoSLAM 系统中对每帧图片提取 Shi-Tomasi 角点[10]，采用主动搜索进行特征点匹配[11]。因为 EKF 的引进，该系统计算复杂度较高且存在线性化而造成的不确定性问题，所以只能处理几百个点的小场景。

同年，Murray 和 Klein 发表了实时 SLAM 系统 PTAM（Parallel Tracking and Mapping）[12]，它将姿态跟踪（Tracking）和建图（Mapping）分为两个单独的任务并行处理，前者不修改地图，只利用二维-三维的匹配为每一输入帧计算摄像机参数，后者则是负责使用局部集束优化来优化局部结构，从而达到地图的建立、维护和更新。PTAM 系统是第一个用多线程处理 SLAM 的算法，也是 SLAM 系统第一次将地图优化整合到实时计算中，这使得它成为现代实时 SLAM 系统的标配。另外，如果成功匹配点数不足造成跟踪失败，PTAM 系统会将当前帧与已有关键帧的缩略图进行比较，选择最相似关键帧进行重定位[13]。

2011 年，Newcombe 等人提出了基于直接法的单目 DTAM（Dense tracking and mapping in real-time）系统[14]。DTAM 系统最显著的优点是可以实时恢复场景三维模型，所以它既允许 AR（增强现实，Augmented Reality）应用中的虚拟物体和场景发生物理碰撞，又能保证在场景特征缺失、图像模糊等情况下稳定地直接跟踪[15]。在 DTAM 系统种用逆深度（Inverse Depth）[16]来表示深度信息，即将空间离散为 $M \times N \times S$ 三维网格，$M \times N$ 表示图片分辨率，S 表示逆深度分辨率，然后通过帧率的整幅图像对准从而获得相对于稠密地图的相机的 6 自由度位姿。因为 DTAM 系统对为每个像素都恢复稠密的深度图，所以它对特征缺失和分辨率较低的图像有好的鲁棒性，但稠密的深度图的建立需要的计算量大，并且其采用全局优化，这使得 DTAM 系统即使在 GPU（Graphic Processing Unit）上能达到实时的效果，但效率依旧很低。

在 2013 年 Engel 等人提出可同样基于直接法的视觉里程计系统，并且在 2014 年扩展为 LSD-SLAM 系统（Large-scale Direct SLAM）[17]。该系统与 DTAM 系统相比，仅恢复半稠密深度图，每个像素深度独立计算，提高了计算效率。另外系统采样关键帧表达场景，前台线程采用直接法计算当前帧和关键帧之间的相对运动，后台线程从关键帧中抽取像素点并搜索在当前帧上的对应点，从而得到新的逆深度观测值和方差，然后采用扩展卡尔曼滤波更新逆深度图和方差，这些保证可该系统能够得到高度准确的姿态估计和实时重构关键帧的姿态图和半稠密深度图。LSD-SLAM 系统还使用一种基于 $sim(3)$ 的直接跟踪法，计算出尺度漂移的公式，对尺度变化较大的场景有很好的鲁棒性。2015 年，Enge 等人还将 LSD-SLAM 系统扩展到双目相机[18]和全景相机[19]上。

苏黎世大学的 Forster 等人在 2014 年提出了一种半直接的单目视觉里程计方法（Semi-direct monocular Visual Odometry, SVO）[20]。半直接相比于直接法不是对整幅图像进行直接匹配而是通过对稀疏特征块使用直接配准。在位姿计算上，系统通过半直接法得到当前帧位置摄像机位姿的粗略估计，然后计算光度的误差求取更精确的投影位置，根据更精确的投影位置进行位姿与地图点的优化。在深度计算上，SVO 系统采用一种由四个参数描述的高斯-均匀混合分布的逆深度[21],[22]对深度进行推导和更新，即被称为深度滤波器（Depth Filter）。SVO 虽然运行速度快，而且选取的关键点分布比较均匀，但是它没有考虑闭环和重定位，当跟踪丢失后，系统也无法运行，所以 SVO 不能称为完整的 SLAM 系统。2016 年，Forster 等人有对 SVO 系统增加了边缘跟踪，并考虑了 IMU（Inertial measurement unit）的运动先验信息，可以支持大视角相机和多相机系统[23]。

2015 年，Mur-Artal 等人提出基于特征的单目 ORB-SLAM 系统[24]，并于 2016 年拓展为可以支持双目和 RGBD 视觉传感器的 ORB-SLAM2[25]。该系统延续了 PTAM 系统的框架的同时增加了回环检测线程，并对大部分组件都做了改进，主要有以下几点：对所有任务（追踪、地图构建、重定位和闭环控制）采用同一个 ORB 特征[26]，这使得系统更加简单可靠，另外 ORB 特征在没有 GPU 的情况下可以运用到实时图像中，并且具有很好的旋转不变特性；因系统中视图内容关联，所以追踪和地图构建可独立于全局视图工作，在局部视图关联中处理；系统采用了统一的词袋模型（Bag of Words, BoW）进行闭环检测和重定位，并构建数据库提高检测效率；改变人工干预选择初始化的视图，通过使用平面视图的单映射和非平面视图的基本矩阵全自动的选择初始化模型。2017 年，Mur-Artal 等人在 ORB-SLAM 系统中融合了 IMU 信息（Visual Inertial ORB-SLAM）[27]，并采用了预积分[28],[29],[30]的方法对 IMU 的初始化过联合视觉信息进行优化。

TUM 机器视觉组的 Engel 在 2016 年又提出了一种新的基于直接法和稀疏法的视觉里程计系统——DSO 系统[31]（Direct Sparse Odometry）。系统中结合最小光度误差模型和模型参数联合优化方法。DSO 不进行关键点检测和特征点描述计算，而是尝试把每个点投影到所有帧中，计算在各帧中的残差，只要残差在合理范围内，就可以认为这些点是由同一个点投影的。另外，DSO 提出了完整的光度标定方法，认为对相机的曝光时间、暗角、伽马响应等参数进行标定后，能够让直接法更加鲁棒。对于未进行光度标定的相机，DSO 也会在优化中动态估计光度参数。光度标定对于由相机曝光不同引起的图像明暗变化有较好的鲁棒性。但是 DSO 系统仍然不是一个完整的 SLAM 系统，不包含回环检测和重定位，因此仍然存在累积误差。

1.2.2 国内研究概述

相对于国外对于 SLAM 的研究来讲，国内的研究时间短，研究团队少，机器人技术水平低，

在国际上仅有很少重大成果展示出来，特别是单目 SLAM。虽然随着国内对人工智能越来越重视，各个研究单位和企业认识到 SLAM 在人工智能方面的重要性，但是还是无法改变国内对视觉 SLAM 研究的先天不足，而成果也主要集中在少数国家重点高校和一些科研院所。中国科学院自动化研究所的温丰、柴晓杰等人[32]设计了一种新型的人工路标系统-MR 二维码，机器人可用通过视觉系统识别 MR 码来确定自己位置或识别出物体，然后建立机器人运动模型和视觉传感器观测模型，并提出了一种实用的里程计位置估计误差模型，最后用扩展卡尔曼滤波融合视觉信息和里程计信息。虽然该算法提高了机器人定位和构图精度，但依赖于 MR 二维码，不适合室外和大场景环境。针对一点 RANSAC 算法在无人机为载体（摄像机多个轴上的角速度都快速变化）的 SLAM 上存在滤波发散的风险，徐伟杰等人提出了结合扩展卡尔曼运动模型的先验信息的 2 点 RANSAC 算法[33]。该算法可以满足微小型无人机自主飞行 20m 左右。谭伟[34]在硕士论文中提出了一种可以鲁棒处理动态场景的实时单目 SLAM，该系统可以允许部分场景动态变化，场景中有较大遮挡的运动物体的定位和建图有较好的鲁棒性。2017 年，香港科技大学的沈绍劼课题组提出了融合 IMU 和视觉信息的 VINS 系统[35]（Monocular Visual-Inertial Systems），该系统在 ISO（**Independent System Operator**）设备上和无人机控制上取得较好的效果。系统主要采用四元数法对对视觉信息和 IMU 信息进行融合，闭环检测仍采用 BoW 词袋模型，通过全局位姿图对累积误差进行实时校正。

1.3 论文的内容和结构安排

1.3.1 主要工作

本文主要对单目机器人的视觉 SLAM 算法进行研究，研究的主要对象是视觉 SLAM 的前端即视觉里程计部分。通过对视觉 SLAM 的研究现状和基本原理的分析，实践和研究单目视觉 SLAM 的一些流行算法，提出对相应算法的改进，最后通过实验证明改进算法比原有算法更加适用于室内光线变化较大和具有较多移动物体的场景。本文主要的内容如下：

1) 首先介绍了单目视觉 SLAM 的相关国内外研究，并对其原理和框架进行了详细的介绍，阐述了单目视觉 SLAM 算法的重要意义；

2) 介绍前端部分特征点检测算法、匹配算法和位姿估计算法，详细介绍了 SURF 检测算法，并提出引入栅格提高检测的效率。然后介绍基于视网膜的特征点匹配算法 FREAK，在简化采样模型的同时引入线性插值，提高算法在光线变化和分辨率变化上的鲁棒性。最后对求位姿状态的算法基于 EKF 的 1 点 RANSAC 提出了加入区分动静态特征点的算法，提高算法对于有动态物体运动的场景的鲁棒性。

3) 通过实验证明三种改进算法得到了理想的结果，整个系统对于室内这种光线变化较大和动态物体较多的环境中呈现对较好的鲁棒性。

1.3.2 论文框架安排

第一章：绪论。首先介绍了课题研究的相关背景和意义，然后对国内外的单目 SLAM 的研究现状和主要成果进行介绍，最后简述了本文研究的主要内容和结构安排。

第二章：单目 SLAM 算法概述。本章首先对单目 SLAM 的通用的数学原理进行介绍，其中观测模型主要介绍了常见的针孔相机的模型。然后对基于特征的单目 SLAM 框架进行了详细介绍，为后文的研究奠定基础。

第三章：基于栅格的特征点检测。首先说明了好的特征点检测算法的要求，然后详细介绍 SURF 检测算法，并提出该算法的不足。提出在栅格环境对于特征点个数较少的栅格进行 FAST 检测，从而使得整张图片的特征点分布更加均匀。最后通过实验得到，该算法可以改善 SURF 检测算法的不足，增加图片中特征点的数量和扩大了特征点的分布空间。

第四章：基于 FREAK 的特征点匹配。首先简述了如今比较经典的匹配方法，及其优缺点，在其中选择了 FREAK 算法。然后对 FREAK 算法进行详细介绍，因算法在光线变化和分辨率变化上的不足，从而提出对视网膜采样模型进行简化，然后对感受野对进行线性插值。实验证明，改进算法对于光线变化和分辨率变化的鲁棒性较原算法有所改进。

第五章：首先介绍了 RANSAC 算法、1 点 RANSAC 算法以及基于 EKF 的 1 点 RANSAC 算法的原理及其优缺点。然后针对基于 EKF 的 1 点 RANSAC 算法，提出了对特征点进行静动态区分从而减小 EKF 的发散。然后在栅格检测、改进的 FREAK 算法的基础上进行位姿估计，实验发现本文单目 SLAM 前端系统对于室内环境有较好的鲁棒性。

第六章：结论。对全文进行总结，阐述全文的主要工作，以及工作中的不足，对未来的研究方向提出相应的意见。

第二章 单目 SLAM 算法概述

根据对图像信息使用的不同，可以把视觉 SLAM 算法分为基于特征的 SLAM 方法和直接 SLAM 方法。直接法是直接根据像素点的强度来估计相机的运动，避免了特征点的提取和描述。另外根据使用像素点的数量将直接法分为稀疏、半稠密和稠密三种。而基于特征的 SLAM 算法需要对输入的图像中比较显著的特征点（比如角点）进行检测和描述，然后基于 2-D 或 3-D 的特征匹配进行摄像机的位姿估算和对环境进行建图。不同于直接法，基于特征的方法不需要对整幅图像进行处理，计算复杂度较低，所运用比较广泛。但是基于特征的方法会忽略特征点以外的所有信息，如特征点过少或分布不均，可能会导致摄像机位姿估计失败或误差较大。因本文对基于特征的 SLAM 算法进行研究，所以本章主要对其进行介绍。

2.1 视觉 SLAM 的数学定义

因为只考虑摄像机拍摄的视频中的每帧图片所显示的运动情况，所以将某连续时间段的运动离散化成时刻 $t=1,\cdots,K$ 当中的运动状态。设 x 为摄像机的位置，则每个时刻的位置记为 $x_1,\cdots,x_K$，构成了摄像机运动的轨迹。设地图由 N 个路标组成，设路标点为 $y_1,\cdots,y_N$。

设摄像机自身运动的数学模型是：

$$x_k = f\left(x_{k-1},u_k,\omega_k\right) \tag{2-1}$$

其中，u_k 是输入值，即运动传感器的读数，ω_k 为噪音。f 是一个指代任意的运动情况的通用方程，被称之为运动方程。

摄像机在 x_k 位置上观测到路标 y_j，从而产生一个观测数据 $z_{k,j}$，用一个函数 h 描述它们之间的关系得：

$$z_{k,j} = h\left(y_i,x_k,\upsilon_{k,j}\right) \tag{2-2}$$

$\upsilon_{k,j}$ 是观测噪音。

(2.1)和(2.2)是最基本的 SLAM 问题。为了求解定位问题（估计 x）和建图问题（估计 y），把 SLAM 问题建模成一个状态估计问题。状态估计与(2.1)、(2.2)的具体形式和噪音服从的分配规则有关。按照运动和观测方程是否为线性划分系统的线性/非线性，噪音是否服从高斯分布划分系统是高斯/非高斯。其中最简单的是线性高斯系统（Linear Gaussian, LG 系统），它可以通过卡尔曼滤波器给出无偏最优估计。对于复杂的非线性非高斯的系统（Non-Linwar Nor-Gaussian,

NLNG 系统），采用 EKF 滤波器或非线性优化去求解。

2.1.1 三维刚体运动模型

为了求出摄像机自身运动模型，要对摄像机的位置进行参数化，即求其位姿进行描述。位姿包括旋转和平移，因为平移的运动比较简单，以下主要对旋转进行分析。

摄像机在运动过程中，摄像机坐标系一直不断移动，所以对于摄像机的运动变化可以简化成两个坐标系之间的变换关系。为了求这个变化关系，常见的做法是设定一个固定不变的惯性坐标系（或称世界坐标系），然后将点通过矩阵 T 从摄像机坐标系转换到惯性坐标系上。因为摄像机运动是刚体运动，所以同一个点在不同坐标系下满足长度和夹角不变，所以称在此状况下的坐标系变换为欧式变换。欧式变换一般由一个旋转和一个平移部分组成。对于旋转部分，设一个单位正交基 (e_1,e_2,e_3)，经过一次旋转变为 $\left(e_1',e_2',e_3'\right)$。对于一个不随坐标旋转而运动的向量 a，设它在两个坐标系下的坐标分别为 $[a_1,a_2,a_3]^T$ 和 $[a_1',a_2',a_3']^T$，由此得到：

$$\left[e_1,e_2,e_3\right]\begin{bmatrix}a_1\\a_2\\a_3\end{bmatrix}=\left[e_1',e_2',e_3'\right]\begin{bmatrix}a_1'\\a_2'\\a_3'\end{bmatrix} \tag{2-3}$$

左右两边同时左乘 $[e_1^T,e_2^T,e_3^T]^T$ 得：

$$\begin{bmatrix}a_1\\a_2\\a_3\end{bmatrix}=\begin{bmatrix}e_1^Te_1' & e_1^Te_2' & e_1^Te_3'\\e_2^Te_1' & e_2^Te_2' & e_2^Te_3'\\e_3^Te_1' & e_3^Te_2' & e_3^Te_3'\end{bmatrix}\begin{bmatrix}a_1'\\a_2'\\a_3'\end{bmatrix}\triangleq Ra' \tag{2-4}$$

从而得到两个坐标之间关系，而其中矩阵就称为旋转矩阵 R。因为矩阵 R 是一个正交矩阵，所以可以得到：

$$a'=R^{-1}a=R^Ta \tag{2-5}$$

把摄像机的平移向量 t 代入后：

$$a'=R^Ta+t \tag{2-6}$$

这就是欧式空间的坐标变换关系。但是为了让这个变化关系变成线性关系，所以对(2.6)作如下变换：

$$\begin{bmatrix}a'\\1\end{bmatrix}=\begin{bmatrix}R & t\\0^T & 1\end{bmatrix}\begin{bmatrix}a\\1\end{bmatrix}\triangleq T\begin{bmatrix}a\\1\end{bmatrix} \tag{2-7}$$

这种把三维向量的末尾添加 1 所变为四维向量叫做齐次坐标。称式中矩阵 T 为变换矩阵，

$\tilde{a}$ 为 a 的其次坐标。所以两次坐标变化的累加就可以表示为：

$$\tilde{b}=T_1\tilde{a},\tilde{c}=T_2\tilde{b}\Rightarrow\tilde{c}=T_1T_2\tilde{a} \tag{2-8}$$

2.1.2 观测模型

大多单目 SLAM 是采用针孔相机进行图像采集的，所以这里随针孔相机进行模型参数化。针孔相机模型[36]如图 2.1 所示。

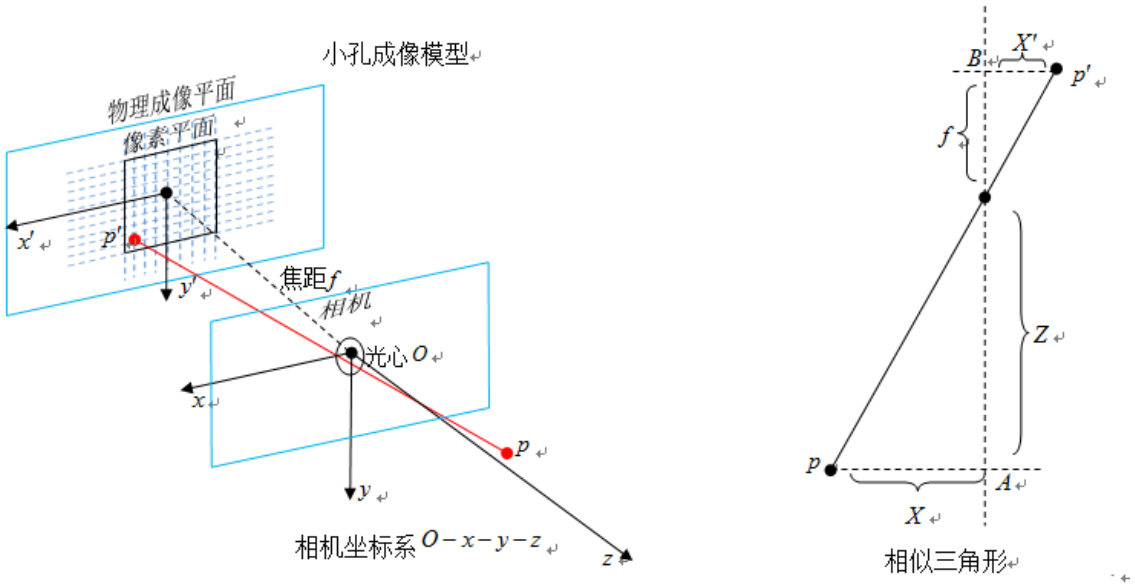


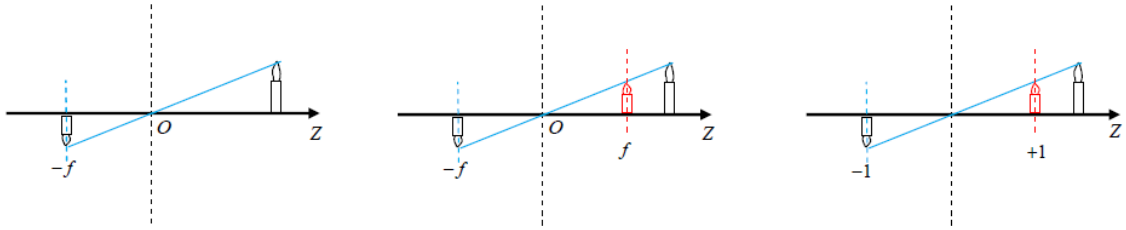
图 2.1 针孔相机模型

设 $O-x-y-z$ 为相机坐标系，设现实世界空间点为 P，经过 O 点投影，在物理成像平面 $O'-x'-y'$ 上的成像点为 P'。设 P 的坐标为 $[X,Y,Z]^T$，P' 点坐标为 $[X',Y',Z']^T$，物理成像平面到小孔的距离为焦距 f。所以由三角相似关系得：

$$\frac{Z}{f}=-\frac{X}{X'}=-\frac{Y}{Y'} \tag{2-9}$$

负号表示成的像是倒立的。为了简化模型，将公式中的负号去掉，可以将成像平面对称到相机前方，使得其与三维空间点一起放在摄像机坐标系同一侧，如图 5.2(b)所示。最后(5.1)可简化为：

$$\frac{Z}{f}=\frac{X}{X'}=\frac{Y}{Y'} \tag{2-10}$$



(a) 真实成像平面　　(b) 对称成像平面　　(c) 归一化成像平面

图 2.2 成像平面

所以：

$$\begin{aligned} X' &= f\frac{X}{Z} \\ Y' &= f\frac{Y}{Z} \end{aligned} \tag{2-11}$$

