

适用于无人驾驶车辆的地图构建与道路提取算法研究

于华超

2018 年 3 月

中图分类号：TP391.41

UDC 分类号 : 621.3

适用于无人驾驶车辆的地图构建与道路提取算法研究

| | |
|---------------|-------------------|
| 作 者 姓 名 | <u>于华超</u> |
| 学 院 名 称 | <u>自动化学院</u> |
| 指 导 教 师 | <u>王美玲</u> |
| 答 辩 委 员 会 主 席 | <u>孙长胜教授</u> |
| 申 请 学 位 级 别 | <u>工学硕士</u> |
| 学 科 专 业 | <u>控制科学与工程</u> |
| 学 位 授 予 单 位 | <u>北京理工大学</u> |
| 论 文 答 辩 时 间 | <u>2018 年 3 月</u> |

Research on map building and road extraction algorithm for intelligent vehicles

Candidate Name: Huachao Yu

School or Department: School of Automation

Faculty Mentor: Meiling Wang

Chair, Thesis Committee: Prof. Changhai Sun

Degree Applied:

Major: Control Science and Engineering

Degree by: Beijing Institute of Technology

The Date of Defence: March, 2018

适用于无人驾驶车辆的地图构建与道路提取算法研究

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

签名：

日期：

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签名：

日期：

导师签名：

日期：

摘要

无人驾驶依赖高精度地图完成导航规划等任务，而易于获取的卫星影像图由于精度差、更新慢等原因无法使用，需要设计适用于无人驾驶的地图，辅助无人车导航规划。在无人车路径规划中，道路信息决定无人车辆的决策方案，但是地图中的道路信息无法直接应用于无人车辆规划，为了提高地图使用效率，需要从已有的地图中自动提取道路信息，形成拓扑道路网络，为无人车规划提供先验信息。

针对这一问题，旨在从一系列图像中产生一个二维或三维地图，同时提取其中的道路区域，使得无人车辆可以依据地图提供的先验信息完成决策规划任务，本文从二维、三维地图的构建与二维图像中道路的提取这几个方面展开论述。

首先，本文介绍使用全景图像构建二维地图的方法，将全景相机与高精度定位系统固定于无人车上，采集全景图像与其对应的车辆定位信息，将全景图像由球面坐标系转换到车体坐标系中，得到以无人车为中心的俯视效果的逆透视投影图，而后依据定位信息将一系列逆透视投影图拼接为大范围二维场景地图；基于此，在GIS数据库中构建包含道路拓扑网络的二维地图，并添加道路的各种属性信息，作为无人车导航决策的依据。其次，本文提出使用多视图几何知识从航拍图像中重建地理空间三维结构的方法，提取所有航拍图像数据集中图片的特征点，根据特征点描述子匹配不同图片特征点，采用对极约束过滤错误的匹配点，据此将无序图片“有序化”；而后使用两张图片初始化，得到3D点坐标与相机位姿，增量式加入其余图片；在相机位姿与3D点坐标的优化过程中，本文除了最小化3D点的重投影误差，还增加了最小化相机中心距离，优化相机位姿与稀疏点云；并通过计算深度图与深度图融合得到稠密点云用于无人车导航规划。最后，本文提出一种从二维图片中提取道路的方法，使用最小二乘法拟合道路区域得到二值图，开闭运算等算法用来平滑二值图，而后采用拓扑细化等方法得到单像素构成的道路区域，最后去除单像素道路中的小枝丫得到最终的拓扑路网。

本文最后介绍了地图构建、道路提取的实验平台搭建与实验精度等结果，通过实验在不同场景下验证本文提出的方法的有效性。

关键词：SfM，三维重建，无人车，地图，道路提取

Abstract

This is abstract.

Keywords:

目录

| | |
|--------------------------------|----|
| 第1章 绪论 ······ | 1 |
| 1.1 研究背景与意义 ······ | 1 |
| 1.2 国内外研究现状 ······ | 3 |
| 1.2.1 地图构建方法研究 ······ | 3 |
| 1.2.2 道路提取方法研究 ······ | 6 |
| 1.3 主要研究内容 ······ | 7 |
| 1.3.1 二维与三维地图构建 ······ | 8 |
| 1.3.2 二维图像中的道路提取 ······ | 10 |
| 1.4 本章小结 ······ | 10 |
| 第2章 基于全景图像的高精度地图构建 ······ | 11 |
| 2.1 引言 ······ | 11 |
| 2.2 全景图像与定位信息采集 ······ | 11 |
| 2.3 全景图像处理 ······ | 11 |
| 2.4 地图的构建与使用 ······ | 14 |
| 2.4.1 图像拼接 ······ | 14 |
| 2.4.2 基于GIS数据库生成无人车行驶地图 ······ | 16 |
| 2.5 本章小结 ······ | 17 |
| 第3章 基于无人机航拍图像的三维构建 ······ | 18 |
| 3.1 引言 ······ | 18 |
| 3.2 相机模型 ······ | 18 |
| 3.3 相关坐标系 ······ | 20 |
| 3.4 特征点提取 ······ | 25 |
| 3.5 特征点匹配 ······ | 29 |
| 3.6 多视图重建 ······ | 34 |
| 3.6.1 两视图重建 ······ | 34 |
| 3.6.2 三角定位法 ······ | 39 |
| 3.6.3 2D-3D位姿求解 ······ | 40 |
| 3.7 三维重建中的优化 ······ | 42 |
| 3.7.1 光束平差法 ······ | 42 |
| 3.7.2 最小化相机中心位置误差 ······ | 45 |

| | |
|---------------------------------|----|
| 3.8 稀疏点云的稠密化 | 47 |
| 3.8.1 深度图计算 | 47 |
| 3.8.2 深度图融合 | 50 |
| 3.9 本章小结 | 50 |
| 第 4 章 二维图像中的道路提取 | 51 |
| 4.1 引言 | 51 |
| 4.2 图像二值化 | 51 |
| 4.3 非道路区域移除 | 52 |
| 4.3.1 开闭运算 | 52 |
| 4.3.2 轮廓提取 | 53 |
| 4.4 拓扑处理 | 54 |
| 4.4.1 拓扑细化 | 54 |
| 4.4.2 去除小枝丫 | 56 |
| 4.5 本章小结 | 57 |
| 第 5 章 实验结果分析 | 58 |
| 5.1 引言 | 58 |
| 5.2 全景图像构建地图实验结果 | 58 |
| 5.2.1 全景图像地图实验平台搭建 | 58 |
| 5.2.2 全景图像地图构建效果及精度分析 | 58 |
| 5.3 地理空间三维重建实验结果 | 60 |
| 5.3.1 地理空间三维重建实验平台搭建 | 60 |
| 5.3.2 三维重建效果及精度分析 | 60 |
| 5.4 道路提取精度分析 | 64 |
| 5.5 本章小结 | 65 |
| 总结与展望 | 67 |
| 参考文献 | 69 |
| 攻读硕士学位期间发表论文与研究成果清单 | 73 |

第1章 绪论

1.1 研究背景与意义

近年来智能移动机器人越来越普及，这些智能移动机器人可以独立或辅助人类完成很多任务，这依赖其对环境的感知、导航和运动规划等过程。作为轮式移动机器人，地面无人车辆在未知区域导航时，对感知、规划等的要求也很高，所以一般的无人车辆都依赖详细的地图进行导航规划^[32]，完成既定任务。而易于获取的卫星图像，由于更新不及时，室内无法使用，精度难以满足车道线级别的精度要求等原因不被高精度导航的无人车辆采用。目前移动机器人领域构建高精度地图的方法较多，一般采用车载或机载传感器等设备，总体上可以分为基于摄像头和基于激光雷达两种方案。

近年来实时定位与地图构建(SLAM, Simultaneous Localization and Mapping)作为移动机器人构建地图的方法获得长足发展，较为著名的有ORB-SLAM^[27]，LSD-SLAM^[28]，LOAM^[29]等，图 1.1 是运用车载传感器得到的三维地图，其中左图使用的传感器是激光雷达，右图使用的传感器是单目摄像头。这种采用车载传感器的构图方法已经发展地较为成熟，但是运用车载传感器得到大范围地图仍有困难，这是由于车载传感器视野受限，易受遮挡，当灾区发生道路阻塞时，车辆由于无法通行而无法采集道路信息，而且使用车辆对所有道路遍历的方法效率低下。然而无人机飞行灵活，视野宽阔，可以高效地对地面拍摄并通过算法建立地理空间地图，为无人地面车辆行驶提供先验信息。

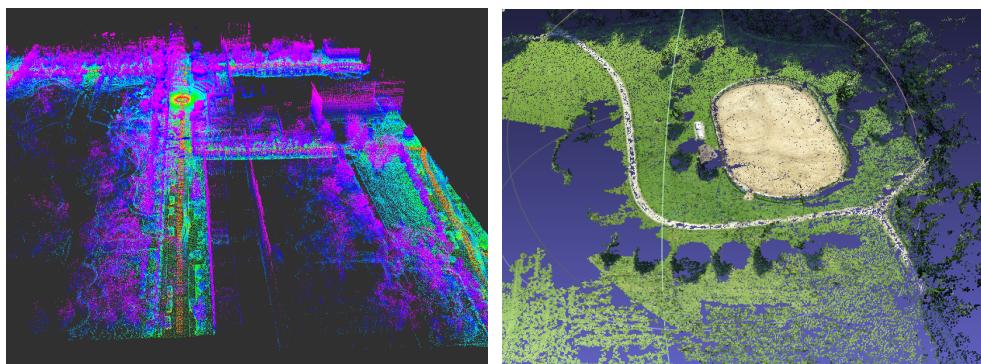


图 1.1 高精度地图

与视觉SLAM类似，SfM(Structure from Motion)也是一种构建三维场景的技术，SfM能通过2D图像得到场景3D 结构，然而SfM更多使用无序图片，运行过程不受时

间限制，从而构建比SLAM更精确的3D地图。与传统的航空激光扫描(ALS, Airborne Laser Scanning)和LiDAR(Light Detection And Ranging)绘制地图相比，SfM可以得到更高分辨率的包含空间纹理的3D模型，更重要的是，SfM得到如此丰富的数据结果使用的设备更廉价。这是由于SfM使用算法匹配图像特征点，计算相机位姿，从而重建拍摄场景的“稀疏”3D点云，而后通过MVS(Multi-View Stereo)将点云稠密化，并渲染得到逼真的三维地理空间模型^[30]，基于此SfM作为一种地形测绘技术被广泛应用于地理科学中。我们可以通过学术文献数据库Web of Knowledge观察SfM的学术研究情况，从1995年至2015年共有1000条关于Structure from Motion主题的记录，其中属于计算机科学范畴的最多，而属于地理科学范畴的记录数量排在第9名^[30]，如图 1.2所示。

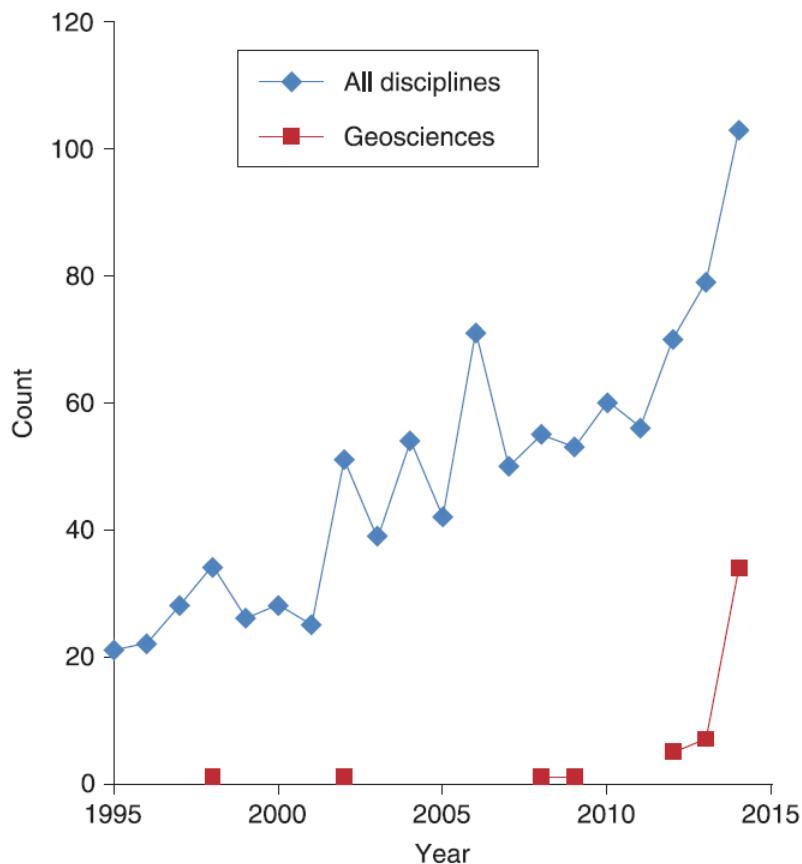


图 1.2 SfM主题统计曲线图

在无人地面车辆自主导航中，地图中的道路信息显得尤为重要，道路信息决定了无人车辆使用的规划模型和决策方案等，但是SfM构建的三维地图无法直接用于无人驾驶导航规划。所以需要提取地图中的可通行区域，也就是道路区域，为无人地面车辆导航提供先验信息。近年来道路提取跟踪在无人车导航、地形分类等领域广泛应用

用，但这些算法针对无人车视角的图像取得良好的效果，然而对于俯视视角的地图这些算法基本失效。这是因为无人车辆的相机朝前安装，其视野中道路一般收敛为消失点，针对消失点的提取道路的方法无法应用于大范围航拍地图。对于航拍地图中提取道路，需要分析道路区域的特征：

(1) 光谱特征：道路区域的光谱与大部分的非道路区域有着明显的不同，具体表现在道路区域的RGB通道值在三维空间的分布差异。

(2) 几何特征：道路区域通常是狭长的区域，道路宽度变化不会特别明显，而且道路的曲率也不会超过一定的限度。

(3) 辐射特征：对于单一的道路区域或者非道路区域，其颜色比较均匀，灰度值的方差较小；如果同时包含道路和非道路区域，则因为其光谱特征差异明显，灰度值的方差较大。

(4) 拓扑特征：一幅卫星影像图中的道路区域会呈现出一定的拓扑结构，并且会交叉形成结构化的道路网络，具有一定的拓扑特征。

目前，结构化道路网络的提取过程主要包括图像预处理、道路边缘提取、道路二值化、道路骨架提取等步骤。道路提取总体方案主要采用全自动或者半自动的方法，全自动方法不需要人工干预、速度较快，但是由于地图道路区域较为复杂，因此提取的效果欠佳；半自动的结构化道路提取^[31]将自动提取过程与人工干预相结合，在道路提取的初始时刻人工选取道路区域的初始点或者初始区域并进行迭代，或者加入人机交互，对通过自动的道路提取方法提取得到的道路网路进行后续的处理，完善自动提取得到的道路骨架，但人机交互也降低了道路提取的效率。

采用SfM构建三维地理空间地图，为无人车未知区域导航提供先验信息，利用道路区域的特征，提取道路区域，通过提取的道路区域构建无人驾驶的拓扑网络，为无人驾驶路径规划导航服务。

1.2 国内外研究现状

1.2.1 地图构建方法研究

构建适用于无人车辆行驶的地图是无人驾驶领域的关键技术之一，是进行路径规划与自主决策的重要信息源。许谦^[26]利用3D激光雷达和摄像头获取无人车环境信息，将得到的环境感知质点模型与环境感知区域添加到GIS数据库中，实现基于GIS的地图动态更新的过程。宋晓虎^[25]构建基于OpenStreetMap的电子导航地图，将SRTM高程

数据巧妙添加到原始OSM地图信息中，合并多层平面金字塔图与多级精度高程图获得立体地图。国内发力自动驾驶领域的百度公司使用地图车绘制高精度地图，为无人车自动自动驾驶服务，其使用高精度定位系统，结合高精度3D激光雷达绘制三维场景，并使用算法从激光点云中提取车道线，使高精度地图达到车道级别的精度，这种方法也被广大无人驾驶厂商采用，但是地图车价格昂贵，同时该方法不适合野外等条件的地图采集。

近年使用SLAM技术进行实时地图构建的方法被越来越多的无人驾驶研究机构采纳，李玉^[24]使用车载3D激光雷达获得稠密点云信息，采用迭代最近点(Iterative Closest Points, ICP)算法求取相邻帧3D点云的旋转平移关系，并评估得到的拼接点云中每个粒子的优劣，选择最优粒子加入全局地图，通过匹配拼接激光雷达数据获得地理空间的三维结构。段华旭^[22]采用双目视觉基于ORB特征提取和BoW模型闭环检测的方法，对地理空间进行三维重建，后端通过高斯-牛顿迭代法最小化三维场景的重投影误差，在优化重投影误差过程中考虑误差函数雅各比矩阵的稀疏性，从而加快整个优化过程。张毅^[21]围绕环境构图相关问题，提出了环境建图中基于粗集理论的数据融合算法，探索了构图过程中二维地图全新表示方法。邹智荣^[20]提出动态环境下自主车SLAM技术，针对大部分SLAM构图是基于静态环境完成的，本文提出了使用数据关联技术消除动态障碍干扰的方法，区分动静障碍，并通过实验验证动态SLAM的有效性。

由于无人机的视角原因，无人机的地图构建更加灵活高效，谢斌^[19]提出使用无人机通过D-S证据理论在未知环境构建适航地图的方法，该方法有效提高无人机对地面目标跟踪的准确性。刘洋^[18]为解决无人机路径规划问题，提出改进的概率地图模型，使用蚁群算法增加概率地图窄通道的采样点数量，获得可以覆盖规划环境的概率决策地图。陈含欣^[17]采用自适应重采样FastSLAM算法，利用无人机进行视觉定位与3D构图。蒙山^[16]提出一种基于几何特征的视觉三维重建SfM方法，使用单目相机基于二维直线匹配生成三维空间直线，并通过点线特征结合及直线增强算法减少三维空间表达的冗余信息。马跃龙^[15]借助SLAM技术得到无人机航拍视频关键帧间的相机关系，据此得到关键帧间的单应性关系，最后得到二维图像的拼接结果形成正摄投影地图。杨详红^[14]针对传统被动SLAM构图低效性，在SLAM中加入规划的思想，使构图更加高效。

在移动机器人地图构建领域，国外的研究起步较早，Leonard^[13]提出使用移动机

机器人的机载传感器实现同时定位与构图的过程，虽然使用的传感器是水下声呐，但与当前视觉、激光SLAM算法相似。A Typiak使用SICK单线激光雷达，测斜仪，相机等传感器，通过多传感器信息融合构建地图，提高无人车远程控制的性能。Cole^[12] 使用3D激光雷达采集数据，使用简单的分割算法将点云分类并将点云转换到车体坐标系下，作者提出了一种新的点云配准算法，可以更鲁棒的将点云拼接为三维地图。Diosi^[11] 使用激光雷达和声呐为移动机器人大构图导航，使用问答式的人机交互为地图添加语义特征，同时使用Kalman滤波器融合雷达与声呐信息，创建带有语义特征的地图为机器人导航。近来随着计算机性能的提升，基于视觉的三维重建技术得到很大发展，使用网络上大量图片构建场景的三维结构的相关研究中，Snavely^[10] 的工作对后来的研究者影响较大，Snavely设计了交互式可浏览查看图片的3D重建系统，拥有良好的用户体验，系统解决了网络图片涉及的相机模型、水印、清晰度、光线等因素多样性带来的问题，完成场景建模得到稀疏点云图。著名的“一日重建罗马”^[9] 就是通过网络上不同类型的图片重建罗马的各个景观，使用并行计算的方法提高运行效率，一天内可处理超过15万张照片，图 1.3 展示重建得到罗马三维结构。

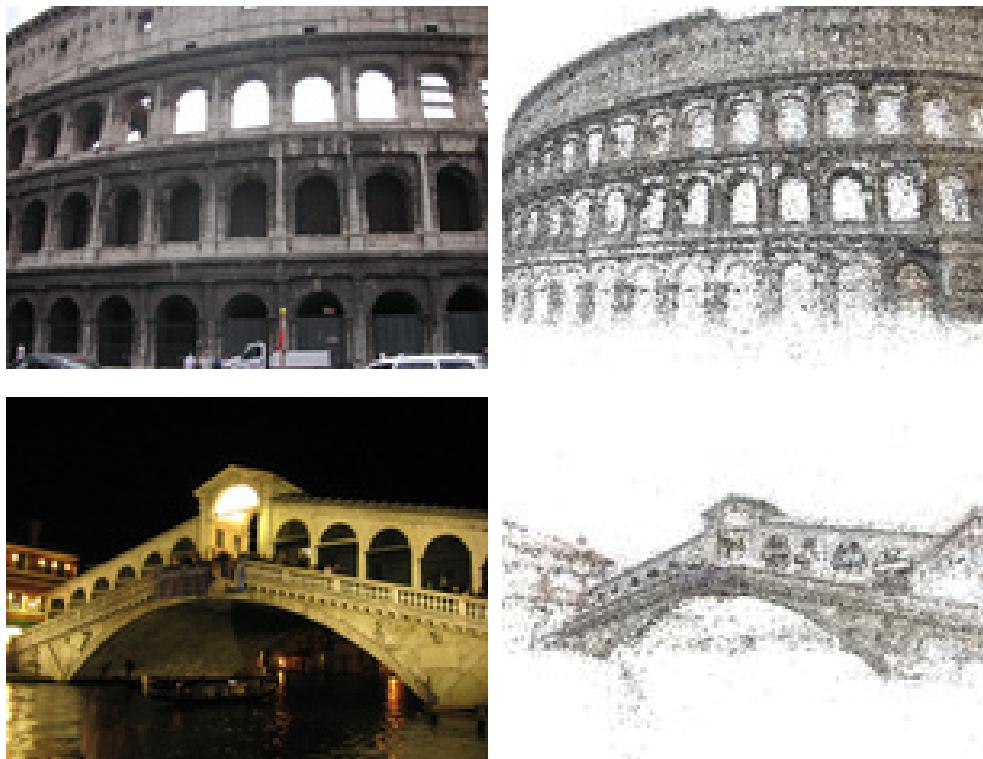


图 1.3 Building Rome in a day

针对SfM耗时效率低的缺点，Wu Changchang^[8] 提出线性时间的增量式SfM算法，

针对增量式SfM时间复杂度 $O(n^4)$, 其中 n 为照片数量, 提出了一种新的光束平差(BA, Bundle Adjustment)优化方法, 使时间复杂度降为 $O(n)$, 同时兼顾构图准确性。同时Wu Changchang^[7]为加速SfM使用GPU优化BA过程, 通过PCG(Preconditioned Conjugate gradients)改进BA, 最大化利用CPU进行并行计算等。

从以上研究现状可以看出构建地图的方法多种多样, 没有统一的标准格式, 甚至为满足特定需要而构建的地图在其他机器人上无法使用, 但是总体而言可以分为两类, 一种是使用激光雷达, 一种是使用双目视觉或单目视觉构建场景地图。3D激光雷达具有测距准确, 精度高等特点, 同时对点云分类算法的改进使实时激光构图, 并对点云分类成为可能, 但是激光雷达价格昂贵, 制约了其大范围商业化应用。使用视觉传感器, 无论双目单目甚至三目基本原理相似, 可以用于实时导航构图, 但要获得稠密而精确的点云, 类似图 1.3所示, 一般需要离线处理, 因为其中涉及到匹配优化问题很难在现有计算条件下完成。

