



重庆交通大学  
CHONGQING JIAOTONG UNIVERSITY

# 本科毕业设计（论文）

题目： 基于 ORB 特征的视觉定位与建图系统设计

学 院： 信息科学与工程学院

专 业： 通信工程

学 生 姓 名： 唐杨洋

学 号： 631907030623

指 导 教 师： 何立兰

评 阅 教 师： 蒲树祯

完 成 时 间： 2023 年 05 月 20 日

重庆交通大学

CHONGQING JIAOTONG UNIVERSITY

本人郑重声明：所提交的毕业设计（论文），是本人在导师指导下，独立进行研究工作所取得的成果。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文研究做出过重要贡献的个人和集体，均已在文中以明确方式标明。

作者签名（亲笔）： 年 月 日

本毕业设计(论文)作者完全了解学校有关保留、使用学位论文的规定,本科生在校攻读期间毕业设计(论文)工作的知识产权单位属重庆交通大学,同意学校保留并向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅;本人授权重庆交通大学可以将毕业设计(论文)的全部或部分内容编入有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存、汇编毕业设计(论文)。

导师签名（亲笔）：                      年        月        日

## 摘 要

本课题是基于 ORB 特征算法的即时定位与建图系统，主要实现基于稀疏地图点的定位与地图构建功能，旨在分析特征点法在 SLAM 中的应用，并根据实验结果改进算法，提升定位与建图的精度。本系统是基于 Android(单目模式)手机所采集的视频数据作为分析依据，实现以低成本的方式完成对系统的开发。

系统采用 Linux 操作系统为开发平台并以 C++语言为开发环境，结合图像处理库完成对系统的开发。

首先，需要使用规范标定板完成对相机参数的标定，使用正确参数的相机获取数据，根据 ORB 算法提取图像特征，获取到关键点，计算关键帧，系统根据相机采集到的数据和特征点信息计算出位置姿态，完成稀疏建图并以 GPS 作比照分析。

本系统对性能和精度进行了分析，得出系统在官方数据集运行后的绝对位姿误差、相对位姿误差，方向误差，分析系统运行的轨迹精确性，得出误差分布范围和主要频率，以及在不同相机模组下的系统精度，从而获知系统性能。

针对 ORB 特征点法在定位与建图功能的优缺点，本课题运用深度学习和三角化提出一种基于 PCL 的三维建图，旨在还原传感器周围的实际场景，通过三维重建和单目预测，可以有效分析周围的环境信息。同时，为了增加系统与使用者的交互性，在系统中扩展了增强现实的功能。

针对不同的环境进行实验，获取各种场景下系统的性能，包括室内、室外、复杂、简单等场景。测试对不同场景的应用能力，对轨迹数据进行可视化分析，得出系统性能优良的结论，误差在可接受范围内。

**关键词：**ORB 特征点；SLAM；深度学习；三维建图

## • Design of Visual Positioning and Mapping System Based on ORB Features

### Abstract

This project designs a real-time positioning and mapping system based on the ORB feature algorithm. It focuses on sparse point-based positioning and map construction to analyze the application of feature point methods in SLAM. The system improves accuracy through algorithm enhancements based on experimental results. It utilizes video data from Android smartphones (monocular mode) for analysis, enabling cost-effective development.

The system is developed on Linux using C++ and image processing libraries. It involves camera parameter calibration using a standardized calibration board, extracting image features with the ORB algorithm, computing keyframes, and calculating the camera's position and orientation using collected data and feature points. Sparse mapping is performed, and a GPS comparison analysis is conducted.

The system conducts performance and accuracy analysis by evaluating absolute and relative pose errors, direction errors, trajectory accuracy, error distribution, main frequency analysis, and system accuracy under different camera modules based on official dataset runs.

To address ORB's pros and cons, this project proposes a PCL-based 3D mapping approach using deep learning and triangulation. It reconstructs the sensor's surroundings and analyzes the environment through 3D reconstruction and monocular prediction. Augmented reality functionality is also included for enhanced user interaction.

Experiments are conducted in different environments, including indoor, outdoor, complex, and simple scenarios, to evaluate the system's performance and its ability to handle various situations. Trajectory data is visualized and analyzed, leading to the conclusion that the system exhibits excellent performance with errors within an acceptable range.

**Key Words:** ORB feature point; SLAM; deep learning; Three-dimensional drawing

## 目 录

摘    要 .....	II
Abstract.....	III
第 1 章  绪论 .....	1
1.1  研究背景及意义 .....	1
1.2  国内外研究现状 .....	2
1.2.1  国内外计算机视觉研究现状 .....	2
1.2.2  国内外计算机视觉在导航定位的发展分析 .....	3
1.3  系统主要内容以及结构 .....	3
1.4  本章小结 .....	4
第 2 章  系统环境分析 .....	5
2.1  系统运行平台 .....	5
2.1.1  Linux 操作系统.....	5
2.1.2  Ubuntu 系统 .....	5
2.1.3  双系统分析 .....	6
2.2  依赖环境 .....	8
2.2.1  主要三方库 .....	8
2.2.2  PCL 库与 VTK 库.....	10
2.3  软硬件参数 .....	11
2.4  本章小结 .....	13
第 3 章  关键技术概述 .....	14
3.1  视觉 SLAM 技术概述 .....	14
3.1.1  ORB-SLAM 原理.....	14
3.2  ORB 算法原理 .....	16
3.2.1  ORB 特征及特征点 .....	16
3.2.2  ORB 特征点提取 .....	16
3.3  单目相机成像原理与相机标定 .....	19
3.3.1  单目相机成像原理 .....	19
3.3.2  相机标定原理 .....	21
3.4  误差分析原理 .....	22
3.5  本章小结 .....	24
第 4 章  系统功能设计与实现 .....	25

4.1	系统框架结构 .....	25
4.2	相机标定 .....	26
4.3	特征点提取 .....	29
4.4	单目深度估计与三维重建 .....	31
4.4.1	Zoedepth 单目深度估计 .....	31
4.4.2	三维重建 .....	32
4.4.3	位姿三角化与增强实现 .....	33
4.5	误差分析 .....	34
4.6	本章小结 .....	35
第 5 章	系统测试 .....	36
5.1	官方数据集结果分析 .....	36
5.1.1	EuRoC 数据集 .....	36
5.1.2	结果评估 .....	36
5.1.3	位姿计算 .....	39
5.2	室内数据集结果分析 .....	45
5.3	回环检测结果分析 .....	47
5.4	轨道交通三号线结果分析 .....	49
5.5	三维重建分析 .....	51
5.6	三角化与 AR 模型 .....	54
5.7	本章小结 .....	56
第 6 章	总结与展望 .....	57
6.1	本文工作总结 .....	57
6.2	展望 .....	58
致 谢	.....	59
参 考 文 献	.....	60

## 第 1 章 绪论

### 1.1 研究背景及意义

随着我国经济的快速发展和国际地位的提升，对于高精度、高可靠性的导航定位技术的需求日益增长。导航定位技术在军事、航空航天、交通运输、智能交通、物流、测绘等领域具有重要应用，因此对于国家的发展和安危至关重要。从以往需要借助他国的定位系统来实现导航，到如今我国自主研发定位系统并给他国提供技术支持，这是一个从无到有的质的飞跃，而在如今这个蓬勃发展的时代，更是需要不断革故鼎新，牢牢把导航定位技术的发展命运把握在自己手中。

如文献[1]所述，导航在人们的日常生活生产起着至关重要的作用。从古至今，回顾导航技术的发展历程，这不仅仅是人类文明的发展，更是人们在科学技术上的智慧结晶。目前各类导航系统的研究几乎都是在各类研究院的指导下进行，而且研究成本较大，定位的应用场景和功能有限。于是利用现行的导航定位技术系统对小工程应用就具有了很大局限性。考虑研究一种新兴的导航定位方法，实现导航定位技术在更多领域以低成本、更简便的应用。

而计算机技术的发展，当前社会已经逐渐步入了数字化，因此带动了计算机视觉、人工智能、神经网络等高新技术领域的进步，同时也给传统技术领域带来了新的发展机会<sup>[2]</sup>。将导航定位技术与计算机视觉结合，是一个具有重要意义的方向，不仅可以推动计算机视觉在工业领域的应用，同时也为导航技术的发展指出一条全新的方向<sup>[3]</sup>。

因此，选择一种较好的计算机视觉技术运用到导航定位技术的性能改善上至关重要。而 SLAM，即自主定位与地图构建技术，能够很好的应用到导航定位系统中。

在计算机科学和导航领域，定位通常指的是通过使用传感器或其他相关信号设备来确定移动设备与构建好的环境地图进行匹配，确定机器人在当前地图中的位置，在定位过程中只有当地图提供准确的环境信息时，定位系统才能正确地将传感器数据与地图进行匹配，从而准确地确定目标物体的位置。而地图构建是将当前传感器感知到的环境信息转化为地图的过程，这个地图是以机器人当前位置为参考的，因此只有在定位准确的情况下，构建的地图才能与真实环境相匹配。基于 SLAM 的这种特性，可以将之与导航定位的实时运行相结合，通过使用传感器实时采集数据，采用 SLAM 技

术对采集到的数据进行处理，从而获取运动的实时数据，包括行驶轨迹、行驶速度、行驶的姿态角度等信息参数，以到达获知传感器周围的环境信息。

在文献[4]中论述了 ORB-SLAM3 系统。作为 SLAM 技术的一类，这种系统不管是在大场景或者小场景，室内室外都能实时运行，而且有很高的精确度。

基于此，本课题将会把 SLAM 技术与导航定位技术相结合，通过对传感器的测速与定位，提供一种全新的定位测速方法，降低导航成本、实现在小场景的导航应用。

## 1.2 国内外研究现状

### 1.2.1 国内外计算机视觉研究现状

室外定位技术已经发展得较为成熟，目前比较流行的就是美国的 GPS 系统 (Global Positioning System)<sup>[5]</sup>，而国内的定位技术也取得关键进步，我国的 BDS 系统 (北斗导航系统)<sup>[6]</sup>。而上述定位系统多适用于室外定位，对于室内相对封闭的环境而言，还没有成熟的定位系统，针对这种情况，出现了许多参差不齐的室内定位技术，而 SLAM 从中脱颖而出，成为当时乃至目前所公认的一种前沿定位技术。

SLAM 作为一种计算机视觉技术，其发展与计算机视觉的发展密不可分。在 20 世纪 60 年代中，麻省理工学院的 Roberts 在他的论文中介绍了一种关于计算机理论的方法，该方法能够从二维图像中推断出物体的三维信息，这一突破开辟了机器视觉领域以解析物体三维场景为目标的理论与实践研究<sup>[7]</sup>。到了 20 世纪 70 年代中，Marr 领导的研究学者们建立了体系化的计算机视觉理论，对机器视觉研究发挥了重要的理论基础作用。这一理论是以二维图像信息为基础，实现了从中恢复出物体的三维几何形状<sup>[8]</sup>。因而，在国外的发达国家计算机视觉技术发展的相对较快，相应的应用领域也较为宽广。

由于计算机的普及在国内比较晚，因此，计算机视觉技术在国内算是一种全新的技术，甚至直至目前，我国还有许多与计算机相关的技术任然需要依靠国外。但是，国内的计算机视觉技术也在近几年得到巨大的提升，在我国的大数据技术的支持下，目前的计算机视觉技术主要运用于人工智能、图像识别、工业产品检测等方面，在定位方面的应用较少，但我国的定位技术日趋完善，这也为将来计算机视觉在定位系统方面的应用埋下伏笔<sup>[9]</sup>。



### 1.2.2 国内外计算机视觉在导航定位的发展分析

计算机视觉在导航定位领域的发展较晚，在 21 世纪初期，国外也属于刚开始起步研究阶段，研究主要集中在理论基础和算法研究上，研究者们主要依赖于传统的匹配算法和特征提取法，并逐步应用到基础的导航定位问题。随着深度学习技术的发展，国外的发达国家开始关注深度学习在计算机视觉中的应用，也将之与视觉导航结合，深度学习算法在图像处理和物体识别方面取得了显著进展，为导航定位问题的解决提供了新的方法和工具。同阶段的 2009 年，Sim 等人提出了 SLAM 视觉导航的方法<sup>[9-10]</sup>。视觉 SLAM 技术也得到了广泛关注，科研人员开始探索使用视觉信息进行实时定位和地图构建，通过结合视觉传感器（如相机）和惯性传感器（如陀螺仪和加速度计）进行导航定位。而如今，计算机视觉在导航定位领域的研究和应用取得了重大进展，由于计算机硬件性能的提升和深度学习的发展，导航定位问题在诸多领域得到了普遍的应用。

我国的研究在这方面的稍落后于发达国家。在早期，主要集中在室内导航和机器人导航等方面，通过摄像头或激光雷达等传感器进行数据采集和图像处理，实现简单的定位和导航功能。在全球兴起计算机技术浪潮后，我国的研究者也积极探索计算机视觉和导航定位结合并取得相应的成果。如一些大型商场、展览馆和机场等场所开始应用计算机视觉技术，结合室内地图和导航算法，提供室内定位和导航服务。

近几年，计算机性能得到较大改善，图像视频数据的处理技术蓬勃发展，各类导航系统相继革新，由视觉导航的成本低廉，因此此项技术获得了越来越多的应用，已逐渐走向成熟。但在我国目前正处于起步阶段，对 SLAM 技术的研究正在完善中，因此催生了不少与 SLAM 技术相关的职业，相信在未来一定能填补 SLAM 技术在定位领域的空白。

## 1.3 系统主要内容以及结构

本系统主要是将 ORB-SLAM3 系统的定位与建图功能应用到导航定位系统<sup>[11]</sup>，此外，可以将扩展到更多的应用场景，包括汽车驾驶场景，室内阴暗场景等一些日常生活中常见的环境，主要利用 Android 手机的相机作为视频数据采集工具，电脑运行 SLAM 算法提供数据处理。本系统需要实现如下功能：

（1）SLAM 定位。SLAM 定位是指在未知环境中同时进行自主定位和地图构建的过程。旨在通过使用传感器数据来估计机器人或移动设备的位置和姿态，同时使用这些位置估计来更新和构建环境的地图。这一过程通常通过使用优化方法来实现。

（2）地图构建。地图构建通常依据传感器获取信息及数据处理算法的差异而不同。SLAM 建图过程中通常需要包含载体的相对位姿信息以及周围环境景物的三维位置信息。用绝对轨迹误差（Absolute Trajectory Error, ATE）和相对位姿误差（Relative Pose Error, RPE）来评估系统算法的精确度。

（3）回环检测。在 SLAM 系统中，当机器人或相机移动并建立地图时，存在误差累积的问题，尤其是系统在长时间运行或遇到重复环境时。回环检测的目的就是通过识别之前访问过的位置或环境，并将其与当前观测到的数据进行匹配，来减小累积误差并纠正机器人或相机的位置估计，帮助系统解决误差累积问题，提高定位的准确性和鲁棒性。在回环检测模块中，用词袋模型作为支撑，该模型可以帮助识别相似的场景，通过匹配特征向量的相似性来判断是否存在回环。

（4）传感器标定。视觉传感器标定的目的是估计相机的内部参数和外部参数，从而实现准确的图像测量和场景重建。标定过程通常涉及将已知的标定模式（例如棋盘格）放置在不同的位置和角度，并使用相机拍摄图像。通过分析图像中标定模式的形状和位置变化，可以推断出相机的参数。相机标定是一切三维视觉的基础。本系统采用 Android 手机的相机作为传感器，为单目模式的相机。

（5）三维重建。本系统的地图构建是基于稀疏点云模式的，存在着明显的缺点。因此为了实现对障碍物、环境信息更加准确的分析，本系统提出一种三维重建方法，使用最新的深度估计模型以最大程度上实现实际场景的构建。同时，为了增添系统与使用者的交互性，为系统增加了 AR 的功能。

## 1.4 本章小结

本章通过对导航定位技术和视觉定位技术的国内外研究背景分析，指出本系统设计的可行性，明确本次课题需要完成的工作以及需要实现的功能。

## 第 2 章 系统环境分析

本章主要介绍系统开发中所需的开发环境、开发语言、开发所使用的工具包以及使用到的软硬件信息。

### 2.1 系统运行平台

本系统是基于 Linux 操作系统的，原则上可以选择使用虚拟机搭建，但是由于本系统运行需要较大的运行算力，故此选择 Linux 系统的另一种运行方式——基于笔记本电脑的双系统模式，即 Windows 系统与 Linux 系统共存一台计算机的方式，双系统可以在开机时自主选择所需系统，不管是哪一个系统，都能充分使用到电脑的硬件资源，从而实现最大化分析系统运行状况。

#### 2.1.1 Linux 操作系统

本次系统设计采用的是基于 Linux 系统架构的 Ubuntu20.04 桌面版。Linux 系统类似于 Windows 系统，它具有开放和免费的特点，同时允许用户自由地使用、修改和分发，也是一种支持多任务和多用户的网络操作系统，它拥有优秀的可移植性，可在多种硬件设备上广泛应用，包括个人电脑、服务器、嵌入式系统和超级计算机等。其次，它支持各种开发工具和编程语言，为开发者提供了丰富的开发环境，Linux 系统还拥有丰富的应用程序和工具，包括办公套件、图形界面、媒体播放器、网页浏览器等，满足了各种用户需求，支持多任务和多用户环境，允许多个用户同时访问和操作系统<sup>[12]</sup>。

Linux 操作系统有自己开源社区的贡献和支持。许多开源软件和项目都与 Linux 系统紧密相关，形成了一个庞大的开源生态系统。这个生态系统不断推动着 Linux 操作系统的发展和创新，为用户提供了丰富的选择和优秀的软件资源。因此，本系统选用 Linux 作为开发主平台。

#### 2.1.2 Ubuntu 系统

Ubuntu 是一种基于 Linux 操作系统的发行版，它以易用性和用户友好性而脱颖而出。Ubuntu 致力于提供一个免费、开放和可定制的操作系统，旨在使个人用户和企业用户都能够轻松使用和享受 Linux 的优势。每个版本的 Ubuntu 都包含 GNOME 桌面环境，Ubuntu 具有以下优点：

（1）易于使用：Ubuntu 注重用户友好性，提供直观的界面和简单的操作方式，使得用户可以轻松上手并快速适应系统。

（2）软件支持丰富：Ubuntu 拥有庞大的软件仓库，用户可以方便地通过软件中心或终端安装各种应用程序和工具。

（3）安全性：Ubuntu 系统注重安全性，通过及时的安全更新和强大的权限管理机制来保护系统和用户的数据安全。它默认以普通用户权限登录，执行与系统相关的任务需要提升权限，从而减少了潜在的风险。

（4）稳定性和可靠性：Ubuntu 经过广泛的测试和稳定性改进，提供了可靠的系统性能和稳定性。它的发行版本经过严格的质量控制，确保用户获得一个可靠的操作环境，这也是本系统选用 Ubuntu 系统的主要原因。

截至目前，Ubuntu 已经发行到了 22.04 版本。经过对比，Ubuntu 20.04 系统性能优良，BUG 少。与上一版本相比，Ubuntu 20.04 系统有更好的软硬件兼容性和更强的安全性，满足本系统的开发环境要求，能够长时间支撑系统稳定运行。