上式是描述点 P 和 P' 之间的空间关系。由于最终需要获得一个个像素，所以在成像平面上对像进行采样和量化。为了将成像点转换成像素点，把一个像素平面 $o-u-v$ 固定在物理成像平面上，从而得到 P' 点的像素坐标为：$[u,v]^T$ 。

像素坐标系中一般定义图像的左上角为原点 o'， u 轴平行于 x， v 平行于 y 。对其进行缩放和平移就得到了成像平面。假设在 u 轴缩放了 α 倍，在 v 轴缩放了 β 倍，原点平移了 $[c_x,c_y]^T$ 。

所以得到 P' 和坐标 $[u,v]^T$ 的关系为：

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \tag{2-12}$$

将(2.12)代入(2.11)并令 $f_x = \alpha f$ 和 $f_y = \beta f$ ，得：

$$\begin{cases} u = f_x\frac{X}{Z} + c_x \\ v = f_y\frac{Y}{Z} + c_y \end{cases} \tag{2-13}$$

其中， f 的单位是 m ， α,β 的单位是像素/m，因此 f_x, f_y 的单位为像素。将(2.5)式转化为矩阵形式得：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z}\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \triangleq \frac{1}{Z}\mathrm{KP} \tag{2-14}$$

两边同时乘以 Z 得：

$$Z\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}f_x & 0 & c_x\\0 & f_y & c_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix}\triangleq \mathrm{KP} \tag{2-15}$$

以上式中把中间的量组成的矩阵称为相机的内参数矩阵（Camera Intrinsics）K 。虽然相机的内参在出厂之后是固定的，但是有时还是需要自己确定内参，即标定。(2.7)中 P 是在相机坐标系中的坐标，而 P 在相机运动中坐标，是由相机的旋转矩阵 R 和平移向量 t 对其世界坐标 P_ω 进行变换到相机坐标系下。具体转换如下：

$$ZP_{uv}=Z\begin{bmatrix}u\\v\\1\end{bmatrix}=\mathrm{K}\left(RP_\omega+t\right)=\mathrm{K}TP_\omega \tag{2-16}$$

其中，相机的位姿 R,t 成为相机的外参数（Camera Extrinsics）。外参是相机的运动轨迹，是 SLAM 中待估计的值。

为了获得更好的成像效果，需要在相机的前方加上透镜，但是透镜的加入会因透镜本身的形状对光线传播的影响引起径向畸变。这种现象在图像的边缘更加明显。因为实际中透镜一般是中心对称的，所以畸变通常也是径向对称的。另外，因为相机组装过程不能使透镜与成像平面严格平行，会引入切向畸变。为了尽量减小这两种畸变对计算的影响，对其用数学形式进行描述。令平面上任意一点 p 的笛卡尔坐标为 $[x,y]^T$，写成极坐标形式为 $[r,\theta]^T$。r 是点 p 离坐标原点的距离，θ 是和水平轴的夹角。径向畸变可以被看作坐标点在长度方向上即距离原点的长度发生了变化 δr，切向畸变则是坐标点在切线方向上即水平夹角发生了变化 $\delta\theta$。因为径向畸变随着离中心距离的增加而增加，用以下的多项式函数进行纠正：

$$\begin{aligned}x_{corrected}&=x(1+k_1r^2+k_2r^4+k_3r^6)\\y_{corrected}&=y(1+k_1r^2+k_2r^4+k_3r^6)\end{aligned} \tag{2-17}$$

其中 $[x,y]^T$ 是未纠正点的坐标，而 $[x_{corrected},y_{corrected}]^T$ 是已纠正点的坐标，另外它们都是归一化平面的点。

对于切向畸变，使用参数 p_1， p_2 纠正：

$$\begin{aligned}x_{corrected}&=x+2p_1xy+p_2(r^2+2x^2)\\y_{corrected}&=y+p_1(r^2+2y^2)+2p_2xy\end{aligned} \tag{2-18}$$

所以通过式(2.11)(2.12)，可以用五个畸变系数得到相机坐标系中点 $P(X,Y,Z)$ 的正确位置：将三维空间点投影到归一化图像平面，并设归一化坐标为 $[x,y]^T$；对归一化平面的点进行径向畸变和切向畸变的纠正；

$$\begin{cases} x_{corrected} = x(1+k_1r^2+k_2r^4+k_3r^6)+2p_1xy+p_2(r^2+2x^2) \\ y_{corrected} = y(1+k_1r^2+k_2r^4+k_3r^6)+2p_2xy+p_1(r^2+2y^2) \end{cases} \tag{2-19}$$

将已纠正的点通过内参数矩阵投影至像素平面，得到点在图像的正确位置。

$$\begin{cases} u = f_x x_{corrected} + c_x \\ v = f_y y_{corrected} + c_y \end{cases} \tag{2-20}$$

虽然在实际的系统中，研究人员提出过许多模型，但对于普通摄像头来说，针孔模型和径向畸变和切向畸变纠正已经足够满足对观测状态的描述。

2.1.3 系统状态表示

因为基于滤波的单目 SLAM 系统的状态向量不能直接观测到，所以只能依据摄像机的运动和观测状态对其进行估计。由此设系统的状态为 x_t，

$$x_t = \left(x_v, y_1, \cdots, y_n\right)^T \tag{2-21}$$

其中，x_v 是根据摄像机运动模型设定的摄像头状态，y_i 是根据观测模型设定的特征点的状态向量。采用对多数摄像头运动估计准确度比较高的 Moncular SLAM 系统中的运动模型。所以 x_v 可表示为：

$$x_v = (r^{WC}, q^{WC}, v^{WC}, \omega^{WC})^T \tag{2-22}$$

其中，r^{WC} 是摄像头在惯性坐标系中的 3 维位置坐标，q^{WC} 表示单位四元组向量，是记录摄像机相对于世界坐标的方向信息，v^{WC} 和 ω^{WC} 分别是摄像机相对于惯性坐标系 W 的线速度和角速度。

设：

$$y_{iP} = (x_c^W, y_c^W, z_c^W, \theta_i, \beta_i, \rho_i)^T \tag{2-23}$$

x_c^W, y_c^W, z_c^W 是摄像机光心（Optical Center）的场景中的三维位置，是特征首次被摄像机观测到时摄像机的状态；θ_i, β_i 分别是摄像机坐标系下的特征点相对于摄像机中心点的方位角（Azimuth）和仰角（Elevation）；ρ_i 是特征点距离摄像机的逆深度值。其转换成三维空间坐标为：

$$\begin{gathered} y_{iE} = \begin{bmatrix} X_i^W \\ Y_i^W \\ Z_i^W \end{bmatrix} = \begin{bmatrix} x_c^W \\ y_c^W \\ z_c^W \end{bmatrix} + \frac{1}{\rho_i} m(\theta_i, \beta_i) \\ m(\theta_i, \beta_i) = (\sin\theta_i \cos\beta_i, -\sin\beta_i, \cos\theta_i \cos\beta_i)^T \end{gathered} \tag{2-24}$$

$m(\theta_i,\beta_i)$ 是从摄像机指向特征点的单位向量。

2.2 基于特征的单目 SLAM 算法框架

无论是直接视觉 SLAM 还是基于特征的视觉 SLAM 的基本框架都是视觉传感器图片采集、视觉里程计（Visual Odometry, VO）、后端优化（Optimization）、回环检测（Loop Closing）、建图（Mapping）（如图 2.3）。其主要区别是在视觉里程器这个部分，基于特征的 SLAM 主要是对相邻两帧图像的特征点进行检测，然后通过匹配特征点计算摄像机的变换矩阵。而直接 SLAM 则是把图像中大部分像素写进位姿估计中，求出帧之间的相对运动。

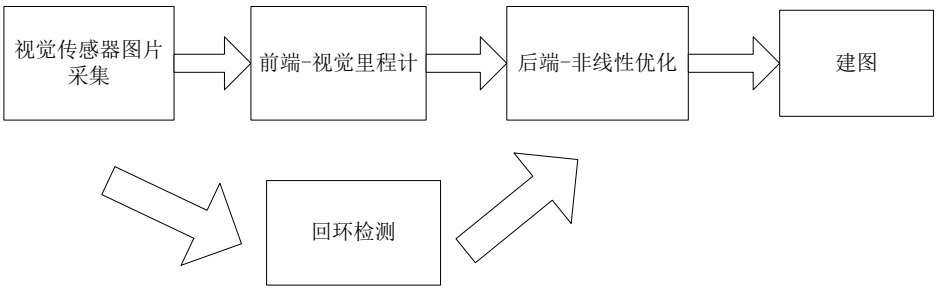


图 2.3 视觉 SLAM 算法框架

特征点法不同于直接法的地方有两处：特征点法是通过最小化重投影误差来确定相机的位姿和地图点的位置，而直接法是由图像间的误差决定的最小化目标函数计算得到相机的位姿和地图点的位置（最小化光度误差，photometric error）；对于数据关联（data association）和位姿估计（pose estimation），直接法是将它们放在统一的非线性优化中，而特征点法是对它们进行分步求解。特征点法先通过特征点匹配得到数据之间的关联，然后根据关联来估计位姿。

2.2.1 视觉里程计

目前基于特征的视觉里程计是 SLAM 算法中的主流方法，这种方法是通过检测两个相邻两帧图像中的特征点，然后对特征点进行描述、匹配，从而得到摄像机的变化姿态，这也是单目 SLAM 中最常用的点特征算法。点特征算法要求选取的特征点不仅可以在视角和光照变化下具有一定的不变性，对模糊和噪音也具有一定的弹性。如今主要的特征点选取方式可以分为基于斑点检测和角点检测。斑点检测主要是对着与周围有着颜色、像素和纹理差别的区域进行检测，主要代表方法有 SIFT 算法（Scale Invariant Feature Transform）[37]和 SURF（Speed-up Robust Feature）算法。角点检测是检测两个或更多灰度边缘交叉的点，主要方法有 Moravec[38]算法、Harris[39]算法和 FAST（Features From Accelerated Segment Test）[40],[41]算法。这两种检测方法中

斑点检测具有更好的辨识度，但是检测速度较慢，而角点检测则相反。为了让相邻两帧图像的特征点进行有效的匹配，还要对特征点进行描述。因为随着计算机计算能力的提高，主流的描述方法是将特征点及其周围的点进行比较，建立一个二进制的矩阵，这种方法相对于其他方法计算速度快，且具有较好的鲁棒性。最具有代表性的算法有 BRIEF（Binary Robust Independent Elementary Feature）[42]、ORB（Orient Fast and Rotated BRIEF）[43]、BRISK（Binary Robust Invariant Scalable Keypoints）[44]以及 FREAK（Fast Retina Keypoint）[45]算法。以上的检测和描述都是为了解决视觉 SLAM 的最关键一步—特征匹配。最简单的特征匹配方式是暴力匹配（Brute-Force Matcher），直接对两帧图像中特征点两两之间进行描述子距离的计算，然后排序，去最近一个作为匹配点。一般对于浮点类型的描述子用欧氏距离进行相似性度量，而二进制描述子使用汉明距离作为相似性度量。暴力匹配法的运算量大，对于特征点数量大时，将无法满足 SLAM 的实时性需求。为了解决这个问题，快速近似最邻法（FLANN）[46]被提出，这种算法更加适合匹配点数量极多的情况。

最终，对已匹配完成的点进行对极几何和三角测量计算，从而得到图像间的运动以及地图点的深度信息。然而对于得到的信息中，存在着各种误差，这不可避免的会出现累计漂移，导致无法建立一致的地图。为了解决这个问题，还需要进行后端优化和回环检测模块。

2.2.2 后端优化

后端优化主要对 SLAM 过程中的噪音问题进行处理。前端算法很多都是基于理想的情况，但是实际上，无论多么精确的传感器都有噪音。这些噪音在每一步计算上造成的误差经过一段时间的累积会造成轨迹出现了大幅度偏差，甚至算法无法继续运算。所以要在估计摄像机位姿运动的同时，还要考虑估计中所带有的误差，上一步计算的误差是如何传递到下一步的，这些误差对于运动估计有多大的影响。后端优化就是考虑如何从这些带有误差的数据中，估计整个系统的状态以及这个状态的估计的不确定性有多大，即最大后验概率估计（Maximum-a-Posterior, MAP）。

在视觉 SLAM 系统中，视觉里程计作为前端部分，为后端提供初始数据，而后端对这些数据进行优化，与传感器无关。所以后端优化是不仅仅是视觉 SLAM 所要研究的问题，它早期被称为空间状态不确定估计（Spatial Uncertainty）[47],[48]，也直接被认为是 SLAM 的研究。最早 Smith 等人采用扩展卡尔曼滤波（Extended Kalman Filter, EKF）实现 SLAM。主要思想是用状态方程存储相邻两帧之间的运动方程和观测方程，然后根据以往的数值计算一个卡尔曼增益来补偿状态方程内噪音影响。但是由于 EKF 的原因，SLAM 算法会存在计算量大，和线性化导致的不确定性等问题。后来，SLAM 研究者们提出将捆集优化方法（Bundle Adjustment, BA）[49]引入 SLAM 算法中[50],。它与滤波器方法的不同的是，它是考虑之前所有帧的信息，然后通过

优化将误差平均分到每一次的观测当中。BA 在 SLAM 中多数以图优化方法[51],[52]（Graph Optimization）展现，图优化不仅可以直观表示优化问题，还可以利用稀疏代数快速求解，慢慢成为视觉 SLAM 主流的优化方法。

2.2.3 回环检测

回环检测，又称闭环检测（Loop Closure Detection），主要解决位置估计随时间漂移的问题，是指让机器人识别曾经到达过的场景的能力，其实际上是一种检测数据相似度的算法。现在在视觉 SLAM 中多数采用比较成熟的词袋模型。词袋模型现将局部特征描述子集合聚类，建立视觉词典，然后分别寻找每个图中包含的“单词”。

回环检测会出现两种错误的结果：假阳性（False Positive）和假阴性（False Negative）。假阳性又称感知偏差，是指实际场景中不同的场景被当成同一个。假阴性又称感知变异，是指实际场景中同一个场景被当成两个。回环检测主要是为了解决假阴性，所以假阳性的存在严重影响最后的地图构造。研究中通常采用准确率 Presicion 和召回率 Recall 曲线来评价算法。

2.2.4 建图

建图是对场景进行构建地图的过程。因为 SLAM 的应用环境不同，构建的地图也是不同的，它可以是简单的空间点的集合也可以是一个漂亮的 3D 模型。地图可以大体分为度量地图和拓扑地图两种。

度量地图（Metric Map）通常指 2D/3D 的网格地图，它比较注重精确地表示地图中物体的位置关系，通常可以划分为稀疏型和稠密型。稀疏型地图比较抽象，一般只表示具有意义的路标，不是路标的部分可以忽略。而稠密型地图则要把所有可看的东西进行建模。稀疏型地图主要用于定位，而导航则需要稠密型地图。度量地图相较于其它地图需要大量的存储空间，另外大规模的度量地图可能会出现一致性问题，很小的误差都有可能导致地图失效。

相较于度量地图拓扑地图（Topological Map）是一种更加紧凑的地图，它将地图抽象为点和边，只考虑节点之间的连通性，放松了对精确位置的需要，抛弃了细节问题。但是对于具有复杂结构的场景，拓扑地图的表达可能就会出现问题。

2.3 本章小结

本章主要介绍了单目视觉 SLAM 的物理模型，并将其转换成数学模型，其后又介绍了视觉 SLAM 的基本框架，并对每一个过程进行了具体分析，重点介绍了基于特征的视觉 SLAM，为后续章节的分析和改进做了铺垫。

第三章 基于栅格的特征检测方法

图像特征检测是计算机视觉和图像处理方面的一个初级运算，是作为数字图像算法中的最基础、最重要的步骤之一，也是进行视觉 SLAM 算法研究的重要组成部分之一。视觉 SLAM 中比较常用的特征点检测算法有 SIFT 特征检测、SURF 特征检测以及 FAST 角点检测等。特征点选取的合理性对视觉 SLAM 算法的性能有直接的影响，所以在特征点提取过程中，希望特征点可以满足尺度不变性，旋转不变性，有较高的重复率以及可以均匀分布在图片中。本章主要为了实现特征点可以更均匀的分布在图片中，采用栅格法[53]对 SURF 进行相应的改进。

3.1 SURF 特征检测

SURF(Speed Up Robust Features)是由 Herbert Bay 提出的一种稳健的局部特征点检测和描述算法。SURF 的检测方法是基于 Hessian 矩阵，但是使用一个非常简单的估计方法。它通过采用积分图像来减少计算时间，因此 SURF 检测方法又可以称作“FAST-Hessian”检测方法。它是对 SIFT 算法的改进，提高了算法的实时性。

3.1.1 积分图像

积分图像[54]是为了能更快的计算矩形特征而使用的对图像的一种转换方式。积分图像中，点(x,y)的像素值是其上面和左面的像素值的和（如公式(3-1)所示）。

$$ii(x,y)=\sum_{x'\le x,y'\le y} i(x',y') \tag{3-1}$$

其中$ii(x,y)$ 是积分图像，$i(x,y)$是初始图像。

使用下面的复式：

$$s(x,y)=s(x,y-1)+i(x,y)$$

$$ii(x,y)=ii(x-1,y)+s(x,y) \tag{3-2}$$

其中$s(x,y)$是行的累积和，$s(x,-1)=0$，和$ii(-1,y)=0$。积分图像可以通过原始图像用上式计算得到。

在积分图像中，任意矩形内像素值和都可以通过四个数组参数计算得到。那么两个矩形内像素值和的差就可以通过八个参数计算得到。如图 3.1 中位置 1 的像素值等于 A，位置 2 的像

素值等于 $A+B$，位置 3 的像素值等于 $A+C$，位置 4 的像素值等于 $A+B+C+D$，所以矩形 D 内像素值的和等于 $4+1-(2+3)$。所以，六个参数可以计算上面被定义的两个矩形内像素值和（包括毗邻的矩形内像素和），八个参数可以计算三个矩形，以及九个可以计算四个矩形。

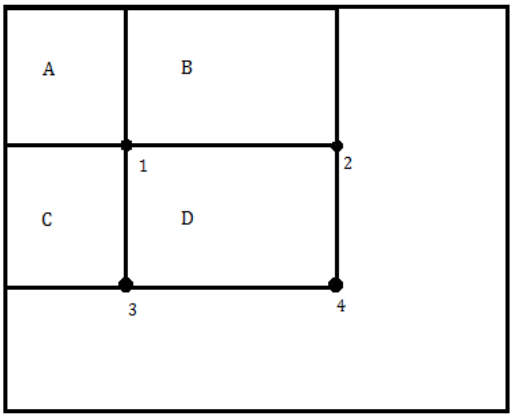


图 3.1 积分图像计算示意图

3.1.2 FAST-Hessian 检测

因为 Hessian 矩阵在计算时间和精度上具有的良好的性能，SURF 在选择在 Hessian 矩阵上构建兴趣点。区别于采用不同的方式去表示位置和尺度的检测方法（比如 Hessian-Laplace 检测方法[55]），SURF 仅使用 Hessian 矩阵的行列式对其进行表示，这样大大提高了兴趣点检测的实时性。在图片 I 中选取一个点 $X=(x,y)$，在尺度为 σ 时，X 的 Hessian 矩阵定义如下：

$$H(X,\sigma)=\begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix} \tag{3-3}$$

