浙江工业大学

硕士学位论文

论文题目：基于单目 SLAM 的稠密重建

作者姓名 毛栋炳

指导教师 陈胜勇教授、张剑华副教授

学科专业 计算机科学与技术

培养类别 全日制学术型硕士

所在学院 计算机科学与技术学院

提交日期 2018 年 5 月

浙江工业大学硕士学位论文

基于单目 SLAM 的稠密重建

作者姓名：毛栎炳

指导教师：陈胜勇 教授

张剑华 副教授

浙江工业大学计算机科学与技术学院

2018 年 5 月

Dissertation Submitted to Zhejiang University of Technology
for the Degree of Master

Dense Reconstruction Based on Monocular SLAM

Candidate: Libing Mao

Advisor: Professor Shenyong Chen

Associate Professor Jianhua Zhang

College of Computer Science and Technology
Zhejiang University of Technology
May 2018

浙江工业大学

学位论文原创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的研究成果。除文中已经加以标注引用的内容外，本论文不包含其他个人或集体已经发表或撰写过的研究成果，也不含为获得浙江工业大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

作者签名：毛栋炳　　日期：2018 年 11 月 7 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权浙江工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、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密□，在一年解密后适用本授权书。

2、保密□，在三年解密后适用本授权书。

3、不保密☑。

（请在以上相应方框内打"√"）

作者签名：毛栋炳　　日期：2018 年 11 月 7 日

导师签名：　　日期：2018 年 11 月 7 日

基于单目 SLAM 的稠密重建

摘　　要

随着人工智能的蓬勃发展，计算机视觉作为目前人工智能中一个不可或缺的一部分受到越来越多研究者的密切关注，而三维重建又是计算机视觉中相当热门的研究点。三维重建主要研究的是根据相机得到的图像恢复出周围环境的三维信息。

本课题针对目前基于单目相机进行稠密三维重建方法的不足，从基于直接法的稠密三维重建和基于特征点法的稠密三维重建两方面展开研究，具体完成的工作如下：

1、铺垫了课题中需要的技术原理比如三维空间的刚体运动、欧式变换、对极约束、本质矩阵、单应矩阵以及基于特征法和直接法视觉里程计的算法基础。

2、在基于直接法的稠密重建算法中，着重介绍了整个流程，包括了全局相机的光度标定，视觉里程计前端，数据桥接，稠密三维重建后端四个主要部分。针对视觉里程计提供的半稠密点云、相机位姿结合图优化分割算法实现对环境平面估计，使用稠密建图算法优化稠密点云的结果。本文提出了数据点云选择策略，相机坐标系统一策略以及数据交换容纳池从而更好的实现数据融合，提高重建效率，降低资源消耗。并对 SLAM 公共数据集进行多次实验测试，算法结果表明该三维重建算法高效鲁棒实用。

3、在基于特征法的稠密重建算法中，首先介绍了结合深度学习的边缘检测流程，然后在基于特征点的 SLAM 系统获取的关键帧中融合边缘检测和基于图的分割结果，提出了一种新的融合算法，计算得到更为鲁棒、效果更为明显的轮廓图。通过此图，再结合稀疏点云可以实现稠密重建。对于重复点云的显示来说，我们采用了一个基于字典相似度计算的稠密点云融合算法，减小了点云存储空间，提高了重建显示效果。实验中，我们使用了公共的数据集，达到了比较不错的重建结果和效率。

本文使用的数据集均为公共数据集，我们的算法均能够很好的完成视频帧的跟踪以及对周围环境的实时三维重建，实验结果表明了我们的算法具有相当的高效性和实用价值。

关键词：单目 SLAM，三维重建，图像分割，边缘检测

DENSE RECONSTRUCTION BASED ON MONOCULAR SLAM

ABSTRACT

With the rapid development of artificial intelligence, computer vision is an indispensable part of current artificial intelligence and attracts more and more researchers' close attention. 3D reconstruction, on the other hand, belongs to a rather hot research point in computer vision. The main research of 3D reconstruction is to recover the three-dimensional information of the surrounding environment based on the images obtained by the camera.

This thesis mainly studies in the following two aspects: the dense three-dimensional reconstruction based on direct method and the dense three-dimensional reconstruction based on the feature point method, and makes some improvements on the reconstruction effect. The completed work is as follows:

1. Paving the technical principles needed in the subject such as the rigid body movement in three-dimensional space, the Euclidean transformation, the pole constraint, the essential matrix, the homography matrix, and the algorithmic basis of the feature-based and direct-method visual odometry.

2. In the dense reconstruction algorithm based on the direct method, the whole process is emphasized, including the four main parts of photometric calibration of the global camera, visual odometer front end, data bridging and dense three-dimensional reconstruction back end. Aiming at the semi-dense point cloud provided by visual odometer and the camera pose algorithm combined with graph optimization and segmentation algorithm to realize the environment plane estimation, the dense map algorithm is used to optimize the result of dense point cloud. The proposed data point cloud selection strategy, a camera coordinate system strategy, data exchange storage pool to achieve better data fusion, improve reconstruction efficiency and reduce resource consumption. The SLAM public dataset was tested experimentally many times. The algorithm results show that the 3D reconstruction algorithm is efficient and robust.

3. In the dense reconstruction algorithm based on the feature method, this paper mainly introduces the process of learning the edge information of the image by combining depth with the keyframes provided by SLAM for image segmentation and edge detection. We propose an image fusion algorithm to obtain a more obvious contour map and combine with the sparse point

cloud for dense reconstruction. We use a fusion algorithm based on dictionary similarity calculation for the repetitive point cloud display, which reduces the point cloud storage space and improves the reconstruction display effect. We use the dataset in our experimental process, to achieve a relatively good reconstruction results and efficiency.

The datasets used in this paper are all public datasets, and the experimental results show that our algorithm has high efficiency and practical value.

Key Words: monocular SLAM, three-dimensional reconstruction, image segmentation, edge detection

目　　录

第 1 章 绪 论

1.1 研究背景与研究意义

近四十年以来，计算机视觉的各个领域取得了长足的进步与发展。实时单目即实时同步定位和地图构建系统（SLAM）和三维空间环境重建技术俨然成为当下学术界流行的研究课题之一。这里主要有两个原因：这些技术广泛使用在机器人领域，特别是用于无人飞行器的自主导航中；同时还有近几年基于增强和虚拟现实技术的各种各样的应用正在逐渐走进大众的生产生活之中。随着信息科技的不断发展，机器人或是移动端对于外界环境的感知要求越来越高，精度的要求越来越细。为此很多研究者尝试了三维激光扫描做三维重建，这种方法是通过主动地投射光线到物体表面然后再通过光线接收器接收物体表面反射的光线完成深度测量，还原出目标物体的形状。与此同时，有些激光设备依赖于红外光线，因而只能适用于室内的环境，室外的强光环境会对设备测量精度造成很大的误差影响。激光扫描设备虽然重建结果好、精度十分高，但是在现阶段价格还是比较的昂贵，因此主要应用在大型的逆向工程中、3D 打印技术以及一些公司产品的质量检测中。此外还出现了利用几何造型方法重塑模型，涌现出了很多不错的制作软件，如 MAYA，3DMax 等。这个方法能达到最为精细的结果但是极大的消耗了人力和物力。而且使用这些软件一般需要有设计专业方向的人员，一般人很难接触并达到精良的模型制作，但是以上的这些都很难普及在机器人的应用产品中。

而基于计算机视觉的三维重建方法因为设备要求低，资源需求量小在当今社会越来越受到欢迎。在计算机视觉众多不同类型的传感器中，相机又是属于其中比较廉价的，并且能够采集相当丰富的环境信息，受到了各国研究者的青睐。同样视觉 SLAM 也是受到国内外空前的关注，它仅仅通过一个简单的相机就能完成各种各样的任务。很多研究者关注着基于 RGB-D 相机的 SLAM 重建算法，包括 DTAM[6]，PTAM[21]，KinectFusion[5]，ElasticFusion[9]，DPPTAM[3]等。这些算法利用深度或立体视觉相机，这些具备深度信息的传感器，它们能提供可靠的深度信息范围。此外，一些基于 CPU 运行的单目视觉 SLAM 三维重建算法目前也非常火爆，例如 ORB_SLAM2[24]，SVO[20]，DSO[1]等。近几年伴随着深度学习的浪潮，三维重建也出现了一些 SLAM 结合深度学习的案例，如

CNN-SLAM[10]：它通过卷积神经网络进行深度预测，然后通过单目 SLAM 测量数据进行数据融合，可以对弱纹理区域实现很好的重建效果，有较强的鲁棒性和准确性。

三维重建能够减少大量的模型设计费用，缩短设计周期，能够为社会带来巨大的经济效益。因此，三维重建在社会的生产生活应用中有着诱人的前景，同时兼顾着相当高的研究和应用价值。研究视觉 SLAM 的三维重建既是一件充满挑战又是一件富含趣味的事情。

本文着重研究基于单目 SLAM 系统的三维重建算法，在研究初期，我们考虑到通过一些图像处理的方法结合非稠密视觉里程计系统，为之添加上稠密重建的效果。主要使用的是一个基于半稠密直接法的稠密三维重建，此方案能够得到较为精确的重建效果，能够很好的完成对环境中的弱纹理平面的估计和生成，但是算法稠密重建的前提条件却依赖于相对丰富的点云数据。因此在后续的第二个实验中，我们希望能够通过使用更少的地图点同时结合基于图的分割技术进行环境的稠密三维重建。我们开始关注于基于深度学习的边缘检测去净化提升图像轮廓图的方法，提出了一种将边缘检测图的信息和基于图的分割图进行信息融合的算法，使我们得到的图像轮廓更加精确，这样对于后续的地图点要求就大大减少。同时通过使用最小二乘法估算出鲁棒的三维平面，其次通过字典相似性计算排除冗余平面，使最终得到的重建效果更加的优秀。

1.2 国内外现状

相较于人们开始认识图像的历史来说，三维立体视觉的历史则是相对较短的。其中国外开始研究三维重建的时间相对较早。多视图三维重建是三维重建中的一个研究热点。主要通过摄像机从三维模型拍摄一系列图像来恢复三维场景，这个通常被称为从相机运动中恢复结构（SFM[36]），解决了大部分计算机图形学领域中计算机视觉的三维建模问题。这具有非常重要的作用，它能够改善三维模型逼真度和重建实时的大规模和复杂的场景。目前，人们主要通过三种途径获得 3D 模型：第一种方式属于传统的几何造型技术；第二种方法是通过三维扫描设备扫描目标，然后重建它的 3D 模型；第三种方式基于多个图像采取不同的角度，通过运用计算机视觉的理论最终重建出目标对象的三维模型。在上面介绍的三种方法中，传统的几何造型技术是最成熟的。3DMAX，AUTOCAD 和 CREATOR 是最常见的专业软件。使用这种方法可以得到非常精确的 3D 模型，让人们更好地控制光线和纹理，因而它被广泛应用于机械和建筑这些工程领域，以及电影和电视动画等娱乐行业。但是使用这种建模技术需要一个漫长的周期和一些专业的操作人员，对于许多不规则

的自然或人造的对象，与真实的场景的差异仍然会出现。基于深度相机的三维重建也是一个重要分支，深度相机可以分为双目相机，红外相机和 TOF 相机。微软剑桥研究院于 2011 年提出的 KinectFusion 是基于红外相机重建领域的开山大作，通过使用 Kinect 深度相机能够完成对室内环境实时的重建。但是由于基于深度相机会受到红外光线以及距离的限制，如 Kinect 支持的深度探测距离为 0.5 到 5 米，室外的红外光线会对深度探测产生干扰，所以它比较适合于一些室内场景的重建。但是我们从三维重建的实际应用角度来看的话，Kinect 需要外接电源，这对于相机移动是相当不够方便的。对于基于双目相机的算法来说，Kuschk[31]等人提出了使用一个高分辨率双目相机通过变分法对大尺度场景进行三维重建；汪神岳[28]等人提出了一种基于双目立体相机的室外场景三维重建系统，双目相机选用了 stereolabs 公司的二代 ZED 双目立体相机，同时使用并行计算架构 CUDA，很大程度上优化了三维重建的运行效率。TOF 相机利用设备发射出信号，根据遇阻后的回波时间，从而来推算出空间的三维距离。Kinect2 就是 TOF 相机中的典型代表，有很多的算法基于它进行三维环境的稠密重建[29][30]。随着设备小型化和移动化，如何单纯的使用单目相机来实现场景的实时三维重建也受到了越来越多的关注和研究，基于普通单目相机的三维稠密重建系统也变得越来越流行。这里面比较著名的算法系统有 DPPTAM[3]，它的一个主要贡献就是在采用 LSD_SLAM[2]系统的基础上，添加了额外的并行线程用分割的室内平面场景超像素执行稠密重建；REMODE[32]，解决了单个相机移动时候稠密的深度估计，提出了一个同时结合了贝叶斯估计和在图像处理方面的凸优化理论来生成的深度图估计方法。

国内在三维重建方向上的研究起步较晚，中科院自动化研究所机器人视觉研究组的雷成博士实现了 CVSuite 软件，该软件具有特征点的提取和匹配、摄像机的自动标定、模型的三维显示功能。该软件在使用上十分的方便，而且能够处理来自来源不同的二维图像，但是同时它的缺点也比较明显，首先，因为使用了 Kruppa 方程[33]进行相机的自标定，所以它需要的图像对的数目为三幅，它所使用的其他方法和原理都是基于双目立体视觉，能够得到较为不错的三维模型，但是该算法在实际的匹配过程中存在速度较慢，效率不高的问题。2015 年上线的 Altizure[34]由香港科技大学发明，旨在为中小企业和个人用户提供三维重建服务和为大公司提供图像处理服务。它通过计算机视觉技术来识别图片中的信息，加以人工智能以及深度学习技术来实现二维图像到三维模型的建立工作，取得了不错三维重建效果。

1.3 三维重建框架的介绍

1.3.1 基于直接法的三维重建系统介绍

基于直接法的三维重建系统与基于特征点法的三维重建系统不同，基于特征点的重建系统的耗时主要集中在特征点的提取和匹配上，而直接法根据图像的像素灰度信息来计算相机运动，从而避免了大量特征点匹配的计算，提高了系统运行速度。基于直接法的三维重建系统是最近几年才逐渐开始崭露头角。

直接法中不同于特征点法中需要知道每一个点与点的对应关系，而是通过优化光度误差来求解。DSO[1]是近期效果最不错的一个典型案例，来自于慕尼黑理工大学的 Jakob Engel。它的鲁棒性、精确度以及计算速度都超过了 LSD-SLAM[2]和 ORB-SLAM2[24]。本文使用 DSO 作为视觉里程计前端，结合图优化分割算法确定图像边缘信息，通过边缘信息与里程计中的半稠密点云信息的位置约束估算出一个鲁棒的平面，再通过稠密建图算法实现环境的稠密重建，以下是本文基于直接法的三维重建系统的流程图：

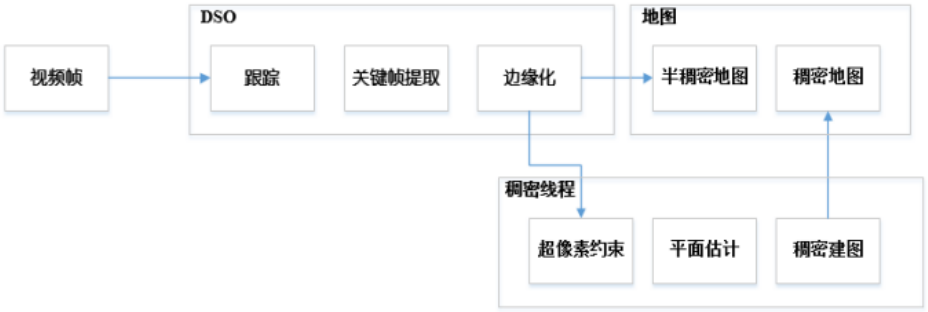

图 1-1 基于直接法的三维重建系统流程图

本系统主要包含了三大模块，分别是：DSO 模块，稠密线程模块，地图模块。

DSO 模块主要由视频帧跟踪、关键帧提取和边缘化构成。基于直接法的视频帧跟踪不需要通过特征匹配点来实现。对于每一个三维点，从某个主导帧出发，乘上深度值之后投影至另一个目标帧，从而建立一个投影残差。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。同时在 DSO 中，每个投影点，除了自身的位置之外，一共提供八维残差，这是为了更好利用该点的信息。这八维残差由该点与周围几个点组成的 Pattern 定义。Pattern 的八维位置表示如下图所示：

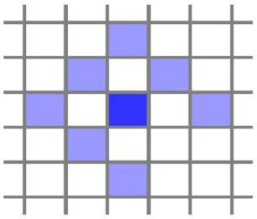

图 1-2　DSO 八维残差 Pattern

选八个点是为了方便 SSE（Super Strong Erection）。在 DSO 的直接法中假设了这八个点在不同图像中保持灰度不变。并且，它们在优化中共享中间点的深度，所以也不能简单地看成八个独立的点。这样通过优化最后 7 个滑动窗口内的所有的残差，使得残差最小求得相机的位姿。此外系统的边缘化步骤遵循以下几个准则：如果一个点已经不在相机视野内被观察到，那么这个点将会被边缘化；如果滑动窗口内的关键帧数量已经超过了 7 个，那么选择其中一个关键帧进行边缘化；当某个关键帧被边缘化时，以该关键帧为主导的地图点将从整个地图中被移除，同时不再参与以后的计算。