### 2.1.3 双系统分析

由于之前电脑主机中已经安装 Windows10 系统，占用了系统磁盘，因此为了使得 Ubuntu 系统与 Windows 系统共存，需要分出一块磁盘空间用于 Ubuntu 系统的安装。结构如图 2.1 所示：

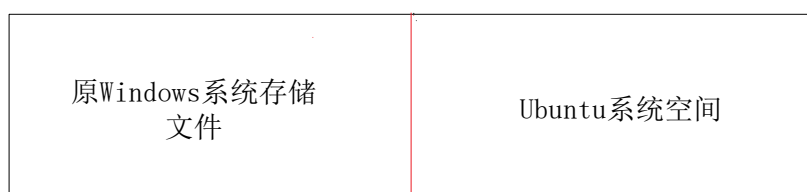


图 2.1 磁盘划分图

在将磁盘空间划分给 Ubuntu 系统之后，需要对其进行系统分区，主流的有两种分区方式，如图 2.2 和图 2.3 所示：

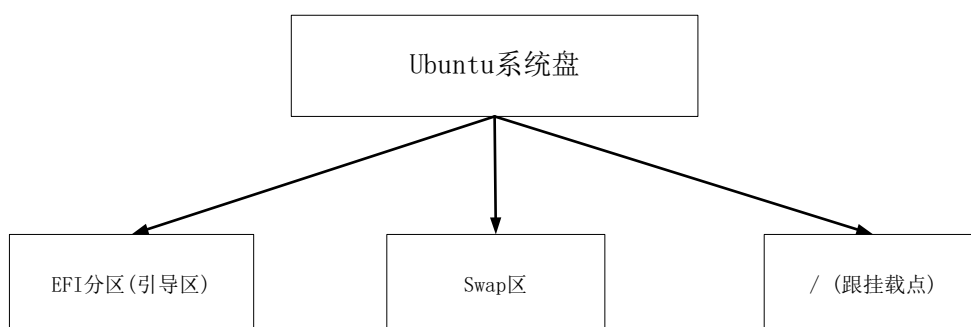


图 2.2 分区方式 1

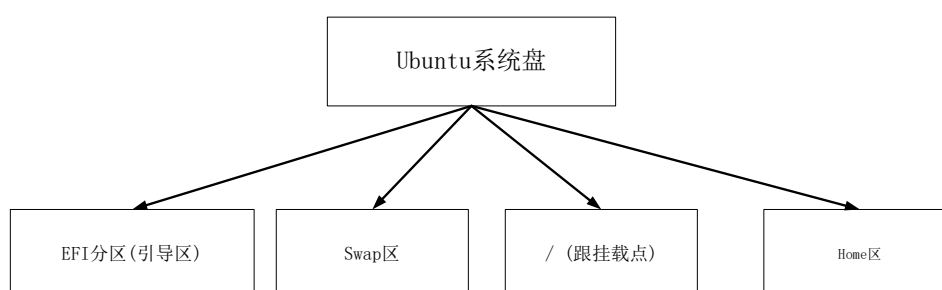


图 2.3 分区方式 2

本系统开发使用到的是分区方式 2，目的是为了将系统开发数据保存在 Home 分区，存储在 Home 分区的优点之一就是用户数据与 Ubuntu 系统数据分离，如需重装系统可以保留用户数据而只装系统驱动文件数据。Ubuntu 系统文件及其功能如图 2.4 所示：

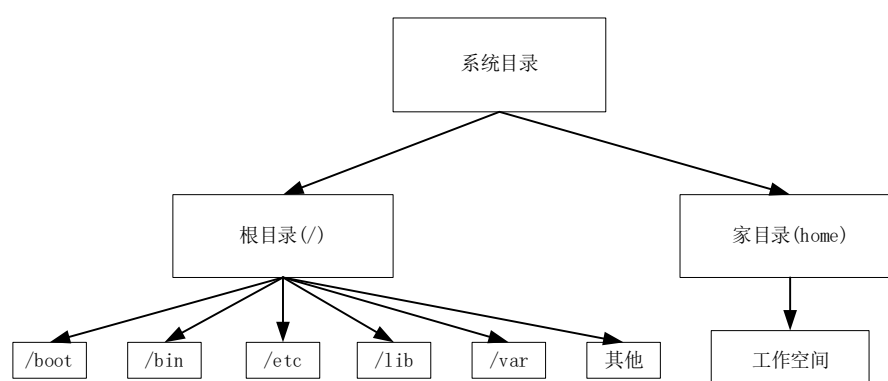


图 2.4 系统文件目录

## 2.2 依赖环境

系统使用 C++ 为主要开发语言，通过使用三方库，实现系统功能。以下介绍本系统所使用到的主要三方库及其作用。

### 2.2.1 主要三方库

本系统使用了多个主要的第三方库，其名称及功能如下：

（1）Eigen: Eigen 是一个开源的 C++ 模板库，由于 Eigen 的接口易于使用和学习，本课题使用 Eigen 库来进行矩阵运算、线性代数等各种数学操作。

（2）g2o: g2o 是一个用于图优化的开源库，使用 g2o 来进行图优化，优化相机位姿和地图点的位置，从而提高系统的精度和稳定性。

（3）DBoW2: DBoW2 是一个用于词袋模型的开源库，使用 DBoW2 来进行特征点描述子的聚类 and 匹配，实现回环检测。

（4）Pangolin: Pangolin 是一个轻量级的跨平台可视化库，使用 Pangolin 来可视化系统的运行结果，包括相机轨迹、地图点云等，本系统使用 0.6 版本。

（5）OpenCV（Open Source Computer Vision）是一个开源的计算机视觉和图像处理库，旨在提供一套丰富的功能和算法。该开源库兼顾了充分发挥 CPU 性能和广泛应用于处理各种类型的视觉数据，如图像、视频、深度图像等。它包含了众多图像处理和计算机视觉算法，如图像滤波、边缘检测、形态学操作、特征提取、目标检测和跟踪等，是计算机视觉必不可少的一个库文件。它还提供了一些工具，如图像和视频的读写、显示、存储等。目前 OpenCV 已经更新到第四版本，其核心模块组成如下图。本系统主要通过使用 OpenCV 部分模块的功能实现特征点的提取、跟踪。其主要功能模块如图 2.5 所示：

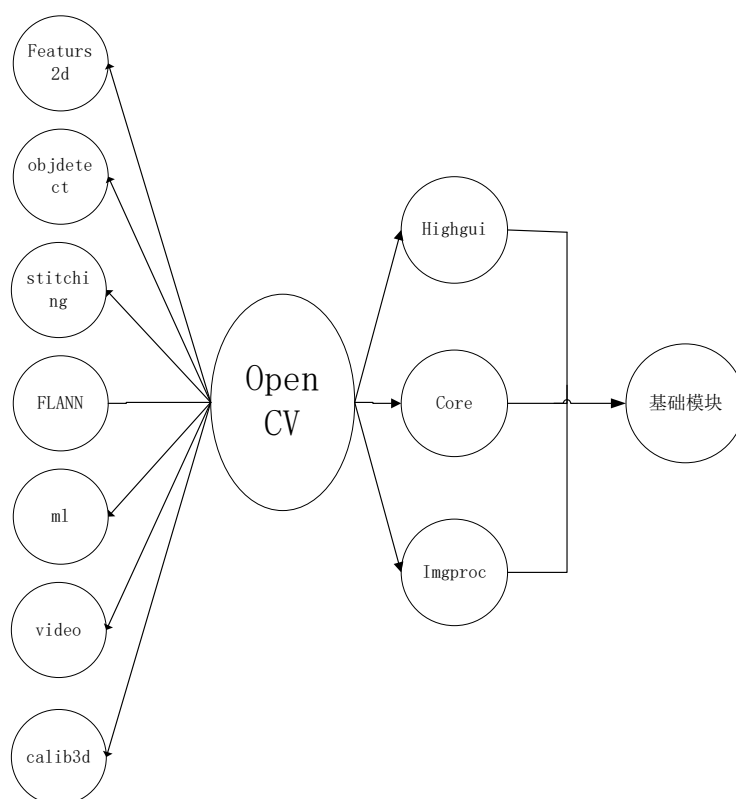


图 2.5 OpenCV 组成模块

(1) **Core 模块：**OpenCV 的核心模块，提供了基本的数据结构和函数，包括矩阵、向量、数组等。它包含了基本的图像处理功能。

(2) **Imgproc 模块：**提供了图像处理和图像操作的函数，包括滤波、边缘检测、形态学操作、图像转换等。它是进行图像预处理和增强的关键模块。

(3) **Highgui 模块：**提供了图形用户界面相关的函数，包括显示图像、创建图像窗口、处理鼠标和键盘事件等。它用于创建交互式图像处理应用程序的用户界面。

(4) **Features2D 模块：**Features2D 模块提供了图像特征检测和描述相关的算法和函数。它包括了一些经典的特征检测器和描述子，如 SIFT、SURF 和 ORB 等，可用于目标识别、图像匹配和图像配准等应用。

(5) **Objdetect 模块：**提供了目标检测和人脸识别相关的函数和算法，可用于检测和识别图像中的目标和人脸。

（6）**Stitching** 模块：用于图像拼接和全景图像生成。它提供了图像对齐、特征提取、图像融合和全景图像渲染等功能。

（7）**FLANN** 模块：用于高效的近似最近邻搜索。

（8）**ML** 模块：提供了各种机器学习算法和工具。它支持分类、回归、聚类和降维等任务，并提供了训练和预测的功能

（9）**Calib3D** 模块：用于摄像机标定和三维重建。它提供了相机标定、立体视觉、立体匹配和三维重建等功能。

（10）**Video** 模块：包含了视频处理相关的函数和算法，包括视频捕捉、视频录制、光流估计、目标跟踪、运动检测等。它为实时视频处理和分析提供了丰富的工具和功能。

### 2.2.2 PCL 库与 VTK 库

在文献[13]中详细论述了 PCL 的结构及功能。PCL（Point Cloud Library）也是一个开源的计算机视觉库，专门用于点云数据处理和三维几何图像处理。它提供了一系列算法和工具，用于获取、处理、分析和可视化点云数据。PCL 包含了许多先进算法，比如滤波，特征估计，表面重建，模型拟合和分割等，适用于各种点云处理任务和应用。本系统使用 PCL 来构建数据得点云信息，读取图像和深度图像数据，构建点云数据，并使用 PCL 进行点云的可视化显示，结合系统运行得到的关键帧信息，计算出运行轨迹，将三维重建信息进行可视化输出。

VTK（Visualization Toolkit）是一个开源的三维数据可视化和图形处理库，用于创建、处理和呈现各种类型的三维数据。它提供了丰富的功能和算法，可用于科学可视化、医学图像处理等领域。VTK 是一个基于 OpenGL 三维函数库的面向对象设计的开发库，它由一个 C++ 类库和解释封装层组成<sup>[14]</sup>。VTK 提供了一套丰富的可视化算法和工具，包括体绘制、表面绘制、体数据重建、插值、图像处理、曲面重建、等值面绘制、计算流体力学等，同时还提供了基于 Qt 的图形用户界面开发工具和 Python 接口。VTK 的使用如下结构图 2.6 所示：



图 2.6 VTK 使用图



**VTK** 主要由三个部分组成：数据模型、算法和可视化。

（1）数据模型：**VTK** 支持多种数据类型，包括点、线、面、体和多边形等。**VTK** 的数据模型主要包括数据集、网格和拓扑结构等，其中数据集是最基本的数据类型，网格和拓扑结构是数据集的扩展类型。

（2）算法：**VTK** 提供了丰富的算法和工具，包括数据转换、滤波、插值、曲面重建、等值面绘制、光照和阴影、交互式绘制等。其中最常用的算法是可视化管线（**Visualization Pipeline**），它由多个模块组成，每个模块负责处理数据的一部分，并将处理结果传递给下一个模块，最终生成可视化结果。

（3）可视化：**VTK** 提供了多种可视化技术和工具，包括体绘制、表面绘制、体数据重建、等值面绘制、计算流体力学等。**VTK** 的可视化技术主要基于现代计算机图形学技术，如 **GPU** 加速、渲染管线等。

## 2.3 软硬件参数

本系统的视觉模块采用的是小米 9 **Android** 手机，如图 2.7 所示。安卓手机集成了惯性与视觉信息，是一个轻量化的平台能够作为 **ORB-SLAM3** 的载体进行运行，使用 **Android** 手机的自带相机作为数据采集传感器，本系统的视频采集参数如表 2.1 所示。



图 2.7 **Android** 手机实物图

表 2.1 硬件采集模组参数

单目相机	
工作距离	0.5m——5m
相机类型	三摄像头
相机分辨率	800*600
帧率	30Hz

根据 Android 手机自带的视频拍摄软件无法获取本系统所需的数据，故需要开发匹配的 Android 软件，本系统所采的软件为使用 Java + Android SDK 工具开发，软件名称为 VideoIMUCapture，软件界面如图 2.8 所示：

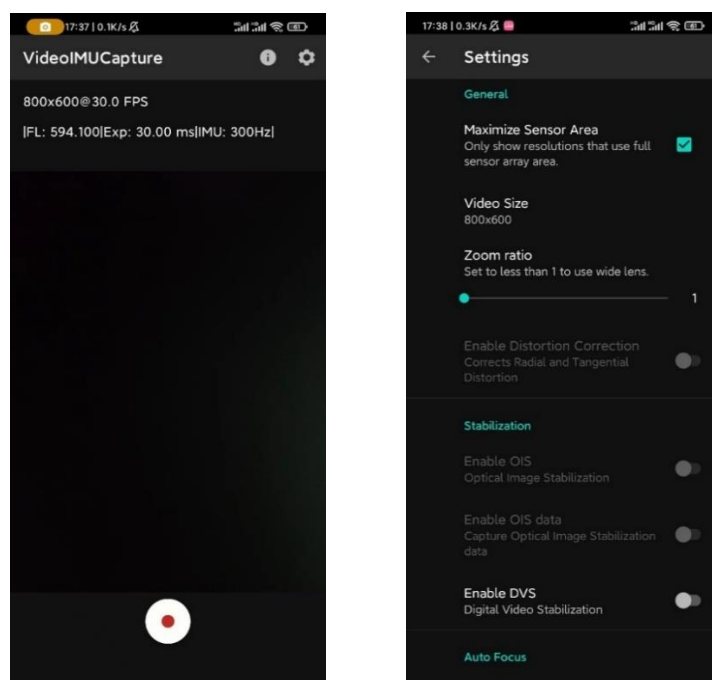


图 2.8 VideoIMUCapture

该软件将相机的分辨率设置为 600\*800 像素，FPS 设置为 30Hz，IMU 帧率为 300Hz。同时，为了适应更多的场景，可以通过设置功能修改参数。

## 2.4 本章小结

本章介绍了系统运行的主要环境和所需的主要三方库，对系统环境的构成和文件组成做了详细分析。同时，对三方库的主要功能模块和本系统中的作用做了详细分析。介绍使用的传感器模组，给出具体的参数信息，将数据采集软件的开发和参数做一个详细呈现。

## 第 3 章 关键技术概述

本章简要介绍 SLAM(Simultaneous Localization and Mapping)技术的算法原理，介绍 ORB 特征点技术的图像数据处理方法和本系统所涉及到的主要技术原理。

### 3.1 视觉 SLAM 技术概述

SLAM 的全称是自主定位与地图构建技术，用通俗的话来说就是将智能机器人放入一个未知的环境中，让机器人通过传感器数据一边描绘出所在环境的完全地图，同时一边决定机器人的前进方向。因此，SLAM 是旨在通过传感器数据，实现对自身位置的准确估计以及对周围环境的地图构建。

视觉 SLAM 可以分为前端、后端、建图和回环检测四个核心部分，将这四个部分划分为三个线程。前端负责从连续的图像帧中提取特征，并通过特征匹配和跟踪来计算相机的姿态变化；后端则利用前端提供的姿态估计结果进行优化，以得到更准确的位姿估计。而建图则是根据姿态估计和传感器数据，构建当前环境或坐标系下的地图。回环检测是指通过识别已经到达过的地点或场景，进一步优化位姿轨迹和地图，以提高整体的一致性和准确性。框架如图 3.1 所示：

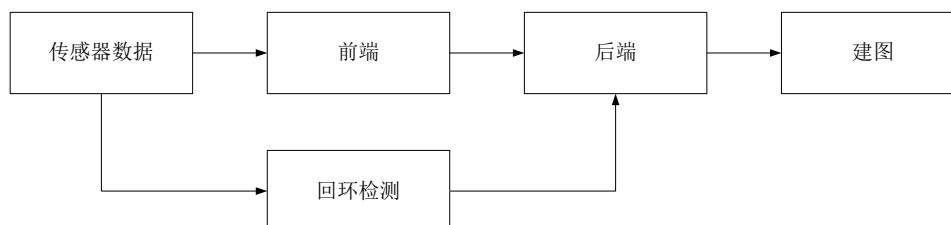


图 3.1 SLAM 框架图

#### 3.1.1 ORB-SLAM 原理

ORB-SLAM 是一种视觉 SLAM 系统，系统结合了特征点提取、相机姿态估计和地图构建等功能，实现了实时的视觉定位和建图，整个系统由三个并发线程模块构成：跟踪线程、局部建图线程、回环检测线程<sup>[15-17]</sup>。三大线程如图 3.2 所示：

（1）跟踪线程（Tracking）是本系统的核心线程，主要实现实时跟踪相机的姿态和地图点。该线程通过提取图像特征并进行特征匹配，估计相机的姿态。它还负责检测和处理丢失跟踪的情况，并进行重定位以恢复跟踪。

（2）局部地图构建线程（LocalMapping）是负责构建和更新局部地图的线程。它的主要任务是处理跟踪线程提供的关键帧和地图点数据，并在局部地图中添加新的关键帧和地图点，从而提供实时的定位和地图可视化。

（3）回环检测线程（LoopClosing）是负责检测回环并进行闭环优化的关键线程。它的主要任务是识别相机在不同时间下经过的相似地点，并将这些回环信息用于地图的优化和姿态校正来提高定位的准确性和鲁棒性。通过回环检测和闭环优化，本系统能够更好地处理场景中的动态变化和尺度漂移等问题。