其中，$L_{xx}(X,\sigma)$ $L_{xy}(X,\sigma)$ 和 $L_{yy}(X,\sigma)$ 同是图像 I 在点 X 处与标准高斯函数的二阶导数 $\frac{\partial^2}{\partial x^2}g(\sigma)$ 的卷积。

高斯函数是尺度空间的分析最理想的方法。实际上，高斯函数在使用过程中还是需要离散化和修正的（如图 3.2），即使使用高斯滤波器，结果图像一旦被采样，混叠仍旧会发生。另外，在一维的情况，可以证明低分辨率下不会出现新的结构的原理不能应用在二维的情况，因此高斯函数的重要性似乎在这方面被高估了。因为在任何情况下，高斯函数滤波都不是最理想的，Bay 等人采用盒子滤波对高斯二阶微分模板进行近似处理（如图 3.3）。通过使用积分图像可以快速的计算近似高斯二阶导数，且独立于尺度。其结果在性能上相当于使用了离散化和被修正后的高斯函数。

图 3.2 和图 3.3 中[56]，是模板尺寸为 9×9 的作为最小尺度空间的盒子滤波在 σ=1.2 对高斯

二阶导数的近似估计值。记近似值为 D_{xx}、D_{xy} 和 D_{yy}。为了提高计算效率，简化矩形区域的权值，然后由几个矩形区域组成的简化后的模板上对每个区域内填充相同的值，如图 3.3 所示，黑色区域为负数，白的为正数，灰色为 0。为了保持高斯核与近似高斯核的一致性，还需要更近一步平衡 Hessian 的行列式的相关权重 $\frac{|L_{xy}(1.2)|_F|D_{xx}(9)|_F}{|L_{xx}(1.2)|_F|D_{xy}(9)|_F}=0.912...\simeq 0.9$，其中 $|x|_F$ 是 Frobenius 范数。所以行列式的值为：

$$det\left(H_{approx}\right)=D_{xx}D_{yy}-\left(0.9D_{xy}\right)^2 \tag{3-4}$$

另外，响应值要根据滤波器的大小进行了归一化处理，以保证任意大小的滤波器的 Frobenius 范数的一致性。

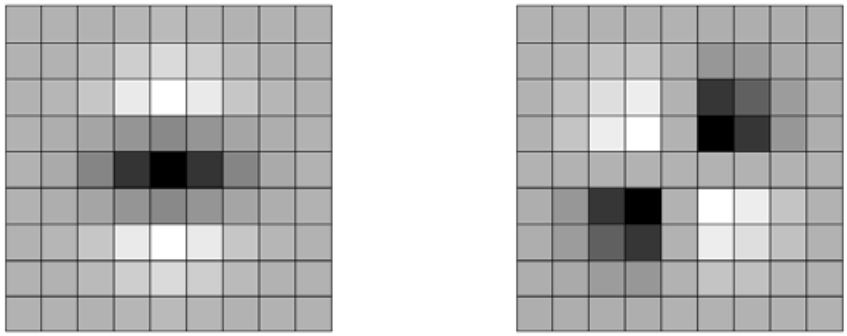


图 3.2 *y* 和 *xy* 方向局部高斯二阶微分模板

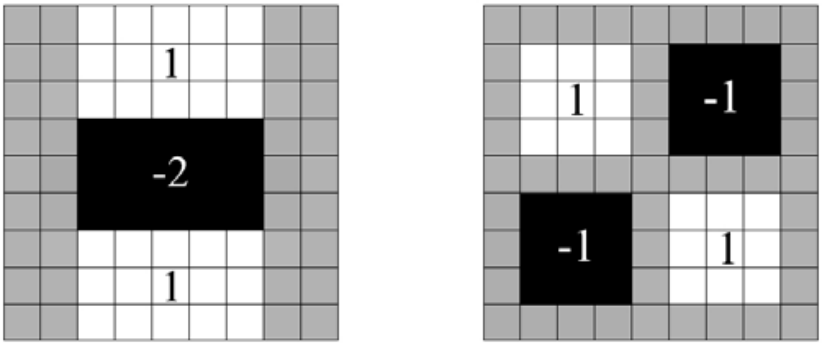


图 3.3 *y* 和 *xy* 方向经过盒子滤波后的近似值

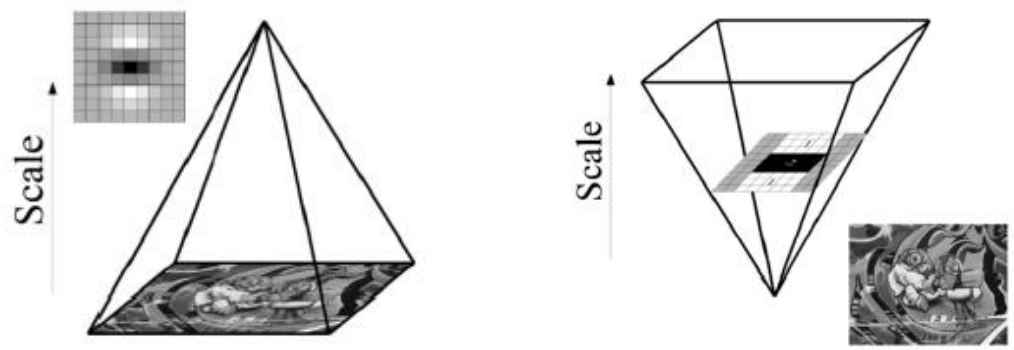
通常通过构造图像金字塔来构造尺度空间。在金字塔中原图像作为最底层，然后通过不断对图像进行高斯平滑再采样得到更高的金字塔层。高斯金字塔中高斯模板尺寸不变，原图像尺寸不断变化（如图 3.4(a)），而且每一层的建立必须建立在底层的构造完毕的基础上，这使得构造速度很慢，而且对底层的图片依赖性较强。因为盒子滤波和积分图像的使用，SURF 金字塔

（如图 3.4(b)）每层的构建不必再对上一层使用滤波，可以同样的速度下对原图像使用任意大小的滤波，甚至可以并行计算。所以不同于高斯金字塔通过缩放原图像构造尺度空间，SURF 是通过改变滤波大小来构造尺度空间的，这样使得每一层金字塔的构造都是独立的不仅可以降低计算对上层的依赖性，还可以通过并行计算减少计算时间。在 SURF 算法中，Bay 等人用9×9滤波器作为初始滤波器，记尺度大小$s=1.2$（近似等于$\sigma=1.2$）。由于积分图像的离散性，所以金字塔层之间最小变化量是由l_0决定，它是在微分上高斯二阶微分器对正负斑点的响应长度。l_0等于盒子滤波模板尺寸的$\frac{1}{3}$。为了保证滤波器的结构比例不变（即保证一个中心像元的存在），下一层的响应长度至少应该在l_0的基础上增加 2 个像元，所以$l_0=5$，模板尺寸为15×15。由此可以推出模板序列为9×9，15×15，21×21，27×27等。其模板尺度可以由公式(3-5)计算的出。

$$FilterSize=3\times\left(2^{octave}\times interval+1\right) \tag{3-5}$$

其中，octave 表示组数，interval 表示层数。在第 0 组第 0 层时，octave=1，interval=1。

SURF 中所用滤波模板的比例仍与尺度变化保持一致，近似于高斯滤波器的参数的变化。比如，当盒子滤波模板为27×27，其对应的高斯参数为$\sigma=3\times1.2=3.6=s$。另外，因为盒子滤波中 Frobenius 范数保持不变，所以滤波器的比例总是尺度标准化的[57]。



(a) 高斯金字塔　　(b) SURF 金字塔

图 3.4 金字塔模型

为了确定兴趣点在不同尺度图像上的位置，Bay 等人采用了$3\times3\times3$邻域的非最大值抑制法。首先要将所有小于所设定的 Hessian 矩阵行列式的阈值的特征点丢弃；然后对其进行检测，如图 2.5 所示。图中标记为'X'为待检测的点的像元，将其与上下两尺度层各 9 个点的值和自身尺度层剩余 8 个点（共$9+9+8=26$点）的值进行比较，若该点的值大于这 26 个点的值，则被选为候选特征点，否则被丢弃。

在确定候选特征点之后，要对特征点进行定位。因为以上的极值点搜索是在离散空间进行

的，不能算作真正意义上的极值点，所以需要对空间尺度进行插值从而得到亚像素级特征点的坐标。SURF 中选用 Brown 等人[58]提出的方法在图像的尺度空间对 Hessian 矩阵的行列式的最大值进行插值。尺度空间的泰勒展开式如下：

$$H(x)=H+\frac{\partial H^{T}}{\partial x}x+\frac{1}{2}x^{T}\frac{\partial^{2}H}{\partial x^{2}}x \tag{3-6}$$

对 $H(x)$ 进行求导，并使其等于 0 即 $\frac{dH}{dx}=0$，从而得到亚像素级。

$$\hat{x}=-\frac{\partial^{2}H^{-1}}{\partial x^{2}}\frac{\partial H}{\partial x} \tag{3-7}$$

其中：

$$\frac{\partial^{2}H}{\partial x^{2}}=\begin{bmatrix}d_{xx} & d_{yx} & d_{sx}\\ d_{xy} & d_{yy} & d_{sy}\\ d_{xs} & d_{ys} & d_{ss}\end{bmatrix}\cdot\frac{\partial H}{\partial x}=\begin{bmatrix}d_{x}\\ d_{y}\\ d_{s}\end{bmatrix} \tag{3-8}$$

所以求得 $\hat{x}=(x,y,\sigma)$，为三个方向的偏移量。当偏移量大于 0.5，则说明偏离原像素点，然后继续进行插值计算。因为尺度空间中每一组的第一层之间的差值相对较大，所以差值计算是非常重要的。

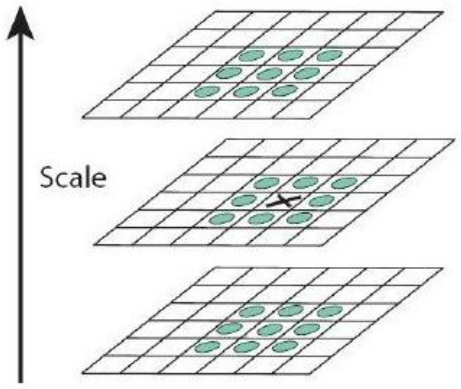


图 3.5 3×3×3邻域非最大值检测示意图

3.2 基于栅格的 SURF 检测

一个比较好的特征点检测方法，应该具有以下特点：定位精确（在位置和尺度上）、重复性（在下一张图像上也能检测出大量特征点）、计算效率、鲁棒性（对于噪音、压缩和模糊）、特殊性（可以同一特征点在不同的图片上仍可以被检测出来）、不变性（对于光度（例如照明）和几何变化（旋转、尺度、远近景））。另外，对于 SLAM 算法，特征点的分布对其结果也具有显著的影响，特征点越多且分布的越广，摄像机位姿估计的结果越好[59]。所以为了可以让特征点

更均匀的分布在图像上，可以将图片栅格化，然后检测每个子栅格内的特征点。

SURF 检测方法是斑点检测中经典的检测方法之一，不仅具有斑点检测的优点—具有更好的辨识度、稳定性好抗噪能力强，而且在旋转和视角变化比较大的两张图片上检测出的特征点重复率高。但是斑点代表一个区域对于角点比较丰富的环境会忽略这些角点，导致某些环境下提取的特征点过少，从而影响最后的运动估计。所以为了克服以上的问题，本章提出了基于栅格的 SURF 检测方法。方法流程图如图 2.7 所示。

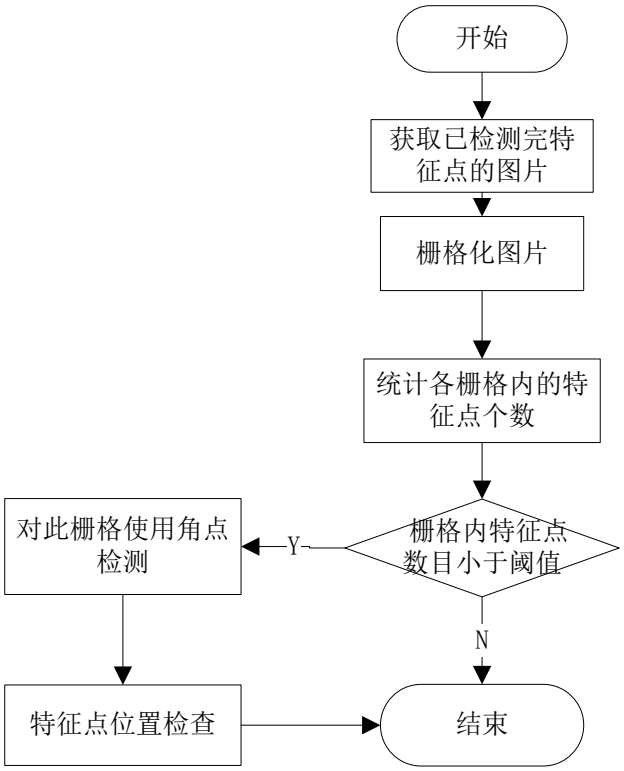


图 3.6 算法流程图

获取一张已用 SURF 检测方法检测后的图片，将图片均匀地分割成一定大小的子区域，这些子区域被称之为栅格(Grid)。将图片划分为宽 M 个和高 N 个，共 $M \times N$ 个子栅格，每个栅格标记为 g_{ij}，其中 i 和 j 表示为第 i 行和第 j 列。对这些栅格进行特征点个数的统计，当特征点个数小于阈值，则采用栅格选取，而阈值 r_t 的计算方法为：

$$r_t = \frac{\sum_{i=1}^{M \times N} f_i}{M \times N} \tag{3-9}$$

其中 f_i 为各个栅格内特征点分布的个数，$M \times N$ 为总的栅格个数。其中阈值的选择为算术平均数，主要是考虑到该算法的不需要精确了解到特征点的分布情况，而且算术平均数计算简单，降低了算法的复杂性。

为了避免提取的采样点出现重叠，用角点检测前对已经存在的采样点进行矩形参数间隔，

判断采用角点采样的特征点是否在矩形参数内，如果不在则保留。

3.3 实验结果及对比

对于栅格的选取，如果栅格数过多就会产生更多的边缘，而在检测特征点时一般会剔除边缘附近的特征点，所以过多的栅格可能会导致大量有效的特征点丢失。如果栅格数较少，则会导致重复检测，算法可行性较差。所以通过实验，本文采用 3×4 来对图片划分栅格。为了能提取更多的特征点的同时，尽量减少运行时间，本文选用对于特征点个数不满足阈值的栅格图片检测速度较为快的 FAST 算法进行检测。对于 SURF 算法采样出的点设置 $(x-r,y+r)$，$(x-r,y-r)$，$(x+r,y-r)$ 和 $(x+r,y+r)$ 四个顶点构成的矩形作为间隔矩形（其中 r 的值为每个检测点的不确定度的半径）。使用 FAST 算法采样时，在这个范围外的点保留，实验结果如下。

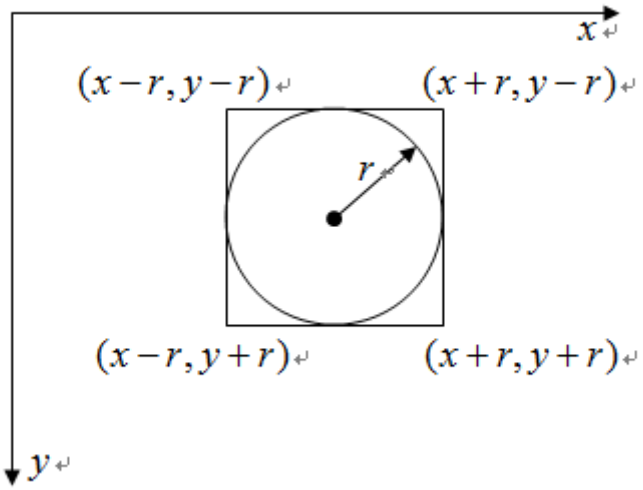


图 3.7 间隔矩形

3.3.1 检测结果比较

使用的图像数据是引用 Mikolajczyk 和 Schmid[60]中的，包括高斯模糊度变化的图像集、光照变化的图像集、尺度和旋转变化图像集、分辨率变化的图像集以及视角变化的图像集。所有的算法都是在 Window10 的 VS2010+Opencv2.4.9 上运行的。

(a) 原图 1　　(b) 原图 2

图 3.8 光线变化的两个原图

表 3.1 SURF 检测算法与本章算法的对比

算法	检测时间/s	特征点个数(原图 A/B)	匹配个数	正确匹配个数
SURF 检测算法	0.52	402/213	202	91
本章检测算法	0.57	513/255	308	122

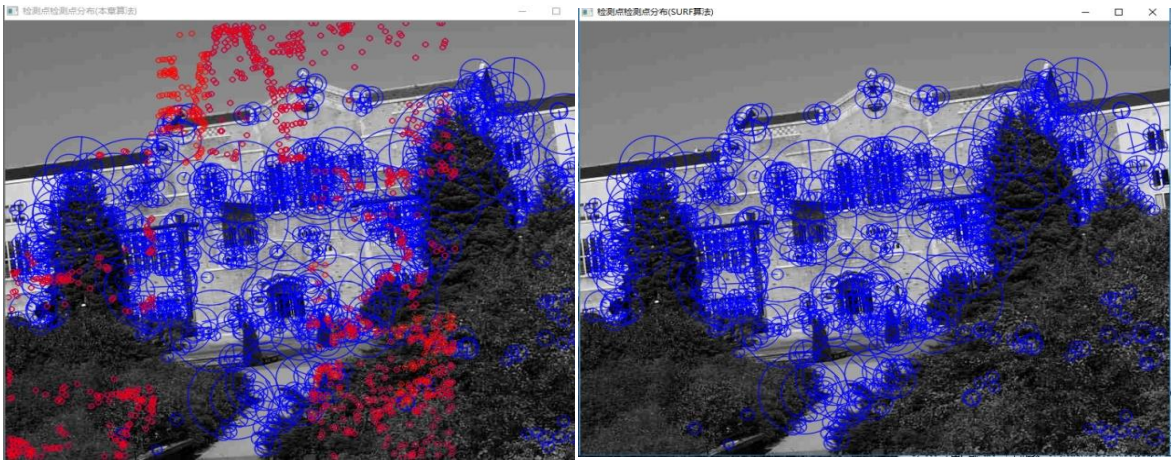
表 3.1 中对图 3.8 中两幅图进行检测并进行了特征点匹配。图 3.8 中两幅图虽然在室外，但遮挡物比较多，光线变化较大，灰度值变化层次多，有较多角点。从表 3.1 可以看出，SURF 检测的特征点个数数量较少，导致正确匹配个数很少，这样对于最终位姿估计的造成很大的误差。而本章算法虽然检测时间有所增加，但是特征点检测个数明显增多，即使在图 3.8(b)这种灰度变化不太明显的场景中也表现出较好的检测效果。下面以图 3.8(a)作为特征点分布情况的原图。

图 3.9 SURF 检测特征点分布情况

图 3.10 本章算法中特征点分布情况

图 3.9 和图 3.10 分别展示的是 SURF 检测和本章检测算法在同一幅图片上检测点的分布情况。从图 3.9 可以看出，SURF 检测的特征点中，在大范围灰度较平滑的区域内检测点数较少（如图片中间和下方区域），特征点分布比较集中，图片中几乎有一半区域没有特征点分布。图 3.10 中蓝色代表 SURF 检测算法检测到特征点，红色代表本章算法检测到的新增的特征点，从图中