地图模块主要负责半稠密点云和稠密点云的显示，其中半稠密的点云由上述 DSO 产生，稠密的点云通过稠密线程生成，最终将两者的点云合并现实得到稠密点云效果。

同时，本系统在当前 DSO 框架的基础之上，我们额外构建了一个稠密重建线程，通过视觉里程计提供的半稠密地图、相机位姿以及基于图的分割图重建出稠密的环境效果；与此同时，我们还设计了一些为稠密重建需要的数据做转换的步骤，包括：坐标系的转换和统一、对半稠密点云的选择策略以及一个可以满足半稠密线程和稠密线程之间的数据交换容器。DSO 中每当一个关键帧生成的时候会产生四种类型的特征点数据，分别是初始不成熟的点，hessian 点，边缘化后的点和 hessian 异常点，假如我们将所有的特征点都传入稠密线程进行计算的话，会造成巨大的误差。因此我们的半稠密点云选择策略将排除异常的误差点，使传递进入稠密线程的半稠密点云更加的精确。此外，我们考虑到半稠密点云线程运行的效率高于稠密点云线程，这样数据的传递需要用容器兼容，不然会造成数据的丢失。我们为此设计的容器包含了半稠密点云以及图像等，使不对等效率的线程能够共用同样的数据。通过以上步骤从而实现将半稠密点云转换成稠密的点云，实现真实环境的三维稠密重建。

1.3.2　基于特征点法的三维重建系统介绍

基于特征点法的三维重建系统主要依赖于对单目相机提供的图像进行特征点的提取和特征点的相互匹配来计算相机位姿并优化出地图点，同时结合基于深度学习得到的边缘检

测图、稀疏的地图点和图优化的分割图像得到环境的稠密重建结果图。

此系统在原先的 ORB-SLAM2 系统中跟踪、建图、重定位、闭环检测四个部分组成的流程框架中额外增加了一个稠密重建线程，其中新增加的线程中包含了边缘检测算法与图优化分割算法以及两者结合的融合算法，最后根据稠密点云的融合算法实现了稠密的重建结果。下图为本文中基于特征点法的三维重建系统的流程图：

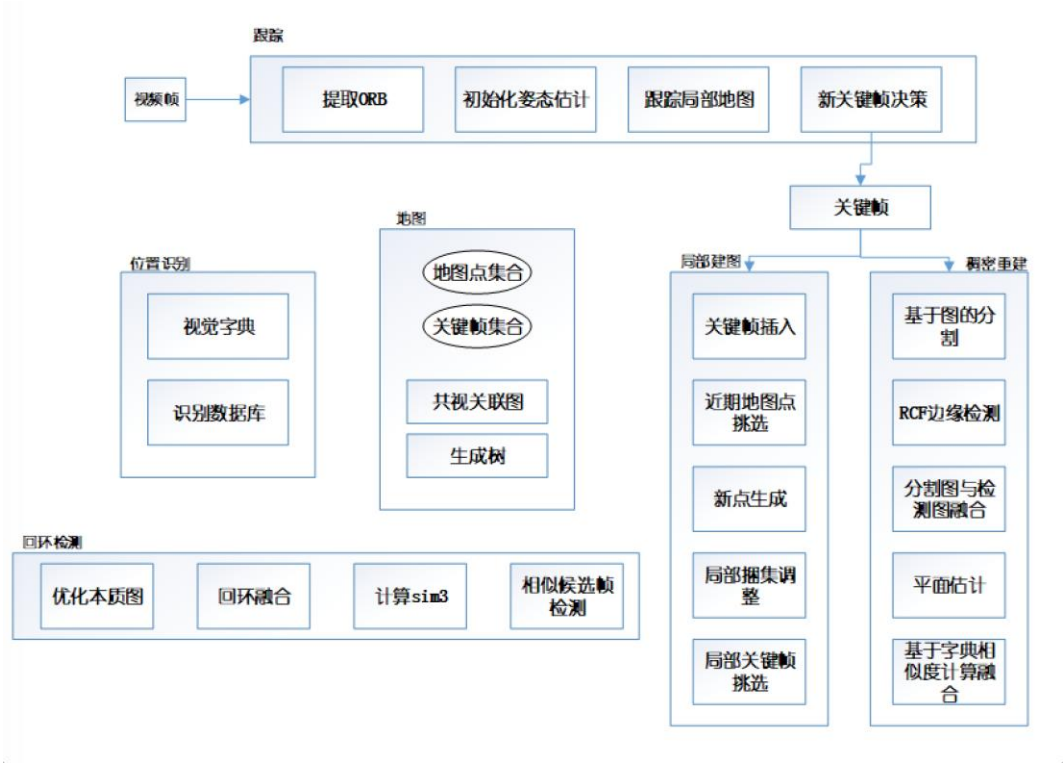

图 1-3　基于特征点法的三维重建系统流程图

跟踪模块主要由 ORB 特征提取，初始化位姿，跟踪局部地图，新关键帧生成这四部分组成；关键帧插入、近期地图点挑选、新地图点生成、局部捆集调整以及局部关键帧挑选这五个部分组成了局部建图模块；回环检测模块包含了优化本质图、回环融合、计算 sim3 以及相似候选帧检测；视觉词典和识别数据库组成重定位的核心部分；稠密重建模块则包括了图优化分割和 RCF 边缘检测的结果融合算法和基于字典相似度计算的稠密点云融合算法。系统内部主要由四个并行的线程组成。

一、跟踪线程：此线程主要的工作内容是在当前帧和地图之间寻找更多的对应关系，来优化相机位姿。首先成功跟踪前面帧后，通过一个假设的匀速运动模型来初始化位姿，这样上一帧的位姿可以被用来估计当前帧的位姿，当运动模式能匹配到的点数比较少时，会启用关键帧匹配模式，此时会拿当前帧与最近关键帧匹配，用词袋模型加速匹配，利用

特征之间的匹配来计算出相机位姿。但是当上一种匹配策略也失败的时候，需要通过重定位方式来继续跟踪，通过和前面所有的关键帧进行匹配来找到是否有合适的位置，当存在足够多的特征点的时候，利用 RANSAC[27]迭代再使用 PnP 求解位姿。此外跟踪线程还负责关键帧的生成，选择关键帧的标准有四个原则：在上一个进行过的全局重定位后，系统又运行了 20 帧图像；当局部建图闲置的时候或在上一个关键帧插入后又过了 20 帧；当前帧跟踪到的点数量大于 50 个；当前帧跟踪到的点比参考关键帧少 90%。合适的插入关键帧以后系统就更为的鲁棒，同时也为后面操作提供源数据。

二、局部建图线程：局部建图主要负责三个部分，分别是插入关键帧，剔除冗余地图点和关键帧以及捆集调整（Bundle Adjustment[35]）。根据前面跟踪线程得到的关键帧作为新的节点插入到关系图（Covisibility Graph）中，同时那些与当前帧共享地图点的关键帧节点所相连接的边也将会被更新。更新关键帧的生长树以及计算关键帧的词袋（Bag Of Words[26]）。当我们判断一个地图点是否被插入到整个地图中需要两个条件，分别是：可预测的观察到这个点的所有关键帧中的关键帧数量必须超过 25%；在单目 SLAM 中，这个构建出来的地图点必须被超过两个关键帧观察到。随后假如一个地图点已经被创建起来了，但当少于 3 个关键帧能观察到这个点的话，此点将从地图中被删除。对于关键帧来说，随着系统的运行关键帧的数目将不断的增加，因此当此帧关键帧内部有 90%的点能够被三个关键帧观察到的话，这个关键帧将会被认为是冗余的。

三、回环检测线程：通过执行位姿图的优化来更正累计漂移误差。在跟踪线程中仅仅考虑到图像在相邻时间上的关联，这样产生的误差将不断的积累，导致无法构建出一个全局一致的相机轨迹和地图。因此，在位姿优化之后，会启动第四个线程来执行全局捆集调整算法，来计算整个系统最优结构和运动的结果。系统的重定位模块由基于 DBoW2[26] 嵌入式位置识别模型组成，仅仅依靠特征的匹配会相当的费时，同时环境中还会存在着光照变化时特征对图像描述的不确定性。词袋通过构建描述图像的特征种类（单词）代替了特征匹配，结合相似度计算函数区分图像的类别。

四、融合稀疏点和边缘检测的稠密重建线程：在本文设计的稠密重建线程中，为了降低对系统地图点数量的要求，不依赖复杂又数量繁多的半稠密点云数据，因此我们考虑通过优化轮廓图的方式来实现我们的目的，采取的策略是把图优化和基于深度学习的边缘检测（RCF）的效果融合，RCF[11]发表于 2017 年的 CVPR，通过深度学习的方式实现对图像的边缘检测，同时 RCF 能在端到端图像处理上达到每秒至少 8 帧的速度，因此用 RCF 的边缘检测结果去优化图优化的分割结果是一种非常不错的尝试。这样在提高了边缘轮廓的效果以后，我们对系统地图点的需求量会大大的降低，从而实现用 ORB 稀疏点估计鲁

棒平面的方案。基于 ORB 稀疏地图的鲁棒平面恢复可以通过最小二乘算法来估计，每个轮廓对应的 ORB 稀疏点可能会存在多个，因此需要求解一个超定方程，最小二乘法通过估计每个点的方差距离实现最优化平面生成。同时我们将这个误差值作为该对应编号平面的属性之一为后续做冗余平面剔除提供支持。此外我们的重建线程会对 ORB_SLAM2 系统提供的关键帧做处理，这样对于关键帧之间拥有公共平面的区域部分必须得做处理，不然会导致系统在三维重建过程有冗余的地图点，使系统的三维重建结果没有达到预期的效果。本文通过计算词典的相似度来区分平面是否为同一个平面，避免对同一个平面的重复重建，使误差达到最小。在处理图像轮廓的时候我们通过图优化的分割图将轮廓编号与分割出来的超像素区域绑定，同时通过稀疏点与已知轮廓的位置约束关系，将轮廓编号与稀疏 ORB 特征点进行绑定，这样我们在后续需要判定不同平面的时候，可以通过其绑定的特征点的属性对其进行区分，而基于词典的相似度计算又是区分特征点相似度中一个经典的算法，通过建立起词带模型，每次通过相似度计算两个编号之间的单词，当存在两者为同一个单词描述时，即是当前两个平面是属于同一个，此后通过判断两个平面的误差大小来选择平面的去留，实现冗余平面的删除。

1.4 论文的组织结构

本文总共分五章，其主要的组织结构安排如下：

第一章 绪论。介绍了本课题的研究背景、研究意义、研究现状，介绍了 SLAM 技术的发展和规划以及目前的研究热点。后续罗列出了基于两大视觉里程计的三维重建框架，即基于特征点法的三维重建系统和基于直接法的三维重建系统，介绍了系统中的具体流程和每一步骤中的重要知识点，并融入了个人的创新想法。

第二章 基于视觉里程计的三维重建理论基础。主要介绍一些关于三维重建的基础内容，包括三维空间的刚体运动，图像特征点的提取与匹配，基于特征法视觉里程计的算法基础。

第三章 基于直接法的稠密三维重建。本章是本文的重点之一。主要分四部分来介绍系统：全局相机的光度标定、视觉里程计前端、数据桥接以及稠密三维重建后端。视觉里程计前端介绍了视频帧的跟踪，关键帧的提取以及边缘化的生成。在数据桥接中使用适当的坐标系转换和数据交换容器，使得视觉里程计的数据能够更好的兼容稠密重建模块，同时我们通过筛选排除了不准确的三维点，使得稠密建图效果更加准确。在图分割算法和位

置约束判断之后进行鲁棒的平面估计，最后结合半稠密的视觉里程计的点图和环境平面的重建实现稠密的三维重建，系统与当前先进的稠密重建算法进行了对比，实验结果显示我们的算法在跟踪和建图上都达到了不错的效果。

第四章 基于稀疏点特征法的稠密三维重建。这个是本文的另一个重点。总共介绍了两个算法，分别是：基于图的分割图与 RCF 边缘检测图的融合算法，基于稀疏特征点的平面估计融合算法。它包括了 RCF 边缘检测算法、边缘检测图与图优化分割图融合策略、ORB 稀疏特征点、基于最小二乘的平面估计以及稠密点云基于字典的相似度计算融合算法。我们使用 RCF 对关键帧图像的边缘检测效果与基于图的分割图进行权重融合，得到更为精确的轮廓图，借助更加精确的轮廓图以及使用更少的特征点信息进行稠密重建是本章的创新点。最后本章系统在开放的公共的数据集上进行了一些测试，并分析和总结了重建效果。

第五章 总结与展望。总结了本文的工作，并为后续工作进行了展望。

第 2 章　三维重建理论基础

2.1　三维空间刚体运动

2.1.1　向量和坐标系

向量的几何意义表示的是一个线程空间中的元素，从一个点指向一个地方的箭头。向量也经常与坐标系互相联系，当我们确定了空间中的一个坐标系的时候，我们才能确定这个向量的坐标位置，也就是能找到该向量的实数对应值。我们假设一个线性空间的基**错误!未找到引用源。**，那么向量**错误!未找到引用源。**在当前坐标系下的坐标可以表示成：

$$a = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_1\boldsymbol{e}_1 + a_2\boldsymbol{e}_2 + a_3\boldsymbol{e}_3 \tag{1}$$

空间下坐标的具体取值跟坐标系的选取和向量的大小都有关系，坐标系通常由三个正交的坐标轴构成。同时根据定义的方式不一样，如图 2-1 所示，坐标系又分为左手系和右手系。当我们在右手系下，通过已知的**错误!未找到引用源。**，**错误!未找到引用源。**坐标轴，通过右手法则由**错误!未找到引用源。**得到坐标轴**错误!未找到引用源。**。向量之间的代数运算包括数乘、加法、减法、内积、外积都会运用在我们的 SLAM 系统中。

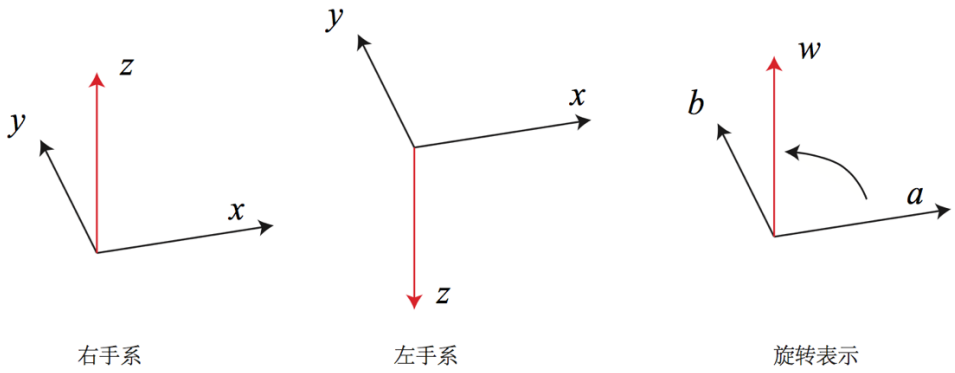

图 2-1　向量、坐标系

2.1.2 欧式变换

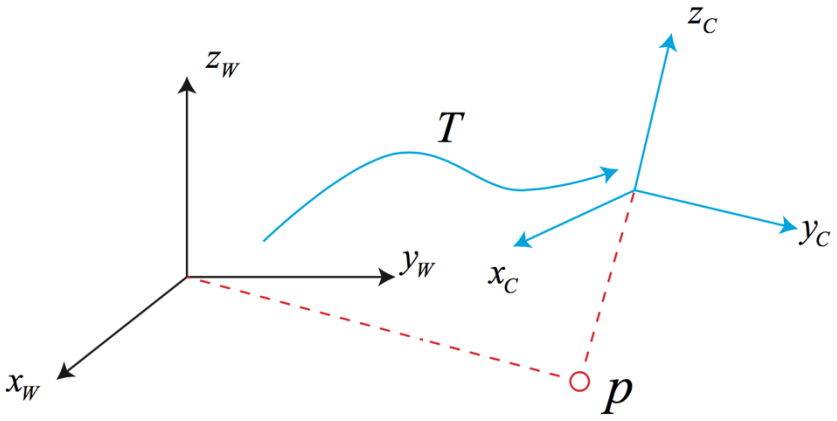

图 2-2 欧式变换

现实生活中，相机的运动是一个刚体运动，在运动过程中始终能保持向量的长度和夹角没有发生变化。一个欧式变换由旋转和平移两部分构成，对于旋转矩阵来说，我们假设一组正交基**错误!未找到引用源。**经过旋转变化得到一组新的正交基**错误!未找到引用源。**，对同一个没有随着坐标系旋转而发生运动的向量 **a** 来说，它在这两个坐标系下的坐标分别为**错误!未找到引用源。**，**错误!未找到引用源。**，可以得到：

$$[\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = [\boldsymbol{e}'_1, \boldsymbol{e}'_2, \boldsymbol{e}'_3]\begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} \tag{2}$$

化简上述式子得：

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_1^T\boldsymbol{e}'_1 & \boldsymbol{e}_1^T\boldsymbol{e}'_2 & \boldsymbol{e}_1^T\boldsymbol{e}'_3 \\ \boldsymbol{e}_2^T\boldsymbol{e}'_1 & \boldsymbol{e}_2^T\boldsymbol{e}'_2 & \boldsymbol{e}_2^T\boldsymbol{e}'_3 \\ \boldsymbol{e}_3^T\boldsymbol{e}'_1 & \boldsymbol{e}_3^T\boldsymbol{e}'_2 & \boldsymbol{e}_3^T\boldsymbol{e}'_3 \end{bmatrix}\begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} \triangleq \mathbf{R}\boldsymbol{a}' \tag{3}$$

将这两个正交基的内积矩阵称为旋转矩阵，它是一个行列式值为 1 的正交矩阵，同时加上平移矩阵，有：

$$\boldsymbol{a}' = \boldsymbol{R}\boldsymbol{a} + \boldsymbol{t} \tag{4}$$

除了欧式变换还有其他三种变换，分别为：相似变换、仿射变换、射影变换。相似变换相较于欧式变换多了一个缩放自由度，由于存在缩放比例，相似变换将不再保持图形的面积不变。仿射变换相较于欧式变换来说对于旋转矩阵不必要求是一个正交矩阵，经过仿射变换以后正方形就变成了平行四边形。射影变换是一个最一般的变换，2D 的射影变换一共有 8 个自由度，3D 则共有 15 个自由度。从真实世界到相机的照片就是一个射影变换。

