



硕士学位论文

面向无人机自主堆体体积测量的视觉定位 及重建方法研究

姓 名：何士波
学 号：1732940
所在院系：电子与信息工程学院
学科门类：工学
学科专业：控制工程
指导教师：岳继光 教授
副指导教师：董延超 副教授

二〇一九年十二月



同濟大學
TONGJI UNIVERSITY

A dissertation submitted to
Tongji University in conformity with the requirements for
the degree of Master of Science

**Visual localization and reconstruction method for
autonomous bulk measurement of unmanned aerial
vehicle**

Candidate : He ShiBo
Student Number : 1732940
School/Department : College of Electronics and
Information Engineering
Discipline : Engineering
Major : Control Engineering
Supervisor : Prof. Yue Jiguang
Associate Supervisor : V.P. Dong Yanchao

December, 2019

学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构递交论文的复印件和电子版；在不以盈利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：

年 月 日

同济大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

摘要

在现代工业中，随着工业生产水平的提高，为了加强对堆体物料库存量的监控，针对堆体积测量的需求也逐步提升。传统测量方案一般都是选用激光雷达，测重地磅等复杂仪器来完成测量工作，这些仪器可以一定程度的完成测量工作并获取测量真实值。但依旧存在很多问题限制以上测量技术在行业内的广泛应用，例如测量仪器成本过高，安装复杂，单一场景内的仪器难以复用；测量过程需要人力参与，难以实现全自动化的测量流程；测量精度年久失衡，难以保证测量精度。针对以上情况，本文提出了一种在封闭电厂环境下，通过无人机的自主定位和飞行，对堆体进行图像采集工作，随后根据采集的图像进行三维重建得到高精度堆体点云模型，对点云模型估计体积值以获取堆体体积的完整测量流程。本文所提出方案完全基于计算机视觉实现，成本可控，且能在各种场景中复用。

针对无人机的自主定位和飞行方法研究。本文提出在封闭无 GPS 的环境中，使用纯视觉的方式来完成无人机的自主定位和飞行工作。通过在场景中放置二维码，来解决视觉 SLAM 无法确定尺度的问题，并且结合 SLAM 坐标系和二维码坐标系对无人机提供真实世界坐标系下的位姿信息。

针对连续图像的高精度堆体三维重建方法研究。本文提出一种结合 SLAM 结果的堆体三维重建方法，以有序图像作为 SLAM 输入，关键帧数据集作为输出提供给三维重建系统以获取高精度点云。解决传统三维重建图像匹配耗时，点云精度低的问题。

针对堆体点云进行体积估计方法研究。本文提出一种基于计算机视觉方案的堆体体积估计方法，依次对生成的点云进行滤波获取高精度点云，估计三维点云的实际尺度，确定点云模型水平面等工作，最后根据纯点云信息估算场景中堆体的体积大小。

基于真实环境下，对堆体体积测量进行分模块测试。实验结果表明，无人机的定位误差在高度方向可以控制在 0.2m 以内，整体定位精度在 5% 以内，该精度完全可以提供给无人机循迹使用；改进后的三维重建模块则能加快匹配流程，获取较高精度的堆体三维点云，以满足体积测量的需求；最后的体积测量模块，可以估计出准确的尺度大小和水平面解析方程，通过对点云体积的求解获得堆体体积，测量误差可以控制在 2% 以内。

关键词：立体视觉，SLAM，无人机定位，三维重建，堆体体积测量

ABSTRACT

In modern industry, with the improvement of industrial production level, in order to strengthen the monitoring of heap material inventory, the demand for heap volume measurement is also gradually increasing. Traditional measurement schemes usually use laser radar, weighometer and other complex instruments to complete the measurement work, these instruments can complete the measurement work to a certain extent and obtain the real value of the measurement. However, there are still many problems that limit the wide application of the above measurement technology in the industry. The measurement process needs human participation, so it is difficult to realize the fully automated measurement process; The measurement accuracy is out of balance for years, so it is difficult to guarantee the measurement accuracy. In view of the above situation, this paper puts forward a closed environment of power plant, through independent positioning and flight of the unmanned aerial vehicle (uav), the pile body image acquisition work, then according to the collected image three dimensional reconstruction of pile body point cloud model has high precision, the point cloud model to estimate the volume value for the pile body volume of the complete measuring process. The scheme proposed in this paper is completely based on computer vision, with controllable cost, and can be reused in various scenarios.

Research on autonomous positioning and flight method of uav. This paper proposes to use pure vision to complete the autonomous positioning and flight of uav in closed environment without GPS. The two-dimensional code is placed in the scene to solve the problem that visual SLAM cannot determine the scale, and the combination of the SLAM coordinate system and the two-dimensional code coordinate system provides the positioning information of the uav in the real world coordinate system.

Research on 3d reconstruction of continuous image with high precision. In this paper, a method of three-dimensional reconstruction of the heap combined with SLAM results is proposed, in which the ordered image is taken as the input of SLAM, and the key frame data set is provided to the three-dimensional reconstruction system as the output to obtain the high-precision point cloud. To solve the problem of time-consuming matching of traditional 3d reconstruction image and low accuracy of point cloud.

The volume estimation method of the point cloud is studied. In this paper, a method of heap volume estimation based on computer vision scheme is proposed, which suc-

sively filters the generated point cloud to obtain the high-precision point cloud, estimates the actual scale of the three-dimensional point cloud, determines the horizontal plane of the point cloud model, and finally estimates the volume of the heap in the scene based on the pure point cloud information.

Based on the real environment, the volume measurement of the reactor is tested by module. The experimental results show that the positioning error of uav can be controlled within 0.2m in the altitude direction, and the overall positioning accuracy can be controlled within 5%, which can be fully used for tracking of uav. The improved 3d reconstruction module can speed up the matching process and obtain 3d point cloud of the heap with high accuracy to meet the demand of volume measurement. Finally, the volume measurement module can estimate the accurate scale size and the analytic equation of the horizontal plane. The volume of the heap can be obtained by solving the volume of the point cloud, and the measurement error can be controlled within 2%.

Key Words: keywords

目录

插图索引	VI
第 1 章 绪论	1
1.1 研究背景和意义	1
1.1.1 研究背景	1
1.1.2 研究意义	2
1.2 国内外研究现状	3
1.2.1 SLAM 的研究现状	3
1.2.2 三维重建的研究现状	4
1.2.3 基于视觉体积测算的研究现状	6
1.3 待解决问题	6
1.4 主要研究内容和技术路线	7
1.5 章节安排	8
第 2 章 面向堆体测量的无人机自主定位研究	10
2.1 引言	10
2.2 二维码识别与使用方法研究	11
2.2.1 二维码检测与识别过程	11
2.2.2 利用二维码估计相机外参	12
2.3 联合二维码的 SLAM 系统描述	12
2.3.1 引入二维码对 SLAM 优化研究	12
2.3.2 包含二维码的地图描述	13
2.3.3 联合二维码的 SLAM 过程描述	15
2.4 面向堆体无人机定位系统设计	19
2.4.1 无人机相机坐标变换数学描述	19
2.4.2 获取真实世界坐标系方法	22
2.4.3 无人机自主定位设计	24
2.5 本章小结	28
第 3 章 面向堆体的三维重建方法改进	30
3.1 引言	30
3.2 三维重建方法研究	30
3.2.1 三维重建流程概述	30
3.2.2 三维重建详细过程解析	32
3.2.3 当前三维重建存在的问题	38
3.3 三维重建优化方案研究	38
3.3.1 增加实效性方案	38
3.3.2 增加精确性方案讨论	41

3.4 本章小结	44
第4章 基于纯视觉的堆体体积测量方法研究	45
4.1 引言	45
4.2 堆体水平面方程解析方法	46
4.2.1 2D/3D 点坐标检测	46
4.2.2 平面方程优化方法	47
4.3 堆体尺度估计方法	48
4.3.1 位姿法估计	48
4.3.2 空间坐标点估计尺度	50
4.4 堆体体积测量方法研究	50
4.4.1 有效 3D 点集获取	51
4.4.2 Delaunay 三角网	53
4.4.3 积分三棱柱	54
4.5 本章小结	55
第5章 无人机视觉定位与堆体重建及体积测量测试与分析	56
5.1 引言	56
5.2 无人机视觉定位系统测试	56
5.2.1 无人机视觉定位系统实验步骤设计	56
5.2.2 视觉定位系统实验结果分析	58
5.3 三维重建系统测试	61
5.3.1 匹配实效性测试	61
5.3.2 点云精度测试	63
5.4 堆体体积测量系统测试	64
5.4.1 解析水平面方程	64
5.4.2 估计堆体实际尺度	66
5.4.3 堆体体积估计	67
5.5 本章小结	72
第6章 总结和展望	73
6.1 全文总结	73
6.2 未来工作展望	74
致谢	75
参考文献	76
个人简历、在学期间发表的学术论文与研究成果	80

插图索引

图 1.1	经典 Marr 视觉信息处理过程示意图	1
图 1.2	三维重建结果示意图	4
图 1.3	技术路线图	8
图 2.1	二维码原始图和梯度图	11
图 2.2	二维码识别结果示意图	12
图 2.3	包含二维码的构建地图示意图	14
图 2.4	基于二维码的 SLAM 流程图	15
图 2.5	针孔相机模型	21
图 2.6	坐标系转化示意图	21
图 2.7	二维码布置示意图	23
图 2.8	获取真实位姿完整流程图	24
图 2.9	PointGrey 工业相机示意图	26
图 2.10	无人机采集堆体图像序列	28
图 3.1	三维重建流程图	31
图 3.2	三维重建具体实现步骤图	31
图 3.3	堆体图像采集示意图	32
图 3.4	FAST 特征点	34
图 3.5	特征提取与匹配结果示意图	35
图 3.6	误差投影示意图	36
图 3.7	极线搜索示意图	37
图 3.8	堆体稀疏重建示意图	38
图 3.9	自然场景三维重建结果示意图	39
图 3.10	融合 SLAM 结果的三维重建流程图	40
图 3.11	前视 SLAM 整体框架	42
图 3.12	SLAM 系统框架图	43
图 3.13	离线 3D 重建技术流程图	43
图 4.1	体积测算流程图	45
图 4.2	二维码检测示意图	46
图 4.3	根据空间点估计尺度流程图	50
图 4.4	纯视觉测量体积流程图	51
图 4.5	三维重建感兴趣区域示意图	52

图 4.6	点云统计滤波前后效果对比示意图	52
图 4.7	Delaunay 三角网示意图	54
图 4.8	箱形图示意图	54
图 5.1	实验场景示意图	57
图 5.2	视觉算法生成地图	57
图 5.3	无人机自主飞行轨迹设计图	58
图 5.4	二维码 Z 方向数据	59
图 5.5	各方向 GPS 和视觉算法对比图	60
图 5.6	各方向 GPS 和视觉算法误差对比图	60
图 5.7	无人机轨迹对比示意图	61
图 5.8	地下车库环形图像	63
图 5.9	地下车库三维重建对比图	64
图 5.10	堆体场景图像序列	65
图 5.11	三维重建点云结果示意图	65
图 5.12	包含水平面的三维重建结果示意图	66
图 5.13	两帧之间获取绝对尺度	67
图 5.14	不同堆体场景预览图	69
图 5.15	不同堆体场景稀疏点云示意图	70
图 5.16	不同堆体场景稠密点云示意图	71

第1章 绪论

1.1 研究背景和意义

1.1.1 研究背景

时人类通过双眼来探索与发现世界，在接收外部信息的方式中，有不到三成来自于听觉、触觉、嗅觉等感受器官，而超过七成、最丰富、最复杂的信息则通过视觉进行感知的。计算机视觉便是一种探索给计算机装备眼睛（摄像头）与大脑（算法）的技术，以使计算机能够自主独立的控制行为、解决问题，同时感知、理解、分析外部环境。

20世纪60年代，计算机视觉得到了最初的发展，该阶段的研究重心主要体现在如何从二维图像中恢复出如立方体、圆柱体等立体化的三维形状，解释各个物体的空间位置关系。1982年David Marr从信息处理的角度对数学、神经生理学、计算机图形学等学科的研究成果进行了归纳总结，并在此基础上提出了一系列计算机视觉理论，经典Marr视觉信息处理过程如图1.1所示。得益于这个完整明确的理论体系，计算机视觉得到了蓬勃的发展。

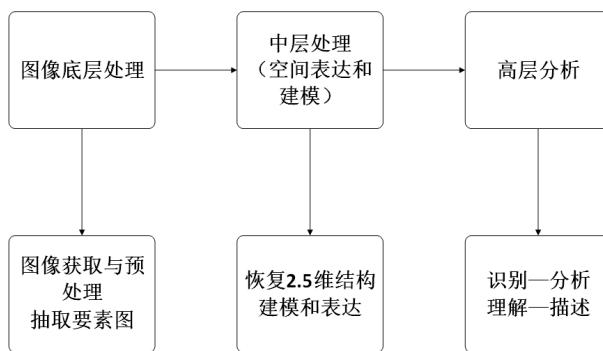


图1.1 经典Marr视觉信息处理过程示意图

无人机在监控、搜救、侦查等领域有着广泛的应用，且具备灵活性高、维持低速飞行的优点，目前人们对无人机也提出了更高的要求，指出无人机需要向更加智能化、实用化的方向发展。因此无人机的自主定位和飞行就是必须要达到的要求。对于控制无人机自主飞行这一问题上，多种传感器和处理器都协同发挥着重要作用，通过传感器来估计自身位姿，构建环境地图。传统无人机的自主定位和飞行都是依托于激光雷达来实现的，但是考虑到激光雷达成本高、体积大，随着研究的深入和需求的加大，科学家发现摄像机凭借其重量轻、功耗低等优点，可以很好地取代激光雷达来识别图像几何结构，因此计算机视觉技术和无人机

飞行技术的关联越来越密切。

近年来，图像的三维重建工作在计算机视觉领域所占比重越来越大，无人驾驶，智能快递概念的出现也加速这三维重建技术的发展。三维重建在质量和性能方面相较之前都有了极大的提升。其主要应用是自动地对自然场景进行建模，可以使用于室内和室外的场景，而不受控制的环境通常影响室外场景，如密集建筑群，或者复杂的原始森林等。对于这些场景，虚拟现实和计算机模拟可以被用来分析工作环境和工作难度。

在当前的一些工业场景中，需要对其中场景中的物料进行定量监控进，主要包括场景的高度，面积，体积，甚至于温度，湿度分布等，传统方式中主要采用雷达扫描场景的方式来进行，但考虑到该雷达本身成本较高，受场景的限制也很大，在室外环境或者大尺度环境下，就难以发挥作用。现在更多采用计算机视觉的方式来解决这些问题，可以根据三维重建的点云结果测量以上所描述的几何特性，这样做只需要结合摄像机和测量算法即可实现。

1.1.2 研究意义

本文的研究目标是让无人机能够在全封闭或者半封闭的环境中，完全基于纯视觉的方法完成自主飞行任务，并且在飞行的过程中，采集待测堆体的实拍图像，进行三维重建，最终根据三维重建的点云结果结合数学处理和视觉处理测量待测堆体的体积值。

传统无人机的飞行都需要依靠操作员手动控制，或者是基于 GPS 定位结果进行巡航，本文提出了一种完全基于纯视觉的方法来进行无人机定位的导航得到的方法，该方法可以在全封闭无 GPS 的环境中，为无人机提供定位信息，解决了场景限制的问题，并且整个无人机的飞行过程可以完全自动化的运行，并对场景进行图像采集。对无人机采集的图像进一步可以进行三维重建，以获取更加直观的 3D 观测模式，更加精确的点云模型和跟快的重建速度都将是三维重建未来发展的目标。以三维点云来测量体积，和传统的测量方式相比较也能极大的节省成本，软件算法的复用也利于这一需求的广泛推广。

因此展开面向无人机自主堆体体积测量的视觉定位及重建方法的研究一方面可以融合计算机视觉领域内的多项技术，另一方面也具有十分重要的工程意义。

1.2 国内外研究现状

本文主要涉及到的理论方法有及时定位与地图构建（Simultaneous localization and mapping，简称 SLAM），三维场景重建，视觉体积测量等几个方面对国内外研究现状和发展动态进行描述。

1.2.1 SLAM 的研究现状

SLAM（即时建图与定位）是一种在定位导航的同时，进行构图的技术^[1]。最早的 SLAM 技术还不是使用视觉的方法，而是使用声波传感器或者激光以及惯性测量单元实现环境建模和自身定位，直到 21 世纪，Stephen Se 等人首次使用图像的特征点实现视觉 SLAM^[2]，之后由 Davison 使用 EKF 框架实现了最早的单目实时 SLAM 系统^[3]，奠定了单目系统的基础；Davison 在 2007 年成功实现基于单相机的纯视觉 SLAM 系统，算法的关键是在线建立 2D 点到 3D 点的映射关系，并且使用实时运动模型估计相机的位置^[4]；Mur-Artal 使用 ORB 特征点作为地图构建特征点，大幅度降低了点云的数量，并且使用回环检测的方法使定位与建图的精度都大幅提升^[5]；随着硬件计算能力和数据储存的提升，提取目标深度信息的技术得到了很大的发展，戚传江等人使用 2D slam 的解决方案，采用多传感器数据融合的方法，完成多自由度位姿检测，拓展了 SLAM 的应用场景；Whelan 的实验通过使用体积融合的方法实现了实时大范围的稠密 RGB-D 的 SLAM 系统^[6]。

Durrant-Whyte 和 Bailey 首先对前 20 年里 SLAM 的发展做出了详细的历史回顾，并提出了概率方法和数据融合^[7, 8]，Aulinas 等人提出在 SLAM 中添加滤波方法减少噪音影响^[1]，Grisetti 等人就 SLAM 后端进行详细阐述^[9]，Dissanayake 研究了 SLAM 的基本性质，包括可观测性、收敛性和一致性等^[10]。近几年来，SLAM 的发展更多的开始和机器人领域相结合，Saeedi 提出了多协同机器人 SLAM 解决方案^[11]，Stachniss 发布了在 SLAM 领域机器人开发手册^[12]。

应用到目标跟踪领域，单纯点云集还是无法满足要求，因此需要将点云数据语义化，Reiger 使用关系树的方法实现物体的语义识别^[13]，这项技术对于目标跟踪是很重要的；之后 Sarkar 在 Reiger 的研究基础上结合 FastSLAM 的方法，使得识别速度更快，鲁棒性更强；Zhang, G 等人使用基于线条的 SLAM 算法^[14] 提高物体识别的准确率，该方法能够对物体的边沿与轮廓进行稳定的识别。

当使用单目相机运行 SLAM 算法时，依旧还存在很多的挑战，ORB-SLAM2^[15] 和 LDSO^[16] 可能是目前单目 SLAM 中最先进的方法。然而，这些方法还存在很多的局限性，无地图复用，无法纯旋转，场景需丰富等，并且用于重定位的方法^[17] 在视点变化、重复模式和随时间变化时的性能有限。另一种

估计相机姿态的方法是使用放置在环境中的人工标记，最近的 SPM-SLAM^[18] 解决了之前描述的一些限制，它使用二维码而不是自然特征，但是也存在场景中摆放大量二维码的问题。UcoSLAM^[19] 则提出了一种结合自然点和人造二维码的 SLAM 运行方法。

1.2.2 三维重建的研究现状

照相机/摄像机是将一个三维场景或物体投影到二维平面上，但是在降维的过程不可避免地会损失存在信息，而利用三维重建技术，就是从获取到的二维图像中复原原始三维场景或物体，三维重建的结果如图 1.2 所示。

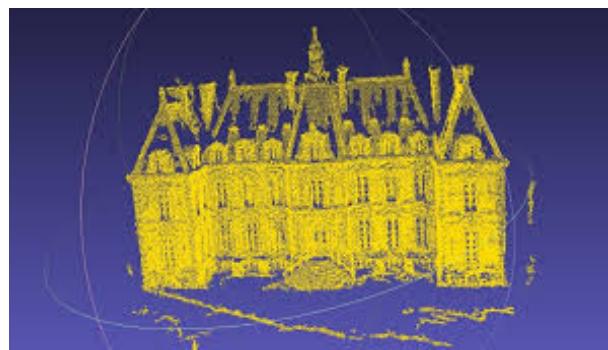


图 1.2 三维重建结果示意图

三维重建技术就是获取环境中物体的三维信息，从而确定不同物体所在位置并建立相应空间模型^[20]，具有高效率的三维重建技术也已经被应用在众多领域当中，其中包括城镇建模、无人作战、文物保护和地图重建等，相关的重建算法及流程也愈加完善^[21]。基于图像的三维重建从获取多视图图像开始，逐步通过相应的算法恢复场景中物体的三维结构^[22]。基于图像的三维重建技术是当今计算机视觉领域的热门方向^[23]，常见的三维重建的方法如表 1.1 所示。

基于单目视觉的三维重建技术是仅使用一台相机来进行三维重建的方法，这种方法简单方便、灵活可靠、使用范围广，可以在多种条件下进行非接触、自动、在线的测量和检测，主要方法为运动恢复结构法 (SfM)，常用的 SfM 方法采用的是多视角几何法。通常，包括以下四个步骤：

1) 特征提取与匹配。特征提取是首先用局部不变特征进行特征点检测，再用描述算子来提取特征点。Moravec^[31] 提出了用灰度方差来检测特征角点的方法。Harris^[32] 在 Moravec 算法的基础上，提出了利用信号的基本特性来提取图像角点的 Harris 算法。Smith 等人^[33] 提出了最小核值相似区，即 SUSAN 算法。Lowe^[34] 提出了一种具有尺度和旋转不变性的局部特征描述算子，即尺度不变特征变换算子，这是目前应用最为广泛的局部特征描述算子。Bay^[35] 提出了一种更快的加速鲁棒性算子。特征匹配是在两个输入视图之间寻找若干组最相似的特

表 1.1 常见三维重建方法

方法	复杂度	实时	效果	应用场景
明暗恢复形状法 ^[24]	简易	否	光照影响大, 效果较差	难以用于室外
几何光学测量法 ^[25]	人工调节	否	效果精细	难以用于贫纹理环境
投影光栅相位法 ^[26, 27]	至少 3 幅图像	否	弱光环境效果好	难以用大尺度场景
单激光线扫描法 ^[28]	设备复杂	是	效果精细, 速度快	适用于小尺度场景
彩色条纹结构光法 ^[29]	仅需要一张彩图	是	精度低, 鲁棒性强	适用于彩色场景
复合结构光测量法 ^[30]	仅需要一张完整图	是	精度高	适用于投影和实物距离远
双目视差法 ^[28]	设备复杂且耗时	是	大尺度效果好	适用于大多数场景

征点来形成匹配。传统的特征匹配方法通常是基于邻域灰度的均方误差和零均值正规化互相关这两种方法。Grauman 等人^[36] 提出了一种基于核方法的快速匹配算法, 即金字塔匹配算法。Photo Tourism 系统在两两视图间的局部匹配时采用了基于近似最近邻搜索^[37] 的快速算法。

2) 多视图几何约束关系计算。多视图几何约束关系计算就是通过对极几何将几何约束关系转换为基础矩阵的模型参数估计的过程。Longuet-Higgins^[38] 最早提出多视图间的几何约束关系可以用本质矩阵在欧氏几何中表示。Luon^[39] 提出了解决两幅图像之间几何关系的基础矩阵。与此同时, 为了避免由光照和遮挡等因素造成的误匹配, 学者们在鲁棒性模型参数估计方面做了大量的研究工作, 在目前已有的相关方法中, 最大似然估计法、最小中值算法、随机抽样一致性算法三种算法使用最为普遍。

3) 优化估计结果。当得到了初始的射影重建结果之后, 为了均匀化误差和获得更精确的结果, 通常需要对初始结果进行非线性优化。在 SfM 中对误差应用最精确的非线性优化方法就是光束法平差。光束法平差是在一定假设下认为检测到的图像特征中具有噪音, 并对结构和可视参数分别进行最优化的一种方法。近年来, 众多的光束法平差算法被提出, 这些算法主要是解决光束法平差有效性和计算速度两个方面的问题。Ni^[40] 针对大规模场景重建, 运用图像分割来优化光束法平差算法。Engels^[41] 针对不确定的噪声模型, 提出局部光束法平差算法。Lourakis^[42] 提出了可以应用于超大规模三维重建的稀疏光束法平差算法。

4) 得到场景的稠密描述。经过上述步骤后会生成一个稀疏的三维结构模型, 但这种稀疏的三维结构模型不具有可视化效果, 因此要对其进行表面稠密估计, 恢复稠密的三维点云结构模型。近年来, 学者们提出了各种稠密匹配的算法。

Lhuillier 等人^[43] 提出了能保持高计算效率的准稠密方法。Furukawa^[44] 提出的基于面片的多视图立体视觉算法是目前提出的准稠密匹配算法里效果最好的算法。

1.2.3 基于视觉体积测算的研究现状

在测量领域, 测量精度、测量速度、测量数字化和自动化程度要求不断提高, 传统的接触式测量已经无法满足需要, 基于纯视觉的非接触式测量方法具有结构简单、测量精度高、实时性好可复用等特点以满足测量要求^[45]。目前, 国内外很多的科研机构对光斑定位、CCD 自动聚焦调整、测量误差等不足问题进行了研究, 并提出了相应的优化方法^[45]。在实际应用领域, 如德国 OPTO NCDT 系列和日本 LK 系列、LC 系列等激光三角测量传感器在测量精度、速度等性能方面也比以前有较好的提高^[46]。