1.2.2 道路提取方法研究

道路作为移动机器人导航规划的重要依据, 无论事先提取道路添加到地图中作为机器人规划的先验信息, 还是实时提取道路区域引导移动机器人导航, 道路提取都是至关重要的。在这方面的研究很多, 由于遥感图像视野开阔, 包含信息丰富, 使用遥感影像提取道路的研究很多, Wang Xia^[6]采用K-Means聚类方法对卫星图像进行分类, 得到关于道路区域的二值图像; 之后通过基于拓扑细化的形态学方法, 对二值化图像进行处理, 得到了道路骨架。汪夕明^[5]采用模糊C均值(Fuzzy C-Means, FCM)聚类的方法对卫星影像图二值化, 对于图像中的交通流干扰, 采用形态学重构和Mean-Shift方法加以去除, 对于阴影部分采用HSV变换加以去除, 实现了对有干扰图像的道路提取。魏宇峰^[4]应用SVM算法对遥感图像进行分类, 并提出一种改进的小波变换算法提取道路, 取得较K-mean更好的效果。小型航拍无人机易于获得, 对其视野中的道路提取方法也很多, Yucong Lin^[3]提出了一种从无人机航拍图像中快速而鲁棒地提取道路的方法, 作者使用梯度直方图检测道路区域, 进而使用概率Hough变换和聚类算法提取更精细的道路区域, 实验表明该方法可以提取野外沙漠环境中97%的道路区域。Wegner^[2]提出一种新的高阶条件随机场模型用来提取航空图像中的道路, 首先提取超像素构成的道路区域, 而后使用CRF(Conditioned Random Field)优化图割结果, 实验证明该方法可以更好地保留道路的拓扑结构, 更好地区

分道路与非道路像素。Shenlong Wang^[1]针对半自动地图构建，以及未来无人车更加依赖地图信息的趋势，提出以OpenStreetMap(OSM)数据为输入，采用深度学习方法，检测并标记地图中车道，人行路，停车位等各种信息，精炼地图；针对航空图像与车载传感器输出图像的特点，融合二者提取的道路区域，相对于GPS+IMU系统，这种融合道路区域的方法有更好的精度。Shenlong Wang^[32]提出使用航空图像道路网络改进更新开源地图的方法，采用马尔科夫随机场模型对道路中线与道路宽度建模，对OSM中的图片进行道路提取，据此可以纠正OSM道路图层的错误；作者也通过结构SVM方法训练OSM数据集，得到人行道、绿化区域的模型用于提取测试集的道路相关区域。图 1.4展示了从OSM中提取道路的结果。



图 1.4 Enhancing Road Maps

遥感图像提供了对地理空间更全的覆盖区域，但是由于分辨率较低，提取道路难度较大；随着航空飞行器的普及，使用小型低空飞行器获取地面影像，并从中提取道路区域更加灵活方便。从以上研究现状分析，道路提取的方法虽各不相同，但是都利用道路特点——光谱特征、几何特征等。首先提取初始道路区域，而后根据道路细长的特点得到道路区域；也有使用部分道路作为训练集，利用深度学习的方法让机器自己学习道路特征，而后使用该模型提取道路区域。尽管经过多年的研究，道路检测的方法很多，但是提取道路仍然不是件容易的事，大范围自动提取道路仍然是一个有待解决的问题。^[32]

1.3 主要研究内容

三维重建技术由来已久，尤其广泛应用于移动机器人导航、定位与决策中。然而移动机器人直接使用三维地图存在困难，因为移动机器人很难在三维地图中分辨可通行区域以及障碍物等信息。而道路作为引导机器人导航的重要信息，从地图中提取道路变得愈发重要。基于这样的背景，本课题的研究内容主要包括以下几点：

- 融合高精度定位信息与视觉传感器信息，构建适用于无人车的二维道路地图，为无人车执行全局路径规划提供先验信息。
- 使用无人机三维重建技术，离线构建高精度三维地理空间地图，为未知环境的无人车导航提供地图信息。
- 提取二维地图中的道路区域，构建道路拓扑网络，依托GIS软件为无人车导航规划提供依据。

1.3.1 二维与三维地图构建

传感器选取

在开始研究之前，本文先讨论研究所涉及到的传感器选型问题。如图 1.5 所示是通常使用的视觉传感器、3D 激光雷达，图(a)是双目相机，双目相机可以通过左右摄像头计算得到场景深度图，但不如图(c)激光雷达得到的点云精度高，但是相对激光雷达更廉价。而且随着计算机视觉的发展，相信利用双目可以得到更理想的结果。图(b)是全景相机，可以获取周围 360° 场景，相机模型较为复杂。图(d)是最常见的单目相机，相机的成像可以近似抽象为小孔成像模型，处理其成像较全景相机更方便，但无法像双目相机直接得到景深图，单目相机尺度未知问题可以通过算法解决。



图 1.5 传感器示例

针对以上分析，结合本文的实际使用需要，本文设计的构建二维地图方法旨在为无人车导航提供先验信息，无人车通过定位信息匹配到二维地图的某区域后，无人车即可通过地图知道当前的车道、限速、甚至距离路口远近以及是否有红绿灯等信息，这些先验信息将为无人车导航规划提供巨大帮助。三维数据存储、运算等耗费资源，所以在二维地图构建中倾向于选择全景相机，这也是考虑到单目相机视野有限的原因。

三维地图虽然构建复杂、存储空间较大，但是三维地图可以提供更多的地理空间信息，随着计算机硬件与算法的发展，3D地图必将是未来无人车地图发展的方向。由于本文要使用无人机视觉的方案构建三维场景地图，在传感器的选取上只能选择单目相机，因为无人机飞行高度较高，此时双目相机的基线基本上可以忽略，两个双目相机的输出基本相同，无法解算得到景深图。

二维地图构建

二维图像由于操作简单，存储方便等原因被许多无人车采用，本文为了获取视野更大的二维图像选取全景相机，将全景相机架设在无人车辆顶部，可以得到无人车周围环境的全景图。全景相机成像在球面坐标系中，其中车道线扭曲严重，所以本文将全景图投影到地面坐标系中，使车道线之间保持平行，得到类似俯视视角的由全景视频变换后的视频流。通过高精度定位系统，将每帧视频流对应的车辆定位坐标信息编码为二维码放置于每帧视频的右下角，如此设计一套采集地图系统。解算采集的视频流中每帧视频对应的二维码信息为位置、航向，而后根据位置航向将图片旋转平移后拼接成完整的路网地图。

以拼接的路网地图为底图，通过自动或半自动的方式构建道路的拓扑网络，拓扑网络由点、线以及道路属性（车道数量、限速、是否路口等信息）构成，为无人车导航决策提供依据。

三维地图构建

三维地图由于包含的信息丰富，在仿真与实车实验中应用广泛，本文构建三维地图的方法是使用无人机搭载的单目摄像机拍摄地理空间的二维图片，将一系列无序的二维图片作为输入，重建地理空间三维场景。首先对所有图片提取其特征点，将二维图片矩阵变为一维特征点向量，而后匹配特征点，将无序图片通过匹配点联系起来，

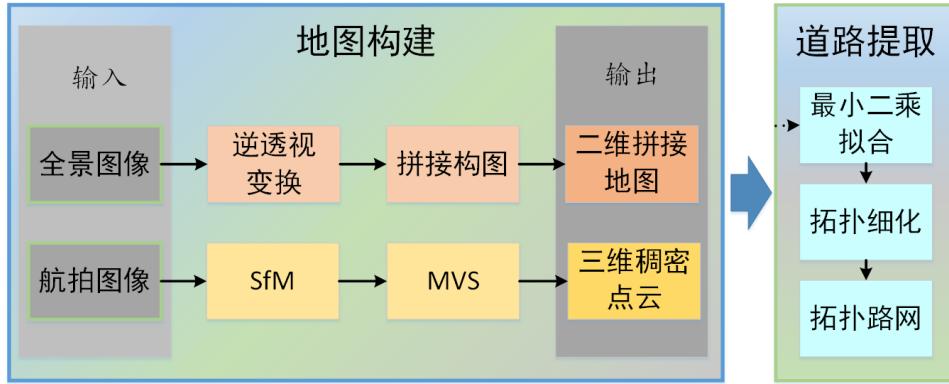


图 1.6 本文研究内容

使用多视图几何相关知识恢复匹配点的三维坐标和相机位姿，最后采用光束平差法优化稀疏点云与相机位姿。采用Multi-View Stereo(MVS)算法计算每幅图片的深度图，融合深度图得到完整的稠密点云。

1.3.2 二维图像中的道路提取

道路检测对无人驾驶至关重要，本文采用最小二乘法拟合测试集道路区域的光谱特征，从而得到初始的道路区域，用二值化的图片表示，但是二值图存在非道路区域的错误检测，本文采用开闭运算平滑二值图，并采用矩形检测等方法移除开闭运算无法消除的大面积的非道路区域，至此可以得到较为理想的道路二值图。在此基础上，通过拓扑细化及短枝丫剔除的方法得到单像素构成的道路拓扑图片，使用GIS软件可以直接生成拓扑路网，用于无人车导航规划。

1.4 本章小结

本章概述了课题研究背景与意义，针对无人驾驶导航与规划高度依赖地图的现象，构建适合无人驾驶的地图变得十分关键。针对这一问题，本文通过查阅当前地图构建的研究现状，设计了新的适合无人驾驶使用的二维地图与三维地图构建方法。同时由于道路信息在无人驾驶导航规划中有着重要意义，提出从二维地图中提取道路网络，构建道路拓扑，为无人车全局路径规划提供先验信息。本文研究总体内容可以用流程图 1.6 表示。

第2章 基于全景图像的高精度地图构建

2.1 引言

全景图像（panorama）是指水平角度包含完整360°的相机得到的二维平面图像。相对于普通单目相机，全景相机可以完整的呈现无人车周围环境。在本章节中，讨论使用全景相机构建车道级别的用于无人车行驶的高精度地图的方法。首先介绍了全景图像与定位信息的采集，而后介绍了全景相机的标定与全景图像预处理，最后阐述了如何通过全景图像构建大范围的高精度地图的方法。

2.2 全景图像与定位信息采集

为了同时采集全景图像信息与高精度定位导航系统信息，本文采用将车辆的定位信息编码为大小一定的QR code (Quick-Response code)，放置于每帧全景图像左上角位置，完成系统的信息采集任务。在此介绍一下二维码的编码与解码过程。由于全景图像的帧率为7Hz，惯性导航系统的帧率为20Hz，考虑拼接图像的精度，在没有使用硬件触发的条件下，本文采用最近邻原则，即当接受到一帧全景图像时，编码离当前时刻最近的惯性导航定位信息。由于编码的信息有时间、经纬度、方位角、偏航角与俯仰角6个整数，所以需要的编码信息量很小，本文只采用 25×25 像素的二维码编码定位信息。

QR code作为一种非常流行的矩阵式二维码，具有空间利用率高，存储数据量大，解码速度快，可以包含字符，数字等不同内容等优点。如图 2.1所示，二维码可分为两部分：功能区和编码区。功能区包含用于二维码定位的位置探测区域，用于校正二维码位置的校正图形区域等，在编码区中，包含数据和纠错码等。二维码的纠错级别越高，则实际存储的数据信息就越少。按照QR code的规则，可以很方便的将定位信息编码为固定像素大小的二维码，进而将该大小的区域放置在每帧全景图像的左上角，完成信息的采集过程。当回放该视频时，可以很容易地对大小固定的编码区进行解码，读取到每帧全景图对应的车辆当前的位置与角度信息等，为图像拼接做准备。

2.3 全景图像处理

Ladybug全景相机具有6个独立的摄像头，如图 2.2所示，可以在球形坐标系下将6个相机图像实时合并为一张全景图，呈现在二维平面空间中。ladybug相机提

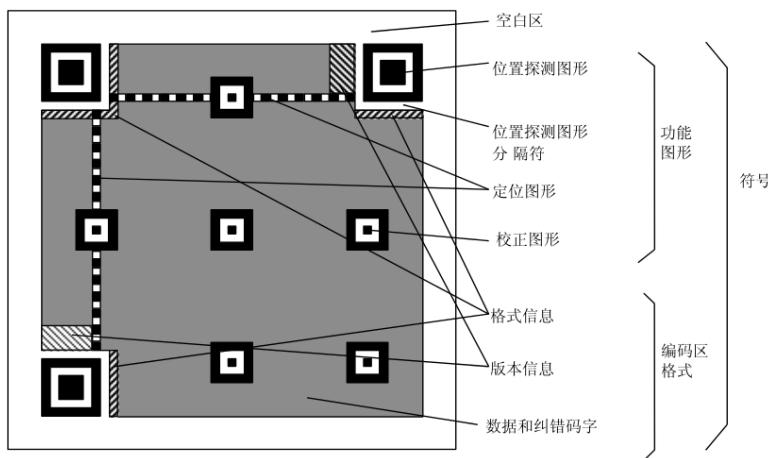


图 2.1 QR code示意图

供360°视频串流功能，可以覆盖90% 的可视球面，具有图像采集、处理和校正功能。在得到Ladybug相机的全景图后，使用逆透视投影（IPM）将全景图像投影成俯视图。而后将车辆的高精度定位信息编码为二维码，放置于对应的逆透视投影得到的图片的左下角，从而完成图像与对应车体位置的采集与处理，方便后续构建大范围地图。



图 2.2 Ladybug全景相机

由于LadyBug成像描述在球面坐标系中，IPM投影需要表示在车体坐标系中，地图拼接需要在世界坐标系中完成，在此简单介绍全景图像处理过程涉及的相关坐标系的变换关系。如图 2.3所示，代表全景相机的球面坐标系与车体坐标系的关系，其中在车体坐标系中，绕自身x, y和z旋转，分别代表俯仰角，横滚角和方位角，使用道路方向形象地代表车辆的行驶方向；在全景坐标系中，球面狭缝与全景图分块成对应关系，如图中虚线箭头所示，得到的全景图的长宽，如图 2.4所示，图片长

宽分别代表相对于车体的方位角 μ 与俯仰角 ν 。为了得到全景图像的IPM俯视图，需要标定全景相机相对于车辆的外参：俯仰角 θ ，横滚角 ϕ 和方位角 φ ，以及Ladybug相机光心与地面的高度 h ，同时假定车辆周围的环境与地面高度一致。现定义车体坐标系为 $\{F_\omega\} = \{X_\omega, Y_\omega, Z_\omega\}$ ，坐标原点位于车辆后轴中点；定义全景相机坐标系为 $\{F_c\} = \{X_c, Y_c, Z_c\}$ ；定义全景图像坐标为 $\{F_i\} = \{\mu, \nu\}$ 。如图 2.3所示，车体坐标系中任一点与全景相机坐标系中的点的对应关系为：

$$P_c = R_\phi R_\theta R_\varphi^T (P_\omega + T) \quad (2.1)$$

其中

$$R_\phi = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \quad (2.2)$$

$$R_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad (2.3)$$

$$R_\varphi = \begin{bmatrix} \cos(\varphi - \frac{\pi}{2}) & \sin(\varphi - \frac{\pi}{2}) & 0 \\ -\sin(\varphi - \frac{\pi}{2}) & \cos(\varphi - \frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$T = \begin{bmatrix} 0 \\ 0 \\ -h \end{bmatrix} \quad (2.5)$$

进而可以得到全景图像坐标系中的坐标 $P_i = [\mu_i, \nu_i]^T$ ：

$$P_i = \begin{bmatrix} \arctan \frac{y_c}{x_c} \\ \arccos \frac{z_c}{\|P_c\|} \end{bmatrix} \quad (2.6)$$

其中 $\|P_c\| = \sqrt{X_{P_c}^2 + Y_{P_c}^2 + Z_{P_c}^2}$ ，即 P_c 的二范数。通过以上算法可以将全景图像（图 2.4）变换得到车体坐标系下的IPM图，显示效果如图 2.5所示，图中长宽对应真实地

理空间30m。

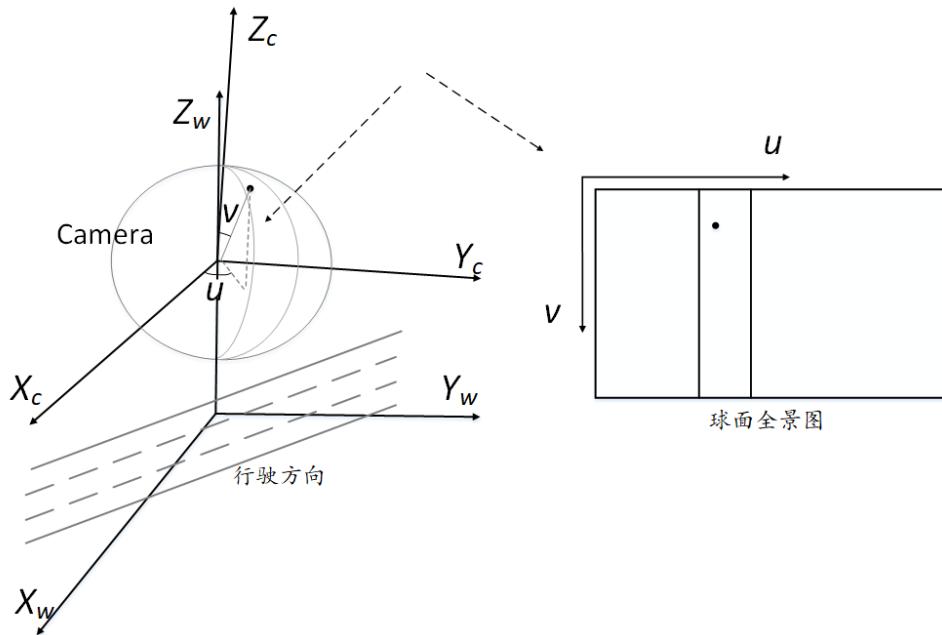


图 2.3 逆透视坐标变换关系



图 2.4 基于球面坐标的全景图

2.4 地图的构建与使用

使用以上的方法可以得到全景图像的逆透视投影图，以及每张投影图对应的此时车辆位置与方位等信息。进而我们可以得到大范围地图信息，制作大范围行驶地图，为无人车规划与感知提供先验信息。

2.4.1 图像拼接

图像拼接的历史可以追溯到上个世纪，使用的方法也不尽相同，总体上可以分为两类，一类是基于特征匹配的图像拼接方法，另一类是基于图片相互间位置关系的拼



图 2.5 逆透视变换结果

接方法。由于本文采用的惯性组合导航系统，在无遮挡，多路径效应较弱的空旷场地，实验精度达到 5cm 以内，所以在此本文采用基于图片之间相互位置关系的方法。具体就是将视频流中每帧逆透视投影图片按照二维码解算的航向旋转一定角度，然后按照解算的位置平移一段距离，对所有视频按照上述方法处理，最终得到完整的城区地图。

对于每帧IPM俯视图，车身为无效信息且占据图像大量位置，对地图拼接造成干扰，所以在拼接地图前，需要将车身移除。由于全景图变换为IPM投影图时，将IPM俯视图变换到车体坐标系中，所以任何一张IPM俯视图的车辆位置与方位是固定的：车辆后轴中心位于图片横轴的中心，纵轴靠下的 $\frac{1}{3}$ 位置处，航向朝上。如图2.6所示。据此设计一个mask，在拼接过程中将车辆去掉，只拼接剩下的像素不为零的部分。

在拼接过程中，因为所需拼接的地理空间过大，图片过多，而内存限制了单张图片的大小，所以本文设计了区号标记的方法。具体做法为：将视频流中图片按照位置和航向拼接为多张 28000×7000 像素的图片，如图2.6中橙色方框所示，由于每个像素代表实际地理空间的大小为 $0.06m$ ，即每张拼接后的图片代表的物理空间大小为 $1680m \times 420m$ 。设置整个地图的坐标原点（图2.6红色圆点所示）为某一固定经纬度坐标，IPM图对应的经纬度与该点对比后，计算出该帧IPM图应该放置于哪个区，而后可以拼接成多张橙色大小的图片。一个橙色方框代表一个区域，对应一个区号，这种存在重叠区域的区号法则有效的解决了图像拼接过程中出现的边缘缺失的问题，所以即使橙色方框代表的区域出现边缘缺失现象，由于重叠区域的存在可以有

效解决该问题。最后只需要保留图 2.6 的绿色框表示的区域和该该区域的区号，这样就可以按照区号和图 2.6 的样式将拼接的图片在 Photoshop 中拼接为更大的图片，为后续 GIS 数据库中构建无人车底图服务。

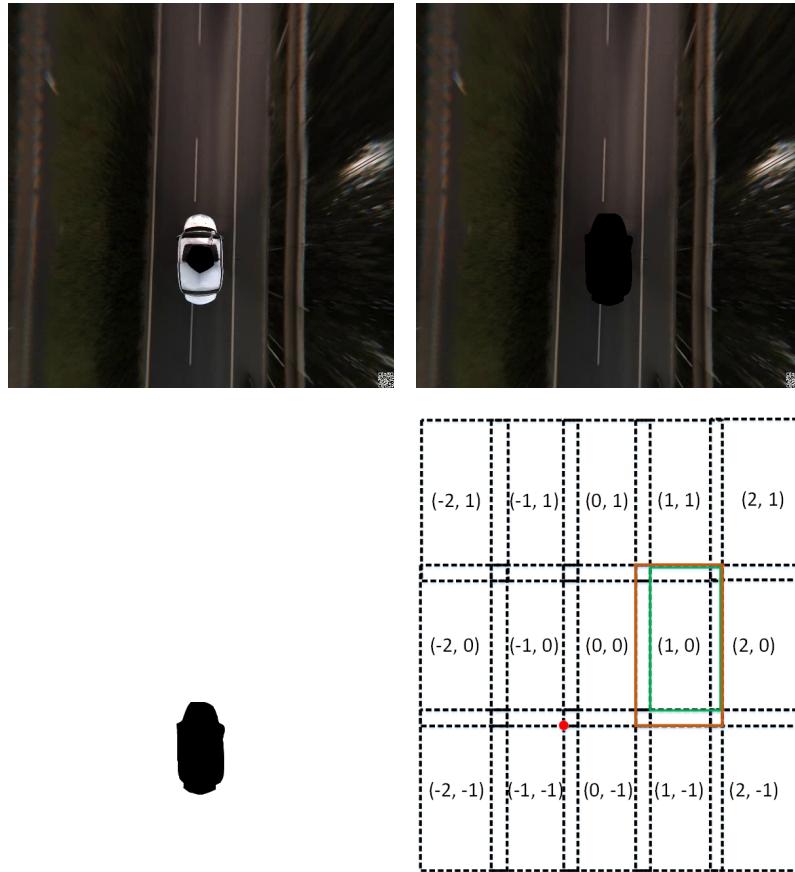


图 2.6 去掉车辆

2.4.2 基于GIS数据库生成无人车行驶地图

在得到数张 Photoshop 拼接的更大的图片的同时，按照图 2.6 的分区原则也很容易得到每张图片四个顶点的地理坐标。依托 Supermap(GIS) 软件，可以很容易完成金字塔地图的生成，作为 GIS 数据库的一个图层。如图 2.7 所示。这样构建的地图并不能直接用于无人车导航或规划，我们将得到的底图作为 GIS 数据库的一个图层，根据底图可以得到车道级别的道路拓扑结构，该拓扑作为 GIS 数据库的另一个图层，如图 2.7 中的绿色线段，在线段与线段的节点上添加各种属性，例如车道数量，限速，红路灯和路口等，即可实现全局路径规划等功能，这些功能不是本文讨论的内容。