2.2 特征提取与匹配

2.2.1 特征点

图像分析与图像识别的大前提是事先对图像完成特征提取，对于一副普通的三通道图像来说，计算机内部存储的是图像的二进制信息，直接使用这些信息会使我们很难从中获取出有用的图像信息，那么当我们需要使用这些高维图像信息的时候，我们必须从这些二进制数据提取出图像中的关键信息，一些基本元件或它们之间的相互关系。

特征点是全局高维图像信息很好的局部表达，它反映的是图像上一些具有局部区域或特征的性质，因此特征点经常用在图像匹配，检索等应用中，而对于一些深层的图像理解则不怎么适合。深层的图像理解则需要关心一些图像的全局特征，如颜色分布，纹理特征，主要物体的形状等。光照，旋转，噪声等不利因素都会影响到全局特征，因此全局特征是十分容易受到外界环境的干扰。相比而言，局部特征点，往往对应着图像中的一些线条交叉，明暗变化的结构中，受到的干扰也相对少，具有一定的鲁棒性。

局部特征点包括了斑点和角点。对于周围有着颜色和灰度巨大差别的区域我们用斑点来描述，如在卧室内一把纯红色的椅子或是足球场草坪上的足球门框。斑点表示的是一个集中区域，因此在抗噪能力上比较强，稳定性也相对较好。而角点则是图像中一个物体的拐角或者线条之间的交叉部分。Harris 角点是一种经典的角点检测算法，通过计算图像灰度的一阶导数矩阵实现。检测器的主要思想计算在某个局部窗口内图像块与在各个方向微小移动后的窗口内图像块的相似性。基于加速分割测试的 FAST 算法可以快速地提取出角点特征。该算法以一个像素点 p 为圆心，在给定阈值的条件下，半径为 3 个像素的离散化 Bresenllam 圆周上，如果在圆周上存在 n 个连续的像素灰度值大于或小于该阈值，则为角点。此外可以使用一种快速的方法来完成 FAST 角点检测，并不需要把圆周上的所有点都比较一遍。首先比较上下左右四个点的像素值关系，至少要有 3 个点的像素灰度值大于或小于阈值，则 p 为候选点，然后再进一步进行完整的判断。使用基于机器学习的 ID3 贪心算法来构建决策树同样可以达到加速角点的检测速度。

而对于一般的直接法系统来说，因为在直接法中，提取的是图像中的高梯度点，对于图像来说，梯度表示的是当前位置点与邻域间的差异程度。假如把图像看成一个二维的离散函数的话，梯度的数学含义就是对这个离散函数进行求导运算，而在一些常见的图像处理中，图像的边缘信息经常是通过对图像的梯度运算后得到，因此很多的高梯度的位置往往伴随的是图像十分有区分性的一些边缘或巨大的像素变化位置。

2.2.2 特征匹配

特征点匹配就是寻找不同图像上由同一个位置点投影而成的特征点对。根据匹配条件不同，匹配算法在目前基本可以分为以下两大类：基于窗口的匹配，窗口表示了当前像素位置附近的二维矩阵的灰度值，这其中最常见的是通过交叉相关性来进行特征匹配，同时这也是目前大多数匹配算法的基础；基于特征的匹配，在匹配前先要在图像中抽取边或区域等特征。这些特征可以看作是图像内容的抽象描述，在不同的光照下具有更多的不变性。但是特征匹配往往有很高的计算代价。除此之外，各种匹配方法所采用的优化算法也不尽相同，有的使用全局优化算法，如动态规划法、穷举法、凸规划法和松弛法等；还有的使用一些非全局最优算法，如贪婪算法，模拟退火算法和随机搜索算法等。以上的大多数方法都隐含地引入了一些约束，如动态规划法就需要顺序不变约束。当这些约束不满足时，相应的方法就无法使用。Maciel 等使用线性规划的方法来解决匹配中产生的歧义问题，这种方法可以得到某种意义上的全局最优解，并且由于线性规划法已经很成熟。因此可以保证算法有较高的效率，但是线性规划法需要消耗大量的内存并且要预先估计正确的匹配数。

基于特征点匹配方法的主要研究课题是如何建立特征点之间的对应关系，虽然特征点的匹配技术起步还是相对较晚，但是因为它迫切的应用需求和广泛的前景使它在计算机视觉领域中引起了众多学者们的关注，并且在最近的十几年里得到了充分的发展，并出现了大量的研究特征匹配的方法。不同的图像变换模型又具有不同的匹配方法，如在一些应用领域中，根据已有的先验知识可以直接计算出缩放比例，如在地球资源卫星图像的配准中，可以直接利用给定图像的比例信息，只需要考虑在一些区域被遮挡的情况下存在的平移与旋转。这时相当于提供了一个不变量，即一幅图像中任意两个特征点的距离与另一幅图像对应的特征点之间的距离相等。这样使匹配算法能够简化为只考虑有平移、旋转的情况下求解点集之间的对应关系，但在很多实际应用中，由于存在图像噪声及视场变换还有不同时间和不同传感器获取的图像等因素，待匹配的两幅图像中不可避免出现虚假点、丢失点以及非刚性形变。

基于特征点匹配方法主要分为如下几类：1)直接基于特征点属性的匹配方法(基于描述符的匹配方法)。这类方法在提取特征点后，对特征点进行不同的描述，用来区别其它特征点，然后用描述符进行匹配。利用特征描述符进行匹配的方法是在特征点提取过程中，不仅得到特征点的位置，同时将得到特征点在其他仿射变换下不变的特征点描述。如经典的 SIFT[39]描述符将特征点周围的 16x16 窗口分割成 16 个 4x4 的子窗口，然后统计每个子窗

口的方向梯度直方图。将每个子窗口的方向分成 8 个方向计算。一共具有 4x4 个子窗口，每个窗口描述是 8 位的，描述 8 个方向的梯度的大小值，那么这样形成的描述符是 844128 维，然后在匹配时利用描述符进行配对。另外，ORB[19]特征匹配中经常使用的是 BRIEF[22]描述子，BRIEF 描述子通过计算二进制串之间计算汉明距离来进行匹配，这样做极大减少了特征点的存储压力，因此被广泛运用在实时系统的特征匹配中。2)基于特征点几何结构的匹配方法。这类方法是利用特征点之间稳定并且相似的几何结构进行匹配，而不是直接对特征点逐一进行匹配，如图像中的边或三角形等信息来等进行匹配。

近年来还出现了许多特征点匹配方法，Skea D 提出了累加器算法[12]，算法的框架主要通过平面点模式匹配实现，对噪声、缺少点及伪点来说，该算法还是较为鲁棒的，但是不足的是其计算复杂度相对较大。Ranade 和 Rosenfeld 等人通过使用松弛法来进行点的匹配[36]，而陈志刚等在前面的基础上通过构建出一个点之间的三角形关系并通过判定其相似度来代替点特征的匹配度，这样提出的算法就具有了比例与旋转不变特性的特点[13]。Chang 等人利用二维聚类进行匹配[14]，张立华等利用不可约矩阵和相对不变量理论提出了几种点模式匹配新算法[15]，它们可分别用来解决相似变换和仿射变换下具有相同点数的两个点模式的匹配问题。Li 利用了几何不变量来进行点模式匹配[37]，Spirkovska 和 Reid 利用了高阶神经网络进行匹配[17]。

基于图像特征的匹配方法可以克服利用图像灰度信息进行匹配的缺点，由于图像的特征点是提取的局部信息，因此在图像中的数量又是非常少，这样就大大减少了特征点在匹配过程的计算量；而且特征点的提取过程中经常考虑了光照不变性和尺度不变性，减少了噪声的影响，对灰度突变，图像的形变以及图像中物体被遮挡等都有十分不错的适应能力。

2.3 基于特征法视觉里程计的算法基础

2.3.1 位姿估计

位姿估计是视觉里程计中最为重要的一部分，很大程度上决定了系统的精度。根据求解位姿数据源的不同大致可以分为三类：

一、二维点之间的对极几何转换关系

对极约束：对极约束普遍存在于单目的运动相机和双目视觉中，类似我们的人眼一样通过不同的视角观察到同一个物体，然后通过匹配得到同一个图像点在不同相机中的偏差，再根据三角测量计算出三维信息。

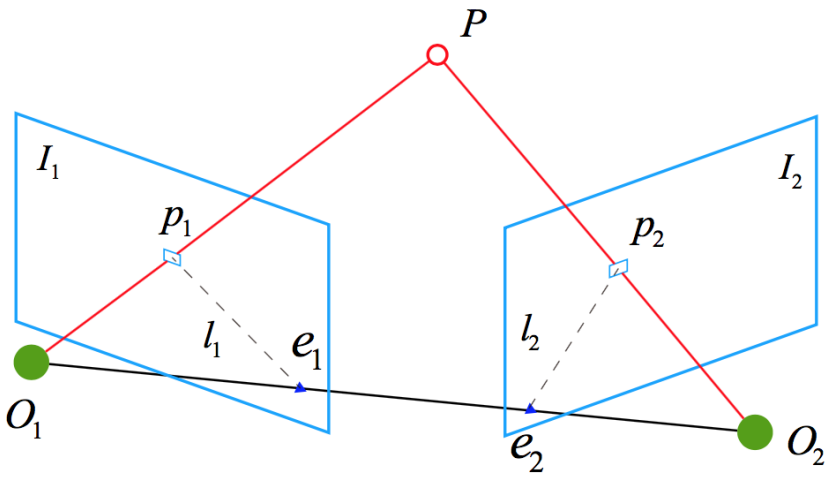

图 2-1 对极几何

对极约束刻画了两个有重叠区域的图像之间的几何关系，它只与相机本身的内参和相机的相对位置关系有关，与拍摄的场景无关，被常用来估算两帧之间的相机运动。如上图所示，**错误!未找到引用源。**，**错误!未找到引用源。**分别代表两个相机的中心，**错误!未找到引用源。**之间的直线称为基线，P 为空间中的一个三维点，**错误!未找到引用源。**，**错误!未找到引用源。**分别为**错误!未找到引用源。**点与左视图和右视图在成像平面上的交点，**错误!未找到引用源。**连线与左右视图成像平面的交点为**错误!未找到引用源。**，**错误!未找到引用源。**，分别被称为左极点和右极点。这个时候**错误!未找到引用源。**，**错误!未找到引用源。**，**错误!未找到引用源。**构成一个平面，称为极平面。而极平面与两个成像平面**错误!未找到引用源。**，**错误!未找到引用源。**之间的相交线**错误!未找到引用源。**，**错误!未找到引用源。**叫做极线。这样已知点**错误!未找到引用源。**或**错误!未找到引用源。**可知其对应点一定在对极线上，这种几何关系称为对极约束。

对极约束的代数公式可以由如下推导出来。假设三维点 $P=[X,Y,Z]^T$，根据相机小孔成像模型，通过相机内参矩阵 K 以及两个坐标系的相机运动 R，T，用齐次坐标表示，我们可以得到像素点**错误!未找到引用源。**，**错误!未找到引用源。**的像素位置为：

错误!未找到引用源。 （5）

定义**错误!未找到引用源。**，**错误!未找到引用源。**为两个像素点的归一化平面上的坐标。代入上式，得:

$$x_1 = K^{-1}p_1,\ x_2 = K^{-1}p_2,\ x_2 = Rx_1 + t \quad (6)$$

两边同时对**错误!未找到引用源。**做外积，并同时左乘**错误!未找到引用源。**，整理以后重新

带入**错误!未找到引用源。**，**错误!未找到引用源。**得到：

$$p_2^T K^{-T} t^{\wedge} R K^{-1} P_1 = 0 \tag{7}$$

这个式子称为对极约束，它的几何意义代表了**错误!未找到引用源。**，**错误!未找到引用源。**，**错误!未找到引用源。**三点共面。中间部分可以记作两个矩阵，分别是基础矩阵(Fundamental Matrix)和本质矩阵(Essential Matrix)，这样对极约束可以简化为两个更简单的式子：

$$\mathbf{E} = t^{\wedge} R,\ \mathbf{F} = K^{-T} E K^{-1},\ x_2^T E x_1 = P_2^T F p_1 = 0 \tag{8}$$

在三维重建中以及相机的初始标定中，可以利用特征点的匹配关系计算出 F，E，进一步计算出相机的 R，T。此外，在直接法的 SLAM 中，已知相机姿态的情况下，也通过极线搜索进行特征点匹配，减少了很多计算量极大提高了效率。

本质矩阵：本质矩阵**错误!未找到引用源。**，是一个 3*3 的矩阵，因为平移和旋转各有三个自由度并且由于尺度等价性，其实**错误!未找到引用源。**只有五个自由度。同时**错误!未找到引用源。**的内在非线性性质会对求解五对点的线性方程会带来麻烦，所以一般使用八点法求解。

单应矩阵：单应矩阵描述了两个平面之间的映射关系，当图像中提取的特征点都落在同一个平面上，比如墙或者地面时，那么可以通过两者之间的单应性来估计运动。

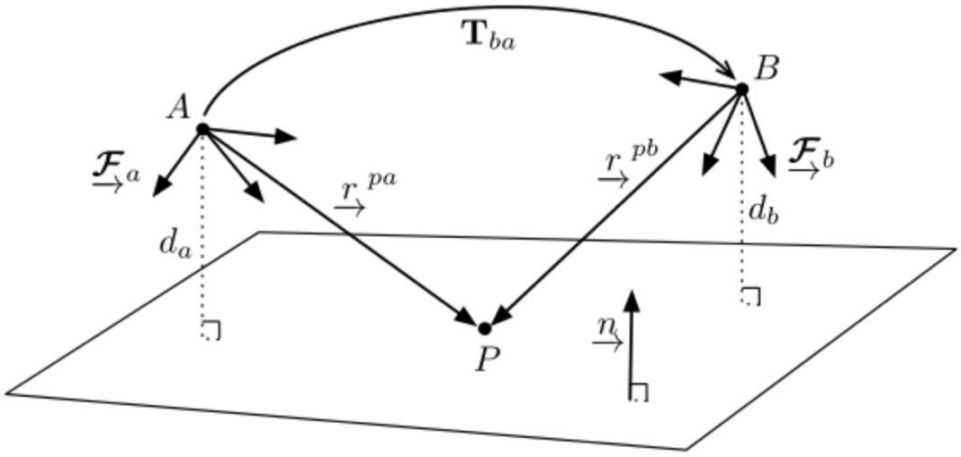

图 2-2　单应矩阵

假设一个三维空间点**错误!未找到引用源。**和三维空间点到像素坐标的齐次转换为：

$$q_i = K_i \frac{1}{z_i} p_i \tag{9}$$

单应矩阵的几何意义是当这个观察的点落在某个平面上的时候，我们可以利用平面方程估算出其深度值。具体的推导为：

当已知相机中心到平面距离以及法向量，可以得出平面的法线方程：

$$\boldsymbol{n}_i^T \boldsymbol{p}_i + d_i = 0 \tag{10}$$

将公式 10 中的三维空间点坐标替换为像素坐标可得：

$$z_i = -\frac{d_i}{\boldsymbol{n}_i^T \boldsymbol{K}_i^{-1} \boldsymbol{q}_i} \tag{11}$$

那么从像素齐次坐标转换到三维空间坐标的公式变为：

$$\boldsymbol{p}_i = -\frac{d_i}{\boldsymbol{n}_i^T \boldsymbol{K}_i^{-1} \boldsymbol{q}_i} \boldsymbol{K}^{-1} \boldsymbol{q}_i \tag{12}$$

再结合空间点在前后两帧的坐标系下的三维坐标约束关系，得到：

$$\boldsymbol{\rho}_b = \boldsymbol{C}_{ba} \boldsymbol{\rho}_a + \boldsymbol{r}_b^{ab} \tag{13}$$

将该式子带入到第一个公式，得到两帧之间同一空间点的像素点坐标的约束关系，经过简化整理得：

$$\boldsymbol{q}_b = \boldsymbol{K}_b \boldsymbol{H}_{ba} \boldsymbol{K}_a^{-1} \boldsymbol{q}_a, \ \boldsymbol{H}_{ba} = \frac{z_a}{z_b} \boldsymbol{C}_{ba} (1 + \frac{1}{d_a} \boldsymbol{r}_a^{ba} \boldsymbol{n}_a^T) \tag{14}$$

其中**错误!未找到引用源。**代表单应矩阵，它反映了两个平面之间的转换关系。

二、三维点与二维点的 PnP 转换关系

PnP 是求解三维点到二维点对的运动方式。在单目 SLAM 中因为没有深度，所以必须先进行过初始化后才能使用 PnP，它比对极约束使用更少的匹配点但是获得较好的相机运动估计。而捆集调整（Bundle Adjustment）将 PnP 问题构建成一个定义与李代数上的非线性最小二乘问题，是一种求解 PnP 中经典的解法。在 PnP 中，捆集调整产生一个最小化重投影误差问题。通过优化该值达到相机运动的估计。构建的重投影误差可以用下图表示：

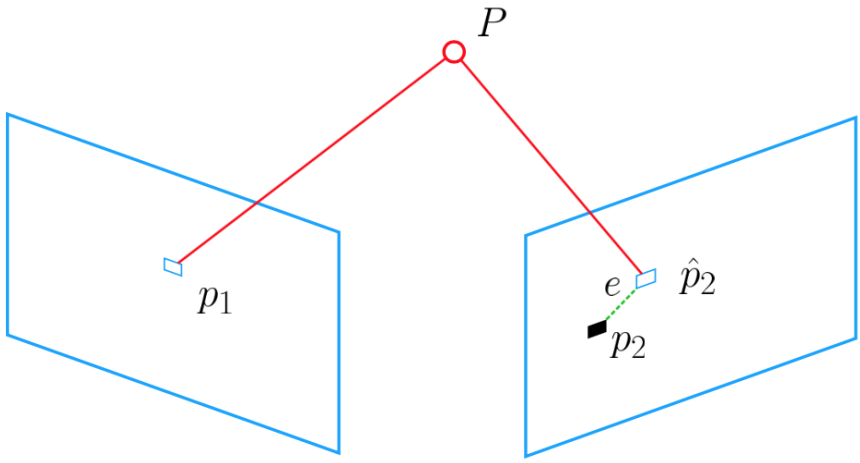

图 2-3　重投影误差示意图

2.3.2　单目深度估计

在单目 SLAM 中深度估计技术主要用极线搜索与块匹配实现。在上一个小节得到相机位姿的基础上，考虑从两个角度一起观察空间的同一个坐标点，根据夹角的大小确定该空间点的距离，示意图如下：