物料体积的测量是工矿企业、港口码头等领域进行库存盘点和物料管理重要的工作, 其中堆积物体是典型的物料体积测量, 如钢铁冶金企业和矿石企业对物料体积的测量^[47]、港口码头对集装箱及堆积货物量^[48] 或船体^[49] 的测量、甚至集装箱或船舱内货体积的测量^[50], 都为合理利用资源和安排生产, 提高生产效率发挥了重要作用。根据测量原理的不同, 物料体积测量技术主要分为激光传感测量^[51] 和计算机视觉测量^[52, 53], 其中激光测量具有非接触、长距离和抗干扰的优点^[54-56], 更适合在户外大范围测量的应用。在视觉测量方面, 主要在获取被测物体的点云之后采用剖分法^[57, 58]、凸包^[59]、切片法^[50] 等方法进行体积计算。

对于室内场景大型堆体的测量工作, 熊友辉^[60] 提出一种便携式激光测量仪, 通过多站人工打点的方式进行盘点, 虽然该仪器存在测量周期长、精度差、人为影响大等缺点, 但仍是现在堆场测量的常用方法。张德津等^[61] 提出了一种基于多传感器的固定式激光测量方法, 该方法将激光扫描仪安装在外界载体(如取料机等)上对堆场进行测量。盛业华等^[62] 提出了一种基于三维激光扫描点云的多站拼接方法, 该方法围绕被测目标布设多个测量点, 将各测量获得的点云进行拼接进而获得完整的被测堆体模型。张小虎等^[63] 提出了激光投影线辅助下的堆场三维形貌摄影测量方法, 该方法围绕被测目标布设多个测量点, 将各测量获得的点云进行拼接进而获得完整的被测堆体模型。王海波等^[64] 提出了车载式多站盘点系统。

1.3 待解决问题

当前基于无人机自主飞行采集图像数据进行三维重建以及体积测量的方案还存在很多的待解决问题, 如:

-
- (1) 无人机在完全封闭环境中依靠纯视觉进行定位和建图，难以获取高精度的飞行位姿和地图信息。
 - (2) 基于三维重建算法对场景进行三维重建时，面临整个流程耗时长，输出点云噪音点大，需要对传统三维重建进行提升，以获取高精度强鲁棒性的大尺度地图。
 - (3) 基于三维点云的体积测算，点云中缺少水平面信息，尺度信息，导致无法直接获取到体积真值。

1.4 主要研究内容和技术路线

本文将针对目前无人机在无 GPS 的密闭环境中进行自主飞行存在的问题，采用计算机视觉的方式建立一套稳定，高精度的无人机自主定位系统，并对采集到的图像作为输入开发出一套能够生产高精度，强鲁棒的三维重建系统。针对获取到的三维点云，提出估计尺度，确定水平面的方法以获取感兴趣区域的堆体体积本文将针对上述功能开发出一套完整、全自动化的系统。所研究系统将得到以下指标：

- 1) 建立一套无人机自动定位系统，使得无人机自主定位结果与真实 GPS 定位数值误差在 0.5 米以内，位置差距在 2% 之内。
- 2) 建立一套基于视频流，并融合 SLAM 结果的和高精度三维重建系统，能够针对各种室内外的大尺度建筑物场景实现三维重建，场景场能将误差控制在 20cm 以内。
- 3) 建立一套堆体体积自主测量系统，可以快速估计出点云的尺度，水平面方程与堆体体积，测量误差控制在 2% 之内。
- 4) 建立一套基于无人机自主飞行采集图像数据进行三维重建以及体积测量的完整方案，实现快速全自动的测量流程。

结合研究内容，完成理论研究，系统实现以及测试实验与分析，技术路线如图 1.3 所示。

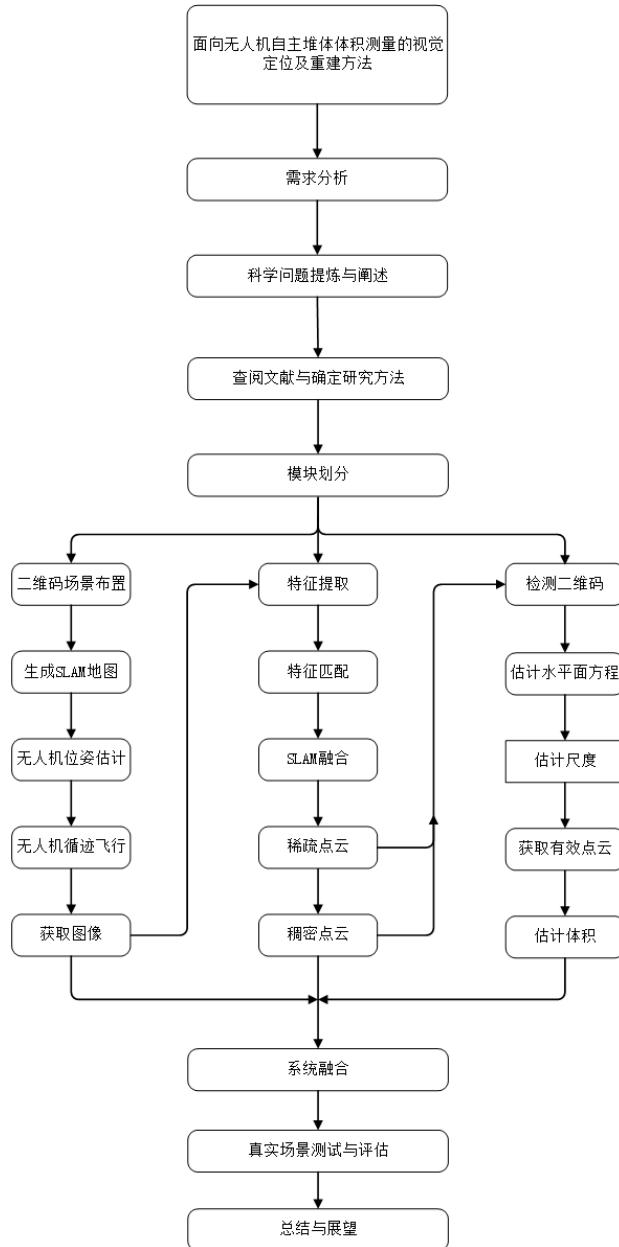


图 1.3 技术路线图

1.5 章节安排

总结研究问题，依据技术路线的规划，本文的章节安排如下所示：

第1章：查阅国内外“SLAM”、“三维重建”以及“体积测量”等相关领域研究文献及技术资料，对视觉 SLAM 的基本框架，三维重建的实现流程以及基于纯视觉的体积测量方法对比作了详细调研和综述，为相关研究工作的开展提供必要性依据，并在明确研究对象的基础上，提炼存在的科学问题，确定研究方法、目标和技术路线。

第2章：研究面向堆体体积测量的无人机自主定位系统，分析二维码的识别

和检测方法，利用二维码求解相机位姿的方法。深入分析结合二维码视觉标签后对视觉 SLAM 系统的优化情况，包含二维码的 SLAM 地图描述情况以及具体的过程描述。设计一种面向实际环境，依靠纯视觉 SLAM 系统的无人机定位系统，该系统能够提供真实尺度和世界坐标系的实际位姿。

第 3 章：研究基于传统三维重建的改进方法，简述传统视觉三维重建的一般过程，分析三维重建所存在的问题以及对应原因，针对每种问题提出改进和提升方式，结合 SLAM 生成 KeyFrame DataBase 增强三维重建生成点云结果。

第 4 章：研究基于纯视觉的对体体积测量方法，提出一种基于纯视觉的堆体体积测量流程，结合二维码视觉标签，解析堆体点云与实际场景之间的尺度大小，优化堆体水平面方程参数，提出获取有效点云方法，结合 Delaunay 三角网积分堆体点云体积，获取实际堆体场景体积。

第 5 章：分别对本文所提的无人机视觉定位，视觉三维重建方法改进以及纯视觉体积测量方法进行测试和分析。定量和定性分析无人机依靠纯视觉方法进行定位的精度以及误差原因，分析比较改进后的三维重建方法效果提升情况，设计实验测试堆体点云尺度估计结果和水平面方程估计结果，设计分组实验验证纯视觉方法下的堆体体积测量精度和稳定性。

第 6 章：总结本文的研究工作，指出面向无人机自主堆体体积测量的视觉定位及重建方法研究工作中存在的不足和改进空间，并为后续研究提出若干研究方向。

第 2 章 面向堆体测量的无人机自主定位研究

2.1 引言

基于 SLAM 方法的即时建图和定位方法目前已经发展到的相当成熟，尤其是在移动设备定位领域，基于视觉 SLAM 方法进行定位有着积极的应用。对于实际落地的场景，该系统输出的相机位姿和构建的地图必须具备真实尺度才能够进行导航与定位，此外依靠视觉方法构建的地图，其坐标系往往取决于初始化成功后的第一帧所建立的坐标系，难以与真实世界坐标系有所关联，这些问题的存在都限制着视觉 SLAM 在实际场景中的存在。

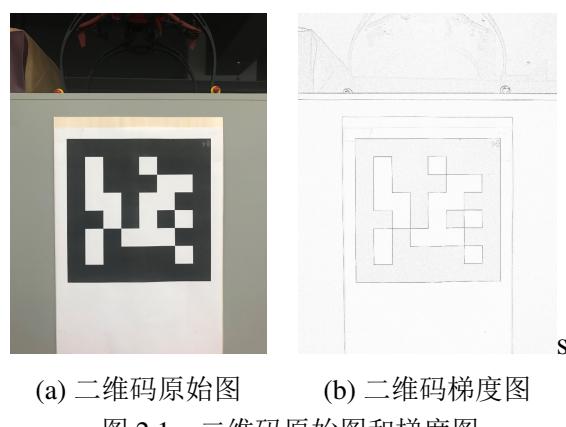
对于单目视觉 SLAM 算法，设备简易，处理数据较少，满足实时性的要求，但存在无法获取真实世界坐标系下的相机位姿以及无法获取地图实际尺度的问题；对于双目视觉的 SLAM 算法，可以解决地图尺度的问题，但面临成本较高，相机标定难度大且精度不高，鲁棒性较差等问题；对于融合视觉和 IMU 传感器的 SLAM 系统，可以获取真实尺度，以及得到真实世界坐标系下的相机和地图位姿，但该系统对 IMU 的精度要求较高，存在引入 IMU 后容易产生累计误差，难以初始化成功等问题。

针对上述问题，本章提出一种面向堆体场景进行测量功能的无人机自主定位方法，该方法将融合二维码视觉标签，在传统 SLAM 功能的基础上，可以得到相机和地图的真实尺度，并且根据一定的坐标转换，实时获取真实世界坐标系下的相机位姿和地图。在真实的堆体场景中，布置的二维码和普通的自然关键点相比，更加容易捕获，此外，二维码的布置固定，在重定位的流程中，往往会有更好的效果。该系统能够具备以下优良特性：

1. 能够对地图进行保存，复用和更新，通过不断完善的先验地图提高无人机定位的精确度；
2. 引入二维码视觉标签后，通过二维码的先验尺度在仅用单目相机的情况下估计出带有真实尺度的 SLAM 输出；
3. 结合二维码中坐标系和真实世界的坐标系的转化，可以获取无人机在堆体场景中的实际位姿。

2.2 二维码识别与使用方法研究

通过在堆体场景中添加二维码视觉标签，可以使得视觉 SLAM 算法在实际场景中得以应用，二维码和一般的自然关键点相比较，具备比周围环境更易检出的显著特点，如图 2.1 所示；且二维码本身是一个四边形的区域，可以凭借该特点估计出相机包含绝对尺度的位姿；并且每一个二维码通过解码都可以获得一个独一无二的 ID 序号，在 SLAM 重定位的过程中，可以避免相似区域的误检情况，在包含二维码的视觉 SLAM 中，希望尽可能多的检出场景中的二维码视觉标签，随后再通过编码公式对误检值进行剔除。本节将验证二维码得到检测和实现过程，以及如何通过二维码解算相机位姿。



(a) 二维码原始图 (b) 二维码梯度图

图 2.1 二维码原始图和梯度图

2.2.1 二维码检测与识别过程

二维码的检测和识别过程主要包括检测出二维码 4 个角点在图像中的位置，以及检出二维码对应 ID 序号，在检测 4 个角点的位置时，需要检测图像中的线段和构成二维码的四边形。

在线段检测阶段，首先会计算图片中每一个像素梯度强度的大小和方向，随后对计算得到的梯度进行聚类，对所有满足聚类条件的像素点进行合并，得到一组连续点，即检测出线段。

检测完线段后，进一步需要检测构成二维码边缘的四边形，针对上一步中获取到的所有线段，对线段进行分组，若满足，上一线段的末端点和下一线段的起始点之间的距离小于某一阈值，即可首尾按照逆时针进行连接，若所连接的线段数量达到 4 时，即认定生成的闭环可能为一个二维码的边缘四边形。

在检测二维码 ID 值之前，会先设定好一个包含所有二维码的字典，字典的大小即包含所有二维码的数量。检测出构成二维码的四边形后，需要对图像进行透视变化规范图像，随后通过设定阈值分离出二维码上的黑色位和白色位，通过位数情况判断出该二维码在字典中对应的特定 ID 值，识别结果如图 2.2 所示。

通过这样的方式检测和识别二维码具备非常好的鲁棒性，并且可以对错误值进行筛选。

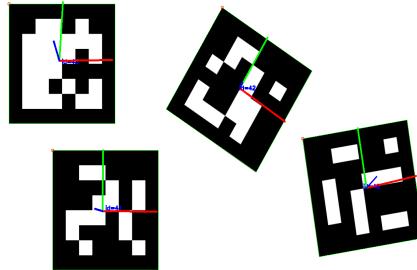


图 2.2 二维码识别结果示意图

2.2.2 利用二维码估计相机外参

对于单帧图像，在提取完二维码的四边形轮廓，4 个角点以及检测出对应的唯一 ID 值后，接下来可以估计相机位姿，包括平移向量和旋转向量，相机的位姿主要是通过 PnP 方法来求解。以二维码坐标系作为世界坐标系，二维码边长为 S ，则四个角点的坐标分别为 $A (-S/2, S/2, 0)$, $B (S/2, S/2, 0)$, $C (S/2, -S/2, 0)$, $D (-S/2, -S/2, 0)$ ，分别在图像中对应的像素点为 $a(u_a, v_a)$, $b(u_b, v_b)$, $c(u_c, v_c)$, $d(u_d, v_d)$ 。因为相机的内参 K 提前标定，则三维空间中的点和像素坐标中的点之间的转换关系可以表示为：

$$\begin{aligned} Z \begin{bmatrix} a \\ 1 \end{bmatrix} &= Z \begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} A \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} -s/2 \\ s/2 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (2.1)$$

通过公式 2.1，可以通过二维码的角点信息求解出相机的旋转矩阵和平移矩阵。

2.3 联合二维码的 SLAM 系统描述

2.3.1 引入二维码对 SLAM 优化研究

SLAM 是一个在导航的过程中同时进行建图的工作，在使用单目相机时，即使目前表现较好的 ORB-SLAM 或者 LSD-SLAM 等算法也很难达到预期想要的效果。第一个问题：所构成的地图的尺度未知，这样就造成无法在实际的导航任务中

发挥作用；其次，在纯旋转的移动过程中，会导致算法失效；第三个就是单目视觉 SLAM 往往需要场景中存在比较丰富的材质才能便于跟踪；最后当场景中的视点发生变化，或者场景重复度较高时，对于纯视觉 SLAM 的重定位，表现效果会很受到很大的限制。

除了用自然点参与视觉 SLAM 外，还可以使用人为设定的二维码来估计相机位姿，例如 SPM-SLAM 方法等，该方式可以较好的解决上述问题。因为二维码的放置不需要按照特定的规则，对于堆体场景，可以直接在堆体周围放置二维码标签，这样做就可以的得到一个由二维码构成的地图。但是这样做就需要在场景中布置足够多的二维码视觉标签，那么在大尺度的堆体场景中，这样做就会提高实验的难度，因为在相机移动的过程中，要保证每帧图像中至少有两张二维码出现。

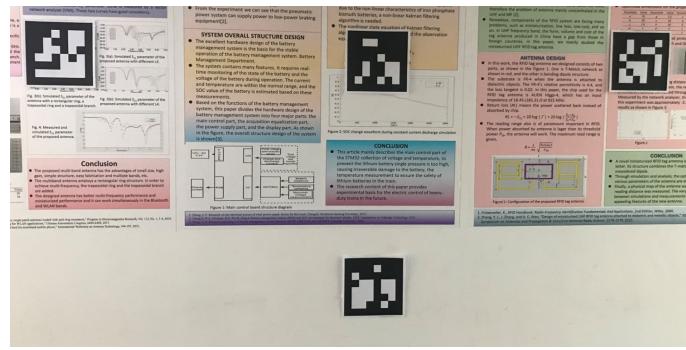
基于上述两种情况，本文提出了一种融合自然特征点和人工放置的二维码的 SLAM 方法来解决各自的问题和限制。对于实际堆体场景中，结合了自然点和二维码的 SLAM 方法可以只检测自然点，或者只检测二维码来进行位姿估计。因此对于大多数场景，都可以有更好的鲁棒性，此外，每一个二维码都有一个单独准确得到的 ID 值，在高度重复的堆体环境中，也能保证匹配的正确性，最后对于大尺度的堆体场景，可以结合自然点进行跟踪，二维码进行重定位来保证 SLAM 系统的长期稳定。

如图 2.3所示，对自然场景进行建图，图 2.3(a)为自然区域的原图，在该区域中，部分区域纹理丰富，其余部分则为贫纹理的白墙面，则在白墙面上人为布置二维码标志，图 2.3(b)为根据该区域重建出的 3D 地图，可以发现，在丰富纹理的区域，自然点和人为布置的二维码都可以检出，但在贫纹理区域则，则基本只有二维码被检出，因此在贫纹理的区域就可以依靠二维码发挥作用。

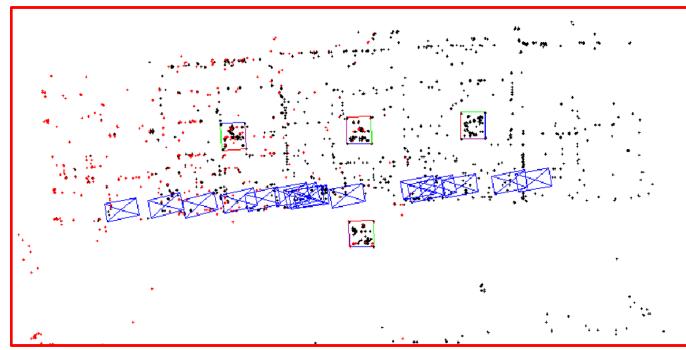
2.3.2 包含二维码的地图描述

在 SLAM 运行的过程中，可以生成一套地图系统，对于同一场景，该地图可以作为先验复用于后续的建图和定位，通过这样的方式可以添加对系统的约束，提高待估计量的精度。传统 SLAM 系统组成地图的元素主要包括关键点和关键帧，通过对关键点的描述符进行匹配，可以较好的的估计出相机的位姿。但关键点描述符的计算和匹配过程一般都比较耗时，而且对于堆体场景中，极易出现重复，容易出现出现匹配错误的情况，考虑到这一情况，本文在此基础上，又添加二维码信息作为构成地图的元素，来进一步优化地图以获取更加准确得到相机位姿和地图信息。

本文中地图的构成包括以下 3 个集合：关键点集合 \mathbf{p} ，关键帧集合 \mathbf{f} 和二维



(a) 自然场景示意图



(b) 自然场景 3D 地图

图 2.3 包含二维码的构建地图示意图

码 \mathbf{f} 集合，每个集合之间的数据元素耦合关联。其中关键点集合

$$\mathbf{p} = \{\mathbf{x}, \mathbf{v}, \hat{\mathbf{d}}\} \quad (2.2)$$

每个元素代表三维空间中的一个点，该点的描述包括在地图坐标系中的三维坐标 \mathbf{x} ，观测方向 \mathbf{v} 以及该点的描述符信息 $\hat{\mathbf{d}}$ ，考虑到 SLAM 系统的实时性，选择 BRIEF 描述符来加快匹配过程。关键帧集合

$$\mathbf{f} = \{t, \mathbf{T}, \delta\} \quad (2.3)$$

其中每一个关键帧包含一个外参矩阵矩阵 \mathbf{T} ，该外参矩阵是全局参考坐标系到相机参考坐标系的转化， δ 是相机的内参矩阵，该参数包括相机焦距，光学中心以及畸变参数，这些参数提前标定。二维码集合

$$\mathbf{m} = \{s, M, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4\} \quad (2.4)$$

其中的每一个二维码包括边长 s （需要保证场景中的所有二维码的为正方形且所有二维码的尺度完全一致），二维码由其自身坐标系到全局坐标系的转换矩阵 M ， $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$ 分别代表二维码四个角点在其自身坐标系下的坐标。通过对包含二维码的自然场景进行离线建图，可以得到如图 2.3(b)所示的地图，其中正方形代表检出的二维码，蓝色矩形代表检出关键帧，红色点代表检出关键点。

上述地图元素之间还具备一定耦合关系：

1. 对于任意一个关键点，除了自身的属性外，还包括被观测帧序列，即所有可以观测到该特征点的关键帧的集合，以及在这些帧中出现的像素坐标位置
2. 对于任意一个关键帧，还包与其关联的其他关键帧序列，以及在该关键帧中所有观测到的关键点和坐标。

通过这些约束可以使得估计结果有更好的鲁棒性。

2.3.3 联合二维码的 SLAM 过程描述

本文所提出的基于二维码的 SLAM 系统运行框图如图 2.4 所示，与一般 SLAM 系统相比较，本文主要提出了添加二维码视觉标签的 SLAM 系统。该系统一直对地图进行维护，有新的信息加入时，则会对更新地图。在对堆体场景运行 SLAM 时，该地图为空，所以需要对其进行初始化。

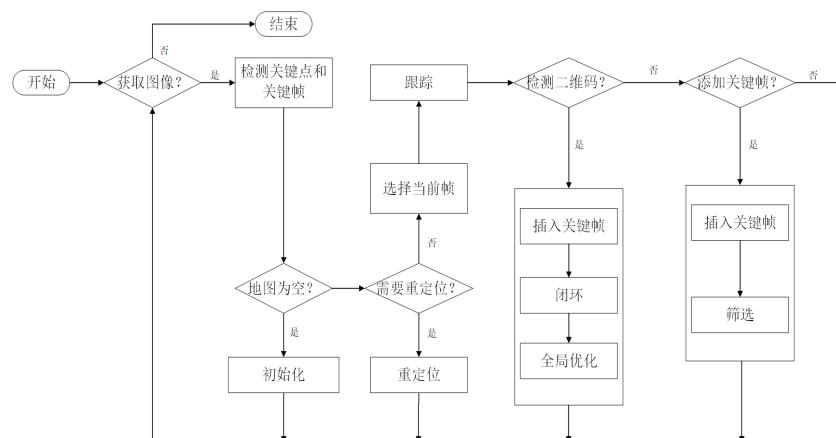


图 2.4 基于二维码的 SLAM 流程图

地图初始化：在该 SLAM 中对地图的初始化可以获取相对尺度信息和参考坐标系的信息，可以依靠图像序列中的关键帧或者二维码来完成，在一般情况下初始化的流程为：首先选择第一和第二帧 (f_0, f_1) 运行两套初始化方法，如果任意一套方法初始化成功，那么就直接进入跟踪环节；如果两套方案都初始化成功，则优先考虑使用二维码的初始化方式；如果没有成功，那么则固定第一帧，换取第三帧重复上述过程 ($f_2, f_3\dots$)。

对于基于二维码的初始化方法，帧之间需要足够的视差，随后得到帧和二维码在地图中的位姿信息，当在某一帧中检测到多个二维码标志时，则可以通过选择在该帧中最小化投影误差最小的二维码来估计初始化信息。使用二维码来进行初始化的一个优点就是，地图中的参数都具备正确的尺度，因此就可以直接用来为无人机的定位提供信息。

跟踪模式：如果在某一帧中估计出相机的位姿，则 SLAM 系统就会以前一

帧作为起始点来估计当前帧的相机位姿，相机的位姿估计是将当前帧中观测到的一组地图三维点和二维码角点进行重投影误差最小化的过程，因此在该过程中主要包括两个步骤：寻找地图的对应关系和通过最小化投影误差获取相机位姿。

在寻找地图点对应关系环节，首先，查找出前一帧中观察到的地图点与参考关键帧之间的匹配，由于该帧很可能再次出现在当前帧中，通过这些匹配可以解算出当前相机的初始估计；然后，通过将地图点投射到参考关键帧的相邻帧中来找到额外的匹配。对于每个地图点，首先需要检查其视角与相机方向之间的夹角是否小于阈值，若大于则丢弃该点，然后，检查相机与地图点之间的欧氏距离是否在关键点的尺度不变性区域内，随后，计算其投影，最后，将地图点的描述符与图像中搜索半径内的关键点进行比较。选择搜索半径中距离最小的两个关键点，且仅当描述符之间的距离比大于 0.8，以及最小的描述符距离低于最低阈值 τ_d ，则与匹配成功。一旦所有的匹配都被计算出来，重复的匹配就会被移除。最终，关键点结合检测到的二维码标记获得最终的精确位姿。对于相机的位姿估计，以

$$\Upsilon_p^f = \{(\mathbf{p}, \mathbf{f}, \mathbf{g})\} \quad (2.5)$$

和

$$\Upsilon_f^f = \{(\mathbf{m}, \mathbf{f}, \mathbf{c}_{m,f})\} \quad (2.6)$$

表示通过上述步骤得到的 f 帧中地图点和二维码标记的观测值，对于该帧的位姿 \mathbf{f}_T 可以通过最小化重投影误差来估计：

$$\mathbf{f}_T = \underset{T}{\operatorname{argmin}} \left(\mathbf{w}_p^f H \left(\Upsilon_p^f, T \right) + \mathbf{w}_m^f H \left(\Upsilon_m^f, T \right) \right) \quad (2.7)$$