可看出，本章算法虽然在某些区域的特征点数依然较少，但整个图片中特征点覆盖面积增大，分布较广，特别是中间几乎没有明显特征的区域也可以检测出较多的特征点。

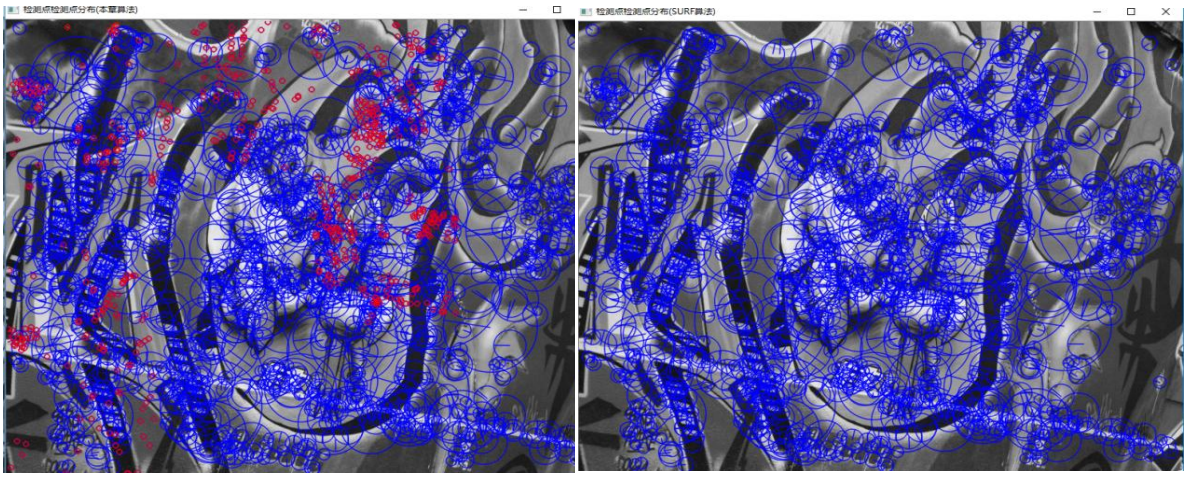


(a) 本章算法　　(b) SURF 检测

图 3.11　分辨率变化下的检测

(a) 本章算法　　(b) SURF 检测

图 3.12　高斯模糊变化下的检测

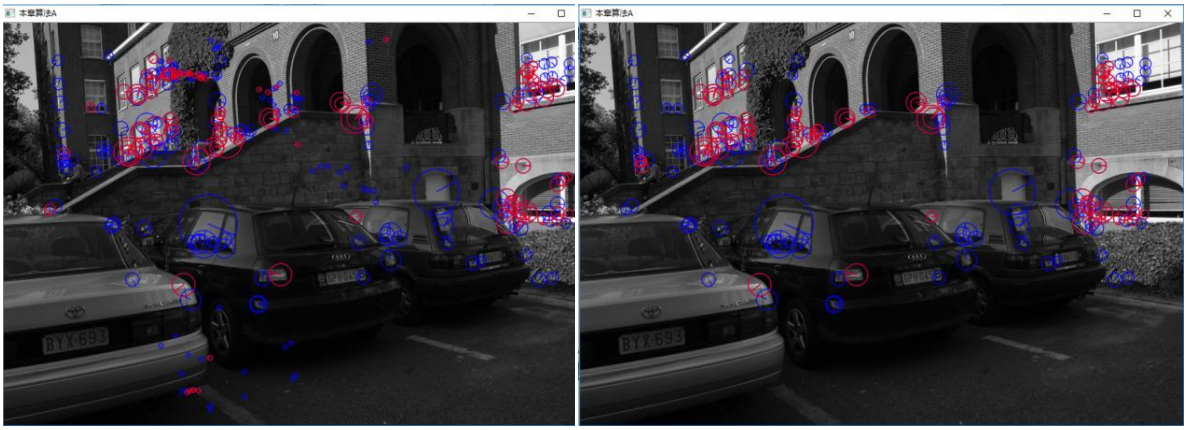


(a) 本章算法　　(b) SURF 检测

图 3.13 视角变化下的检测

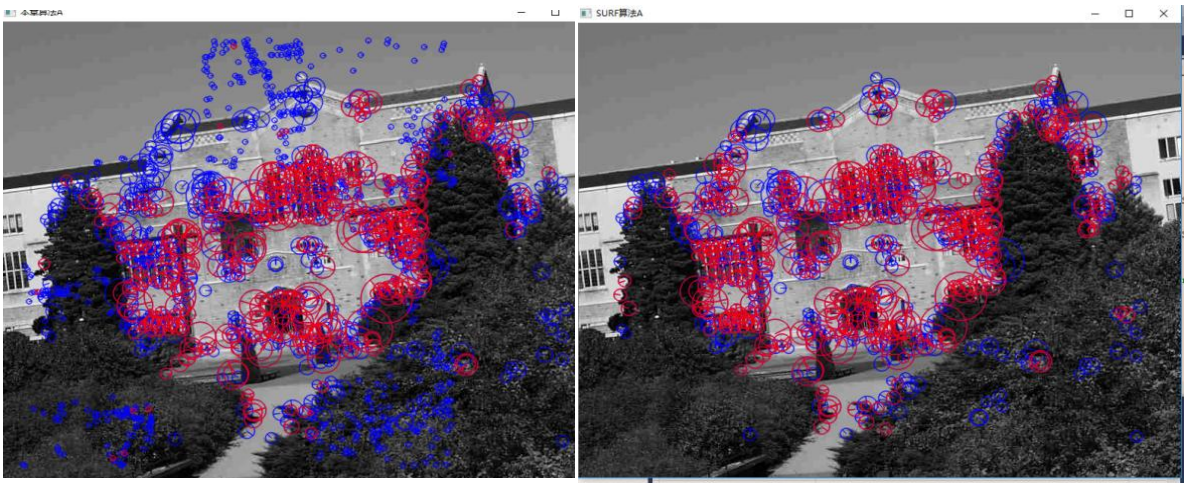
从图 3.11 和图 3.12 可以看出该算法对于分辨率有变化的图像有较好的鲁棒性，对于 SURF 检测算法中无法检测到的区域可以有所补充，且对于 SURF 检测较好的场景（如图 3.13）也可以进行有效的检测，并尽量使特征点分布到场景中的每个区域，这样对于场景中有较大动态物体的场景可以尽量检测到没被遮挡到的特征点，对后面的位姿估计有很好的保障。

3.3.2 加入匹配结果的比较



(a) 本章算法　　(b) SURF 算法

图 3.14 光线变化检测的特征点和内点

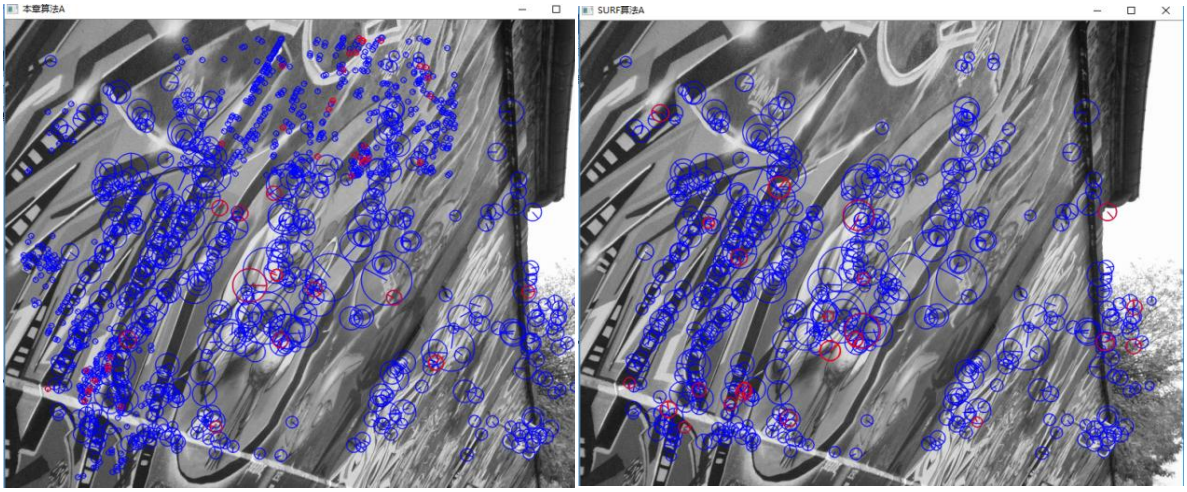


(a) 本章算法　　(b) SURF 算法

图 3.15 分辨率变化检测的特征点和内点

(a) 本章算法　　(b) SURF 算法

图 3.16 高斯模糊变化检测的特征点和内点



(a) 本章算法　　(b) SURF 算法

图 3.17 视角变化检测的特征点和内点

从图 3.14 到 3.17 中红色表示正确的匹配点数，蓝色表示错误的匹配点数。从中可以看出，本章算法中正确匹配点数分布更加均匀，且个数比较多，这对于后面的位姿运算提供了好的基础。

3.4 本章小结

本章对 SURF 检测算法进行详细的描述，并阐述了 SURF 检测算法具有的优缺点，为了弥补其不足。提出加入角点检测的基于栅格的 SURF 算法，主要对图片划分栅格，对特征点数量较少的区域用 FAST 角点进行重新检测。最后实验得出，本章算法可以弥补 SURF 检测算法对于场景中对于角点的不敏感，尽量增加特征点分布的广度，使得特征点可以覆盖到场景中大部分区域。

第四章 基于简化的 FREAK 模型的特征点匹配算法

图像匹配是基于特征的视觉 SLAM 方法中最重要的一步，是摄像机是否可以准确定位和构建地图的关键。视觉 SLAM 中的图像匹配问题只要是将摄像机采集的视频中相邻两帧的图像在空间上进行“对准”。因为摄像机的运动，图像之间存在着空间变换，如旋转变换、尺度变换等，而且在移动过程中，图像中的光线以及运动物体也在不停变换，所以要想得到精准的匹配结果要把这些影响因素考虑进去。本章主要对基于特征的匹配算法进行讨论。基于特征的匹配算法过程最关键就是对特征点进行描述，一个好的描述子不仅可以减小计算时间，还可以提高匹配精度。因为计算机技术的发展，计算机计算能力和存储能力的提高使得建立二进制矩阵成为可能，所以用二进制矩阵对特征点描述的方法逐渐成为主流，这种方法相较于以往的算法（SIFT 和 SURF）计算速度更快，而且匹配方式更加简单。本章从主流的二进制匹配算法中选取各种光线变化场景中匹配点数量比较稳定的算法[61]，FREAK 算法。为了尽量在不影响实时性的情况下，提高 FREAK 算法在光线变化较大和图片较模糊的场景的正确匹配率，选用两种方法同时进行改进。

4.1 FREAK 算法

现如今比较经典的二进制匹配算法有 BRIEF 算法、ORB 算法、BRISK 算法以及 FREAK 算法。BRIEF 算法是较早提出的一种二进制算法，它主要通过对已经进行高斯平滑的 512 个采样对进行特征点描述，然后用汉明距离进行匹配。虽然 BRIEF 算法运算速度很快，但是它不具有旋转不变性和尺度不变性，且对噪音比较敏感。为了解决 BRIEF 的不足，Rublee 等提出了 ORB 算法。ORB 算法在特征点提取方面仍采用 FAST 算法，但在检测上提出计算特征点的主方向来解决 BRIFE 的旋转问题。描述方法是在 BRIEF 算法基础上加入尺度空间来解决尺度问题，以及在解决噪音方面采用积分图像和在旋转问题方面对随机对进行旋转判别对。同一时间，Leuteneggar 等提出了 BRISK 算法（与 An Efficient Dense Descriptor Applied to Wide Baseline Stereo:DAISY[62]算法相似），不同于上面两种算法它是使用同心圆图案来进行特征点采样的。为了建立二进制描述矩阵，BRISK 采用同心圆中的采样点，并对这些采样点进行短距离和长距离划分。长距离子集被用来估计方向，而短距离子集则用来在采样图像旋转时进行二进制描述。虽然 FREAK 算法与 BRISK 算法同样是采用同心圆图案，但是 FREAK 算法的同心圆是基于人类的视网膜模型，圆与圆之间存在重叠区域。另外，在方向计算上 FREAK 算法抛弃了 BRISK 算法中的长距离子集计算方法，改用了固定的中心对称对进行计算，将几百个点对减少为 45 个点对，大大减少了算法的计算量。下面对 FREAK 算法进行详尽介绍。

4.1.1 采样模型

FREAK 算法采用类似人眼的视网膜接收图像信息的采样模型，此模型一共有八层，每层有六个半径相同的圆圈即感受野，圆心是采样点，但是每层感受野的半径是不同的，大小等于每个感受野上相对应用于图像平滑的高斯核的标准差。所以越靠近内层的特征点，点的密度就越大（如图 4.1）。

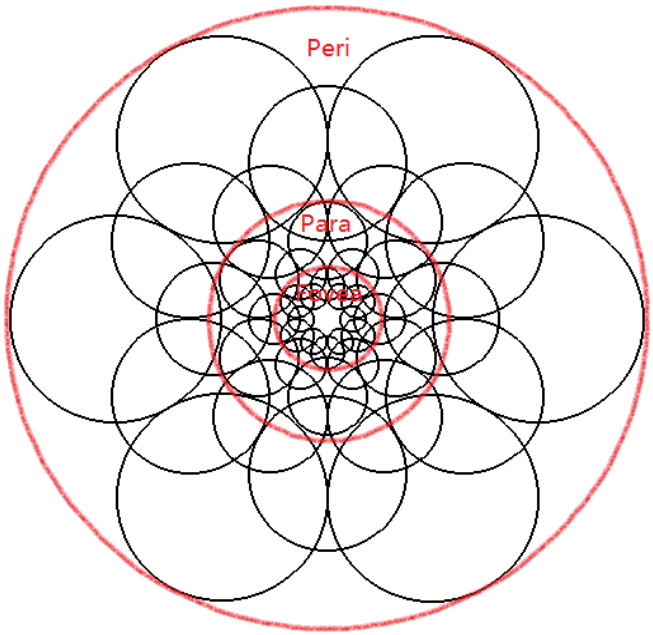


图 4.1 采样模型

Alexandre 等人在实验中，发现将高斯核尺寸引入到对极数视网膜图案中性能会更好，另外感受野的重叠可以捕捉更多的信息，从而可以提高算法的性能。算法中感受野 A、B、C 的像素值 I_i 符合以下规则：

$$I_A > I_B, I_B > I_C, and, I_A > I_C \tag{4.1}$$

如果这些感受野不重叠，那么不能判别 $I_A > I_C$，而且感受野的重叠使得部分新的信息可以被采样。总的来说，增加冗余可以减少感受野的使用，这也是一种用于压缩传感或字典学习的一种策略[63]。另外 Olshausen 和 Field[64]也提出在视网膜的感受野是存在冗余的。

4.1.2 由粗到细的描述方法

通过比较对应高斯核的感受野对建立二进制描述子 F，它是由一位高斯差分（Difference of Gaussians）数组构成的：

$$F = \sum_{0 \le a \le N} 2^a T(P_a) \tag{4.2}$$

其中， P_a 是感受野对， N 是描述子预计的大小。以及：

$$T(P_a) = \begin{cases} 1 & if\ (I(P_a^{r_1}) - I(P_a^{r_2}) > 0), \\ 0 & otherwise. \end{cases} \tag{4.3}$$

$I(P_a^{r_1})$ 是感受野对中第一个平滑的像素值。

感受野越多，构成的二进制描述子就越大。但是对于描述一幅图片，许多点对的描述很可能是无用的，所以为了选择有效的点对，BRISK 利用空间距离进行筛选。但是这种方法选出的点对具有高相关性以及不具有辨识度。FREAK 使用了一种与 ORB 相似的方法从训练数据中选择更好的点对。

1. 首先构造一个有近 5 万被提取的特征点的矩阵 D，每一行对应一个特征点的描述符。每一个表示特征点的描述子又由视网膜模型中的采样点对的计算所得。所以矩阵有 $43 \times 42 / 2 = 903$ 列。

2. 计算矩阵 D 中每列的平均值。为了得到一个具有辨识度的特征点，期望每列的值具有高方差，而均值为 0.5 说明这组二进制数具有最高的方差。

3. 选择矩阵 D 中符合最高的方差的列数。

4. 存储最优的列数，然后在剩下的列数中选取与已选的列具有低相关性的列。

在实验中发现前 512 对是最有效的，增加剩余感受野对不能提高算法的性能。所以选取前 512 列，然后将这 512 个感受野对分为 4 组，每组 218 个野对，对每组野对进行连线观察。发现这样描述方式跟人类视觉系统很相似，都是通过 perfoveal 区域对物体的位置进行估计，然后通过 fovea 区域进行验证，从而得到物体的位置。

4.1.3 扫视搜索

人类看一个场景不会通过固定视角来看，一般都是眼睛单独不连续的移动来观察环境，这种行为叫做扫视。Fovea 区域可以具有的高密度的感光细胞让其可以捕捉到更高分辨率的信息，所以在识别和匹配中有着关键性的作用。而 perifoveal 层不能捕捉细节的信息，如低频观测值，所以它用于在第一步的物体位置的估计上。

通过对描述子的分解来模拟扫视搜素。首先用 FREAK 描述子的前 16 个字节作为扫视信息，如果距离小于一个阈值，更进一步对剩下的位数进行比较来获得更好的信息。结果发现，这种级联加快了匹配速度。在 FREAK 描述子的前 16 个字节中，超过 80%的候选点被抛弃。值得注意的是，选择 16 字节作为第一级联是为了满足硬件需求。

4.1.4 方向

为了估计特征的旋转，FREAK 算法中概述了与 BRISK 算法相似的方法，即对选择的感受野对进行局部梯度估计的方法。然而，不同于 BRISK 使用长距离感受野对来计算总体的方位，FREAK 主要选择的是中心对称的感受野对（如图 4.3）。

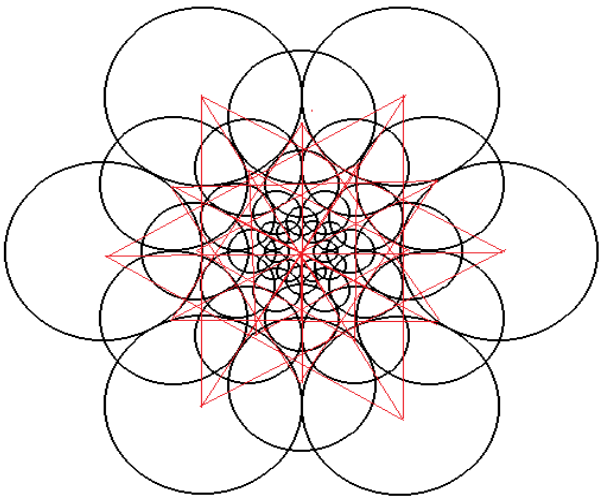


图 4.2 计算方向的点对

设G是计算方向梯度的所用感受野对的子集：

$$O=\frac{1}{M}\sum_{P_o\in G}(I(P_o^{r_1})-I(P_o^{r_2}))\frac{P_o^{r_1}-P_o^{r_2}}{\|P_o^{r_1}-P_o^{r_2}\|} \tag{4.4}$$

其中，M 是在G中感受野对的数量，$P_o^{r_i}$是感受野的中心点的空间二维坐标。

FREAK 算法中选用 45 个感受野对替代了 BRISK 中几百个感受野对。另外，在 perifoveal 区域中的感受野半径比 BRISK 中的大，可以允许在方向估计上有更多的误差，因此可以将方向空间离散化，从而使内存负载减少了 5 倍以上（从 40MB 减少为 7MB）。

4.2 基于简化 FREAK 采样模型的改进算法

FREAK 算法属于二进制匹配算法，匹配速度较快，但是对于场景发生大变化的情况下鲁棒性较差。所以国内产生了许多对 FREAK 算法进行改进的方法。例如谢红等人[65]提出了将具有尺度不变的 SURF 中的检测算法与 FREAK 算法结合的新的图像匹配算法(SURF-FREAK)，该算法对图像的尺度、旋转和光照差异上有一定的提高，但实时性有所降低。针对 FREAK 的描述算法，李晶皎等人提出了 Improved FREAK (IFREAK)[66]算法，该算法通过采样多个位来增