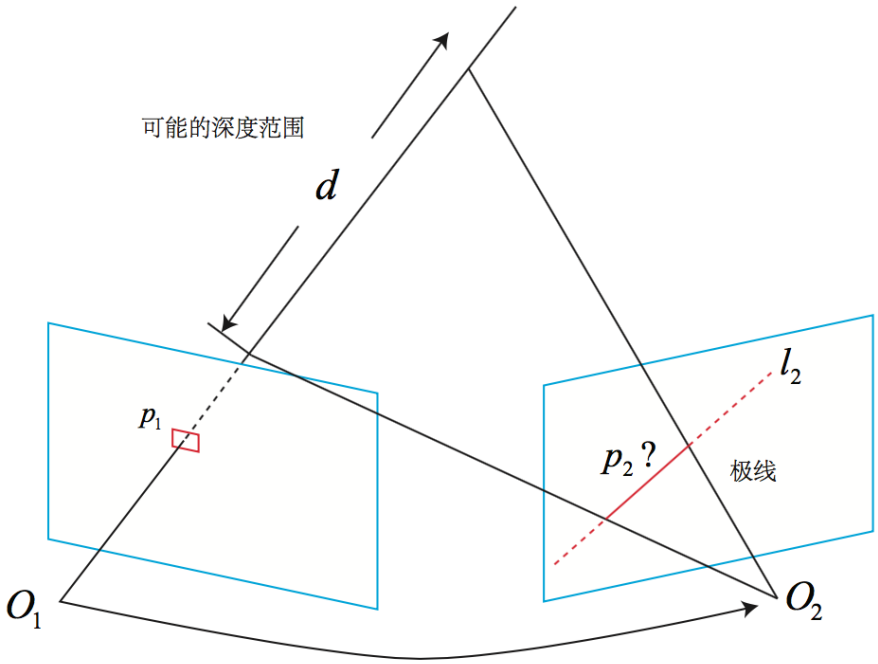

图 2-4　极线搜索示意图

在左侧图像**错误!未找到引用源。**中的特征点**错误!未找到引用源。**，假如我们只考虑这个图像那么我们是无法知道这个特征点的深度，现在我们假设**错误!未找到引用源。**的深度值在实际值的最小值和无穷大之间，设为**错误!未找到引用源。**。这样从右侧图像中看起来这个深度区域对应的空间点在图像**错误!未找到引用源。**上就是一条线段。一种最为直接的思路是将极线上每个点均与**错误!未找到引用源。**匹配，然后这会存在单个像素的亮度区分度很小的问题，直接匹配将会产生很大的误差，此时我们考虑在像素**错误!未找到引用源。**周围选取一个 w*w 的小块区域，然后在对应极线上也选取相同大小的小块进行匹配，即块匹配技术。假设**错误!未找到引用源。**周围的小块为**错误!未找到引用源。**，极线上 N 个小块记成**错误!未找到引用源。**,i=1,…,N，通过计算小块之间的平方差优化出最优的深度值，公式如下：

$$S(\boldsymbol{A},\boldsymbol{B})_{SSD}=\sum(\boldsymbol{A}(i,j)-\boldsymbol{B}(i,j))^2 \qquad (15)$$

虽然通过块匹配技术大致估算出了两个特征点所对应的地图点在三维空间中的坐标，但是估算的精度我们不得而知，也不能确定空间点**错误!未找到引用源。**是否真的是**错误!未找到引用源。**，**错误!未找到引用源。**的深度点。因此我们需要一个策略确定前面估计出的深度值是精确度，评估原理如下图所示：

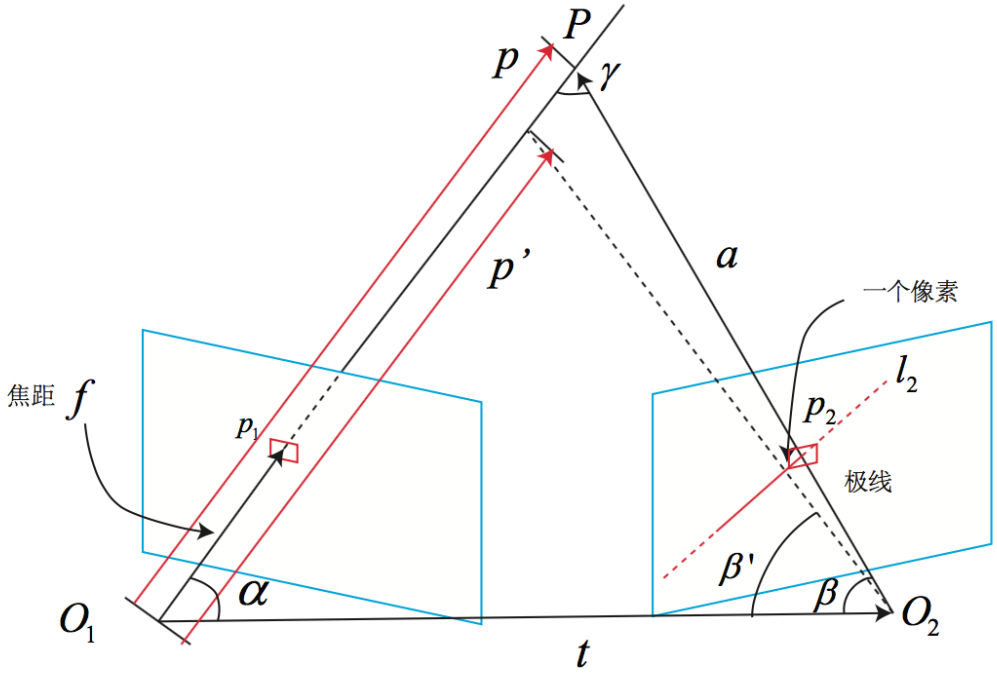

图 2-5　不确定性分析

我们假设像素**错误!未找到引用源。**扰动一个最小的像素单位，这会导致夹角**错误!未找到引用源。**产生一个变化量**错误!未找到引用源。**，考虑到相机焦距**错误!未找到引用源。**，得到：

$$\delta\beta = \arctan\frac{1}{f} \tag{16}$$

结合三角形三个角度之和以及正弦定理可得：

$$\|\boldsymbol{p}'\| = \|\boldsymbol{t}\|\frac{\sin\beta'}{\sin\gamma} \tag{17}$$

这样我们可以得到一个最后的误差像素表示函数：

$$\sigma_{obs} = \|\boldsymbol{p}\| - \|\boldsymbol{p}'\| \tag{18}$$

那么深度不确定性分析可以大致分为以下三个步骤，一：当新的数据产生时，利用极线搜索和块匹配确定投影点位置；二：根据几何关系和深度不确定性分析；三：将观测估计值不断融合，如果深度收敛则停止计算，否则返回第一步骤。

2.4 基于直接法视觉里程计的算法基础

2.4.1 位姿估计

在直接法中并不像特征点法中需要知道每一个点与点的对应关系，而是通过优化光度误差来求得它们。

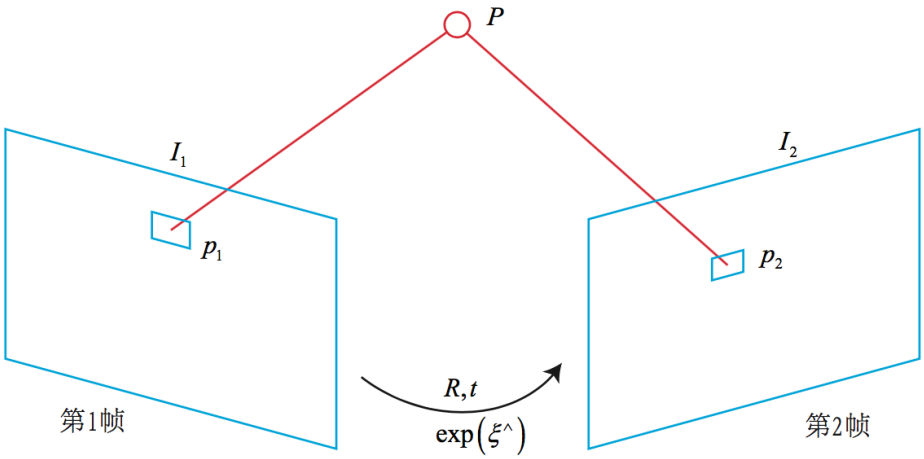

图 2-6 直接法示意图

直接法的目标是求得第 1 帧到第 2 帧之间的相机位姿的相对变换情况。假设以第 1 帧为参考系，从第 1 帧到第 2 帧之间的旋转和平移为 R，t，当这种情况是单目相机的时候这两个帧的相机内参是相同的，设为 K。这样可以得到一个投影方程：

$$\boldsymbol{p}_1 = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_1 = \frac{1}{Z_1}\boldsymbol{KP} \tag{19}$$

$$\boldsymbol{p}_2 = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_2 = \frac{1}{Z_2}\boldsymbol{K}(\boldsymbol{RP}+\boldsymbol{t}) = \frac{1}{Z_2}\boldsymbol{K}(\exp(\xi^\wedge)\boldsymbol{P})_{1:3} \tag{20}$$

错误!未找到引用源。是**错误!未找到引用源。**的深度，**错误!未找到引用源。**是**错误!未找到引用源。**在第 2 帧中的深度。在直接法中是无法确定**错误!未找到引用源。**，**错误!未找到引用源。**的对应关系，直接法会去寻找一个最优的相机姿态估计值去寻找**错误!未找到引用源。**的位置。这样就变成了一个优化问题：通过优化相机位姿来减小**错误!未找到引用源。**，**错误!未找到引用源。**的差别。此时优化的不是特征点法中的重投影误差而是**错误!未找到引用源。**在两个像中的亮度误差，即光度误差**错误!未找到引用源。**。

$$e = \boldsymbol{I}_1(\boldsymbol{p}_1) - \boldsymbol{I}_2(\boldsymbol{p}_2) \tag{21}$$

此优化的前提是**错误!未找到引用源。**的灰度不变假设，空间点**错误!未找到引用源。**在两个不同视角下，成像的灰度是不变的。选取优化目标为光度误差的二范数，为：

$$\min J(\xi) = \sum_{i=1}^{N} e_i^T e_i, \quad e_i = \boldsymbol{I}_1(\boldsymbol{p}_1, i) - \boldsymbol{I}_2(\boldsymbol{p}_2, i) \tag{22}$$

通过给**错误!未找到引用源。**左乘一个小扰动**错误!未找到引用源。**来求解这个优化问题：

$$\mathrm{e}(\xi \oplus \delta\xi) = \boldsymbol{I}_1\left(\frac{1}{Z_1}\boldsymbol{KP}\right) - \boldsymbol{I}_2\left(\frac{1}{Z_2}\boldsymbol{K}exp(\delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\,\boldsymbol{P}\right) \tag{23}$$

$$\approx \boldsymbol{I}_1\left(\frac{1}{Z_1}\boldsymbol{KP}\right) - \boldsymbol{I}_2\left(\frac{1}{Z_2}\boldsymbol{K}(1+\delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\,\boldsymbol{P}\right) \quad \text{错误!未找到引用源。}$$

错误!未找到引用源。

化简得到：

$$\mathrm{e}(\xi \oplus \delta\xi) = \boldsymbol{I}_1\left(\frac{1}{Z_1}\boldsymbol{KP}\right) - \boldsymbol{I}_2\left(\frac{1}{Z_2}\boldsymbol{K}\exp(\boldsymbol{\xi}^\wedge)\,\boldsymbol{P} + \mathbf{u}\right) \tag{24}$$

错误!未找到引用源。

错误!未找到引用源。

链式法则第一项为**错误!未找到引用源。**处的像素梯度，后两项合并可以写成：

$$\frac{\partial \boldsymbol{u}}{\partial \delta \boldsymbol{\xi}} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} & -\frac{f_x XY}{Z^2} & f_x + \frac{f_x X^2}{Z^2} & -\frac{f_x Y}{Z} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} & -f_y - \frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X}{Z} \end{bmatrix} \tag{25}$$

最后此优化问题的雅克比**错误!未找到引用源。**通过高斯牛顿（G-N）或列文博格（L-M）计算出最优的相机位姿。

第 3 章　基于直接法的稠密三维重建

3.1　引言

本章主要介绍基于 Direct Sparse Odometry(DSO)的稠密三维重建技术，其主要内容主要可以归结为通过图像处理技术建立半稠密三维点云和环境中平面的约束进而得到稠密的环境三维点云，经过事先由全局相机光度标定程序计算相机参数的前提下，根据视觉里程计计算出的关键帧并恢复出三维场景中的半稠密点云，最后通过基于图的分割算法建立数据关联，增加点云密度以及通过稠密匹配算法增加三维点云精度。以下将分为五个步骤做介绍：摄像机标定、视觉里程计前端、数据桥接、稠密三维重建后端、实验的结果与分析。主要系统框架图与实例结果如下图所示：

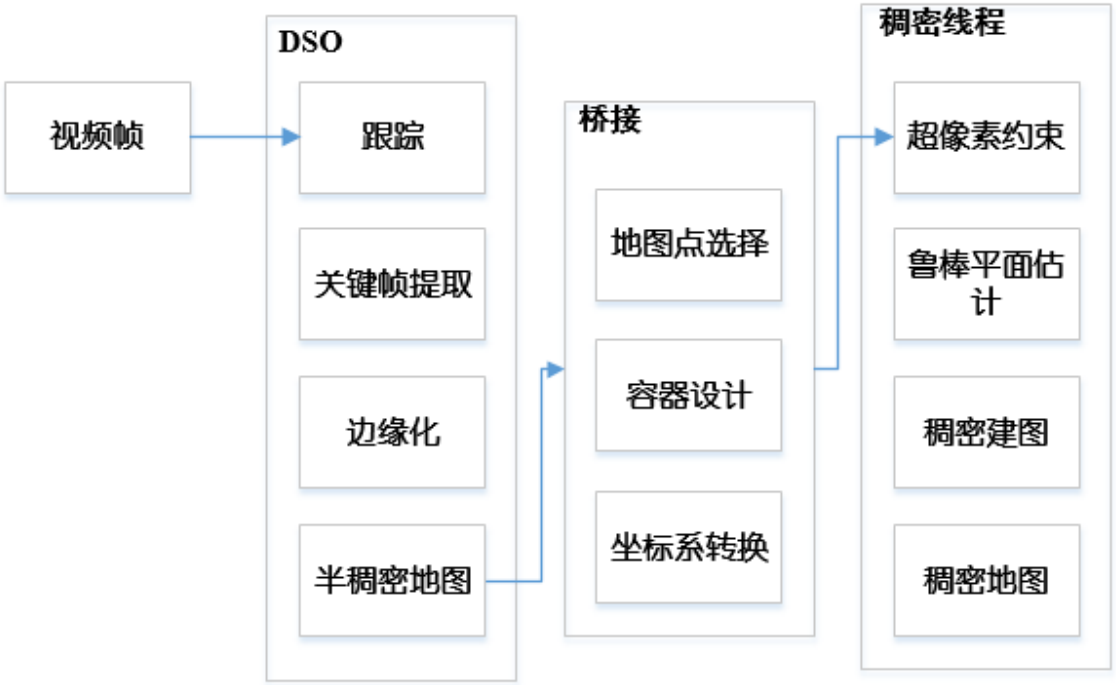

图 3-1　基于 DSO 的稠密三维重建框架

在稠密三维重建中，本文提出的地图点选择策略，数据交换容纳池，点云坐标系转换方法使后续得到的半稠密点云的精度和鲁棒性更高，该方法通过图像分割技术，通过点云

的投影位置和二维图像的位置约束关系产生出估计的环境中的平面，最后使用稠密建图生成稠密点云。

3.2 全局相机光度标定

光度标定策略的设定主要是为了去除在做直接法的时候环境光照对图像信息的干扰。下图是一个相机的成像过程：

图 3-2　相机成像原理

此成像过程中涉及到场景辐射指物体表面单位面积上的能量分布，它与光的方向有关。而辐射度表示的是物体表面单位面积总的入射能量，因此与方向是没有关系的；光学模块除了常见的小孔成像和畸变模型之外，还有存在光学晕影。光学晕影由相机元件相互遮挡引起，导致有效的入射光减少，现实中我们拍摄一个光度均匀的物体会出现图像中心和边缘亮度不一致，看起来就像是光晕一般。快门主要影响的是曝光时间，不同场景中的曝光时间不一致会导致图像产生不一致的灰度值。而传感器则是负责将输入的曝光值输出为亮度值，两者之间的关系称为响应函数，它通常是非线程的，因此光度标定可以用来修正这些非线性误差。使用的图像转换模型的公式如下：

$$I_i(x) = G\left(\mathrm{t}_i V(x) B_i(x)\right) \tag{26}$$

其中，**错误!未找到引用源。**表示曝光时间，每一帧的曝光时间都是不同的；**错误!未找到引用源。**表示不受光学模块影响的辐射度传感器；**错误!未找到引用源。**表示一个返回值为离散型的响应函数；**错误!未找到引用源。**表示光学模块对误差的影响，用一个跟原始图像一样大的权重矩阵来表示光学模块对每个像素的影响。这样可以通过事先标定好的响应函

数，然后使用迭代的方法求解得到**错误!未找到引用源。**，再反解出**错误!未找到引用源。**，再去做直接法会更加的准确。

3.3 视觉里程计前端

本节主要介绍在视频数据流进入后，对每一帧的图像进行跟踪计算出旋转平移矩阵、关键帧提取以及关键帧对应的半稠密点云的生成。在视频帧跟踪过程中，主要使用基于滑动窗口的光度误差优化法进行相机位姿的计算。根据三种关键帧选取策略选择关键帧，随后根据相机位姿计算得到深度点，生成半稠密地图，其中相机位姿的计算可以参考 2.4 小节。