其中

$$H \left(\Upsilon_m^f, T \right) = \sum_{(\mathbf{m}, \mathbf{f}, \mathbf{c}_{m,f}) \in \Upsilon_m^f} \sum_{i=1}^4 \left\| e \left(T \cdot \mathbf{m}_M, \mathbf{m}_{x^i}, \mathbf{f}_\delta, \mathbf{c}_{m,f}^i \right) \right\|_2^2 \quad (2.8)$$

是当前帧观测到的二维码标记的重投影误差之和，其中

$$H \left(\Upsilon_p^f, T \right) = \sum_{(\mathbf{p}, \mathbf{f}, \mathbf{g}) \in \Upsilon_p^f} h_\alpha \left(e(T, \mathbf{p}_x, \mathbf{f}_\delta, \mathbf{g}_u) \Omega_g e(T, \mathbf{p}_x, \mathbf{f}_\delta, \mathbf{g}_u)^\top \right) \quad (2.9)$$

是当前帧观测到的地图点的重投影误差之和。对于 Huber 损失函数

$$h_\alpha(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \alpha \\ \alpha(|a| - \frac{1}{2}\alpha) & \text{otherwise} \end{cases} \quad (2.10)$$

是为了降低优化过程中异常值的相对权重。

对于公式 2.7, 二维码视觉标签和地图点的权重分别由 \mathbf{w}_m^f 和 \mathbf{w}_p^f 决定, 因为同一帧中地图点的数量远远多于二维码的个数, 所以必须平衡两者之间的权重, 定义

$$\mathbf{w}_p^f = 1 - \mathbf{w}_m^f \quad (2.11)$$

和

$$\mathbf{w}_m^f = \frac{1}{2} \min \left(1, \frac{\mathbf{n}_f}{\tau_m} \right) \quad (2.12)$$

其中 \mathbf{n}_f 代表追踪过程中该帧有效的二维码标记的个数, τ_m 代表数量阈值, \mathbf{w}_m^f 的值将介于 $[0, \frac{1}{2}]$ 之间, 假设在追踪的过程中, 没有检测到有效的二维码, 则 \mathbf{n}_f 的值将为 0, 即该帧的位姿的结果将全部由地图点来估计。

插入关键帧: 关键帧只有在向系统添加新信息时才会被添加到地图中, 以便实现平稳可靠的跟踪。因此, 必须设定一些规则来确定是否需要将当前帧添加至关键帧的集合。本文提出以下关键帧的提取规则:

1. 如果当前帧至少有一个新的二维码标记出现 (该标记没有出现在原本的地图中), 则将该二维码和该帧添加到地图中。二维码是第一次在该帧中被发现时添加的, 因此也不能在帧中明确地估计该二维码的姿态。在这种情况下, 标记的位姿被设置为无效, 当观察标记的其他关键帧被添加到地图中时, 再对其进行估计。
2. 如果该帧包含至少一个二维码标记, 且该二维码标记在地图中具有无效位姿, 如规则 1 所描述, 则该二维码的位姿可以在该帧中进行估计, 则添加该帧并更新该二维码在地图中的位姿。
3. 如果帧包含至少一个二维码标志, 并且距当前帧最近的关键帧的距离大于一个阈值 τ_b , 该帧也会被添加进关键帧。
4. 如果当前帧中匹配的映射点数量低于在参考关键帧中检测到的映射点总数的百分比 τ_k , 该帧也会被添加进关键帧。

前三条规则的目的是考虑到当前关键帧中的可见二维码视觉标签来检查是否添加关键帧, 由于这些规则而没有被认为是关键帧的帧则通过最后一条规则来添加。

本文中的 SLAM 系统在添加地图点时会遵循一种强鲁棒的策略, 当一个新的关键帧被添加到地图时, 系统有机会添加新的地图点, 以及增加对现有地图点的支持。对于每个关键点, 搜索参考关键帧的相邻关键帧之间可能的对应, 由于关键帧的姿态是已知的, 所以采用了外极限制来减少误检。一旦在地图上添加了一个点, 将以一下原则验证该点是否会被加入地图基本思想是, 该点必须至少在接下来的三分之二帧中可见, 直到再向地图添加两个关键帧。

地图优化: 在添加新的关键帧或检测到闭环时, 需要更新地图, 以便加入新的信息, 并剔除不正确的地图点。需要优化的元素包括关键帧位姿 k_T 、地图点位置 p_x 和二维码位姿 m_M 。全局优化是一个缓慢耗时的过程, 在 SLAM 中一般会启用多线程的方式来管理地图。

筛选关键帧: 与地图点的保存策略相似, 关键帧剔除策略避免了关键帧的无限增长。但是, 相似处理关键帧的同时必须同时处理关键点和二维码。其主要流程是, 每个二维码和地图点都应至少被三个关键帧所观测到, 以便在进行优化过程时实现良好的三角化。对于每一个二维码标记 \mathbf{m} , 从关键帧集合中找到可以观测的关键帧, 并筛选出三个和该关键帧距离最近的关键帧, 这些关键帧不会被筛选。对于剩余的关键帧, 在不显著影响观察到的地图点的情况下会筛选掉, 即某个关键帧中的地图点如果存在 τ_c 的比例以上会被至少三个其他关键帧观测到, 那么该帧会被认为是冗余的, 而使用其他三个关键帧代替。

回环检测和校准: 在视觉 SLAM 中, 对于相机位姿的解算都是由上一帧来计算的, 那么这样就会造成误差的一直传递, 产生累积误差, 回环检测的提出就是通过建立当前帧和历史帧之间的关系以减少累积误差。词袋模型的提出可以快速简单地完成回环检测, 词袋模型使得每一帧都可以用单词来描述, 表达为:

$$F = 1 \cdot \omega_1 + 0 \cdot \omega_2 + 0 \cdot \omega_3 + \dots + 1 \cdot \omega_{n-1} + 0 \cdot \omega_n \quad (2.13)$$

每一帧 q 都可以由单词 w 和权重 η 构成:

$$A = \{(w_1, \eta_1), (w_2, \eta_2), \dots, (w_N, \eta_N)\} \triangleq v_A \quad (2.14)$$

可以通过计算 q, d 两帧之间字典的相似度判断是否构成回环:

$$\begin{aligned} s(v_q, v_d) &= 1 - \frac{1}{2} \left| \frac{v_q}{|v_q|} - \frac{v_d}{|v_d|} \right| \\ &= \frac{1}{2} \sum_{i|q_i \neq 0, d_i \neq 0} (|v_{qi}| + |v_{di}| - |v_{qi} - v_{di}|) \end{aligned} \quad (2.15)$$

以上是对关键帧的回环处理, 对于二维码的回环, 必须立即检出由二维码标记引起的闭环, 如果在跟踪之前不纠正回环漂移, 会导致相机的跟踪问题, 因为未纠正的帧位姿可能与二维码所计算出来的位姿差异较大。例如在某一个场景中, 系统从一个二维码开始初始化, 相机在场景中移动并回到最初的起始位置, 在这个过程中, 会不断地叠加漂移误差, 当相机第二次到达初始位置并再次观察二维码时, 估计的相机姿态与实际会较大, 当使用二维码标记或关键点检测到循环闭合时, 必须纠正漂移。

重定位: 如果在跟踪的过程中失败, 则必须进行重定位, 重定位也是首先考虑使用二维码来完成, 如果在某一帧中明确检测到二维码标记, 则利用该二维码

来重新计算位姿，随后，通过查找地图对应关系和重新估计相机姿态来优化位姿。如果当前帧位姿不能从二维码标记中准确地估计出来，就使用关键帧结合数据库进行查询，查询与当前帧相似的关键帧，并分析最佳关键帧。在回环检测出中，将当前帧的关键帧与所选关键帧进行匹配，并采用 RANSACPnP 法估计相机的位姿 T ，如果内置点的数量足够高，则认为重新定位成功，系统重新进入跟踪模式。

2.4 面向堆体无人机定位系统设计

2.4.1 无人机相机坐标变换数学描述

无人机上携带的相机可以看成三维空间中的刚体（保证了同一个向量在各个坐标系下的长度和夹角都不会发生变化），对于相机的平移和旋转，平移可以利用向量来表示，旋转则利用旋转矩阵，旋转向量，欧拉角，四元数等方式来描述，本文也将主要说明旋转过程的数学描述。

旋转矩阵：对于任意向量 \mathbf{p} ，在两个坐标系的描述为：

$$[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = [\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3] \begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} \quad (2.16)$$

等式两边同时左乘 $[\mathbf{e}_1^T, \mathbf{e}_2^T, \mathbf{e}_3^T]^T$ 可得

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} e_1^T e'_1 & e_1^T e'_2 & e_1^T e'_3 \\ e_2^T e'_1 & e_2^T e'_2 & e_2^T e'_3 \\ e_3^T e'_1 & e_3^T e'_2 & e_3^T e'_3 \end{bmatrix} \begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} = \mathbf{R}\mathbf{a}' \quad (2.17)$$

对于旋转矩阵 \mathbf{R} ，可描述旋转过程，并且 \mathbf{R} 可以定义为特殊正交群：

$$SO(n) = \{\mathbf{R} \in R^{n \times n} | \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\} \quad (2.18)$$

旋转向量：旋转矩阵在描述旋转时存在参数冗余和自身约束的问题，因此提出一个更加紧凑的数学描述 θ_n ，旋转向量方向为旋转轴 \mathbf{n} ，大小为旋转角 θ 。旋转向量和旋转矩阵的转换可以用罗德里格斯公式表示：

$$\mathbf{R} = \cos \theta \mathbf{I} + (1 - \cos \theta) \mathbf{n} \mathbf{n}^T + \sin \theta \mathbf{n}^\wedge \quad (2.19)$$

反之则有

$$\text{tr}(\mathbf{R}) = \cos \theta \text{tr}(\mathbf{I}) + (1 - \cos \theta) \text{tr}(\mathbf{n} \mathbf{n}^T) + \sin \theta \text{tr}(\mathbf{n}^\wedge) = 1 + 2 \cos \theta \quad (2.20)$$

欧拉角：将旋转运动分解成分别绕三个坐标轴的旋转 $(r, p, y)^T$ 来表示的，三个旋转的总和即为总的旋转，表述最为直观。

四元数：由于欧拉角和旋转向量存在奇异性，且不存在不带奇异性的三维向量描述方式，因此提出四元数这种紧凑且没有奇异性的数学描述。四元数可以描述为：

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} = \{s, \mathbf{v}\} \quad (2.21)$$

可以用单位四元数来描述空间中的任意旋转，假设旋转的描述为 θ_n ，则：

$$\mathbf{q} = \left[\cos \frac{\theta}{2}, n_x \sin \frac{\theta}{2}, n_y \sin \frac{\theta}{2}, n_z \sin \frac{\theta}{2} \right]^T \quad (2.22)$$

旋转描述为 \mathbf{R} 时，则：

$$\mathbf{R} = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix} \quad (2.23)$$

反之则有：

$$q_0 = \frac{\sqrt{\text{tr}(\mathbf{R}) + 1}}{2}, q_1 = \frac{m_{23} - m_{32}}{4q_0}, q_2 = \frac{m_{31} - m_{13}}{4q_0}, q_3 = \frac{m_{12} - m_{21}}{4q_0} \quad (2.24)$$

通过以上方式都可以描述刚性运动中的旋转过程 \mathbf{R} ，结合物体的平移 \mathbf{t} ，则变化后的坐标为：

$$\mathbf{a}' = \mathbf{R}\mathbf{a} + \mathbf{t} \quad (2.25)$$

在实际的使用中，由于以上等式非线性，可以引入齐次坐标 \mathbf{T} ，可以将上式改成：

$$\begin{bmatrix} \mathbf{a}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}' \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{a}' \\ 1 \end{bmatrix} \quad (2.26)$$

对于变换矩阵 \mathbf{T} ，可以定义为特殊欧氏群：

$$SE(3) = \left\{ T = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in R^{4 \times 4} | \mathbf{R} \in SO(3), \mathbf{t} \in R^3 \right\} \quad (2.27)$$

对于针孔相机模型，如图 2.5 所示，（其中 f 为相机焦距）存在 4 个坐标系分别为：世界坐标系，相机坐标系，图像坐标系和像素坐标系，对于真实世界中的空间点 P_w (X_w, Y_w, Z_w)，其对应的相机坐标系坐标为 P_c (C_w, C_w, C_w)，对应的图像坐标系为 P' (X', Y')，对应的像素坐标系为 p (u, v)。

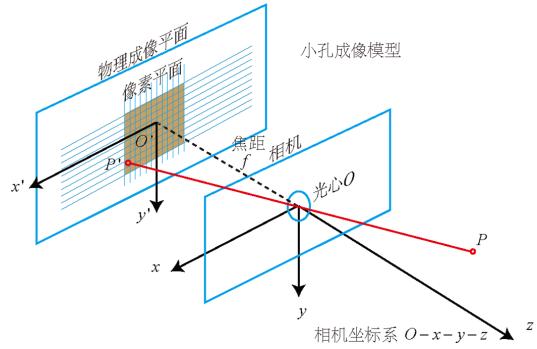


图 2.5 针孔相机模型

对于世界坐标系到相机坐标系之间的转化:

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t \quad (2.28)$$

对于相机坐标系到图像坐标系之间的转化, 可以利用如图 2.6(a)所示的相似三角形来求解:

$$\begin{aligned} \frac{Z_c}{f} &= \frac{X_c}{X'} = \frac{Y_c}{Y'} \\ \begin{cases} X' = f \frac{X_c}{Z_c} \\ Y' = f \frac{Y_c}{Z_c} \end{cases} \end{aligned} \quad (2.29)$$

对于图像坐标系到像素坐标系的转化, 如图 2.6(b)所示:

$$\begin{aligned} \begin{cases} u = \frac{X'}{d_x} + u_o \\ v = \frac{Y'}{d_y} + v_o \end{cases} \\ \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & 0 & u_o \\ 0 & \frac{1}{d_y} & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} \end{aligned} \quad (2.30)$$

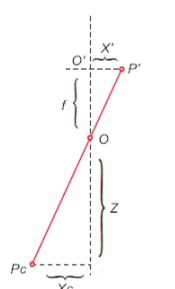
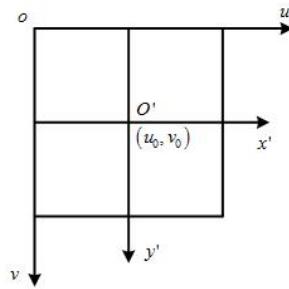
(a) 相机坐标系
到图像坐标系
转换(b) 图像坐标系到像素坐
标系转换

图 2.6 坐标系转化示意图

其中 d_x, d_y 分别表示沿着 x, y 轴的实际物理尺寸, (u_o, v_o) 表示光心对应到像素坐标系得到坐标. 由公式 2.29 和 2.30 可以转化得到相机坐标系到像素坐标系的关系:

$$Zc \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{d_x} & 0 & u_0 \\ 0 & \frac{f}{d_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2.31)$$

由公式 2.28 和 2.31 得到世界坐标系到像素坐标系的转化关系为:

$$Zc \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K(RPw + t) = K \left(R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + t \right) \quad (2.32)$$

其中 K 表示相机内参矩阵, R, t 表示相机外参。

2.4.2 获取真实世界坐标系方法

在 SLAM 系统中所描述的定位信息都是基于地图坐标系或者相机坐标系, 但是在实际的应用中, 就需要将所有的定位信息全部转换至真实世界坐标系下, 才能为无人机后续的移动提供控制方法。传统的 SLAM 方法, 只能够得获取相机在地图坐标系下的位姿, 本文结合二维码自带坐标系的特性, 可以经过一定的转换获取相机在真实世界坐标系下的位姿, 为后续控制提供真实世界下的位姿信息。

首先, 声明所有可以得到的坐标信息, 二维码在 SLAM 地图坐标系下的位姿 T_{marker}^c , 相机在 SLAM 地图坐标系下得到位姿 T_{camera}^c , 二维码在真实世界坐标系的位姿 T_{marker}^w 。

经过坐标变化, 可以的得到相机在二维码下的位姿

$$T_{camera}^m = \left[T_{marker}^c \right]^{-1} T_{camera}^c \quad (2.33)$$

真实世界坐标系在二维码坐标系下的位姿

$$T_{world}^m = \left[T_{marker}^w \right]^{-1} \quad (2.34)$$

那么相机在真实世界坐标系下的位姿

$$\begin{aligned} T_{camera}^w &= \left[T_{world}^m \right]^{-1} T_{camera}^m \\ &= T_{marker}^w \left[T_{marker}^c \right]^{-1} T_{camera}^c \end{aligned} \quad (2.35)$$

其中, 二维码在真实世界坐标系的位姿可以按照实际场景进行设置, 为了计算方便, 直接设定第一个二维码的中心点为世界坐标系三维坐标的中心点, 且 x 轴朝

右, y 轴朝上, z 轴垂直于二维码平面向外, 如图 2.7 所示, 那么第一个二维码在世界坐标系下的坐标为 $(0,0,0)$, 其余二维码则可以按照真实尺度获取。

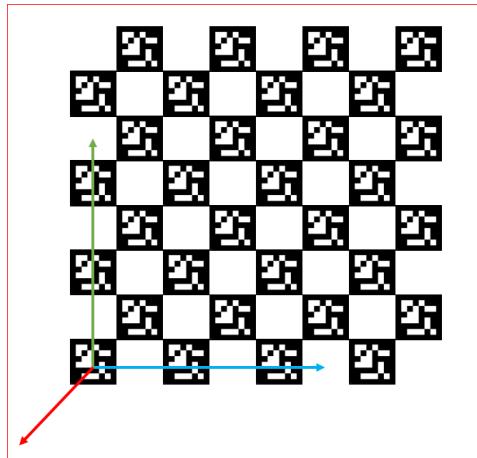


图 2.7 二维码布置示意图

在 SLAM 实际运行前, 应该保证二维码的实际布置坐标和设定的二维码在实际世界坐标系下的坐标一致, 避免 SLAM 过程中位姿的抖动问题, 先验地图可以以配置文件的形式记录, 这样可以针对不同的场景和不同的二维码布置情况, 直接修改配置文件即可。在 SLAM 运行过程中, 应该尽可能保证每一帧都可以看到至少一个二维码, 对于观测不到二维码的图像帧, SLAM 算法将利用特征点匹配进行定位, 当观测的图像帧中有多个二维码时, 那么就需要考虑到多个二维码的协同影响, 本文提出以下几种处理形式:

1. 提前选择某一个参考二维码, 无论相机观测到哪一个二维码, 都根据观测到的二维码和参考二维码之间的位姿关系, 求解相机在世界坐标系下的位姿。
2. 对于观测到的多个二维码标记, 根据上一帧计算出来的位姿, 可以计算出当前帧距离最近的一个二维码, 以该二维码作为参考, 计算出此时相机在世界坐标系下的位姿。
3. 对于观测到的多个二维码标记, 分别结合各个二维码计算出此时相机在世界坐标系下的位姿, 对于所有位姿, 求取平均值即可。

以上三种方法都可以解算出相机在真实世界坐标系下的位姿, 其中方法 1 的计算过程最为简略, 并且全程只利用一个参考二维码可以避免在频繁更换参考二维码的过程中造成的位姿抖动问题, 但是会存在线性放大相机位姿误差的情况, 该方法难以对误差进行优化。方法 2 会应用到历史位姿信息, 可以有效降低相机位姿估计的误差, 但会频繁更换参考二维码, 这样也会造成相机位姿估计过程的抖动问题。方法 3 和方法 2 类似, 在处理误差时会结合多个参考二维码的信息, 使得估计位姿的系统鲁棒性提升。针对不同的场景, 需要结合具体的方法, 对于二维码远距离线性摆放的场景, 应该避免使用方法 1, 优先考虑方法 2 和方

法 3；对于实时性要求较高的场景，则优先考虑方法 1。

2.4.3 无人机自主定位设计

在根据以上章节，可以确定出在真实世界下相机的位姿结果，在具体的工程实践中，除了结合二维码的 SLAM 算法的研究，还需要考虑到先验地图的设定，数据的传输，相机的设定等问题，具体流程如图2.8所示。

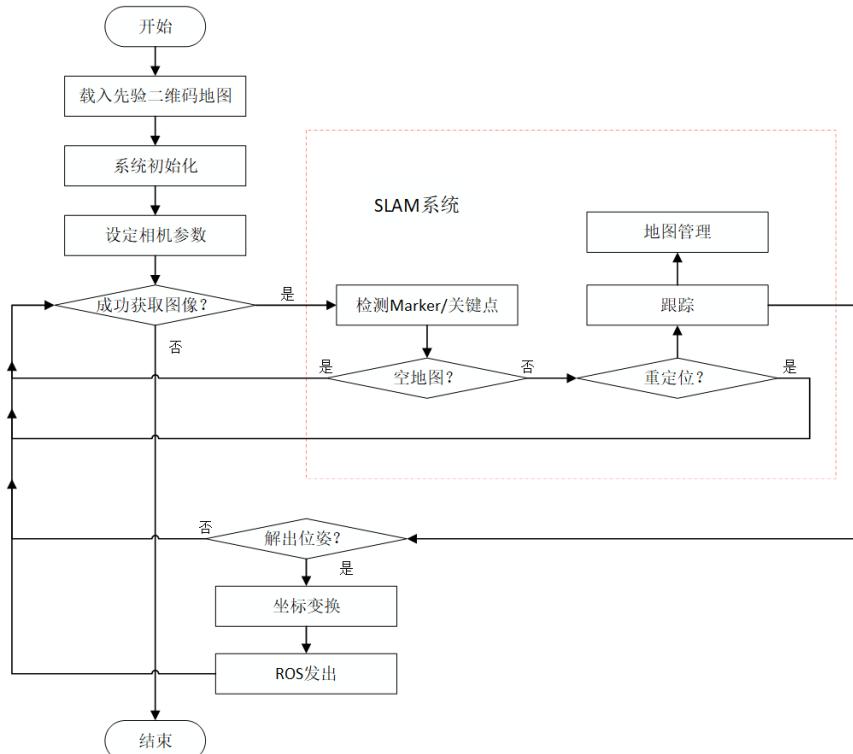


图 2.8 获取真实位姿完整流程图

地图管理: 在无人估计位姿时，如果有先验地图的加入，则可以添加约束进一步优化位姿结果。在载入地图的过程中，如果该场景是第一次测量，没有先验的地图信息，则创建一个空的 globalmap 来存储地图信息，并将 MapFlag 字段设定为 NotFirstMap；反之，则直接载入历史先验地图。在整个定位流程结束后，将更新后的地图保存在 globalmap 中，代码如 2.1 所示。

Listing 2.1 地图管理 C++ 实现

```

1 //载入地图
2 if (Map_Flag == NotFirst_Map)
3 {
4     globalmap->readFromFile(map_path);
5 }
6 else
7 {
8     globalmap->readFromFile(map_path);
9     Map_Flag == NotFirst_Map;

```

```

10         }
11 //保存地图
12 if (Map_Flag == Save_Map)
13 { globalmap->saveToFile(map_path); }

```

载入先验二维码地图:首先准备先验地图配置文件,包括二维码的真实尺度,所有二维码的ID值,以及每一个二维码特定的id值和在真实世界坐标系下的位姿(\mathbf{R}, \mathbf{t})。如代码2.2所示,需要依次遍历每个二维码的信息,最终将所有数据都存储在markermmap中。

Listing 2.2 载入先验二维码地图 C++ 实现

```

1 先验地图配置文件:
2 %YAML:1.0
3 markersize: 0.73
4 markers_id: [0,1]
5 id_0: !!opencv-matrix
6 rows: 4
7 cols: 4
8 dt: d
9 data: [1.0, 0.0, 0.0, 0.0,
10        0.0, 1.0, 0.0, 0.0,
11        0.0, 0.0, 1.0, 0.0,
12        0.0, 0.0, 0.0, 1.0]
13 id_1: !!opencv-matrix
14 rows: 4
15 cols: 4
16 dt: d
17 data: [1.0, 0.0, 0.0, 4.0,
18        0.0, 1.0, 0.0, 0.0,
19        0.0, 0.0, 1.0, 0.0,
20        0.0, 0.0, 0.0, 1.0]
21
22 先验地图载入代码:
23 void loadMarkermmap(string path){
24     cv::FileStorage fs(path, cv::FileStorage::READ);
25     fs["markersize"] >> markersize;
26     fs["markers_id"] >> markers_id;
27     for (std::size_t i = 0; i < markers_id.size(); i++) {
28         cv::Mat cvmarker ;
29         Eigen::Matrix<double, 4, 4> marker;
30         fs["id_" + to_string(markers_id[i])] >> cvmarker;
31         cv2eigen(cvmarker, marker);
32         markermmap[markers_id[i]] = marker;
33     }
34 }

```

设定 SLAM 参数:对于结合二维码的SLAM系统,需要对系统中的参数进行设定,主要包括SLAM的运行模式(SLAM模式,重定位模式),特征点的提取方式(orb、brief、SIFT等),二维码的检测类别(ARUCO_MIP_25h7、ARUCO_MIP_36h12等),关键帧筛选阈值,线程的数量,是否复用地图等等,这些SLAM相关的参