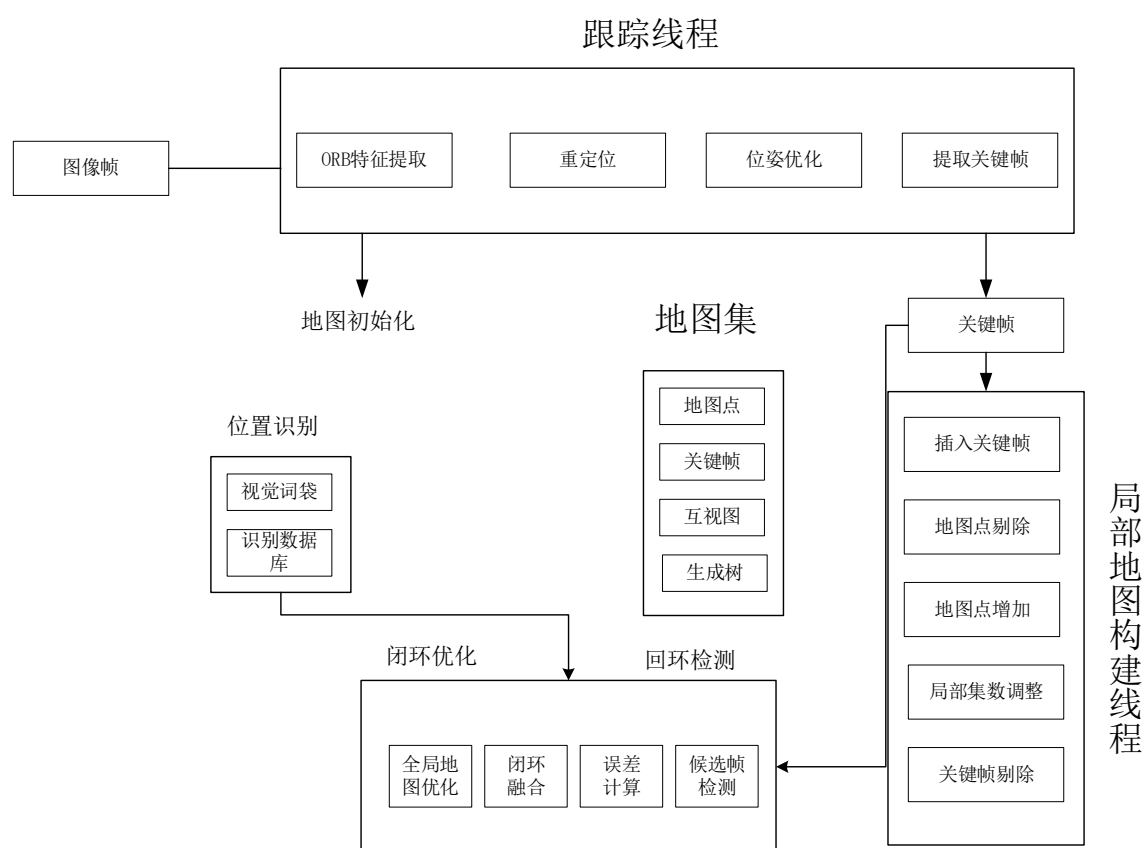


图 3.2 算法框架图

## 3.2 ORB 算法原理

ORB 算法是一种用于图像处理的算法，它的全称是 **O**riented **F**AST and **R**otated **B**RIEF 算法<sup>[18]</sup>。该算法使用 FAST 算法<sup>[19]</sup>来提取图像中的特征点，并使用 BRIEF 算法来构建这些特征点的描述子。同时，ORB 算法还对这些描述子进行修正，以实现特征点的尺度和旋转不变性。也就是说，经过缩放和旋转后的特征点仍然能够产生与原始特征点相似的描述子。

### 3.2.1 ORB 特征及特征点

图像特征是图像中具有一定特点和可辨识度的局部区域或描述符，用于表示图像的关键信息。图像特征可以通过各种算法和方法进行提取和描述，用于表示图像的关键信息，从而实现图像分析、识别和处理任务。

ORB 特征是一种常用的图像特征，其最大的优点是提取速度快。它结合了 FAST 角点检测和 BRIEF 描述子，具有快速计算的良好性能，其在计算速度和性能之间取得了平衡，适用于实时的图像处理、特征匹配和 SLAM 等应用领域。

特征点是图像中具有显著性质的点或区域，可以用于图像分析、匹配和定位等任务。它们通常具有以下特征：特征点具有三大特点：显著性、不变性、易于检测和描述性。基于上述特性，因而特征点在图像特征信息中具有举足轻重的地位。关键点及描述子是特征点中的两大极其重要的信息。关键点（**Key Points**），是图像中具有显著性质的局部区域，在本系统中的作用和特征点类似，但是高于特征点。描述子

（**Descriptor**）是用于描述关键点周围图像区域特征的向量，通过描述子提取关键点周围的图像信息，并将其表示为具有可区分性和稳定性的特征向量，可以对关键点进行匹配和识别。将二者结合通过关键点的位置、朝向和描述子，可以进行关键点的匹配、特征的识别和目标的跟踪。

### 3.2.2 ORB 特征点提取

ORB 特征提取流程图如图 3.3 所示。首先将输入的图像进行预处理，利用改进的 FAST 方法来计算特征点，这一步既保留了 FAST 非常快的检测速度的优点，又在特征点中增加了方向信息，通过灰度质心来计算方向，当图像的某一部分像素发生了明显的变化，这个像素点就会被检测到。在角点检测后，需要对每个关键点周围的图像区

域进行描述，以生成特征向量。ORB 特征使用的是 BRIEF 描述子，它通过对关键点周围的像素对进行二进制编码来描述图像特征。

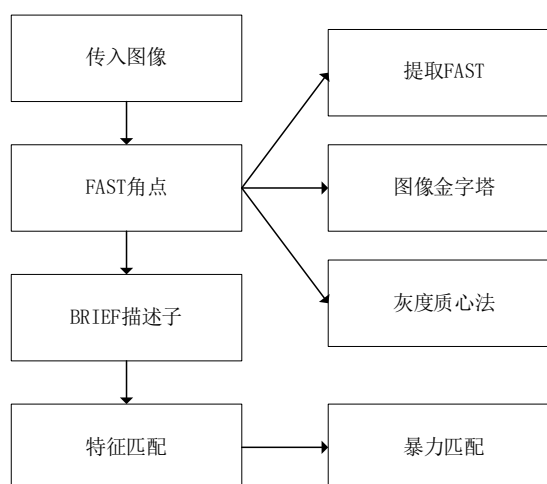


图 3.3 ORB 特征提取流程图

因为 FAST 只需要计算像素点，所以才有非常快的检测速度，如图 3.4 所示。其像素点的提取过程大致如下所示：

（1）建立图像金字塔，将图像按照一定的比例因子扩大，建立  $nlevel$  层金字塔，以供在多个尺度上进行 FAST 角点检测，实现特征点的多尺度不变性。

（2）确定一个以目标像素为圆心，半径为  $r$  的圆，若该圆上有连续  $k$  个像素值与目标像素值的差均大于某阈值  $T$ ，则判断目标像素为角点。通常情况下取  $r = 3$ ，ORB 算法中的  $k$  值取 9。

（3）然后循环上一步，对每一个像素点如此反复。

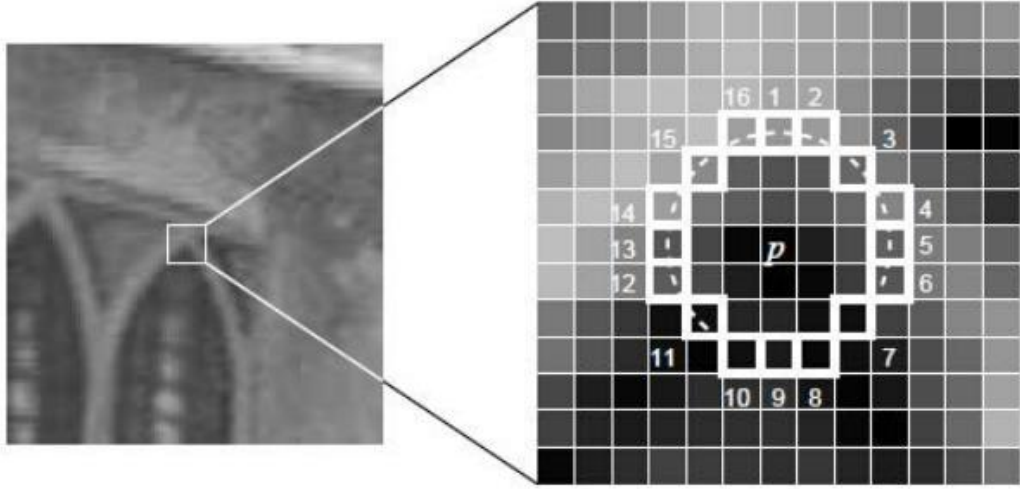


图 3.4 FAST 角点

FAST 算法会产生很多特征点密集的分布在图像的特征区域,可以选择只保留区域中最好的一个特征点,以减少数据量,提高后续算法的效率。这一过程可以通过非最大值抑制与计算 Harris 响应来完成。非最大值抑制是指,首先对于每一个特征点  $I_c$  计算一个  $S$  值,为其与提取时圆上各像素  $I_i$  值之差的绝对值之和,如式 3.1:

$$S = \sum_i |I_c - I_i| \quad (3.1)$$

对于每一个特征点,检查其邻域内的所有特征点,如果该特征点的  $S$  值不是邻域内所有特征点  $S$  值最大的,则舍弃该特征点。

而 Harris 响应则是根据 Harris 角点的算法,对整张图片所有特征点各计算一个响应值  $R$ ,取  $R$  最大的前  $N$  个特征点保留。

Harris 角点算法基于以下基本思想:在图像中,具有临近位置灰度差值较大的点被认为是角点。为了实现这一目标,我们在图像上构建一个小窗口,并在图像上滑动该窗口。如果窗口在滑动前后的位置上出现灰度变化较大的情况,那么认为该位置是一个角点。设点  $(x, y)$  处的窗口在滑动  $(\Delta x, \Delta y)$  之后的灰度变化可以用公式 3.2 表示:

$$C(x, y, \Delta x, \Delta y) = \sum_{(u, v) \in W(x, y)} \omega(u, v) (I(u, v) - I(u + \Delta x, v + \Delta y))^2 \quad (3.2)$$

其中  $W(x, y)$  是  $(x, y)$  附近与窗口等大的邻域。 $\omega(u, v)$  是以  $(x, y)$  为中心的窗口的值,通常选择使用高斯分布,也可以使用全 1。 $I(u, v)$  表示图像上  $(u, v)$  点的灰度值。



### 3.3 单目相机成像原理与相机标定

本系统所采用的数据采集工具类型为单目模式的相机，不同于从正规单目相机，Android 手机相机参数不符合 SLAM 系统所需的配置文件要求，故需要对相机各项参数进行标定，相机参数的正确标定直接影响系统运行的精度。

#### 3.3.1 单目相机成像原理

相机结构根据物理知识最基本的解释就是透镜模型，所有平行光穿过如果由透镜中心的射入，则在射出时不会改变方向，经过其他角度的射入光线会经过折射作用在一定程度改变射出的方向。如图 3.5 所示：

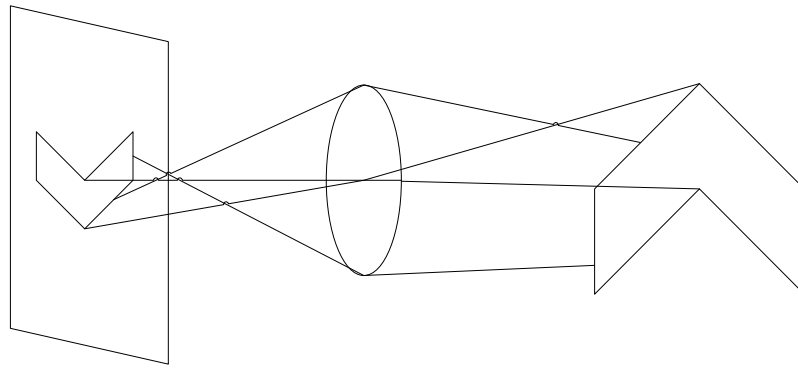


图 3.5 透镜原理

当将透镜模型简化为小孔成像模型时，光线通过小孔后的成像过程可以用连线来描述。如图 3.6，在小孔成像模型中，光线从物体通过小孔进入相机后并没有改变方向，而是直线传播到成像平面上形成图像。因此，图像中的每个像素点都与原始物体上的点通过一条连线相对应。这种模型假设了小孔的尺寸非常小，因此光线在通过小孔时几乎不发生散射或折射。本系统所有相机算法均属于小孔模型算法，将单目相机归类为小孔相机，因此可以直接使用小孔相机成像理论。

在相机的成像过程中存在三个坐标系：相机坐标系（也称为光心坐标系）、物理坐标系、像素坐标系。整个成像过程可以理解为从相机坐标系下的点转为物理坐标系，再从物理坐标系转为像素坐标系。

##### （1）相机坐标系到物理坐标系

假设相机坐标系下有一点 $P(x, y, z)$ ，投影到成像平面坐标系下为 $P'(x', y')$ ，已知  $x$ 、 $y$ 、 $z$  和  $f$ （焦距），由于是小孔成像模型，成像平面与相机平面平行，且两坐标系原点同在相机坐标系  $Z$  轴。依据相似三角形原理，可以得出 $P'(x', y')$ 的坐标，根据相似三角形原理，且小孔成像导致图像会旋转  $180^\circ$ ，可得如下公式 3.3、3.4 和 3.5：

$$-\frac{x'}{f} = \frac{x}{z} \quad (3.3)$$

$$x' = f \cdot \frac{-x}{z} \quad (3.4)$$

$$y' = f \cdot \frac{-y}{z} \quad (3.5)$$

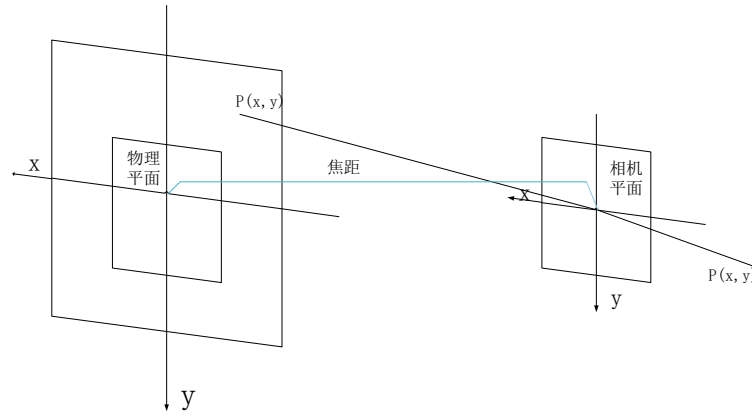


图 3.6 小孔成像原理图

## （2）从物理坐标系到像素坐标系

得到 $P'(x', y')$ 后，将物理平面的所有点都映射到像素平面，需要先进行缩放，然后再加上点的偏移。 $P'(x', y')$ 映射到像素平面后为 $P(u, v)$ ，假设 $x'$ 轴和 $y'$ 缩放倍数为 $\alpha$ 和 $\beta$ ，则 $P(u, v)$  的坐标由式 3.6 和 3.7 可得：

$$u = \alpha x' + C_x = \alpha f \frac{-x}{z} + C_x \quad (3.6)$$

$$v = \beta y' + C_y = \beta f \frac{-y}{z} + C_y \quad (3.7)$$

小孔成像会使得成像图像翻转，但是将成像平面看做是在相机前面，如此一来，就可以把公式中的“ $-$ ”去掉。

现在将上述式子中 $\alpha f$ 的替换为 $f_x$ ， $\beta f$ 替换为 $f_y$ ，则有式 3.8 和 3.9：

$$u = f_x \frac{x}{z} + C_x \quad (3.8)$$

$$v = f_y \frac{y}{z} + C_y \quad (3.9)$$

变换为矩阵形式有式 3.10:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{f_x}{z} & 0 & \frac{C_x}{z} \\ 0 & \frac{f_y}{z} & \frac{C_y}{z} \\ 0 & 0 & \frac{1}{z} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (3.10)$$

提取出 $\frac{1}{z}$ ，设为  $K$ ，于是得到内参矩阵为式 3.11:

$$K = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.11)$$

相机的内参矩阵是描述相机内部几何属性和成像特性的重要参数，它对像素坐标与相机坐标之间的转换提供了必要的几何信息和参数。

### 3.3.2 相机标定原理

理论物理模型都是在特定的理想情况下对真实事物的模拟，然而在实际的工程应用中或多或少会因为各种原因存在误差，小孔相机的成像模型同样是在理想状态下得出的结论。相机的成像误差主要有两类：一种是在传感器制造时产生，这是由生产工艺及技术决定的；另一种就是镜头在制造和安装时产生的误差，镜头一般存在非线性的径向畸变；镜头与相机传感器安装不平行，还会产生切向畸变。如图 3.7 所示：

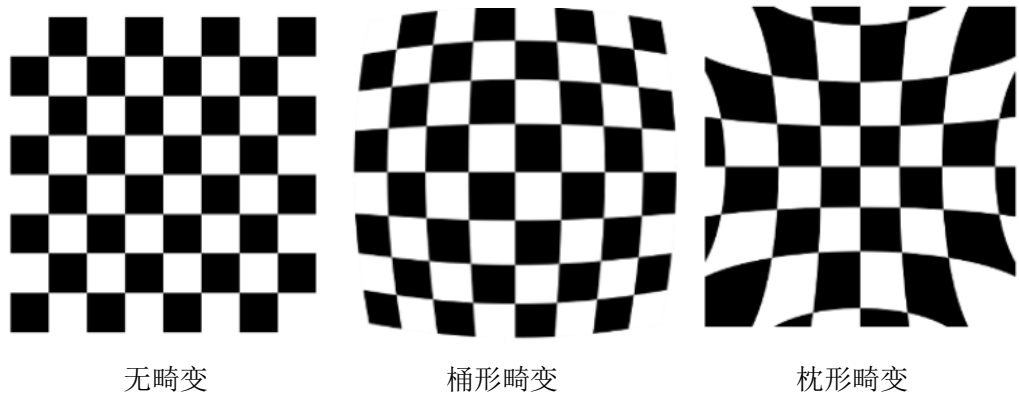


图 3.7 镜头畸变示意图

在将采集到的数据运用于 SLAM 系统时，如果未能准确标定相机参数则会对视频数据产生较大失真，只有完成相机参数的标定校正工作，才能得到准确的成像数据。

相机拍摄的图片是二维图片，但现实场景是三维立体的，相机就是把三维环境映射到二维世界，用函数关系式可以理解为输入是一个三维信息，输出是一张二维图片。

相机标定就是通过输入带有标定图样的标定板来计算相机参数，其成像过程可以用一个简易的数学模型来表示。而数学模型，就是需要求解的相机的参数，包括相机的内参、外参以及畸变参数。得到了数学模型就得到了相机的参数，就可以得到相机拍摄的正确图片。标定模型如图 3.8 所示：

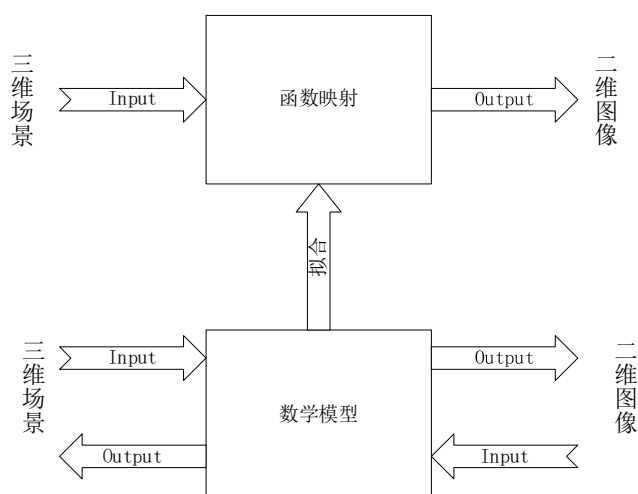


图 3.8 相机标定模型

### 3.4 误差分析原理

当评估一个 SLAM 算法的表现时，我们通常从时间消耗、复杂度和精度等多个角度进行考量。其中，对于精度的评价最为关键。在评估精度时，常常需要使用两个指标，即绝对轨迹误差估计和相对轨迹误差估计。