3.3.1 视频帧跟踪

当一个新的帧到来时，我们提取高梯度点，并计算参考帧和目标帧之间的光度误差，这里光度指的是每个高梯度点的像素亮度。误差则表示在一块像素区域上的亮度平方误差。然后我们计算所有帧里面的点的全部光度误差[1]。同时为了获得更精确的相机位姿，我们还保留了一个能够拥有 7 个关键帧的活动窗口，并使用高斯-牛顿算法优化所有滑动窗口中的总光度误差，这样计算的结果会更加的鲁棒和精确。

3.3.2 关键帧提取与边缘化

由于系统运行的资源限制，我们不能保持每一帧视频帧都能够存储到系统中，从每一帧视频帧都拿来更新地图的计算量过于庞大，因此需要对视频帧进行关键帧的提取，以便提高系统运行效率的同时生成清楚可用的地图信息。对于关键帧生成的策略，本文考虑了以下三个选取策略：

$$f \coloneqq \left(\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{p}-\boldsymbol{p}'\|^2\right)^{\frac{1}{2}} \tag{27}$$

式 27 表示了当相机的视场角变化的时候，需要提取关键帧。在初始化时的不精确的跟踪过程中，计算出前一帧关键帧和最近的视频帧之间的平均平方光流值。

$$f_t \coloneqq \left(\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{p}-\boldsymbol{p}'\|^2\right)^{\frac{1}{2}} \tag{28}$$

式 28 表示了当相机的平移运动导致对景物的遮挡和非遮挡，此时场景也发生了巨大的变

化，因此需要提取出更多的关键帧。这种情景需要计算出不带旋转信息的平均光流值，**错误!未找到引用源。**表示去除了相机旋转矩阵后的值。

$$\mathrm{a} := \left|\log\left(e^{a_j - a_i} t_j t_i^{-1}\right)\right| \tag{29}$$

当相机的曝光时间显著增加的时候，公式 29 通过估算出两帧之间的相对亮度因素来决定是否需要提取关键帧。其中**错误!未找到引用源。**表示一个亮度传递函数的参数，**错误!未找到引用源。**表示照片的曝光时间。

综合考虑上述三个选取策略，我们用式子**错误!未找到引用源。**表示最后是否真的要提取出这个关键帧，其中**错误!未找到引用源。**，**错误!未找到引用源。**，**错误!未找到引用源。**分别表示三个选取策略的相对权重关系，当它们的总和大于 1 时，系统认为这个视频帧为关键帧。

同时我们可能不会保留系统中的所有激活状态的关键帧数据，因为这也需要极大的存储资源。我们需要从以前的所有处于激活状态的关键帧变量的分布中获得这些关键帧子集的概率分布，也就是说，需要使用关键帧边缘化。当我们有**错误!未找到引用源。**，**错误!未找到引用源。**…**错误!未找到引用源。**个处于激活状态的关键帧的时候，这其中**错误!未找到引用源。**是最新的关键帧，**错误!未找到引用源。**是最老的关键帧。对于最新的两帧关键帧，我们总是保留它们；同时在**错误!未找到引用源。**中可见点数少于 5%的帧将被边缘化；当超过 7 个关键帧处于激活状态，那么我们将边缘化距离分数最高的那个视频帧，距离分数公式如下：

$$\mathrm{s}(I_i) = \sqrt{d(i,1)} \sum_{j \in [3,n] \setminus \{3\}} (d(i,j) + \epsilon)^{-1} \tag{30}$$

其中**错误!未找到引用源。**表示关键帧**错误!未找到引用源。**和**错误!未找到引用源。**之间的欧几里得距离，**错误!未找到引用源。**表示常量，这个启发式公式的设计可以使处于激活的关键帧在三维空间中保持良好分布，同时还能满足更多关键帧接近最近的关键帧，从而达到边缘化的目的。

3.3.3 半稠密地图

DSO 通过一个候选点激活策略来稳定所有活动帧中的所有特征点数。然后它在系统中生成了四种不同类型的点（见 1.3.1 小节）。此外，随着系统的运行一些候选点需要被激活变成活跃点，同时一些活跃点因为相机的位姿变化而被边缘化。最后我们选择适合的边缘化和 hessian 点云来生成更密集的场景，因为它们更加的稳定。虽然增加点云数量在系统准

确性或鲁棒性方面几乎没有任何好处，但是更加密集的点云对二维坐标点和轮廓以及平面估计（见 3.5.2 小节）之间的位置约束的确有一定的影响，二维坐标点的数量越丰富，最后估计出来的平面结构越加鲁棒。所以在本文中，考虑到速度和效率的关系，我们选择 8000 个活动点作为折中选择。这样的操作使我们最后生成的地图更加的稠密，满后续稠密重建的需求，使最终稠密重建效果更加优秀。

当然，随着半稠密地图的生成完成，我们的半稠密线程的任务也告一段落。根据半稠密线程和稠密线程运行效率的不对等性，随后我们需要将关键帧图像，相机姿态，相机参数和点云准备到由列表容器构成的数据缓冲区中，以防止数据丢失。

3.4 数据桥接

在视觉里程计产生图像的关键帧和半稠密点云以后，本节主要考虑如何得到更加鲁棒和精确的半稠密点云以及如何在尽量小的内存消耗和计算量下得到高效的稠密三维重建，因此本节分三部分介绍数据桥接的具体内容。先给出两者之间的坐标系转换系统，同时考虑到重建实时性和效率两者的关系，我们设计了一个数据交换容纳池以及地图点选择策略。

3.4.1 坐标系转换与数据交换

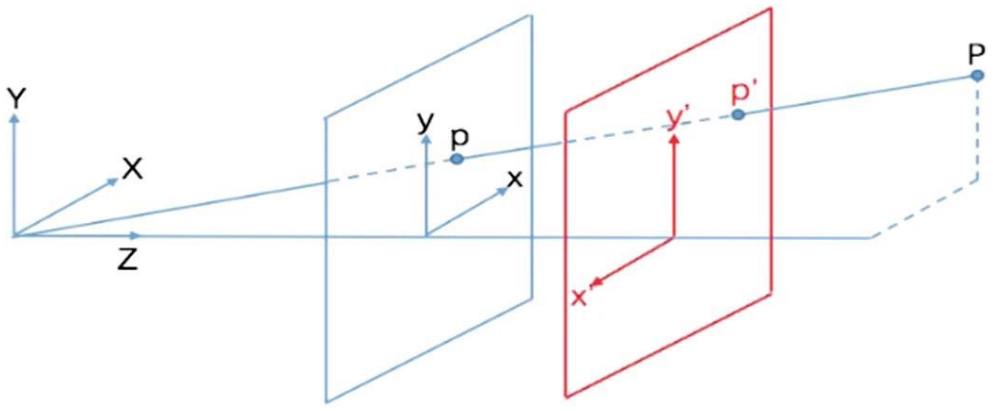

图 3-3 坐标系转换

我们观察到 DSO 中的坐标系和稠密重建系统中是不同的。如上图所示，点**错误!未找到引用源。**的 X 轴位置成为点**错误!未找到引用源。**的相反方向。所以如果我们直接将它们结合起来，稠密重建结果将与其他 DSO 半稠密结果分离，导致我们无法获得完美的最

终重建结果。因此，在桥接部分摄像机位置和地图点位置也从 DSO 坐标系统转换为稠密重建坐标系。

当 DSO 生成关键帧时，我们无法立即将数据传递给稠密重建线程，因为稠密重建线程可能正在运行。所以使用容器来保存关键帧数据是非常必要的。但是如果我们保留所有关键帧信息，将会逐渐增加我们系统的成本，之后系统内存将崩溃。因此必须使用容器的删除和更新机制来减轻系统的负担。同时，当我们稠密重建系统线程一次运行结束时，应该将包含的数据交换到该线程。列表容器包含 5 个列表对象。在每个对象中包含关键帧的索引 ID、灰度图像、旋转矩阵、平移矩阵和精确的半稠密点云。首先准备来自 DSO 中关键帧的三通道图像，然后将旋转平移矩阵从世界坐标系变换为相机坐标系，最后，半稠密点云也需要从摄像机坐标转换为世界坐标。然后把它们放在一起作为一个列表容器。一旦稠密建图线程结束，我们立即交换容器数据。另外，当即将到来的数据超过容器的限制时，根据索引 ID 立即移除最早的数据。

3.4.2 地图点选择策略

为了在所有的激活帧中保持一定数量的特征点数量，我们需要对关键帧中生产的特征点进行动态选择和改变。DSO 中每一个新生成的关键帧会伴随着四种不同类型的点：边缘化后的点，hessian 点，不成熟点和 hessian 异常点。边缘化点由舍尔补（schur complement）后产生，当一些点被边缘化后，需要激活一些额外的候选点（不成熟点）并去替换掉边缘点，这样使得活跃点总数保持一致。Hessian 点是 DSO 中的活动点，用于基于滑动窗口的相机位姿优化。不成熟点是不精确的摄像机跟踪的原始候选点，一旦该点被激活，系统就会初始化这些点的深度。Hessian 异常点是潜在的系统异常点，通过在相机跟踪过程中沿着极线搜索而消除。此时如果我们将所有四种类型的点都用于稠密建图线程，那么我们的系统中可能会出现很大的重建误差。考虑到这四类点的不同属性，我们提出了一个地图点选择策略：其中前两种类型即边缘化后的点和 hessian 点更适合于生成更稠密的场景，因为它们更加的稳定。尽管当大量增加点云密度在系统精度或鲁棒性方面几乎没有好处，但是对稠密的点云在后续二维点和轮廓之间的位置确定以及平面估计具有一定的影响作用。为了权衡速度和效率，本文最终选择了在所有激活帧中保持 8000 个激活点。

3.5 稠密三维重建后端

随着前面视觉里程计中半稠密点云的生成，我们需要通过额外的图像处理技术进行环

境的稠密三维重建，主要是对视觉里程计中忽略的平面结构进行稠密的三维重建。本文主要介绍一种基于图的图像分割算法进行稠密三维重建方法，主要将二维图像进行图像分割并提取出分割完后物体的轮廓位置，将半稠密点云反投影到二维图像上，通过两者在图像坐标系上的位置判断其约束关系，随后通过 SVD 奇异值分解计算出平面的方程并用 RANSAC 算法排除异常点，最后通过稠密建图算法去除误匹配点提高点云的精度。

3.5.1 基于图的图像分割算法

本文采用了基于图的分割算法，主要依赖贪心聚类算法，实现简单，速度比较快，精度也较高。通常来说，主要使用基于图的表示来测量两个图像区域的边界。首先将**错误!未找到引用源。**定义为无向图。其中顶点**错误!未找到引用源。**代表图像中的像素。每个边**错误!未找到引用源。**对应一个权重**错误!未找到引用源。**，然后取强度、颜色、运动、位置或其他局部属性的差异来度量两个像素之间的不相似度。这样最终得到的结果是每个图像区域的最小生成树，每个树表示一个分割结果。算法使用了自适应阈值代替全局阈值，全局阈值太大，会使原本差距较少的区域产生合并，导致分割结果太粗；全局阈值太小，高频区的图像信息会被分割成诸多小块，导致分割过细。因此自适应阈值与全局阈值相比可以做到更好的效果。这样通过增加阈值使得高频区信息分割为同一块，通过降低阈值使低频区信息分割为不同块。极大的增加分割效果。自适应阈值策略通过额外两个附加信息来实现，分别是：类内差异 Int 和类间差异 Diff，其中，

$$\mathrm{Int}(C) = \max_{e \in (MST,E)} e \tag{31}$$

$$\mathrm{Diff}(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i v_j) \in E} \Psi(v_i, v_j) \tag{32}$$

类内差异可以看成是一个区域内部最大的亮度差异值，即最小生成树中不相似度最大的一条边，而类间差异可以描述成连接两个区域所有边中，不相似度最小的边的不相似度，也就是两个区域最相似的地方的不相似度。此外，两个区域是否合并的标准被定义为如下：

$$\mathrm{Diff}(C_i, C_j) \le \mathrm{MInt}(C_i, C_j) = \min\left(Int(C_i) + \mathrm{r}(C_i), Int(C_j) + \mathrm{r}(C_j)\right) \tag{33}$$

其中：

$$\mathrm{r}(C) = \frac{k}{|C|} \tag{34}$$

错误!未找到引用源。，**错误!未找到引用源。**分别是区域所能忍受的最大差异，当二者

都能忍受当前类间差异**错误!未找到引用源。**的时候，那么这两个区域就会被融合在一起。此外算法为了避免出现过分割还额外增加了**错误!未找到引用源。**，它的**错误!未找到引用源。**表示区域**错误!未找到引用源。**中所包含的像素点的个数，随着像素区域块大小的增加，增加项的值会越来越小。通过控制**错误!未找到引用源。**值来实现对当前图像的区域块大小的控制，**错误!未找到引用源。**越大分割后的图也越大。

我们使用以上算法对一帧关键帧处理的结果如下图所示：

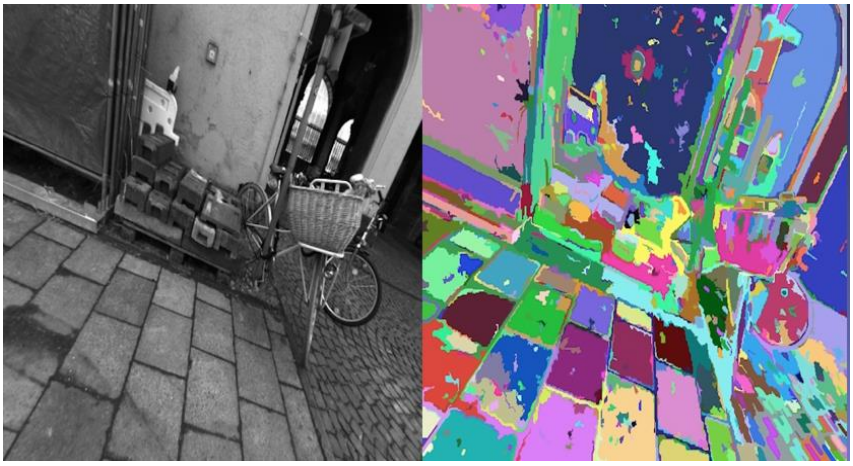

图 3-4　基于图的分割效果

3.5.2　位置约束和鲁棒平面估计

对于图像中的像素之间的关系来说，梯度突变的位置总是存在于颜色或亮度等突然改变的地方或一些物体对象的边缘。所以我们根据前面得到的分割图像和 DSO 的半稠密点云，可以确定半稠密点云和超像素轮廓之间的位置约束关系如下：

$$\boldsymbol{Q_i} = \lambda_i \boldsymbol{K}^{-1} \boldsymbol{q_i} \tag{35}$$

$$\text{distance}(\boldsymbol{Q_i}, \boldsymbol{S_i}) < \varepsilon \tag{36}$$

其中式 35 表示标准的针孔模型，**错误!未找到引用源。**表示三维地图中的位置，**错误!未找到引用源。**表示深度常数，**错误!未找到引用源。**表示相机参数，**错误!未找到引用源。**表示图像中特征点的二维位置。阈值**错误!未找到引用源。**表示围绕投影点位置相邻的八个像位置。如果距离小于**错误!未找到引用源。**。我们则认为投影点在超像素轮廓之内。

此后，通过先前一段定义的半稠密点和分割轮廓去估计平面。 我们计算图像中符合条件的每个轮廓，然后使用 SVD 和 RANSAC 来拟合一个鲁棒的平面。此外，我们还使用额外的评估标准来评估平面的质量。通过归一化残差测试，主动搜索、时间一致性以及考虑图像退化情况。

$$\frac{dis(p_u, \pi_i)}{dis(p_i, p_j)} < \xi \tag{37}$$

分子代表三维点到平面的距离。分母代表三维点之间的距离，阈值**错误!未找到引用源。**为-0.05。退化的情况通过求解 SVD 中的退化秩来排除错误轮廓。主动搜索是减少三维轮廓和二维超像素之间误差的一个非常重要的部分。因为在基于图的分割和鲁棒平面估计之后，我们无法区分哪个超像素与单一视图中的轮廓相对应，因为每个轮廓可能至少有两个相邻的超像素。为此，我们计算当前帧轮廓和相邻帧轮廓之间的重投影误差。然后，我们排除重投影中的低重叠区域。

3.6 实验结果与分析

本系统用了德国慕尼黑工业大学开源的单目视觉里程计的数据集作为我们的测试数据集。我们在该测试数据集上做了大量的测试实验。以下我们将分别展示一些我们算法三维稠密重建过程中的中间结果和最终结果来说明我们的算法是切实有效实用的。我们首先展示了关于一帧关键帧生成的半稠密点云的和进行平面估计后的稠密重建结果（见图3-5）。图 3-5 最左边一列是在 DSO 中提取的一帧关键帧。中间一列是对应于图的半稠密点云，从图中大致可以看出那些灰色的半稠密点云位置几乎对应于图像中物体的一些边界。最右边一列是算法的稠密重建结果。纯白色突出显示的区域是我们最终稠密结果的一部分。这样结合上一小节中的基于图的分割图的效果，我们可以看出我们的算法在原先半稠密地图的基础之上，通过很好的对地面、墙面的平面估计，将环境中的平面有效的重建了出来。整个稠密三维重建的算法流程就是在这个生成框架下执行的。