数都需要在运行前设定, 具体内容如代码 2.3 所示。

Listing 2.3 slam 启动和参数设置 C++ 实现

```

1 void setSlam(int argc) {
2     UcoSlamParams.runSequential=false;
3     UcoSlamParams.detectMarkers=true;
4     UcoSlamParams.aruco_markerSize=markersize;
5     UcoSlamParams.aruco_Dictionary="ARUCO_MIP_36h12";
6     UcoSlamParams.nthreads_feature_detector =1;
7 }
```

设定相机参数: 相机的选择对基于纯视觉的无人机定位任务也很重要, 作为 SLAM 系统的输入, 必须选择一台成像质量高, 稳定性强的相机。在实际工程中选择 PointGrey (CM3-U3-31S4M-CS) 工业相机来接受视觉信息, 外形如图 2.9 所示, 具体参数如表 2.1 所示。



图 2.9 PointGrey 工业相机示意图

表 2.1 pointgrey 相机参数表

参数	数值	参数	数值
Frame Rate	30 FPS	Megapixels	1.3 MP
Lens Mount	CS-mount	Readout Method	Global shutter
Sensor Format	1/3"	Chroma	Color
Sensor Type	CCD	Part Number	CM3-U3-13S2C-CS
Pixel Size	3.75 µm	Sensor Name	Sony ICX445
Resolution	1288 × 964	Required Accessories	Lens, Cable, Host Adapter (USB 3.1 Gen 1), Tripod Mount Adapter (ACC-01-0003)

相机在使用过程中需要设定的参数一般包括相机内参, 以及曝光, 快门时间和增益等, 对于相机内参需要提前进行标定, 后面三个参数需要根据实际场景调试后确定出最佳数值, 代码如 2.4 所示

Listing 2.4 相机参数设定 C++ 实现

```

1 int SHUTTER=7 ; int BRIGHTNESS=0.5; float GAIN=20.0;
2 void setCamera(){
3     for (int frameNumber = 0; ;frameNumber++) {
4         char key = static_cast<char>(cv::waitKey(1));
5         if (key == w || key == W) {break;}
6         if (key == u || key == U){SHUTTER = SHUTTER+1;}
7         if (key == j || key == J){SHUTTER = SHUTTER-1;}
8         if (key == i || key == I){BRIGHTNESS = BRIGHTNESS+1;}
9         if (key == k || key == K){BRIGHTNESS = BRIGHTNESS-1;}
10        if (key == o || key == O){GAIN = GAIN+1;}
11        if (key == l || key == L){GAIN = GAIN-1;}
12        mypointgrey->pointgreyInit(SHUTTER,BRIGHTNESS,GAIN);
13        mypointgrey->getData();
14    }
15 }
```

设定数据传输系统：在工程实践中，一般都采用 ROS（机器人操作系统）来对数据进行传输，其用于编写机器人软件程序的一种具备高灵活轻度的软件架构，可以使得数据在多个进程中进行传输，本文主要涉及 SLAM 进程和无人机飞控进程，这两个进程分别对同一个节点进行数据得到发布和订阅以完成数据传输，代码如 2.5 所示。

Listing 2.5 ROS 初始化设定和数据传输 C++ 实现

```

1 void publishPose(ros::Publisher odom_pub){
2     Eigen::Quaterniond q = Eigen::Quaterniond(t.block<3,3>(0,0));
3     nav_msgs::Odometry odometry;
4     ros::Time current_time= ros::Time::now();
5     odometry.header.stamp = current_time;
6     odometry.header.frame_id = "world";
7     odometry.pose.pose.position.x = t(0,3);
8     odometry.pose.pose.position.y = t(1,3);
9     odometry.pose.pose.position.z = t(2,3);
10    odometry.pose.pose.orientation.x = q.x();
11    odometry.pose.pose.orientation.y = q.y();
12    odometry.pose.pose.orientation.z = q.z();
13    odometry.pose.pose.orientation.w = q.w();
14    odom_pub.publish(odometry);
15    ros::spinOnce();
16 }
17
18 for (int frameNumber = 0; keyPressed!=27;frameNumber++) {
19     ...\\dots
20     publishPose(odom_pub);
21 }
22
23 ros::init(argc, argv, "talker");
24 ros::NodeHandle n;
25 ros::Publisher odom_pub =
26             n.advertise<nav_msgs::Odometry>("odom", 1000);
27 ros::Rate loop_rate(10);
```

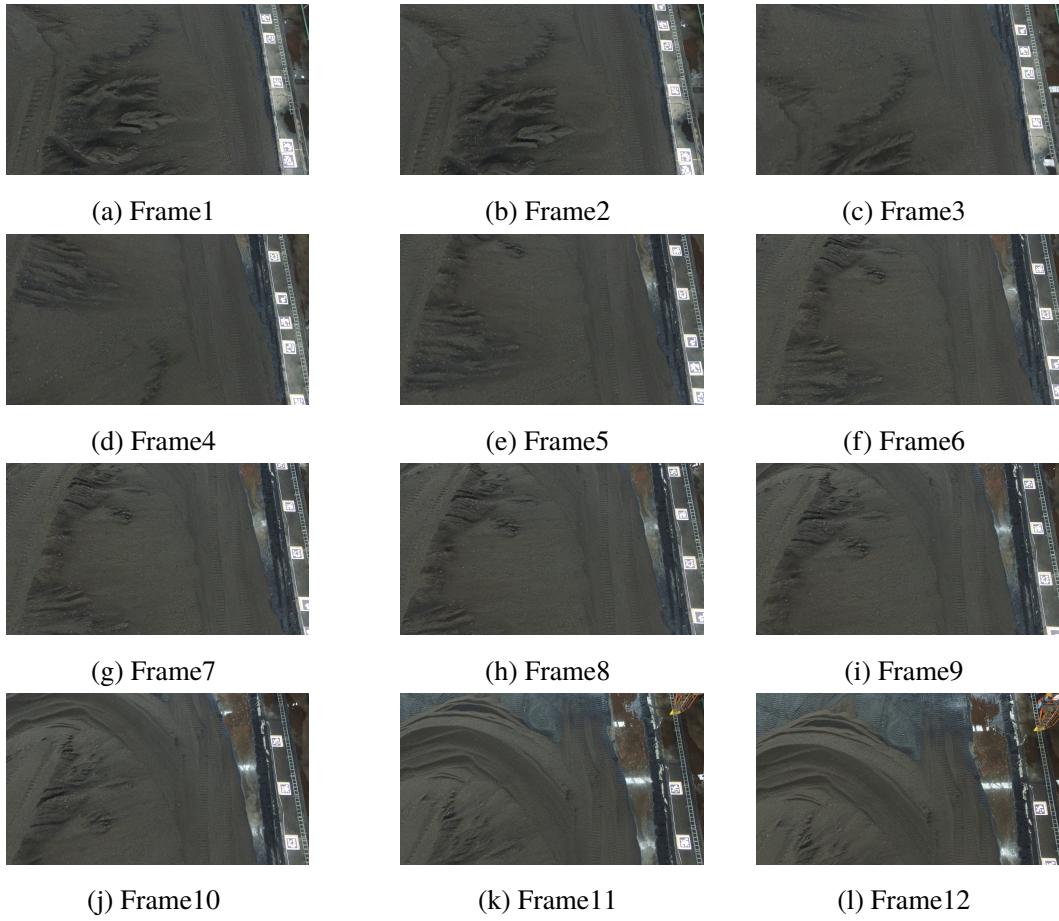


图 2.10 无人机采集堆体图像序列

根据以上流程，可以针对堆体，在不适用 GPS 的条件下进行无人机的自主定位和控制，无人机所拍摄的堆体场景如图 2.10 所示。二维码视觉标签位于图像左侧，连续放置，为无人机的自主定位提供世界坐标系得到真实位姿。

2.5 本章小结

本章阐述了二维码的检测和识别原理，通过线条和角点的检测，判断区域中二维码的坐标位置以及每一个二维码对应的 ID 值，推导了通过二维码解算相机位姿的流程。

然后重点研究二维码对视觉 SLAM 带来的优化作用以及 SLAM 地图的描述情况，包括关键点，关键帧以及二维码的属性等。说明引入二维码对视觉 SLAM 各个环节的影响，包括地图初始化、跟踪情况、关键帧的插入筛选、地图优化、回环检测和重定位等。解决了针对堆体场景下，单目视觉 SLAM 在无人机自主定位工作中，无法确定尺度和不能确定坐标系的问题。

最后本文针对具体堆体场景，进行软件设计，提出了一种为无人机在封闭环境中依靠纯视觉进行自主定位的方案，包括地图管理，图像处理，无人机位姿信

息传递等模块，模块之间相互协作，为无人机提供准确的位姿数据，供其后续进行循迹飞行。

在无人机飞行的过程中，会携带图像采集装置，不断采集堆体的图像信息，为后续的堆体三维重建以及体积测量提供支持。

第3章 面向堆体的三维重建方法改进

3.1 引言

在计算机视觉中，三维重建是指基于对环境或者物体的一系列不同视角的照片，通过特定流程的处理，获得环境或者物体的三维模型，模型表达方法众多，常见形式包括点云格式，网格格式，深度地图模式等。三维重建的整个操作流程简易，只需要将采集到的 2D 图像，或者截取视频中的图像作为输入，传递给三维重建系统即可，通过一系列的处理即可得到所拍摄场景的三维维点云模型，每一个元素具备三维位置信息和 RGB 颜色信息。三维重建的应用广泛，在自动驾驶，VR，AR 等众多领域都有涉及，在未来也会进一步的和计算机视觉中的各种识别方法相互结合，更好的服务于现代科学。

基于视觉的三维重建流程已经相对成熟，在一般情况下，也能得到一个较为满意的结果，但是传统三维重建方法还是存在很多的问题，三维重建大多数都应用于离线环境，一旦输入数据量较大时，流程本身会十分耗时，且对处理设备的要求也会提升；此外三维重建对场景的要求也比较高，在光照条件不良，或者场景重复度较高的环境中，三维重建最终输出的点云会存在无法闭合，噪音点对以及场景歧义等现象，这些都限制了三维重建的应用范围。

基于以上问题，本章将提出一种结合 SLAM 结果的优化三维重建方法，以解决三维重建耗时和精度不高的问题。三维重建的输入为无序的图像序列，因此在匹配和解算位姿时都会耗费较大的算力和内存，并且匹配结果因为确实时序信息无法保证匹配的准确性，从而导致后续的位姿解算也会发生错误。但在 SLAM 系统中，由于考虑到图像的时序信息，在匹配时不需要进行完全匹配，可以从相邻帧或者回环中检测出匹配对从而节省算力，此外更加精确的匹配关系能得到更加准确的位姿信息，以为后续三维重建中的稀疏建模和稠密建模获取更高的精度和鲁棒性。

3.2 三维重建方法研究

3.2.1 三维重建流程概述

三维重建是指从三维图像中复原三维场景或者物体的过程，整个流程的输入为无序的图片即可，输出可以得到三维重建后的稀疏点云和稠密点，大致流程

如图3.1所示。

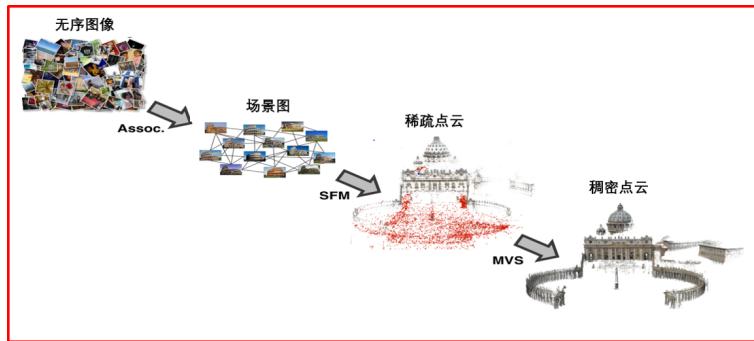


图3.1 三维重建流程图

对于以上流程进一步进行细化，整个三维重建的过程可以划分为以下几个主要的步骤：

1. 2D 图像采集：多角度拍摄或者从视频中提取到一组图像序列，将图像序列作为整个系统的输入；
2. 特征点提取和匹配：根据拍摄到的图像，提取每张图像之间的特征点，并进行特征点的匹配；
3. 稀疏点云：根据匹配结果估计特征点的深度，提取出稀疏点云，并估计相机的位姿和参数；
4. 稠密点云：根据优化后的相机参数和匹配结果，获得稠密点云；
5. 纹理映射：根据以上点重建物体表面，进行纹理映射。

三维重建的一般步骤可以简化为如图3.2所示。

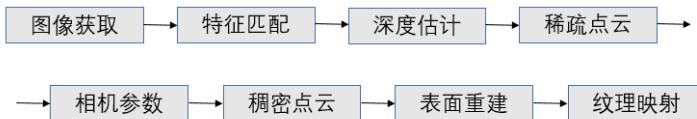


图3.2 三维重建具体实现步骤图

三维重建主要分为多个步骤，目前很多开源的系统都可以完成其中的部分环节，对于完整的三维重建流程还需要多个系统相互连接实现，表3.1是当前三维重建系统的简要对比，其中绿色框图代表该重建系统包括该流程，灰色则不包括。

表 3.1 常见三维重建系统对比表

系统名称	稀疏点云	稠密点云	重建表面	纹理映射
Bundler	是	否	否	否
CMVS	否	是	否	否
Colmap	是	是	是	否
Meshlab	否	否	是	是
MVE	是	是	是	否
MVS-texturing	是	是	是	否
openMVG	是	是	否	否
openMVS	否	否	是	是
Theia	是	否	否	否
VisualFSM	是	是	否	否

3.2.2 三维重建详细过程解析

3.2.2.1 图像获取

目前三维重建仅需要输入无序图片，在构图的过程中，可以极大地降低操作的复杂度，并且对于相机的内参和外参也无需提前提供给整个三维重建的系统，在重建过程中，这些参数都可以计算得到。对于输入的图像，可以通过随着时间流单帧拍摄的方式获取或者通过截取视频流的方式得到，且图像之间不能仅有纯旋转，这样无法估计深度。对于堆体场景的图形采集，如图 3.3 所示，需要注意保证每连续两帧图像之间尽可能保证 30% 的重叠区域，相邻两帧之间的旋转角度为 30 度到 45 度之间，且物体中的每一个点至少能被三帧图像观测到。

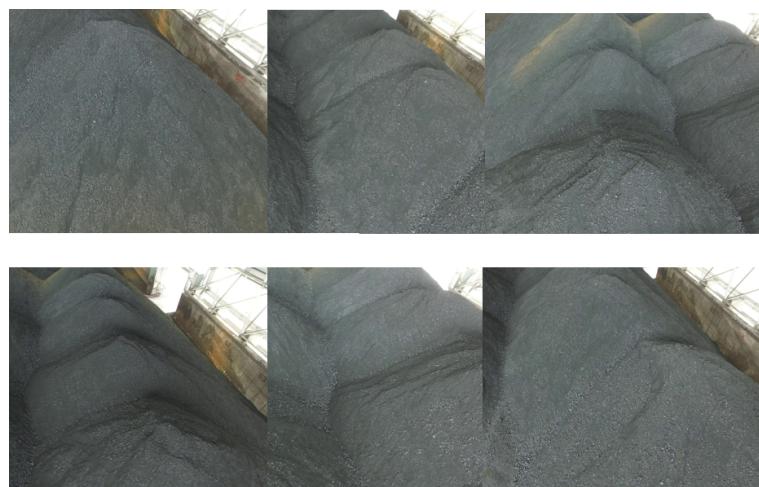


图 3.3 堆体图像采集示意图

3.2.2.2 特征点提取与检测

特征是图像信息的另外一种数字表现形式，良好的特征应该是不受光线，噪音，几何变形影响的。目前，在三维重建领域中图像的特征匹配就是以特征点为基础而进行的，所以，如何定义和找出一幅图像中的特征点就非常重要，常见的特定点检测和匹配主要包括 FAST, SIFT, ORB, Harris 角点等。特征提取的目的是为了后续能够尽可能准确、稳定地估计出相机的运动，且特征点应该具备可重复性，高效率，可区别性以及本地性的特点，本文列举了在图像处理领域几种常用的特征检测和提取的方法。

Harris 角点：当从不同的方向去移动一个视觉窗口，假设该区域内的灰度发生了很大的变化，则认定存在角点。对于图像 $I(x,y)$ ，当在点 (x,y) 处平移 (x,y) 后的对应窗口的像素点灰度变化描述为：

$$c(x,y; \Delta x, \Delta y) = \sum_{(u,v) \in W(x,y)} (I(u,v) - I(u + \Delta x, v + \Delta y))^2 \quad (3.1)$$

结合泰勒分解公式，上式可以化简为：

$$c(x,y; \Delta x, \Delta y) \approx \sum_w (I_x(u,v)\Delta x + I_y(u,v)\Delta y)^2 = [\Delta x, \Delta y]M(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (3.2)$$

其中

$$\begin{aligned} M(x,y) &= \sum_w \begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix} \\ &= \begin{bmatrix} \sum_w I_x(x,y)^2 & \sum_w I_x(x,y)I_y(x,y) \\ \sum_w I_x(x,y)I_y(x,y) & \sum_w I_y(x,y)^2 \end{bmatrix} \\ &= \begin{bmatrix} A & C \\ C & B \end{bmatrix} \end{aligned} \quad (3.3)$$

公式 3.1 即可转化为

$$c(x,y; \Delta x, \Delta y) \approx A\Delta x^2 + 2C\Delta x\Delta y + B\Delta y^2 \quad (3.4)$$

又提出角度响应值 R 来判断该点是否为角点：

$$R = \det M - \alpha(\text{trace } M)^2 \quad (3.5)$$

通过以上公式可以看出，Harris 角点具备以下特点：

1. 对亮度和对比度的变化不敏感：因为微分运算对图像密度的变化不敏感，即亮度或者对比度的变化对 Harris 的检出影响较小；

2. 具有旋转不变性: Harris 角点检测算子的本质可以表示为一个椭圆, 但椭圆旋转时, 并不会影响 R 值的大小;
3. 不具有尺度不变性。

FAST 特征点: 对图像中的中的一个像素 p , 假设其亮度值为 I_p , 阈值为 t , 如图所示, 取以其为中心, 半径为是三个像素的圆, 圆上共有 16 个像素点, 假设这 16 个点中有连续 N 个点都比 $I_p + t$ 大或者比 $I_p - t$ 小, 则认定其为 Fast 角点, 如图 3.4所示, 为了加快检测过程, 会直接检测第 1,5,9,13 这 4 个点, 如果有三个满足要求, 也会认为是角点。

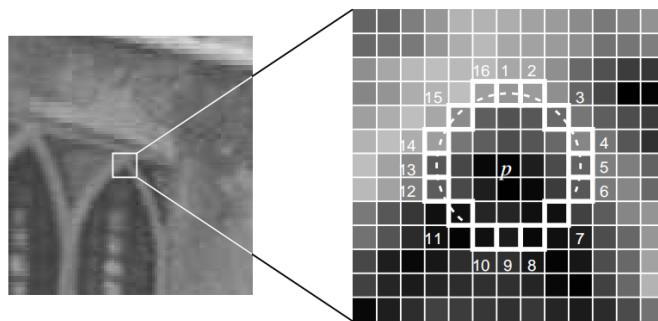


图 3.4 FAST 特征点

ORB 特征点: 结合了 FAST 特征点的检测和 BRIEF 描述子, 其中 RIEF 特征描述符的计算过程: 首先平滑图像, 在某个像素点的周围选择一个领域, 根据特定的点对选择方法挑选出 n_d 个点对, 比较每个点对之间亮度值的大小, 即可得到一个长度为 n_d 的二进制串。由于 FAST 特征点不具有方向, ORB 在此基础上进行了改良, 即利用灰度质心法求解出灰度质心之间的偏移方向, 首先定义特征点 p 的邻域像素的矩:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (3.6)$$

图像的质心为:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3.7)$$

那么偏移方向可以定义为 FAST 的特征点方向:

$$\theta = \arctan(m_{01}, m_{10}) \quad (3.8)$$

以上说明了图像处理中常用的特征点提取方式, 如图 3.5(a)和 3.5(b)先分别对两张堆体图像进行特征提取。接下来将以此为基础说明特征点的匹配过程: 在获取到每张图像上的特征点后, 需要对图像之间建立匹配关系, 常用的方式可以采用计算欧式距离的办法:

1. 完全匹配，对所有的特征点都进行穷举，计算其对应距离。
2. 邻近搜索，建立 KD 树，缩小搜索范围，能提高效率，但也有可能不是最优，所以邻域取值是关键，越大越准确，越大计算量越大。

如图 3.5(c)所示，两帧之间大多数特征点都可以正确匹配，但依然存在存在部分匹配是错误的，在本文中，选择了 RANSAC（随机抽样一致性）的方式来剔除错误的匹配对，以更加准确的估计相机位姿，RANSAC 是指可以从一组包括局外点（错误匹配）的观测数据中，通过迭代的方式估计数学模式中的参数，通过 RANSAC 处理后的匹配结果如图 3.5(d)所示，误检的匹配已经被明显的降低。

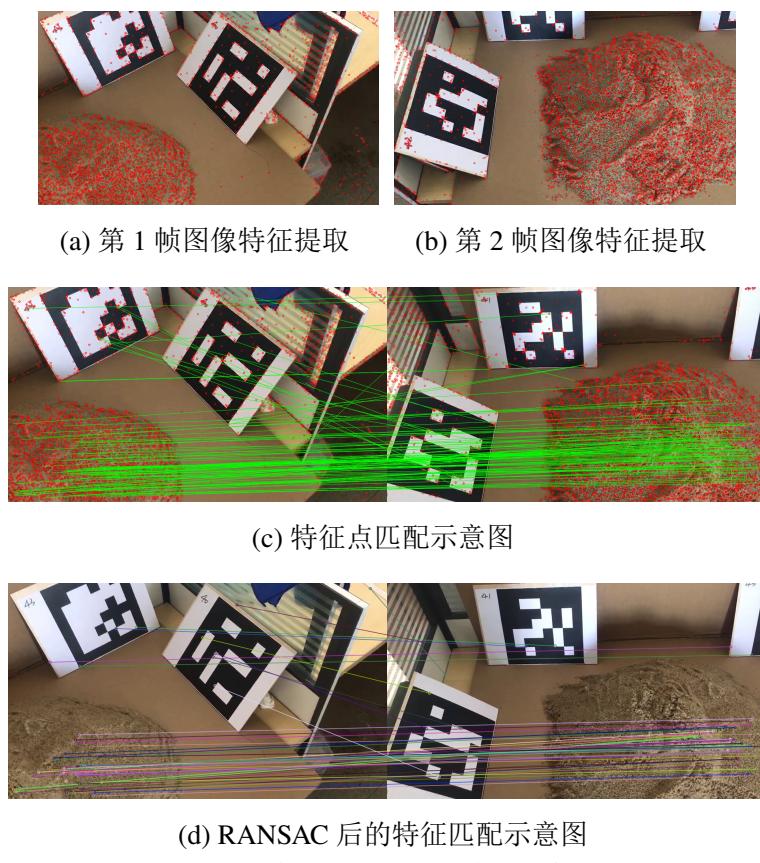


图 3.5 特征提取与匹配结果示意图

3.2.2.3 SfM

在 3.2.2.2 小节中，可以获得初始的匹配关系，但是这种匹配关系不完全可靠，需要添加几何约束进行检测，该集合约束完全依赖于场景中的客观事实。可以通过基本矩阵 F 将匹配好的两帧图像之间的像素坐标 $(x,y), (x',y')$ 进行关联，假设一个符合条件的匹配对像素坐标需要满足以下公式：

$$\begin{bmatrix} x' & y' & z' \end{bmatrix} F \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0 \quad (3.9)$$

找到相机基线最大的像对，根据该像对，通过 RANSAC 八点法计算本征矩阵，再通过对本征矩阵 SVD 分解得到第二个图像的 R 、 T ，在这一步需要进行畸变校正，然后根据 R 、 T 和矫正后的像点坐标三角计算出三维点。当所有的两两匹配图像对被确定以后，可以开始计算相机的位姿（ 3×3 的旋转矩阵 R ， 1×3 的平移向量 t ），摄像机的内参（焦距 f ，畸变参数 k_1, k_2 ）。几何场景提供轨迹中的每个 3D 点 X_j ，通过投影方程，将 3D 点投影到摄像机的 2D 成像平面上，投影误差的定义为投影点和图像上真实点之间的欧式距离，如图 3.6 所示。

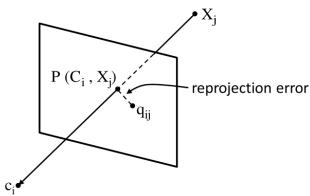


图 3.6 误差投影示意图

对于 n 个视角和 m 个轨迹，投影误差的目标优化方程为：

$$g(C, X) = \sum_{i=1}^n \sum_{j=1}^m \omega_{ij} \|q_{ij} - P(C_i, X_j)\|^2 \quad (3.10)$$

其中 $\|q_{ij} - P(C_i, X_j)\|^2$ 就是摄像机 i 中的轨迹 j 的投影误差累积和，SfM 算法的目标就是找到合适的相机和场景参数去优化这个目标函数， g 是采用一个非线性最小二乘的优化方法求解，常用 BA（光束平差法）来优化上述过程。最后，不断添加新的摄像机和 3D 点进行 BA。这个过程直到剩下的摄像机观察到的点不超过 20 为止，说明剩下的摄像机没有足够的点可以添加，BA 结束。得到相机估计参数和场景几何信息，即稀疏的 3D 点云。