绝对轨迹误差（Absolute Trajectory Error, ATE），主要用于衡量系统估计的相机轨迹与真实轨迹之间的差异，如图 3.9。通过将系统估计的相机位姿与真实位姿对齐，然后计算它们之间的距离或差异。最后，将所有时间步的误差进行累加或者求取平均值，得到整体的绝对轨迹误差。

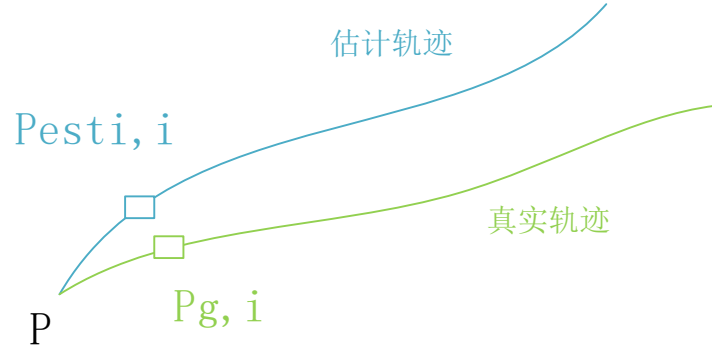


图 3.9 绝对轨迹分析图

记  $P_{1:n}$  代表待估计的轨迹,  $Q_{1:n}$  代表真值轨迹, 则时间戳  $i$  处的绝对估计误差为:

$$F_i = Q_i^{-1} S Q_i \quad (3.12)$$

则针对所有时刻定义平移分量的均方根误差(RMSE)、和方差 (SSE):

$$RMSE(F_{1:n}, \Delta) := \left( \frac{1}{m} \sum_{i=1}^m \|trans(F_i)\|^2 \right)^{\frac{1}{2}} \quad (3.13)$$

$$SSE := \sum_{i=1}^n \|trans(F_i)\|^2 \quad (3.14)$$

相对位姿误差 (Relative Pose Error, RPE) 测量了轨迹在一个固定的时间区间  $\Delta t$  内的局部准确度, 如图 3.10。因此, 相对位姿误差对应的是轨迹的漂移, 与 ATE 不同的是, RPE 是分别计算每条轨迹上两个时刻 (也就是两个点) 的变化, 再求这两个变化的误差, 其中  $trans$  表示取括号内部变量的平移部分。

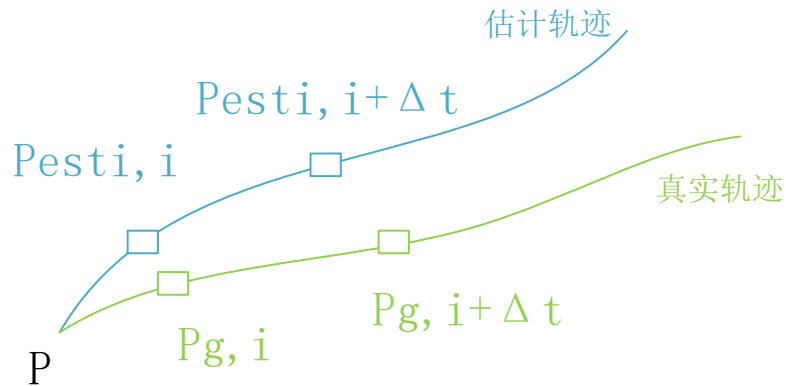


图 3.10 相对轨迹分析图

先定义时间步  $i$  处的相对位姿误差如下，可知相对位姿误差计算的是相隔固定时间差  $\Delta$  两帧位姿差，由式 3.15 计算为：

$$E_i := (Q_i^{-1}Q_{i+\Delta})^{-1}(P_i^{-1}P_{i+\Delta}) \quad (3.15)$$

根据式 3.15，对于一个有  $n$  个相机位姿的序列中，我们获得  $m = n - \Delta$  个独立的沿着序列的相对位姿误差。则可以定义平移分量的所有时刻的均方根误差 **RMSE** 由式 3.16 可得：

$$RMSE(E_1:n, \Delta) := \left( \frac{1}{m} \sum_{i=1}^m \|trans(E_i)\|^2 \right)^{\frac{1}{2}} \quad (3.16)$$

### 3.5 本章小结

本章对系统涉及到主要原理做了详细说明，包括 **SLAM3** 的三大主要线程，以及这三大线程之间运行的作用；详细阐述了 **ORB** 特征点的提取原理，相机的成像原理，相机的标定原理，对误差的计算方法以及涉及到的两类误差计算公式给予详细说明。

## 第 4 章 系统功能设计与实现

基于 ORB 特征的即时定位与地图构建系统的整体功能是在 Ubuntu20.04 系统上完成，包括系统环境搭建、功能设计、定位与建图、三维重建、AR 模型引入。数据采集和处理通过利用 Android 手机来获取模组图像数据、获取视觉数据，将采集到的数据处理为 SLAM 系统所规范的数据形式，并通过 EVO 评测工具分析结果。

### 4.1 系统框架结构

本系统主要分为前端、后端和回环检测三个主要模块。前端负责实时处理输入图像序列，提取特征点并进行跟踪，以估计相机的运动和位姿。在前端中，使用 ORB 特征点提取和描述子生成算法，通过 FAST 角点检测器检测特征点，然后使用 BRIEF 描述子对特征点进行描述。同时，前端还负责对新的关键帧进行选择 and 插入，以保持地图的稀疏性；后端主要负责对地图进行优化和维护，以提高定位和地图的准确性。在后端中，使用图优化技术，如位姿图优化和非线性优化，来最小化重投影误差和优化地图点的位置。后端还负责处理闭环检测产生的回环，并进行回环优化，以减小地图的漂移；回环检测模块负责检测并处理闭环，即当相机经过先前访问过的位置时，通过回环检测来纠正地图的漂移和误差。通过这三个模块的协同工作，系统能够实现实时的单目视觉定位和建图的功能。

根据上文所述，结合 ORB-SLAM 的优缺点，本系统对建图功能做了一定程度的优化，通过使用单目深度估计模型实现三维地图重建。系统运行流程图如图 4.1 所示：

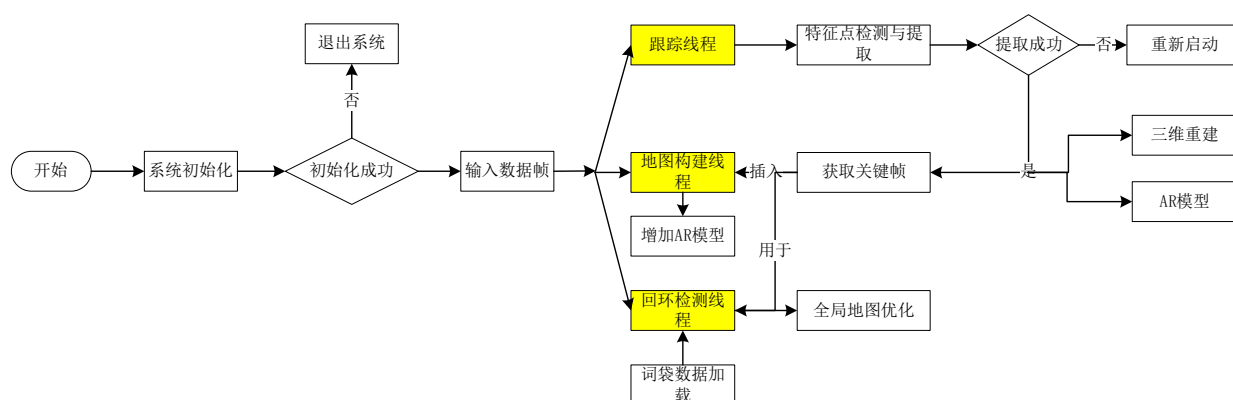


图 4.1 系统流程图

## 4.2 相机标定

本次设计的相机标定工作在基于 Linux 内核的 Ubuntu20.04 系统下完成，使用一款通用的标定工具 Kalibr 完成标定工作。可以在 docker 容器内完成，也可以独立安装 Kalibr 库，前者需要耗费比较大的主机内存，后者需要添加一些处理数据用的脚本文件，本次标定选用的方法为后者。

在视觉 SLAM 系统中，拥有一个参数正确的传感器直接决定了在系统运行的准确性，相机参数标定工作是所有机器视觉项目开发基础。

首先需要准备一张标准标定板，标定板的选择理论上，标定的物体可以是任何适当表征的物体，比如三维、二维、圆网格或随机图案，一般选择使用平面物体的多个视图，本系统采用了一张 4×6 的标准棋盘格标定板，如图 4.2 所示。

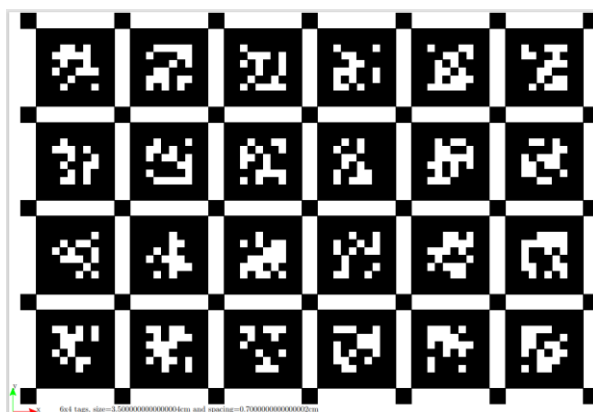


图 4.2 标准标定板

用 A4 纸打印下标准标定板，贴到平整的墙面上方便数据录制；第二步采用需要标定的相机对着标定板录制一定时间的视频，一般在两分钟以内，然后将数据转移到 Ubuntu 系统中；第三步就是用数据处理工具处理采集到的数据，将数据转换为 Kalibr 可处理的数据。标定流程如图 4.3 所示：

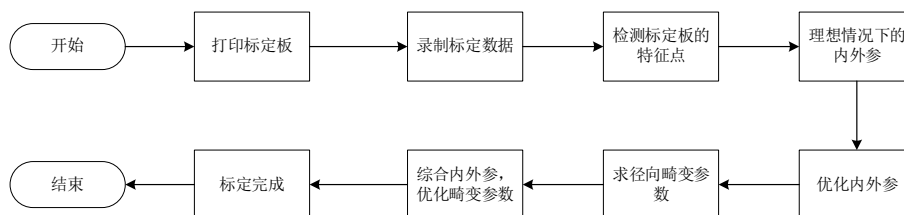


图 4.3 标定流程



标定过程如图 4.4 所示：

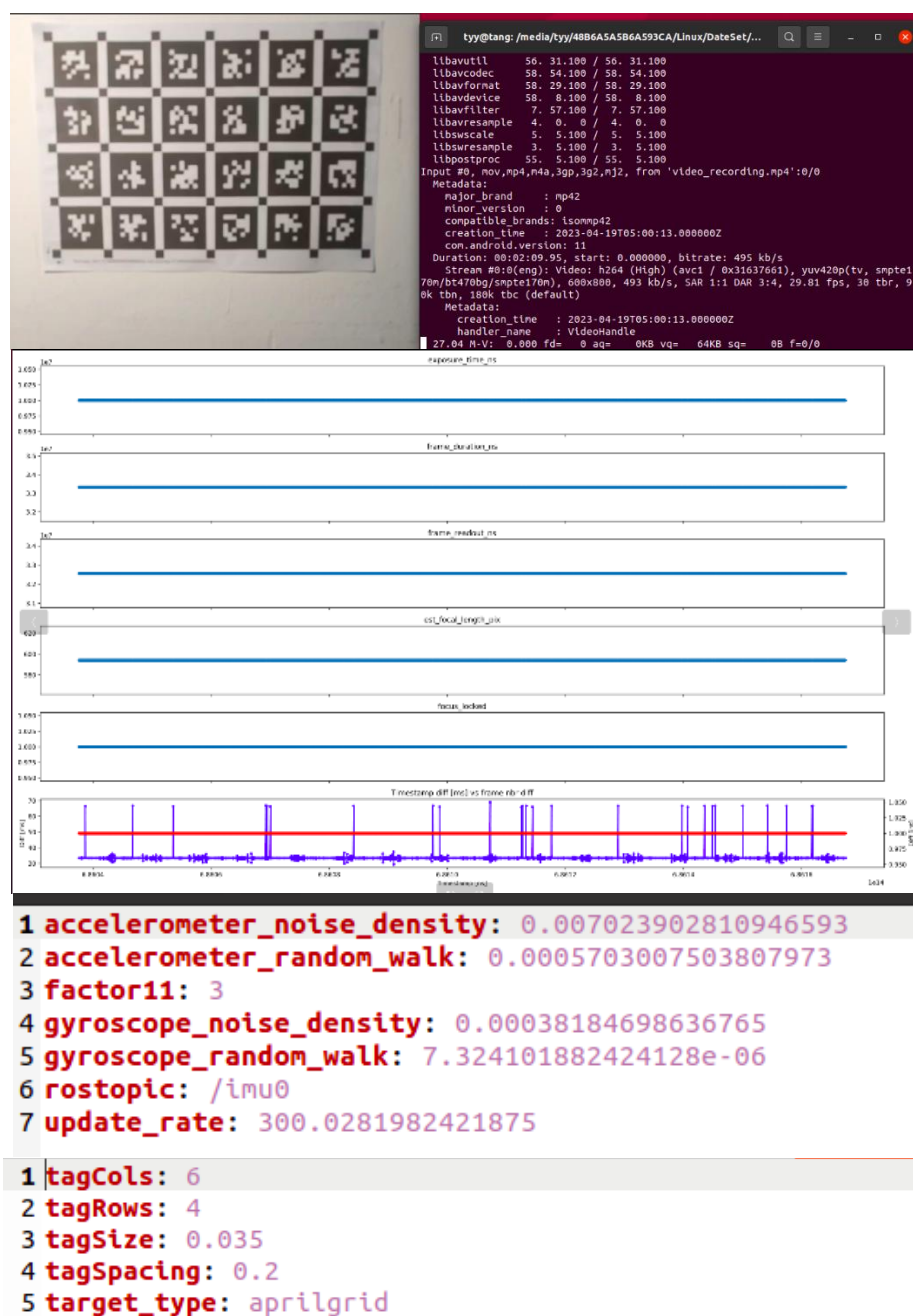


图 4.4 标定过程及结果展示图

数据参数有随机噪声，加速度噪声，IMU 参数，同时将标定板的类型给与说明，这些参数是对相机进行校正后得到较为准确得相机参数。

将标定得到的数据写入配置文件完成标定，如图 4.5 所示：

```

1 %YAML:1.0
2
3 # from EuRoc.yaml
4
5 -----
6 # Camera Parameters. Adjust them!
7 -----
8 File.version: "1.0"
9
10 Camera.type: "PinHole"
11
12
13 # intrinsics: [562.7316337110448, 560.3423735300665, 317.2140542719598, 391.6093401190242]
14 # [fu fv pu pv]
15
16
17 # distortion_coeffs: [k1 k2 r1 r2]
18 # [0.06553264584007061, -0.11595512032462152, -0.001117813380511963, 0.005219962734861495]
19
20 # Camera calibration and distortion parameters (OpenCV)
21 Camera1.fx: 595.7182282949835
22 Camera1.fy: 594.9814912688832
23
24 Camera1.cx: 295.23243453934515
25 Camera1.cy: 404.43474572668373
26
27 Camera1.k1: 0.06592973662768427
28 Camera1.k2: -0.12009953832701149
29 Camera1.p1: 0.0034511942358520452
30 Camera1.p2: -0.0023522955824459312
31
32 # Camera resolution
33 Camera.width: 600
34 Camera.height: 800
35
36 Camera.newWidth: 600
37 Camera.newHeight: 800
38
39 # Camera frames per second
40 Camera.fps: 30 #.0
41 # 29.79
42
43 # Color order of the images (0: BGR, 1: RGB. It is ignored if images are grayscale)
44 Camera.RGB: 0
45
46
47 IsColor: 0
48

```

图 4.5 单目标定结果

在将采集到的数据运用于 SLAM 系统时，如果未能准确标定相机参数则会对视频数据产生较大失真，如图 4.6 所示。因此，只有完成参数标定校正工作，才能使得数据准确成像。



图 4.6 相机标定前后对比

### 4.3 特征点提取

在使用正确参数的相机获取到视频数据后即可用于系统。系统首先会对传入的数据提取特征点，在获取到特征点之后，需要对特征点进行变化，转换为一串能够表示特征点信息的数据，ORB 中采用调整过的 BRIEF 算法来完成。

BRIEF 算法要在目标点的邻域内选取若干对点，一般取 128, 256 或 512，然后对每个点对 $(p_x, p_y)$ 通过比较  $p_x$  与  $p_y$  的灰度大小生成一位二进制数，得到的二进制串即为描述子。通过 Brief 算法得到的描述子可以通过汉明距离判断两个描述子之间是否匹配。汉明距离是指二进制串中对应位置不同的次数，即异或后 1 的出现次数。

BRIEF 算法首先在图像中使用角点检测算法选择关键点，这些关键点将作为特征点进行描述，对于每个选定的特征点，选择一组固定的像素对作为描述子的采样点，对于每对采样点，比较它们的灰度值，并将比较结果转化为二进制码。例如，如果第一个采样点的灰度值大于第二个采样点，则对应的二进制位设为 1，否则设为 0。这样，每对采样点生成一个二进制码，所有二进制码串联在一起形成最终的特征描述子，最后使用汉明距离或汉明权重距离等衡量两个特征描述子之间的相似性。较小的汉明距离表示两个特征相似度较高

本系统在基于上述理论下，获取到 ORB 特征点。首先需构建图像金字塔并初始化用于计算的描述子，在系统实现中，描述子已经确切给出，为 256 对坐标点，图像金字塔会对图片进行缩放和填充。填充完成后即可进行提取特征点并进行筛选，要求特征点需要尽量分布均匀，直接影响系统运行的效果。系统的筛选过程为：首先分块搜索特征点，如果特征点响应值普遍较小就降低阈值重新搜索；然后对得到的所有特征点进行八叉树筛选，如果某一块区域特征点数目过于集中则取响应值最大的一个。具体流程如图 4.7 所示：