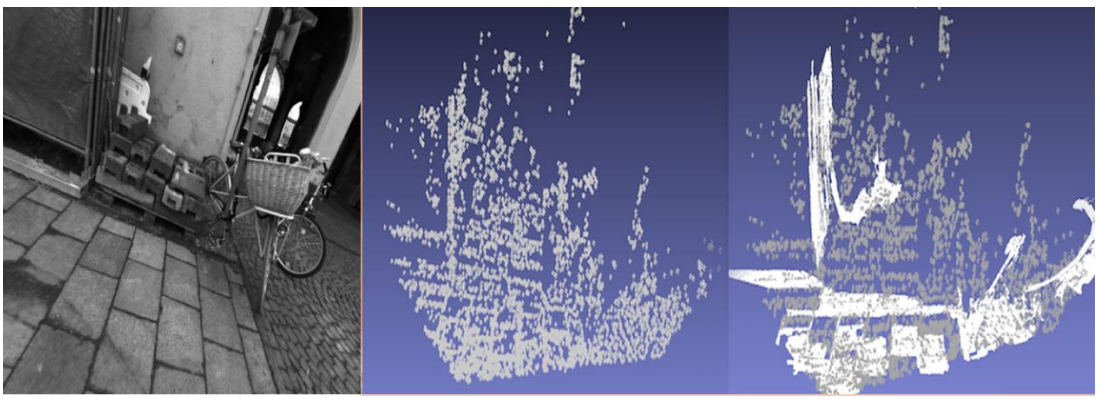

图 3-5　基于直接法三维重建的中间结果

同时我们的最终实验结果还表明，我们的算法可以解决高梯度点和低纹理区域之间（平面）的区域无法重建的问题，实现重建效果从半稠密向稠密的转换。如图 3-6 中所示，我们的算法与对比算法均运行在同一个测试数据集下。左边第一列是一帧视频帧中 DSO 提取出来的关键帧，中间列的结果是 DPPTAM 算法的重建结果，它重建出来的点云比较少，这是因为 DPPTAM 算法是基于 LSD-SLAM 的视觉里程计前端，其因为存在累积误差和几何误差，会造成在重建的时候系统跟踪丢失和初始化不准确的情况，而且对弱纹理区域的跟踪和亮度较低区域的准确跟踪几乎不可能实现。最右边列是我们的算法重建出来的效果，不仅继承了 DSO 鲁棒稳定的视频帧跟踪以及优秀的半稠密建图效果，图中红色区域是我们能够估算出来的平面，可以看出我们的算法能够很好的填补弱纹理平面区域，是对半稠密视觉里程计的重建效果上一个很好的稠密补充。因为 DPPTAM 的前期跟踪的丢失，我们这一组对比实验仅仅反映了在视频流初始阶段的一些三维重建的稠密效果对比。此外，我们的数据集能够完整的跑完数据集，很少会出现丢失情况。如图 3-7 中展示的一些其他数据集的运行结果，其中白色高亮区域的是我们算法的重建结果，可以看到算法对环境中一些地面和墙面完成了很好的重建效果。我们的算法在 2.3 千兆赫、因特尔酷睿 i5-6300HQ 处理器、8GB 内存的联想 Y700-15ISK 笔记本上能够达到实时稠密重建的效果。另外在 3.5.1 小节中基于图的分割算法处理一帧 640*480 大小的关键帧的时间大约在 0.18 秒左右。

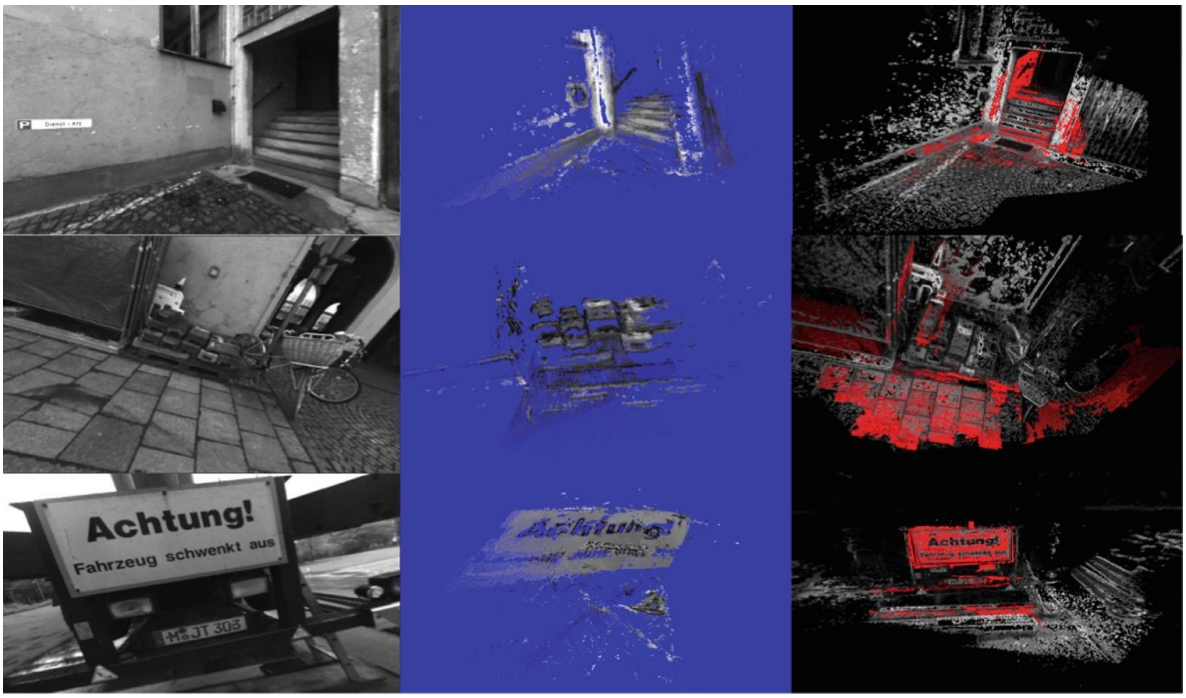

图 3-6　两种实验的三维重建结果，最左边一列是关键帧，中间一列是 DPPTAM 算法的结果，最右边一列红色部分是我们算法得到的结果

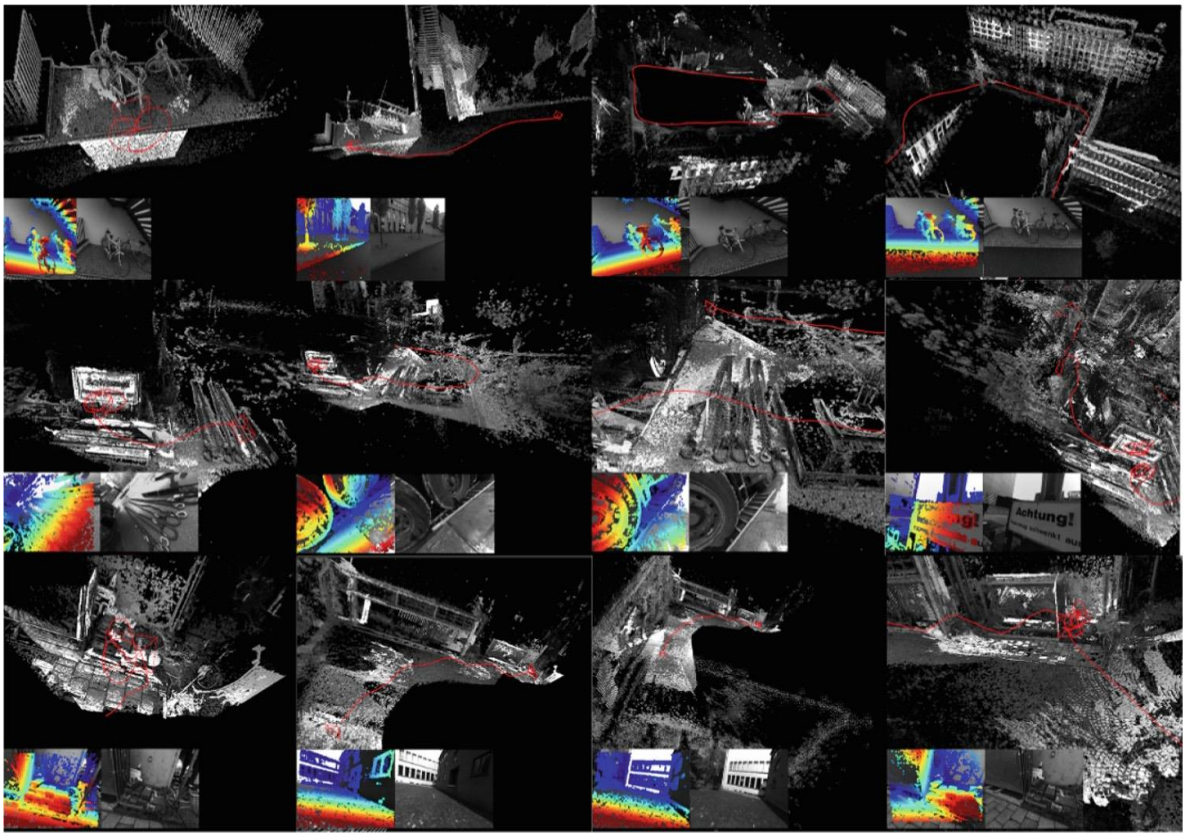

图 3-7　多个数据集上的运行效果，白色高亮区域为本文算法的重建区域

3.7 本章小节

本章主要介绍基于直接法的稠密三维重建。我们提出了一种单目稠密重建系统，它比当前最先进的技术有效并且快速，而且对于相对较严格的环境，我们的重建系统能够进行鲁棒的跟踪。我们将先进的视觉里程计和稠密的重建线程通过精心制作的桥接部分相结合。通过与当前先进的单目稠密重建方法相对比，我们的系统即使在弱纹理的场景中也可以在室内，室外环境下成功运行。实验结果表明，基于直接法的稀疏或半稠密里程计能够很好的与基于图的分割算法以及平面检测算法相结合，实现环境的稠密三维重建。此外，我们的系统在 CPU 上运行并且是实时的，因此可以应用于许多领域。

第 4 章　基于特征法稀疏点的稠密三维重建

4.1　引言

本章主要介绍一种基于特征法稀疏点的稠密三维重建。在 ORB_SLAM2 提供的关键帧和 ORB 稀疏特征点的基础上，通过使用基于深度学习得到的图像轮廓信息和基于图的分割信息优化最终轮廓效果图。我们同时使用了八邻域位置约束判断将 ORB 稀疏点与轮廓图进行位置绑定，再用最小二乘法估计出一个鲁棒的平面，同时将特征信息和轮廓信息都附加到平面对象中。最后使用词袋模型来完成冗余平面的剔除。本章节中主要的研究贡献在于提出了一种基于图的分割图和 RCF 边缘检测图的融合算法来取得更为精确鲁棒的图像轮廓图，同时为了减轻整个系统的负载和提高重建精度，提出了一个稠密点云基于字典的相似度计算融合算法。

4.2　稀疏 ORB 特征点与 RCF 边缘检测算法

4.2.1　稀疏 ORB 特征点

ORB_SLAM2 系统中提取 ORB 特征由 Oriented FAST 关键点和 BRIF 描述子构成，FAST 角点检测算法最初是由 Edward Rosten 和 Tom Drummond 提出，该算法的基本原理是使用圆周长为 16 个像素点，即通过一个半径为 3 的圆，来判定其圆心像素 P 是否为角点。

通过对圆周上以顺时针方向从 1 到 16 的顺序对圆周像素点进行编号。如果在圆周上有 N 个连续的像素的亮度都比圆心像素的亮度加上阈值 t 还要亮，或者比圆心像素的亮度减去阈值 t 还要暗，则圆心像素被称为角点。因此 FAST 角点反映了局部图像灰度变化显著的地方。仅仅依靠这种策略的 FAST 角点容易出现扎堆的现象，因此需要对第一遍图像检测结果用非极大值抑制的方法进行处理，以避免角点过于密集。

上述得到的 FAST 角点不具有方向信息和尺度信息，在 ORB_SLAM2 中为了得到更为鲁棒的 ORB 特征匹配，为之添加了旋转和尺度信息。通过构建图像金字塔，对图像进行不同层次的降采样，获得不同分辨率下的图像。在保持旋转不变性上，采用了灰度质心法，

将图像块灰度值作为权重的中心，对于一个小的图像块 P 中，设矩为：

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y), \quad p,q = \{0,1\} \tag{38}$$

这样通过矩寻找到图像的质心**错误!未找到引用源。**为：

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{39}$$

连接圆心和质心得到方向向量**错误!未找到引用源。**，定义特征点的方向为：

$$\theta = \arctan(m_{01}/m_{10}) \tag{40}$$

这样通过图像金字塔和灰度质心法，使得 ORB 特征拥有尺度和旋转信息，在后续的特征匹配中更为鲁棒和准确。

ORB 特征匹配则是通过 BRIEF 描述子实现的，它则是一种二进制描述子，它由多个 0 和 1 构成一组描述向量，通过计算二值串的捷径避免了高维向量的对比运算。在平滑后的图像周围选取一个小区域，通过概率挑选的原则选出 n 个点，两两构成一个点对，对于每一个点对来说，计算两者之间的亮度值。如果前者较大，那么二进制值为 1，如果后者较大，二进制值为-1，否则为 0。所有点对都进行比较之后就能得到一个二进制串。之后，通过计算两个二进制之间的汉明距离结合阈值设定来判断是否属于同一个点。同样 ORB 稀疏点与前一小节的轮廓约束关系依旧可以用 3.5.2 小节中的位置约束描述，保持八邻域的位置约束关系。

4.2.2 RCF 边缘检测算法

本文使用的边缘检测算法是 2017 年在 CVPR 发表的 RCF[11]算法，传统的边缘检测算法一般以池化最后一层的结果作为输出，没有考虑到卷积层之间的互补信息，RCF 则是考虑了所有的卷积层信息，设计了更为丰富的卷积特征，以下是 RCF 的核心流程图：

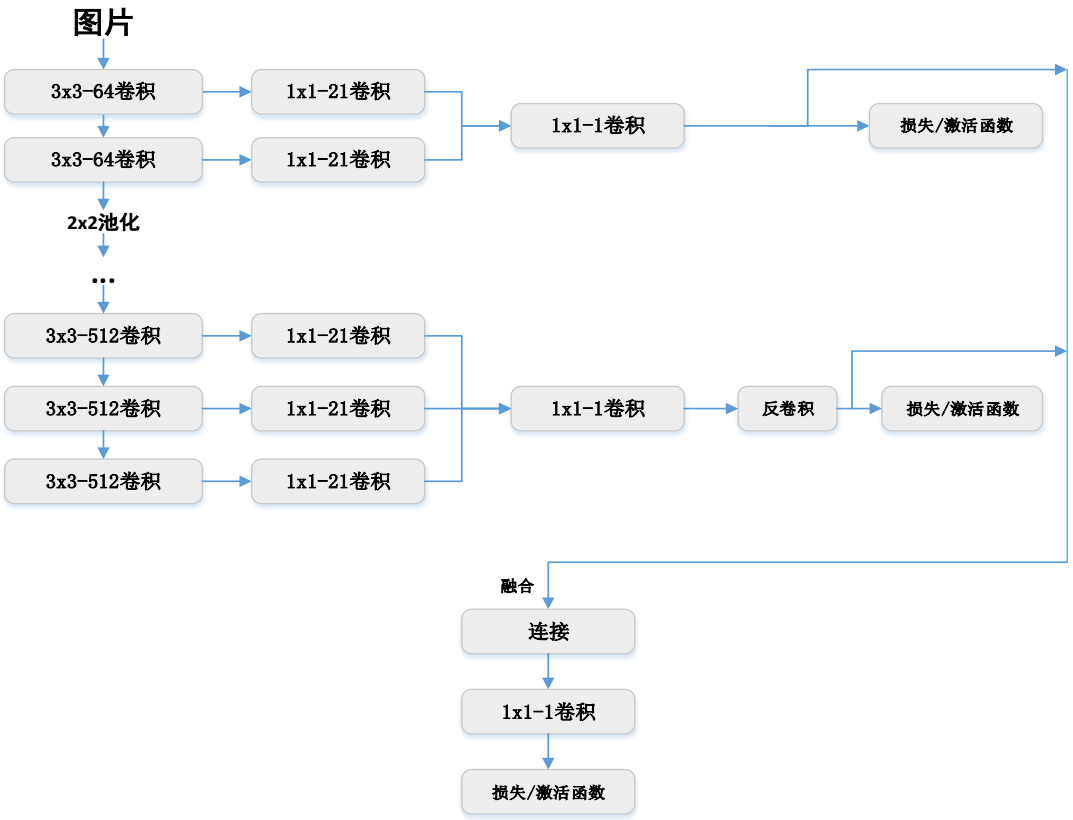

图 4-1 RCF 网络结构示意图

RCF 一共考虑了五个阶段，相邻两个阶段通过池化层降采样得到不同尺度的特征信息。RCF 对 VGG16 做了以下五点的改动，一：将全连接层和第五池化层除去了，去除全连接层可以得到全卷积网络，同时第五池化层对图降采样，不利于边缘的定位；二：对于 VGG16 中每个卷积层使用了一个 1x1 大小的核以及 21 个通道深度的卷积层，每个阶段里面所有的 1x1x21 卷积输出进行一个简单累加操作，从而得到一个符合要求的特征属性；三：每一个相加后的卷积层后面添加了一个 deconv 层，主要用于放大特征尺寸；四：在每一个上采样层后面使用了一个交叉熵损失函数（S 形函数）层；五：对所有上采样的层进行关联，随后使用一个 1x1 的卷积层进行特征融合，最后使用交叉熵损失函数 S 形函数层得到输出结果。

同样 RCF 设计了十分鲁棒的损失函数，不考虑一些有争议的边缘点。这些点包括在多个标记中只有部分标记某个像素为边缘点，而不是全部。这样针对每个像素，设计的损失函数如下：

$$\mathrm{l}(X_i;W)=\begin{cases}a\cdot\log\left(1-P(X_i;W)\right) & if\ y_i=0\\ 0 & if\ 0<y_i\le\eta\\ \beta\cdot\log P(X_i;W) & otherwise\end{cases} \tag{41}$$

其中，

$$\alpha = \lambda \cdot \frac{|Y^+|}{|Y^+| + |Y^-|} \tag{42}$$

$$\beta = \frac{|Y^-|}{|Y^+| + |Y^-|} \tag{43}$$

同时虽然神经网络自带了多尺度信息，但是显式的使用多尺度融合对于图像的边缘效果依然是能够做到有一定的提升。以下是一些单目数据集里面使用 RCF 算法得到的边缘检测效果展示：