3.2.2.4 MVS

SfM 是指从运动中恢复结构的过程，而 MVS 则是多视角立体视觉生成统，SfM 生成的是稀疏点云，恢复相机之间的几何关系，MVS 生成的是稠密点云，由 SfM 获得的一些相机参数和相机之间的几何关系，来进行 MVS。在 SfM 中，重建的点都是由特征匹配的点，这些点集本身就就不稠密，因此无法获取到稠密的点云，而在三维重建的过程就需要通过 MVS 的方式获取稠密的点集，MVS 利用图像中的像素点来实现点云的重建，将图像中的每一个像素点估计其三维坐标，构成稠密点云。

在稠密点云的估计过程中，无法将每一个像素点按照特征点的方式计算其描述子，因此提出了极线搜索和快匹配技术来匹配图像中的某一像素点在其他图像中的对应的点，在找到每一个像素点在其他图像中出现的位置之后，就可以

利用三角测量的发放确定其深度，但是因为用一个像素点会出现在多个图像中，所以就期望通过多次三角化让该点的深度值收敛。

对于极线搜索和块匹配技术，如图3.7所示。

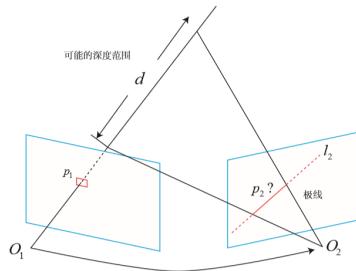


图3.7 极线搜索示意图

首先对于极线搜索和块匹配技术，如图所示，左侧的相机观测到了像素点 p_1 ，由于该相机为单目相机，没法确定深度，可以先假设该点的实际位置在0到正无穷的区间中，因此该像素对应的空间点就分布在某条线段上，对于右侧相机，上述线段可以在成像平面上生成一段投影，在已知两帧图像之间的相对位姿时，是可以确定该极线的位置的，接下来就需要在极线上寻找与像素点 p_1 所对应的 p_2 点的位置。

因为单个像素没有对应的特征值，无法进行匹配，此外依靠单个特征点的亮度来进行匹配也并不可靠，可以基于图像块在前后变化得到过程中的灰度不变性特征，在 P_1 周围选育一块 w^*w 得到小块，然后在右侧的极线上选择多个这样的小块来提高区分度。对于计算两个小块之间的差异，一般可以通过以下方法来计算：

1.SAD 取两个小块的差的绝对值之和

$$S(A, B)_{SAD} = \sum_{i,j} |A(i,j) - B(i,j)| \quad (3.11)$$

2.SSD 取两个小块比较求平方和

$$S(A, B)_{SSD} = \sum_{i,j} (A(i,j) - B(i,j))^2 \quad (3.12)$$

3.NCC 取两个小块计算相似性

$$S(A, B)_{NCC} = \frac{\sum_{i,j} A(i,j)B(i,j)}{\sqrt{\sum_{i,j} A(i,j)^2 \sum_{i,j} B(i,j)^2}} \quad (3.13)$$

在极线上，计算了 A 与每一个 B_i 的相似性 y_{ij} 度量，那么将得到一个沿着极线的数值分布，该分布的形状取决于图像本身信息，在搜索距离较长的情况下，通

常会得到一个非凸函数：该分布存在许多峰值，但是真实值只有一个，接下来需要使用深度滤波器使用概率分布来描述深度值从而找到对应的 p_2 点。

根据以上流程，可以对图 2.10 所示的连续堆体图像进行三维重建，堆体点云重建结果如图 3.8 所示。

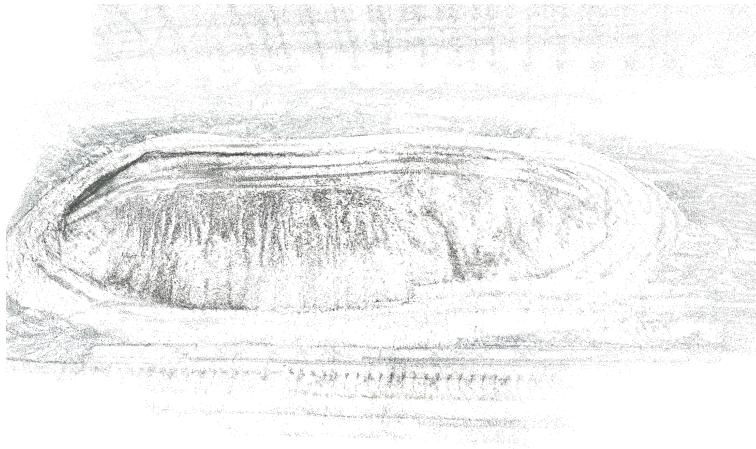


图 3.8 堆体稀疏重建示意图

3.2.3 当前三维重建存在的问题

根据 3.2.2 节的描述，可以发现通过现有的三维重建技术流程可以实现对多张连续图像到三维点云的转化，整个过程操作简便，且能够得到一个较好的结果。但是从整体耗时情况，三维点云结果的精确性来看，依然还存在以下问题：

1. 整个三维重建的过程十分耗时，由于三维重建中的图像输入没有时序信息，关键帧之间的匹配通过完全枚举的方式进行，当输入 N 张原始图像时，系统的时间复杂度高达 $O(N^2)$ ，由单帧图像到稀疏点云一般耗时为几分钟，由单帧图像到稠密点云耗时更是会高达几个小时。
2. 三维点云的精度不高，噪音点较大，一方面是由于三维重建往往采取增量式的重建方式，导致最终场景重建的结果无法闭合；另外一方面由于对于某些重复度较高的场景，关联帧之间的匹配由于缺少时序信息，从而导致匹配精度较低，在解算相机位姿时，得到的结果也无法保证正确，如图 3.9 所示。

3.3 三维重建优化方案研究

3.3.1 增加实效性方案

对无序的图像进行三维重建时，会有大量的时间耗费在图像的匹配过程中，在三维重建中常见的图像匹配方法包含以下几种模式：

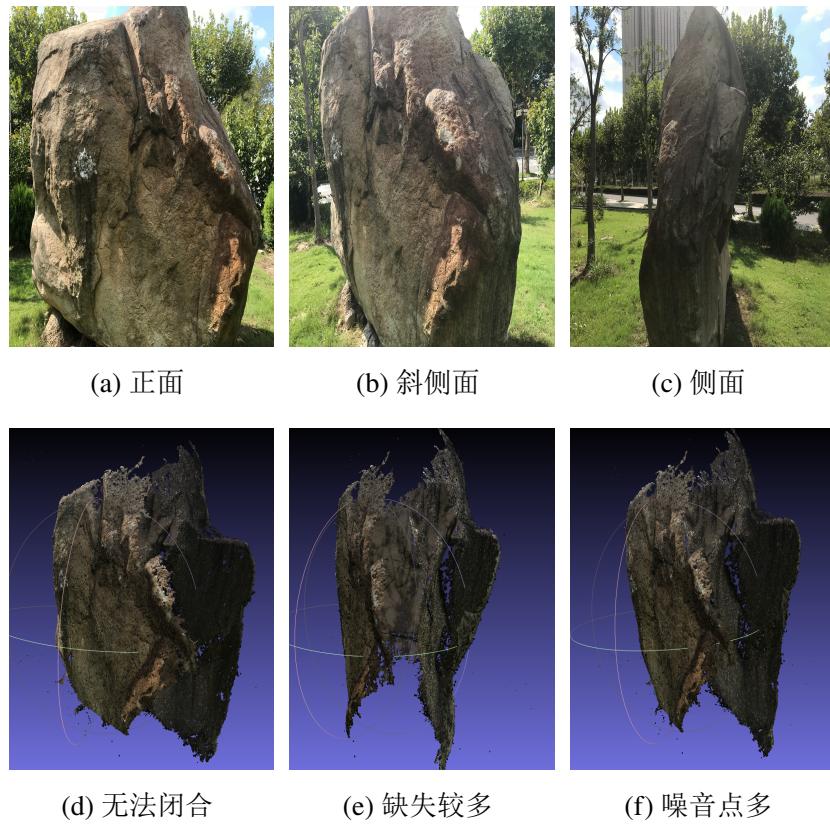


图 3.9 自然场景三维重建结果示意图

1. 完全匹配：适用于三维重建的图像数量较低时，这种匹配模式一般可以较快的完成，并且产生一个比较好的匹配结果，在这种模式里，每一帧图像都会和其他所有的图像进行匹配验证。但是一旦图像的数量较大，那么这种图形匹配方式将极其耗时，延缓整个三维重建的实效性。

2. 序列匹配：一般用于图像帧连续获取的情况下，存在视觉重叠部分，这样就不需要对所有帧进行完全匹配，只需要考虑前后帧即可。

3. 空间匹配：考虑每一帧图像的空间位置，通过空间位置获取到和其相邻的匹配帧，每一帧的空间位置可以通过图像自带的 GPS 信息获取，该模式对提供的图像空间位置精度要求较高。

4. 传递匹配：当确定帧 A 和帧 B 都和帧 C 有匹配关系时，那么就默认帧 A 和帧 B 也具备匹配关系，这样的匹配模式实效性较快，但是误差较大。

5. 自定义匹配：即在系统进行匹配查找时，就将所有的匹配关系自定义给出，以自定义的匹配结果代替三维重建中的匹配方法。

对于上述几种匹配模式，需要同时考虑到匹配时效和匹配精度的问题。排除空间匹配的方法，因为在密闭的环境中很难获取到每一帧图像对应的真实 GPS 信息，图像所自带的 EXIF 信息也会受到 GPS 强弱的影响，此外通过无人机获取到得到图像，即使有两帧之间的空间位置十分接近，也无法保证两帧图像具备匹配关系，例如在无人机正反来回的的两帧即使空间位置十分接近，也不一定是匹

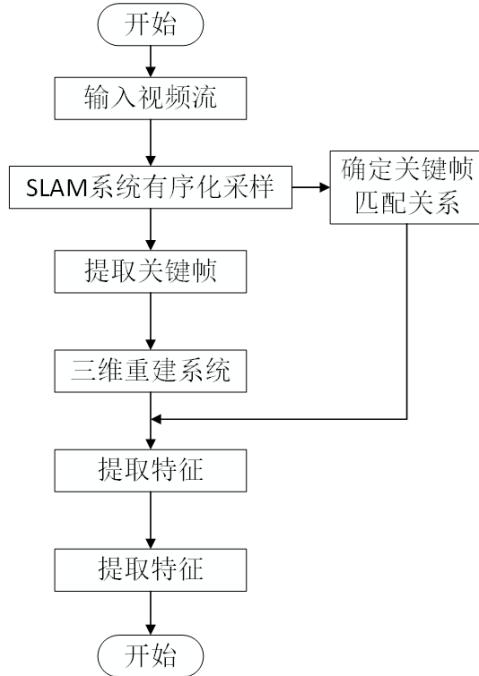


图 3.10 融合 SLAM 结果的三维重建流程图

配帧。

对于自定义的匹配模式，本文将以 SLAM 的结果提供给三维重建进行图像匹配，即以有序化的图像输入代替无序化的图像输入，一方面可以避免完全匹配带来的耗时问题，同时具备时序信息的 SLAM 所产生的匹配结果更加精确。对于具体的实现过程，以连续视频流作为 SLAM 系统的输入，筛选出其中的关键帧和所有关键帧之间的对应关系，将 SLAM 中的关键帧作为三维重建的图像输入，关键帧之间的匹配关系作为三维重建的先验匹配结果，具体流程如图 3.10 所示。以下将讨论 SLAM 中关键帧的提取和匹配策略。关键帧相当于 SLAM 的框架，是在局部一系列普通帧中选出一帧作为局部帧的代表，记录局部信息，但相机在场景中的某个区域内移动时，普通帧的集合会不断增减，导致连续的两个普通帧之间会存在大量的信息冗余，如果所有普通帧全部参与计算，就会极大的浪费算力和内存，因此为了保证整个 SLAM 系统的良好运行，都会选择关键帧来作为优化对象。此外，关键帧选择时还会对图片质量、特征点质量等进行考察，一定程度上也发挥了滤波的作用，防止无用的或错误的信息进入优化过程而破坏定位建图的准确性。

选择关键帧主要从关键帧自身和关键帧与其他关键帧的关系两个方面来考虑。一方面，关键帧自身质量要好，例如不能是非常模糊的图像、特征点数量要充足、特征点分布要尽量均匀等等；另一方面，关键帧与其他关键帧之间的关系，需要和局部地图中的其他关键帧有少量的共视关系，但大部分特征点是新特征点，以达到既存在约束，又尽量少的信息冗余的效果，例如局部地图点投影到此帧的点数低于一个阈值或前一个关键帧的特征点在此帧里已经有 90% 观测不

到等等。对于关键帧的选择，一般有以下策略：

1. 距离上一关键帧的帧数是否足够多，主要从间隔的时间上来考虑，例如可以直接间隔固定帧数选择一个关键帧，这样操作流程简单，但是整体取帧效果不好，因为对于一些运动缓慢的场景，会选择大量相似的关键帧，再次造成冗余的情况，在运动较快的场景，又容易造成大量有效帧的缺失。

2. 距离上一关键帧的距离是否足够远，主要从空间位置上来考虑，在 SLAM 运行的过程中，可以根据相机位姿得到帧和帧之间的位置关系，如果运动的距离足够大，那么可以直接将其确定为下一个关键帧，但在对某一个场景进行重复来回运动时，就会收集大量重复的关键帧。

3. 跟踪质量，主要从共视特征点上来考虑，在 SLAM 的过程中会记录下当前视角中的信息，一旦检测到离开当前视角则加入新的关键帧，和上两种方法相比较能够更有效的获取到关键帧。

对于本文中三维重建图像输入的选择，优先选择第三种关键帧提取方案，避免了提取帧率过小，而丢失一些三维重建时关键的帧，提取帧率过大而造成图片集合存在大量冗余的问题。此外通过 SLAM 的筛选策略也可以提前过滤掉存在运动模糊和质量过低的图像。

3.3.2 增加精确性方案讨论

针对 3.2.3 节所提出的问题，本文提出一种结合 SLAM 结果的三维重建方法，包括 SLAM 生成的 KeyFrame DataBase 以及位姿信息，以提高点云结果的鲁棒性和准确性。本文所提出的建图过程分为两个阶段：在线 SLAM 阶段和离线 3D 重建阶段。

在线 SLAM 阶段实时获取单目相机的图像信息，结合 SLAM 技术快速建立 scene graph 及点云环境地图。由于在线 SLAM 注重实时性能，只是对局部窗口进行增量优化，即使在闭环时也只进行轨迹和部分点云优化，可以动态获取 KeyFrame DataBase 和较为准确的位姿信息。所建立的 scene graph 及点云环境地图的精度较低。随着时间的推移不可避免的会出现误差累计，导致闭环处出现轨迹和环境地图的断口问题。

离线 3D 重建阶段首先基于在线 SLAM 阶段所建立的粗糙 scene graph 及点云环境地图，以大尺度地图及运动轨迹为优化目标，利用高性能计算机的快速处理能力进行全局优化，得到更加精确的全局稀疏地图和运动轨迹；然后基于 MVS 技术建立稠密/半稠密环境地图。下面分别详细说明在线 SLAM 过程和离线 3D 重建过程。

3.3.2.1 在线 SLAM 阶段

本文所提出的 SLAM 系统整体框架如图 3.11 所示。系统的输入为单目相机（内参标定结果已知）。SLAM 系统有前端-后端两个部分构成，前端进行特征提取及数据关联，后端进行图模型建模及 Maximum a Posteriori 优化。当前主流

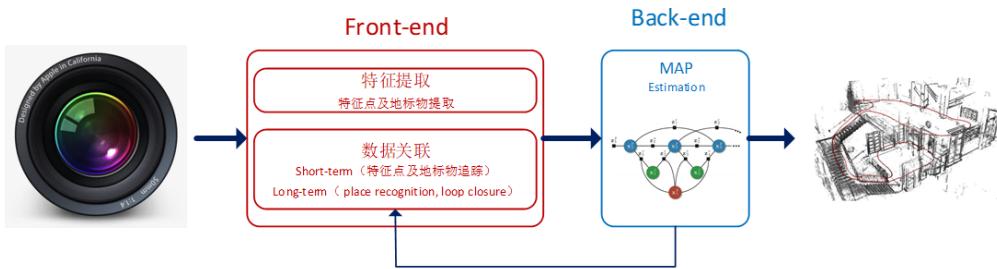


图 3.11 前视 SLAM 整体框架

vSLAM 通过多线程技术维护 3 个模块，实现前端-后端的各项功能，如图 3.12 所示。系统以地图数据为核心，以 BA 图模型优化为手段，实现了相机追踪、地图维护、闭环优化。图中绿色模块代表传统点特征数据，通过特征提取、数据关联、地图维护、词袋维护，闭环检测等流程被不断的加入到 factor graph 中并进行优化估计。

追踪: 模块主要负责计算相机在这两帧之间的相对运动，通过相机姿态累计得到相机的运动轨迹。首先通过图像预处理提取点特征及语义地标特征，基于两帧间特征匹配利用 Motion-only BA 进行运动姿态估计，基于图像特征与地图数据关联结果利用闭环进行姿态优化，最后通过计算当前帧的信息增量的多少判断其如果是关键帧则将其输送给地图构建模块。

地图构建: 模块将追踪模块新输送过来的关键帧数据与已有关键帧数据根据可共视性约束进行数据匹配，在 factor graph 图模型中加入新的点特征节点并完成数据关联，根据可共视性约束确定局部优化范围，通过 Local Mapping BA 建立并优化地图数据。

闭环: 模块通过检测匹配环境地标以判断相机是否再次运动到之前曾经到过的区域，并根据这一信息对相机的运动轨迹及地图数据进行全局修正，降低累积误差。本项目拟基于 BoW 词袋技术进行闭环检测。在闭环修正时时首先对地图中的重复点特征进行融合，然后对闭环主干网络进行全局 BA，实现轨迹和地图数据的正确闭环。

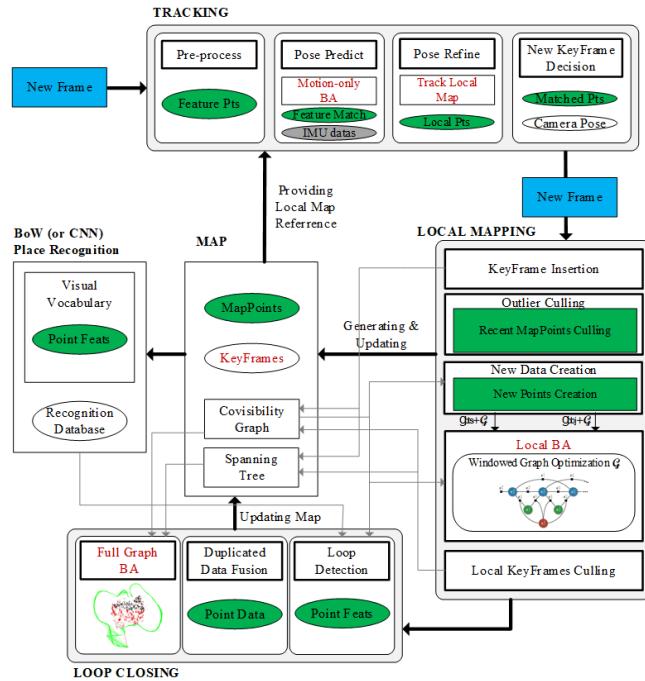


图 3.12 SLAM 系统框架图

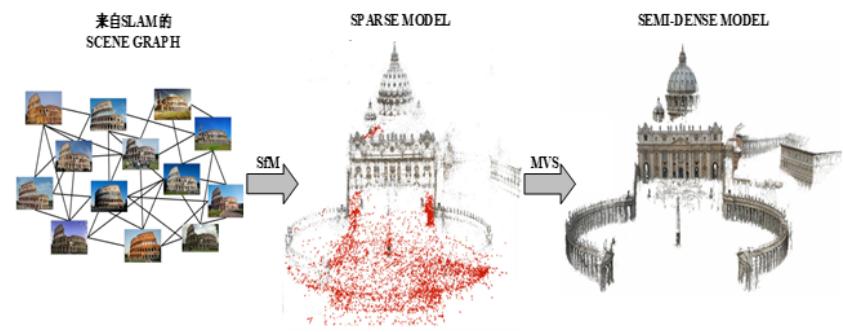


图 3.13 离线 3D 重建技术流程图

3.3.2.2 离线 3D 重建阶段

本文 3D 重建以提高建图精度为目标，其流程如图 3.13 所示。首先基于在线 SLAM 结果建立粗糙 Scene Graph 及点云环境地图，以全局大尺度地图及运动轨迹为优化目标，使用鲁棒性更高的特征描述算子，利用高性能计算机的快速处理能力进行全局 BA 优化，得到更加精确的全局稀疏地图和运动轨迹。然后利用 MVS 技术对图像中的高梯度变化区域建立半稠密环境地图。利用 MVS 进行半稠密建图首先提取关键帧中的边沿特征，基于 SfM 优化过的可共视图模型利用 triangulation 计算融合边沿特征的深度信息。

3.4 本章小结

本章研究了基于传统三维重建系统的实现流程，包括数据的采集，特征的提取匹配，SfM 过程获取稀疏点云，MVS 过程获取稠密点云等，只需要向系统中输入围绕堆体拍摄的连续图像，即可得到关于堆体场景的点云模型。

本章揭示了传统三维重建存在的问题，包括系统耗时以及点云模型结果精确度不高等问题，提出了基于传统三维重建的改进方法，即结合实时运行的 SLAM 系统得出的粗糙 scene graph 及点云环境地图结果，为三维重建提供先验信息，结合三维重建全局大尺度地图及运动轨迹的优化目标，使用鲁棒性更高的特征描述算子，进行全局 BA 优化，得到更加精确的全局稀疏地图和运动轨迹。

通过本章基于改进后方法获取到的堆体点云模型，可以提供给基于纯视觉方法测量堆体体积。因此，本章需要获取到高精度的堆体点云结果，以减小后续体积测量误差。

第4章 基于纯视觉的堆体体积测量方法研究

4.1 引言

利用三维重建的技术流程，可以通过输入二维图片序列的方式获取到场景中的三维稀疏点云或者稠密点云，以视觉特征的方式来表达场景中的信息。再进一步，可以利用重建后的堆体点云来测量其体积，这些点云可以描绘出空间信息，对于体积测量的实现，还存在以下两个问题需要解决：

1. 在三维重建的过程中，由于点云坐标系以相机坐标系为参考坐标系，因此点云坐标系和真实世界坐标系无法对齐；
2. 对于大尺度的堆体场景，需要得到该场景的水平面所在方程，由于物体的遮挡，水平面很难直接通过视觉的方法构建出来，从而导致整个场景非闭合；
3. 由于三维重建的信息来源于单目相机，所构建出的点云也没有绝对尺度，无法得到实际的体积真值。

在解决上述两个问题的基础上，就可以对堆体进行高度，面积或者体积的测量，对比传统方法中用激光雷达，或者称重器械来测算体积的方式，现在就可以通过单个摄像头以纯视觉的方式来完成上述过程，并获取到一个精确的结果。本章将以三维重建的点云结果为基础，为了解决点云的尺度问题，以及求解堆体三维点云的水平面方程，本文将在场景中引入二维码视觉标签，依靠其因为携带尺度信息和唯一 ID 值的特性估计出尺度信息，并且还具备易检出的角点坐标用来估计水平面方程。在获取到尺度和水平面方程后，结合点云信息可以再进一步计算三维场景的体积，具体流程如图 4.1 所示。

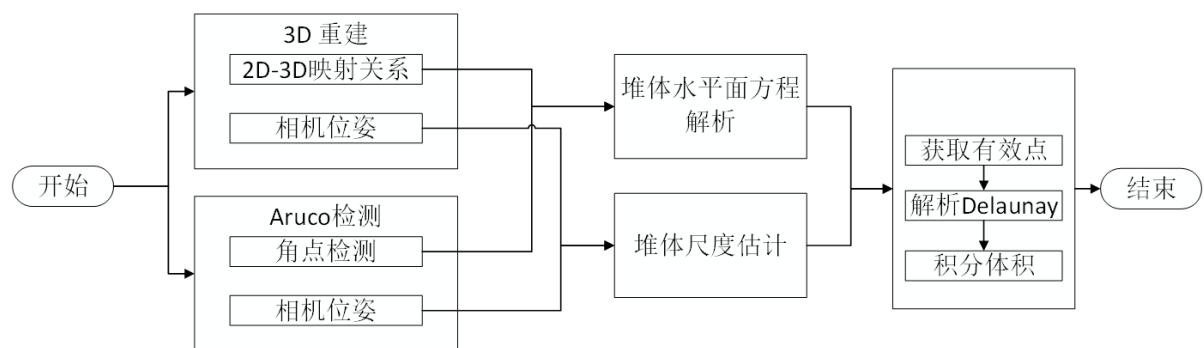


图 4.1 体积测算流程图

4.2 堆体水平面方程解析方法

4.2.1 2D/3D 点坐标检测

对于水平面的求解，其基本思路为求解落在堆体水平面上二维码的角点所在的坐标平面，或者求解已知角点和堆体水平面的实际空间位置关系，通过角点求解堆体实际水平面方程，二维码如图 2.2 所示，将其作为媒介来估计平面方程。

首先检测 2D 点，可以利用二维码检测算法检测出场景中二维码的 ID 值，位置，大小，角点坐标等信息，检测结果如图 4.2 所示。因为在场景中，二维码的布置都会有下侧两个角点直接落在水平面上，因此可以认为所有二维码的下侧角点所构成的平面即是场景的水平面。那么接下来即可将水平面的求解转化为二维码角点所在平面的求解。在检测的过程中，需要将所有 2D 图片的索引和角点的位置记录下即可。