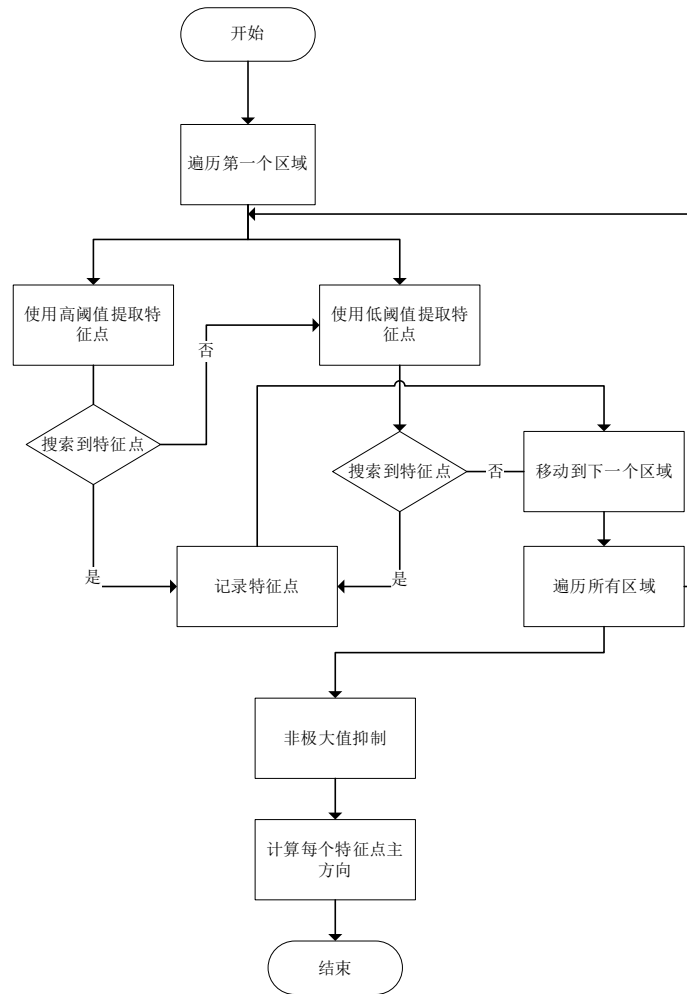


图 4.7 系统特征点提取

在本系统中使用对应的算法实现对特征点的提取以及筛选，如图 4.8。

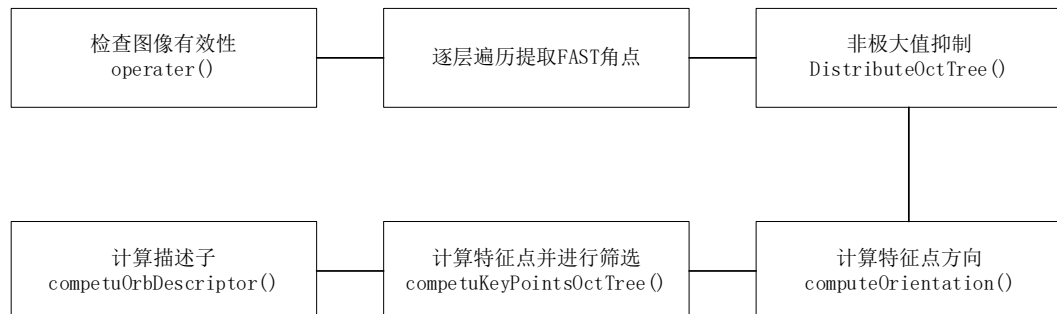


图 4.8 提取函数

## 4.4 单目深度估计与三维重建

由于 ORB-SLAM 系统构建的地图是稀疏点云地图，仅保留了图像数据中的一部分特征点作为关键点，这些关键点在空间中固定定位，难以准确描述地图中的障碍物。因此本系统设计了基于 Zoedepth 的三维建图，目的在于还原传感器所在周围真实环境，用以分析传感器周围的实际信息。

### 4.4.1 Zoedepth 单目深度估计

深度估计<sup>[20]</sup>，顾名思义就是指通过计算机视觉算法去推断图像中物体或场景的深度信息，它的主要目标是根据给定的单张或多张图像，估计每个像素点到相机的距离或相对深度。而单目深度估计<sup>[21]</sup>就是使用单目相机，即本系统中的 Android 手机的相机。单目深度估计指的是通过单目相机获取的图像信息，推断出图像中每个像素点到相机的深度信息，单目深度估计是视觉 SLAM 系统中的关键技术之一，能够帮助相机获取更多的场景信息，实现更加精准的定位和地图构建。

当前，单目深度估计可以分为度量深度估计（也称为绝对深度估计）和相对深度估计两个大类。在当代的单目深度估计研究中，一些方法过于注重泛化性能而忽略尺度，使用相对深度估计，而另一些方法则专注于给定数据集的结果，采用绝对深度估计。本系统采用了 Intel 的开源深度估计系统——Zoedepth。

ZoeDepth 是一种结合了相对深度和绝对深度的方法，它在保持度量尺度的同时，具有出色的泛化性能。本系统采用了 ZoeDepth 的旗舰模型 ZoeD-M12-N。该模型使用相对深度在 12 个数据集上进行预训练，并在两个数据集上使用绝对深度进行微调，从而在现有 SOTA 上实现了明显的提升。该模型还在多个数据集上进行了联合训练，但其性能并未明显下降，而且在室内外数据集上实现了零样本泛化性能。此外，ZoeDepth 通过在更多的数据集上进行微调和定义更细化的域，并在更多的绝对深度数据集上进行微调，进一步改进了网络性能。这使得 ZoeDepth 在现有 SOTA 方法的基础上有了明显的提高，可以应用于多种任务和应用领域。

本系统将采集到的视频数据按采样率为 30 帧的速率采集图片，使用 ZoeD-M12-N 模型进行训练，得到每张图片的深度图，与系统结合，还原三维场景。

#### 4.4.2 三维重建

三维建图是计算机视觉领域中备受关注的的一个重要研究方向，它涉及到从二维图像或其他形式的传感器数据中，推断出三维场景的几何结构和拓扑信息。三维重建是指从二维图像或视频中恢复物体或场景的三维几何形状和外观信息的过程。目前，建立三维物体模型的方法主要有三种：一种是基于图像的三维重建，通过多方向的传感器(如相机)拍摄物体，利用机器视觉技术从图像中构建出三维数学模型；另一种是运用传统的几何建模技术手动建立三维模型；第三种则是基于点云的三维重建方法，通过对物体进行扫描或拍摄，生成该物体在点云视角下的三维模型。

本系统的默认建图为基于点云的稀疏建图，有着明显的缺点，就是很难描绘出障碍物的存在。因此，本系统考虑到定位时需要考虑到周围环境的具体情况，使用 PCL 建图工具快速构建三维模型，使用的是基于图像的三维重建方法。

三维重建需要图像的原信息和深度图信息，而本系统采集的数据配合 Zoedepth 模型训练深度数据；得到深度信息后，由于深度信息的限制，为了提高基于深度图像的应用效果，需要对深度图像进行增强处理，如去噪和修复；经过预处理后的深度图像呈现二维形式，其中每个像素值表示物体表面到传感器的直线距离。接下来进行点云配准，将不同时间、角度、照度下获取的多帧图像匹配到统一的坐标系中，以公共场景为基准。配准后的深度信息仍然是散乱无序的点云数据，仅能提供部分物体信息，因此需要对点云数据进行融合处理，以获取更精细的重建模型。最后，通过表面生成，使用体素级方法处理原始灰度体数据，构造物体的可视等值面<sup>[22-23]</sup>。流程如图 4.9 所示：

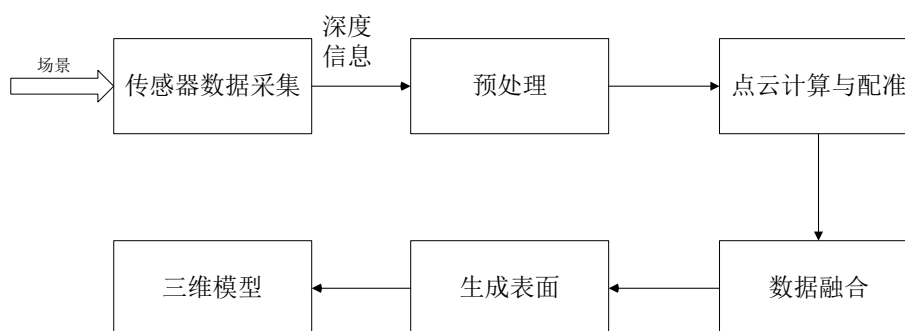


图 4.9 三维重建流程

### 4.4.3 位姿三角化与增强实现

三角化（Triangulation）是计算机视觉中的一项重要技术，用于从多个视角的观测中恢复三维场景的几何信息。它是通过在不同视角下观察到的图像特征之间进行几何推理来计算三维点的位置。

三角化的基本原理是利用多个视角下的视线交汇关系来确定三维点的位置。这可以通过找到至少两个视角的图像特征之间的对应关系，并使用相机的投影几何关系来实现。三角化在计算机视觉中具有广泛的应用，包括三维重建、立体视觉、SLAM 等领域，它是从多个视角中获取三维信息的重要手段之一。求解三角化的方法有多种，本系统使用 SVD（奇异值分解）法求解三角化。

在 SLAM 中，主要使用三角化来估计像素点的距离，即通过某个特征点  $P$  在不同帧图像中的像素位置，计算出该特征点在三维空间中的坐标，从而获取特征点  $P$  的深度信息。

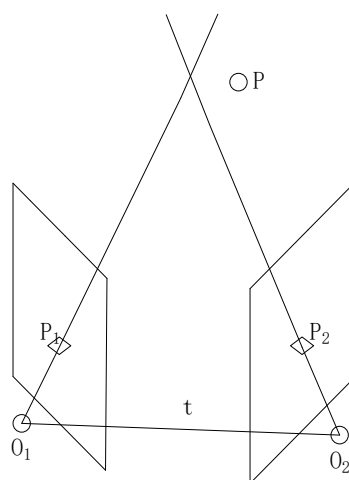


图 4.10 三角化

如图 4.10 所示， $O_1$  和  $O_2$  分别表示为两个相机光心。在图像  $I_1$  中存在特征点  $P_1$ ，同理在图像  $I_2$  中存在对应的特征点  $P_2$ 。由理论可知，直线  $O_1P_1$  和  $O_2P_2$  应该在场景中的一点  $P$  处相交。这个点  $P$  表示两个特征点对应的地图点在三维场景中的位置。

增强现实(Augmented Reality, AR)是一种技术，将虚拟的数字内容叠加到真实世界中，创造出与现实环境交互的增强体验<sup>[24]</sup>。AR 技术通过识别和跟踪真实世界中的物体

或场景，并在其上叠加虚拟的图像、视频、3D 模型或其他数字内容，使用户能够在真实环境中与虚拟内容进行交互，它为用户提供了与数字内容的实时交互和沉浸式体验，同时保留了真实环境的感知和交互性。本系统为了增强系统的可操作性，增加了 AR 的功能，将 3D 模型增添到跟踪线程中。将 AR 技术与 SLAM 相结合，可以为用户提供更加丰富、实时的视觉体验，并拓展 SLAM 应用的功能和应用场景。无论是在导航、场景理解、可视化还是协作方面，AR 都可以提供增强的效果，提升系统体验。

## 4.5 误差分析

误差分析是评估和量化系统或算法的性能和准确度的过程，误差分析是非常重要的，因为它可以帮助我们了解系统的稳定性、准确性和鲁棒性。本系统使用 EVO 进行误差分析，EVO 是一款用于视觉里程计和 SLAM 轨迹评估工具。它具有两个核心功能：轨迹绘制和误差评估。通过 EVO，可以将算法输出的相机轨迹可视化绘制出来，以便更好地理解相机在场景中的运动轨迹。同时，EVO 还能评估算法输出轨迹与真实轨迹之间的误差，从而量化算法的性能，依此提高算法的精度和稳定性。EVO 支持多种数据格式进行测试，包括主流的 KITTI、EuRoC、TUM 以及 ROSbag 包，同时支持这些数据格式之间的相互转换。

EVO 自带有文件格式转换工具，只需执行相应的命令即可转换。常用的 EVO 命令如表 4.1：

表 4.1 EVO 常用命令

命令	功能
evo_ape	评估绝对位姿误差
evo_rpe	评估相对位姿误差
evo_traj	显示轨迹、输出轨迹文件、转换数据格式
evo_res	比较 evo_ape 或 evo_rpe 生成结果
evo_fig	重新打开序列化图
evo_config	evo 工具全局设置和配置文件操作
correct_scale	单目尺度变换



本课题采用 EuRoC 数据集作为分析数据，该数据集提供了轨迹真值，可以与自己系统运行的关键帧轨迹做详细对比，通过表 4.1 的命令将结果可视化分析，将相机的轨迹可视化绘制出来，以便直观地观察相机的运动路径，用于分析系统在轨迹输出的性能。同时，将误差结果可视化输出，以便直观的分析误差分布的范围以及分布频率。

#### 4.6 本章小结

本章系统阐述了系统设计的流程以及实现的功能，分析了系统实现工程中需要完成的工作，解释了涉及到的部分原理。将系统运行的整个过程做了分析，给出相机标定的详细过程，实现三维重建和 AR 模型的引入，并详细给出误差分析工具的详细使用方式以及作用。

## 第 5 章 系统测试

在本章会对系统进行实际场景实验结果展示，并根据实验的数据结果做系统分析，通过系统测试的数据结果分析，评判系统性能状况。

### 5.1 官方数据集结果分析

#### 5.1.1 EuRoC 数据集

本系统首先使用官方数据集对总体性能进行评估，由于官方数据集自带轨迹真值，可以直接对比实际运行效果与传感器采集的轨迹。可以根据对特征点的追踪、地图构建的准确度来检验系统效果是否优良。