加描述子从而改善匹配效果的，但是该算法因为增加了计算的复杂度也因此增加了描述时间。为了减少计算的复杂度，提高算法的实时性，Jianyong Wang 等人提出了 Center Symmetry FREAK(CS-FREAK)[67]算法，该算法中将八层视网膜模型简化成五层视网膜模型，从而减少其描述时间，但其匹配精度有所降低，所以作者又在匹配方法上加入中心对称，使得算法实时性提高，在各方面的鲁棒性也有所提升。FREAK 的匹配算法与大部分的二进制匹配算法相同，采用的是汉明算法，所以改进汉明距离的精度是主流的改进方式。房贻广等人[68]提出对汉明距离进行加权，改进原来汉明距离粗略的计算，使计算更加精确，使得算法更适用于旋转尺度变化较大的环境。

虽然 FREAK 算法在各种光线变化的场景，正确匹配点的数量变化较小，但是匹配正确率不高。为了尽量保证实时性的情况下，提高算法在光照变化的场景匹配率，本章对 FREAK 算法进行了改进，改进方法可以分为两个部分，首先对在相同光照变化下和分辨率变化下对不同层数的视网膜模型进行分析，采用最合适的简化模型，即将 FREAK 算法中的八层视网膜模型简化成五层。然后对相对距离大于一定阈值的采样点之间进行线性插值，感受野对和插值点间按顺序进行像素值比较，从而将描述子从一位变为四位，从而增加其精度。

4.2.1 简化的视网膜采样模型

通过图 4.3 可得，采样模型的层数的变化对光线变化和分辨率变化下的算法鲁棒性影响较小，所以综合匹配的正确率和时间（表 4.1），本章中将 FREAK 算法中的 8 层采样模型简化成 5 层（如文献[68]），43 个采样点减少为 25 个，每层与原 FREAK 算法相似，半径大小等于每个感受野上相对应用于图像平滑的高斯核的标准差，简化模型如图 4.4 所示。

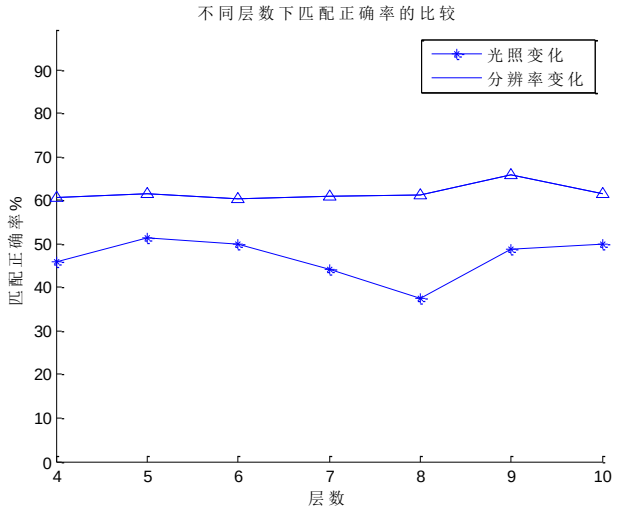


图 4.3 不同层数下匹配正确率的比较

表 4.1 不同层数的模型的描述时间

层数	4	5	6	7	8	9	10
描述时间/s	0.135	0.156	0.164	0.176	0.188	0.204	0.212

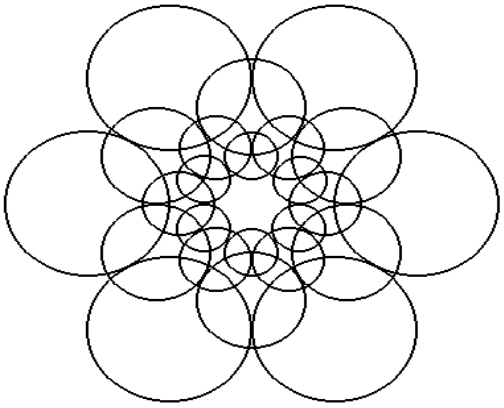


图 4.4 简化的采样模型

从简化模型可以看出，虽然靠近特征点的感受野密度减少，但冗余仍然存在，对于感受野 A、B、C 依然满足公式(4.1)。因为图片的像素远远达不到人眼所呈现的画面，所以八层视网膜采样模型中可能存在距离过小的感受野之间的比较，这些比较不存在意义，造成存储空间的浪费和计算成本的增加，所以简化层数对于光线变化和分辨率变化下的匹配结果影响较小。但因为考虑到方向梯度计算，所以折中选择 5 层的视网膜模型。

4.2.2 加入线性插值的描述子

线性插值是数学、计算机图形学等领域广泛使用的一种简单的插值方法，通过使用线性插值可以求出不在表上的值。从 FREAK 原文中了解到，冗余的存在可以加强光感，所以为了增加采样模型中相隔距离较大的感受野，增加其冗余，对描述子进行线性插值。线性插值对于像素值变化较大的两个感受野的像素描述更加详细，这样对于图片中光线变化更加敏感，有利于增加在光线变化较大和分辨率较低环境中特征点的匹配率。为了不增加不必要的冗余，本文中只对模型中距离较大两个感受野进行线性插值，然后依次进行比较，这是一种考虑了空间结构的描述符，方法如图 4.5 所示。

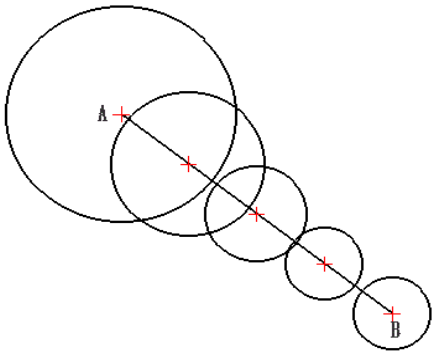


图 4.5 感受野中心 A 和 B 之间的位取样

本文描述子分成两个部分，对于 $dx^2+dy^2>=threshold$ 的情况将两个采样点之间均分 S（$S>1$）份，从 A 点到 B 点依次进行采样比较，让每对采样点间的描述值从一位增加到 S 位，其它情况则不进行线性插值。所以算法描述符 F' 的公式如公式 1 所示。

所以改进算法的描述子为：

$$F'=\begin{cases}\sum\limits_{0\le a\le N}\left(2^4\right)^a P_{A,B}^a & dx^2+dy^2>=threshold\\ \sum\limits_{0\le a\le N}P_{A,B}^a 2^a & otherwise\end{cases} \tag{4.5}$$

其中，dx^2+dy^2 是两个感受野中的距离，而 $P_{A,B}^a$ 的值为：

$$P_{A,B}=\begin{cases}\sum\limits_{i=1}^{S}2^{i-1}b_i & dx^2+dy^2\ge threshold\\ b_i & otherwise\end{cases} \tag{4.6}$$

$$b_i=\begin{cases}1 & I\left(v_i\right)>I\left(v_{i+1}\right)\\ 0 & otherwise\end{cases} \tag{4.7}$$

在本文中选取 S=4，阈值 *threshold* 选择原 FREAK 算法中 fovea 的范围内最大圆的直径的平方。$I(x)$ 表示某一图像区域 X 的灰度平均强度。每个位 b_i 表示一个当前点 v_i 和 v_{i+1} 的比较。对所有对 $(A,B)\in P$，其中 P 是选出来的对的有序集，结果位向量描述符是 $P_{A,B}$ 的级联。

4.2.3 方向计算

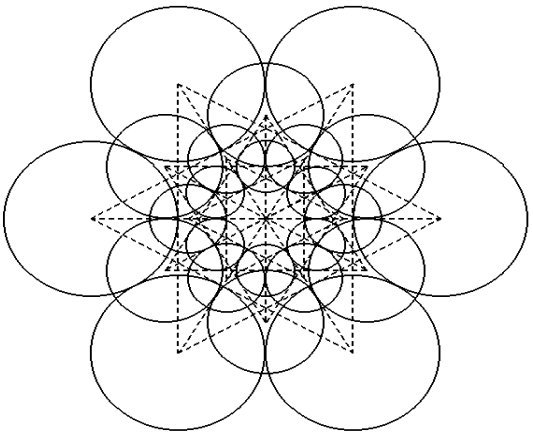


图 4.6 计算所取对的方向说明

本文方向计算与原 FREAK 算法相同，因为原有的 45 个采样点减少为 30 个（见图 4.6），主要是选择相对于中心对称的采样点对，其计算公式如下所示。

$$O=\frac{1}{M}\sum_{P_o\in G}\left(I\left(P_o^{r1}\right)-I\left(P_o^{r2}\right)\right)\frac{P_o^{r1}-P_o^{r2}}{||P_o^{r1}-P_o^{r2}||} \tag{4.8}$$

其中，其中，M 是 2D 矩阵 G 和 P_o^{ri} 点对的数量。

4.3 实验结果及分析

为了测试本文算法的性能，与 FREAK、CS-FREAK 和 IFREAK 算法进行了对比。使用的图像数据是仍然引用 Mikolajczyk 和 Schmid 中的，包括高斯模糊度变化的图像集、光照变化的图像集、尺度和旋转变化图像集、分辨率变化的图像集以及视角变化的图像集。且其检测算法采用第三章所提出的基于栅格的改进算法。

为了对匹配的特征点进行错误点剔除，从而来估算匹配的正确率本文使用了 RANSAC 算法。RANSAC 最早是由 Fischler 和 Bolles 提出，主要用于根据一组包含异常数据的样本数据集，计算出数据的数学模型参数，得到有效样本数据的算法。如今，RANSAC 算法被经常用于计算机视觉中。

4.3.1 实时性对比

表 4.2 四种算法所用时间

算法	FREAK	CS-FREAK	IFREAK	本文算法
检测时间/s	0.6273	0.6186	0.6470	0.6236
描述时间/s	0.0777	0.0437	1.5978	0.5473
匹配时间/s	0.0287	0.0199	0.0736	0.0365

通过表 4.2 的时间对比，发现本文算法虽然仍没有原始算法的用时短，但相较于 IFREAK 算法时间缩短近三倍，实时性有所提高。但因将两个采样点的描述符将一位变为四位，在描述矩阵的建立上增加了运算量，虽然简化了采样模型，已将描述时间有所缩短，但仍不能缩短至原有的时间。

4.3.2 图像集中各图像匹配点的正确率比较

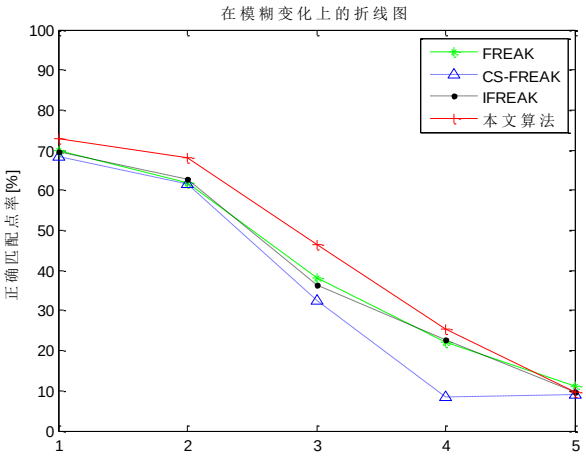


图 4.7 在高斯模糊变化上的折线图

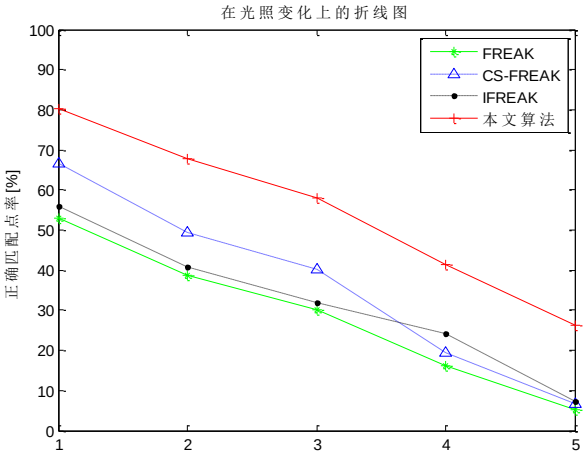


图 4. 8 在光照变化上的折线图

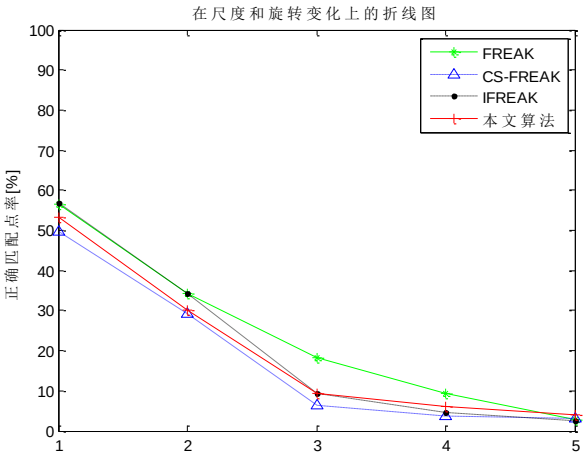


图 4.9 在尺度和旋转变化上的折线图

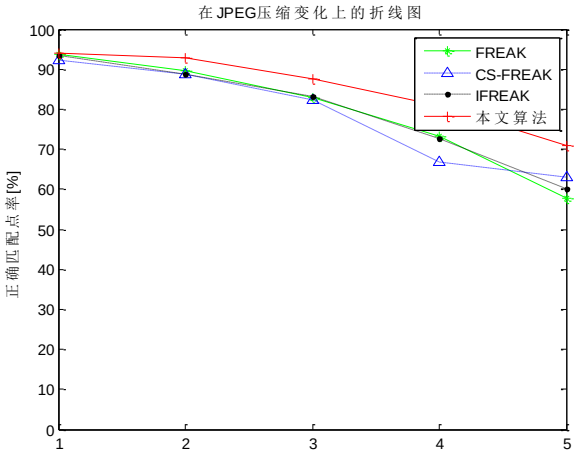


图 4.10 在 JPEG 压缩变化上的折线图

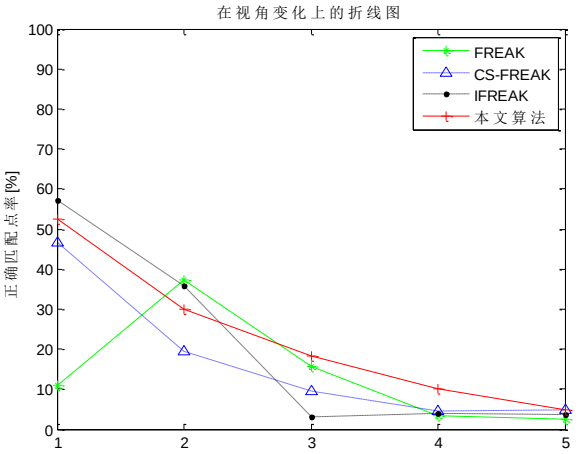


图 4.11 在视角变化上的折线图

由图 4.7 至图 4.11 可知，本文算法在模糊变化、光照变化和 JPEG 压缩变化上有明显的匹配优势，特别是在光照情况下正确率提高近 30%。但是在视角变化和旋转尺度变化上性能却没有明显的改善，甚至有些变差。这是因为简化了视网膜模型，计算方向变化梯度的采样点对减少，可能会导致方向计算结果与实际结果偏差较大。而对于视角变化，采用线性插值增加描述子的描述精度无法准确的描述透视投影改变的像素。

4.3.3 在各种变化图像匹配点的正确配率对比

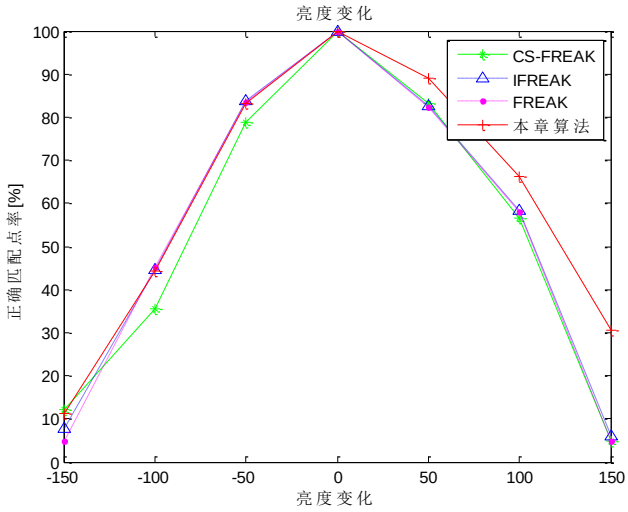


图 4.12 图像亮度变化下的正确匹配率比较

从图 4.7 可以看出，在亮度变化上，本文算法具有较好的鲁棒性，相较于其它三种算法在曝光过度的情况下也可以具有较高的匹配率，而对于较为暗的环境中也可以进行良好的匹配，比较适合室内这种光线变化较为多的环境。

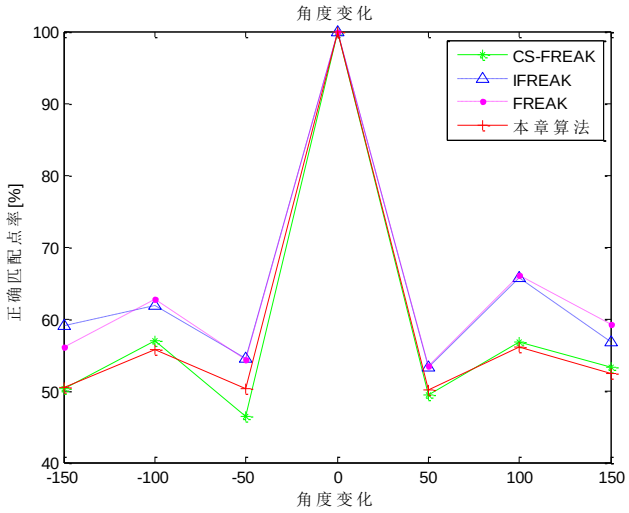


图 4.13 角度变化下正确匹配率的比较

从图 4.8 可以看出，虽然在角度变化的情况下本章算法不能达到原有算法的匹配效果，但平均匹配的正确率依然可以达到 50%左右，可以满足室内机器人的运动估计需求。

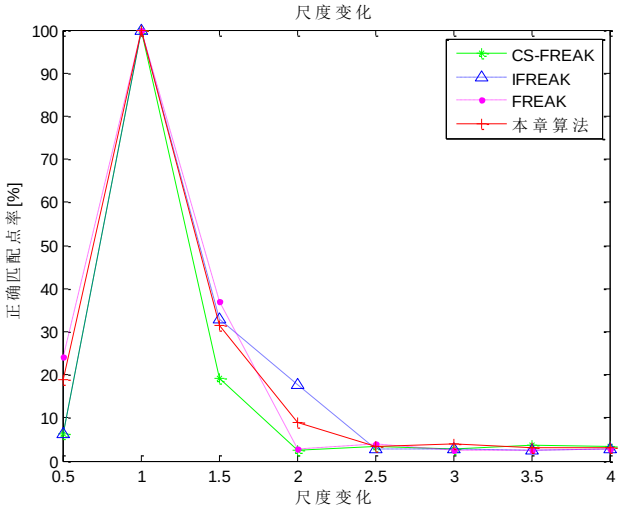


图 4.14 尺度变化下正确匹配率的比较

图 4.9 展示了四种算法在尺度变化下的正确率，可以看出，当图像尺度变化过大时，四种算法都没法得到很好的匹配效果，但是本章算法和 IFREAK 算法在尺度较大时匹配率相对于其它算法有所提高，特别是在大于 1.5 和小于 2.5 之间，对于一般工作环境来讲，可以满足计算需求。