图 4.2 二维码检测示意图

其次再检测 3D 点，通过上述步骤，可以获得连续视频帧中每一帧的二维码角点的坐标值，同样的，可以直接将这些包含二维码的视频帧序列作为三维重建的输入以获取场景的点云，考虑到二维码角点被作为特征点时极易被检出，因此只需要通过稀疏点云，即可获取角点在二维图像中的坐标和三维重建点云中的三维坐标之间的对应关系。

在三维重建的过程中，可以获取到每一帧图像中所有特征点所对应的三维重建点云中的三维坐标，无论是稠密点云或者稀疏点云，都可以得到一个对应关系，但是考虑到遍历的效率问题，则选择稀疏点云中的点击作为遍历对象。在寻找对应关系的实际过程中，会遇到二维坐标无法映射到三维坐标点的情况，那么则会优先选择距离二维点坐标欧氏距离最近的点作为替代对象。此外，还需要注意的是，若所布置的二维码的所有下侧点都位于同一条直线上，那么这样就无法获取准确的水平面方程，因为经过同一直线的平面并不唯一，因此在布置坐标的过程中需要注意将所有二维码尽可能分散放置。

4.2.2 平面方程优化方法

根据上述步骤，可以将待求的堆体水平面方程问题转化为求解 N 个三维点所在水平面方程的非线性优化问题。本文选择谷歌开源的 Ceres solver 库作为解决该非线性问题的工具，Ceres 可以解决边界约束鲁棒非线性最小二乘法优化的问题，表达式如下：

$$\begin{aligned} \min_x \quad & \frac{1}{2} \sum_i \rho_i(\| f_i(x_{i_1}, \dots, x_{i_k}) \|^2) \\ \text{s.t.} \quad & l_j \leq x_j \leq u_j \end{aligned} \quad (4.1)$$

其中 $\rho_i(\| f_i(x_{i_1}, \dots, x_{i_k}) \|^2)$ 为残差块， $f_i(\cdot)$ 为代价函数，代价函数的求解依赖于一系列参数 $[x_{i_1}, \dots, x_{i_k}]$ 所有参数构成参数块，限制边界分别为 l_j, u_j 。 ρ_i 为损失函数，其作用是减少异常值对优化结果的影响。在利用 Ceres 解决非线性问题时，通常分为以下三个步骤：1. 构建代价函数，也就是具体问题所对应的目标式，本小节具体解决 N 个三维点所在水平面的方程；

2. 通过代价函数构建待求解的优化问题；
3. 配置求解器参数并求解问题，在这一过程中主要是设定求解方程的方式。

上述步骤如代码 4.1 所示。

Listing 4.1 获取平面方程参数的 C++ 实现

```

1  vector<vector<float>> plane_data1;
2  vector<float> planex_data1, planey_data1, planez_data1; // 待优化量
3  // 第一部分：构建代价函数模型
4  struct CURVE_FITTING_COST_plane
5  {
6      CURVE_FITTING_COST_plane (double x, double y, double z)
7      : _x(x), _y(y), _z(z) {}
8      // 残差的计算
9      template <typename T>
10     bool operator() (
11         const T* const plane,
12         T* residual) const
13     {
14         residual[0] = plane[0]*T(_x) + plane[1]*T(_y)
15             + plane[2]*T(_z) + plane[3];
16         return true;
17     }
18     const double _x, _y, _z;
19 };
20
21 int main (int argc, char** argv)
22 {
23     double plane[4] = {0.1, 0.1, 0.2, 0};
24     // 第二部分：构建最小二乘问题
25     ceres::Problem problem2;
26     for (int i=0; i<planex_data1.size(); i++) {
27         problem2.AddResidualBlock

```

```

28     (new ceres::AutoDiffFunction<CURVE_FITTING_COST_plane, 1, 4>
29     (new CURVE_FITTING_COST_plane
30     (planex_data1[i], planey_data1[i], planez_data1[i])),
31     nullptr,
32     plane);
33 }
34 // 第三部分：配置求解器
35 ceres::Solver::Options options2;
36 options2.linear_solver_type = ceres::DENSE_QR;
37 options2.minimizer_progress_to_stdout = true;
38 ceres::Solver::Summary summary2;
39 ceres::Solve (options2, &problem2, &summary2 );
40 for ( auto a:plane ) {
41     outfile_kabc<<a*1/plane[3]<<endl;
42 }
43 return 0;
44 }
```

对于任意平面方程，都可以利用以下方程通过确定 4 个参数的方式来确定唯一解析解。

$$Ax + By + Cz + D = 0 \quad (4.2)$$

在迭代的过程中，同时需要确定每个带估计参数的初始值，最终可以得到方程的四个参数 A,B,C,D。

4.3 堆体尺度估计方法

由单目视觉重建出来的点云地图都存在没有确定尺度的问题，地图中的特征点坐标和相机的位姿都是相对尺度，即最常见的采取初始化成功后的前两帧作为单位尺度，后续的所有关键帧都以此为参考确定尺度。这样的做法可以获取到地图中所有描述的绝对尺度，但无法计算出实际尺度，本文提出两种结合二维码估计尺度大小的方法，通过在场景中引入已知尺度的二维码来确定所构点云地图的真实尺度，主要针对同一场景，计算两种 SLAM 的相机位姿或者两种 SLAM 中地图点坐标来确定相对尺度和绝对尺度之间的比例，从而确定绝对尺度。

4.3.1 位姿法估计

首先，获取绝对尺度的数据，由图 ?? 所示同时可以得到相机在某一个确定的二维码下的位姿，因为二维码自带确定的边长信息，因此可以得到带有绝对尺度的相机位姿 (T_x, T_y, T_z)。在实际的工程中需要注意，必须选择不同的相机对于同一二维码下的位姿，可以通过多帧图像序列获取到多个不同位姿。

其次，再获取同一批图像序列的相对尺度，通过三维重建的结果，就可以得到每一帧相机在参考坐标系下的位姿 (T_x, T_y, T_z) ，该位姿为相对尺度，因为只需要考虑到尺度的大小关系，因为不需要讨论不同坐标系之间的相对转化问题。

以上可以获取到多个针对同一场景的两种 SLAM 相机位姿估计结果，本文提出通过位姿来估计尺度的方法，因为两种 SLAM 方法估计出的相机位姿是基于不同的坐标系得到的，本文提出利用相机在不同坐标系下移动的欧式距离之间的比例来确定尺度，简化公式为：

$$K = \frac{\Delta Pose_{rel}}{\Delta Pose_{abs}} \quad (4.3)$$

同 4.2.2 节，利用 Ceres 库得到最优解 K 值，计算如代码 4.2 所示：

Listing 4.2 获取尺度值的 C++ 实现

```

1  vector<vector<float>> colmap, opencv;
2  vector<float> colmap_data1, opencv_data1; // 待优化量
3  // 第一部分：构建代价函数模型
4  struct CURVE_FITTING_COST_k
5  {
6      CURVE_FITTING_COST_k (double x, double y)
7      // 残差的计算
8      template <typename T>
9      bool operator() (
10          const T* const k,
11          T* residual ) const
12      {
13          residual[0] = T ( _y ) - ( k[0]*T ( _x ) ); // y-ax
14          return true;
15      }
16      const double _x, _y;
17  };
18  int main ( int argc, char** argv )
19  {
20      double k[1] = {0};
21      // 第二部分：构建最小二乘问题
22      ceres::Problem problem;
23      for ( int i=0; i<colmap_data1.size(); i++ ) {
24          problem.AddResidualBlock
25          (new ceres::AutoDiffCostFunction<CURVE_FITTING_COST_k, 1, 1>
26          (new CURVE_FITTING_COST_k
27          (opencv_data1[i], colmap_data1[i])),
28          nullptr,
29          k);
30      }
31      // 第三部分：配置求解器
32      ceres::Solver::Options options;
33      options.linear_solver_type = ceres::DENSE_QR;
34      options.minimizer_progress_to_stdout = true;
35      ceres::Solver::Summary summary;
36      ceres::Solve ( options, &problem, &summary );
37      for ( auto a:k ) {

```

```

38     outfile_kabc<<a<<endl;
39 }
40 return 0;
41 }
```

只需要解出唯一参数 K 即可，在计算的过程中可以随机获取任意两帧之间的位姿，以减小误差。

4.3.2 空间坐标点估计尺度

对于尺度的估计，可以通过场景中已知点之间的距离通过投影获得点云之中点的实际坐标，但是空间中的已知点难以确定，而且也不易保持固定不变，符合以上规则的点的数量较少。因此考虑到场景中二维码标记的角点为研究对象，可以通过 SLAM 求得每一个二维码在图像中的位置以及四个角点的位置，因为角点极易识别，且任意两个角点之间的实际空间距离都能够简易测量得到。

通过第 3 章三维重建的流程，可以获取到每一个点在二维图像中和三维空间点之间的对应关系，如图 4.3 所示表示的是同一个点在二维和三维之间的对应。因为图像中的任意两个二维码角点之间的距离都是已知的，那么也可以得到三维坐标中任意两个角点之间的绝对尺度，将这一尺度因子应用到整个地图中进行缩放，即可获得一个带有真实尺度的地图。

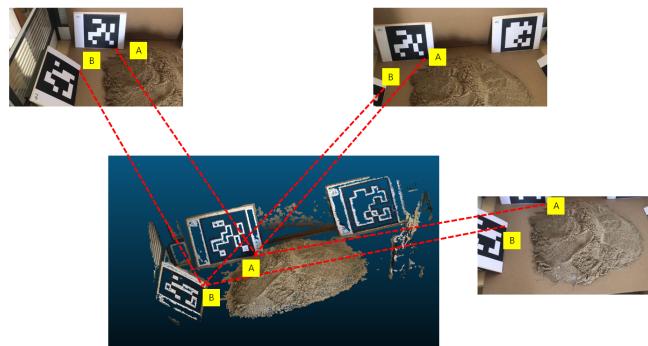


图 4.3 根据空间点估计尺度流程图

4.4 堆体体积测量方法研究

根据以上两节可以获取三维重建后堆体点云的水平面方程和比例尺度，在此基础上可以进一步计算出堆体的实际体积，本文提出一种计算点云实际体积的方法，具体流程如下：

1. 将所有堆体点云中的 3D 点投影在水平面上，根据场景的实际空间约束和过滤方法提纯出有效 3D 点集；

2. 通过空间变换，将三维空间点转化为水平面上的点 2D 点集；
3. 对所有 2D 点集求 Delaunay 三角形，计算每个三角形对应的水平高度，得到三棱柱体积；
4. 剔除异常三棱柱，将所有有效的三棱柱积分求和得到总体积。

流程图如图 4.4 所示。

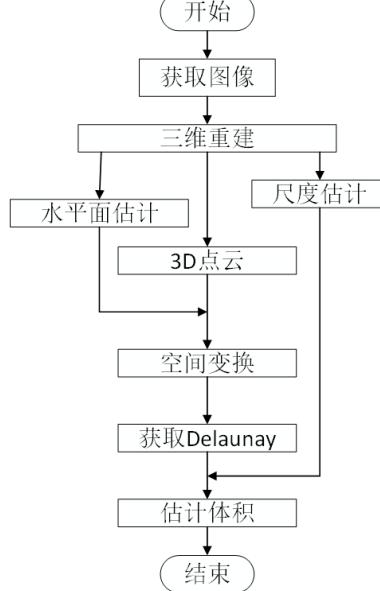


图 4.4 纯视觉测量体积流程图

4.4.1 有效 3D 点集获取

通过第 3 章的方法，可以对堆体场景获取三维稀疏点云和稠密点云，在生成点云的过程中会把部分非感兴趣区域的内容和一些离散的误差噪音点添加至点云结果中。针对这两种情况，本文分别提出以下对应的解决方法：

1. 针对非感兴趣区域，本文通过三维空间平面方程对非感兴趣区域直接进行切割剔除；
2. 针对随机误差噪音点，本文通过离群点检测算法进行剔除。

对于方案 1，对于重建后的三维点云，只需要收集感兴趣区域内的 3D 点集，如图 4.5 所示，只有红色框图内区域是体积测量的感兴趣区域，通过上一节求出的水平面方程和二维码的底边点坐标共同约束求出空间切面方程对场景进行切割。

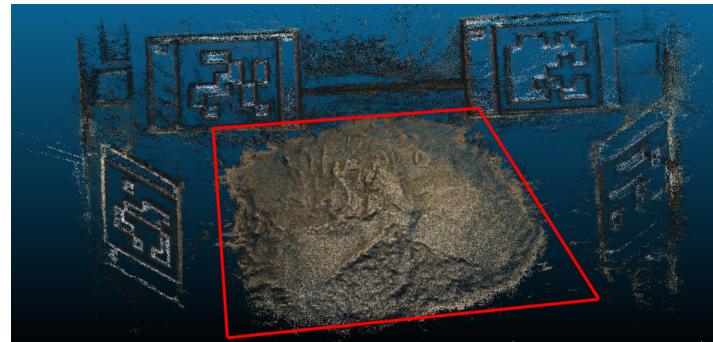
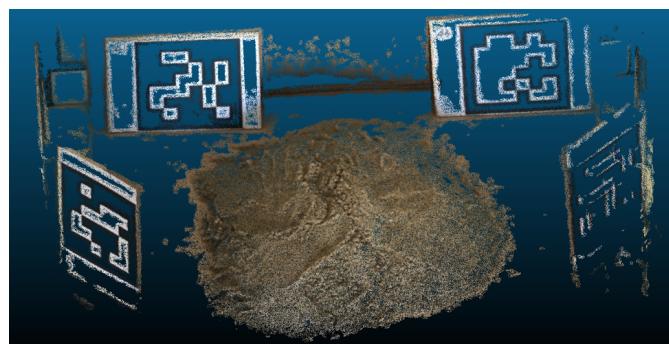


图 4.5 三维重建感兴趣区域示意图

对于方案 2，本文采用统计滤波器即为对每个点的邻域进行一个统计分析，并修剪掉一些不符合标准的点，具体方法为在输入数据中对点到临近点的距离分布的计算，对每一个点，计算它到所有临近点的平均距离（假设得到的结果是一个高斯分布，其形状是由均值和标准差决定），那么平均距离在标准范围之外的点，可以被定义为离群点并从数据中去除。采用统计滤波对点云中的离散点进行滤波对比如图 4.6 所示。



(a) 对点云统计滤波前



(b) 对点云统计滤波后

图 4.6 点云统计滤波前后效果对比示意图

通过以上步骤，可以求出有关堆体的有效 3D 点集，接下来需要将所有三维点云投影至二维平面上获取 2D 点集，简略步骤如下：

1. 通过 4.2 节获取点云水平面和参考坐标系中的 xy 平面方程的法向量分别为 $U(A,B,C)$ 和 $V(0,0,1)$ ；

2. 将所有的 3D 点集 $O(x_o, y_o, z_o)$ 根据以下公式投影水平面上, 获取新的 3D 点集 $P(x_p, y_p, z_p)$:

$$\begin{aligned}x_p &= \frac{(B^2 + C^2)x_o - A(By_o + Cz_o + D)}{A^2 + B^2 + C^2} \\y_p &= \frac{(A^2 + C^2)y_o - B(Ax_o + Cz_o + D)}{A^2 + B^2 + C^2} \\z_p &= \frac{(A^2 + B^2)z_o - C(Ax_o + By_o + D)}{A^2 + B^2 + C^2}\end{aligned}\quad (4.4)$$

同时也可以根据公式计算出每一个三维空间点到水平面的方程的距离

$$d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (4.5)$$

3. 计算旋转角度:

$$\theta = \arccos\left(\frac{U \cdot V}{|U||V|}\right) \quad (4.6)$$

4. 计算旋转轴: 在计算旋转角时可知, 旋转角所在的平面为有 P 和 Q 所构成的平面, 那么旋转轴必垂直该平面。假定旋转前向量为 $a(a_1, a_2, a_3)$, 旋转后向量为 $b(b_1, b_2, b_3)$ 。由叉乘定义得旋转轴 $k(k_1, k_2, k_3)$ 为

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix} \quad (4.7)$$

5. 根据罗德里格旋转公式将旋转角, 旋转轴的表达转化为旋转矩阵 R

$$R = \cos \theta I + (1 - \cos \theta)kk^\top + \sin \theta k^\wedge \quad (4.8)$$

6. 将 3D 点集 $P(x_p, y_p, z_p)$ 右乘上述旋转矩阵 R 可以得到一系列在同一平面上 (所有转变后的点 z 值都相同) 的 3D 点集, 提取所有 3D 点的 x, y 值构成 2D 点集 $p(x, y)$ 。

4.4.2 Delaunay 三角网

通过 4.2 小结, 可以得到在由在同一平面上的 3D 点集转化成的 2D 点集和每一个 2D 点集对应的距离值 d 。在本节中将对 2D 点集进行三角剖分, 获得 Dealunay 三角网, Delaunay 三角网是一系列相连的但不重叠的三角形的集合, 而且这些三角形的外接圆不包含这个面域的其他任何点, 如图 4.7 所示。

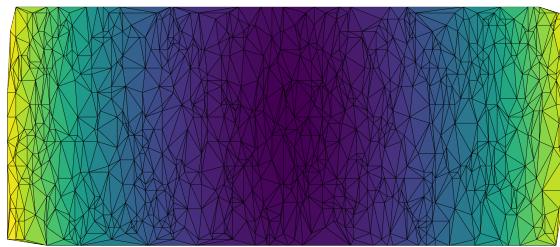


图 4.7 Delaunay 三角网示意图

获取 Delaunay 三角网格的一般步骤为：

1. 构造一个超级三角形，包含所有散点，放入三角形链表；
2. 将点集中的散点依次插入，在三角形链表中找出其外接圆包含插入点的三角形（称为该点的影响三角形），删除影响三角形的公共边，将插入点同影响三角形的全部顶点连接起来，从而完成一个点在 Delaunay 三角形链表中的插入；
3. 根据优化准则对局部新形成的三角形进行优化。将形成的三角形放入 Delaunay 三角形链表；
4. 循环执行上述第 2 步，直到所有散点插入完毕。

由于 Delaunay 剖分具备最接近性和区域性的特性，可以对点集所构成的平面进行拟合。通过以上流程可以获得三角形集合以及三角形中每个顶点对应的距离，取三个顶点距离的平均值作为三棱柱的高，即可获得所有三棱柱的体积。

4.4.3 积分三棱柱

通过以上步骤，可以将三维重建后点云体积的求解转化成对多个三棱柱积分的求解，因为所有三棱柱的体积值都分布在某一个区间内（并不完全服从正太分布），对于部分偏差较大的三棱柱体积值应该通过算法进行剔除。本文采用箱形图的方法来处理这一问题，箱形图不受异常值的影响，能够准确稳定地描绘出数据的离散分布情况，同时也利于数据的清洗，箱形图如图 4.8 所示。

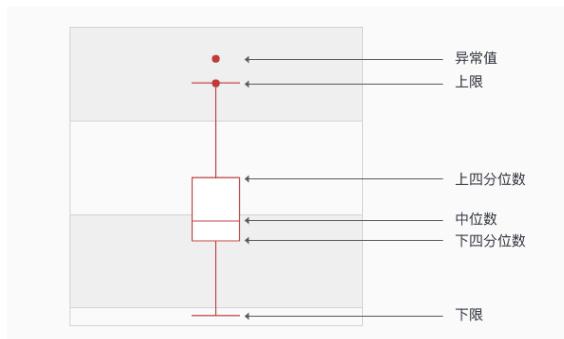


图 4.8 箱形图示意图

主要需要求解出数据中的 5 个特征数据值，包括下四分位数 Q1，上四分位数 Q3，四分位距 $IQR=Q3-Q1$ ，以及上限 $Q3+1.5IQR$ 和下限 $Q1-1.5IQR$ ，对于上下限之外的内容需要剔除，最终结合估算出的尺度，以及所有有效三棱柱的体积按照公式 4.9 即可估计出物体中感兴趣区域的实际体积值。

$$V = K^3 * \sum_i \frac{1}{3} * S_{\Delta} \times \sum_{j=1}^3 d_j \quad (4.9)$$

4.5 本章小结

本章提出了一种面向堆体场景的体积测量方法，主要包括水平面方程的求解，尺度值的估计以及计算 3D 点云体积等流程，该方法完全基于纯视觉方式实现，在获取到堆体三维点云后，可以自动化估计出堆体的整体体积。

本章重点分析了堆体体积测量的可行性和数学描述过程，包含通过位姿法或者空间点坐标值法，利用非线性优化估计出堆体场景实际尺度和堆体点云相对尺度之间的比例关系；通过 2D 角点获取对应 3D 映射点，将水平面方程求解转化求解点集共面方程的问题；最后结合尺度估计值和水平面投影以及空间坐标变换求解 Delaunay 三角形，积分三棱柱以估计出堆体点云的实际体积。

本章以无人机的自主定位和飞行和高精度的三维重建为基础，提出了面对堆体场景的纯视觉体积测量技术，完整流程可以自动化完成，复用于相似场景。

第 5 章 无人机视觉定位与堆体重建及体积测量测试与分析

5.1 引言

本章将对以上章节提出的方法通过实验进行验证，主要包括无人机的自主定位和飞行精度测试，三维重建生成点云的实效性以及精度测试和面向堆体的体积精度测试。本章将对这三部分内容进行分模块处理，包括实验设计，方法测试，数据分析等，对方法表现进行定性分析，对实验数据进行定量比较。

对于无人机的自主定位和飞行精度测试，本章将对实际场景展开无人机的定位和飞行，结合第 2 章所提方法，利用场景中的二维码进行有尺度且基于真实世界坐标系下自主定位，并基于定位结果进行无人机的循迹飞行。在无人机飞行的过程中，实时记录下无人机的飞行状态，和二维码地图的数据，对精度进行定量分析。

对于三维重建生成点云的实效性以及精度测试，本章将对比第 3 章所提出的稀疏点云重建所采用的图像匹配模式，定性比较各匹配模式之间的实效性，并结合 SLAM 将运行后的 keyFrame DataBase 以及先验位姿提供给离线三维重建系统，定性比较三维重建后点云的精度差异。

对于面向堆体的体积测量精度测试，本文选用沙堆作为研究对象，利用其易获取真值且能够任意改变形状的特点，代替其他堆体进行试验。结合第 4 章的方法，测试堆体三维点云和实际堆体之间的尺度估计值，堆体点云的水平面方程，以及利用纯视觉方法测试堆体体积的稳定性和精确度。

5.2 无人机视觉定位系统测试

5.2.1 无人机视觉定位系统实验步骤设计

在第 2 章，本文提出了一种结合二维码的 SLAM 视觉定位系统，利用该系统可以估计出带有真实尺度的相机位姿，以及能够得到真实世界坐标系下的相机位姿。本次实验主要利用上述方法在不利用 GPS 信号的环境下，结合二维码视觉标签进行无人机定位，通过获得到的位置信息进行下一步的飞行控制。本次实验的准备过程主要分为三个步骤，布置场景，生成地图，无人机循迹飞行。

首先是场景布置，选择在空旷场地中，布置二维码视觉标签，保证每个二维码的尺度大小完全一致，且二维码之间尽可能等间距布置，场景中的二维码 ID

完全独立不同，设计图和实际布置图分别如图 5.1(a)和图 5.1(b)所示。在本次实验中，所原选择的二维码实际大小为 0.73m，相邻二维码之间的距离为 4m，整个实验区域面积为 $400m^2$ ($20m \times 20m$)。

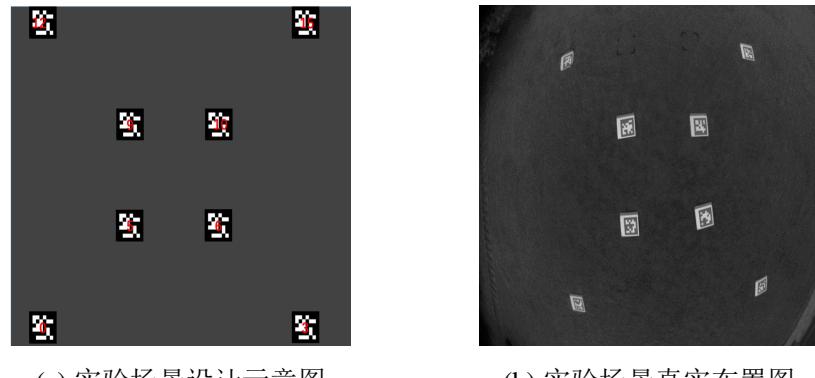


图 5.1 实验场景示意图

在布置好场景后，需要根据实际场景生成地图信息。当无人机首次在无先验地图的环境下飞行时，为了保证后续飞行精度，可以先飞行一次建立场景的先验地图，在无人机的飞行过程中，其飞行区域需要尽可能覆盖所有场景以确保生成完整的地图，根据实际场景生成的地图如图 5.2 所示，其中正方形框代表二维码，蓝色相机表示代表视觉定位产生的关键帧，点代表地图中的特征点。