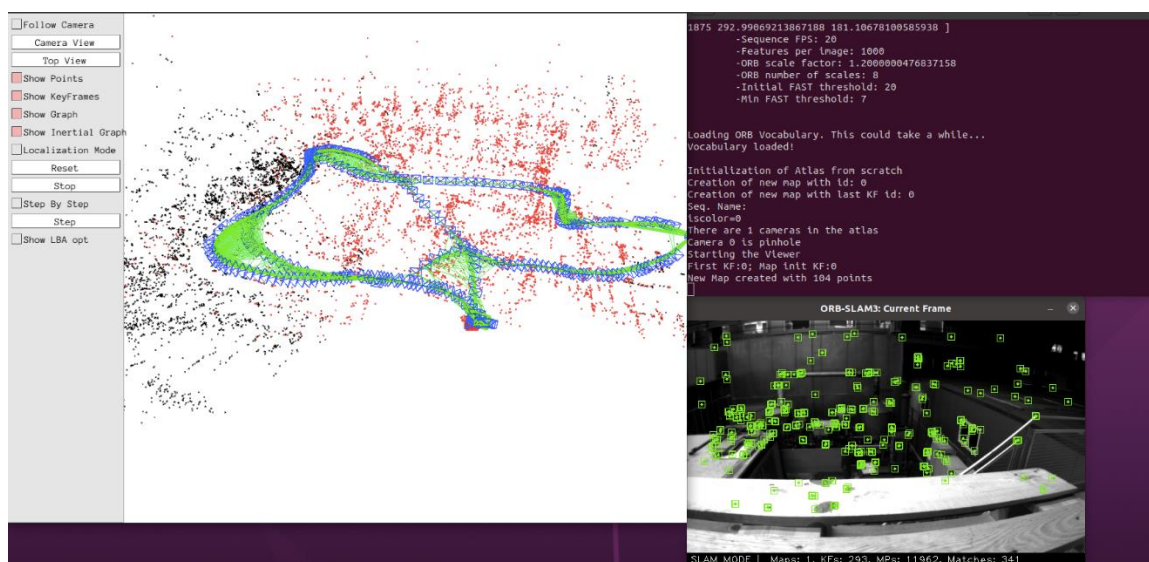


图 5.1 单目无人机数据

根据图 5.1 中的输出结果分析，基本完成了对传感器周围环境的稀疏建图，特征点数量较多，可以判断出本系统角点检测效果良好。

#### 5.1.2 结果评估

对官方数据集的运行结果使用 EVO 工具进行分析，包括运行轨迹的绘制，误差的计算，结果的比较。

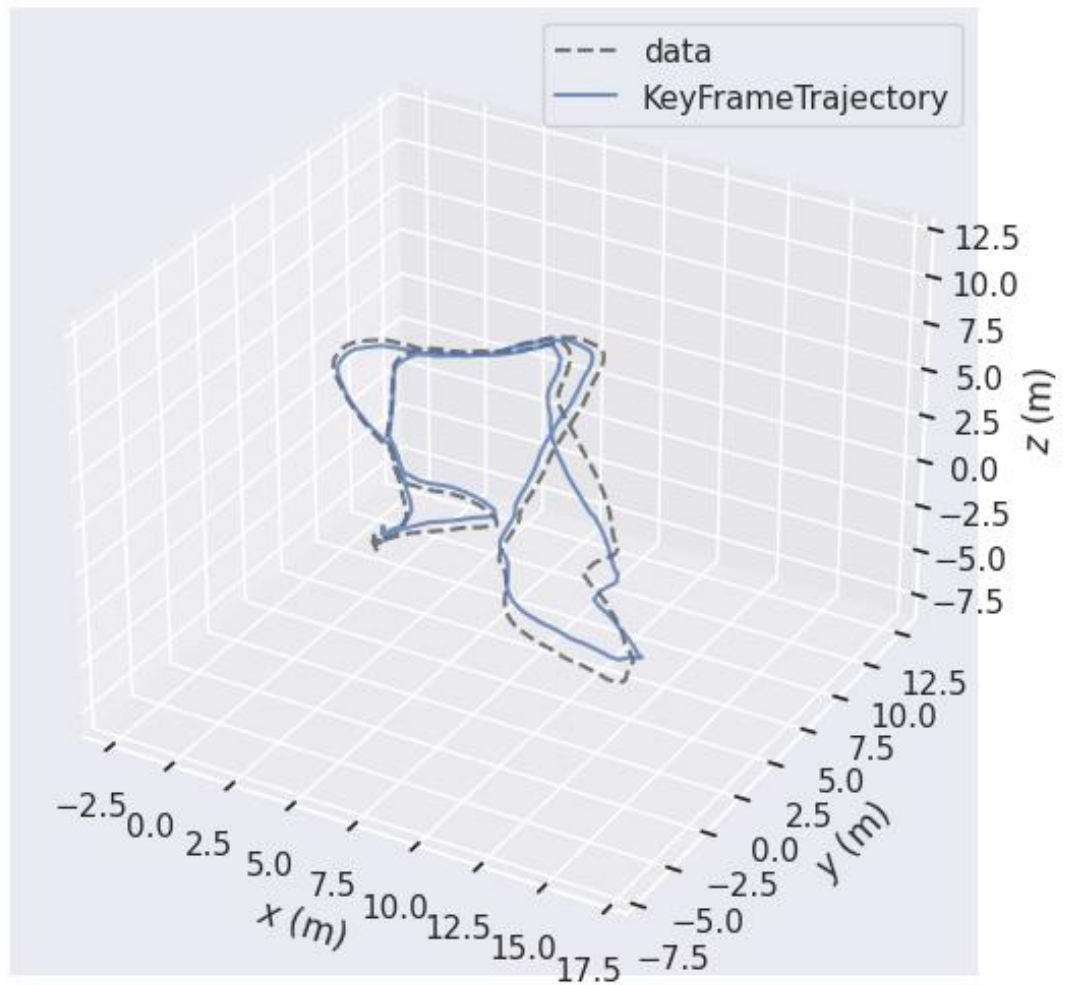


图 5.2 EVO 轨迹输出

表 5.1 单目输出值

	真值	系统值
路径/m	97.593	91.291
时间/s	111.055	111.055

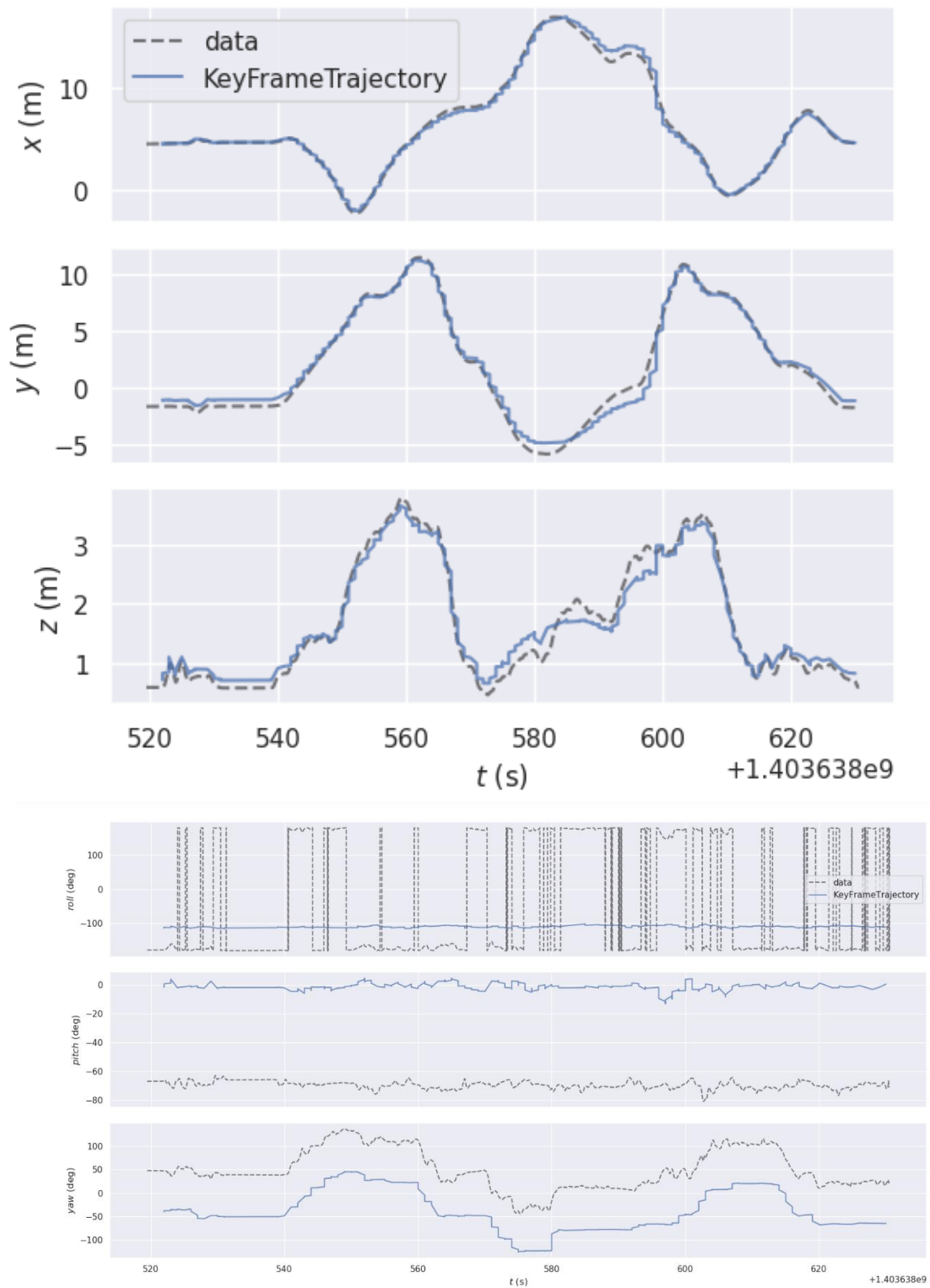


图 5.3 方向误差

根据图 5.2 和表 5.1 输出结果分析，系统运行的轨迹与相机实际运行的结果略微有偏差。具体数据如图 5.3，其中在 x 轴方向上的误差较小，其次是 y 轴上的偏差，z 轴上的误差最大。可能的原因是无人机数据采集时比较抖动，同时算法的对特征点提取过于密集，使得系统对周围环境的感知过于集中导致误差产生。由表 5.1 计算，系统误差度为 6.4%，用于定位，该误差在可接受范围内。

### 5.1.3 位姿计算

在计算位姿时，由于官方数据集的轨迹真值文件与本系统运行后产生的关键帧数据文件在记录上有区别，因此在计算位姿之前需要将二者的数据进行统一，将数据集的真值文件格式修改为系统所需要的格式，这个可以利用上文所述的“evo\_traj”功能。其次，数据转换完成后需要将二者的时间戳格式统一，由于数据量比较庞大，直接写一个 Python 脚本完成数据格式的转换。

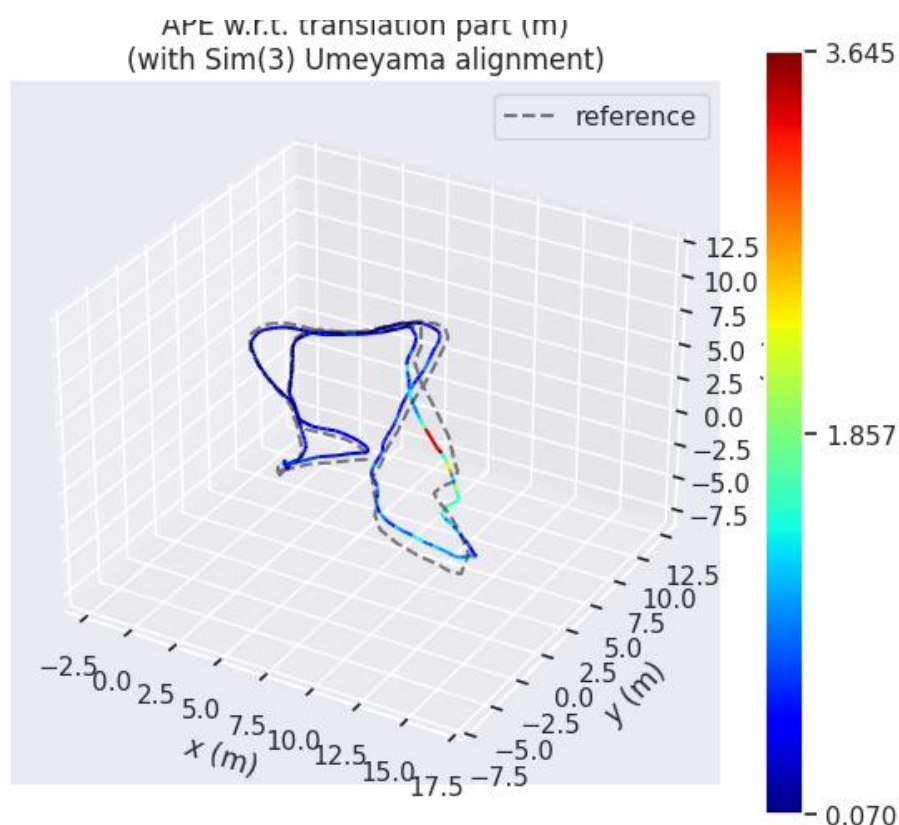


图 5.4 绝对误差

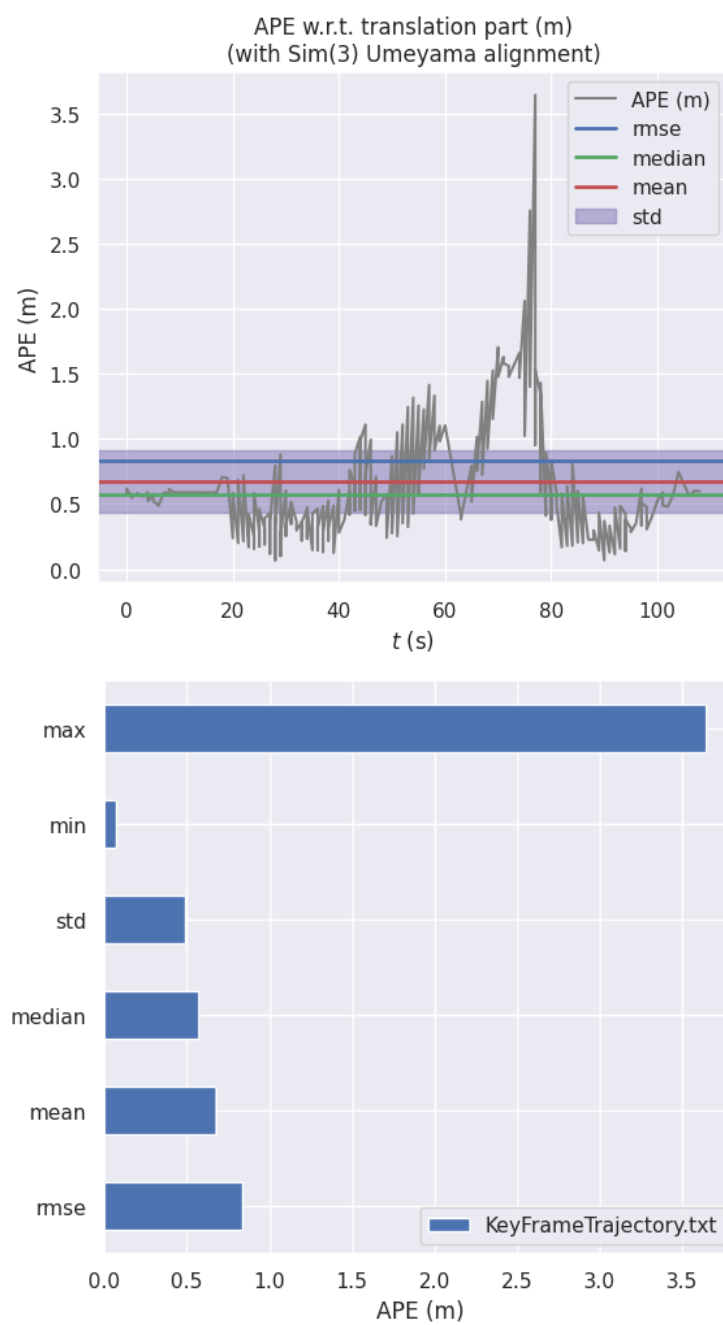


图 5.5 绝对误差表

表 5.2 绝对误差值

max	min	mean	median	rmse	std	sse
3.6446	0.0696	0.6754	0.5708	0.8326	0.4868	219.0756

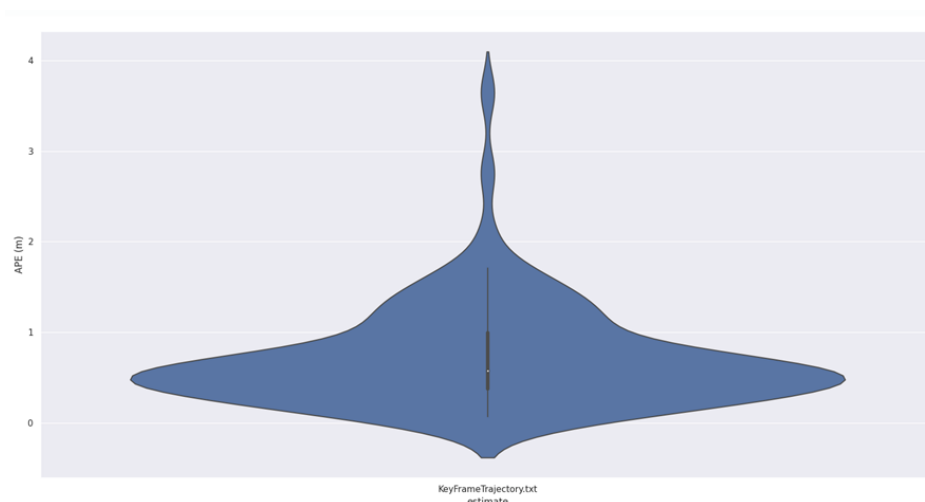


图 5.6 绝对轨迹误差数值频率分布

由图 5.4，图 5.5 和表 5.2 分析，本系统的绝对轨迹误差平均值为 50 厘米，整体定位精度为 6.4%，即路程为 100m 的偏差为 64cm。且根据误差数值的频率分布图 5.6 所示，本系统的误差分布主要在 1m 以内，因此，可以得出系统的定位精度在厘米级。

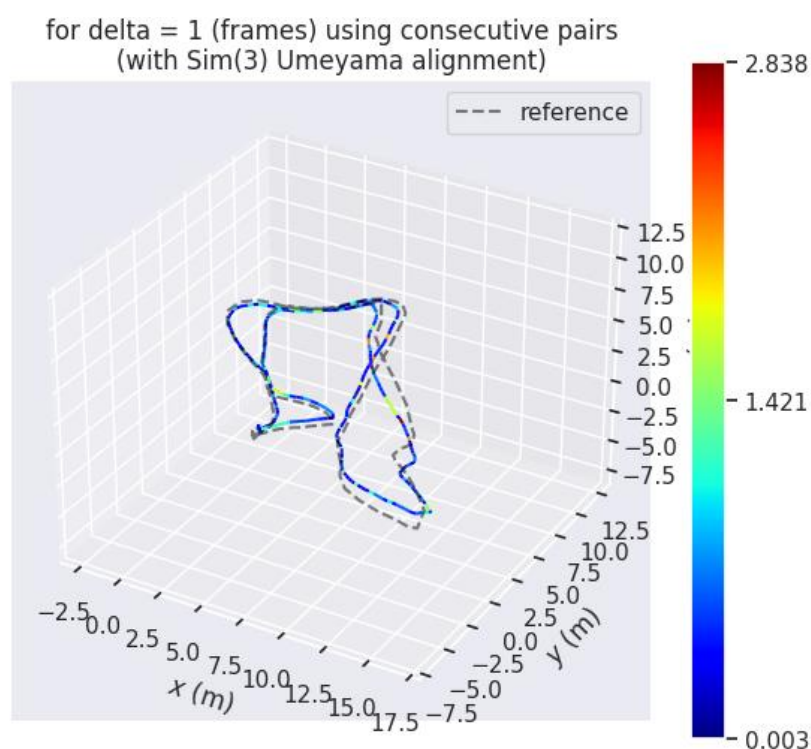


图 5.7 相对误差

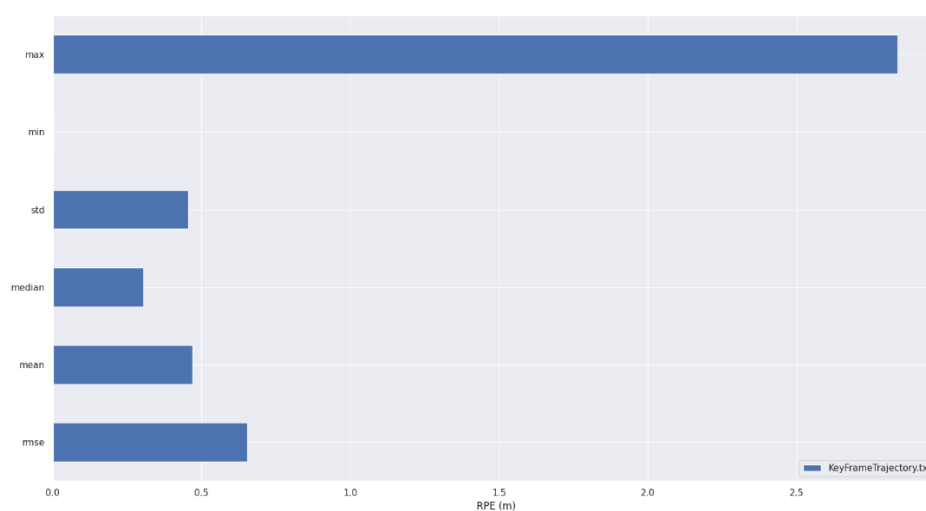
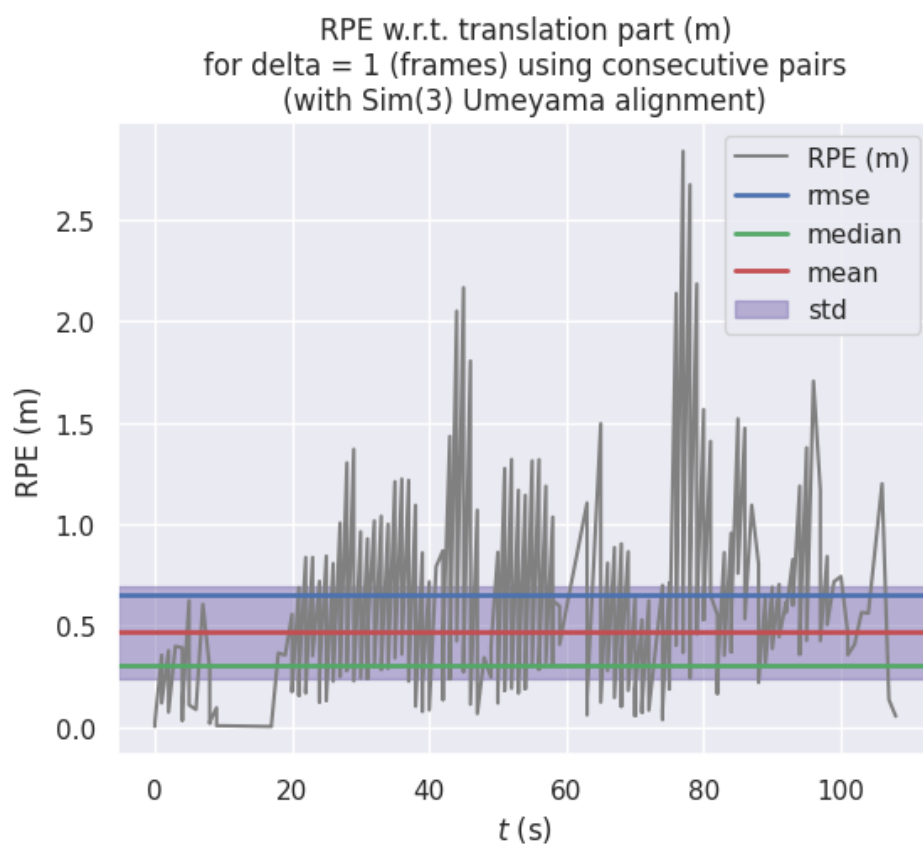


图 5.8 相对误差表

表 5.3 相对误差值

max	min	mean	median	rmse	std	sse
2.8383	0.0033	0.4687	0.3045	0.6540	0.4560	134.7428



综合图 5.7，图 5.8 和表 5.3 以及上述实验结果可以得出结论，不论是相对位姿误差还是绝对位姿误差，都可以看出系统的最小误差几乎可以忽略不计，即使是最大误差也相对较低。通过对算法的输出分析，系统对特征点检测有时不稳定，综合原情况分析，是上位机算力不足从而导致有时可能出现追踪失败，故而会重新加载地图导致误差出现。