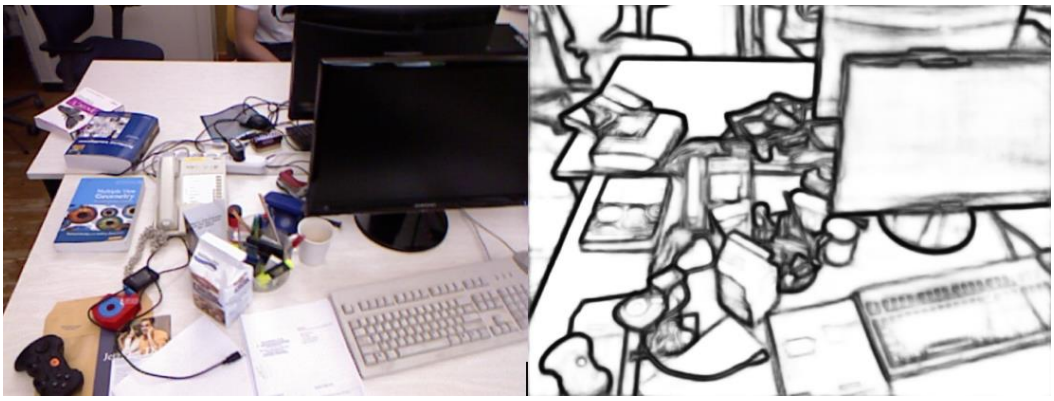

图 4-2　RCF 边缘检测效果图

从 RCF 的边缘检测图中我们不难发现，检测图很好的展示了物体的复杂边缘特性，同时检测图中的灰度信息的不一致一方面也说明了 RCF 算法对物体边缘性质的不确定性。

4.3　基于稀疏特征点的稠密重建算法

为了能够通过使用 ORB 稀疏特征点对环境实时进行稠密的三维重建，在原先的 ORB_SLAM2 系统的基础上，我们提出以下三个策略，分别是：通过边缘检测图与基于图的分割图进行融合提高图像轮廓信息，基于最小二乘法完成平面的提取，而为了防止环境中重复平面的生成，我们提出了稠密点云基于字典的相似度计算的融合算法。

4.3.1　边缘检测图与基于图的分割图融合策略

基于图的分割图生成图像的轮廓主要是通过比较不同超像素块之间的颜色差别来判定。如图 4-3 所示，该算法得到的轮廓图效果较差，对于聚集在一块的物体的边缘几乎很难分辨出来，而对于一些边缘较窄的物体来说，基于图的分割算法最终估算出来的边缘相

对比较粗糙，较难分辨。因此对后续位置约束判断的时候所需的特征点数量要求就越高。本文希望能够使用更少的特征点数目来进行稠密的三维重建，通过结合图分割的轮廓和 RCF 的边缘检测轮廓，从而得到更为精确鲁棒的轮廓图。从而减少对特征点数量的判定要求。

图 4-3　基于图的分割后的轮廓图

RCF 边缘检测的效果图为一张灰度图像，RCF 检测图颜色越深接近 0，代表边缘越明显，越接近 255，图像的边缘信息变得越模糊不清，不容易判断。本文算法中考虑的是 RCF 灰度图与图优化分割图，为此设计了一个基于分段函数的融合算法，算法考虑了两者相对权重关系，同时由于 RCF 的边缘检测效果出众，所以最后结果以 RCF 的边缘效果为导向，辅助融合了基于图的分割轮廓。图的分割轮廓算法可以参考小节 3.5.2。这里我们用**错误!未找到引用源。**表示 3.5.2 小节里判定位置点**错误!未找到引用源。**是否为轮廓点的布尔值，如果该位置点是位置约束的轮廓点，其结果为 true，否则为 false。**错误!未找到引用源。**表示位置点**错误!未找到引用源。**经过 RCF 算法后得到的当前位置的灰度值。

$$\begin{cases} 1, \ if\ 0 < G\left(\boldsymbol{P}_{i_{RCF}}\right) < 0.8 * \max\left(G\left(\boldsymbol{P}_{i_{RCF}}\right)\right) \ and \ B\left(\boldsymbol{P}_{i_{SEG}}\right) = true \\ M(\boldsymbol{P}_i), \ if\ 0 < G\left(\boldsymbol{P}_{i_{RCF}}\right) < 0.8 * \max\left(G\left(\boldsymbol{P}_{i_{RCF}}\right)\right) \ and \ B\left(\boldsymbol{P}_{i_{SEG}}\right) = false \\ N(\boldsymbol{P}_i), \ if\ G\left(\boldsymbol{P}_{i_{RCF}}\right) \geq 0.8 * \max\left(G\left(\boldsymbol{P}_{i_{RCF}}\right)\right) \ and \ B\left(\boldsymbol{P}_{i_{SEG}}\right) = true \\ 0, \ if\ G\left(\boldsymbol{P}_{i_{RCF}}\right) \geq 0.8 * \max\left(G\left(\boldsymbol{P}_{i_{RCF}}\right)\right) \ and \ B\left(\boldsymbol{P}_{i_{SEG}}\right) = false \end{cases} \tag{44}$$

其中：

$$\mathrm{M}(\boldsymbol{P}_i) = \frac{\max\left(G\left(\boldsymbol{P}_{i_{RCF}}\right)\right) - G\left(\boldsymbol{P}_{i_{RCF}}\right)}{\max\left(G\left(\boldsymbol{P}_{i_{RCF}}\right)\right)} + \frac{Ner(\boldsymbol{P}_i)}{8} \tag{45}$$

当位置点的 RCF 检测图的灰度值小于检测图最大灰度值的 80%以及位置约束轮廓点判定为真的时候，我们认为当前位置点的融合值为 1，即该位置点是融合后的轮廓点；当

位置点的 RCF 检测图的灰度值小于检测图最大灰度值的 80%以及位置约束轮廓点判定为非的时候，我们的融合算法将会考虑当前位置约束轮廓点的八领域位置点是否为轮廓点，当**错误!未找到引用源。**的时候，我们认为该位置点是融合后的轮廓点；当位置点的 RCF 检测图的灰度值大于等于检测图最大灰度值的 80%以及位置约束轮廓点判定为真的时候，即**错误!未找到引用源。**，它表示最终的 RCF 检测图的灰度值大小在 20 之内，我们认为当前位置的融合值是 1，其余是 0；当位置点的 RCF 检测图的灰度值大于等于检测图最大灰度值的 80%以及位置约束轮廓点判定为非的时候，我们认为当前位置点的融合值为 0，即该位置点我们不认为是个轮廓点。

基于图的分割后的轮廓图与使用 RCF 融合分割图后的轮廓效果图如下：

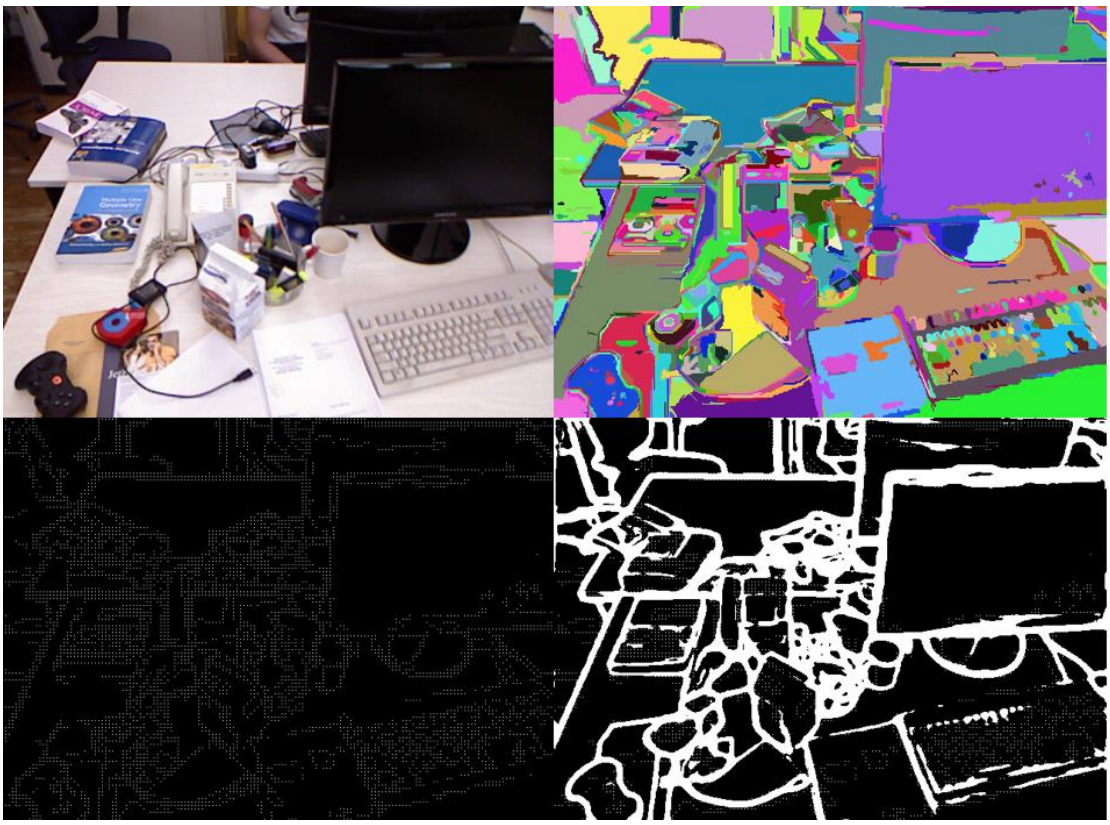

图 4-4　基于图的分割轮廓图与 RCF 融合分割图后的轮廓效果对比图

对比图 4-4 和图 4-3 可以发现，单纯的使用基于图的分割算法，轮廓的位置点比较稀疏，而且在轮廓边缘处容易出现多条轮廓线。因此后续需要使用该轮廓的时候需要大量的反投影点进行位置约束验证。而通过使用 RCF 的边缘特征信息后，我们的图像轮廓更加的清晰，融合算法统筹考虑两者的权重关系，对于基于图的分割算法中有些不明显的轮廓进行了剔除，同时也对一些不准确的小区域进行了排除，融合得到的轮廓包含平面的概率更大，减少了后续工程中需要遍历的代价。这样在精确轮廓图的基础上，我们可以大大降低

后续需要验证位置约束关系所提供的点云数量，同时对后续的稀疏点与轮廓图的位置约束关系判断提供更好的数据支撑，达到预期的稠密重建效果。

4.3.2 基于最小二乘法的平面估计

既然拥有了 ORB 稀疏点与轮廓的位置关系，我们使用最小二乘法来估算最终的平面。最小二乘法是一种回归分析中的标准方法，用于近似超定系统求解，即其中方程组多于未知数的方程组。最小二乘主要是通过最小化整体解决方案中每个单一方程结果中残差的平方和，它最重要的一个应用是用来做数据拟合。最小二乘意义上的最佳拟合使残差的平方和最小（残差为：观测值与由模型提供的拟合值之间的差值）。当问题在自变量中存在很大的不确定性时，则简单回归和最小二乘方法会存在问题；在这种情况下，可以考虑拟合变量模型所需的方法而不是最小二乘法。最小二乘问题分为两类：线性或普通最小二乘和非线性最小二乘，取决于残差是否在所有未知数中是线性的。线性最小二乘问题出现在统计回归分析中；它有一个封闭的解决方案。非线性问题通常是通过迭代来求解来，在每一次迭代中，系统都近似为一个线性系统，因此核心计算在两种情况下都是相似的。

通过前一小节的约束关系之后，我们对每个轮廓图跟周围的 ORB 特征点进行了一一绑定。随后本文使用最小二乘法结合 ORB 稀疏特征点估计出一个鲁棒的平面。首先，对于每一个需要计算的关键帧来说，我们遍历每个关键帧所有的轮廓，通过每一个轮廓对应的所有的 ORB 特征来计算三维平面。在这里，我们认为 ORB_SLAM2 系统估计出来的 ORB 特征是十分鲁棒的，先将特征点的坐标投影到世界坐标系下，再通过构建如下平面方程来计算三维平面：

$$\mathrm{Ax} + \mathrm{By} + \mathrm{Cz} + \mathrm{D} = 0, \quad (\mathrm{C} \neq 0) \tag{46}$$

记：

$$a_0 = -\frac{A}{C} \quad a_1 = -\frac{B}{C} \quad a_2 = -\frac{D}{C} \tag{47}$$

可得：

$$\mathrm{z} = a_0 x + a_1 y + a_2 \tag{48}$$

此时在三维空间中假设由 n 个二维 ORB 特征点投影到空间中的三维点**错误!未找到引用源。**：

$$(x_i, y_i, z_i), \quad i = 0,1,\cdots,n-1 \tag{49}$$

我们将用这些三维点来估算出一个鲁棒的三维平面，有：

$$S=\sum_{i=0}^{n-1}(a_0x_i+a_1y_i+a_2-z_i)^2 \tag{50}$$

此时必须得使 S 最小，应该满足：

$$\frac{\partial S}{\partial a_k}=0,\quad k=0,1,2 \tag{51}$$

即：

$$\begin{cases}\sum_{i=0}^{n-1}2(a_0x_i+a_1y_i+a_2-z_i)x_i=0\\ \sum_{i=0}^{n-1}2(a_0x_i+a_1y_i+a_2-z_i)y_i=0\\ \sum_{i=0}^{n-1}2(a_0x_i+a_1y_i+a_2-z_i)z_i=0\end{cases} \tag{52}$$

化简得：

$$\begin{cases}a_0\sum_{i=0}^{n-1}x_i^2+a_1\sum_{i=0}^{n-1}x_iy_i+a_2\sum_{i=0}^{n-1}x_i=\sum_{i=0}^{n-1}x_iz_i\\ a_0\sum_{i=0}^{n-1}x_iy_i+a_1\sum_{i=0}^{n-1}y_i^2+a_2\sum_{i=0}^{n-1}y_i=\sum_{i=0}^{n-1}y_iz_i\\ a_0\sum_{i=0}^{n-1}x_i+a_1\sum_{i=0}^{n-1}y_i+a_2n=\sum_{i=0}^{n-1}z_i\end{cases} \tag{53}$$

将每个点带入到公式中计算得到a_0，a_1，a_2，从而得到三维平面的函数方程以及计算得到的误差**错误!未找到引用源。**值，随后将S值作为当前平面的误差值并对当前平面的属性做一次绑定。属性绑定是因为我们考虑到系统有时候建出的平面会存在大误差，后续构建过程需要根据S值的大小剔除大误差平面，在本实验中当S的值大于 2 的时候，我们考虑这个平面建立的时候误差较大，应当舍去。

4.3.3 稠密点云基于字典的相似度计算融合算法

随着系统的运行，对关键帧处理完后的平面会越来越多，不一定能排除同一个平面存在多次重建的情况。而三维空间中对于两片点云相似的计算基本依赖点云的配准，但是点云的配准算法一般都比较耗时。因此在本文中我们通过对每个估计出来的平面设计一个相应的 ORB 特征点容器，将 ORB 特征代替平面去描述是否需要进行关键帧之间的平面融合，

使用字典生成词袋，得到的当前关键帧都与该关键帧的邻帧进行相似度计算，设计一个阈值区分邻帧之间的相似性，当相似性超过阈值，则代表邻帧存在相似平面，然后根据每个平面估算时候的误差（稀疏点与平面的距离误差）选取误差最小的平面，将其余平面抛弃。

当每进入一个关键帧进行相似检测的时候，我们得到该关键帧的所有邻帧，依次将每一个邻帧加入到字典中直到最后一个，随后我们通过这些关键帧建立起一个字典，方便后面做相似检索。我们选取的是拥有与轮廓约束关系的特征点，没有轮廓约束关系的特征点会进行剔除。

本文中使用 DBoW3 库完成字典的建立，将 ORB 特征点全部加入到字典中完成。相似度计算通过构建在文本检索中常用的 TF-IDF 实现。TF（译频率）表示的是某个单词在图像中经常出现，那么区分度就算很高。IDF（逆文档频率）表示的是某个单词在字典中出现的频率越低。那么它的区分度就越是明显。在当前词带模型中，采用统计某个叶子节点**错误!未找到引用源。**中的特征数量相对于所有特征的数量的比例看作是**错误!未找到引用源。**部分。我们假设所有的特征的个数为**错误!未找到引用源。**，同时有**错误!未找到引用源。**个**错误!未找到引用源。**节点，那么这个单词的**错误!未找到引用源。**可以表示成下面这个形式：

$$IDF_i = \log\frac{n}{n_i} \tag{54}$$

那么表示某个特征在整个图像中出现的频率**错误!未找到引用源。**可以表示成：

$$TF_i = \frac{n_i}{n} \tag{55}$$

其中，某个具体的图像中，单词**错误!未找到引用源。**出现了**错误!未找到引用源。**次，而一共出现的单词次数为**错误!未找到引用源。**。这样词带可以用一个向量**错误!未找到引用源。**描述，随后我们通过计算两个向量之间的**错误!未找到引用源。**范数来表达这两个图像的相似度：

$$s(v_A - v_B) = 2\sum_{i=1}^{n}|v_{A_i}| + |v_{B_i}| - |v_{A_i} - v_{B_i}| \tag{56}$$

通过比较当前的关键帧与字典里面所有邻帧的相似度分数，确定相似度最高的一帧关键帧，因为此时得到的相似性因为字典中图像数量较少结果比较粗略，所以后续我们结合 ORB 特征匹配将相似结果更加精细化。根据上文得到的在每一帧关键帧里的平面里均有绑定的 ORB 特征点属性，这样遍历当前关键帧里面每个平面，将每个平面里面对应的特征

点与相似度最高匹配的帧做特征匹配。当有一定数量的特征点匹配上时，说明这两个平面是在空间中是属于同一个平面，从而根据每个平面绑定的误差大小关系决定平面取舍，留下误差小的平面，去除误差大的平面，从而达到两个平面融合的效果，减轻了系统存储负载，提高系统三维重建的精度。