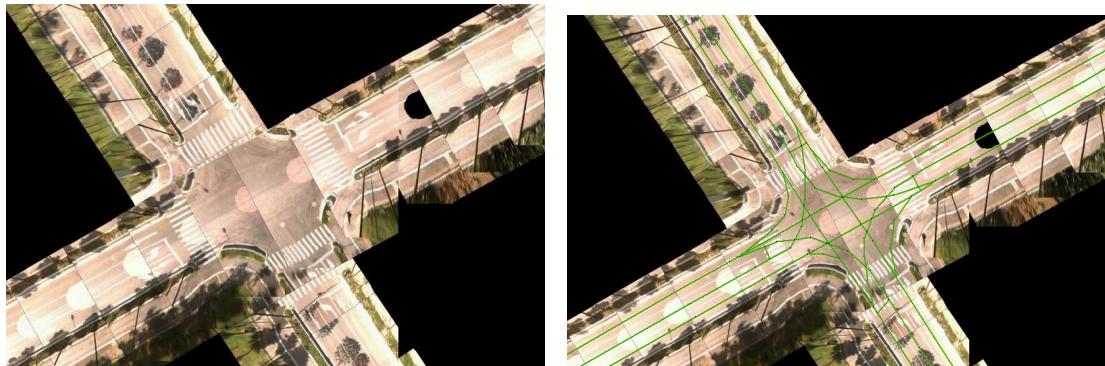


图 2.7 拼接的路口展示

2.5 本章小结

本章介绍了如何通过Ladybug全景相机生成的球面全景图与高精度定位导航组合，构建车道级别地图的过程。根据Ladybug与车体的固定位置关系，将全景图像旋转与平移转换到车体坐标系中，并完成逆透视投影的过程，并将每个IPM 投影的图像通过车辆的高精度定位导航信息，旋转平移拼接为完整的道路地图，最后将地图生成GIS数据库的图层，完成道路拓扑构建与道路属性添加的任务，即可将该地图用于无人驾驶导航与规划等任务中。

第3章 基于无人机航拍图像的三维构建

3.1 引言

大部分无人车辆行驶依赖环境地图导航与感知^[32]，这限制了无人车在未知环境中的行驶。为了解决该问题，当前许多技术利用车载传感器构建大范围地图，例如SLAM(Simultaneous Localization and Mapping) 和第二章提到的用车载Ladybug构建地图的方法。然而车载传感器仅能获取车辆周边环境信息且易被阻挡，而且在野外或灾害地区等环境中，车辆可通行区域变少。然而无人机拥有比无人车更广阔的视野，不受地面障碍物的影响，可以灵活地从空中俯瞰地面。本章介绍了应用无人机构建三维环境地图的方法，为无人车导航提供先验信息。

三维重建涉及到的内容可以分为两部分：稀疏点云的获取与点云稠密化。其中稀疏点云的获取使用的是Structure from Motion(SfM)，在技术层面上与当前热门的Visual-SLAM(VSLAM)比较类似，只是SfM侧重于三维重建，而SLAM侧重于实时定位，并需要预测下一时刻的位置。SfM可以分为视觉前端与优化后端，前端涉及到的是相机模型，坐标转换，特征提取与匹配以及多视图几何相关的内容，最后得到粗略的相机位姿与3D点坐标，后端涉及到的是优化前端得到的相机位姿与3D点坐标，去除错误的结果等。点云稠密使用的技术称为Multi-View Stereo(MVS)，MVS的实现方式多种多样，本文采用深度图融合的方法，对每张图片寻找参考图，而后根据参考图与SfM恢复的相机位姿计算该张图片的深度图，最后融合所有图片的深度图形成稠密点云。

3.2 相机模型

数码相机拍摄实际是一个光学成像过程，这涉及到摄像机最基本的原件——透镜——的成像原理，如图 3.1所示，这是最基本的透镜成像原理：Z是物体距透镜光心的距离，简称物距；f是焦距；b是相距，即成像平面与透镜光心的距离。三者满足式3.1。

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{b} \quad (3.1)$$

在机器视觉中，利用摄像机可以将三维场景记录在二维平面上，不同的相机模型导致不同的成像效果，比如第二章中Ladybug使用的球面成像模型，使不同相机的拼

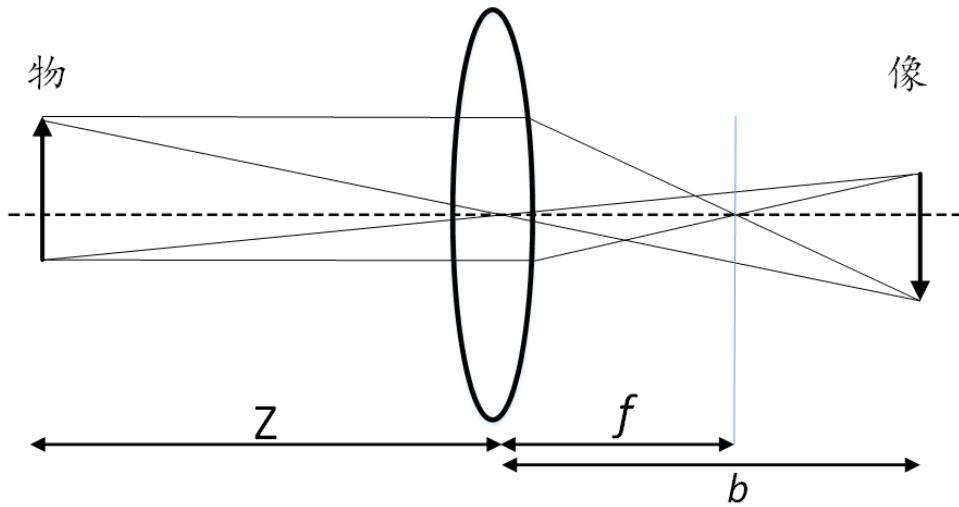


图 3.1 透镜成像原理

接简化为绕光心的旋转。但是常见的相机模型还是小孔成像模型，如图 3.2 所示。数码相机的镜头相当于一个凸透镜，感光元件就处在这个凸透镜的焦点附近，将焦距近似为凸透镜中心到感光元件的距离时就成为小孔成像模型。这可以类比为艺术家画一幅画的过程，将眼睛放在图 3.2 光心 C 处，在物体与眼睛之间放置一个画布（图 3.2 中虚像位置），按照光线直线传播的原则，则物体反射的光线与眼睛的连线交画布一点，如此物体可以在画布上成像。只是相机的成像在光心后面，所以说存在一个虚拟成像平面。

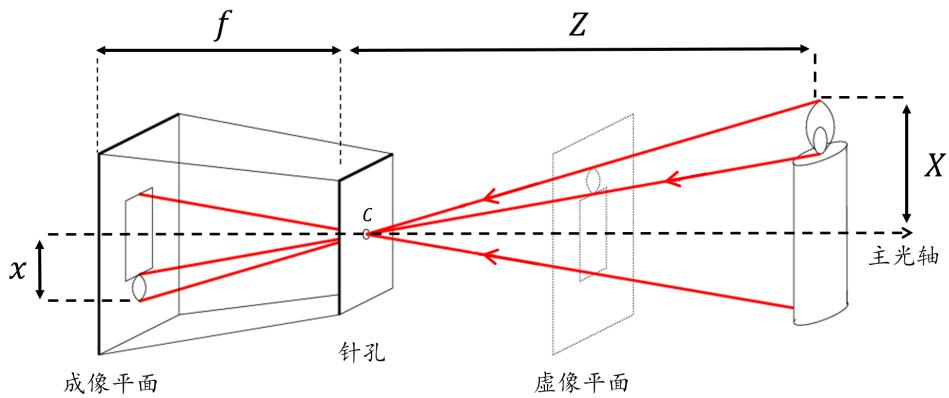


图 3.2 小孔成像模型

在三维空间中，物体反射的光线经过小孔形成倒立二维图像，根据其数学模型，可以得到

$$\frac{-x}{X} = \frac{f}{Z} \quad (3.2)$$

其中, f 是小孔到成像平面的距离, 即焦距; Z 是物体距光心的距离, 简称物距; x 是物体在成像平面的投影长度; X 是物体实际长度。为了简化以上模型, 用图 3.3 表示小孔成像的等价模型, 其中 C 是相机中心, $P(X, Y, Z)$ 为空间中的 3D 点, $p(x, y, 1)$ 表示 3D 点成像在感光元件上的点, 小孔成像近似后, 成像平面与相机中心的距离为焦距 f 。该图也表示了两个坐标系, 分别为图像坐标系和像素坐标系, 在下节中详细介绍。

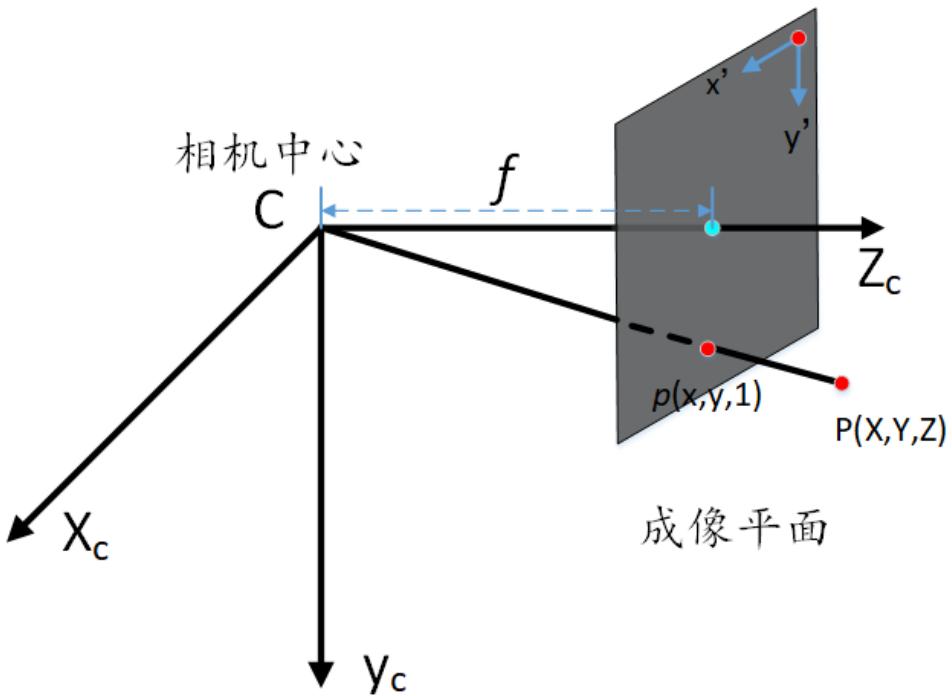


图 3.3 小孔成像等价模型

3.3 相关坐标系

从上一节知道, 仅仅相机内部就存在多个坐标系, 那么摄像机在空间中涉及到的坐标系转换问题将在这节详细探讨。相机成像的过程是一个 3D 物体显示在 2D 成像平面的过程, 在此涉及物体, 真实空间与相机三者的位置关系。物体在真实空间的位置即是物体在世界坐标系中的位置坐标表达, 在图 3.3 中, (X, Y, Z) 为 3D 点 P 在相机坐标系 (X_c, y_c, Z_c) 下的坐标表示, 这里就存在两个三维坐标的转换, 即同一个点在世界坐标系与相机坐标系下表达方式的转换关系, 如式 3.3, 该变换可以看做三维坐标系的刚体变换。

$${}^cP = {}^wR_w^cP + {}^cT_w \quad (3.3)$$

其中 c 表示相机坐标系, w 表示世界坐标系, wP 表示世界坐标系下 P 点3D坐标, cP 是相机坐标系下 P 点3D坐标, cR_w 和 cT_w 分别表示由世界坐标系到相机坐标系的旋转矩阵和平移矩阵。

将世界坐标系下的3D点表示为齐次坐标 $(X, Y, Z, 1)$, 式3.3可以改写为:

$${}^cP = \begin{bmatrix} {}^cR_w & {}^cT_w \\ 0^{1 \times 3} & 1 \end{bmatrix}_{4 \times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.4)$$

其中 cR_w 是 3×3 矩阵, 三个列向量俩俩正交; cT_w 是 3×1 矩阵, 代表两个坐标系原点连线的向量, cP 坐标表示为 (x_c, y_c, z_c) 。

从图3.3中可以看到成像平面内的二维坐标系 x', y' , 该二维坐标系称为图像坐标系, 可以推导相机坐标系与图像坐标系(3D到2D)变换的公式表示:

$$x' = f \frac{x_c}{z_c} \quad (3.5)$$

$$y' = f \frac{y_c}{z_c} \quad (3.6)$$

改为矩阵表达形式为

$$z_c \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (3.7)$$

相机在生成图片的过程中, 由于成像平面由许多CCD感光元件组成, 所以成像的横纵坐标是不连续的, 引入像素的概念, 同时也存在从图像坐标系到像素坐标系(2D到2D)的转换关系:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{s_x} & s & p_x \\ 0 & \frac{1}{s_y} & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (3.8)$$

其中, dx 是u轴方向单像素的宽度, dy 是v轴方向单像素的宽度, 当前大部分ccd的像素宽度, $dx = dy$, 只有以前老式电视机使用的是矩形ccd, 导致 $dx \neq dy$ 。 p_x, p_y 是主点(图3.3)与图像左边界的距离与 p_y 是主点与图像上边界的距离。 s 是衡量主光轴与成像平面的倾斜程度, 如果主光轴垂直于成像平面, 则 $s = 0$ 。这些参数被称作相机内参, 一般可以通过查询相机手册得到包括焦距在内的相机内参。

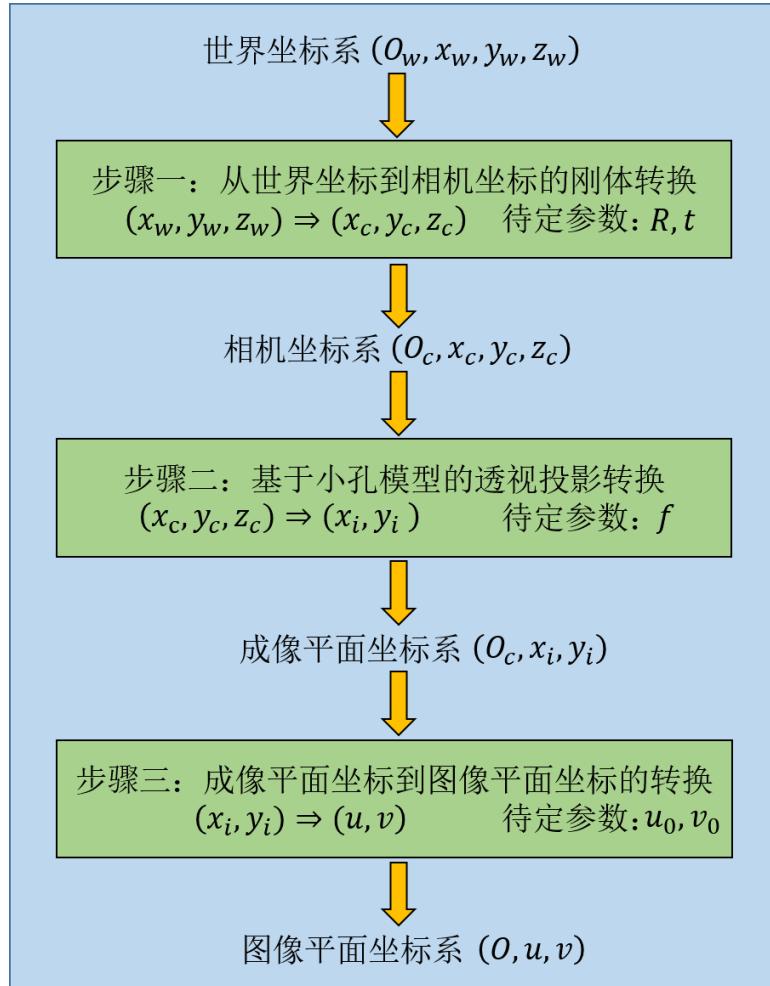


图3.4 坐标转换流程图

至此我们可以得到从世界坐标系到像素坐标系变化过程, 如图3.4和3.5。图3.4过程用公式表达为3.9

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{s_x} & s & p_x \\ 0 & \frac{1}{s_y} & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cR_w & {}^cT_w \\ 0^{1 \times 3} & 1 \end{bmatrix}_{4 \times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.9)$$

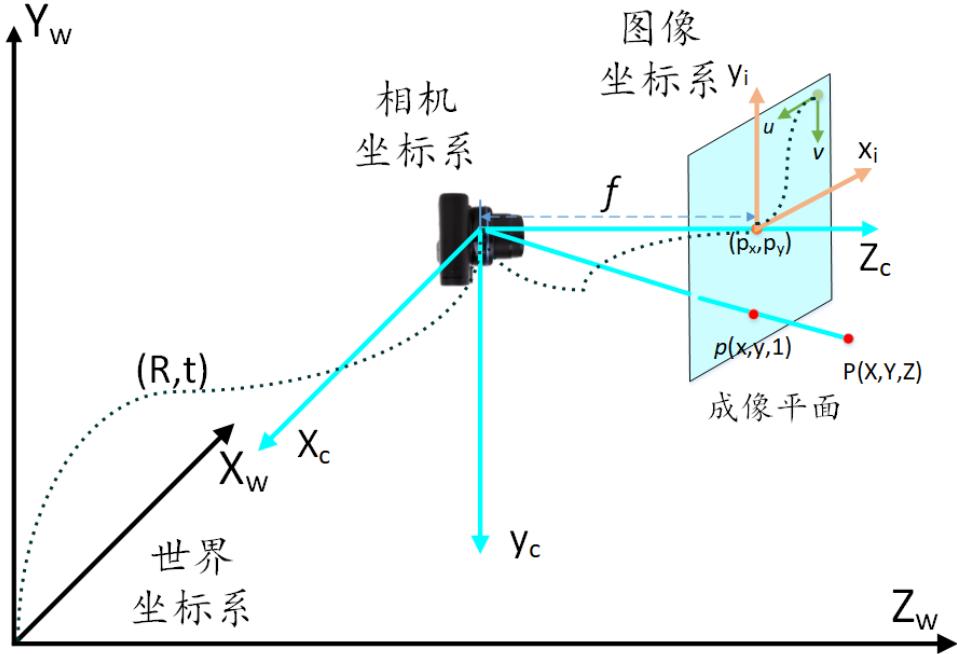


图 3.5 坐标转换关系图

至此我们知道了相机成像过程中某点如何在不同坐标系下转换，从而完成摄像过程。式 3.9 涉及到相机参数 f, p_x, p_y 以及空间变换相关的参数（外参 R, t ），实际上在相机成像过程中除了这些参数，我们还会遇到无法线性建模的参数：像 Gopro 等鱼眼镜头中，我们可以看到世界中的直线不再是直线，如图所示，这些弯曲的线有一些特别的特性：他们都是被关于中心扭曲的，我们称这种畸变为径向畸变。这意味着像素点是沿着从中心放射的径向成比例扭曲的。为了去除相机的畸变，需要对畸变用多项式进行建模，如式 3.10。

$$u^{dist} = u(1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots) \quad (3.10a)$$

$$v^{dist} = v(1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots) \quad (3.10b)$$

$$\text{where } r^2 = u^2 + v^2 \quad (3.10c)$$

其中， k_1, k_2, k_3 是未知参数，需要标定才可以得到，而实际校正径向畸变时一般也只用到 2 个或 3 个参数，如图 3.6 所示，左图为存在径向畸变的图片，右图为经过校正后的图片。



图 3.6 径向畸变的校正

在相机校正的过程中，实际上还需要校正图像的切向失真，这是由于摄像机安装时成像平面与透镜没有平行导致的，如果说径向畸变是坐标点距离中心点的长度放生了变化，那么切向畸变就是坐标点与中心点的水平夹角发生了变化。对切向畸变的建模如式 3.11，一般使用 p_1, p_2 两个参数。

$$u^{dist} = u + 2p_1uv + p_2(r^2 + 2u^2) \quad (3.11a)$$

$$v^{dist} = v + 2p_2uv + p_1(r^2 + 2v^2) \quad (3.11b)$$

$$\text{where } r^2 = u^2 + v^2 \quad (3.11c)$$

综上所述，本节详细介绍了相机模型与相机相关坐标系转换关系，以及相机成像的校正等相关内容，这部分属于图像处理的基础，熟悉该部分内容将为下面几节多视图几何相关内容作铺垫。以下 3.4-3.7 节的内容概况了本文 SfM 的主要流程，如图 3.7 所示，输入为无序单目航拍图像，输出为稀疏点云与相机位姿，以下将分步介绍各个环节。

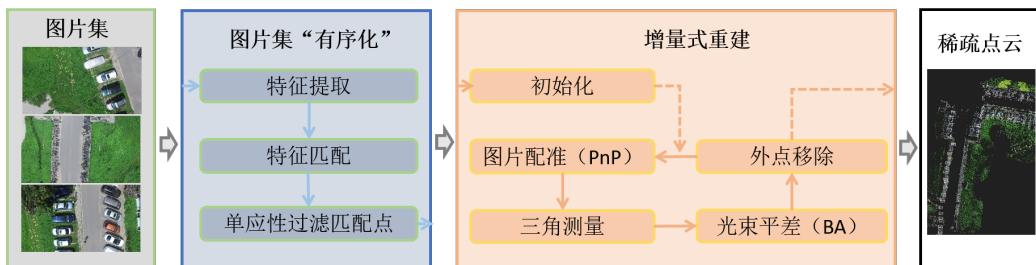


图 3.7 SfM 流程图

3.4 特征点提取

SfM首先需要找到图片集中俩俩图片间的几何关系，这样才能估算相机间的运动与成像的三维点坐标。为了找到图片之间的几何关系，现今有许多方法，例如光流法，特征点法等，光流法适用于匀速运动的视频流中，要求帧间运动比较小，从而找出帧间运动关系；特征点法指对图像特征的提取与存储，是图像中比较有代表性的特征。由于本文使用的数据集为无序的航拍图像，特征点法较为适宜使用。特征点提取更通俗的理解为将二维矩阵描述的图像降维为一维特征描述的图像，突出图像的重点，去除冗余信息的过程。

最简单的图像特征为图像的角点，角点就是图像中线的交点，例如大厦最高处的顶点。两幅图像的角点不会随着图像移动而改变，但是角点会随着离相机的距离太近而变得无法识别，所以许多研究者设计了精巧而互有优劣的特征检测与描述方法，比较著名的有SIFT^[33]，SURF^[34]，ORG^[35]等。相对于简单的角点，这些特征检测具有以下优点^[36]：

- 可重复：相同的特征可以在不同的图片中找到
- 可区别：不同的特征有不同的表示
- 高效率：特征点数远远小于像素数

实际上，特征点可以分为局部特征点和全局特征点，像上面举例的三个特征点实际上是局部特征点，局部特征点顾名思义就是基于图像的突出区域明显的区分图像的特征，一般而言，局部特征需要具有旋转不变性，光线明暗的鲁棒性等。因此图像可以被提取到的一系列称为感兴趣区域的特征描述。相反全局特征是将图像表示为一个向量，向量的值为图像的各个方面，如颜色，纹理或形状。例如像区分图像，一些是海洋，一些是森林，一个全局颜色描述子可以很好的对其归类。在这种情况下，全局描述子涉及图像所有像素的特定属性，这个属性可以是颜色直方图，纹理，边缘等。如图 3.8所示，使用哪种特征描述子取决于图像处理的情景。由于地理空间的航拍图像在颜色或者纹理等全局特征上，图片的相似度极高，不易辨识，难以找到图片之间的匹配关系，所以此文选用局部特征描述子：SIFT。