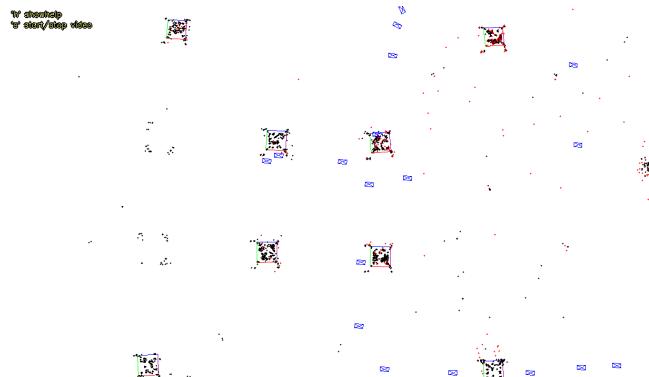


图 5.2 视觉算法生成地图

当获取到完整的地图信息地图后，可以让无人机按照认为规定的轨迹进行自动循迹飞行，首先设计如下图 5.3 所示的轨迹图，起飞点为 ID=0 的二维码处，红色箭头代表无人机的飞行轨迹，对于无人机的飞控过程，按照设定循迹点的方案来实现，在整个轨迹中给定多个循迹点坐标，使得无人机按照设定的坐标顺序进行飞行即可。

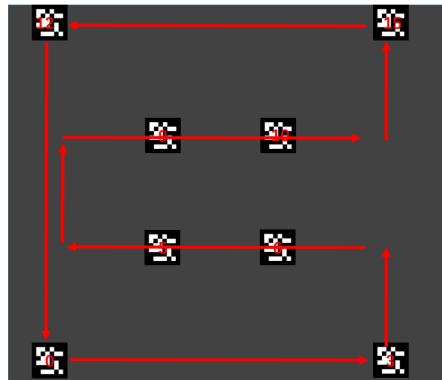


图 5.3 无人机自主飞行轨迹设计图

5.2.2 视觉定位系统实验结果分析

按照上述实验流程让无人机进行自主飞行，可以直接获得视觉算法生成的地图以及无人机在飞行的过程中生成的三维坐标，根据这些数据可以对视觉算法关于无人机自主定位的效果进行定量的分析，本文主要从生成地图的精度，无人机三维坐标的精度以及轨迹精度三个方面进行定量分析。

首先对于地图精度，根据图 5.1(b)生成的地图，通过对地图的解析，可以获取每一个二维码的三维位置坐标，如表 5.1 所示。针对地图的精度，可以提出以

表 5.1 二维码标志位置实际测量值

序号	X	Y	Z
0	-2.63	1.32	7.86
3	-14.69	3.88	7.80
5	-6.05	6.34	7.45
6	-9.83	-9.83	7.45
9	-5.13	10.44	7.30
10	-9.17	11.22	7.24
12	-0.33	13.56	7.34
15	-12.25	16.15	7.27

下两个判断指标：

1. 任意两个二维码之间距离的测量值和理论值的误差比较；
2. 所有二维码是否在同一个坐标平面。

针对指标 1，计算得到表 5.2，经过计算得到地图中二维码的平均误差精度为 3.1%。

针对指标 2，绘制出每个二维码在同一个坐标平面的误差情况，如图 5.4 所示，平均误差为 0.18m，可以认定所有二维码基本都在同一水平面内。

表 5.2 二维码位置误差

ID1	ID2	实际值	理论值	误差率
0	3	12.33	12	2.75 %
0	5	6.07	5.65	7.43%
0	6	9.26	8.94	3.58%
0	9	9.46	8.94	5.82%
0	10	11.86	11.31	4.86%
0	12	12.45	12	3.75%
0	15	17.67	16.97	4.12%
3	5	8.99	8.94	0.56%
3	6	5.85	5.65	3.54%
3	9	11.6	11.31	2.56%
3	10	9.18	8.94	2.68%
3	12	17.32	16.97	2.06%
3	15	12.51	12	4.25%
5	6	3.87	4	3.25%
5	9	4.21	4	5.25%
5	10	5.79	5.65	2.48%
5	12	9.21	8.94	3.02%
5	15	11.61	11.31	2.65%
6	9	5.75	5.65	1.77%
6	10	4.13	4	3.25%
6	12	11.47	11.31	1.41%
6	15	9.33	8.94	4.36%
9	10	4.11	4	2.75%
9	12	5.72	5.65	1.24%
9	15	9.12	8.94	2.01%
10	12	9.15	8.94	2.35%
10	15	5.81	5.65	2.83%
12	15	12.2	12	1.67%

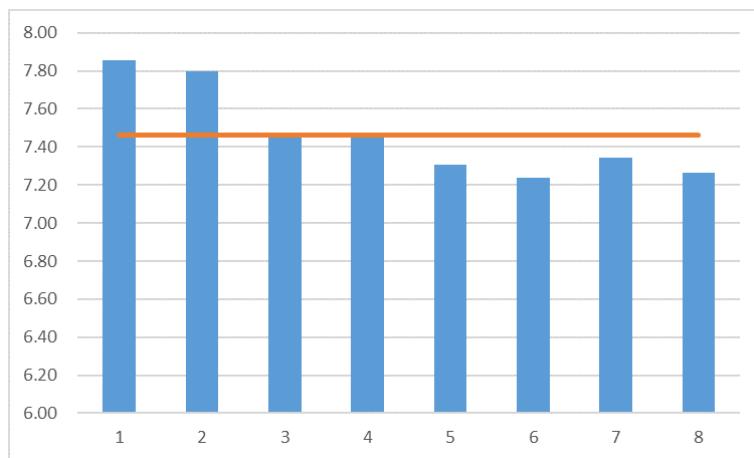


图 5.4 二维码 Z 方向数据

其次对于无人机三维坐标的精度，以无人机自带 GPS 测定出的坐标为真值，和视觉算法计算出来的位置信息进行对比，选择无人机 800 帧的数据，在 X、Y、Z 三个方向得到的结果分别如图 5.5 所示，其中蓝色连线为视觉算法检测出的坐标值，红色连线为无人机 GPS 检测出的真实值。随后，计算真值和测量值之间的

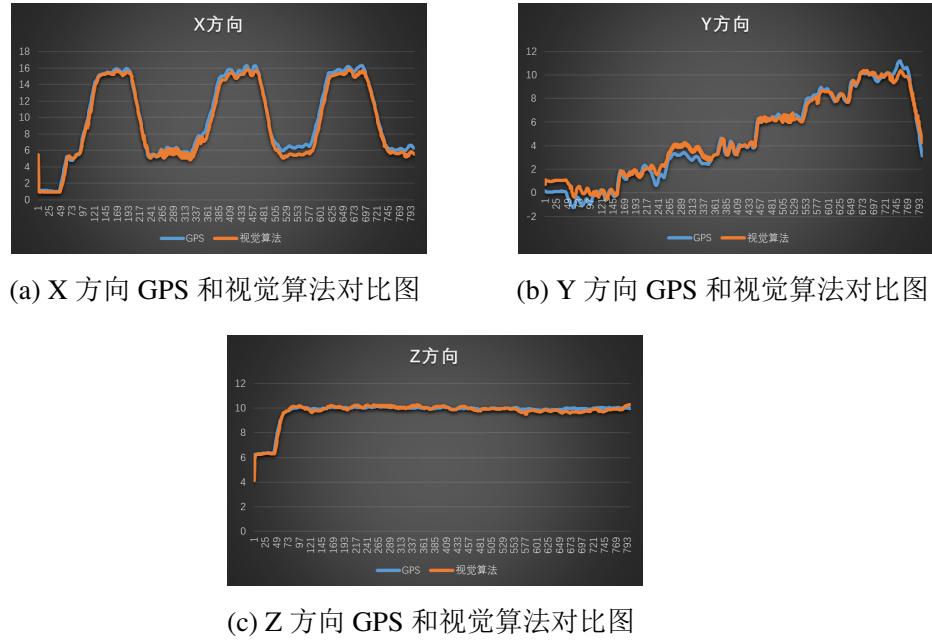


图 5.5 各方向 GPS 和视觉算法对比图

误差，在 X、Y、Z 方向分别得到结果如图 5.6 所示，对于 X、Y、Z 三个方向分别可以得到距离误差为 0.22m、0.37m、0.107m。

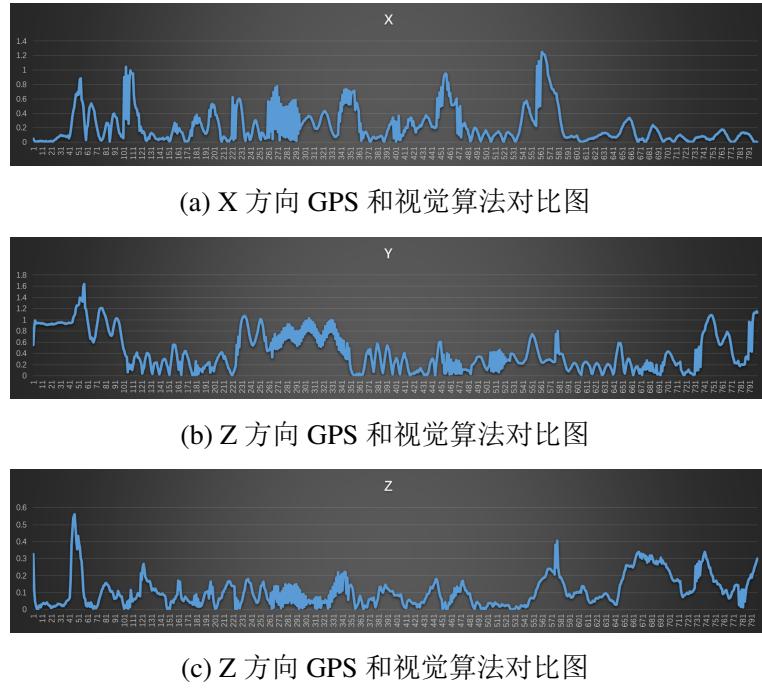


图 5.6 各方向 GPS 和视觉算法误差对比图

最后，对无人机的轨迹精度进行对比。无人机在自主飞行过程中能够产生实

时的相对于真实世界坐标系的定位信息，将其与 GPS 生成的定位信息进行对比，如图 5.7 所示。

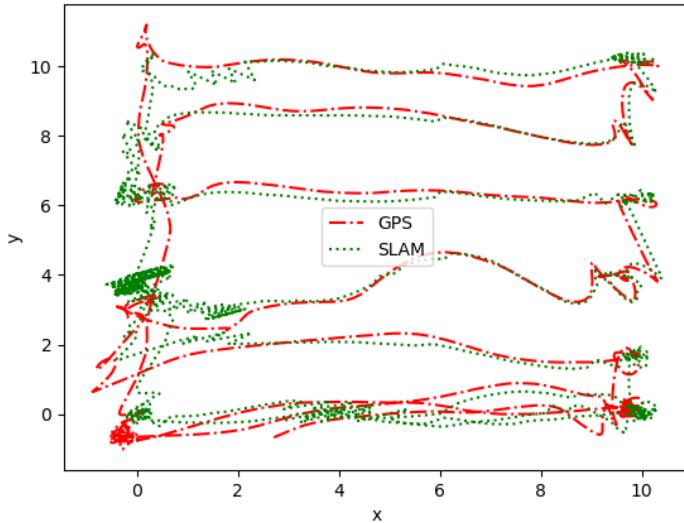


图 5.7 无人机轨迹对比示意图

对于无人机的直飞路线，误差较小，GPS 和视觉算法测算出的轨迹基本吻合，但在转弯方向改变的区间，两者之间的相对误差则较大，考虑其原因为在转弯处所设计的循迹点相对比较稠密，导致在该区域内无人机需要改变的方向更大，视觉算法在测算时由于方法振动的缘故产生较大的误差。

5.3 三维重建系统测试

在本文 3.2.3 节中，提出了三维重建存在的一些问题，包括对场景进行三维重建时，匹配耗时严重和点云准确性较低，甚至无法重建出三维点云，本实验将结合 3.3 节提出的方法对三维重建进行改进。

5.3.1 匹配实效性测试

通过 3.3.1 节对输入图像匹配模式的分析，本实验共选择 150 张图像（图片选择过少的话，各匹配模式之间的耗时差异会过小）作为输入图片，每张图像的大小为 960*544。各匹配模式包括完全匹配，序列匹配，空间匹配，传递匹配和自定义匹配，对于空间匹配，在对室内堆体三维重建时，难以获取准确的空间位置信息，本实验将排除该匹配方式的对比。

其中自定义匹配以实时运行的 SLAM 匹配信息作为匹配结果。SLAM 处理代码如 5.1 下：

Listing 5.1 SLAM 处理数据集 C++ 实现

```

1 //SLAM 处理数据集
2
3 if (mpCurrentKeyFrame->mnId > 4)
4 {
5
6     std::vector<KeyFrame*> cos =
7         mpCurrentKeyFrame->GetBestCovisibilityKeyFrames(covisibleNum);
8     cv::Mat rotation = mpCurrentKeyFrame->GetRotation();
9     cv::Mat translation = mpCurrentKeyFrame->GetTranslation();
10
11    Eigen::Matrix3d r2 = Converter::toMatrix3d(rotation);
12    Eigen::Quaterniond q2(r2);
13
14    fs1 << mpCurrentKeyFrame->mnFrameId << ",";
15    for (int k = 0; k < cos.size(); k++)
16    {
17        fs1 << cos[k]->mnFrameId << ",";
18    }
19
20    fs1 << "|" <<
21    q2.w() << "," << q2.x() << "," << q2.y() << "," << q2.z() << ","
22    << translation.at<float>(0) << ","
23    << translation.at<float>(1) << ","
24    << translation.at<float>(2)
25    << std::endl;
26}

```

对任意一帧图像，在 SLAM 处理后只处理并保存关联关键帧大于 4 帧的图像，依次获取与该帧匹配度最高的 4 帧和该帧所对应的位姿 (R, t)，结果如表 5.3 所示。各个匹配模式之间的耗时情况和匹配准确度如表 5.4 所示。由表可以分析得

表 5.3 SLAM 处理结果

当前帧	匹配帧集合	位姿
4	[5 3 2 1]	[R:0.989 -0.008 -0.137 0.053 t:-0.00 0.017 -0.033]
3	[4 5 2 1]	[R:0.989 -0.031 -0.163 0.056 t:-0.001 0.015 -0.030]
6	[5 7 4 8]	[R:0.987 -0.019 -0.131 0.090 t:-0.018 0.016 -0.064]
2	[3 4 5 1]	[R:0.981 -0.018 -0.184 0.057 t:0.000 0.015 -0.025]
8	[9 7 10 6]	[R:0.989 -0.039 -0.111 0.079 t:-0.034 0.014 -0.104]
...
...
...
...
145	[147 146 148 149]	[R:0.978 -0.118 -0.148 0.083 t:0.026 0.030 0.117]
149	[148 147 146 145]	[R:0.983 -0.109 -0.114 0.083 t:0.0123 0.020 0.033]
146	[147 148 145 149]	[R:0.972 -0.148 -0.168 0.067 t:0.017 0.0334 0.099]
151	[4 5 152 3]	[R:0.989 -0.096 -0.099 0.047 t:0.001 0.010 -0.057]
154	[12 13 11 10]	[R:0.992 -0.080 -0.085 0.021 t:-0.030 0.004 -0.204]

表5.4 各匹配模式耗时与精度情况对比表

匹配模式	耗时 (/s)	匹配精度
完全匹配	504	精度较高
序列匹配	94	精度较高
传递匹配	7	精度较差
自定义匹配	实时得到结果	精度高

到,以SLAM实时运行的匹配结果作为三维重建的匹配输入,一方面可以节省下匹配所耗费的时间,另外SLAM通过对时序图像的分析,可以得到更加准确的匹配信息。

5.3.2 点云精度测试

在面对部分重复度高,表面问题贫瘠的场景中,三维重建往往难以生成有效的点云,按照第3.3.2节提出的方法可以以SLAM生成的keyFrame Database等信息作为先验知识来改善三维重建的结果。本实验考虑到地下车库场景重复度高,且存在反光现象,选择了地下车库作为实验场景,按照环形有闭环的路径采集了一系列图像,部分图像如5.8所示。

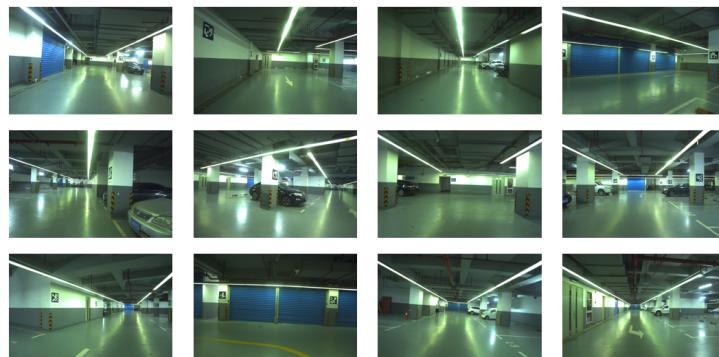


图5.8 地下车库环形图像

分别以传统方法和改进后的方法对其进行三维重建,结果分别如图5.9(a)和5.9(b)所示。对于普通三维重建方法而言,地下车库场景重建场景点云难度较大,因为场景中的场景重复度高,匹配精度较低,且反光场景的加入又会导致相机的位姿估计出现偏差,三维重建无法生成有效的点云结果。因此需要结合SLAM的结果来作为三维重建的先验知识,提供更多丰富的匹配约束以及更加精准的相机位姿,可以重建出较好的结果。

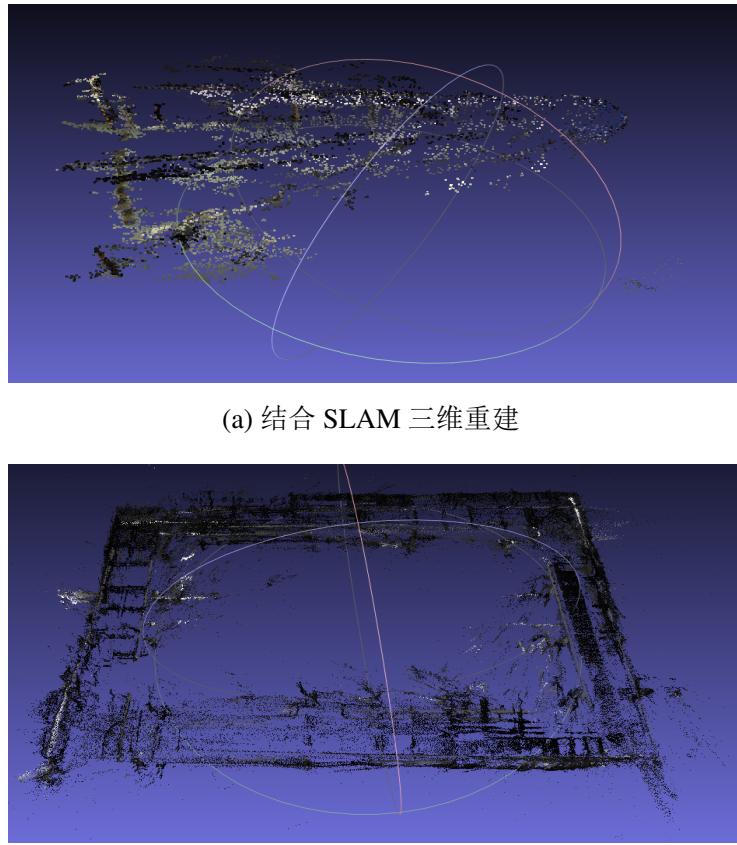


图 5.9 地下车库三维重建对比图

5.4 堆体体积测量系统测试

在第 4 章，本文提出了一种基于纯视觉方法来测量堆体体积的方法，基于对实验场景进行的三维重建结果，整个实验步骤主要包括以下四个方面：

1. 解析水平面方程
2. 估计场景实际尺度
3. 点云提纯
4. 计算三维点云的体积。

在测量堆体场景体积之前，需要收集该场景的连续视频帧以获取其三维重建的结果，考虑到稠密点云的点集数量过大，在遍历和查询时都会比较耗时，且稀疏点云也包含了每个特征点的坐标信息和相机的位姿信息，后续在解析水平面方程和估计场景实际尺度时选择稀疏点云作为分析对象。

5.4.1 解析水平面方程

1. 场景布置：在待测堆体场景中需要布置多个不同的二维码，一方面提高场景三维重建的稳定性和准确性，另外进一步为后续水平面方程的解析提供数据。在布置场景时，需要注意两个问题：1) 布置的二维码的所有下边沿都位于同一

条直线上；2) 应该尽可能将所有 Aruco 的二维码的下边沿都与待求水平面贴合，以保证水平面方程求解的准确性，若无法实现水平面的贴合要求，则需要进一步保证所有下边沿都位于同一水平面上，那么对计算得到的水平面进行空间变换即可得到真实水平面方程。

2. 数据收集：可以通过单目相机对堆体场景进行连续采集，在采集视频的过程中需要保证大部分图像帧中都能够采集到完整的二维码，所有的采集结果如图 5.10 所示。



图 5.10 堆体场景图像序列

3. 三维重建：可以直接将上述这些包含二维码的图像序列作为三维重建的输入，以获取堆体场景的三维点云，重建的结果如图 5.11 所示。

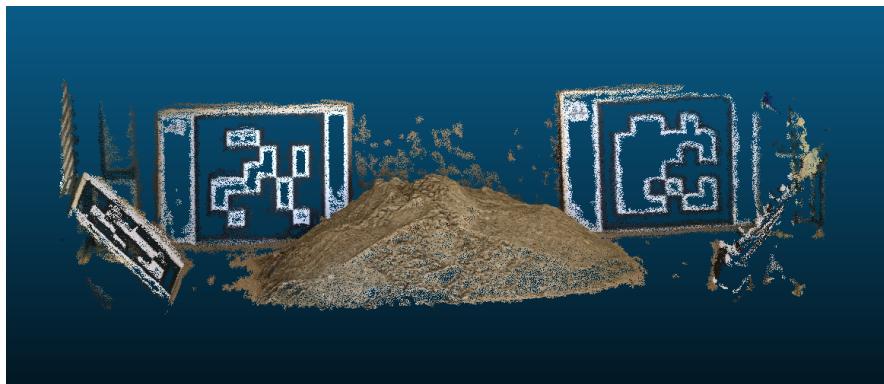


图 5.11 三维重建点云结果示意图

4. 按照 4.2.1 节的方式获取 2D 图像中的角点坐标和对应的三维重建的 3D 坐标，部分对应关系如表 5.6 所示。其中最近的 2D 坐标代表通过三维重建获得特征点提取时的最近坐标，当 3D 为 -1 时，则代表三维重建中的点云没有和该 2D 图像相对应的 3D 点。

5. 添加方程：根据 4.2.3 小节提出的方法，通过上述三维点可以计算出平面方程的参数，可以在原本缺失水平面的点云中按照解析方程添加点云即可，结果如图 5.12 所示，其中红色点云即为水平面。

表 5.5 2D 坐标和 3D 坐标关系对应表

图像名称	2D 坐标	最近点 2D 坐标	3D 坐标
1613.jpg	[724, 344]	[720.478, 343.642]	-1
	[512, 338]	[514.977, 337.341]	[1.75459, -1.35326, 8.97379]
1641.jpg	[901, 327]	[902.283, 326.125]	[3.75985, -1.35022, 8.76246]
	[691, 305]	[686.783, 313.456]	[1.74998, -1.29991, 8.97196]
1631.jpg	[822, 342]	[825.233, 345.432]	[3.75985, -1.35022, 8.76246]
	[615, 324]	[607.253, 323.576]	[1.69161, -1.38207, 8.97119]
1561.jpg	[652, 277]	[649.351, 274.972]	[3.75985, -1.35022, 8.76246]
	[453, 298]	[455.467, 297.046]	[1.75459, -1.35326, 8.97379]
1897.jpg	[775, 283]	[775.325, 282.267]	[3.75985, -1.35022, 8.76246]
	[607, 276]	[607.673, 276.95]	[1.75459, -1.35326, 8.97379]
1897.jpg	[457, 270]	[461.465, 266.988]	-1
	[295, 267]	[296.426, 266.952]	[-2.24158, -1.3894, 9.26188]
1911.jpg	[783, 278]	[780.128, 273.717]	[3.75985, -1.35022, 8.76246]
	[615, 272]	[615.711, 273.065]	[1.75459, -1.35326, 8.97379]
1911.jpg	[464, 267]	[463.551, 266.358]	-1
	[302, 265]	[303.311, 263.918]	[-2.24158, -1.3894, 9.26188]

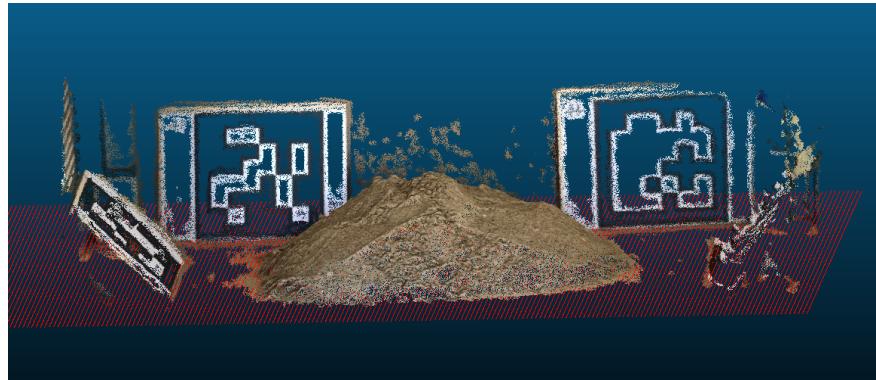


图 5.12 包含水平面的三维重建结果示意图

5.4.2 估计堆体实际尺度

1. 获取绝对尺度: 根据图 5.13 所示, 分别可以获得在 ID = 43 的二维码坐标系下的相机位姿, 如表 5.6 前两行, 可以计算出在这两帧之间相机移动的绝对距离为 0.464m。