通过在公用数据集上的测试结果表明，本文设计的系统在硬件条件良好的情况下可以用于实际环境的定位与建图。

同时，为了分析不同相机模型对定位精度的影响，本系统在测试中增加了双目相机作为对照组，实验结果如图 5.9 和表 5.4 所示：

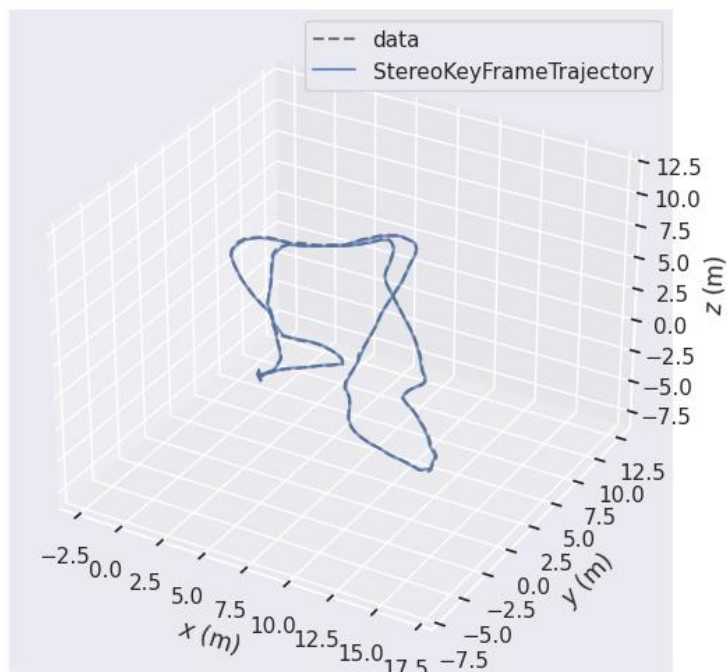


图 5.9 双目轨迹分析

表 5.4 双目输出值

	真值	系统值
路径/m	97.593	96.113
时间/s	111.055	108.000

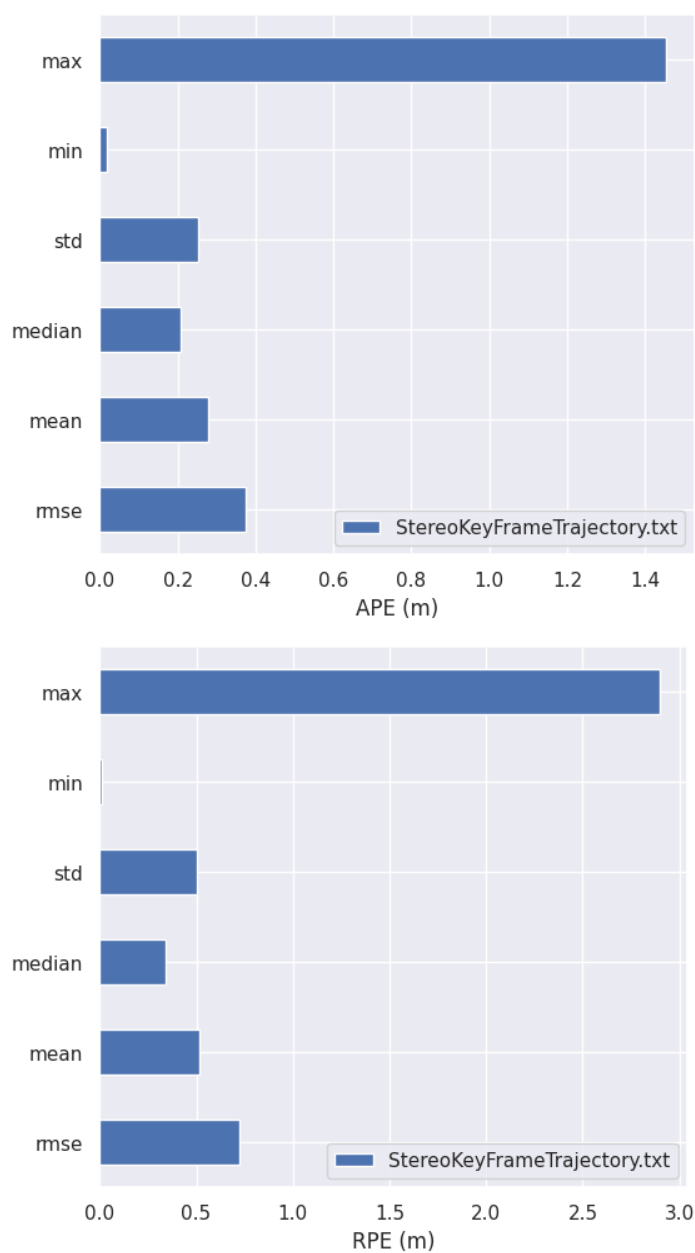


图 5.10 双目误差值

将相机换为双目相机后，由图 5.10 可以得出系统性能得到明显提高，误差降低到 1.5%，误差几乎可以忽略不计。同时不管是相对轨迹误差还是绝对轨迹误差，在双目相机下的数值均有所下降。因此，本系统对双目相机作为传感器将会有更优良的性能。

## 5.2 室内数据集结果分析

SLAM 系统的一大应用就是可以用于室内小场景的定位。本系统采用学校语音楼一楼做为数据集，测试室内建图准确度，同时为了评估本系统的性能效果，用 GPS 定位系统作为对照组。传感器实际运行路程为 60 米左右，路线为一个位移为 0 的室内封闭环形。

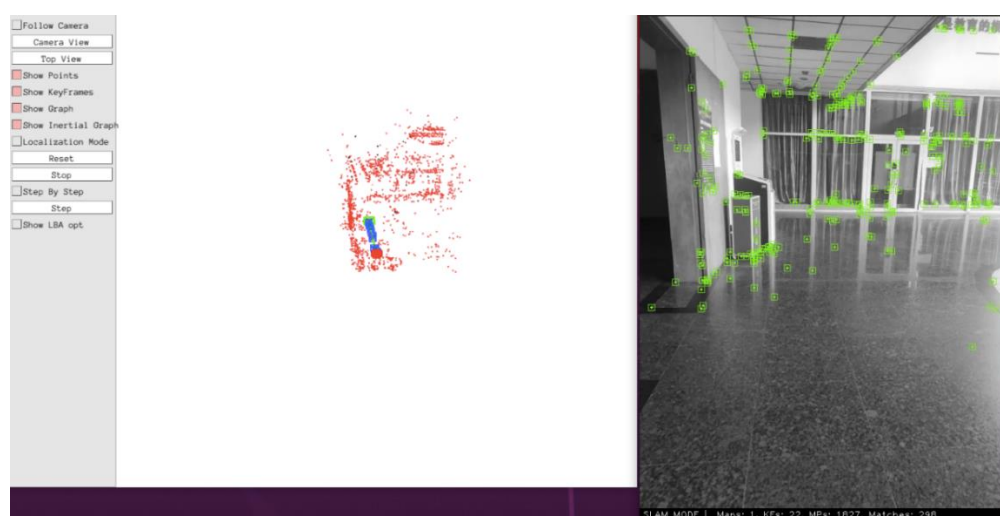


图 5.11 特征点提取

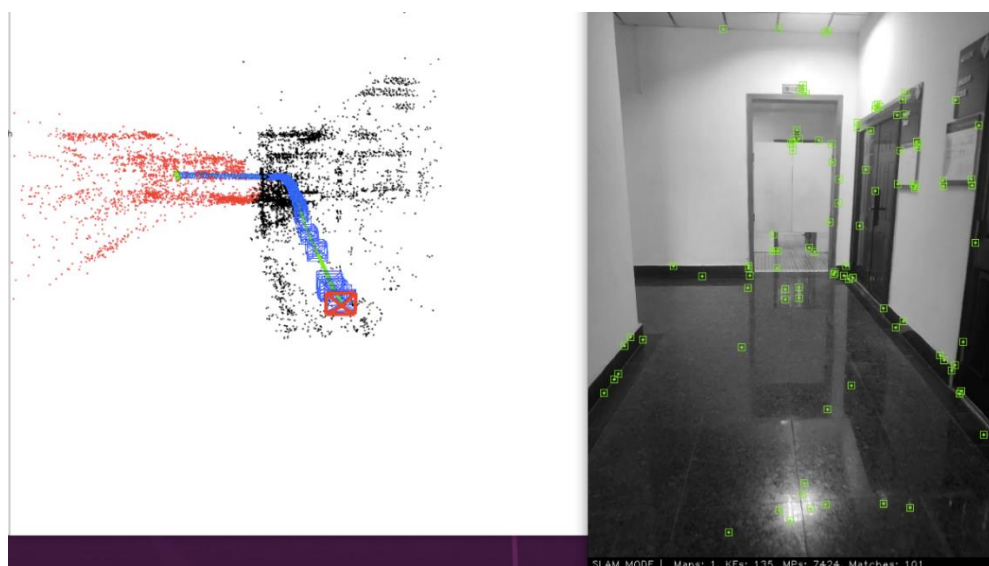


图 5.12 黑暗条件下建图

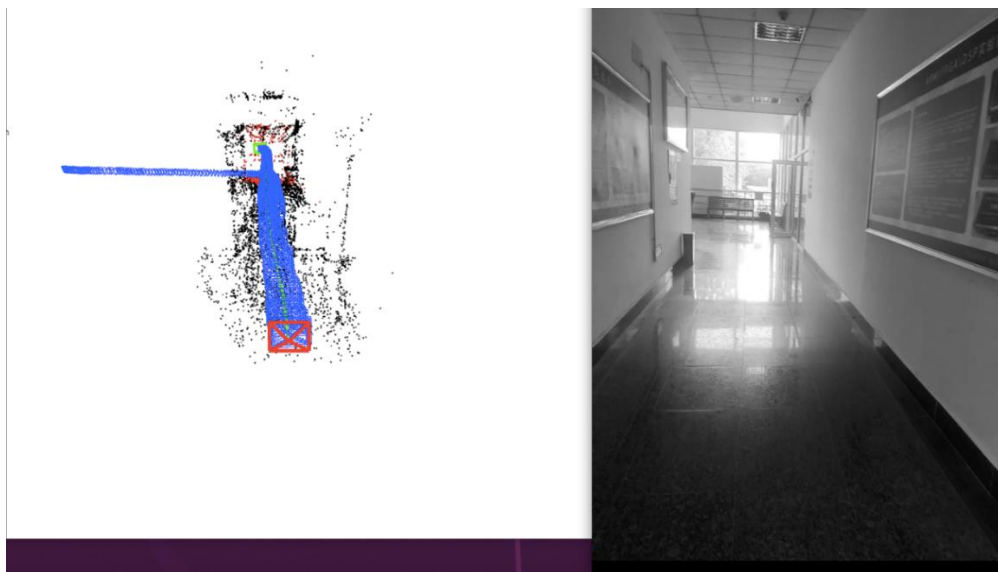


图 5.13 特征点提取失败

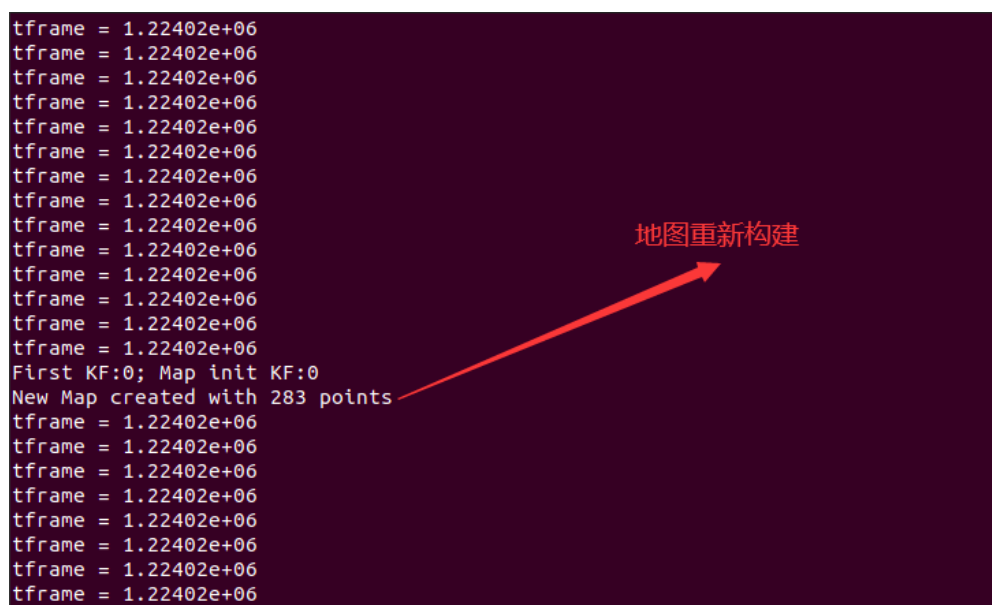


图 5.14 地图重构

如图 5.11，由系统的建图结果可知，在室内场景且环境亮度适中时，系统能够提取到较多的关键点，不会出现定位失败的情况，能够很好的用稀疏点云图还原传感器周围的环境。但是，如图 5.12 和 5.13，当传感器周围环境过于暗淡或者在传感器旋转角度过大时系统会出现特征点跟踪丢失的情况，但是如图 5.14，系统会及时对跟踪失败的情况做出响应，立即进行重定位和地图的重新构建。

综上可以得出本系统在光照环境比较好的场景，具有良好的性能，即使在黑暗条件下出现跟踪失败的情形，本系统也能在快速做出重定位操作。能够很好的绘制传感器周围的环境信息，包括门窗、桌椅等实际物体。

下面，用 GPS 定位系统做对比分析，如图 5.15:

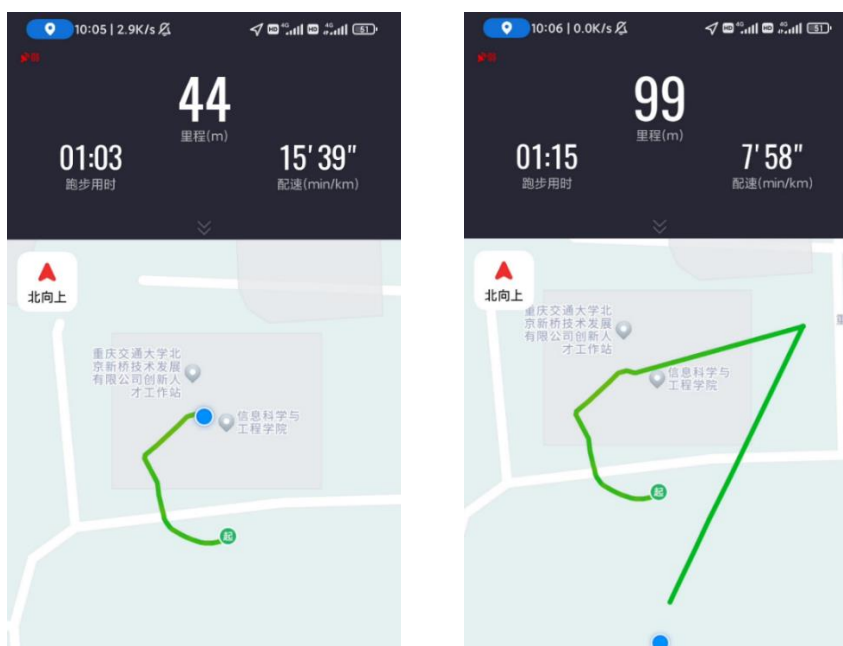


图 5.15 GPS 定位

在同一环境中，根据 GPS 定位的结果显示，在室内 GPS 信号较弱的室内场景，定位效果很差，无法获知传感器周围的具体环境。同时在计算里程时出现较大误差，由 GPS 的信号决定了定位的精度，分析图 5.15 可知，对相同地点的两次定位，数据出现较大误差，且与真实值的误差均在 30% 以上。因此，本系统在室内环境中的定位精度明显高于目前主流的 GPS 定位系统，而且本系统还可以对周围环境进行建图，可以根据建图功能直接获知当前环境的详细情况。

### 5.3 回环检测结果分析

本系统的一个关键的模块就是回环检测，它可以检测到相机是否回到了之前经过的地方，并将这些信息用于优化相机轨迹和地图。因此，评估一个本类系统的另一个重要指标就是看是否有优良回环检测效果，如图 5.16。

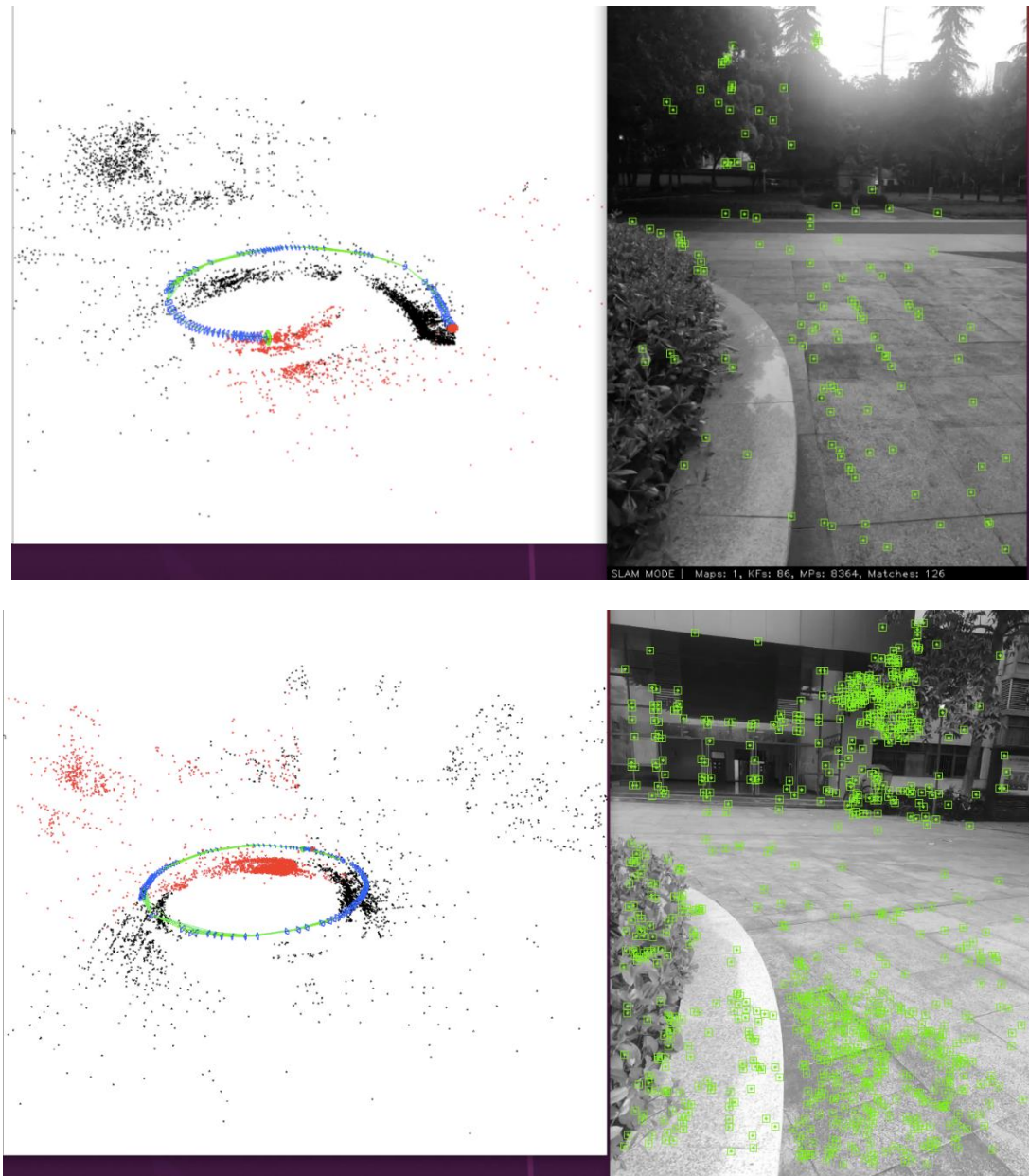


图 5.16 未回环与回环结果对比

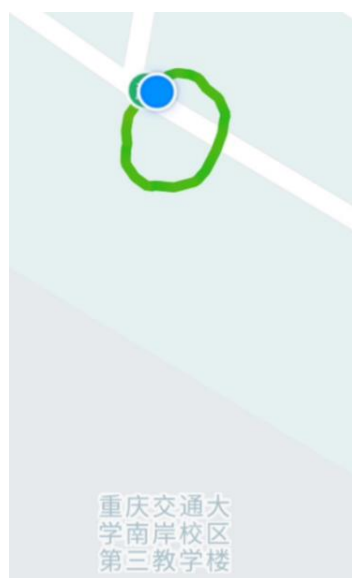
表 5.4 位移量

	帧名	X 位移量/m	Y 位移量/m	Z 位移量/m
初始帧	1224720278559554	0.0000	0.0000	0.0000
结束帧	1224752732902615	0.0349	-0.0431	-0.0145