特征点提取包括特征点检测与特征点描述两部分，在此以SIFT特征点为例，简单介绍一下检测子与描述子的使用方法。检测子有很多，使用较多的有Harris

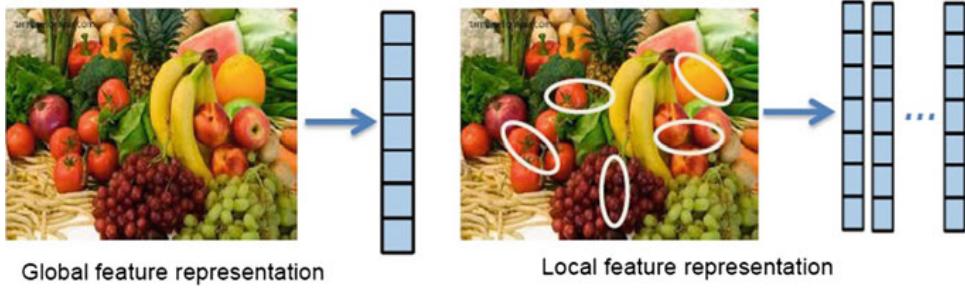


图 3.8 全局特征与局部特征描述

Detector^[38], FAST Detector^[37]。Harris检测子结合边缘与角点检测方法得到图像不同方向的自相关系数变化率,例如强度变化率等,该方法将局部特征描述为对称的自相关系数矩阵。Fast检测子考虑某像素周围一定大小的圆,院上像素强度与该像素比较,如果大于或小于某一阈值则定义该像素为角点,这两种检测子对尺度变化都不具有不变性,不适合直接应用在拍摄高度不同的航拍图形的特征检测中。而SIFT作为一种具有尺度、放缩不变,对不同光线鲁棒的特定,很适合应用于环境复杂的航拍图形的特征特取与描述中。SIFT算法由四个步骤组成:尺度空间极值检测,兴趣点(关键点)定位,方向计算以及兴趣点描述。第一个阶段是使用Difference of Gaussian(DoG)确认潜在的兴趣点,这里使用DoG代替Laplacian of Gaussian (LoG)算子以提高计算速度。

Laplacian of Gaussian (LoG)是二阶导数的线性组合,也是检测斑点(blob)的检测子,给出一个图片 $I(x,y)$,尺度空间 $L(x,y,\sigma)$ 是由图片 $I(x,y)$ 与不同大小的高斯核 $G(x,y,\sigma)$ 卷积得到的,表示为

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (3.12)$$

其中,

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.13)$$

据此可计算Laplacian算子为

$$\nabla^2 L(x,y,\sigma) = L_{xx}(x,y,\sigma) + L_{yy}(x,y,\sigma) \quad (3.14)$$

这对于大小为 $\sqrt{2\sigma}$ 的斑点有最强的响应,然而该算子严重依赖图像中斑点大小与用来平滑图像的高斯核大小,即大小不同的高斯核函数被用来寻找对应大小的blob。为了在一幅图像中自动检测不同大小的斑点,一种多尺度的方法被提出来^[39],其通过

尺度不同的归一化的Laplacian算子对图片进行平滑从而在不同尺度空间找到大小不同的blob，尺度归一化的高斯核函数为式 3.15。

$$\nabla_{norm}^2 L(x, y, \sigma) = \sigma^2 L_{xx}(x, y, \sigma) + L_{yy}(x, y, \sigma) \quad (3.15)$$

值较大的 σ 对应图片的粗略特征，较小 σ 对应图片的精细特征。同时LoG算子是对称的，所以它具有blob旋转对其没有影响，但是LoG算子耗费计算资源，为了加速计算并在尺度空间检测到稳定的兴趣点，Lowe^[33]提出了高斯差分尺度空间，简称DoG scale-space。如式 3.16所示，利用不同尺度的高斯差分核函数与图像卷积生成。

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (3.16)$$

计算DoG的关键步骤是构建图像金字塔：对于某个图像，使用大小不同的高斯核与该图像卷积，可以得到模糊程度不同的但是图像长宽一致的图像，这些图像构成了一个八度(octave)，而后对这一系列图像进行降采样，降采样就是对octave中的图像隔行隔列采集像素，最后图像尺度(scale)变为原来大小的 $\frac{1}{4}$ ，构成下一个octave。这样就避免了对不同尺度空间下的图像进行卷积，减少了计算。为了寻找尺度空间的极值点，在DoG尺度空间中，每个采样点要和其周围所有相邻点（8邻域中，一个像素点周围共有26个像素点）比较，若该点是极大值或者极小值则定位为兴趣点。整个过程可以用图 3.9描述。

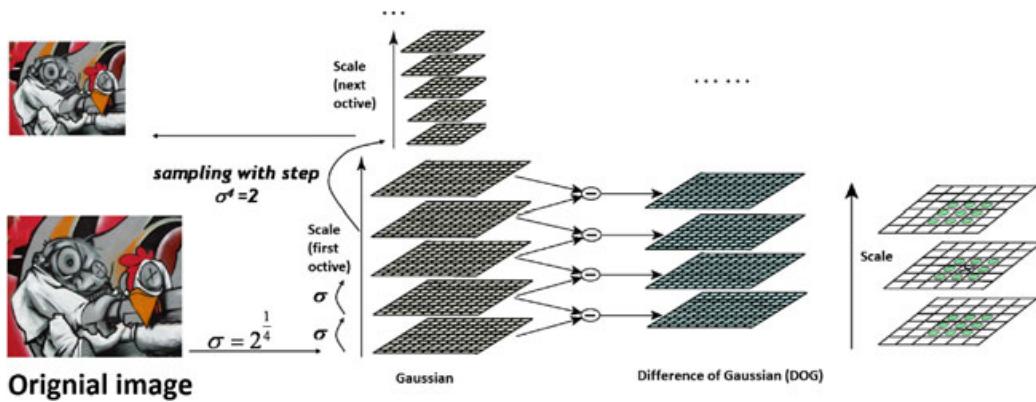


图 3.9 图像金字塔的构建

检测到特征点后，需要定量的表示各个特征点，以为特征点匹配做准备。Lowe在其论文中提出用一个128维度的向量表示每个特征点，具体做法为：如图 3.10，首先对每个关键点周围的 16×16 邻域内所有像素计算其梯度的幅值与方向，用直方图统计邻域像素的梯度方向，梯度直方图将 $0 - 360^\circ$ 分为8个区间，对得到的直方图进行高斯平滑，以减少突变带来的影响。选取直方图的峰值作为该关键点处邻域梯度的主方向。所以在每个 4×4 的象限内，将每个像素的主方向加权到直方图的8个方向区间中的一个，计算一个梯度方向直方图，这样对于每个特征点可以形成 $4 \times 4 \times 8 = 128$ 维的描述子。

至此基本介绍了本文使用的SIFT特征点提取的方法，详细介绍了特征的提取与描述方法，以上是使用SIFT特征点检测多张图像的结果，其中特征点主方向以及特征点的强度分别表示在图中的。。。

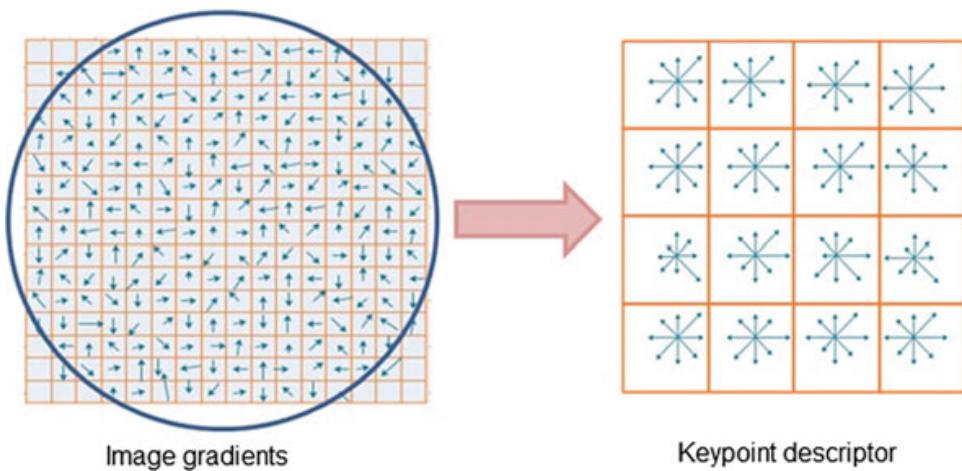


图 3.10 SIFT描述子的构建



图 3.11 SIFT特征点检测结果

3.5 特征点匹配

上面提到本文使用SIFT特征点作为关键点检测的方法，特征点提取并描述后，就应该找出各个图像中以特征点为依据的关系，即每张图像中的特征点如何与其他图像的特征点进行匹配，找到相近的特征点并去除错误的匹配的特征点。特征点匹配解决航拍图像之间的数据关联问题，将无序的图片数据集建立关联关系，为下文通过图像之间的关系，计算相机位姿与还原特征点的三维坐标提供基础。很好理解，一对匹配点就是实际空间的一个点，如果将所有图片对应同一个真实点的特征点连在一起，将使数据集中所有无序图片“有序化”。相同的，如果匹配点中有大量的错误匹配，那对后续的位姿估计与还原三维坐标都产生不利影响，所以如果去除错误匹配更是至关重要的问题。下面就这两方面展开讨论。

假设在图片集 $I = \{I_i | i = 1 \dots N_I\}$ 中找到的局部特征点为 $F_i = \{(x_j, f_j) | j = 1 \dots N_{F_i}\}$ ，其中 x_j 表示特征点位置坐标， f_j 表示特征点描述子，如果是SIFT特征点，则 f_j 是128维向量。对于 F_i 与 F_j 的特征点匹配的方法，最简单的是使用暴力匹配，即每个 $(x_j, f_j) | j = 1 \dots N_{F_i}$ 与 $(x_j, f_j) | j = 1 \dots N_{F_j}$ 中特征点测量两个向量之间的距离，需要进行 $N_{F_i} \times N_{F_j}$ 次距离比较。对于像SIFT这样的浮点描述子，一般采用欧拉距离作为衡量依据，而对于像BRIEF（ORB特征使用的描述子）二进制描述子，则一般使用汉明距离作为度量依据。

但是当图片长宽较大，分辨率较高时，图片特征点会很多，这时采用暴力匹配会导致效率很低，一般实际使用中采用近似最近邻(Approximate Nearest Neighbor, ANN)匹配方法。事实上，无论SIFT特征匹配还是数据库检索本质上是相同的，都是使用距离函数在高维矢量空间中寻找相近对象的过程。常用的方法试K近邻查询，设置查询点与正整数K，从另一个数据集中根据距离公式找到距离最近的K个数据，如果K=1，改方法变为最近邻查询算法。K近邻查询算法是通过构建KD-Tree或者R-Tree等实现的。在此不做详细介绍。

在此简单介绍近似最近邻的快速库(Fast Library for Approximate Nearest Neighbors, FLANN)，FLANN具有一种内部机制，该机制可以根据数据本身选择合适的算法来处理数据集，基于FLANN的匹配非常准确、快速。FLANN在使用的时候需要配置两个参数：indexParams和searchParams。使用FLANN时，indexParams可以选择为LinearIndex、KTreeIndex、KMeansIndex、CompositeIndex和AutotuneIndex，其中KTreeIndex灵活且可被并行处理。对于searchParams，用来指定索引树被遍历的次

数，即check值，该值越大则匹配的时间越长，当然也越准确。

当然即使用上面的方法，也无法保证所有的匹配都是正确的，Lowe^[40]在1999提出一种简单的方法可以剔除大部分的错误匹配。对于某个特征点向量，如果查询数据集中与其距离最近的特征点向量与次近的特征点向量的距离之比大于0.7，可以避免接近90%的错误匹配，但是正确的匹配也因此变少。使用FLANN并采用Lowe提出的方法可以去掉大部分的错误匹配，图 3.12展示了随机选取10个匹配对的结果。



图 3.12 FLANN匹配

以上提到的方法，匹配后仍然存在错误的匹配点，由于刚刚的匹配是单纯基于特征描述子的向量值，无法保证匹配的两个特征点对应相同的场景点。因此SfM使用投影几何的知识估计两张图像的特征点变换。依赖图像对的空间配置，不同投影描述图片间不同的几何关系。例如，单应性变换（又称射影变换^[41]）描述相机拍摄的二维图片之间纯旋转和平移变换。对极几何通过基本矩阵（Essential matrix） E 和基础矩阵（Fundamental matrix） F 描述移动相机的关系，并可以通过三焦张量扩展到三视图的关系。无论哪种变换关系，有效的变换可以满足大部分匹配点的几何关系，那么这种变换或者匹配点被认为是有效的。本文在删除错误匹配时，使用的是单应性变换，一下简单介绍一下单应性变换的详细内容。

单应性又称射影变换、保线变换，与坐标系无关。映射 h 是射影变换的充要条件是：存在一个 3×3 非奇异矩阵 H ，使得任意一个矢量 X 表示的坐标点都满足 $h(x) = Hx$ 。即

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.17)$$

矩阵 H 是一个齐次矩阵，在 H 的九个元素中有八个独立比率，即交比(cross ratio) $h_{11} : h_{12} : h_{13} : h_{21} : h_{22} : h_{23} : h_{31} : h_{32} : h_{33}$ 不会因为 H 因为乘以非零因子而变化，所以单应性变化有八个自由度，而每个二维特征点可提供2个方程，故需要解算 H 需

要至少4对匹配点。已知4对匹配点，使用直接线性法解算 H ，示意图如图 3.13，已知两个视图四对匹配的特征点坐标 X_1, X_2 ，求 H 的过程。线性变换表示为 $X_2 = HX_1$ ，

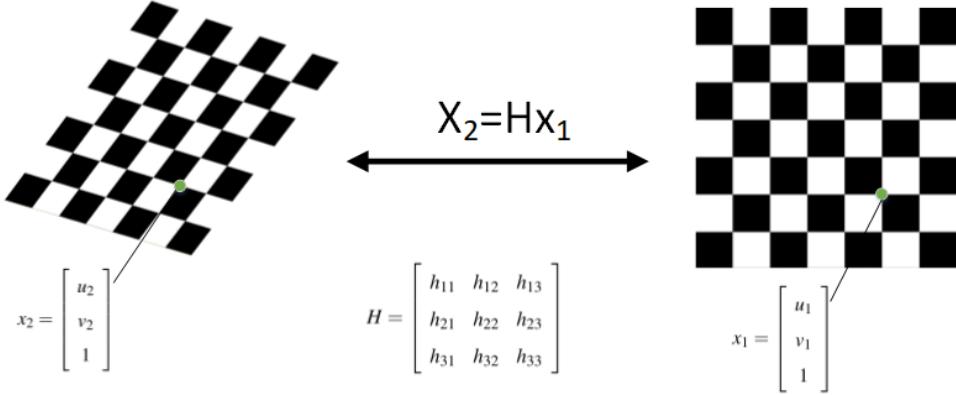


图 3.13 单应性变换

这是齐次矢量方程，三维矢量 X_2 和 HX_1 不相等，实际可以写为 $\lambda X_2 = HX_1$ ，线性变换可以变形为 $X_2 \times HX_1 = 0$ （矢量与自身外积为零），记为 $[X_2] \times HX_1 = 0$ ，设 h_i 为 H 的第*i*行，

$$\begin{aligned} \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \times \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} X_1 &= \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \times \begin{bmatrix} h_1 X_1 \\ h_2 X_1 \\ h_3 X_1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \times \begin{bmatrix} X_1^T h_1^T \\ X_1^T h_2^T \\ X_1^T h_3^T \end{bmatrix} \\ &= \begin{bmatrix} 0 & -1 & v_2 \\ 1 & 0 & -u_2 \\ -v_2 & u_2 & 0 \end{bmatrix} \begin{bmatrix} X_1^T & 0_{1 \times 3} & 0_{1 \times 3} \\ 0_{1 \times 3} & X_1^T & 0_{1 \times 3} \\ 0_{1 \times 3} & 0_{1 \times 3} & X_1^T \end{bmatrix} \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} \end{aligned} \quad (3.18)$$

最后一步是由线性代数的知识得到，即对3维矢量 $x = (x_1, x_2, x_3)$ 的 3×3 的反对称矩阵为式 3.20

$$[a]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.19)$$

合并 3.18 最后一步的前两项得到下式为

$$= \begin{bmatrix} 0_{1 \times 3} & -X_1^T & v_2 X_1^T \\ X_1^T & 0_{1 \times 3} & -u_2 X_1^T \\ -v_2 X_1^T & u_2 X_1^T & 0_{1 \times 3} \end{bmatrix} \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} = 0 \quad (3.20)$$

上式简写为 $Ab = 0$, 其中 A 为 3×9 矩阵, $\text{rank}(A) = 2$, 这是因为每队匹配点只提供两个量: u 和 v , 也可以从 A 矩阵的值看出, 将 A 的第一行乘以 u_2 与第二行乘以 v_2 相加得到第三行。 b 为 9×1 矩阵, 由于只需要维持 b 各项的交比不变即可, 所以有 8 个自由度, 故需要 4 对匹配点即可在这个线性系统上获得足够的条件计算得到单应性矩阵 H 的各项。由以上分析我们可以知道当给定的匹配点为 4 个时, 并将 $\|H\| = 1$, 方程有精确解,。但是大部分的匹配问题匹配点的数目远不止 4 对, 如果多于 4 对, 那么 $Ab = 0$ 是超定的, 如果所有的匹配点的位置是精确的, 那么 $Ab = 0$ 的解仍然是精确的, 但是通常情况匹配点中有噪声, 存在错误匹配等问题, 所以这时候方程存在最小二乘解 (超定解)。除了零解可以忽略外, 方程解可以使用 SVD 找到其近似解。

SVD 又称奇异值分解, 在此不详细介绍, SVD 可以将一个矩阵, 无论是否满秩, 均可以分解为 $A = UDV^T$ 形式, 其中 $UU^T = I$ 和 $VV^T = I$, 即 U 和 V 是正交矩阵, D 是对角阵, 对角元素非负。实际上 SVD 可以看做特征值分解的推广, SVD 在主成分分析, 求解方程的最小二乘解中广泛应用。回到求解单应性变换的问题来, 根据 SVD 求解最小二乘解的原理, 可以得到: 最小二乘解是 $A^T A$ 的最小特征值的特征向量。具体讲, 假设有 n 个匹配点, 则 A 为 $2n \times 9$ 的矩阵, 且 $A = UDV^T$, 且 D 对角阵元素按降序排列, 那么 b 是 V 的最后一列。这是由于 $Ab = 0$ 的解, 即 A 的零空间, 又由于 $\text{rank}(A) = n - 1$, 所以 A 的零空间只有一个基向量, 即 SVD 中最小特征值对应的特征向量 V 的最后一列。

奇异值分解是广泛用于图像处理中的一种算法, 不仅在求单应性矩阵时使用, 下文亦有涉及。但是这里要介绍一种本文使用到的另一种方法——RANSAC(Random Sample Consensus), 该方法在图像领域中也广泛应用, 其根本原理是使用统计原理, 可以通过多次迭代估计含有噪声的数据模型的参数。RANSAC 中文译为随机抽样一致性, 其一般过程为从一组含有外点 (噪声) 的数据中随机选取数个数据, 估计模型参数, 然后将估计得到的模型应用于剩余的其他数据, 从而得到该模型的误差, 重复以上过程, 直到误差达到某设定阈值或迭代次数达到设定最大次数, 停止迭代, 选取

最优的一组模型作为最终结果。根据最终结果与事先设定的阈值可以过滤一部分外点并得到较为精确的模型。由以上内容知，估计单应性变换需要4对匹配点即可计算一个单应性变换，所以RANSAC每次选取4对匹配点迭代计算得到单应性矩阵，而后将单应性矩阵应用于剩余的匹配点，计算误差，过滤外点。如图 3.14所示，其中 a,b 是仅根据SIFT描述子通过最近邻匹配过滤的结果。 c,d 是使用RANSAC方法通过单应性变换过滤的结果，从图中可以直观看出使用RANSAC可以过滤掉一部分错误的匹配，同样从表 3.1中也可以看出，其中对角数字表示图 3.14中三张图片提取到的SIFT特征点数量，其他表格数目，例如55/32表示使用RANSAC方法前得到的匹配点数目，以及使用RANSAC后过滤剩下的匹配点。

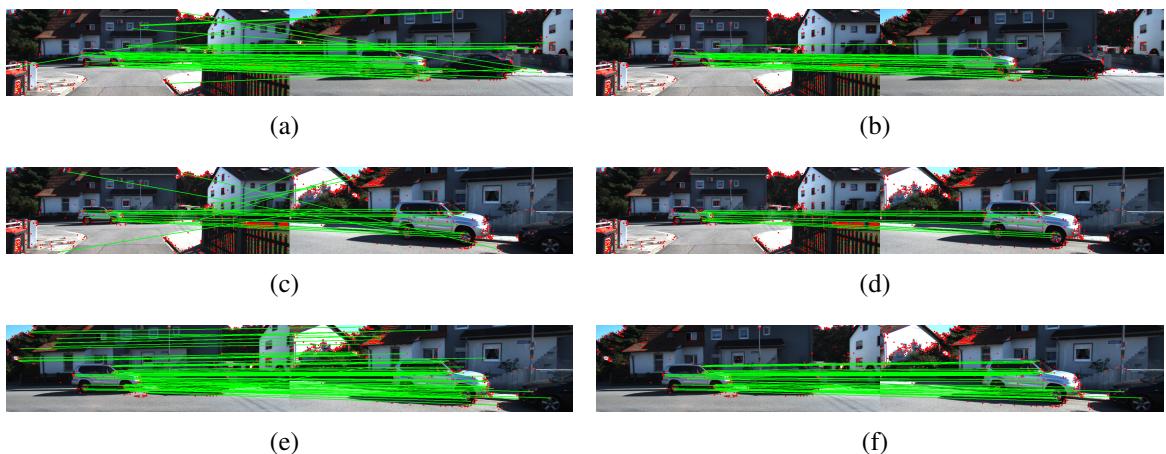


图 3.14 RANSAC筛选的单应性变换结果

表 3.1 匹配点过滤

| 图片名 / 特征点 | KITTI-0400 | KITTI-0415 | KITTI-0424 |
|-------------------|------------|------------|------------|
| KITTI-0400 | 861 | 55/32 | 75/46 |
| KITTI-0415 | | 370 | 19/10 |
| KITTI-0424 | | | 599 |

经过各种方法，得到的图片匹配点基本上是正确的，基于此我们可以对数据集中无序图片构建它们的关联结构，对于多张图片对应的同一个匹配点，设置统一标号，称为track。例如图片一中第125个特征点，图片二中第259个特征点，图片三中第6个特征点...匹配，实际上它们对应实际地理空间的同一个点，所以可以使用同一个track序号标记这些匹配点，这样做使整个数据集通过特征点匹配构成关联关系，为下面的多视图重建打下基础。

3.6 多视图重建

本节内容主要介绍如何通过多视图几何的只是，恢复上节匹配的特征点的三维空间结构，以及摄像机位姿等相关信息，本节内容采用增量式SfM的方法，首次从track点最多的两张图片入手，恢复两视图匹配特征点的三维空间位置与两视图对应的摄像机的相关关系，而后加入第三张图片，计算第三张图片与前两张图片得到的3D点的关系，使用三角测量方法还原第三张图片的特征点3D位置坐标，如此循环。其中每加入一张图片使用光束平差法（Bundle Adjustment，简称BA）优化3D点与相机位姿。最终得到较为理想的点云信息。