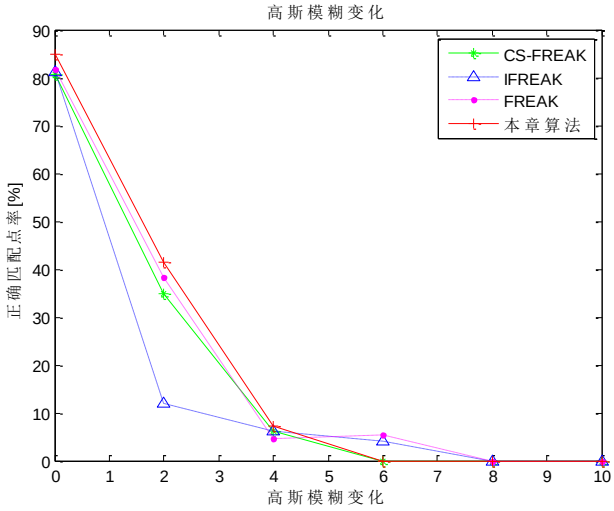


图 4.15 在高斯模糊变化下正确匹配率的比较

由图 4.10 可知，本章算法在高斯模糊变化下的正确匹配率对于原有算法有小幅的提升，对于摄像头发生震荡或者有移动物体在场景中时导致的图像的分辨率下降等有较好的鲁棒性。

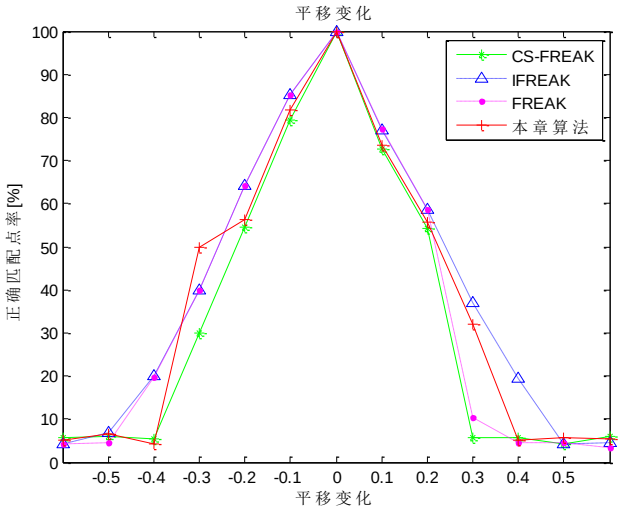


图 4.16 在平移过程中的正确率比较

平移是摄像机运动过程中，最普遍的行为，所以在平移过程中算法应该具备相较于其它变化下的更好的鲁棒性。图 4.11 中是对图片进行 $(x \pm rows \bullet k, y \pm cols \bullet k)$ 平移得出的结果，其中 k 的值为图中横坐标的值。由图可知，算法对于图片平移较大的情况下具有良好的鲁棒性。

4.4 本章小结

本章对原 FREAK 算法进行采样模型的简化，并在描述过程中对大于一定阈值的两感受野之间进行线性插值，增加冗余性。实验证明，本章算法虽然在描述过程中增加了计算时间，但除了旋转变化的情况中，表现出相较于原有算法更好的鲁棒性，特别是在亮度变化和分辨率变化下，对于室内场景的特征点匹配具有一定的优势。

第五章 基于 EKF 算法的 1 点 RANSAC 算法的改进

RANSAC 从 1981 年被 Fischler 和 Bolles 提出后，被运用在很多方向的研究上，而且一直沿用至今。早在 1993 年，Torr 和 Murry 提出将 RANSAC 算法应用到视觉运动估计上[69]，这也是如今在视觉领域上比较常用的筛选特征点的算法。近十年，研究的主要方向集中在通过预先检验和放弃较差的假设模型来减少标准 RANSAC 算法中验证模型的计算量[70],[71]。1-point RANSAC[72]就是基于这个理念提出的具有良好效果的算法。相较于标准的 RANSAC 算法，1-point RANSAC 假定预先知道模型参数的先验概率分布，这使得算法可以使用一个数据点就可以估计模型，大大减少了假设的数量和提高了计算效率。但是 1-point RANSAC 的缺点也很明显，算法对摄像机的运动进行约束了，任意一个超出约束的运动都会造成估计误差。虽然约束模型对于 RANSAC 假设是足够了，但是约束小的模型可以为运动估计提供更好的结果。因此，Civera 等人[73]提出了一种新的 1-point RANSAC 算法，该算法中预测摄像机运动的信息来自于扩展卡尔曼滤波器随时间变化得到的概率分布函数。这种方法在原理上对任何特定的运动没有限制，对 6 自由度估计也适用。虽然 Civer 等人提出的 1-point RANSAC 的 EKF 算法在原有基础上对不在模型内的运动估计有所改进，但是该算法以 EKF 滤波估计为核心，因为 EKF 滤波本身存在的缺点，所以各种因素所引起的模型误差导致的观测值与模型的不对应最终会导致滤波发散[54]。而这些导致模型误差的因素大部分来源于视频中运动的物体，如果可以将运动的物体上的特征点单独剔除，会减小滤波所发散的几率。本章节就是在 1-point RANSAC 单目是视觉 EKF 算法的基础上提出区分静态和动态特征点，筛选出视频中动态特征点，然后进行摄像机运动估计。

5.1 相关 RANSAC 算法

5.1.1 RANSAC 算法

RANSAC 是为了构造一个满足实验数据的模型的一种方法。相比于使用尽可能多的数据来获得初始结果然后剔除无效数据点的传统的滤波算法，RANSAC 使用尽可能少的初始数据集合进行建模，然后扩大数据集合的范围，测试其它数据是否满足这个模型。比如，检测一个二维点的集合是否满足是一个圆形的弧的一个测试。首先选用三个点（三个不在一条直线上对的点可以确定一个圆形），计算出这三个点所在圆的原点和半径。计算其它实验数据点对这个圆形模型的兼容性（即这些点的偏差小到可以被当做测量误差）。当得到足够多的兼容的点，RANSAC 将使用滤波如最小二乘法对圆形的参数做进一步的估计。

RANSAC 具体的过程如下：

● 给定一个能建立模型的最小子集（具有 n 个数据），数据点集合 P 的数量是大于 n [$\#(P)\geq n$]($\#()$表示求集合内的参数的数量)的，从 P 中随机选取有 n 个数据点的子集 $S1$，构建这个模型。用这个建立的模型 $M1$ 去验证 P 中点的其它子集 $S1^*$，如果点在模型 $M1$ 的容许误差内，则称集合 $S1^*$ 是 $S1$ 的一致性集合。

● t 是 P 中点是否适用于模型的点的数量的阈值。如果 $\#(S1^*)$ 大于阈值 t，就使用 $S1^*$ 去计算一个新的模型 $M1^*$。

● 如果 $\#(S1^*)$ 小于 t，随机选择一个新的子集 $S2$，然后重复上面的这个过程。如果验证一些点后，无法计算出一致性阈值 t 或找不到更多一致性的点，那么需要寻找更大的一致性的集合或终止此次模型验证。

从上面的过程可看出，RANSAC 算法中包括三个未设定的重要参数：(a)用于检测一个点是否在这个模型上的误差容忍度；(b)理想的迭代次数；(c)阈值 t，即正确的模型兼容的点的最少数量。下面是对这三个参数的讨论和确定。

a. 误差容忍度

一个模型的数据偏差是由数据误差和模型误差（一部分是用于实例化模型的数据误差）共同产生的。如果模型只是简单的由数据点构建的，那么就可以对误差容忍度设定合理的界限。但是，简单的估计方法是不可行的（比如通过实验来估计），因为抽样偏差一直随着数据的更新、模型的计算以及误差的测量而改变。所以误差容忍度只能设定一个或两个大于平均测量误差的标准偏差。

假定模型中理想的数据偏差只受到数据的影响，因此误差容忍度应该不同于每一个数据。另外，通过对比发现总误差的大小误差容忍度的变化幅度较小，所以对于所有数据一个误差容忍度已经足够。

b. 理想的迭代次数

寻找含有 n 个好的数据点的子集所需的实验的理想次数 k 决定算法是否可以停止寻找新的 P 得子集。w 表示任意被选中的数据点都在模型的误差容忍度内的概率。然后使得：

$$E(k)=b+2\times(1-b)\times b+3\times(1-b)^2\times b+\cdots+i\times(1-b)^{i-1}\times b+\ldots$$

$$E(k)=b\times[1+2\times a+3\times a^2+\cdots+i\times a^{i-1}+\ldots] \tag{5-1}$$

其中 $E(k)$ 是 k 的理想值，$b=w^n$ 以及 $a=1-b$。

由泰勒展开定理得到：

$$\frac{a}{1-a}=a+a^2+a^3+\cdots+a^i+\ldots \tag{5-2}$$

对(5.2)式进行求导，可得：

$$\frac{1}{(1-a)^2}=1+2\times a+3\times a^2+\cdots+i\times a^{i-1}+\ldots \tag{5-3}$$

所以综合上式可得：

$$E(k)=\frac{1}{b}=w^{-n} \tag{5-4}$$

通常情况下，在放弃前通过一个或两个标准差来做超过 $E(k)$ 的测试。记 k 的标准差为 $SD(k)$ 。

$$SD(k)=sqrt[E(k^2)-E(k)^2] \tag{5-5}$$

因为

$$E(k^2)=\sum_{i=0}^{\infty}(b\times i^2\times a^{i-1})=\sum_{i=0}^{\infty}[b\times i\times(i-1)\times a^{i-1}]+\sum_{i=0}^{\infty}(b\times i\times a^{i-1})$$

使用泰勒展开定理并进行二阶求导可得：

$$\frac{2a}{(1-a)^3}=\sum_{i=0}^{\infty}[i\times(i-1)\times a^{i-1}]$$

所以，

$$E(k^2)=\frac{2-b}{b^2}$$

求得：

$$SD(k)=[sqrt(1-w^n)]\times\frac{1}{w^n} \tag{5-6}$$

大部分情况 $SD(k)$ 和 $E(k)$ 的值近似相等，例如当 $w=0.5$ 和 $n=4$ 时，$E(k)=16$，$SD(k)=15.5$ 。这就意味着为了获得超过阈值 t 的抽样集需要对 k 做2次到3次的期望随机数的选择（如上所示）。

观测到稍微不寻常的点，如果要保证至少有一个随机选择是 n 个数据点的无误差集合的概率 z ，必须令至少有 k 个选择使得

$$(1-b)^k=1-z$$

所以

$$k=\frac{\log(1-z)}{\log(1-b)} \tag{5-7}$$

由此可以得到，如果 $w^n\ll 1$ ，那么 $k\approx\log(1-z)E(k)$ 。

c. 可被接受的一致性子集最低数量

阈值 t 是决定一个 P 中含有 n 个值的子集是否满足算法结束要求的基础。所以 t 的值必须足够大到满足两个要求：(1)数据的正确模型可以找到；(2)有足够的互相关的点可以满足最后的滤波过程（即对模型参数进行进一步估计）。

为了避免最后一致性的子集被不正确的模型兼容，假设 y 是任意数据点在不正确模型的误差容忍度内的概率。因为不能简单准确的计算 y 的值，同时要保证 y^{t-n} 尽可能的小，所以假设 $y<w$（w 是任意数据点在正确模型的误差容忍度内的概率）。假设 $y<0.5$，$t-n=5$，任意数据点将有 95% 的概率不会在不正确模型的误差容忍度内。

RANSAC 算法的提出具有很大的意义，它抛弃以往其他算法（如最小二乘法）用全体数据进行构建模型的方法，利用迭代不断随机抽取样本数据，再用剩余的样本数据进行不断检验，剔除不符合要求的样本点。在最大程度上减小畸变点对模型构造的影响。但是它迭代次数没有上限，如果设置上限就会导致结果的不理想。另外，RANSAC 的阈值要随着不同情况进行进一步的设定。

虽然 RANSAC 算法可以减小畸变点对模型构造的影响，但其构建模型的最小数据点和迭代次数的不确定，会造成计算时间的浪费。所以视觉方向上的 RANSAC，各研究人员将最小数据点固定到从 7 点到 1 点，这些点的固定有利于减小计算的复杂性，提高计算效率。本文主要对 1 点进行介绍。

5.1.2 基于 EKF 的 1 点 RANSAC 算法

基于 EKF 的 1-point RANSAC 算法中使用与 1-point RANSAC 算法中相似的改进，即减少随机假设的点的数量，但是 1-point RANSAC 算法中对运动模型有所限制。所以为了放宽限制，1-point RANSAC EKF 算法用更通用的方法去处理从帮助匹配的运动模型获得的额外信息，它可以处理平稳的六自由度摄像机运动。下面从两个部分分析就是 1-point RANSAC EKF 算法。

a. 基于 EKF 的以摄像机为中心的估计

为了克服扩展卡尔曼滤波早期出现的因为线性化误差而引起的不一致性，从而在视觉里程器造成一些影响：一旦场景中的点不再图片上或以前的地方不能被观测到，它就会偏离估计，随着摄像机的不停移动，摄像机位置的不确定性就会随着世界参考系而增长，所以为了解决这个问题提出了基于 EKF 的摄像机中心估计[74]。因为在通常情况下，EKF 计算所得的参数更接近与摄像机坐标系中的值，所以这样就可以通过基于 EKF 的摄像机估计算法来减小线性化误差，这样可以降低不确定性，线性化更加可靠。

设每一时刻 k 的估计参数化为多维高斯分布 $x_k \sim N(\hat{x}_k, P_k)$，其包含了不能被观测到的特征世界坐标系 $\hat{x}_k^C$ 以及地图 y^C。而摄像机运动中速度模型的统一使得保持状态矩阵速度估计在摄像机系 x_v^C 中。

$$\hat{x}_k^{C_k}=\begin{bmatrix}\hat{x}_W^{C_k}\\ \hat{x}_v^{C_k}\\ \hat{y}^{C_k}\end{bmatrix} \qquad P_k^{C_k}=\begin{bmatrix}P_W^{C_k} & P_{W_v}^{C_k} & P_{W_y}^{C_k}\\ P_{vW}^{C_k} & P_v^{C_k} & P_{v_y}^{C_k}\\ P_{yW}^{C_k} & P_{y_v}^{C_k} & P_y^{C_k}\end{bmatrix} \tag{5-8}$$

地图 y^{C_k} 由 n 个点特征 $y_i^{C_k}$ 构成，而且用深度坐标的逆参数化可以得到 $y_i^{C_k}$ 的值。

$$\hat{y}^{C_k}=\begin{bmatrix}\hat{y}_1^{C_k}\\ \vdots\\ \hat{y}_n^{C_k}\end{bmatrix}\quad P_y^{C_k}=\begin{bmatrix}P_{y_1}^{C_k} & \cdots & P_{y_1y_n}^{C_k}\\ \vdots & \ddots & \vdots\\ P_{y_ny_1}^{C_k} & \cdots & P_{y_n}^{C_k}\end{bmatrix} \tag{5-9}$$

速度状态向量 $x_v^{C_k}$ 中存储的是线性角速度，所以使用位置向量和四元数对世界坐标系坐标进行表示可以得到：

$$\hat{x}_v^{C_k}=\begin{bmatrix}\hat{v}^{C_k}\\ \hat{\omega}^{C_k}\end{bmatrix}\quad \hat{x}_W^{C_k}=\begin{bmatrix}\hat{r}_W^{C_k}\\ \hat{q}_W^{C_k}\end{bmatrix} \tag{5-10}$$

无论哪一种坐标系中，计算都包含三个步骤：EKF 预测、更新运算时刻以及将摄像机在 $k-1$ 时刻的坐标系转换到 k 时刻。

对 k 时刻进行预测时，世界坐标系和特征地图维持在 $k-1$ 时刻，用一个新的参数去表示 $k-1$ 和 k 时刻之间的摄像机运动：

$$\hat{x}_{k|k-1}^{C_{k-1}}=\begin{bmatrix}\hat{x}_W^{C_{k-1}}\\ \hat{x}_v^{C_{k-1}}\\ \hat{y}^{C_{k-1}}\\ \hat{x}_{C_k}^{C_{k-1}}\end{bmatrix} \tag{5-11}$$

$$P_{k|k-1}^{C_{k-1}}=\begin{bmatrix}P_W^{C_{k-1}} & P_{W_v}^{C_{k-1}} & P_{W_y}^{C_{k-1}} & 0\\ P_{vW}^{C_{k-1}} & P_v^{C_{k-1}} & P_{v_y}^{C_{k-1}} & P_{v_{C_k}}^{C_{k-1}}\\ P_{yW}^{C_{k-1}} & P_{y_v}^{C_{k-1}} & P_y^{C_{k-1}} & 0\\ 0 & P_{C_k v}^{C_{k-1}} & 0 & Q^{C_{k-1}}\end{bmatrix} \tag{5-12}$$

$Q^{C_{k-1}}$ 是零均值高斯加速度噪音的协方差。通过运用恒定的速度模型计算 $k-1$ 和 k 时刻之间的摄像机运动，$\hat{x}_{C_k}^{C_{k-1}}$ 用四元数和位置向量表示摄像机的位置：

$$\hat{x}_{C_k}^{C_{k-1}}=\begin{bmatrix}\hat{r}_{C_k}^{C_{k-1}}\\ \hat{q}_{C_k}^{C_{k-1}}\end{bmatrix} \tag{5-13}$$

与预期相反，摄像机中心估计方法中的运动不是应用在特征上，而是首先被应用于摄像机，然后坐标系在更新之后变换。运算的延缓是为了更进一步减小线性化误差：当更新的协方差小于预测的，在更新后运算将减小线性化的误差。

使用标准的扩展卡尔曼滤波器等式进行更新：

$$\hat{x}_k^{C_{k-1}}=\hat{x}_{k|k-1}^{C_{k-1}}+K_k(z_k-h_k(\hat{x}_{k|k-1}^{C_{k-1}}))$$

$$P_k^{C_{k-1}} = (I - K_k H_k) P_{k|k-1}^{C_{k-1}}$$

$$K_k = P_{k|k-1}^{C_{k-1}} H_k^{\mathrm{T}} S_k^{-1} \tag{5-14}$$

其中，h_k 是测量等式，是由针孔摄像机模型和径向畸变透镜模型构成。H_k 是模型的雅克比矩阵 $(H_k = \frac{\partial h}{\partial x^{C_{k-1}}}\Big|_{x_{k|k-1}^{C_{k-1}}})$，以及 S_k 是图像的协方差，是衍生的状态协方差和零均值图像噪音协方差 R_k 的和 $(S_k = H_k P_{k|k-1}^{C_{k-1}} H_k^{\mathrm{T}} + \mathrm{R}_k)$。

在运算过程，要对世界坐标系和从先前摄像机坐标系转移到当前坐标系的被估计特征点之间要有一个严格的转换。移除估计中先前坐标系和当前坐标系的转换，所以状态矩阵为：

$$\hat{x}_k^{C_k} = \begin{bmatrix} \hat{x}_W^{C_k} \\ \hat{x}_v^{C_k} \\ \hat{y}^{C_k} \end{bmatrix} \tag{5-15}$$

其中 $\hat{x}_W^{C_k}$，$\hat{x}_v^{C_k}$ 和 $\hat{y}^{C_k}$ 通过坐标 $\hat{x}_{C_k}^{C_{k-1}}$ 之间的运算得到：

$$\hat{x}_W^{C_k} = \Theta \hat{x}_{C_k}^{C_{k-1}} \oplus \hat{x}_W^{C_{k-1}}$$

$$\hat{x}_v^{C_k} = \Theta \hat{x}_{C_k}^{C_{k-1}} \oplus \hat{x}_v^{C_{k-1}}$$

$$\hat{x}_v^{C_k} = \Theta \hat{x}_{C_k}^{C_{k-1}} \oplus \hat{y}^{C_{k-1}} \tag{5-16}$$

用等式 $J_{C_{k-1} \to C_k}$ 的雅克比矩阵计算最终的协方差：

$$P_k^{C_k} = J_{C_{k-1} \to C_k} P_k^{C_{k-1}} J_{C_{k-1} \to C_k}^{\mathrm{T}} \tag{5-17}$$

b. 对于 EKF 的 1 点 RANSAC 的估计

要保证 RANSAC 随机假设建立的数量 n_{hyp} 中至少有一点满足先验概率 p，求得 n_{hyp} 为：

$$n_{hyp} = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^m)} \tag{5-18}$$