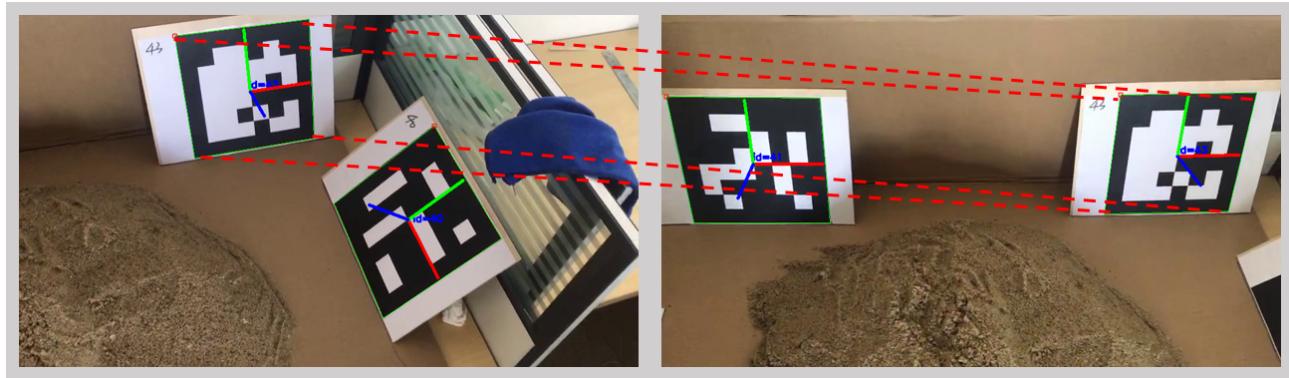


图 5.13 两帧之间获取绝对尺度

2. 获取相对尺度：从三维重建的结果中提取出这两帧在参考坐标系下的相机位姿，如表 5.6 后两行，可以计算出这在这两帧之间相机移动的相对距离为 4.924。

表 5.6 2D 坐标和 3D 坐标关系对应表

序号	T_x	T_y	T_z	距离
1027.jpg	-0.127325	-0.172455	0.980864	
1167.jpg	0.324499	-0.0655995	0.969803	0.464
1027.jpg	-6.06873	1.87705	-2.13389	
1167.jpg	-1.20871	2.09888	-1.37369	4.924

3. 估计尺度：按照公式 4.3 通过多组数据优化估计出堆体三维重建和实际堆体之间的尺度因子大小为 $K = 10.61$ 。

5.4.3 堆体体积估计

针对体积测量实验，本文拟采用如下方式验证所提方法是否能够测量出堆体体积，以及验证测量和测量稳定性。

首先使用测量仪器检测出待测堆体的实际体积真值，随后将堆体分别摆放成单峰堆体，双峰堆体，多峰堆体的形式，如图 5.14 按照第 4 章的流程，通过验证三次实验测量值和真值之间的误差衡量视觉算法的精确度，以及对三次测量值之间的数值进行比较，衡量视觉算法测量的稳定性。具体步骤如下：

1. 将某一堆体放置在水平面上，并在堆体周围放置 5 个二维码以估计尺度和水平面方程。
2. 对上述场景以视频的形式进行图像采集，提取视频中的图像，将数据集的数量控制在 200 张左右，且每张图像的分辨率在 50 万左右。
3. 对上述采集到的图像进行二维码检测与识别，记录下每张图像中的二维码 ID 以及角点坐标。

4. 对上述采集到的图像进行稀疏重建，结果如图 5.15所示，记录下稀疏点云中的 3D 点和每张图像角点之间的映射关系。
 5. 按照 4.2估计堆体出水平面方程。
 6. 按照 4.4.3估计出堆体的尺度大小。
 7. 对稀疏点云进行稠密重建，结果如图 5.16所示，并将所求解出的水平面方程添加至稠密点云中。
 8. 按照 4.4估计出当前堆体的体积大小。
 9. 将单个堆体分别设计成 2 个堆体和 3 个堆体的情况，重复步骤 2 8，计算不同场景对应的体积。
 - 10 根据结果定量分析上述视觉算法测量堆体的准确性和稳定性。
- 3 次实验结果如表 5.7所示。

表 5.7 体积测量结果

批次	图像数	点个数	尺度 K	水平面方程	计算耗时 (s)	堆体体积 (cm ³)	测量精度
1	186	110149	1.945	-0.0259x-0.0331y- 0.0873z+1=0	712	13221.4	1.7%
2	201	86403	8.143	0.0068x-0.0299y- 0.0620z+1=0	675	12758.3	1.8%
3	215	83196	3.522	0.0119x-0.0254y- 0.0683z+1=0	663	12830.2	1.3%

对上述沙堆 使用仪器测量出的体积真值为 13000cm^3 ，通过三次实验， 分别得出测量单峰堆体， 双峰堆体， 多峰堆体的体积为 $13221.4\text{cm}^3, 12758.3\text{cm}^3, 12830.2\text{cm}^3$ ， 测量的误差可以控制在 2% 以内， 且三次测量值接近， 测量稳定。



(a) 单峰堆体场景预览图



(b) 双峰堆体场景预览图



(c) 多峰堆体场景预览图

图 5.14 不同堆体场景预览图



(a) 单峰堆体场景稀疏点云图

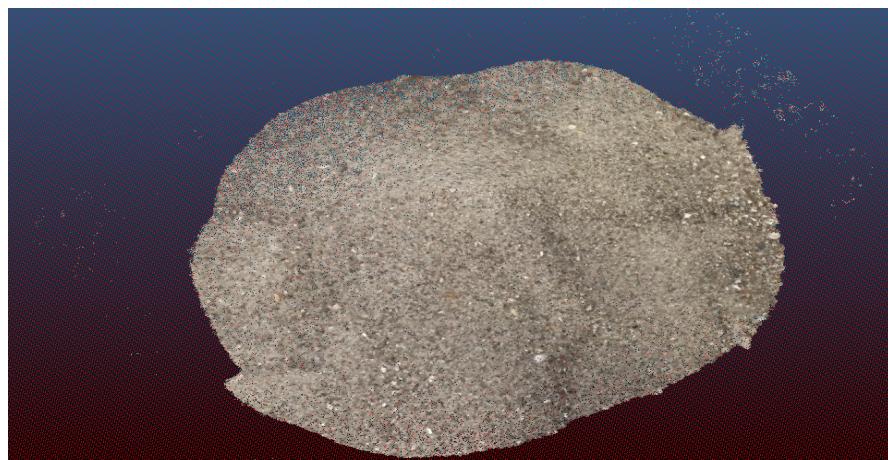


(b) 双峰堆体场景稀疏点云图



(c) 多峰堆体场景稀疏点云图

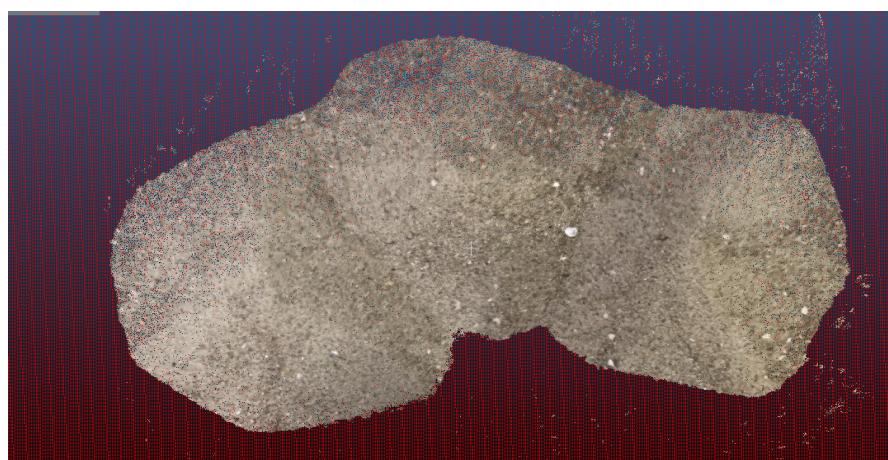
图 5.15 不同堆体场景稀疏点云示意图



(a) 单峰堆体场景稠密点云图



(b) 双峰堆体场景稠密点云图



(c) 多峰堆体场景稠密点云图

图 5.16 不同堆体场景稠密点云示意图

5.5 本章小结

本章分模块测试了无人机在非 GPS 场景下依靠纯视觉进行定位的效果与精度，以及改进后的三维重建的效果和基于纯视觉的堆体体积测量算法的准确度和稳定性。

第6章 总结和展望

6.1 全文总结

本文提出了基于纯视觉的无人机自主定位方法，能够使无人机在无 GPS 环境中围绕堆体进行自主飞行；本文基于传统三维重建方法，在重建实效性和精度方面提出了改进优化方法；本文提出了基于纯视觉的堆体体积测量方法。本文提出了面向无人机自主堆体体积测量的视觉定位及重建方法，可以构成一套完整的自动化测量系统。本文的主要研究成果与总结如下：

1. 实现无人机自主定位。针对部分封闭，无 GPS 的环境中，构建一套无人机自主定位系统，通过结合二维码视觉标签的纯视觉方案，为无人机提供具有真实尺度的位姿数据，且坐标系统可以完全与真实世界对齐，使得无人机可以按照规定路线进行有效的自主飞行。结合二维码的视觉 SLAM 可以给无人机提供一个更加精度的定位数据，添加地图信息使整个系统更加稳定。
2. 改进优化三维重建流程。使用结合视觉 SLAM 的方法对三维重建进行优化改良，三维重建本身可以较为简便地通过连续图像序列获取到场景的三维点云模型，但存在很严重的耗时问题以及点云精度低的问题。针对上述问题，提出了以 SLAM 对具备时序信息的连续帧进行处理，获取包含 KeyFrame DataBase，匹配关系，相机位姿和环境地图等信息，作为先验知识提供给三维重建系统，提升原本系统的建模精度和速度。
3. 实现视觉方法求解堆体体积。本文提出基于纯视觉的估计方法代替传统测量仪器来对堆体进行体积测量，测量过程主要包括对堆体点云尺度的估计，堆体水平面方程的求解以及堆体体积的数学转化和计算。本方案在测量堆体体积时有着较好的准确性和稳定性，且流程简易，能够在多种场景下复用。
4. 系统分模块测试。本文对无人机定位，三维重建改进方法，堆体体积测量三个方面进行分模块测试。其中无人机自主定位的精度可以控制在 51.1% 以内，高度方向的误差在 0.2m 以内；改进后的三维重建点云模型在精度方面有了明显的提升，且可以很好解决实效性的问题；基于纯视觉的体积计算，整体测量误差精度可以控制 2% 以内，且多次测量保证稳定性。

6.2 未来工作展望

限于时间、能力，本文的研究仍存在一些问题以及尚需改进的部分，有待将来有更加深入的研究，以下结合本文的研究，就几个典型问题作出展望。

1. 点云精度定性评价指标。本文对点云精度的评价指标多为定性处理，可以提出一种针对点云精度定量化分析的评价指标。
2. 系统联合。本文提出的三个模块系统的联合较为初步，后续可以结合实际工程使得三个模块融合为完整系统，使得操作更加简易有效稳定。
3. 堆体多场景测试。本文选用的堆体的对象为煤堆和沙堆，后续可以进一步对工农业生产中的多种类别堆体进行验证与测试。

致谢

在此论文成稿之际，谨向我敬爱的导师岳继光教授和董延超副教授以最真诚的感谢与敬意！本文的选题、论文布局在岳老师的指导下完成，论文的创新方向以及细节实施均在董老师的负责下完成。岳老师治学严谨，知识渊博，为人正直谦虚，与他的每一次谈话都让我受益匪浅。我从老师身上学到的不仅仅是知识，还有很多做人的道理，这些令我受用终身。董老师有着丰富的项目经历和理论知识，能够快速定位研究中出现的问题以及痛点，合理安排我的研究方向以及时间，并训练我问题的凝练和解决方法，这对我处理实际问题的能力有了极大的提高，为日后进入工作打下了坚实的基础。同时，岳老师和董老师不仅在学术上给我耐心的指导，还关心我的日常生活，让我在同济的三年研究生生涯过得顺利而又充实，再次向他们表示衷心的感谢。

感谢课题组的苏永清老师、吴继伟老师、李荣艳老师等，谢谢各位老师在论文开题时做出的指导。感谢师兄侯培鑫博士在学习和生活中的帮助，师兄极聪慧，知识面广，介绍给我非常多实用的工具以及技能，且师兄的高标准促使我更加规范地完成每一件事。在侯博的主导下，我们一同完成并维护大论文的 L^AT_EX 模板，在此过程中我受益匪浅，再次感谢侯博，希望师兄在今后的工作生活中一切顺利。感谢师兄王森博、徐晨剑以及施梁，三位师兄在我读研期间给予我非常多的指导和帮助，给我的研究生生活起到了榜样的作用，希望森博师兄顺利毕业，希望徐晨剑及施梁师兄工作顺利。感谢同门徐刚、乔琪和张爽在日常学习中的齐心同行，也祝愿你们在日后的工作中取得好成绩。感谢张鲲鹏师兄、刘志刚师兄、汪胤师兄、唐丹旭师兄、王栗博士、吴琛浩博士，用丰富的生活和学习经验，使我更快融入实验室集体。感谢师弟何士波，师妹林敏静在学术上的探讨和交流。感谢课题组成员，穆慧华、刘雪娇、刘金承、武新然、孙佳妮、李灵聪、冀玲玲、何洪志、寿佳鑫、王浩天、宁少淳带来的欢乐与帮助。此外，感谢室友闫智海，好友胡钱珊、刘梦琪等，在学术上的交流讨论和生活中的娱乐和关心。

最后，感谢我的父母，是他们一直鼓励我、支持我，使我有了不断克服困难，奋勇前进的动力。

何士波

2019 年 12 月于同济

参考文献

- [1] CADENA C, CARLONE L, CARRILLO H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J]. IEEE Transactions on robotics, 2016, 32(6): 1309-1332.
- [2] SE S, LOWE D, LITTLE J. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks[J]. The international Journal of robotics Research, 2002, 21(8): 735-758.
- [3] DAVISON A J. Real-time simultaneous localisation and mapping with a single camera.[C] // Iccv:vol. 3. 2003: 1403-1410.
- [4] DAVISON A J, REID I D, MOLTON N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007(6): 1052-1067.
- [5] 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.
- [6] WHELAN T, KAESS M, JOHANNSSON H, et al. Real-time large-scale dense RGB-D SLAM with volumetric fusion[J]. The International Journal of Robotics Research, 2015, 34(4-5): 598-626.
- [7] GIBBENS P W, DISSANAYAKE G M W M, DURRANT-WHYTE H F. A closed form solution to the single degree of freedom simultaneous localisation and map building (SLAM) problem[C] // Decision and Control, 2000. Proceedings of the 39th IEEE Conference on. 2000.
- [8] DURRANT-WHYTE H, BAILEY T. Simultaneous localization and mapping: part I[J]. IEEE robotics & automation magazine, 2006, 13(2): 99-110.
- [9] GRISETTI G, KUMMERLE R, STACHNISS C, et al. A tutorial on graph-based SLAM[J]. IEEE Intelligent Transportation Systems Magazine, 2010, 2(4): 31-43.
- [10] DISSANAYAKE G, HUANG S, WANG Z, et al. A review of recent developments in simultaneous localization and mapping[C] // 2011 6th International Conference on Industrial and Information Systems. 2011: 477-482.
- [11] SAEEDI S, TRENTINI M, SETO M, et al. Multiple-robot simultaneous localization and mapping: A review[J]. Journal of Field Robotics, 2016, 33(1): 3-46.
- [12] STACHNISS C, LEONARD J J, THRUN S. Simultaneous localization and mapping[G] // Springer Handbook of Robotics. Springer, 2016: 1153-1176.
- [13] SARKAR A, REIGER R, ROY S, et al. SLAM Using relational trees and semantics[C] // Advanced Materials Research:vol. 452. 2012: 648-653.
- [14] ZHANG G, LEE J H, LIM J, et al. Building a 3-D line-based map using stereo SLAM[J]. IEEE Transactions on Robotics, 2015, 31(6): 1364-1377.
- [15] MUR-ARTAL R, TARDÓS J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.
- [16] GAO X, WANG R, DEMMEL N, et al. LDSO: Direct sparse odometry with loop closure[C] // 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018: 2198-2204.
- [17] GÁLVEZ-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.

- [18] MUÑOZ-SALINAS R, MARÍN-JIMÉNEZ M J, MEDINA-CARNICER R. SPM-SLAM: Simultaneous localization and mapping with squared planar markers[J]. Pattern Recognition, 2019, 86: 156-171.
- [19] MUÑOZ-SALINAS R, MEDINA-CARNICER R. UcoSLAM: Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers[J]. ArXiv preprint arXiv:1902.03729, 2019.
- [20] FENG J, TENG Q, HE X, et al. Reconstruction of three-dimensional heterogeneous media from a single two-dimensional section via co-occurrence correlation function[J]. Computational Materials Science, 2018, 144: 181-192.
- [21] LIU Y, MA X, SUN X, et al. The application of 3D reconstruction in investigating the frontal sinus drainage pathway based on computer tomography data[J]. Lin chuang er bi yan hou tou jing wai ke za zhi= Journal of clinical otorhinolaryngology, head, and neck surgery, 2018, 32(3): 171-176.
- [22] RUPNIK E, PIERROT-DESEILLIGNY M, DELORME A. 3D reconstruction from multi-view VHR-satellite images in MicMac[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2018, 139: 201-211.
- [23] ZHANG M, CHAI M, WU H, et al. A data-driven approach to four-view image-based hair modeling.[J]. ACM Trans. Graph., 2017, 36(4): 156-1.
- [24] NAYAR S K, NAKAGAWA Y. Shape from focus[J]. IEEE Transactions on Pattern analysis and machine intelligence, 1994, 16(8): 824-831.
- [25] CHOI T S, ASIF M, YUN J. Three-dimensional shape recovery from focused image surface[C]//1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258):vol. 6. 1999: 3269-3272.
- [26] SRINIVASAN V, LIU H C, HALIOUA M. Automated phase-measuring profilometry of 3-D diffuse objects[J]. Applied optics, 1984, 23(18): 3105-3108.
- [27] 颜国霖, 林琳. 基于光栅相位法三维重构技术研究[J]. 机电技术, 2010, 33(5): 14-17.
- [28] 黄俊春, 章炯民. 基于彩色结构光的实时三维重建[J]. 计算机应用与软件, 2009, 26(12): 235-237.
- [29] IZQUIERDO M, SÁNCHEZ M, IBÁÑEZ A, et al. Sub-pixel measurement of 3D surfaces by laser scanning[J]. Sensors and Actuators A: Physical, 1999, 76(1-3): 1-8.
- [30] TAKEDA M, MUTOH K. Fourier transform profilometry for the automatic measurement of 3-D object shapes[J]. Applied optics, 1983, 22(24): 3977-3982.
- [31] MORAVEC H P. Techniques towards automatic visual obstacle avoidance[J].., 1977.
- [32] HARRIS C G, STEPHENS M, et al. A combined corner and edge detector.[C]// Alvey vision conference:vol. 1550. 1988: 10-5244.
- [33] SMITH S M, BRADY J M. SUSAN—a new approach to low level image processing[J]. International journal of computer vision, 1997, 23(1): 45-78.
- [34] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.
- [35] BAY H, ESS A, TUYTELAARS T, et al. Speeded-up robust features (SURF)[J]. Computer vision and image understanding, 2008, 110(3): 346-359.
- [36] GRAUMAN K, DARRELL T. Pyramid match hashing: Sub-linear time indexing over partial correspondences[C]//2007 IEEE Conference on Computer Vision and Pattern Recognition. 2007: 1-8.
- [37] MOUNT D, ARYA S. ANN: library for approximate nearest neighbor searching, 1998[C]// 2nd annual CGC workshop on computational geometry. 1998.
- [38] LONGUET-HIGGINS H C. A computer algorithm for reconstructing a scene from two projections[J]. Nature, 1981, 293(5828): 133.

- [39] LUONG Q T. Matrice fondamentale et autocalibration en vision par ordinateur[D]. Paris 11, 1992.
- [40] NI K, STEEDLY D, DELLAERT F. Out-of-core bundle adjustment for large-scale 3d reconstruction[C]//2007 IEEE 11th International Conference on Computer Vision. 2007: 1-8.
- [41] ENGELS C, STEWÉNIUS H, NISTÉR D. Bundle adjustment rules[J]. Photogrammetric computer vision, 2006, 2(2006).
- [42] LOURAKIS M I, ARGYROS A A. SBA: A software package for generic sparse bundle adjustment[J]. ACM Transactions on Mathematical Software (TOMS), 2009, 36(1): 2.
- [43] LHUILLIER M, QUAN L. A quasi-dense approach to surface reconstruction from uncalibrated images[J]. IEEE transactions on pattern analysis and machine intelligence, 2005, 27(3): 418-433.
- [44] FURUKAWA Y, PONCE J. Accurate, dense, and robust multiview stereopsis[J]. IEEE transactions on pattern analysis and machine intelligence, 2009, 32(8): 1362-1376.
- [45] CHANG R S, CHIU J H, CHEN F P, et al. A Parkinson' s disease measurement system using laser lines and a CMOS image sensor[J]. Sensors, 2011, 11(2): 1461-1475.
- [46] CIGADA A, MANCOSU F, MANZONI S, et al. Laser-triangulation device for in-line measurement of road texture at medium and high speed[J]. Mechanical Systems and Signal Processing, 2010, 24(7): 2225-2234.
- [47] WANG H, ZHANG D, HE L. Research on large open stockyard laser measurement methods[J]. Zhongguo Jiguang(Chinese Journal of Lasers), 2013, 40(5): 0508002-8.
- [48] CHANG D, LU H, MI W. Bulk terminal stockpile automatic modeling based on 3D scanning technology[C]//2010 International Conference on Future Information Technology and Management Engineering:vol. 1. 2010: 67-70.
- [49] YUNTAO Y, XINGSHU W, ZONGSHENG H, et al. Bias error compensation of laser gyroscope for autonomous measurement of ship hull angular deformation[J]. Chinese Journal of Scientific Instrument, 2014(12): 14.
- [50] SHI J, CAI Y, LIANG H, et al. Study on horizontal tank volume calculating method based on laser tomoscan[J]. J. Electr. Meas. Instrum, 2014, 28: 1021-6.
- [51] REN C, CHEN C, WANG M, et al. Research on a positioning correction method with inertial measurement in combination with distance[J]. Chin. J. Sci. Instrum, 2013, 34: 2513-9.
- [52] DONG Z, LONG Y, JIAN X, et al. Study on multi-cameras and structured-light vision system calibration approach study in large filed view measurement[J]. Chinese Journal of Scientific Instrument, 2015, 36(4): 903-912.
- [53] LIU Z, CHEN T. Distance measurement system based on binocular stereo vision[C]//2009 International Joint Conference on Artificial Intelligence. 2009: 456-459.
- [54] ROSÉN B G, BÅÄTH L, DIMKOVSKI Z. On variation of surface topography and robust product performance[C]//4th International Symposium on Test Automation and Instrumentation (ISTA' 2012), Aug. 8-11, Dandong, China. 2012.
- [55] MEILI Z, ZONGWEN B. Design of the laser displacement measurement system based on 2D-PSD[J]. Foreign Electronic Measurement Technology, 2015(2): 15.
- [56] SHI H, ZHANG J. Study on track gauge measurement system based on laser triangulation principle [J][J]. Chinese Journal of Scientific Instrument, 2013, 34(9): 1934-1940.
- [57] GUO J R, WANG Y L, LEI Y U. Resolution of object surface and method of surface areas and volumes calculation[J]. Journal of Shandong University of Technology, 2014.
- [58] ZHANG W, WU T. Remote 3D measurement & visualization system based on laser scanning & 3D reconstruction[C]//2010 Symposium on Photonics and Optoelectronics. 2010: 1-5.
- [59] BI Y, QI L, CHEN S, et al. Canopy volume measurement method based on point cloud data.[J]. Science & Technology Review, 2013, 31(27): 31-36.

-
- [60] 熊友辉, 李培生, 邹显宏, 等. 便携式激光盘煤系统原理及应用[J]. 中国电力, 2003, 36(6): 48-51.
 - [61] 张德津, 李必军, 何莉, 等. 基于多传感器集成的堆场激光测量技术应用[D]. 2012.
 - [62] 盛业华, 张卡, 张凯, 等. 地面三维激光扫描点云的多站数据无缝拼接[J]. 中国矿业大学学报, 2010, 39(2): 233-237.
 - [63] 张小虎, 欧建良, 苑云, 等. 投影轮廓线辅助下的堆场三维形貌摄影测量研究[J]. 光学学报, 2011.
 - [64] 王海波, 张德津, 何莉, 等. 大型露天料场激光测量方法研究[J]. 中国激光, 2013, 40(5): 0508002.

个人简历、在学期间发表的学术论文与研究成果

个人简历:

1994 年 4 月出生于河南省信阳市。

2013 年 9 月考入中南大学信息科学与工程学院自动化专业，2017 年 7 月本科毕业并获得工学学士学位。

2017 年 9 月考入同济大学控制工程专业系攻读工学学位至今。

发表论文:

- [1] Yang Y, Ren T L, Zhang L T, et al. Miniature microphone with silicon- based ferroelectric thin films. *Integrated Ferroelectrics*, 2003, 52:229-235. (SCI 收录, 检索号:758FZ.)

研究成果:

- [1] 任天令, 杨轶, 朱一平, 等. 硅基铁电微声学传感器畴极化区域控制和电极连接的方法: 中国, CN1602118A. (中国专利公开号.)