3.6.1 两视图重建

在式 3.9 中， z_c 为每个像素的深度，在单目成像时， z_c 是未知的。所以当知道某点的世界坐标系时，可以很容易得到该点的像素坐标，但是反之无法已知像素坐标得到世界坐标。所以从单张图像中无法还原场景的三维结构，所以下文介绍从两张存在特征点匹配的图片中恢复场景的三维结构。两视图几何最核心的原理是对极约束，在阐述堆积约束相关知识之前，我们先复习一下 3.3 节中关于摄像机成像过程中涉及到的相关坐标系。如果三维空间的一个点 X ，其在相机所成的像坐标为 x ，则有 $\lambda x = P X = k[R \ t]X$ 。其中 λ 表示每个像素的深度； $[R \ t]$ ，表示摄像机坐标系与世界坐标系刚体变换。了解这些后下面将详细阐述对极约束的相关知识。如图 3.15 所示，其中 C_1, C_2 表示两个摄像头同时看到三维世界中同一点，该点在 C_1 为原点的坐标系的三维坐标为 X_1 ，在 C_2 为原点的坐标系的三维坐标为 X_2 。在该对极几何中，世界坐标原点选为 C_1 ，坐标轴与该相机坐标系相同，所以对于 C_1 表示的相机成像矩阵 $P = k[I \ 0]$ ，而对于 C_2 表示的相机成像矩阵 $P = k[R \ t]$ ，所以 $X_2 = RX_1 + t$ ，表示 C_1 为原点的坐标系的点经过 R, t 可以转换到 C_2 为原点的坐标系表示的过程。在 C_2 为原点的坐标系中，我们可以得到如图 3.15 中红色三角形三边对应的矢量为 $t, X_2, X_2 - t$ ，其中 $X_2 - t = RX_1$ ，由于三边共面有向量的混合积如式 3.21 所示，当然式中 $[t] \times X_2 = t \times X_2$ 表示垂直于图 3.15 红色平面的向量，由绿色箭头显示。最后我们可以得到简洁的形式 $0 = -X_2^T E X_1$ ，其中 $E = [t] \times R$ ， E 被 Longuet Higgins 称为本质矩阵，他首先发现的这一关系。本质矩阵是 3×3 的矩阵，它将旋转与平移的复杂关系转换为简单的形式，下文将介绍如何计算 E 与如何从 E 中恢复旋转、平移信息。

这里还有几个概念需要介绍一下，所有由 3D 点，相机中心 C_1, C_2 组成的平面交成

像平面于两条直线，如图 3.15 中蓝色直线所示， C_1 成像平面中一点 λX_1 对应的极线为红色平面与 C_2 成像平面的交线，即蓝色线 Ex_1 。如果改变 3D 点的位置形成另外一个红色平面，其与成像平面的交线为另外一条极线，所有形成的极线交于一点，该点称为极点，如图中 e_1, e_2 所示。从极线约束的关系看 $0 = -X_2^T Ex_1$ ，结合齐次坐标的性质点 x 在直线上的充要条件是 $x^T L = 0$ ，所以 Ex_1 是过 X_2 的直线，也就是与 C_2 成像平面的极线平行的直线，由于所有的极线 Ex_1 一直通过同一个点（极点 e_2 ），所以 $e_2^T E = 0, Ee_1 = 0$ 。极点与极线的关系在图 3.16 中展示出来，图 a 中粉色点表示选取的两个特征点，而图 b 中对应的图 a 两点的两条极线标注为红色，这两条线的交点为极点。

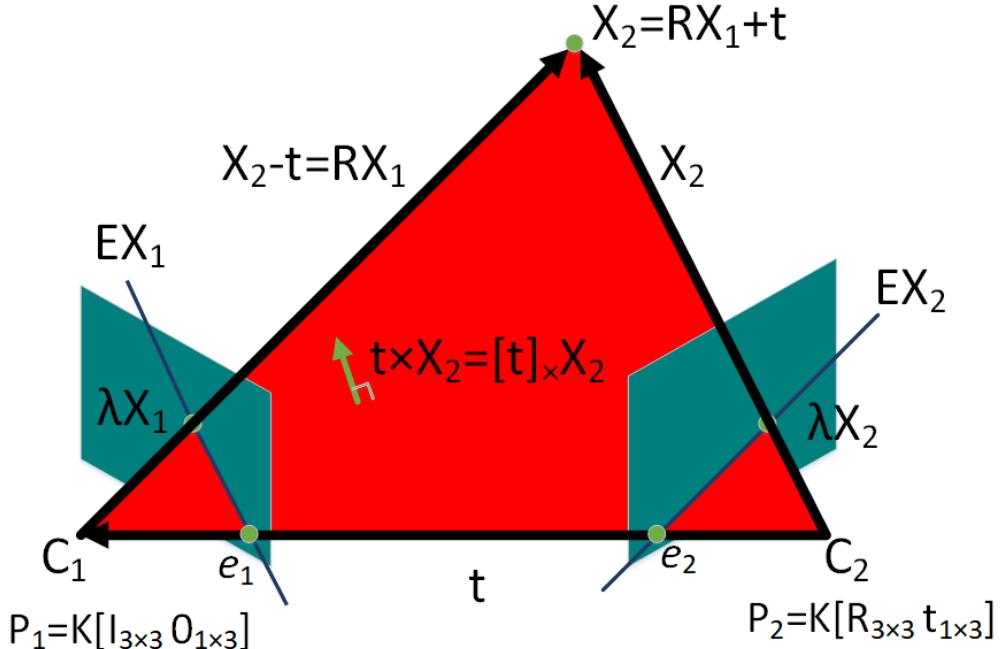


图 3.15 对极几何约束

$$0 = (X_2 - t)^T \cdot [t]_{\times} X_2 = (RX_1)^T \cdot [t]_{\times} X_2 = X_1^T R^T [t]_{\times} X_2 = -X_2^T [t]_{\times} RX_1 = -X_2^T EX_1 \quad (3.21)$$

$$0 = -X_2^T EX_1$$

已知 $KX_1 = x_1$ ，其中 K 为内参矩阵， X_1 表示 C_1 为原点的坐标系中三维点坐标， x_1 表示 C_1 的像素坐标系中二维点坐标。这里和上文相同都是其次坐标，所以是在齐次意义上相等。带入 $-X_2^T EX_1 = 0$ ，中有式 3.22 所示，其中 F 称为基本矩阵， $rank(F) = 2$ ，



(a) a



(b) b

图 3.16 极点与极线的关系

F 的自由度为8。根据式 3.22，一对匹配点可以得到一个方程，所以至少需要8对匹配点才能计算得到矩阵 F 。如式 3.24所示，可以简写为 $AX = 0$ ，与上文计算单应性矩阵的方法相同，由于匹配点远不止8对，所以此时方程存在最小二乘解，使用SVD对左边矩阵分解得 UDV^T ，则基本矩阵即为 V 的最后一列重新排列成 3×3 的形式，当然最后还需要保证 $\text{rank}(F) = 2$ ，这可以通过将 F 进行SVD分解并使最小奇异值为零得到秩为2的 F 。至此我们可以根据至少8对匹配点解算出基本矩阵 F ，该方法称为八点法，在两视图重建中被广泛使用。另外一种方法是使用RANSAC代替SVD分解，避免噪声带来的影响。

$$\begin{aligned} X_2^T E X_1 &= 0 \\ x_2^T k^{-T} E k^{-1} x)) 1 &= 0 \end{aligned} \tag{3.22}$$

$$x_2^T F x_1 = 0$$

基于式 3.22 可以得到下式

$$\begin{bmatrix} u_i^1 & v_i^1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i^1 \\ v_i^1 \\ 1 \end{bmatrix} = 0 \quad (3.23)$$

$$\begin{bmatrix} u_1^1 u_1^2 & u_1^1 v_1^2 & u_1^1 & v_1^1 u_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ u_2^1 u_2^2 & u_2^1 v_2^2 & u_2^1 & v_2^1 u_2^2 & v_2^1 & u_2^2 & v_2^2 & 1 \\ \vdots & \vdots \\ u_8^1 u_8^2 & u_8^1 v_8^2 & u_8^1 & v_8^1 u_8^2 & v_8^1 & u_8^2 & v_8^2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \quad (3.24)$$

在得到基本矩阵 F 后，根据式 3.22 以及已知相机内参矩阵 K 的情况下，可以得到本质矩阵 E ，从本质矩阵可以恢复相机之间的旋转平移关系。

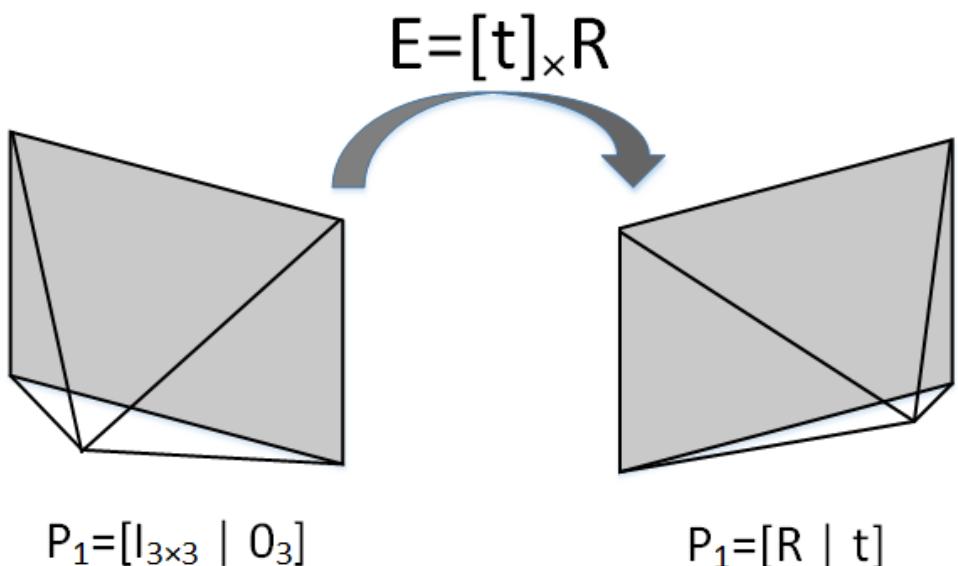


图 3.17 本质矩阵恢复旋转平移关系

由于 $P_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = t$, 所以 $t^T E = 0$, 由SVD知 t 是 E 的左零空间, 也是第二视图极点 $e_2^T E = 0$, 所以 $t = U[:, -1]$, 即 E svd 分解后 U 的最后一列, 即 $t = -u_3, u_3$

$$E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T = [t] \times |R| = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T |U Y V^T \quad (3.25)$$

where $U = \begin{bmatrix} u_1 & u_2 & t \end{bmatrix}$

据此我们可以得到

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} Y \quad (3.26)$$

所以可以得到矩阵 Y 的值, $Y/Y^T = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ 所以 R 有两种可能值, 这是因为 $R = U Y V^T$ 。

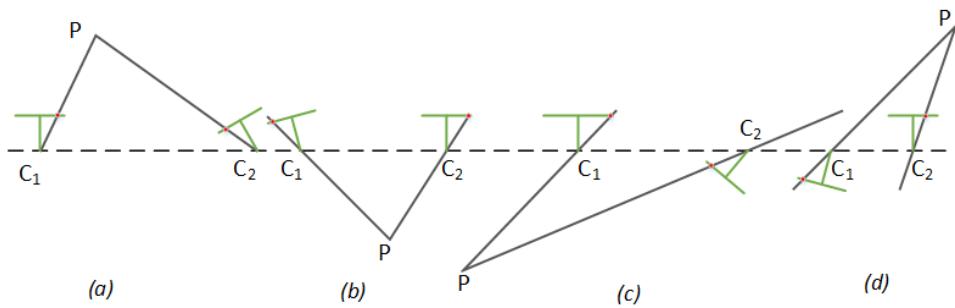


图 3.18 分解本质矩阵得到的四个解

综上通过 E 恢复旋转矩阵 R 和 平移矩阵 t 有四种可能的解, 图 3.18 形象地显示了分解本质矩阵得到四个解的过程。保持成像平面的点 (红点) 不变的情况下, 可以画出四种可能的情况, 但是只有(a)是正确的, 因为其有正深度, 所以恢复旋转与平移后需要将 3D 点带入图中四个解中, 检测该点在两个相机中的深度, 即可确定 R 和 t 。实

际上利用 E 的内在性质，其只有五个自由度：旋转(3)，平移(1)，相机中心(1)。但是使用五点法形式复杂，所以本文还是采用上文介绍的八点法计算旋转与平移关系。

3.6.2 三角定位法

在得到相机的位姿后，需要恢复匹配点的三维坐标信息，该方法称为三角测量(triangulation)，三角测量时指不同视角观察同一个点的夹角，确定该点的位置。三角测量最早由高斯提出并应用于天文学中。在SfM中，我们使用三角测量估计匹配点的像素深度。

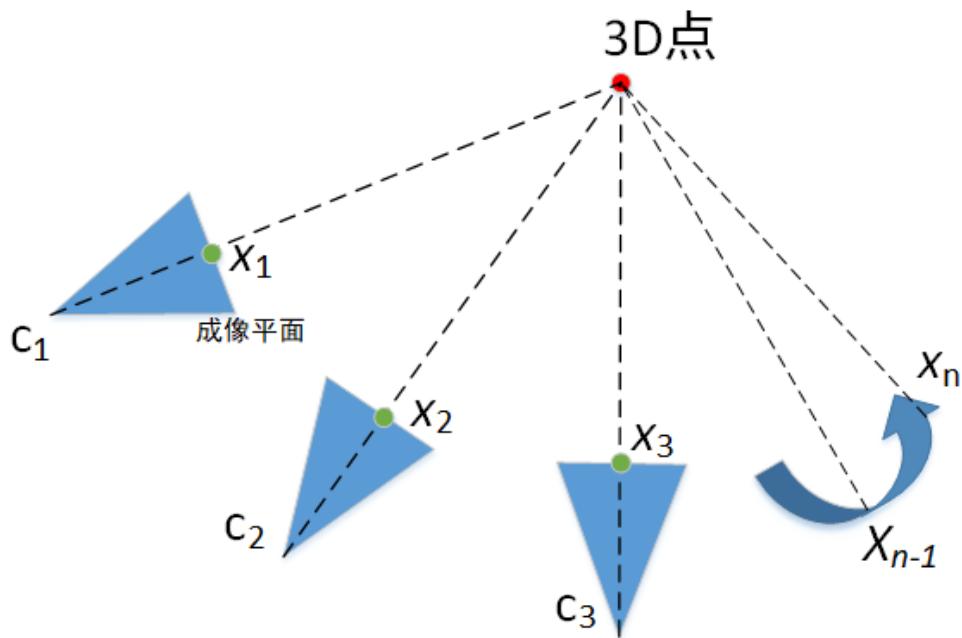


图 3.19 三角测量示意图

小孔相机成像模型简化为 $\lambda \begin{bmatrix} x_1 \\ 1 \end{bmatrix} = P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix}$ ，即 $\begin{bmatrix} x_1 \\ 1 \end{bmatrix} \times P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix} = 0$ ，由于一个视图可以提供两个方程，所以只需要两个视图即可计算得到3D点坐标，但是由于噪声的影响，空间中的两视图射线不一定交于一点，所以使用两视图恢复出来的三维点坐标存在不准确的问题，所以一般情况下使用多视图进行三角测量恢复多视图匹配的3D点坐标，如图 3.19 所示，多个视图匹配的同一个3D点在不同视图成像平面的投影为 x_1, x_2, \dots, x_n ，这些投影点具有相同的track序号（第三章第二节末提到的定义），当从多视图中恢复3D位置坐标，该问题又归为对式 3.27 进行奇异值分解求最小二乘解的问题。由于 $\text{rank}([\])_{3n \times 4} = 2$ 所以对 $[]$ 进行 SVD 分解， V 的最后一列就是 $[]$ 的

右零空间，也就是3D点的位置坐标。

三角测量的多视图必须存在平移关系，否则单纯的旋转无法使用三角测量，因为此时对极约束永远满足。在平移存在的情况下，三角测量存在不确定性，当平移很小时，射线夹角很小，计算得到的像素深度不确定性很大；但是平移太大后，图像变换会较大，导致匹配失效。所以三角测量时，需要考虑多视图夹角，也就是多视图平移大小的影响。

$$\begin{bmatrix} \begin{bmatrix} x_1 \\ 1 \\ x_2 \\ 1 \\ \vdots \end{bmatrix} \times P_1 \\ \begin{bmatrix} x_1 \\ 1 \\ x_2 \\ 1 \\ \vdots \end{bmatrix} \times P_2 \\ \vdots \end{bmatrix}_{3n \times 4} = \begin{bmatrix} X \\ 1 \end{bmatrix}_{4 \times 1} = 0 \quad (3.27)$$

3.6.3 2D-3D位姿求解

经过以上的计算，我们可以得到初始的两视图匹配点的3D位置坐标，也就是我们通过两视图匹配关系得到场景的特征点3D位置坐标，或者称为稀疏点云，以及两视图的空间几何关系，至此我们加入第三张匹配的图片，这时就存在如何计算第三张图片对应的相机与刚刚三角测量得到的点云的空间位置关系，完成这一步才能继续增加其他图片，完成对数据集的增量式重建过程。

PnP(Perspective-n-Point)是求解3D到2D点对运动的方法。新图片能被通过解决PnP问题与当前的点云模型（两视图或者多视图得到的）配准。PnP问题被用来估计相机位姿，包括未矫正相机的内参矩阵。我们知道对极几何估计2D-2D位置关系，使用八点法，但也存在纯旋转等问题。然而，如果两张图像其中一张特征点的3D位置已知，那么最小只需要三对匹配点就可以估计相机运动。特征点的3D位置可以有上文的三角测量或者深度传感器确定。3D-2D方法不需要使用对极约束，其有多种解法，直接线性方法，RANSAC或者非线性优化构建最小二乘问题并接待求解的Bundle Adjustment。

$$\begin{aligned}
& \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} X = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1 X \\ P_2 X \\ P_3 X \end{bmatrix} \\
&= \begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} X^T & 0_{1 \times 4} & 0_{1 \times 4} \\ 0_{1 \times 4} & X^T & 0_{1 \times 4} \\ 0_{1 \times 4} & 0_{1 \times 4} & X^T \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} \\
&= \begin{bmatrix} 0_{1 \times 4} & -X^T & vX^T \\ X^T & 0_{1 \times 4} & -uX^T \\ -vX^T & uX^T & 0_{1 \times 4} \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} = 0
\end{aligned} \tag{3.28}$$

Which is $A_1 b = 0$

下面仅介绍使用直接线性方法结合SVD分解，其他的方法与此原理相似不一一展开，首先回忆最经典的小孔成像模型 $\lambda \begin{bmatrix} x_1 \\ 1 \end{bmatrix} = P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix}$ ，即 $\begin{bmatrix} x_1 \\ 1 \end{bmatrix} \times P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix} = 0$ ，其中 P 为未知量，2D坐标 x 与3D坐标 X 都是已知的，参照求单应性矩阵的方法，我们可以把推导写成如式 3.28 所示，其中 $\text{rank}(A) = 2$ ，这很容易理解，因为每对匹配点仅能提供两个方程， A 为 3×12 矩阵， b 为 12×1 的矩阵，所以需要至少6对匹配点才能计算得到位置关系矩阵 b 。此时等式简写为 3.29 的组合矩阵，其中每对3D-2D匹配点可以写成类似 A_1 的形式。此时可以对式 3.29 中最左边的组合矩阵运用 SVD 求解。对于实际中匹配点大于6对时，SVD求得的是最小二乘解。

$$\begin{bmatrix} A_1 \\ A_2 \\ \dots \\ A_6 \end{bmatrix} b = 0 \tag{3.29}$$

通过以上方法，我们可以得到矩阵 P ，由于 $P = K[R|t]$ ，所以旋转矩阵 $R = K^{-1}P_{[1:3]}$ ， $P_{[1:3]}$ 表示 P 的前三列，由于旋转阵 R 是正交矩阵，为了保证这一性质，

所以旋转阵取式 3.30 中 R_+

$$R_+ = UV^T \quad (3.30)$$

where $UDV^T = R$

那么该如何恢复平移阵 t 呢？如式 3.31 所示，最后 $P = K[R_+t]$

$$t = K^{-1}P_4/\sigma_1 \quad (3.31)$$

where $D = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$

and $\sigma_1 > \sigma_2 > \sigma_3$

3.7 三维重建中的优化

上面介绍了如何使用多视图几何的知识计算相机位姿与特征点的3D坐标，一般是先计算相机位姿，后计算特征点3D坐标，而本章要介绍的非线性优化的问题，将相机位姿与特征点3D位置放在一起优化，实际上SfM中structure就是指特征点3D坐标，而motion指相机位姿，但是以上内容将图像配准（image registration）和三角测量作为独立的步骤，实际上它们是紧密关联的：相机姿态的不确定将导致三角测量的不确定性，反之亦然。没有优化操作，SfM将很快离散到难以恢复的状态。本文一下介绍同时优化相机位姿与3D点坐标的常用方法，称为光束平差法(Bundle Adjustment, BA)，主要手段是最小化重投影误差。

3.7.1 光束平差法

BA是对相机位姿矩阵 P_c 与3D点 X_k 的非线性优化方法，其最小化重投影误差如式 3.32，其中函数 $f(x)$ 表示3D点的重投影误差，函数 π 是3D点在成像平面的投影的像坐标， ρ_j 是损失函数，用来降低外点带来的影响。

$$E = \arg \min_x \sum_{j=1}^k \|f_j(x) - x_j\| = \sum_j \rho_j(\|\pi(P_c, X_k) - x_j\|_2^2) \quad (3.32)$$

为解决该问题，使用最多的方法是梯度下降法，该算法是解决非线性最小二乘问题的通用方法，而BA实际上就是非线性最小二乘问题。实际操作中，我们一般使用Levenberg-Marquardt^[42]，简称LM，LM将非线性问题转变为一系列正则线性问题。

设 $J(x)$ 是 $f(x)$ 的雅各比矩阵，最小化重投影误差 E 如式 3.33

$$\min_x \|f(x) - b\|^2 = \min_x (f(x) - b)^T (f(x) - b) = \min_x f(x)^T f(x) - 2b^T f(x) \quad (3.33)$$

对式 3.33求导数有

$$\frac{\partial E}{\partial x}|_{x^*} = 2 \frac{\partial f(x)}{\partial x}^T f(x) - 2 \frac{\partial f(x)}{\partial x} b = 0 \quad (3.34)$$

其中

$$J = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \vdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \vdots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_m}{\partial x_1} & \vdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (3.35)$$

雅各比矩阵的 m 一般不等于 n ， m 表示组成 $f(x)$ 的等式个数， n 表示与 $f(x)$ 相关的变量的个数，下文会详细解释在光束平差法中各个变量代表的含义。雅各比矩阵反映存在 m 个约束条件的误差函数的变化率。对式中的 $f(x)$ 泰勒展开如式 3.36，最终结果可简写为 $H\Delta x = -J^T \nabla$ ，其中 H 称为Hessian矩阵。

$$\begin{aligned} \frac{\partial E}{\partial x}|_{x^*} &= 2 \frac{\partial f(x)}{\partial x}^T (f(x) + \frac{\partial f(x)}{\partial x} \Delta x) - 2 \frac{\partial f(x)}{\partial x} b = 0 \\ \frac{\partial f(x)}{\partial x}^T \frac{\partial f(x)}{\partial x} \Delta x &= \frac{\partial f(x)}{\partial x}^T (b - f(x)) \\ \Delta x &= (J^T J)^{-1} J^T (b - f(x)) \end{aligned} \quad (3.36)$$

式 3.36的推导结果称为高斯牛顿法，LM是基于高斯牛顿法，只是在迭代步长时不使用 $\Delta x = (J^T J)^{-1} J^T (b - f(x))$ ，而改为 $\Delta x = (J^T J + \lambda D(x)^T D(x))^{-1} J^T (b - f(x))$ ，其中 $D(x)$ 是非负对角矩阵，是 $J^T J$ 的对角元素的平方根，也可以简写为 $H_\lambda \Delta x = -J^T \nabla$ ，式中 H_λ 被称为增广Hessian矩阵。相较高斯牛顿法，LM的优点是可以动态调节 Δx ，当下降太快时，使用较小的 λ ，反之亦然。经过上面的计算我们知道实际上光束平差法是一种非线性最小二乘法，关键是如何计算 $f(x)$ 的雅各比矩阵 J ，即 $f(x)$ 的梯度，并最终计算步长 Δx ，直至误差 E 收敛至最小。下面介绍一下，BA 中针对多个相机多个3D 点使用的LM 算法计算雅各比矩阵的形式。最小化误差函数存在 $6(F - 1)$ 个运动约束和 $3N - 1$ 个结构约束， F 指相机个数， N 指3D点的个数，这可以理解为第一个相机为基准，所以不计入考虑，而每个相机有6 个约束条件，即旋转 (3)、相机中