其中 ε 是假定的内点率，m 是建立模型的最少匹配点数。由公式(4.23)可知，n_{hyp} 的值随 m 的减小而成指数递减，所以当 $m=1$（即 1 点 RANSAC）可以大大减小计算复杂度。而每一个估计通过 EKF 对一个匹配点进行状态更新替代阈值比较，又从另一方面提高了计算效率。

算法主要分为两步，第一步是构成一个可靠的低内点的子集。在测量值 $\mathrm{N}(h_k(\hat{x}_{k|k-1}), S_k)$ 上，在预估的概率分布函数给出并且以 99% 概率出现的区域内的特征点中，用互相关搜索到的点集就是初匹配点集 z（其中可能包含错误匹配点集），z 是整个算法的输入。依靠这种方法（叫做

主动搜索）可以剔除小部分外点。

与标准 RANSAC 算法的主要区别是在假设和对候选点的选择循环上，方法如下：首先从点集 z 中选出随机匹配点 z_i。用 z_i 和 EKF 得到的预测状态值（遵从 $\mathrm{N}(\hat{x}_{k|k-1},\mathrm{P}_{k|k-1})$）共同作用的状态更新。其次在每个匹配点的状态更新后，计算剩余的匹配点。对于低于所要求的阈值的点（在作者的实验中设定的是 1.0 像素）被认为是低内点。如上对模型进行假设和验证，详细说明在公式(4.23)。大部分支持假设的剩余低内点被判定为内点，其它的匹配点可能是外点也可能是高内点。

第二步是提取出高内点。上一步中，剩余的点中不是外点就是高内点，其中的高内点可能是具有不确定深度估计的新初始化的点或被摄像机移动影响的近点。为了从高内点子集中排除错误的匹配，在局部更新消除相关误差后，高内点是互相关的，而外点不是。

5.2 基于 EKF 的 1 点 RANSAC 的改进算法

虽然基于 EKF 的 1 点 RANSAC 有效的降低了 RANSAC 算法的复杂性，但是其算法主要依赖于扩展卡尔曼滤波估计，而扩展卡尔曼滤波会随各种误差（模型的建立、系统干扰以及观测噪声等因素引起的模型误差）的存在发散，所以为了防止滤波发散必须有效的减小观测值与模型的误差。RANSAC 算法总是选择特征点集中满足某一模型的特征点更多的集合进行计算，所以实时跟踪场景中动态特征点数量较多时，模型建立可能就会出现较大误差，从而影响到滤波估计的结果，导致计算出现更大的误差。所以本小节提出对静态和动态特征点进行区分，通过静态特征点用基于 EKF 的 1 点 RANSAC 算法对摄像机运动进行建模。

5.2.1 选取样本

如今，摄像机的平均帧数为 25 帧/s，所以相对于室内机器人的速度，相邻帧的图像内容通常比较相似。所以如果有动态物体在图片中，相邻两帧的图像中动态特征点会发生较大位移，这样可以通过特征点的分布变化判断特征点是否为静态或者动态。其算法过程如下：

a. 首先对整个输入图像平均分为 10×10 的区域，共 100 个箱(bin)。每个箱设为 B_i，设 $B_k=[b_1^k,\cdots,b_{100}^k]$ 为当前时刻 k 每个箱中检测到的特征点个数的集合；$S_k=[s_1^k,s_2^k,\cdots,s_{100}^k]$ 为当前 k 时刻匹配的特征点个数的集合。求得每箱内有效特征点的比例为 $R_k=\left[r_1^k,\cdots,r_{100}^k\right]=\left[\frac{s_1^k}{b_1^k},\cdots,\frac{s_{100}^k}{b_{100}^k}\right]$，若其中 $b_n^k=0$，则使得 $r_n^k=0$。

b. 已知 $k-1$ 时刻点集 R_{k-1} 和 k 时刻点集 R_k。按照 R_{k-1} 点集把 $k-1$ 时刻的 B_i 划分为已知区域，

未知区域，以及空区域。其中，已知区域要满足$0.9 \le r_n^{k-1} \le 1$，空区域要满足$0 \le r_n^{k-1} \le 0.1$，余下的则为未知区域[75]。k时刻时，$r_n^{k-1} \le 0.1$且$0.9 \le r_n^k \le 1$或$r_n^{k-1} \ge 0.9$且$r_n^k \le 0.1$时，设B_i为动态区域；$0.1 < r_n^{k-1} \le 1$且$0.9 \le r_n^k \le 1$或$r_n^{k-1} \le 0.1$且$r_n^{k-1} \le 0.1$时，设B_i为静态区域；其它为未知区域；

c. 设定静态区域和未知区域内的已匹配中的点为获取初始点集Z，在选择构建模型的点时优先选取静态区域；同时设定动态区域和未知区域中的点为动态初始点集M，在构建运动物体模型时优先选取动态区域。

5.2.2 假设模型估计

因为当运动物体较大时，静态区域和未知区域的点较少，很难保证摄像机位姿估计和位置计算的正确性。因此提出建立动态模型估计，当动态物体在摄像机的图像中占据很大位置时，这时候摄像机的位置和动态物体的位置几乎重合，即摄像机和动态物体相遇，而摄像机的运动对相邻两帧的图片影响较小，所以可以默认摄像机此时是静止的，从而对动态物体的运动进行估计。但动态物体不是一直出现在摄像机拍摄的区域内，所以选择当动态区域在图片中的大小等于或稍大于静态区域时，同时对两个模型进行计算。

一般物体的运动多为刚性运动，多以对于运动物体模型的估计可以首先假设摄像机坐标系作为惯性坐标系，求出相对于摄像机坐标系的运动物体的坐标变化矩阵，然后通过摄像机坐标系与世界坐标系的转换关系从而得出，运动物体相对于世界坐标系的运动变化。

已知k时刻，设$S_k = T_k I$（从文章 2.1.1 得到），S_k是摄像机坐标系，I是惯性坐标系，T_k是两者之间的转换关系。同时通过计算得到k时刻运动物体的坐标系相对于摄像机坐标系的转换关系$D_k = T_k' S_{k-1}$，所以得到：

$$D_k = T_k' T_{k-1} I \tag{5-19}$$

当摄像机在k时无法得到估计值时，即动态区域较大时，此时运动物体与摄像机位置几乎一致，所以可以近似得到$S_k = D_k$。

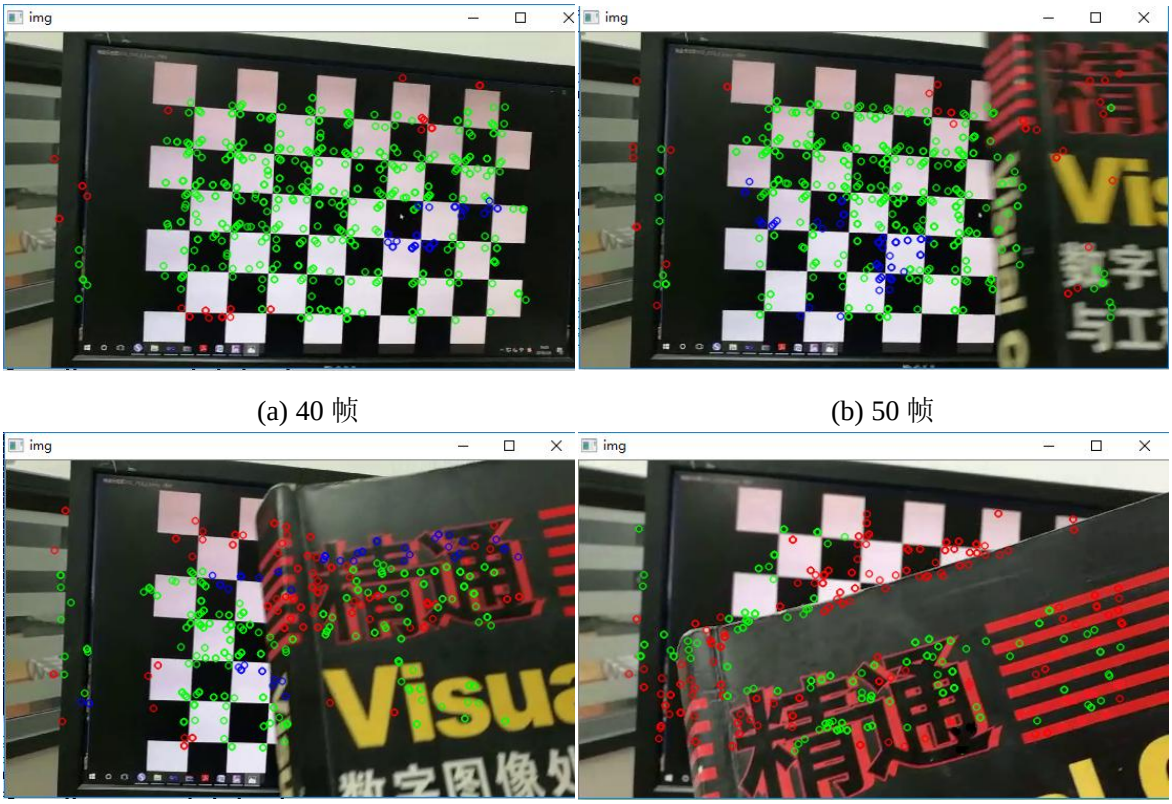
5.3 实验结果及分析

为了使得算法在室内环境有更好的效果，本章中采用的检测方法是本文中第三章节的算法，匹配方法是第四章的算法。

5.3.1 动态区域分解

因为本文中所使用的摄像机采集的视频不够理想，导致对于特征点的检测结果过少，所以背景只能使用棋盘标定图，这样在保证具有较多特征点的情况进行实验。具体操作是以棋盘标定图为背景，在摄像头前，对一本书进行移动，然后显示出静态点和动态点，从而分析该算法

是否可以区分大部分静态点和动态点。

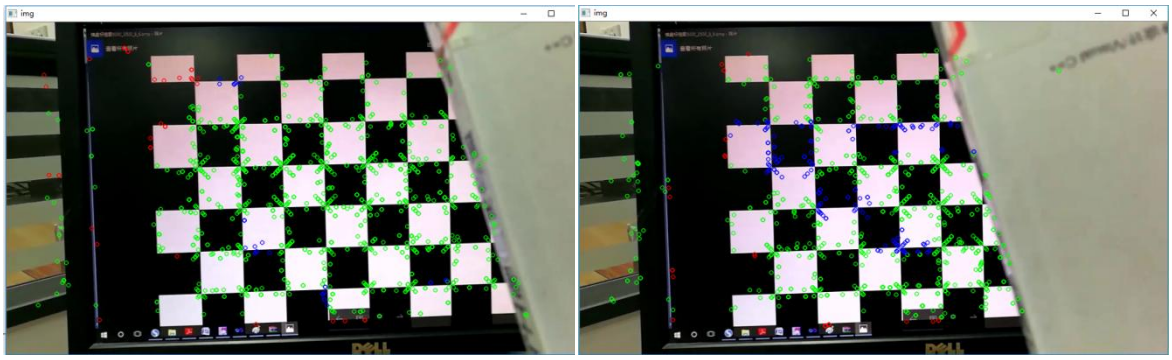


(a) 40 帧　　(b) 50 帧

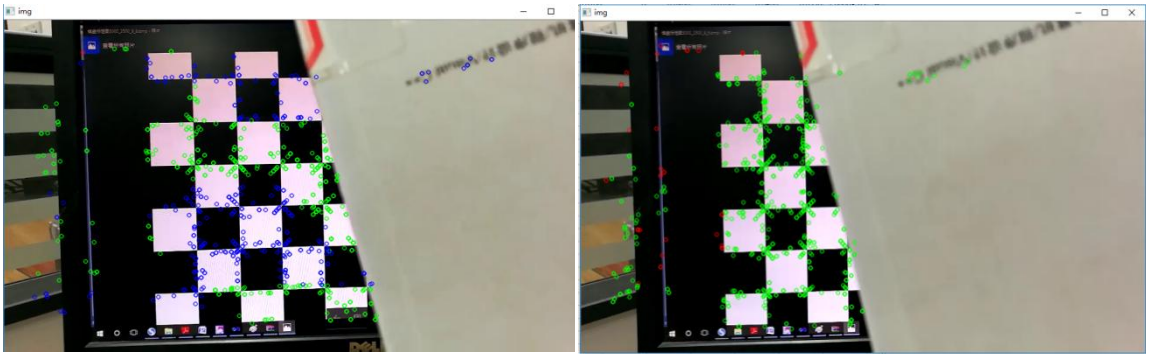
(c) 80 帧　　(d) 120 帧

图 5.1 动态运动物体上特征点较多的时点的划分

图 5.1 中可得，绿色的为分辨出的静态点，蓝色为不确定的点，而红色为动态点。本章算法对于动态物体有较多特征的图片中对于静态点的划分有所不足，但是基本上动态点都集中在动态物体上，当通过静态点无法计算摄像机位姿时，依然可以通过动态点对动态物体的估算，然后间接得到摄像机位姿。



(a) 120 帧　　(b) 150 帧



(c) 200 帧　　(d) 280 帧

图 5.2 动态运动物体上特征点较多的时点的划分

由图 5.2 可以看出，虽然在动态物体上没有特征点，但是在静态物体上大部分的特征点已被划分为静态点和不确定点，所以说本文对于静态和动态点就有较好的区分度，对于位姿估计的计算可以提供较好的铺垫。

5.3.2 位姿估计

为了最终测试本章节的方法，再使用同种检测方法和匹配方法（即第三章和第四章改进算法）的前提下，和 RANSAC 算法和基于 EKF 的 1 点 RANSAC 算法的结果进行比较。实验中采用公开的 Rawseeds 数据集中自然光线下，有动态物体移动的室内环境下前置摄像头所拍的视频，该视频中每帧是 640*280 像素。其中摄像机被安置在机器人上，具有 5 自由度信息（即空间坐标$[x, y, z]^T$ 和摄像机相对于中心的偏航角和俯仰角），匀速行驶在室内的走廊间，期间会有人不断在摄像头前路过。本章实验主要采用视频中有人运动的 200-1000 帧进行实验验证。

表 5.1 每帧计算的平均时间

算法	RANSAC	1PRANSAC	本章算法
每帧计算的平均时间/s	1.173	1.032	1.053

从表 5.1 可以看出虽然本章算法中加入先验方法，但是因为对每帧图像进行静动态点的计算，先采用静态点进行姿态计算，这样需要迭代的次数大大降低，所以每帧计算平均时间较小，但是对于静态点个数不足，还需要代入不明确的点集重新计算，所以可能在某一帧上的需要的计算时间较长，所以平均时间相较于 1PRANSAC 算法有所增加。

下图是三种算法计算所得坐标对于实际坐标$[x, y, z]^T$ 误差值。

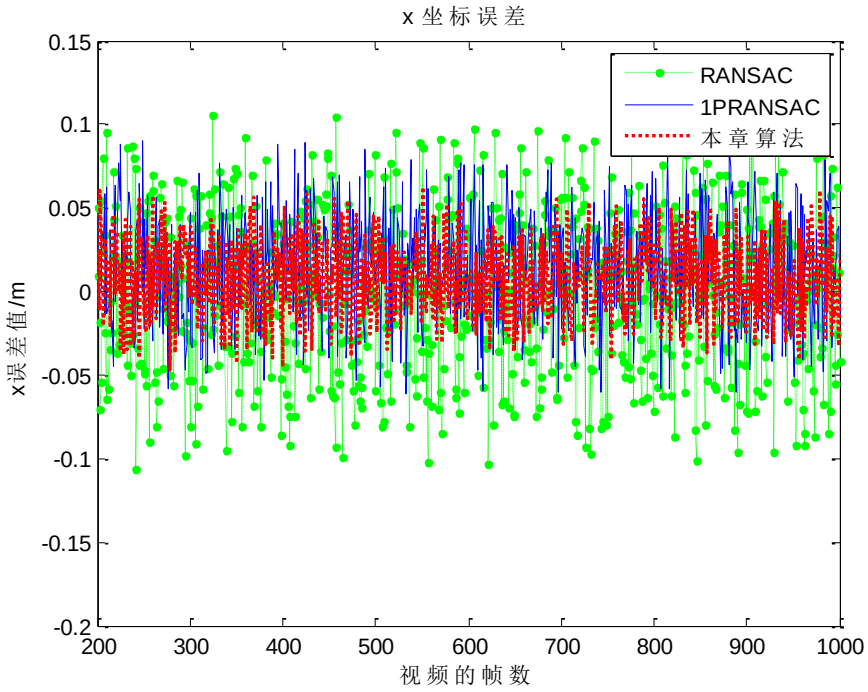


图 5.3 x坐标的误差

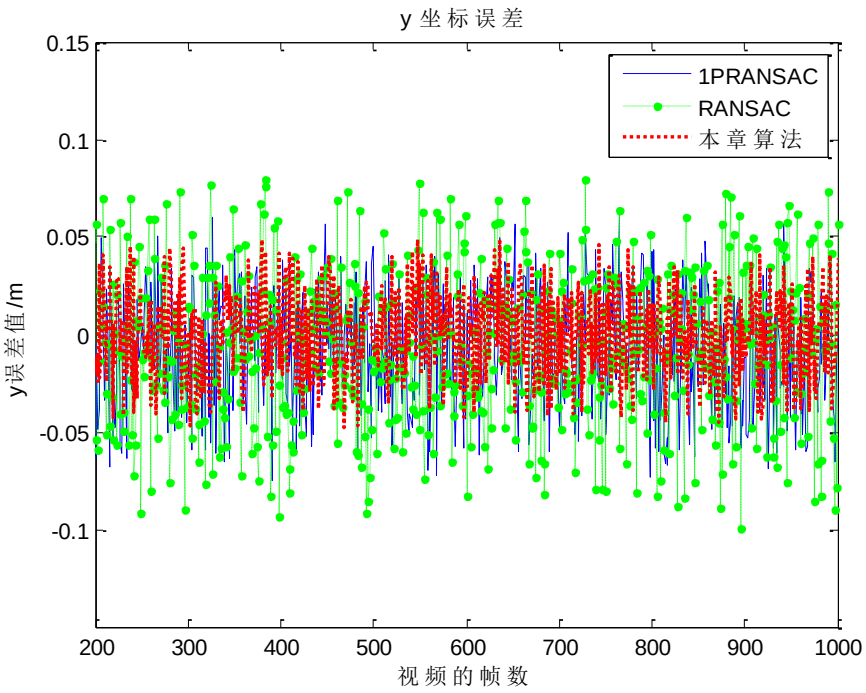


图 5.4 *y* 坐标的误差

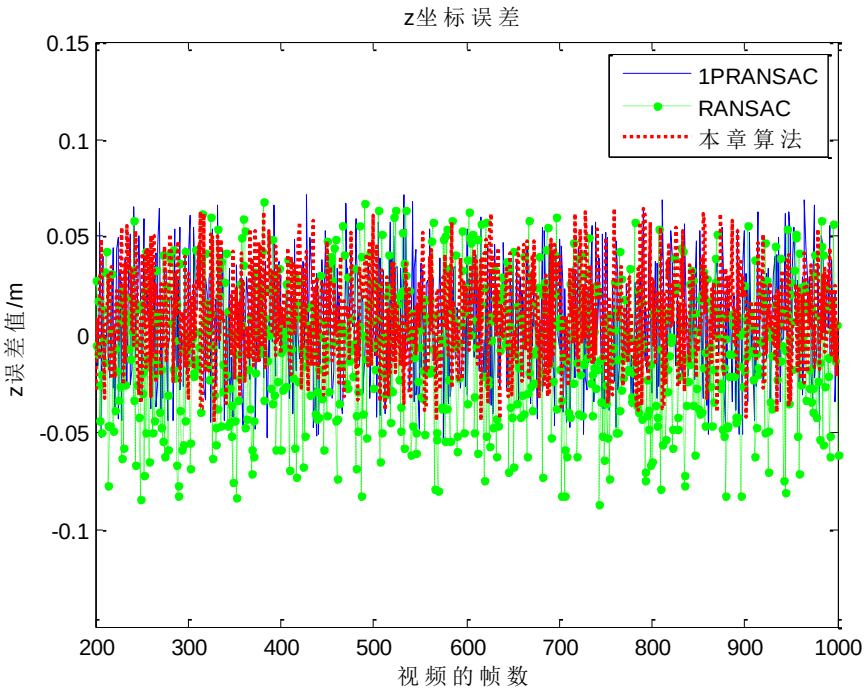


图 5.5 *z* 坐标的误差

从图 5.3 至 5.5 可以看出，相较于 RANSAC 算法和 1-point RANSAC 算法，本章算法在空间坐标上的误差相对较小，这是动态物体的存在会对其它两种算法造成一定的影响，但因为视频中的动态物体较小，场景结构简单，静态特征点分布依旧较多，所以其它两种算法计算结果依旧没出现较大误差，所以三种算法的误差都在 5 厘米到 10 厘米左右的范围。