4.4 实验结果与分析

本文实验主要使用了一个包含 RGB-D 数据和地面实况数据的大型数据集。其中数据集包含了实时轨迹的 Microsoft Kinect 传感器的颜色和深度图像。数据集里面的视频都是以全帧 30 赫兹速率和 640x480 的传感器分辨率记录的，该数据集为了捕捉更为精确的地面实况运动轨迹，使用了高精度运动捕捉系统和 8 个 100 赫兹的高速跟踪摄像机追踪获得。在 ORB_SLAM2 获得的关键帧的基础上，我们首先使用基于图的分割对关键帧的彩色图像进行超像素化，随后使用 RCF 算法对关键帧图像进行边缘检测，两者融合后进行鲁棒平面的提取和去冗余。

图 4-5 展示的是基于 RCF 和分割融合的轮廓与原先基于直接法中的轮廓的对比效果，其中蓝色的点是图像中提取的 ORB 特征点位置，可以发现本文经过融合算法处理后，点与轮廓的位置更加的鲜明，对于后续的位置约束判断能够提供更好的数据支撑。

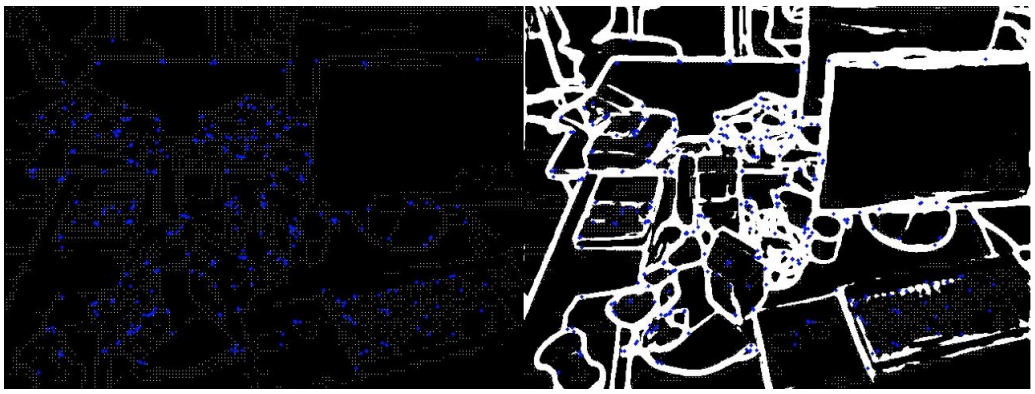

图 4-5 反投影位置约束示意图

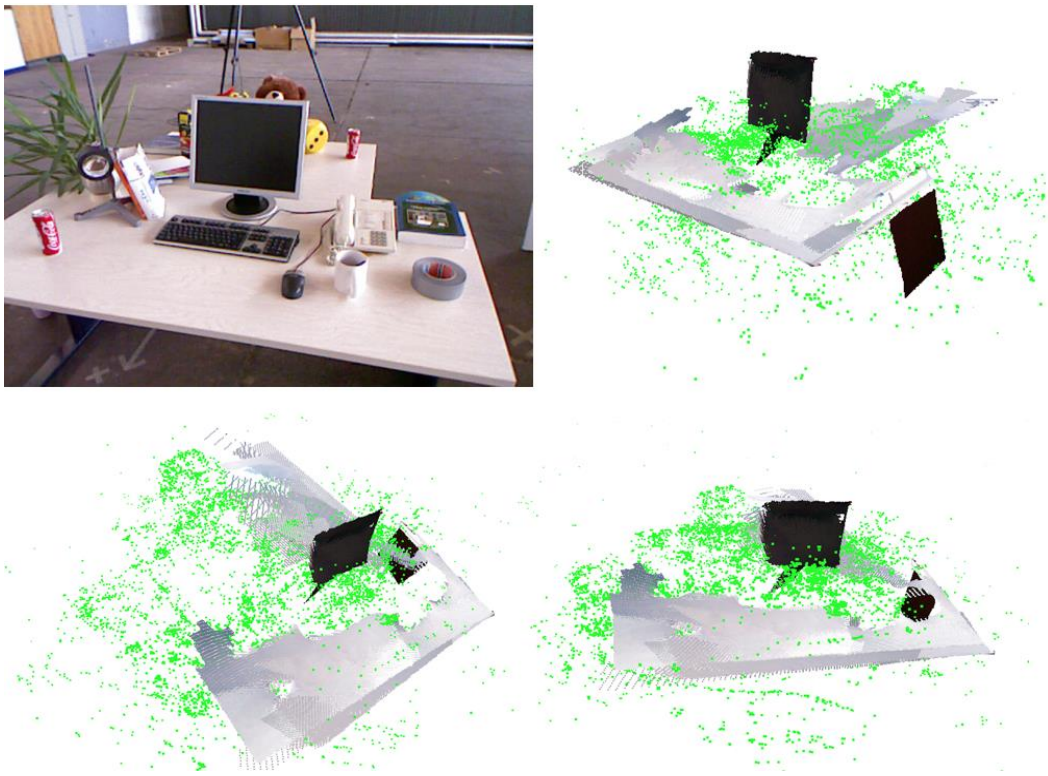

图 4-6　基于 ORB_SLAM2 系统的稠密重建结果

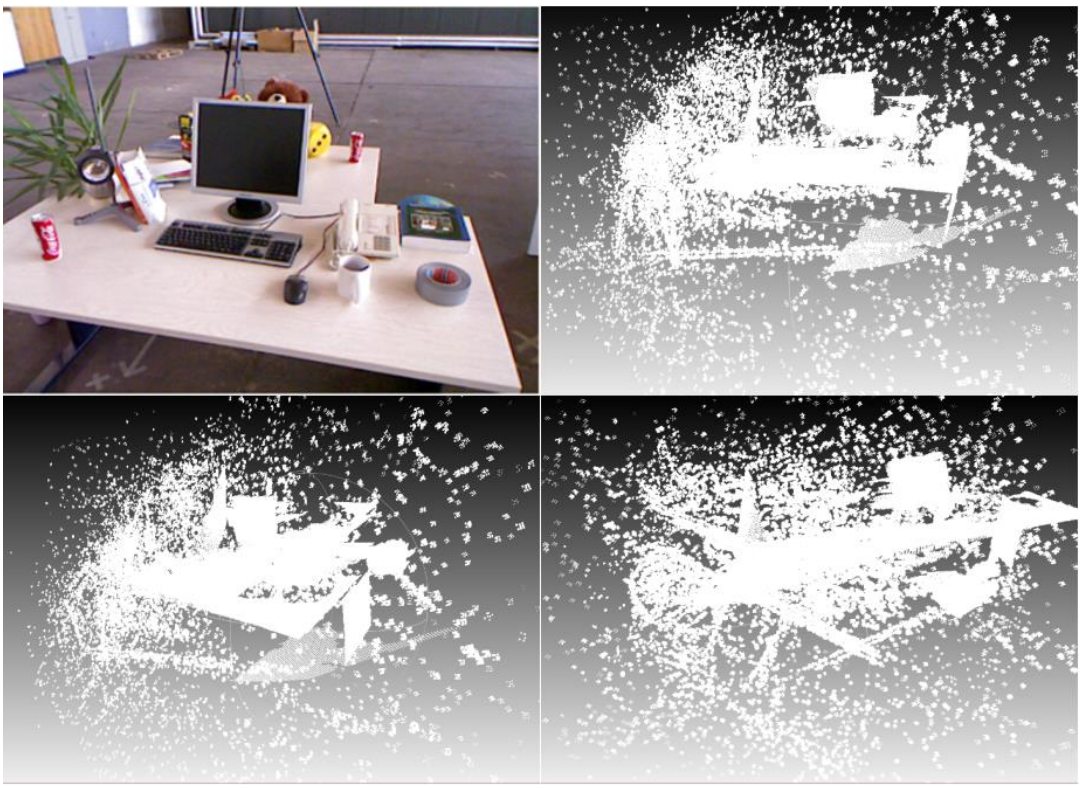

图 4-7 未进行光度标定的基于直接法的算法在 RGBD 数据集上的稠密重建结果

图 4-6 中左上角的图像是环境的截图，其余三张是该环境下重建出来的结果图，其中绿色点云是 ORB_SLAM2 系统原本就自带的稀疏地图，其余三维平面均为本算法经过 RCF

轮廓检测和基于图的分割融合后提取的鲁棒平面。如图 4-7 的实验结果是由本文第三章中的算法运行同一个 RGBD 数据集产生的，原先的基于直接法的稠密重建算法存在对数据集多次的系统初始化，同时后续估计出的相机姿态出现了一定的误差，其对应的半稠密点云出现了轻微的重叠和偏移，从而导致了重建出来的平面区域不能很好的和原始的半稠密点云进行拼接。由上述重建结果可知，本文的基于特征法稀疏特征点的三维稠密重建系统，通过重建初始环境中的平面来结合稀疏地图点云从而达到最终的稠密地图效果。从该点出发，本文的算法基本上达到了最初的预期效果，同时本文的算法通过多线程方式，能够在电脑上实时的运行。但是，在另一方面，本文的算法虽然去除了大部分重叠的平面，但是还是会存在对真实环境中的平面有部分的冗余重建，这也是本文算法在后期工作中需要提高和改进的地方。

4.5 本章小节

本章主要介绍了基于特征法稀疏点的稠密三维重建。介绍了所需的基础知识和基本流程：基于图的分割与 RCF 边缘检测图融合、鲁棒平面估计。通过上述操作可以直接使用 ORB_SLAM2 中的稀疏点进行平面估计和稠密三维重建。此外还使用了基于字典完成了对多重复平面的剔除，极大的提高了重建效率。本章使用来自慕尼黑工业大学的公共数据集进行三维重建，实验结果表明，我们的算法能够很好完成对环境的三维重建。

第 5 章　结论与展望

5.1　总结

现阶段基于单目相机的三维重建算法越来越趋向完善，本文主要完成了两个三维重建算法以及具体的流程，基于直接法的稠密三维重建和基于特征法稀疏点的稠密三维重建。本文主要的研究贡献点和内容主要表现在以下几个方面：

一：描述了三维重建上的一些基础方法，同时对国内外现状进行了详细的介绍，对三维重建系统做了简要的介绍。

二：列举了三维重建系统的一些基础知识点以及一些公式推导。

三：分别列出了基于直接法的三维重建系统和基于特征法的三维重建的流程框架，罗列了整个框架的运行步骤和逻辑，介绍了框架模块之间的相互作用关系和数据传递。

四：通过使用一种基于图的分割方法与半稠密系统的点云的位置约束进行鲁棒的环境平面估计，能够有效的完成平面生成，同时鲁棒的视觉 SLAM 前端可以在弱纹理和黑暗的环境成功完成跟踪。提出了通过深度学习得到的图像边缘信息进行轮廓净化，减少对特征点数量的依赖，成功在特征法 SLAM 上完成稠密重建。两者的算法在公共数据集上运行，实验结果表示算法高效可用。

5.2　展望

本文前期介绍的是基于直接法的稠密三维重建，可以成功完成对弱纹理环境和黑暗环境的跟踪，实现鲁棒的三维稠密重建。因此后续的工作大致在以下几个内容上做改进：

首先，基于直接法的三维重建系统虽然能够成功完成对环境平面的鲁棒估计，但是所依赖的半稠密点云数量还是相对较多，未来在如何减少点云依赖上还需做一些研究，降低资源使用。

其次，希望通过使用基于 CPU 运行的深度学习，同时追求运行效率和运行的结果，通过在基于 CPU 深度学习的图像边缘检测中和平面提取中寻找效率结果平衡点，为后续平台的扩展做准备。

最后，本文主要介绍的两个系统均在笔记本电脑端运行，未来考虑将算法移植到移动端或无人机，设备小型化，更加友好的为增强现实等应用提供环境感知任务。同时鉴于移动端或平板端的计算能力有所下降，后续还需要改进算法，尤其是在运行效率端，更加的需要减少内存使用，同时我们可以考虑使用临时地图等方案将存储要求降低，这样部署在无人机端可以用稠密地图做导航任务。

参 考 文 献

[1] Engel J, Koltun V, Cremers D. Direct Sparse Odometry[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99):1-1.

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

[3] Concha A, Civera J. DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2015.

[4] Felzenszwalb P F, Huttenlocher D P. Efficient Graph-Based Image Segmentation[M]. Kluwer Academic Publishers, 2004.

[5] Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-time dense surface mapping and tracking[C]// IEEE International Symposium on Mixed and Augmented Reality. IEEE, 2012:127-136.

[6] Newcombe R A, Lovegrove S J, Davison A J. DTAM: Dense tracking and mapping in real-time[C]// IEEE International Conference on Computer Vision. IEEE, 2011:2320-2327.

[7] 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.

[8] Whelan T, Kaess M, Fallon M, et al. Kintinuous: Spatially Extended KinectFusion[J]. Robotics & Autonomous Systems, 2012, 69(C):3-14.

[9] Whelan T, Leutenegger S, Moreno R S, et al. ElasticFusion: Dense SLAM Without A Pose Graph[C]// Robotics: Science and Systems. 2015.

[10] Tateno K, Tombari F, Laina I, et al. CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:6565-6574.

[11] Liu Y, Cheng M M, Hu X, et al. Richer Convolutional Features for Edge Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:5872-5881.

[12] Skea D, Barrodale I, Kuwahara R, et al. A control point matching algorithm[J]. Pattern Recognition, 1993, 26(2):269-276.

[13] 陈志刚,宋胜锋,李陆冀,等.基于相似原理的点特征松弛匹配算法[J].火力与指挥控制, 2006,31(1):49-51.

[14] Chang S H, Cheng F H, Hsu W H, et al. Fast algorithm for point pattern matching: Invariant to translations,

rotations and scale changes[J]. Pattern Recognition, 1997, 30(2):311-320.

[15] 张立华, 徐文立.点模式匹配[J].计算机学报, 1999, 22(7):740-745.

[16] Chang S H, Cheng F H, Hsu W H, et al. Fast algorithm for point pattern matching: Invariant to translations, rotations and scale changes[J]. Pattern Recognition, 1997, 30(2):311-320.

[17] Spirkovska L, Reid M B. Robust position, scale, and rotation invariant object recognition using higher-order neural networks[J]. Pattern Recognition, 1992, 25(9):975-985.

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

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

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

[21] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces[C]// IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE, 2008:1-10.

[22] 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.

[23] Mair E, Hager G D, Burschka D, et al. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test[M]// Computer Vision – ECCV 2010. Springer Berlin Heidelberg, 2010:183-196.

[24] 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, 2016, 33(5):1255-1262.

[25] 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.

[26] Galvez-López D, Tardos J D. Bags of Binary Words for Fast Place Recognition in Image Sequences[J]. IEEE Transactions on Robotics, 2012, 28(5):1188-1197.

[27] 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.

[28] 汪神岳,刘强,王超然,等.基于双目立体相机的室外场景三维重建系统设计[J].计算机测量与控制,2017(11):137-140.

[29] Valgma L. 3D reconstruction using Kinect v2 camera[D]. Tartu Ülikool, 2016.

[30] Varanasi S, Devu V K. 3D Object Reconstruction Using XBOX Kinect V2. 0[J]. 2016.

[31] Kuschk G, Cremers D. Fast and Accurate Large-Scale Stereo Reconstruction Using Variational Methods[C]// IEEE International Conference on Computer Vision Workshops. IEEE, 2014:700-707.

[32] 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.

[33] 雷成, 胡占义, 吴福朝,等. 一种新的基于 Kruppa 方程的摄像机自标定方法[J]. 计算机学报, 2003, 26(5):000587-597.

[34] 江盼. 三维重建技术在城市空间数据采集中的运用——以 Altizure 运用为例[J]. 城市建设理论研究(电子版), 2017(32).

[35] Lourakis M, Argyros A. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm[R]. Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece, 2004.

[36] Ranade S, Rosenfeld A. Point Pattern Matching by Relaxation[J]. Pattern Recognition, 1980, 12(4):269-275.

[37] Li S Z. Matching: Invariant to translations, rotations and scale changes[J]. Pattern Recognition, 1992, 25(6):583-594.

[38] Schönberger J L, Frahm J M. Structure-from-Motion Revisited[C]// Computer Vision and Pattern Recognition. IEEE, 2016.

[39] Lindeberg T. Scale Invariant Feature Transform[M]// Scholarpedia. 2012:2012 - 2021.

致　　谢

在论文即将完成之际，谨向教导和帮助过我的师长、同学表达我真挚的谢意。在三年的课题研究过程中，我所取得的进步和成果都离不开计算机视觉研究所，更离不开我的导师陈胜勇教授，以及张剑华副教授的悉心指导和帮助。是他们为我创造了良好的学术氛围，教会了我严谨的治学态度和对研究工作的执着。

在项目完成过程中，导师的指导让我学会了如何分析问题看出本质，并找到解决问题的有效方法。在本文的撰写过程中，也是他们给予了我宝贵的意见，让我顺利的完成和取得最后的成绩。

同时，感谢计算机视觉研究所的全体师生们，其中我的师兄师姐们谢臻、冯余剑、任亲虎、冯缘、步青、王其超、汤帆扬、万富华、张少波等。也感谢我的同窗应高选、王燕燕、黄积晟、王超、金佳丽、邱叶强等，以及我的师弟师妹们吴佳兴、丁福光、陈宏峰等，感谢他们陪我度过了难忘的研究生三年，一起做项目，一起讨论问题。

最后要感谢我的家人，这么多年一直支持我理解我帮助我。并且在经济上无私的支持我，让我能够毫无顾虑的完成研究生的学业。同时也希望父母能够身体健康，万事如意。也祝大家能够在人生的旅途中一帆风顺。

攻读学位期间参加的科研项目和成果

录用和发表的论文

[1] Mao L, Wu J, Zhang J, et al. Monocular Dense Reconstruction Based on Direct Sparse Odometry[M]// Computer Vision. 2017.