心(3), 每个3D点都是3个约束条件, 但是缺失一个尺度, 所以减去1。Hessian矩阵 $J^T J$ 的维数为 $(6F + 3N - 7) \times (6F + 3N - 7)$ 。既然如此, 由于3D点与相机非常多, 所以BA得到的雅各比矩阵与Hessian矩阵将非常大。如图3.21所示, 两视图、同一个3D点对应的矩阵为式3.20中的 J

$$J_{\text{part}} = \begin{bmatrix} \frac{\partial f(R(q), C, X)}{\partial R} & \frac{\partial R}{\partial q} & \frac{\partial f(R(q), C, X)}{\partial C} & \frac{\partial f(R(q), C, X)}{\partial X} \end{bmatrix}$$

$$J = \begin{bmatrix} \text{Left View} & 0_{2 \times 7} & 3D \text{ Point} & 0_{2 \times 3} \\ 0_{2 \times 7} & \text{Right View} & 3D \text{ Point} & 0_{2 \times 3} \\ \text{Left View} & 0_{2 \times 7} & 0_{2 \times 3} & 3D \text{ Point} \\ 0_{2 \times 7} & \text{Right View} & 0_{2 \times 3} & 3D \text{ Point} \end{bmatrix}$$

图3.20 雅各比矩阵

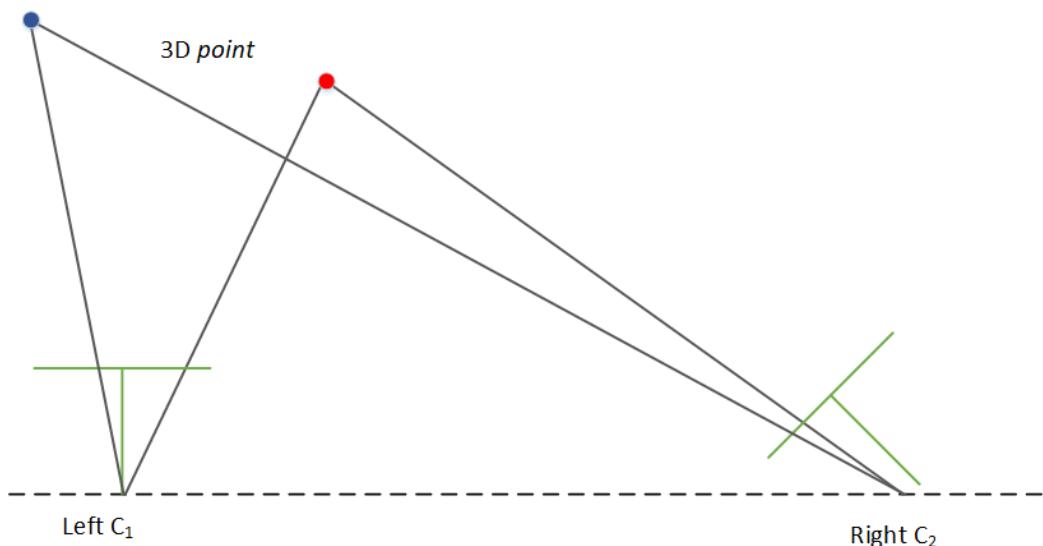


图3.21 雅各比矩阵示意图

$$\begin{bmatrix} U_\lambda & W \\ W^T & V_\lambda \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta x_p \end{bmatrix} = - \begin{bmatrix} J_c^T(b-f) \\ J_p^T(b-f) \end{bmatrix} \quad (3.37)$$

为了下文分析方便，我们设 $U = J_c^T J_c$, $V = J_p^T J_p$, $U_\lambda = U + \lambda D_c^T D_c$, $V_\lambda = V + \lambda D_p^T D_p$, $W = J_c^T J_p$, 下标 c 表示与相机参数有关的向量, 下标 p 表示与 3D 坐标点参数有关的向量, 可以将式 $\Delta x = (J^T J + \lambda D(x)^T D(x))^{-1} J^T (b - f(x))$ 写为分块矩阵形式, 如式 3.37 所示。 U_λ 和 V_λ 是分块对角矩阵, 对此可以采用 Schur Complement 高效求解该方程。考虑求解线性系统 $M \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, 式 3.38 表示分块矩阵 M 的分解

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ CA^{-1} & 1 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & \bar{D} \end{bmatrix} \begin{bmatrix} 1 & A^{-1}B \\ 0 & 1 \end{bmatrix} \quad (3.38)$$

$$\text{where } \bar{D} = D - CA^{-1}B$$

其中, A 必须是非奇异方阵, D 称为 A 在 M 中的 Schur Complement。所以使用矩阵 $\begin{bmatrix} 1 & 0 \\ -CA^{-1} & 1 \end{bmatrix}$ 左乘该线性系统, 有 $\begin{bmatrix} A & B \\ 0 & \bar{D} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ \bar{b}_2 \end{bmatrix}$, 其中 $\bar{b}_2 = b_2 - CA^{-1}b_1$, 至此我们可以得到一个降阶系统 $\bar{D}x_2 = \bar{b}_2$, 求解 x_2 后, 回带解算 x_1 。

回到光束平差法问题中, 可以得到降阶系统 3.39 和 3.40, 矩阵 $S = (V_\lambda - WU_\lambda^{-1}W^T)\Delta x_p$ 是 Schur Complement。

$$(U_\lambda - WV_\lambda^{-1}W^T)\Delta x_c = -J_c(b-f) + W^T V_\lambda^{-1} J_p^T (b-f) \quad (3.39)$$

$$\Delta x_p = -V_\lambda^{-1}(J_p^T(b-f) + W^T \Delta x_c) \quad (3.40)$$

S 是对称正定矩阵, 使用 Cholesky 分解可以求解式 3.39。以上求解 BA 问题的方法之所以有效, 是由于相机数量要远小于 3D 点数量, 所以可以先求解 Δx_c , 再求解 Δx_p 。

3.7.2 最小化相机中心位置误差

上节介绍了如何使用 Schur Complement 简化线性系统, 求解使重投影误差最小的 $\Delta x_c, \Delta x_p$, 其中涉及到的变量有旋转、相机中心(平移)、3D 点坐标, 而本节将介

绍一种优化方法：最小化相机中心位置误差。由于从图片中，我们可以提取相机的EXIF(Exchangeable image file format)信息，从而得到相机的经纬度坐标。EXIF是由数码相机制造商在图片、声音上标记的相机、图片、声音等相关内容的标准格式文件。EXIF包含的信息丰富，可以从中找到经纬高信息，也可以从中提取相机模型，包括焦距等，也包括相机制造厂商，甚至相机位姿等。

对于同一个3D点在两视图中的左视图的相机坐标系中坐标为 X_1 ，在右视图的相机坐标系中坐标为 X_2 ，已知两视图相机位姿关系 R, t ，则坐标可以表示为 $RX_1 + t = X_2$ 。对于右视图相机中心这一3D点有 $X_2 = 0$ ，则该点在左视图的相机坐标系中的坐标为 $X_1 = -R^T t$ ，旋转矩阵是正交矩阵有 $R^T = R^{-1}$ 。所以在多视图中每个视图与第一个视图的位姿关系已知为 R_i, t_i ，则这些视图对应的相机中心坐标在第一视图的相机坐标系中的坐标为 $-R_i^T t_i$ 。这样我们就建立了每个视图的经纬高 $pos = (Lon, Lat, High)$ 与相对于第一视图的位姿关系 $-R_i^T t_i$ ，基于此我们可以建立最小二乘问题如式 3.41。

$$E_{position} = \arg \min_{R, t} \sum_{i=1} \|f_j(x) - x_j\| = \sum_i (\|-R_i^T t_i - pos\|_2^2) \quad (3.41)$$

在此不详细介绍如何解算上式，相对于最小化重投影误差的过程，该过程只调整相机相关参数，没有涉及3D点相关的调整，所以只能作为BA的辅助步骤，也可以认为这属于光束平差法的一部分，下文不再过多强调。



图 3.22 BA对比结果图

本节介绍了基本的稀疏点云生成的过程，包括多视图几何知识从匹配点恢复结构，而后增量式的重建其他图片；在每加入一张图片时，使用BA优化相机位姿与3D点的位置，在此我加上最小化相机中心位置误差，作为BA的辅助步骤。图 3.22(b)是使用CeresSolver库迭代三次优化重投影误差的结果，初始代价函数结果为 $3.483778e+01$ ，BA优化后的代价为 $1.732320e+01$ 。从 3.22两张图对比可以看

出BA最小化重投影误差的作用:(a) 仅使用多视图几何计算相机位姿并使用三角测量恢复3D点位置, 而(b)在(a)的基础上增加BA环节, 图中红色点是提取的2D特征点坐标, 绿色点是3D根据相机 R, t 关系投影回成像平面的点, 从中可以明显看出BA对重投影误差有改善作用, 可以将误差平摊到所有点上。

3.8 稀疏点云的稠密化

通过SfM方法, 不仅可以得到相机位姿, 也可以恢复特征点的三维坐标得到稀疏点云, 但是由于点云过于稀疏难以用于无人车导航与规划, 所以本文使用Multiple View Stereo(MVS)技术将点云稠密化, MVS可以分为基于体素 (voxel), 基于表面演化 (surface evolution)、基于特征点生长 (feature point growing) 和基于深度图融合 (depth-map merging) 四种类型^[45]。本文采用深度图融合的方法^[46]得到稠密点云, 该方法包括深度图计算与深度图融合两部分, 可以用图 3.23 表示整个过程。

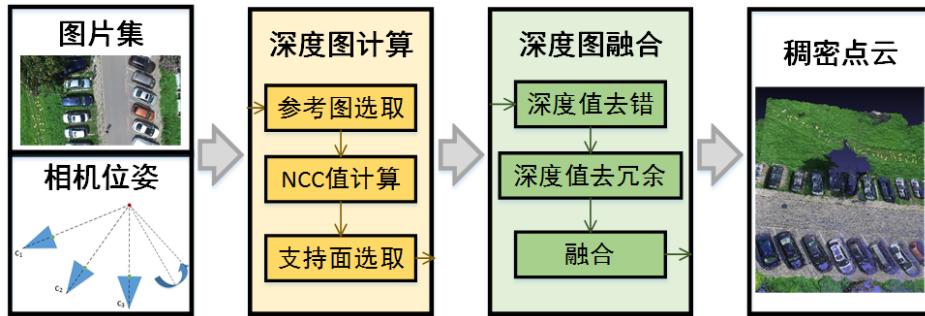


图 3.23 点云稠密化流程图

3.8.1 深度图计算

基于SfM方法, 我们可以得到图像数据集下所有图片对应相机的位姿关系与特征点的3D坐标, 利用这些位姿关系, 可以完成接下来的深度图计算过程。对于单张图片无法计算深度图, 深度图的计算首先需要为每张图片选取其参考图 (Reference Images), 而后根据参考图与自身的旋转平移关系 (通过SfM得到), 计算每张图片的深度图。对于双目相机, 通过双目图片得到深度图, 双目成像互为参考图; 然而对于无序的图片集, 参考图的选取至关重要。本文参考Shen^[2]的方法设置参考图阈值, 假设有 n 张图片, $\theta_{ij}, j = 1, \dots, n$ 表示第 i 张图片与其他图片 $1, \dots, n$ 的相机主轴夹角, θ_{ij} 用 i 与 j 图片匹配的特征点之间夹角的平均值表示。同时每张图片与其参考图对应相机的距离称为基线, 记为 d_{ij} , 基线的距离太长则两张图片重叠区域太小, 太短则使重

建精度降低，据此满足 $5^\circ < \theta_{ij} < 60^\circ$, $d_{ij} < 0.05\bar{d}$ 或 $d_{ij} > 2\bar{d}$ 的图片作为参考图，按照 $\theta_{ij}d_{ij}$ 降序排序参考图，取前10张（如果有）作为参考图。

根据Bleyer^[47]的方法使用上文刷选的参考图计算深度图，主要思想是遍历图片的每个像素，为其找到一个支持面，使该支持面在参考图中有最小汇总匹配代价，整个过程如图3.24所示。其中支持面 f 由法线 n_i 与3D点 X_i 决定，两视图几何中，以一个相机的相机坐标系作为世界坐标系，则有 $P = [I_{3 \times 3}|0_3]$, $P' = [R|t]$, 空间平面的齐次表达式为 $\pi^T X = 0$, 其中 $\pi = (V^T, 1)^T$, $V^T = (V_1, V_2, V_3)$, 由于 X_i 在面 π 上，所以有 $|V^T| = \left| \frac{1}{X_i} \right|$, 而方向与 n_i 相反，故 $V^T = -\frac{n_i^T}{n_i^T X_i}$ 。

相机 i 由 K_i, R_i, C_i 表示，参考图对应的相机用 K_j, R_j, C_j 表示，现在设世界坐标系中心为相机 i ，则相机坐标系间的旋转平移关系变为 $P_i = K_i[I_{3 \times 3}|0_3]$, $P_j = K_j[R_j R_i^{-1}|R_j(C_i - C_j)]$ 根据吴福朝^[48] 有两视图间的单应性可以用式3.42表示。

$$H = K_j(R - tV^T)K_i^{-1} = K_j(R_j R_i^{-1} + R_j \frac{(C_i - C_j)n_i^T}{n_i^T X_i})K_i^{-1} \quad (3.42)$$

图3.24中支持面 f (patch) 以 p 为中心，大小为 7×7 像素的正方形块，对于patch中的每个像素通过式3.42转换到参考图中，记为 H_{ij} ，此时可以求出 7×7 所有像素 q 的代价和 $m(p, f_p)$ ，称为去均值的归一化互相关 (NCC, Normalized Cross Correlation)，如式3.43所示，其中上划线表示所有像素点的平均值。

$$m(p, f_p) = 1 - \frac{\sum_{q \in patch} (q - \bar{q})(H_{ij}(q) - \overline{H_{ij}(q)})}{\sqrt{\sum_{q \in patch} (q - \bar{q})^2 \sum_{q \in patch} (H_{ij}(q) - \overline{H_{ij}(q)})^2}} \quad (3.43)$$

据此可计算数据集中每张图片的深度图与图片中每个像素对应的NCC值，如图3.25所示，图(b),(e)为深度图，图(c),(f)为NCC值图，图中使用冷暖色表示像素值的大小，颜色越暖表示像素值越大。图3.25(c)较冷（蓝色）的区域表示用来计算深度图的图片与参考图对应patch区域像素误差较大，当大于一定阈值时，证明该区域的深度计算是不准确的，故对应图3.25(e)的深度设为零，为最浅色（蓝色）部分。对于深度图而言颜色越暖证明深度越大，图3.25(e)很好反映了一面竖直墙导致图片深度的变化情况。

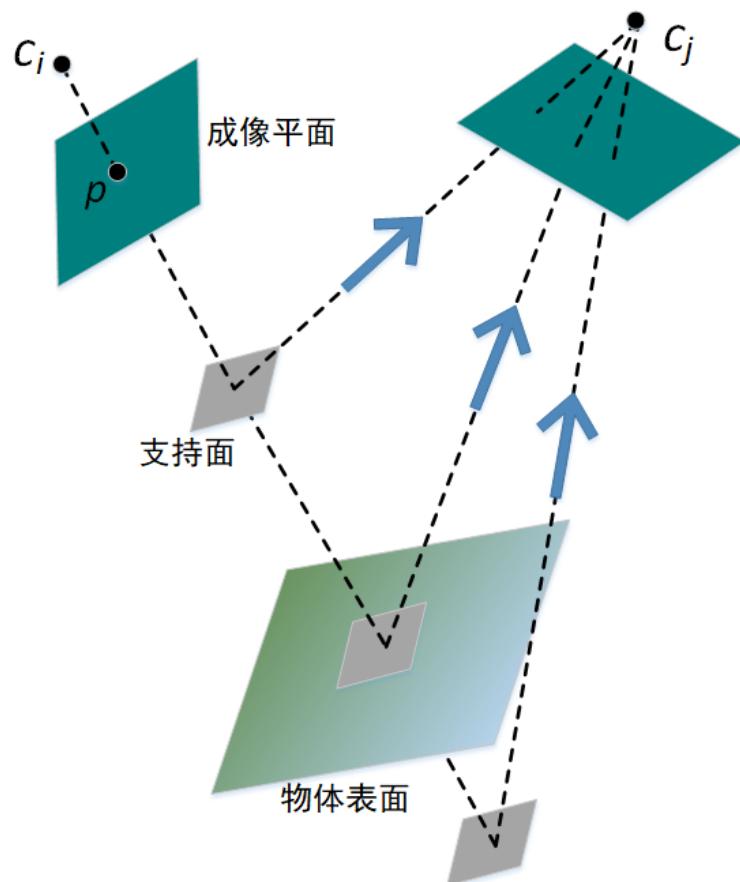


图 3.24 最小汇总匹配代价计算过程

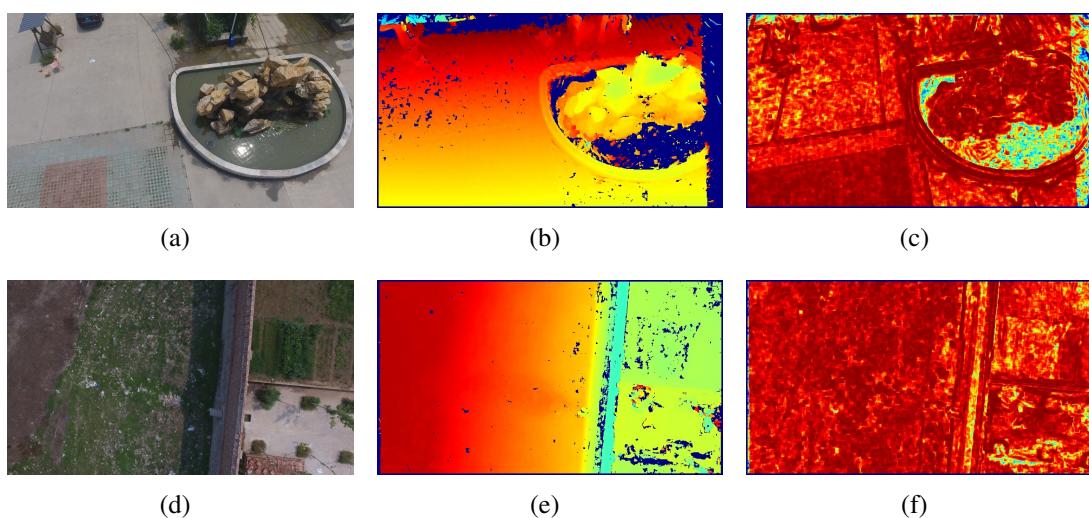


图 3.25 深度图与NCC值

3.8.2 深度图融合

在得到每张图片的深度图之后，需要将各个深度图融合得到完成的场景图，由于图片存在重叠区域，所以直接融合深度图会在重叠区域产生深度冗余，所以深度图的融合就是去除冗余深度的过程。将图*i*每个像素点*x*通过上文计算得到的深度重新投影到3D空间为 $X = \lambda R_i^T K_i^{-1}x + C_i$ ，而*i*的参考图 $j = 1,..h$ 也可以得到对应图*i*中每个像素的深度，这个深度可以通过参考图的深度图计算得到，记为 $\lambda(X, j)$ ，对于依赖图*i*的深度图得到的坐标X可以转换到参考图的相机坐标系中得到X对应的像素深度 $d(X, i)$ ，如果 $d(X, i) < \lambda(X, j)$ 或 $\frac{|d(X, i) - \lambda(X, j)|}{\lambda(X, j)} > \epsilon$ ，则移除参考图中该像素的深度。

3.9 本章小结

本章介绍了如何将点云稠密化的过程，首先采用基于块的多视图理论，以去均值的NCC作为块匹配代价函数得到数据集中每张图片的每个像素的深度，由于图片存在重叠区域，所以深度图的融合也存在冗余深度的问题，所以精炼深度图，深度设为零（图3.25(b)蓝色区域），从而去除冗余深度，从而完成深度图的融合，得到稠密点云。

第4章 二维图像中的道路提取

4.1 引言

最近数十年来许多方法被应用于无人地面车辆自主导航，使用地图提高定位、场景理解的方法是非常成功的。大部分无人机依靠详细路网地图导航，然后这些路网地图基本是靠手工完成的，如此限制了当前无人车辆的大范围普及。提取道路的方法被广泛的研究用于车辆导航与区域分割，这些算法对于UGV有很好的效果，然而对于UAV这些算法基本都失效。这是因为UGV的相机朝前安装而UAV相机俯视地面，UGV视野中道路在视野尽头一般收敛为消失点，针对消失点的提取道路的方法在UAV上效果较差，为了解决这一问题本文提出了使用图像二值化并结合道路拓扑关系相关内容对航拍图像进行道路提取。

4.2 图像二值化

本文采用最小二乘法提取道路区域得到二值化的图像。在实际工程中我们总是希望找到数据的线性关系，不幸的是对于实际数据找到这样完美的线性关系不容易，然而最小二乘法提供了一种可信赖的最小化残差方法，这展示了观测与拟合之间的差别。考虑到某区域的道路颜色特征基本一致的特点，本文采用线性最小二乘的方法拟合图像道路区域的RGB颜色特征，得到道路的二值化图。



图 4.1 训练集

首先从UAV图片中找到一部分道路区域作为训练的样本集，如图 4.1中非黑色部分，再次训练集中的图片分为RGB三个通道展示在三维空间坐标系中，如图 4.2中蓝色点所示，而后最小二乘法被用于拟合这些观测值（蓝色点）为一条直线 $\frac{1}{2}x + \frac{1}{2}a = \frac{1}{2}y + \frac{1}{2}b = \frac{1}{2}z$ ，如图 4.2中红色直线所示，最后可以计算所有观测点与拟合直线之间的平均欧式距离 D 。对于需要处理的图片，也就是测试集数据，遍历图片的每个像素，如果测试集像素与拟合直线的欧氏距离大于 $3D$ 则将该像素赋值为零，否则赋值为一，进而得到和原图大小相同的二值图，按照这种方法进而每张UAV图片转化为二值图。

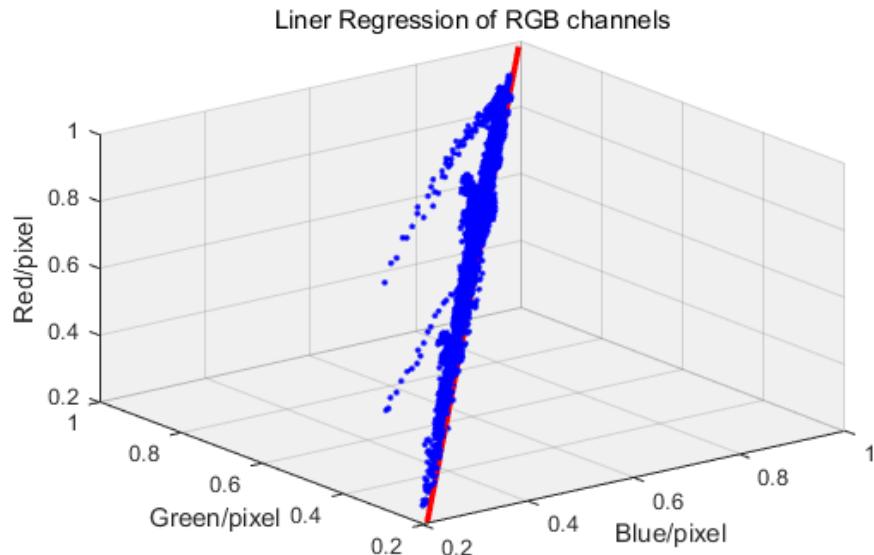


图 4.2 线性回归

4.3 非道路区域移除

通过最小二乘法我们初始的道路区域，然而这有很多的噪声和错误的道路检测，为了减少噪声和误差，本文采用数学形态学精简结果。数学形态学是关于基于拓扑的图像分析学科，基本的操作有开闭运算、俯视膨胀运算和形态学梯度等。道路有显而易见的不同于图片周围物体的形态学特征——瘦长。这些拓扑特征可以被用来消除二值图中的非道路区域，精简二值图。