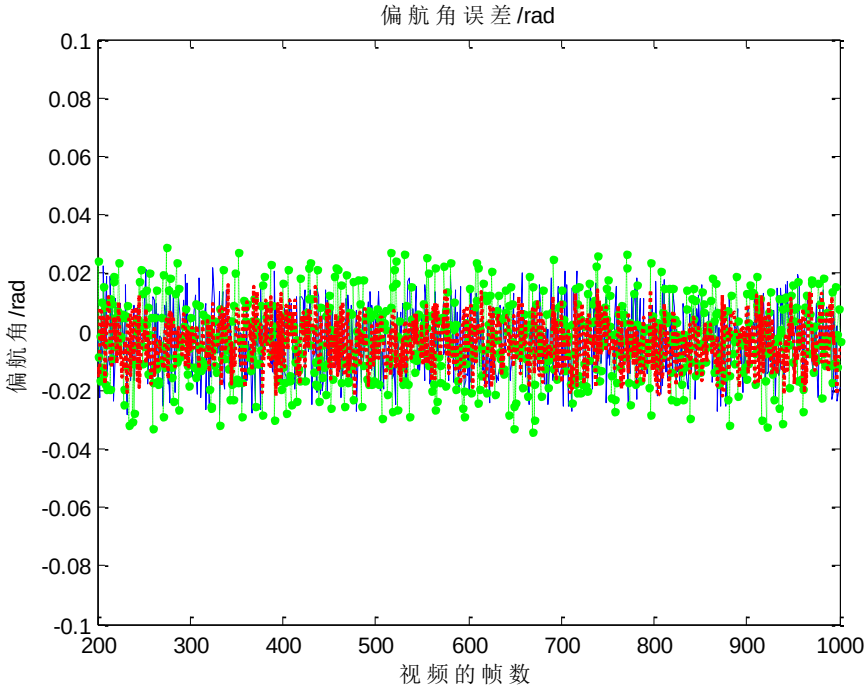


图 5.6 偏航角的误差

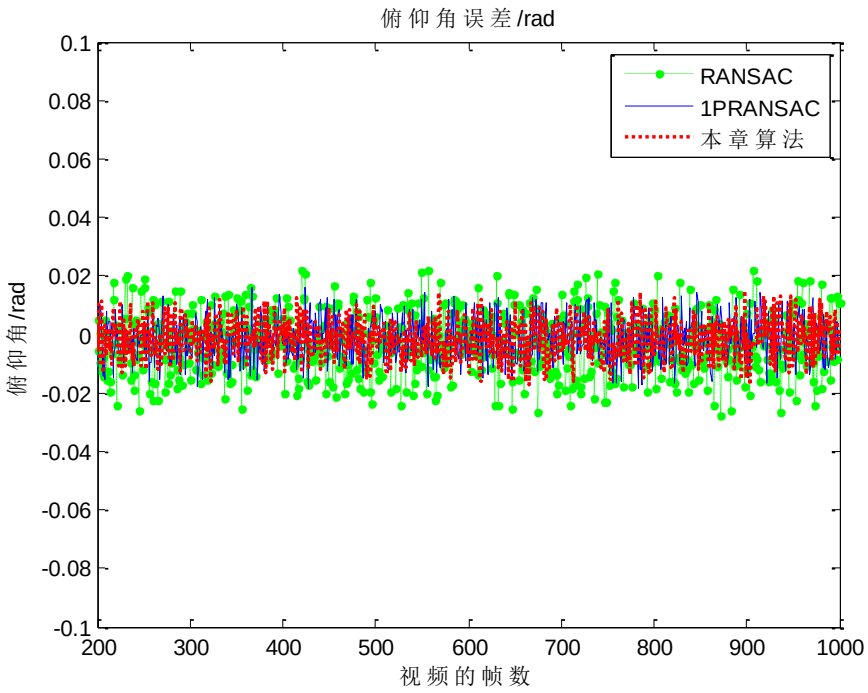


图 5.7 俯仰角的误差

图 5.6 和 5.7 展示的是三种算法在偏航角和俯仰角的误差，算法中 1-point RANSAC 和本章算法误差皆 0.01rad 左右，但相较于 1-point RANSAC 算法，本章算法误差相对较小，而 RANSAC 误差较大，最高的误差达到 0.02rad。

5.4 本章小结

本章主要分析了对摄像机位姿估计和运动估计的算法，比较每种算法的优劣性，从而采用基于 EKF 的 1 点 RANSAC 算法。为了避免算法的发散，提出对静动态点进行区分，然后同时求摄像机位姿和动态物体位姿，解决场景中有运动物体的情况。在最后结果中，检测算法使用的是本文第三章的算法，匹配结算法用的是本文第四章的算法。同 RANSAC 和基于 EKF 的 1 点 RANSAC 进行比较，实验结果得出本文提出的视觉 SLAM 视觉里程计部分的算法适用于室内环境，对于室内环境建图有较好的鲁棒性。

第六章 总结与展望

室内环境一般结构比较简单，地势平坦，视角变化不大，但是因为环境不够开阔，物体较多，光线变化较大，人比较密集，摄像机行进速度慢且其对图像的采集容易受到动态物体的影响。本文中，为了使算法更好的适用于室内的场景对于基于特征的单目 SLAM 算法的前端部分提出了三个改进：

首先，因为室内环境简单，所以对特征点的检测比较困难，如果图片中有大部分单一背景（比如墙），检测到的特征点数量会大幅度减少且分布相对集中，所以提出了基于栅格的特征点检测方法，综合了角点检测和斑点检测，实验证明检测到的特征点数量明显增多，且对大部分环境下可以保证特征点的分布比较均匀。

然后，提出基于简化 FREAK 的特征点匹配改进算法。FREAK 算法采用人类视网膜采样模型，感受野的重叠存在的冗余，更光线变化更加敏感。但实验发现感受野之间过多的重叠会增加计算的复杂度和存储空间，所以对采样模型进行简化，同时对两个距离较大的感受野之间进行线性插值，增加对灰度值变化的描述。实验证明该算法对于光照和分辨率变化下的正确匹配率更高。

最后，对于已匹配的特征点进行摄像机位姿估计。考虑到现实生活中，室内运动物体对摄像机位姿估计的影响，提出具有动态环境鲁棒性的基于 1 点 RANSAC 的 EKF 算法。该算法对静态点和动态点进行区分，同时对这两者进行位姿估计，当动态点占据图片中较大空间时，利用动态物体的位姿确定摄像机位姿。实验证明，本文提出的方法更适合具有较大移动物体的室内环境。

但本文算法也有很多不足。因为算法是纯理论研究，实时性不是作为主要考虑问题，如果加入硬件平台，可能要涉及算法运行时间的考虑。其次，匹配算法不够稳定，对于大幅度的摄像机运动（有大的旋转角度），可能会出现正确匹配点数量较少。最后区分动静态点进行位姿估计的方法的前提是摄像机本身运动速度较慢且有较多分布比较均匀的匹配点，对于快速移动的摄像机和运动物体，可能会出现动静态点的区分不够理想。这些都是待进一步研究的问题。

参考文献

[1] Smith R, Cheeseman P. Estimating uncertain spatial relationships in robotics[C]. Proceedings of the Second Annual Conference on Unceratianty in Artificial Intelligence, 1986(4): 435-461.

[2] Leonard J J, Durrant-Whyte H F. Mobile Robot Localization by Tracking Geometric Beacons[J]. IEEE Transactions on Robotics and Automation, 1991,7(3): 376-382.

[3] 权美香, 朴松昊, 李国. 视觉 SLAM 综述[J]. 智能系统学报, 2016, 11(6):768-776.

[4] Matthies L. Dynamic stereo vision[D]. Ph. D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1989.

[5] Matthies L, Shafer S. Error modeling in stereo navigation[J]. Robotics and Automation, IEEE Journal of, 1987, 3(3): 239-248.

[6] Lacroix S, Mallet A, Chatila R, et al. Rover self localization in planetary-like environments[C]. Artificial Intelligence, Robotics and Automation in Space. 1999: 433-440.

[7] Nistér D, Naroditsky O, Bergen J. Visual Odometry[C]. Proc. IEEE Conf. Computer Vision and Pattern Recognition. 2004: 652-659 .

[8] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[M]. ACM, 1981.

[9] Dacison A J, Reid I D, Molton N D, and Stasse O. MonoSLAM: Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 26(6): 1052-1067.

[10] Shi J, Tomasi. Good features to track[C]. Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94. 1994 IEEE Computer Society Conference on. IEEE, 2002:593-600.

[11] Davison A J. Active Search for Real-Time Vision[C]. Tenth IEEE International Conference on Computer Vision. IEEE Computer Society, 2005:66-73.

[12] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces[C]. IEEE and ACM International Symposium on Mixed and Augmented Reality. Nara, Japan, 2007:

2225-234.

[13] Klein G, Murray D. Improving the agility of keyframe-based SLAM[M]. Proceedings of European Conference on Computer Vision. Heidelberg, Springer, 2008, 2: 802-815.

[14] Newcombe R A, Lovegrove S J, Davison A J. DTAM: Dense tracking and mapping in real-time[C]. Proceedings of IEEE international Conference on Computer Vision. Barcelona, Spain, 2011: 2320-2327.

[15] 刘浩敏, 章国锋, 鲍虎军. 基于单目视觉的同时定位与地图构建方法综述[J]. 计算机辅助设计与图形学学报, 2016, 28(6):855-868.

[16] Civera, J, Davison, A J, Montiel, J. M Martínez. Inverse Depth Parametrization for Monocular SLAM[J]. IEEE Transactions on Robotics, 2008, 24(5):932-945.

[17] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[M]. Computer Vision – ECCV 2014. Springer International Publishing, 2014:834-849.

[18] Engel J, Stückler J, Cremers D. Large-scale direct SLAM with stereo cameras[C]. Proceedings of IEEE International Conference on Intelligent Robots and Systems. IEEE, 2015:1935-1942.

[19] Caruso D, Engel J, Cremers D. Large-scale direct SLAM for omnidirectional cameras[C]. Proceedings of IEEE International Conference on Intelligent Robots and Systems. IEEE, 2015:141-148.

[20] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]. IEEE International Conference on Robotic and Automation. HongKong, China, 2014: 15-22.

[21] Pizzoli M, Forster C, Scaramuzza D. REMODE: Probabilistic, monocular dense reconstruction in real time[C]. IEEE International Conference on Robotics and Automation. IEEE, 2014:2609-2616.

[22] George Vogiatzis, Carlos Hernández. Video-based, real-time multi-view stereo [J]. Image and vision Computing, 2011, 29(7):434-441.

[23] Forster C, Zhang Z, Gassner M, et al. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems[J]. IEEE Transactions on Robotics, 2017, 33(2):249-265.

[24] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2017, 31(5):1147-1163.

[25] 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.

[26] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]. IEEE International Conference on Computer Vision. IEEE, 2011:2564-2571.

[27] Mur-Artal R, Tardós J D. Visual-Inertial Monocular SLAM With Map Reuse[J]. IEEE Robotics & Automation Letters, 2017, 2(2):796-803.

[28] Forster C, Carlone L, Dellaert F, et al. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation[J]. Georgia Institute of Technology, 2015.

[29] Forster C, Carlone L, Dellaert F, et al. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation[C]. Robotics: Science and Systems. 2015.

[30] Forster C, Carlone L, Dellaert F, et al. On-Manifold Preintegration for Real-Time Visual--Inertial Odometry[J]. IEEE Transactions on Robotics, 2017, 33(1):1-21.

[31] Engel J, Koltun V, Cremers D. Direct Sparse Odometry.[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, PP(99):1-1.

[32] 温丰, 柴晓杰, 朱智平等. 基于单目视觉的 SLAM 算法研究[J]. 系统科学与数学, 2010, 30(6):827-839.

[33] 徐伟杰, 李平, 韩波. 基于 2 点 RANSAC 的无人直升机单目视觉 SLAM[J]. 机器人, 2012, 34(1):65-71.

[34] 谭伟. 动态场景下的基于单目摄像头的鲁棒同时定位与建图[D]. 浙江大学, 2015.

[35] Li P, Qin T, Hu B, et al. Monocular Visual-Inertial State Estimation for Mobile Augmented Reality[C]. IEEE International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2017:11-21.

[36] 高翔, 张涛. 视觉 SLAM 十四讲:从理论到实践(M). 北京, 电子工业出版社.

[37] Lowe D G. Object Recognition from Local Scale-Invariant Features[C]. ICCV IEEE Computer Society, 1999:1150-1157.

[38] Moravec H P. Obstacle avoidance and navigation in the real world by a seeing robot rover[M]. Stanford University, 1980.

[39] Harris C G, Pike J M. 3D positional integration from image sequences[J]. Image & Vision Computing, 1988, 6(2):87-90.

[40] Rosten E, Drummond T. Machine learning for high-speed corner detection[C]. European Conference on Computer Vision. Springer-Verlag, 2006:430-443.

[41] Rosten E, Porter R, Drummond T. Faster and Better: A Machine Learning Approach to Corner Detection[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2010, 32(1) : 105-119

[42] Calonder M, Lepetit V, Strecha C, et al. BRIEF: binary robust independent elementary features[C]. European Conference on Computer Vision. Springer-Verlag, 2010:778-792.

[43] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]. IEEE International Conference on Computer Vision. IEEE, 2011:2564-2571.

[44] Leutenegger S, Chli M, Siegwart R Y. BRISK: Binary Robust invariant scalable keypoints[C]. International Conference on Computer Vision. IEEE Computer Society, 2011:2548-2555.

[45] Alahi A, Ortiz R, Vandergheynst P. FREAK: Fast Retina Keypoint[C]. Computer Vision and Pattern Recognition. IEEE, 2012:510-517.

[46] Muja M. Fast approximate nearest neighbors with automatic algorithm configuration[C]. International Conference on Computer Vision Theory and Application Vissapp. 2009:331-340.

[47] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. International Journal of Robotics Research, 1987, 5:56-68.

[48] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[M]. Autonomous robot vehicles. Springer-Verlag New York, Inc. 1990:435-461.

[49] Triggs B, McLauchlan, Philip F, Hartley R I, et al. Bundle Adjustment - A Modern Synthesis[C]. International Workshop on Vision Algorithms: Theory and Practice. Springer-Verlag, 1999:298-372.

[50] Konolige, K, Agrawal, M. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping[J]. IEEE Transactions on Robotics, 2008, 24(5):1066-1077.

[51] Golfarelli M, Maio D, Rizzi S. Elastic correction of dead-reckoning errors in map building[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems, Piscataway: IEEE, 1998:905-911.

[52] Hauke Strasdat, J.M.M. Montiel, Andrew J. Davison. Visual SLAM: Why filter?[J]. Image

& Vision Computing, 2012, 30(2):65-77.

[53] 李捐. 基于单目视觉的移动机器人 SLAM 问题的研究[D]. 哈尔滨工业大学, 2013.

[54] Viola P, Jones M. Rapid Object Detection using a Boosted Cascade of Simple Features[C]. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2003: 511-518.

[55] Mikolajczyk K, Schmid C. Indexing based on scale invariant interest points[C]. Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE, 2001:525-531.

[56] Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[J]. Computer Vision & Image Understanding, 2006, 110(3):404-417.

[57] Lindeberg T, Bretzner L. Real-time scale selection in hybrid multi-scale representations[C]. International Conference on Scale Space Methods in Computer Vision. Springer-Verlag, 2003:148-163.

[58] Brown M, Lowe D G. Invariant Features from Interest Point Groups[C]. British Machine Vision Conference 2002, BMVC 2002, Cardiff, Uk, 2-5 September. DBLP, 2002:656-665.

[59] Fraundorfer F, Scaramuzza D. Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications[J]. IEEE Robotics & Automation Magazine, 2012, 19(2):78-90.

[60] Mikolajczyk K, Schmid C. An Affine Invariant Interest Point Detector[C]. European Conference on Computer Vision. Springer-Verlag, 2002:128-142.

[61] 索春宝，杨东清，刘云鹏. 多种角度比较 SIFT、SURF、BRISK、ORB、FREAK 算法[J]. 北京测绘，2014, 4:2-26

[62] Tola. E, Lepetit. V, and Fua. P. Daisy: An efficient dense descriptor applied to wide-baseline stereo[C]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010, 3, 32(5):815-830,

[63] Candès E J, Eldar Y C, Needell D, et al. Compressed sensing with coherent and redundant dictionaries[J]. Applied & Computational Harmonic Analysis, 2011, 31(1):59-73.

[64] Olshausen B A, Field D J. What Is the Other 85 Percent of V1 Doing?[M]. 23 Problems in Systems Neuroscience. 2006.

[65] 谢红，王石川，解武. 基于改进的 FREAK 算法的图像特征点匹配[J]. 应用科技，2016,43(4):1-7

[66] 李晶皎，赵越，王爱侠，李贞妮，杨丹. 基于改进 FREAK 的增强现实实时注册算法[J]. 小型微型计算机系统，2016,37(1):137-177

[67] Wang Jiayong, Wang Xue-mei, Yang Xiao-gang,et al.CS-FREAK: An improved binary descriptor[J]. Communications in Computer and Information Science, THE 8th Conference on Image and Graphics Technologies and Application(IGTA), 2014, CCIS 437: 129-136.

[68] 房贻广，刘武，高梦珠，谭守标，张骥. 基于 FREAK 描述子的精确图像配准改进算法[J]. 计算机应用，2016, 36(12):3402 -3405, 3410

[69] Torr P H S, Murray D W. Outlier detection and motion segmentation[J]. Proceedings of SPIE - The International Society for Optical Engineering, 1993, 2059:432--443.

[70] Capel D P. An effective bail-out test for RANSAC consensus scoring[C]. British Machine Vision Conference 2005, Oxford, UK, 2008, 9:629-638.

[71] Chum O, Matas J. Optimal randomized RANSAC.[J]. IEEE Transactions on Pattern Analysis & machine Intelligence, 2008, 30(8):1472-1482.

[72] Scaramuzza D, Fraundorfer F, Siegwart R. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC[C]. IEEE International Conference on Robotics and Automation. IEEE, 2009:488-494.

[73] Civera J, Grasa O G, Davison A J, et al. 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry[J]. Journal of Field Robotics, 2010, 27(5):609–631.

[74] Castellanos J A, Neira J, Tardos J D. Limits to the consistency of EKF-based SLAM[J]. Symposium on Intelligent Autonomous Vehicles, 2011.

[75] Wolf D F, Sukhatme G S. Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments[J]. Autonomous Robots, 2005, 19(1):53-65.

致 谢

时间匆匆而过，转眼两年半的时间已过。回想以往的一切，所有的记忆在我眼前划过，在这两年半中，我首先要衷心感谢我的导师范胜林教授，在课题研究的过程中，他给我提出了指导性的意见，并在我有问题的情况下给予耐心的指导。然后要感谢我已经毕业的师兄师姐们，他们在我的研究生生涯给出了许多指导性意见。其次感谢我的朋友们在我对实验进行仿真的过程中，帮助我解决了许多程序上的问题。

在学期间的研究成果及发表的学术论文

攻读硕士学位期间发表（录用）论文情况

侯豆，范胜林，金春杨《基于快速视网膜匹配的图像匹配改进算法》，导航与控制，已录用。

攻读硕士学位期间参加科研项目情况