4.3.1 开闭运算

在最小二乘法得到的二值图中存在许多噪声，例如小孔等，这影响了道路的完整

性，本文中，开闭运算被用来移除孔洞，消除噪声，平滑道路区域。开运算是先腐蚀图片后膨胀图片的运算，而闭运算正好相反，是先膨胀后腐蚀图片，式 4.2 和 ?? 所示。其中二值图像的腐蚀是指选择大小一定的核函数 K ， K 一般是方阵，则二值图像 I 被 K 腐蚀可以理解为 K 在 I 内部移动时， K 的中心经过的区域即为腐蚀后的 I 的值。二值图像的膨胀是指选择大小一定的核函数 K ， K 一般为方阵，则二值图像 I 被 K 膨胀可以理解为 K 的中心在 I 内部移动时， K 所经过的区域。腐蚀膨胀的示意图如 4.3 中左图深蓝色为腐蚀掉的像素，右图浅蓝色为膨胀后的像素。

$$\text{Open}(I, K) = \text{dilate}(\text{erode}(I, k), K) \quad (4.1)$$

$$\text{Close}(I, K) = \text{erode}(\text{dilate}(I, k), K) \quad (4.2)$$

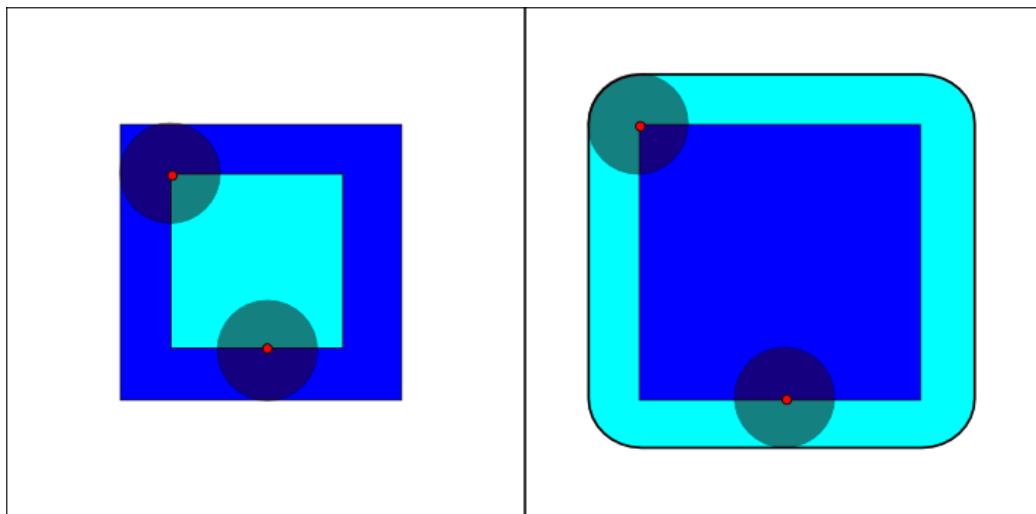


图 4.3 腐蚀膨胀示意图

基于开闭运算的方法，道路区域被平滑，大量干扰（噪声）被移除，但是仍然存在非道路区域，像房子边缘，被错误识别为道路，由于这些区域面积较大，所以基于开闭运算的方法无法移除这些区域，所以采用以下方法去除非道路道路的干扰。

4.3.2 轮廓提取

Ramer Douglas Peucker 算法（简称 RDP）是由 Ramer、David Douglas 和 Thomas Peucker 提出的^[43]，通过物体轮廓估计物体形状。需要移除的房子边缘被检测为

满足式 4.3、4.4 与 4.5 的轮廓，而这些被需要移除。

$$\|C - C'\| \leq 10\%C \quad (4.3)$$

$$60^\circ \leq \theta \leq 100^\circ \quad (4.4)$$

$$\frac{L}{L'} \leq 6 \quad or \quad \frac{L}{L'} \geq \frac{1}{6} \quad (4.5)$$

其中， C 、 C' 代表原始轮廓与近似轮廓的周长， θ 代表 C' 中每两个相邻顶点连线所成的角度。 L 和 L' 代表 C' 的相邻两边。按照该式可以检测到几个两个非道路区域，如 4.4 中红色矩形所示。至此，我们完成了形态学滤波得到较为理想的道路区域，然而

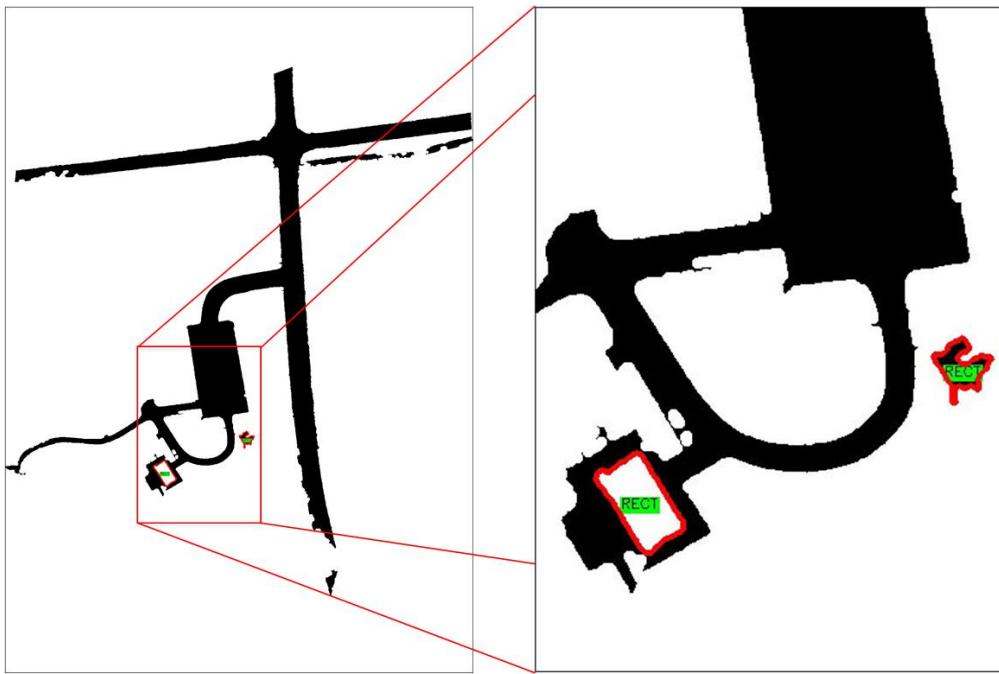


图 4.4 轮廓检测

这样的道路并不适合直接应用于无人车路径规划与导航，因为 UGV 路径规划需要拓扑路网，下面将介绍如何从栅格道路提取道路骨架，得到拓扑路网的过程。

4.4 拓扑处理

4.4.1 拓扑细化

栅格地图可以被简化为无人车路径规划中广泛使用的拓扑地图，这种地图有基本几何元素组成，例如节点和弧线、直线等。Choi^[44] 使用拓扑细化的方法提取道路骨

架，道路的宽度不会影响算法的有效性，图 ??显示了拓扑细化算法的过程，图中左边类似形态学处理后的道路区域，右边类似拓扑处理后的道路骨架，可以看出线段宽度不影响最后得到的道路骨架结果。拓扑细化的目的就是剥离像素，知道无法剥离

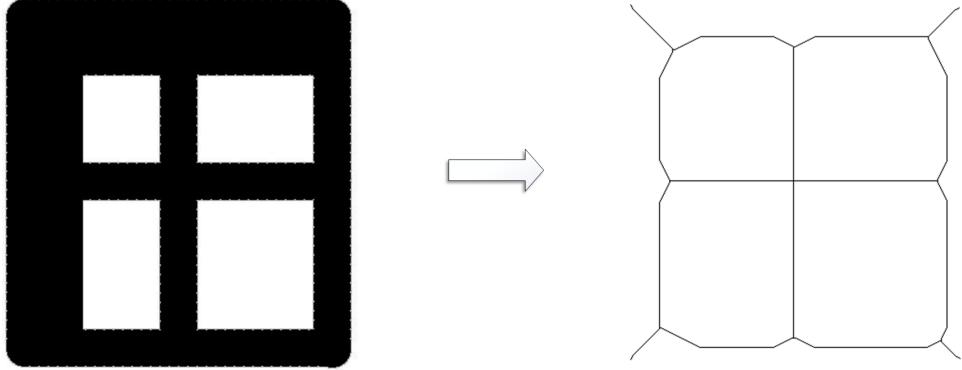


图 4.5 拓扑细化

为止，从而得到单像素构成的拓扑骨架。对于构成道路的像素是否可以剥离取决于像素的连通数，连通数的概念由式 4.6 和图 4.6 定义。

$$C = \sum(N_k - N_k N_{k+1} N_{k+2}), k = 1, 3, 5, 7, N_9 = N_1 \quad (4.6)$$

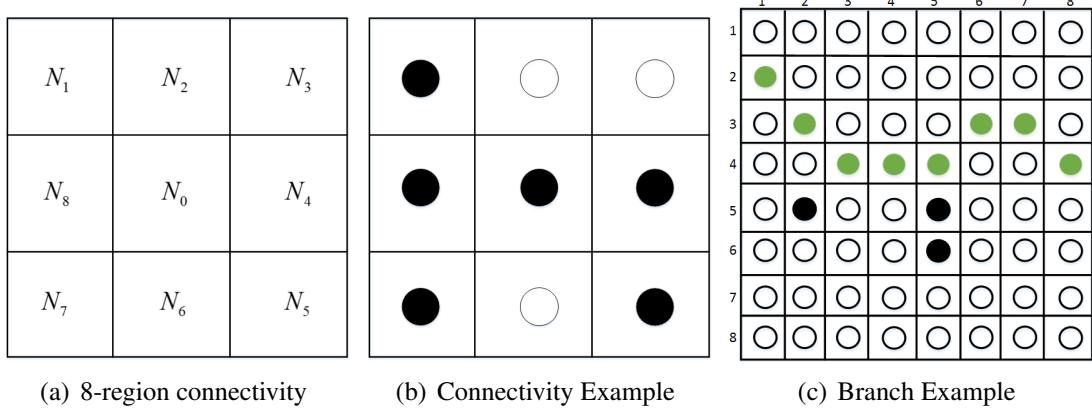


图 4.6 Connectivity

在 4.6(a) 中 N_i 表示以 N_0 为中心的 8 邻域中第 i 个像素大小，例如 $N_1 = 1, N_2 = 0$ ，二值图像中像素值非 0 即 1；根据式 4.6 可以计算某像素的 8 邻域中连通数的大小。如果一个像素的连通数等于 1 并且该像素不是端点（拓扑的起点或者终点），那么我们可以简化删除该像素简化拓扑关系。按照这一原则对于图 4.6(b)，其中黑色圆形对应的像素值为 1，白色为 0， N_0 的连通数 $C = 2$ ，所以 N_0 无法删除；此时如果将图

4.6(b)的 N_6 设为1, N_0 的连通数变为1, 且 N_0 不是端点, 所以 N_0 可以被移除; 同样如果 $N_4 = N_5 = N_7 = N_8 = 0$, N_0 不可移除, 尽管 $C_{N_0} = 1$, 但 N_0 为端点。按照拓扑细化的规则, 遍历图片中所有像素, 对每个像素采用拓扑细化的方法就可以得到基本的道路骨架。

4.4.2 去除小枝丫

拓扑细化的规则虽然能够得到保持道路的连通、完整, 但是同时产生冗余的小枝丫, 如图 4.5所示所示, 本文提出了一种从道路骨架中移除小枝丫的方法。在给出介绍之前, 先定义一下连通度的概念, 不同于应用于拓扑细化的连通数概念, 连通度的定义如下:

- (1) 一个像素的连通度: 在8邻域中, 一个像素直接与之相邻的像素个数, 图 4.6(b)中心像素的连通度是5。
- (2) 道路骨架的端点: 在8邻域中, 一个像素与之相邻的像素仅一个, 即连通度为1。
- (3) 道路骨架的节点: 一个像素与之相邻的像素个数大于2, 例如 4.6(b)中心像素即为拓扑的节点, 换句话说像素连通度大于2。
- (4) 枝丫的长度: 从起点到最近节点的像素个数。

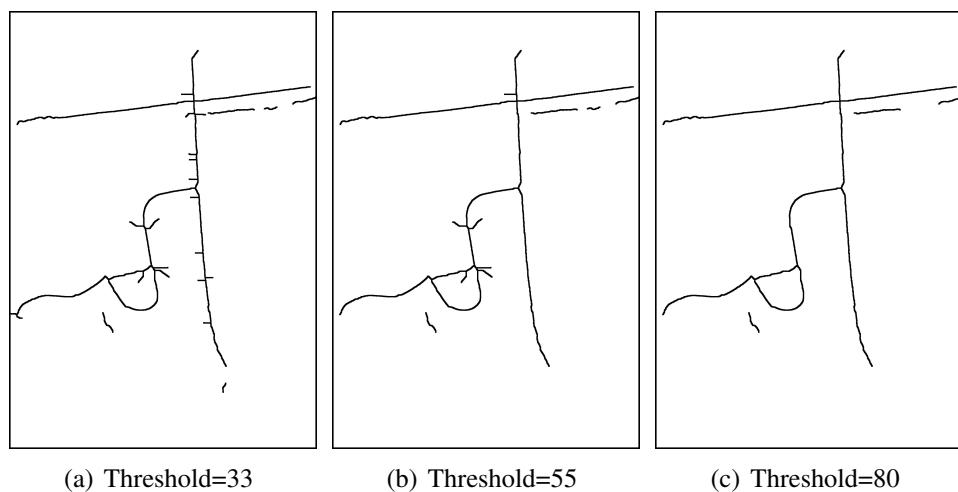


图 4.7 Branches Elimination

基于以上定义, 我们可以推断图 4.6(c)中 $P(1,2), P(2,5), P(5,6), P(8,4)$ 是端点, $P(3,4), P(5,4)$ 是道路节点, 枝丫长度对于端点 $P(1,2), P(2,5), P(5,6), P(8,4)$ 分别为3, 1, 3, 4。

实际道路枝丫通常较长，在 4.5 中短的枝丫很显然并不属于真实道路骨架，设计逐渐增大的阈值，如果枝丫长度小于初始较小阈值，则该枝丫被移除。阈值逐渐增大，小于该阈值的枝丫被删除，直到阈值到达设定最大值，该过程结束。在图 4.6(c) 中，黑色点将被移除，剩下的绿点就是道路骨架。图 4.7 展示了阈值逐渐增大过程中，拓扑路网的简化过程。

4.5 本章小结

本章提出使用最小二乘法拟合道路区域得到二值图而后使用开闭运算等方法平滑道路区域，之后采用 RDP 算法提取道路骨架得到适合无人车辆路径规划的拓扑网络，经过拓扑细化的道路区域虽然保持连通性，但也存在过多的冗余小枝丫，本文设计了通过计数枝丫长度的方法去除小枝丫的方法。本章的方法简单实用，但对于复杂的道路仍然存在少量错误提取的道路网络，而这些需要手动加以修改。

第 5 章 实验结果分析

5.1 引言

本文简述基于全景图像的二维地图构建与基于航拍图像的三维地图构建方法，构建适用于无人驾驶的地图，并对地图中的道路进行提取，构建拓扑路网。本章将从实验结果分析算法的有效性，实验分为三部分，二维、三维地图构建，道路提取，由于实验平台不同，将在各节文章中单独介绍。实验将分析构图或道路提取的精度，并简单介绍实验结果如何应用于无人驾驶中。

5.2 全景图像构建地图实验结果

5.2.1 全景图像地图实验平台搭建

本文提出的使用全景相机构建地图的方法使用的工具包括Ladybug全景相机（图2.2）与高精度组合定位系统，通过标定全景相机与车体后轴中点的相对位置关系，得到全景相机球面坐标与车体坐标的关系，最后将全景图像（图2.4）变换为逆透视图像（图2.5），投影在地面中，依据定位信息将图像拼接为地图。除了传感器外，本节实验使用的计算机为运行Windows 7的台式机，CPU型号为英特尔酷睿I7-5930K，主频为3.5GHz，内存为4通道32GB。所有算法由C++语言实现，使用OpenCV开源库。由于本节构建地图的范围较大，而拼接的图像需要导入GIS软件（Supermap）生成金字塔图以便加快地图的缩放等操作，该过程是逐行逐像素操作的，所以需要较大的计算量与较大的内存。

5.2.2 全景图像地图构建效果及精度分析

图5.1是使用本文提出的方法构建的二维地图，该地图在2016年于江苏常熟举办的无人车“未来挑战赛”中被采用，在无人车辆出发前，通过给定的任务点，依据地图信息进行全局路径规划，同时与GIS软件结合，在无人车行驶过程中，为无人车提供道路信息等。

图5.1(b)为图5.1(a)的细节展示，其中(b)绿色方框内的点为无人车比赛时，车辆出发前路径规划得到的轨迹图，(b)上为左转变道过程，(b)下为U型转弯过程。图5.2(a)为使用本节提出的方法构建的常熟三环路网地图，地图面积大约为 $12000 \times$

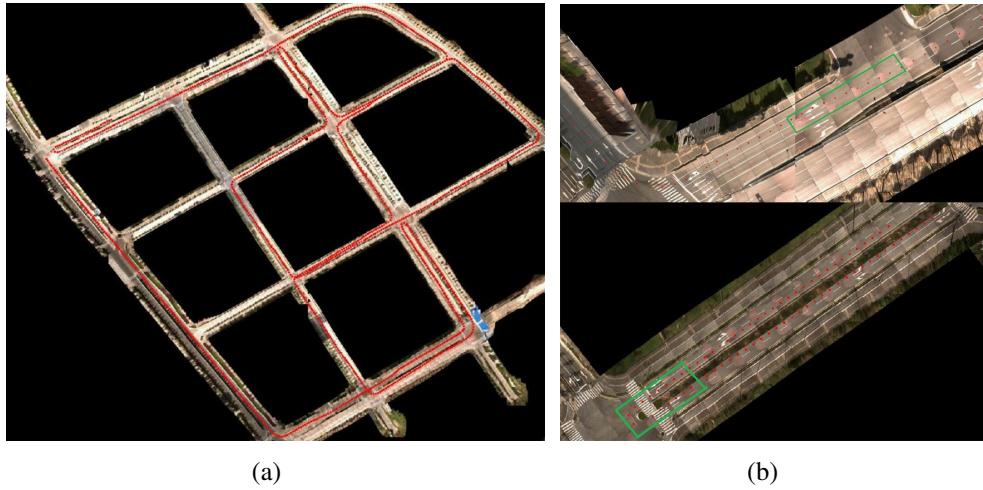


图 5.1 常熟无人车试验场

8000m的区域，三环道路的细节部分展示在图 5.2(b)(c)中，可以看出道路拼接良好，车道线存在轻微没有对齐的现象。

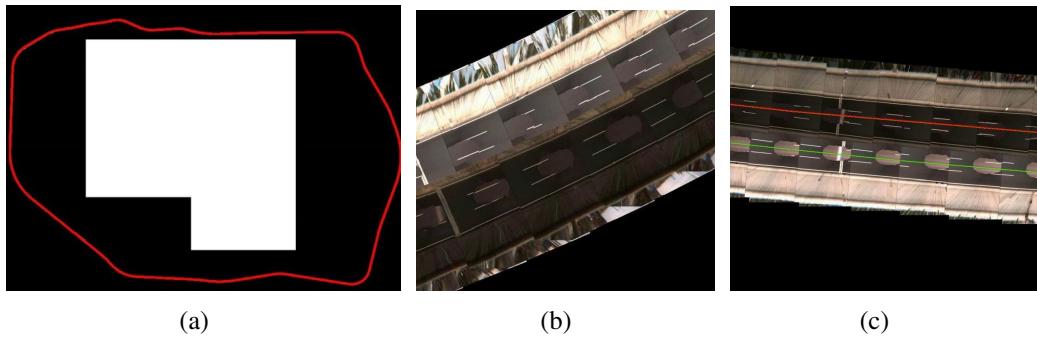


图 5.2 常熟三环路

本节提出的构建适用于无人车行驶的二维地图，地图的精度很大程度取决于高精度定位系统的精度，因为每帧图片是按照位置与航向拼接的；其次需要考虑定位系统与全景相机的硬件同步，由于两个传感器采集频率不同，当采集到一帧图像时，对该帧图像的车体位置（严格讲是全景相机位置）需要严格同步才可能做到高精度地图，最后就是全景相机成像逆透视变换时产生的形变也影响到构图的精度，尤其是本节算法默认除车外的其他区域高度与地面相同，所以路边花坛、树木等变形严重，无法作为后期制作地图的依据，这三方面误差是导致本节提出构建地图误差的主要来源。这些误差受制于传感器算法、地形等因素，所以很难量化，故在此不做详细讨论。本节提出的构建地图的方法分别应用于2016常熟“未来挑战赛”与军方总装的“跨越险阻”比赛中，在“未来挑战赛”中大部分地图区域拥有接近车道级别的构图

精度，但是在“跨越险阻”比赛中，由于山地崎岖、颠簸严重，所以构图精度没有城市环境的“未来挑战赛”出色。

5.3 地理空间三维重建实验结果

5.3.1 地理空间三维重建实验平台搭建



图 5.3 DJI M100

本文是用算法由C++/Python实现，SfM与MVS实现是基于OpenCV、CeresSolver开源库，本研究使用运行Ubuntu 14.04 LTS操作系统的台式机，台式机CPU型号为intel酷睿i5-4570，CPU主频为3.2GHz，并配置8GB单通道内存，没有配置独立显卡。航拍图像采集工具为大疆（DJI）M100无人机，如图 ?? 所示。图像分辨率为 4000×2240 ，直接存储于无人机相机的16GB SD卡中，用于本文实验后处理。

5.3.2 三维重建效果及精度分析



图 5.4 实验耗时测量

图 5.4 表示 SfM 过程两个数据集的耗时情况，其中数据集 1 表示 雾岚山庄（56 张图片），数据集 2 表示 自动化学院（79 张图片）数据集，图片分辨率为 2400×1350 。蓝

色表示提取所有图片的SIFT特征点并存储的耗时，黄色的耗时包括使用FLANN匹配特征点，并使用对极约束去除错误匹配，而后将所有图片的匹配关系用track序号表示的过程，绿色表示接下来的重建过程，包括初始化、增量式重建过程，系统使用Python虚拟的多线程（4线程）实现。

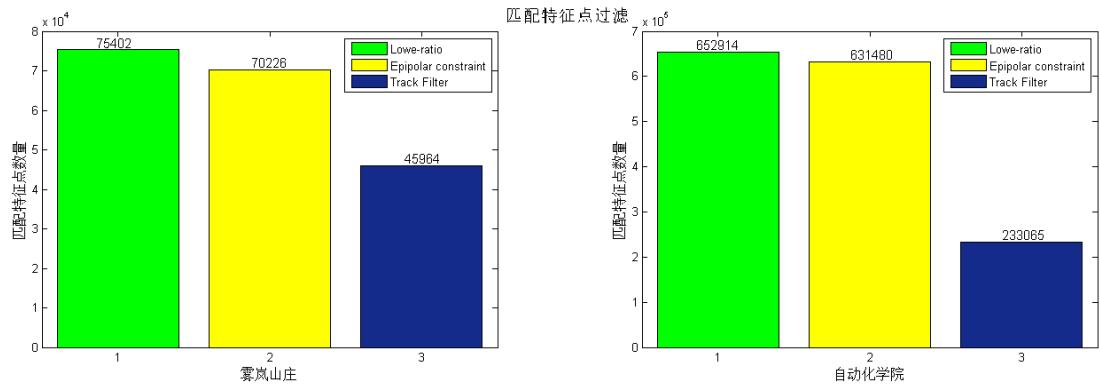


图 5.5 匹配特征点数量变化

重建的基础是正确的匹配数据集中所有图片的特征点，所以过滤错误的匹配特征点至关重要，图 5.5展示了不同方法过滤错误匹配特征点对的情况，使用的数据集与上文描述相同。绿色表示使用Lowe提出的最近邻与次近邻比例小于0.8得到的匹配特征点数目。经过对极约束后剩下的特征点数目为黄色表示，即图片特征点与其对应极线距离不大于9个像素的特征点数目。基于对极约束刷选结果，通过匹配关系为每个特征点设置一个track序号，如果同一个track序号对应特征点所在视图存在名称相同的现象，则说明该匹配关系存在错误，删除该track序号对应的所有匹配特征点，剩余的特征点。

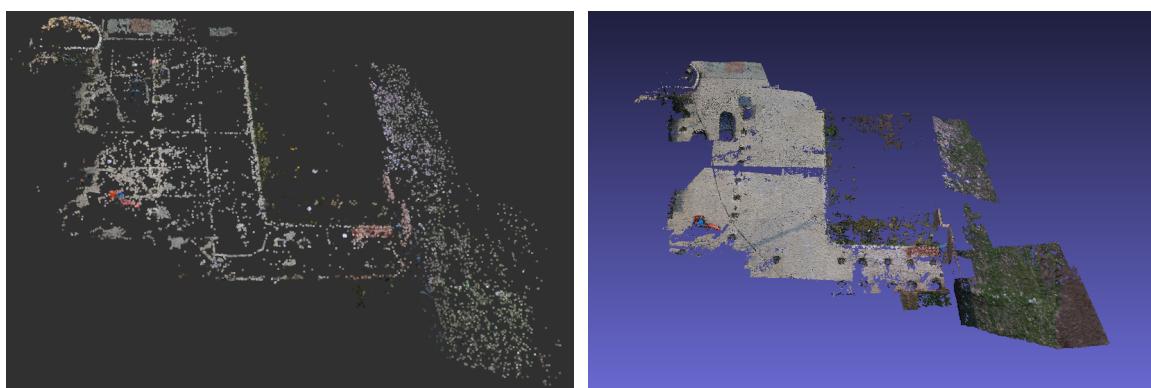


图 5.6 实验结果-雾岚

如图 5.6 所示, 使用 54 张 DJI 航拍图像得到的稀疏点云(图 5.6(a))与稠密点云(图 5.6(b)), 稀疏点云显示在 RVIZ 软件中, 其中每个点都是提取的 SIFT2D 特征点还原的 3D 点, 稠密点云是使用基于深度图融合的方法得到并显示在 Meshlab 中。为了更好的衡量构图的精度, 本文根据 Turner^[49] 测量构图精度的方法, 同样采用设置地面控制点的方式, 如图 5.7(a) 所示, 其中红色圆圈标记的白色纸张为设置的标记点。设置地面控制点的操作就是在构图区域使用高精度卫星定位系统测试真实场景某点(白色纸张的中心)的经纬度坐标, 而后标记该场景所成二维图像中对应点的像素坐标(白纸中心成像对应的像素坐标), 如表 5.1 所示。从而在构建三维地图的同时, 根据已知标注点的绝对地理位置坐标与其像素坐标得到整个三维模型的地理空间尺度, 可以直接作为 GIS 数据库的输入数据, 形成可以使用的地图信息。



图 5.7 地面控制点位置

本文在图 5.6 场景中设置了 9 个地面控制点, 最后根据成像模型测得这 9 个点的坐标与实际使用 RTK GPS 接收机测量的真实结果对比, 如表 5.2 所示, 每个测试点的精度分为东向与北向, 二者的平方根表示在表最后一列为绝对误差, 最后计算 9 个控制点的平均绝对误差为 1.586m, 由于地面控制点分布较均匀, 而地图中某些区域, 一般是边缘区域变形较大, 导致平均误差较大。

表 5.1 地面控制点坐标

| 图片名称 \ 坐标名称 | 东向 | 北向 | 图像 X 方向 | 图像 Y 方向 |
|--------------|-----------|------------|---------|---------|
| DJI_0050.JPG | 415462.60 | 4401226.15 | 2706 | 1615 |
| DJI_0054.JPG | 415462.60 | 4401226.15 | 1312 | 438 |
| DJI_0052.JPG | 415462.60 | 4401226.15 | 2355 | 440 |
| DJI_0077.JPG | 415489.00 | 4401201.80 | 1011 | 1033 |
| | | | | |

表 5.2 地面控制点误差测量

| 控制点序号 | 东向误差 | 北向误差 | 绝对误差 |
|-------|-------|-------|--------|
| 0 | 0.093 | 0.095 | 0.132 |
| 1 | 0.267 | 0.012 | 0.268 |
| 2 | 1.353 | 2.165 | 2.553 |
| 3 | 0.879 | 1.888 | 2.081 |
| 4 | 0.260 | 0.013 | 0.261 |
| 5 | 0.195 | 1.542 | 1.554 |
| 6 | 2.148 | 3.73 | 4.311 |
| 7 | 0.184 | 0.207 | 0.276 |
| 8 | 1.298 | 2.521 | 2.8355 |

与上文雾岚山庄数据集处理方法相似，通过79张图片可以得到自动化学院的三维地图结构，图 5.8(a)为稀疏点云在RVIZ中的显示效果，图 5.8(b)为稠密点云在Meshlab中的显示效果图。该数据集同样设置地面控制点以衡量构图精度，共有三个地面控制点，分布如图5.7(b)所示，该图是展示在Google Earth中的效果，可以看出没有设置落在构图边缘的控制点，控制点整体分布均匀。

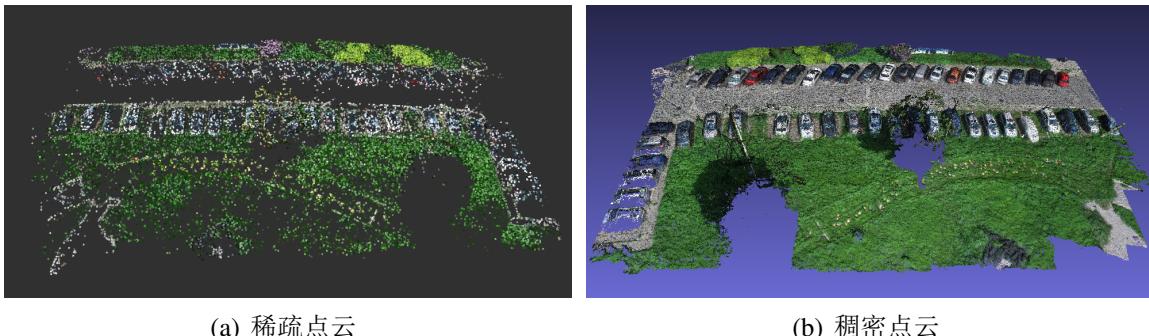


图 5.8 实验结果-自动化学院

表 5.3 地面控制点误差测量

| 控制点序号 | 东向误差 | 北向误差 | 绝对误差 |
|-------|-------|-------|-------|
| 0 | 0.184 | 0.378 | 0.421 |
| 1 | 0.856 | 1.676 | 1.883 |
| 2 | 0.141 | 0.262 | 0.298 |

最后通过实验得到真实值与测量值的对比结果，如表 5.3 所示，各列数据分别表示地面控制点的真实测量值与实验三维重建得到的模型对应点的东向、北向以及绝对误差值。由于三个点没有收到构图边缘变形的影响，所以结果稍好于雾岚山庄，得到平均控制点误差为0.867m。实际上由于本实验使用图片以及图片EXIF 信息中的

经纬高数据，但是实际上EXIF 中的经纬高数据不是很准确，大约误差在 $5m$ 左右，如果想要得到结果更高的地图，需要在构图时引入RTK，改善构图的精度。

5.4 道路提取精度分析

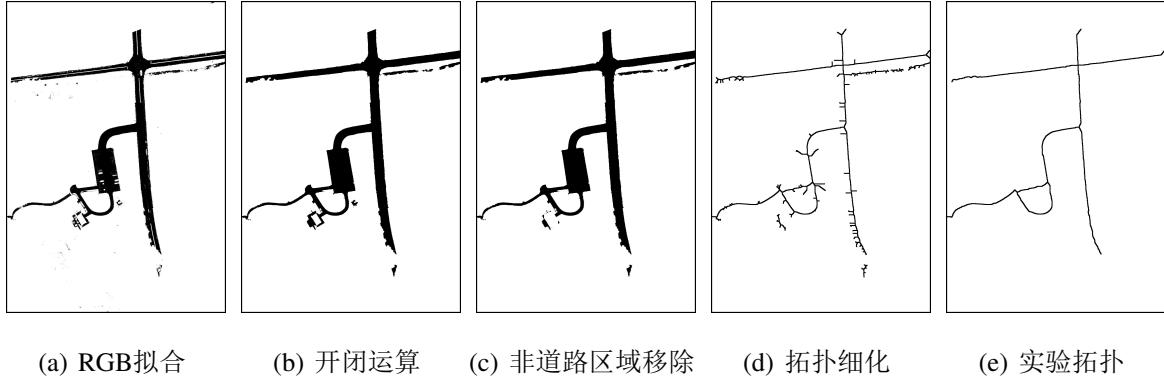


图 5.9 道路提取过程

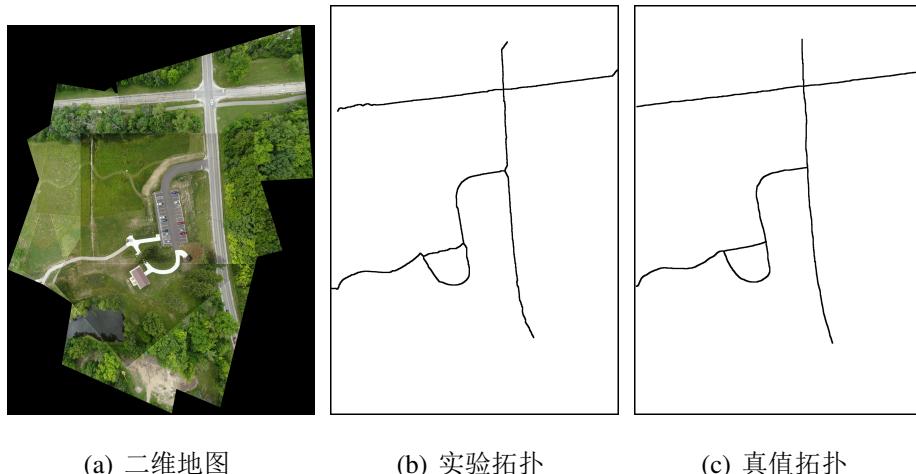


图 5.10 道路提取对比

根据第四章道路提取的原理，本节设计实验验证道路提取方法的精度。图 5.10(a)是用来提取道路的二维地图，由于单张航拍二维图片表示的道路区域无法形成拓扑网络，所以从单张航拍图片中提取道路，而后根据航拍原图间的单应性关系拼接得到如图 5.10(a)（原图拼接结果）和 5.9(a)（单张图片RGB拟合结果），而后应用本文提出的道路提取方法精细道路区域，拓扑细化得到最后的实验结果。整个过程如图 5.9所示，其中图 5.9(b)表示开闭运算平滑后的结果，图 5.9(c)为移除矩形非道路区域

后的结果，从中可以看出图左下方的房屋边沿被移除，图 5.9(d)为采用RDP算法进行拓扑细化后的结果，可以看出初始拓扑有很多小枝丫，使用逐渐递减的阈值去除小枝丫后如图 5.10(b)所示。

使用道路提取的方法得到道路拓扑网络后，图 5.10(a)分别展示了原图，提取道路（图 5.10(b)）与真实道路（图 5.10(c)），两幅图中道路拓扑是由单像素构成，通过膨胀真实道路，计算提取道路的像素不在膨胀后的真实道路区域的比例，绘制图 5.11，其中横轴代表膨胀时使用的核函数的大小，单位是像素，而且一个像素大约代表地理空间 $16cm$ ，核函数越大，膨胀后的真实道路区域越大，不在其区域的提取道路的像素越少。从图中可以看出核函数大小为10个像素，即 $1.6m$ 时，有接近90%提取的道路在膨胀区域内，也就是90%的道路误差小于 $1.6m$ ，同时可以看出80% 的道路误差小于 $1m$ 。

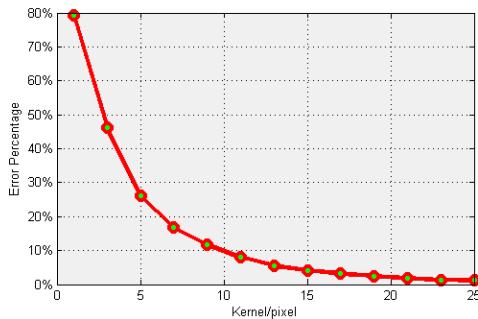


图 5.11 道路提取精度曲线

5.5 本章小结

本章主要是本文提出的算法的实验结果与结论的总结说明，可以分为三部分。首先是二维地图构建使用的传感器以及应用于实际比赛的构图结果，并就影响构图精度的因素做了说明。其次是关于三维重建的实验平台与重建结果精度的说明，定量的计算了两个数据集重建精度。最后是道路提取的精度测量，展示了每个过程的实验结果，并就提取的道路精度进行评价。实验的结果验证了本文提出的方法的有效性与真实性。

总结与展望

工作总结

无人驾驶的快速发展使高精度地图构建显得越来越重要，针对这一问题，本文提出了构建高精度二维地图与三维地图的方法，同时为了更好的将地图应用于无人驾驶中，本文提出了从高空视角的航拍图像中提取道路的方法。本文的主要工作与创新点如下：

1. 本文设计了带有高精度定位信息的全景图像采集系统，并通过坐标变换、图像拼接等流程，提出了构建大范围二维地图的方法，并使用GIS软件配准地图，构建适用无人驾驶使用的地图数据库。该数据库应用于真实无人车比赛，获得较好的效果。
2. 本文使用航拍图像结合当前先进的三维重建技术，获得地理空间的三维场景。其中使用图片EXIF信息中的GPS数据辅助图片匹配过程，提高了数据集图片特征匹配的效率。本文也充分利用相机GPS位置最小化相机与GPS位置的距离多视图间的旋转平移关系，作为光束平差法的一部分。
3. 本文采用多种方法避免将错误匹配点用于重建，包括最近邻匹配去除模糊的匹配，使用对极约束计算匹配点间的几何关系，从而去除一部分误差大的匹配点，通过track序号的方式删除匹配的逻辑关系不正确的匹配点等。
4. 本文研究了航拍视角的道路提出方法，并对提取的道路进行拓扑细化，提取由单像素构成的道路拓扑网络，可以直接用于GIS软件中生成线数据集，用于无人车导航。

未来工作展望

针对无人驾驶的高精度地图的构建一直是一个非常活跃的研究领域，随着相关传感器技术与感知技术的提高，构建高精度的三维地图，并将三维地图用于无人驾驶仿真与路测的研究将进一步深入，结合本文的研究内容与最新进展，有以下方面有待提高：

1. 本文设计的二维地图构建方法对远离全景相机的地面区域存在变形，不准确的特定，在未来可以加入激光雷达，将激光雷达点云与全景相机配准融合，得到高精度三维的带有真实纹理的点云数据。
2. 本文在基本的SfM框架上进行研究，如选取更合适的两视图用于初始化重建，

对于有序图片提高重建效率，使用GPU进行并行计算等，这些问题都没有涉及。因此在未来的研究中可以尝试吸取更先进的SfM算法思想，进一步提高完善三维重建技术。

3. 本文提出的道路提取算法采用简单有效的拟合道路RGB信息，但是对于道路颜色特征差别较大的区域，效果不理想，尤其是房屋的干扰很难消除，所以在未来将采用深度学习的方法提取道路的抽象特征，并应用于大范围道路提取，相信道路训练样本足够多时，算法将获得较好的效果。

参考文献

- [1] Mattyus G, Wang S, Fidler S, et al. HD Maps: Fine-Grained Road Segmentation by Parsing Ground and Aerial Images[C]// Computer Vision and Pattern Recognition. IEEE, 2016:3611-3619.
- [2] Wegner J D, Montoya-Zegarra J A, Schindler K. A Higher-Order CRF Model for Road Network Extraction[C]// Computer Vision and Pattern Recognition. IEEE, 2013:1698-1705.
- [3] Lin Y, Saripalli S. Road detection from aerial imagery[C]// IEEE International Conference on Robotics and Automation. IEEE, 2012:3588-3593.
- [4] 魏宇峰. 高分辨率遥感影像道路信息提取关键技术研究与实现[D]. 北京理工大学, 2010.
- [5] 汪夕明, 遥感影像道路提取方法研究与实现[D], 北京, 清华大学, 2011.
- [6] Xia W, Hongmei T, Yang Y等. Study on Road Extraction Method in Remote Sensing Image[A]. 2012 International Conference on Industrial Control and Electronics Engineering (ICICEE)[C]. 2012: 1578 – 1580.
- [7] Wu C, Agarwal S, Curless B, et al. Multicore bundle adjustment[C] Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011: 3057-3064.
- [8] Wu C. Towards linear-time incremental structure from motion[C] 3DTV-Conference, 2013 International Conference on. IEEE, 2013: 127-134.
- [9] Agarwal S, Furukawa Y, Snavely N, et al. Building rome in a day[J]. Communications of the ACM, 2011, 54(10): 105-112.
- [10] Snavely N, Seitz S M, Szeliski R. Photo tourism: exploring photo collections in 3D[C]//ACM transactions on graphics (TOG). ACM, 2006, 25(3): 835-846.
- [11] Diosi A, Taylor G, Kleeman L. Interactive SLAM using laser and advanced sonar[C]//Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005: 1103-1108.
- [12] Cole D M, Newman P M. Using laser range data for 3D SLAM in outdoor environments[C]//Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. IEEE, 2006: 1556-1563.
- [13] Leonard J J. Large-scale concurrent mapping and localization[C] Proceedings of SPIE. 2000, 4196: 370-376.

- [14] 杨祥红, 王希彬. 无人机主动SLAM 技术研究[J]. 系统仿真技术, 2016 (2016年04): 283-287.
- [15] 马跃龙, 赵勇, 曹雪峰. 一种基于单目SLAM 的无人机序列图像拼接方法[J]. 系统仿真学报, 2016 (2016年10): 2349-2355.
- [16] 蒙山, 唐文名. 单目SLAM 直线匹配增强平面发现方法[J]. 2017.
- [17] 陈含欣. 无人机视觉特征提取及同步定位与地图构建方法研究[D].南京航空航天大学,2016.
- [18] 刘洋, 章卫国, 李广文, 等. 一种三维环境中的无人机多路径规划方法[J]. 西北工业大学学报, 2014, 32(3): 412-416.
- [19] 谢斌, 张开石, 潘华东, 等. 未知环境中无人机适航地图构建与地面目标跟踪[J]. 浙江大学学报: 工学版, 2010 (1): 118-123.
- [20] 邹智荣. 动态环境下自主车同时定位与建图研究[D]. 中南大学, 2012.
- [21] 张毅, 尹亮. 无人车自主导航的环境建图技术[J]. 兵工学报, 2014 (S1): 32-37.
- [22] 段华旭, 闫飞, 庄严, 等. 无人车基于双目视觉的同时定位与地图构建[J]. 华中科技大学学报: 自然科学版, 2015 (S1): 319-323.
- [23] Fan Q, Yi Y, Hao L, et al. Semantic motion segmentation for urban dynamic scene understanding[C]//Automation Science and Engineering (CASE), 2016 IEEE International Conference on. IEEE, 2016: 497-502.
- [24] 李玉. 基于3D激光点云的无人车城市环境SLAM问题研究[D]. 北京理工大学, 2016.
- [25] 宋晓虎, 朱纪洪. 基于OpenStreetMap和90 m SRTM的地面对导航电子地图实现[J]. 地球信息科学学报, 2017, 19(7):901-908.
- [26] 许谦. 基于无人车环境感知信息的GIS动态更新算法研究[D]. 北京理工大学, 2015.
- [27] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [28] Engel J, Schops T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C] European Conference on Computer Vision. Springer, Cham, 2014: 834-849.
- [29] Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time[C]//Robotics: Science and Systems. 2014, 2.
- [30] Carrivick JL, Smith M W, Quincey D J. Structure from Motion in the Geosciences[M]. John Wiley & Sons, 2016.
- [31] Jianhua Wang, Qiming Qin, Xiucheng Yang, Jun Wang, Xin Ye, Xuebin Qin. Automated road extraction from multi-resolution images using spectral information and texture[J]. IEEE Geoscience

- and Remote Sensing Symposium. 2014. 533-536.
- [32] Mattyus G, Wang S, Fidler S, et al. Enhancing Road Maps by Parsing Aerial Images Around the World[C] IEEE International Conference on Computer Vision. IEEE, 2015:1689-1697.
- [33] Lowe D G, Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [34] Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision & Image Understanding, 2008, 110(3):346-359.
- [35] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C] IEEE International Conference on Computer Vision. IEEE, 2012:2564-2571.
- [36] 高翔, 张涛等, 视觉SLAM十四讲: 从理论到实践[B] 电子工业出版社, 2017-3:138
- [37] Rosten E, Drummond T. Machine learning for high-speed corner detection[J]. Computer Vision - ECCV 2006, 2006: 430-443.
- [38] Harris C, Stephens M. A combined corner and edge detector[C] Alvey vision conference. 1988, 15(50): 10.5244.
- [39] Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comput. Vis. 30(2), 79 – 116 (1998)
- [40] Lowe D G. Object Recognition from Local Scale-Invariant Features[C] iccv. IEEE Computer Society, 1999:1150.
- [41] Multiple View Geometry in Computer Vision Second Edition, Richard Hartley and Andrew Zisserman, Cambridge University Press, March 2004:105
- [42] Hartley R, Zisserman A. Multiple view geometry in computer vision[J]. Kybernetes, 2003, 30(9/10):1865 - 1872.
- [43] P. S. Heckbert and M. Garl, “Survey of polygonal surface simplification algorithms,” 1997.
- [44] C.-H. Choi, J.-B. Song, W. Chung, and M. Kim, “Topological map building based on thinning and its application to localization,” in Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, vol. 1. IEEE, 2002, pp. 552 – 557.
- [45] Shen S. Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes.[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2013, 22(5):1901.

- [46] Furukawa Y, Ponce J. Accurate, Dense, and Robust Multiview Stereopsis[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2010, 32(8):1362-1376.
- [47] Bleyer M, Rhemann C, Rother C. PatchMatch Stereo - Stereo Matching with Slanted Support Windows[C]// British Machine Vision Conference. 2011:14.1-14.11.
- [48] 吴福朝. 计算机视觉中的数学方法[M]. 科学出版社, 2008.
- [49] Turner D, Lucieer A, Watson C. An Automated Technique for Generating Georectified Mosaics from Ultra-High Resolution Unmanned Aerial Vehicle (UAV) Imagery, Based on Structure from Motion (SfM) Point Clouds[J]. Remote Sensing, 2012, 4(5):1392-1410.

攻读硕士学位期间发表论文与研究成果清单

项目成果

- [1] 第一作者, UAV-aided Large-scale Map Building and Road Extraction for UGV [C]. IEEE-CYBER, 2017 IEEE International Conference on IEEE, (EI检索)
- [1] 一种适用无人驾驶车辆的双天线差分GNSS与里程计组合导航方法[P]. 北京: CN

致谢