由三维空间的位移公式以及表 5.4 数据计算可得总位移量为 0.05m，理论上位移为 0，存在较小的误差，在实际应用时可以忽略。根据结果显示，当相机在首次经过花坛圆环时，由于没有先前的位姿信息，系统无法判断是否经过了相同的场景，因此无法实现闭环检测，所以建图轨迹比较没有实现闭合。但是，在有先前的位姿信息后，系统通过特征点信息判断，结合先前的位姿，获知传感器正在经过之前的环境，判断出轨迹为环形轨迹，发现回环之后，则本系统将使用回环约束重新优化相机轨迹和地图，以便更好地估计相机的位姿和场景的结构，实现回环检测功能。由本次检测结果可知，该系统的回环检测效果优良。

本系统以 GPS 定位做对比分析。GPS 在定位信号较强的地方可以对运行的大致轨迹进行绘制，但是由图 5.17 可以看出，整体的轨迹存在漂移现象，起始点并未重合。因此，本系统由于回环检测功能可以估计运行轨迹，很好的判断出运行轨迹，大大提升定位的准确度。



5.17 GPS 轨迹

## 5.4 轨道交通三号线结果分析

本系统采用轨道交通三号线大场景应用分析。轨道交通的运行速度较快，周围环境复杂，适合用于检验本系统在高速、复杂环境下精确度。通过在轻轨的窗口使用手机

采集数据，由于录制条件限制，不能获取到正前方的情况，只能通过窗口获知轨旁附近的环境信息，如图 5.18。

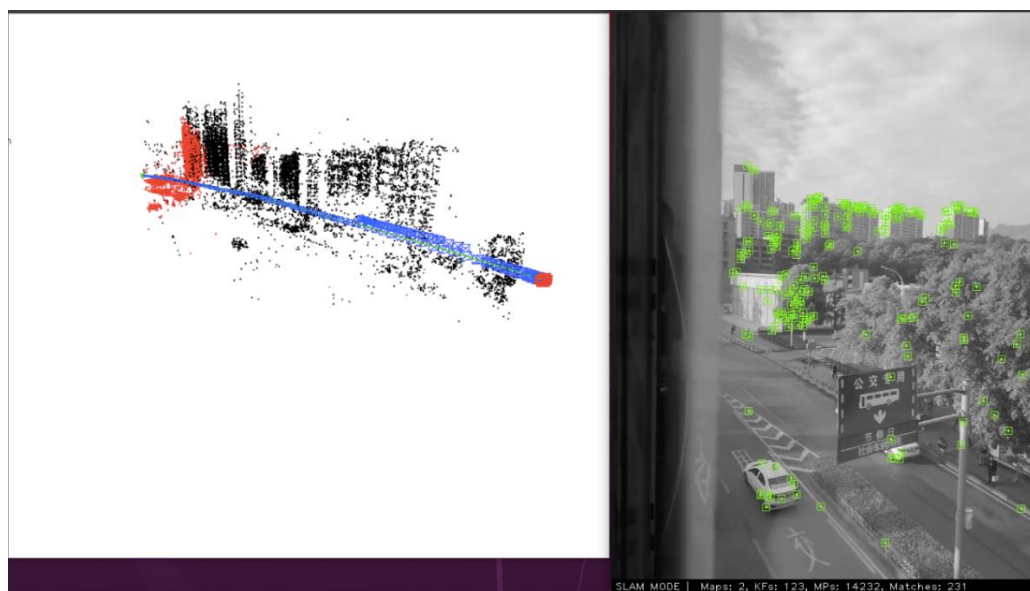


图 5.18 三号线白天特征提取

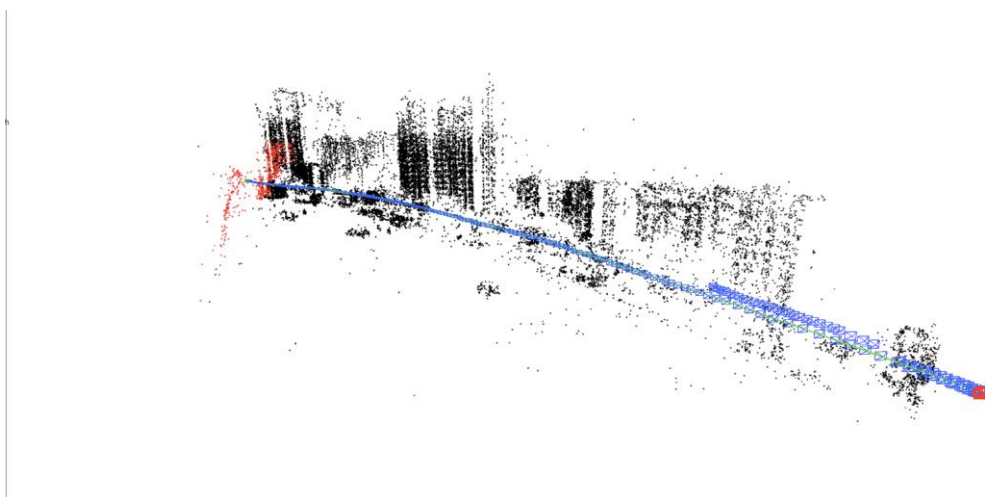


图 5.19 结果图

从结果图 5.19 可以看出在高速运行的轻轨上，系统在运行时出现了跟踪丢失的状况，但是依旧是迅速重构地图。从高到低，均存在特征点，而且对特征点提取的效果很



强，因此对环境还原度较高，地图构建效果良好，可以证明本系统在复杂场景的应用效果良好。

## 5.5 三维重建分析

通过使用单目深度估计模型 Zoedepth 获取到环境的深度图信息，结合关键帧轨迹信息，三维还原现实环境，可以用于分析实际场景，弥补了稀疏点云建图的不足。通过分析看出系统能够实现三维重建，并且效果显著，能够很好的还原实际场景的环境。

首先需要获取原始数据的深度信息，通过深度学习获取，其次，将深度信息与原始数据配合关键帧信息，实现三维位姿估计。输出如图 5.20 和 5.21 所示：

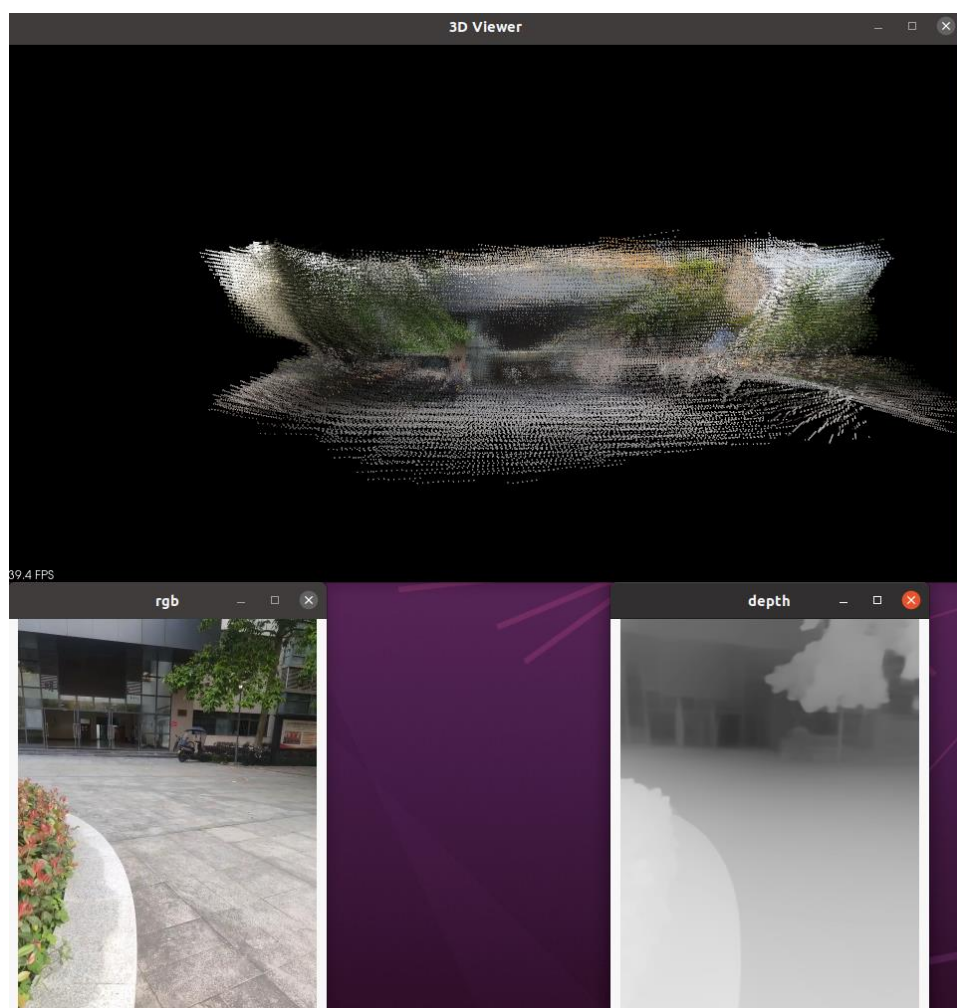


图 5.20 三维还原过程

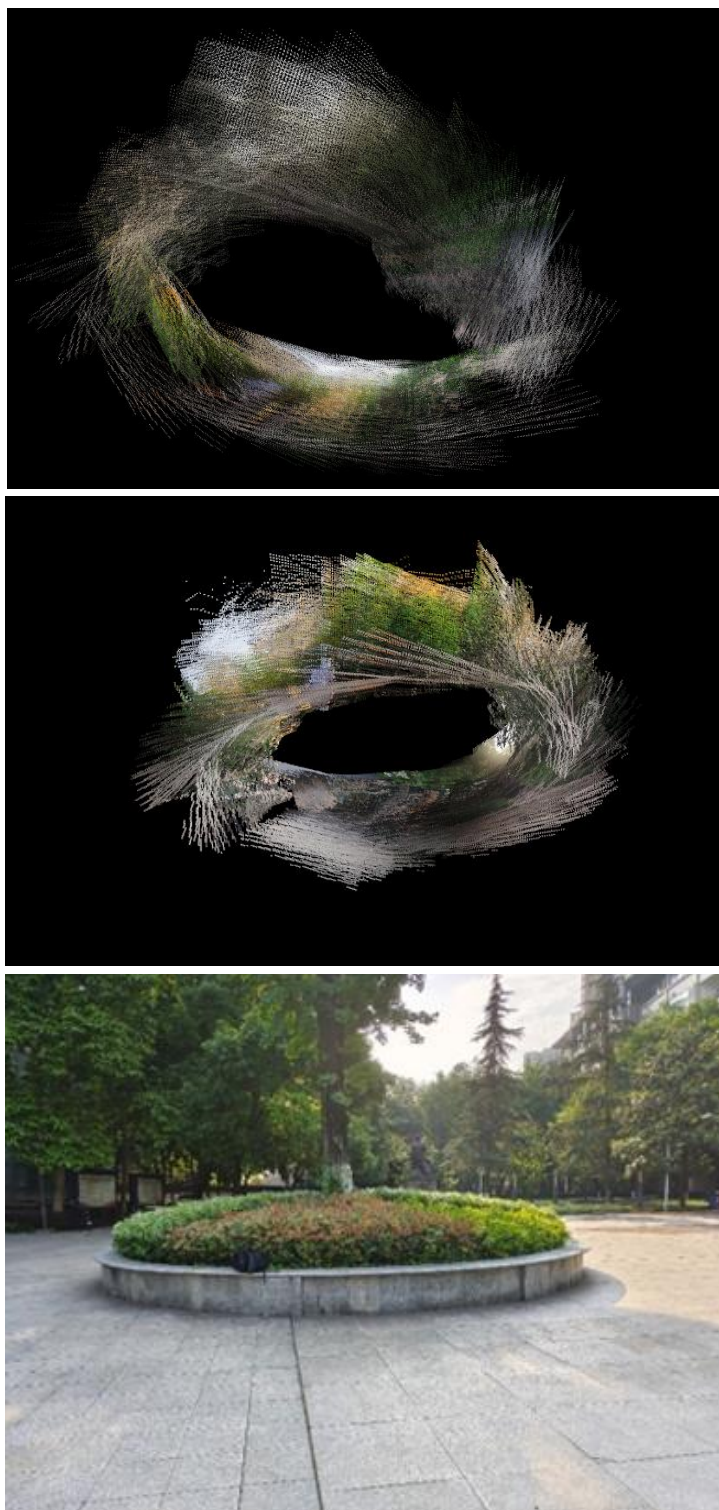


图 5.21 三维还原结果与原场景

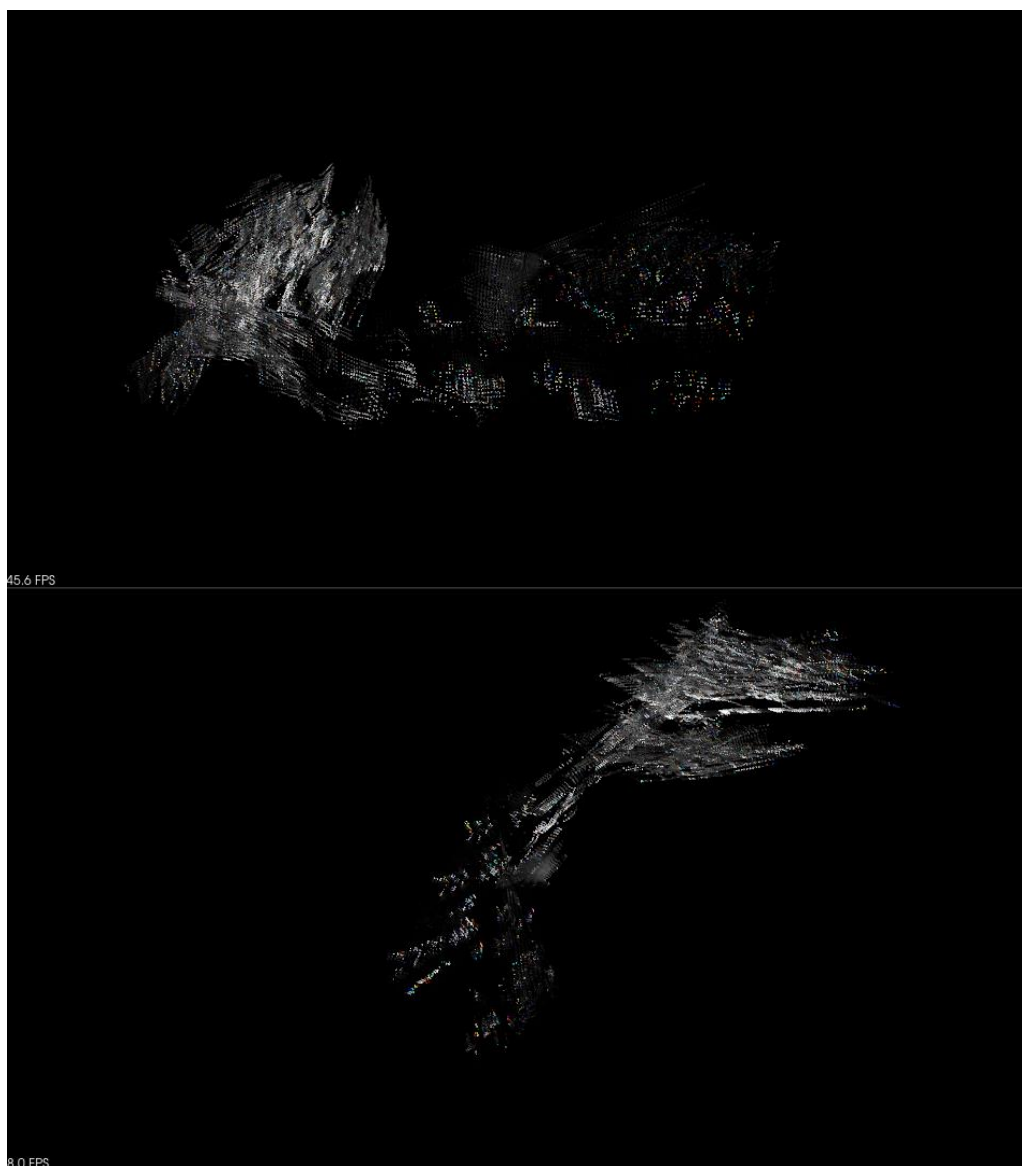


图 5.22 官方数据集三维重建

如图 5.22 根据对不同数据集三维重建，可以对输出进行分析，本系统能够较好的还原传感器周围实际场景，但是根据特征点的提取差异，在使用关键帧进行还原的过程中会出现关键帧不匹配的情况，导致还原的精确度偏低，目前本系统的精确度需要进一步改善，可以通过后续算法的优化，增加关键帧与图像的匹配度，提高三维还原的精度。

## 5.6 三角化与 AR 模型

在 SLAM 中,三角化的精确性和稳定性对于地图的质量和定位的准确性至关重要。通过有效地估计特征点的深度,SLAM 系统可以更好地还原场景的三维结构,提高地图的精确性和稳定性。同时,三角化的结果也可以用于位姿估计、回环检测等其他 SLAM 过程中的关键计算。本系统可视化了三角化的过程,可以详细分析点云之间的匹配情况。

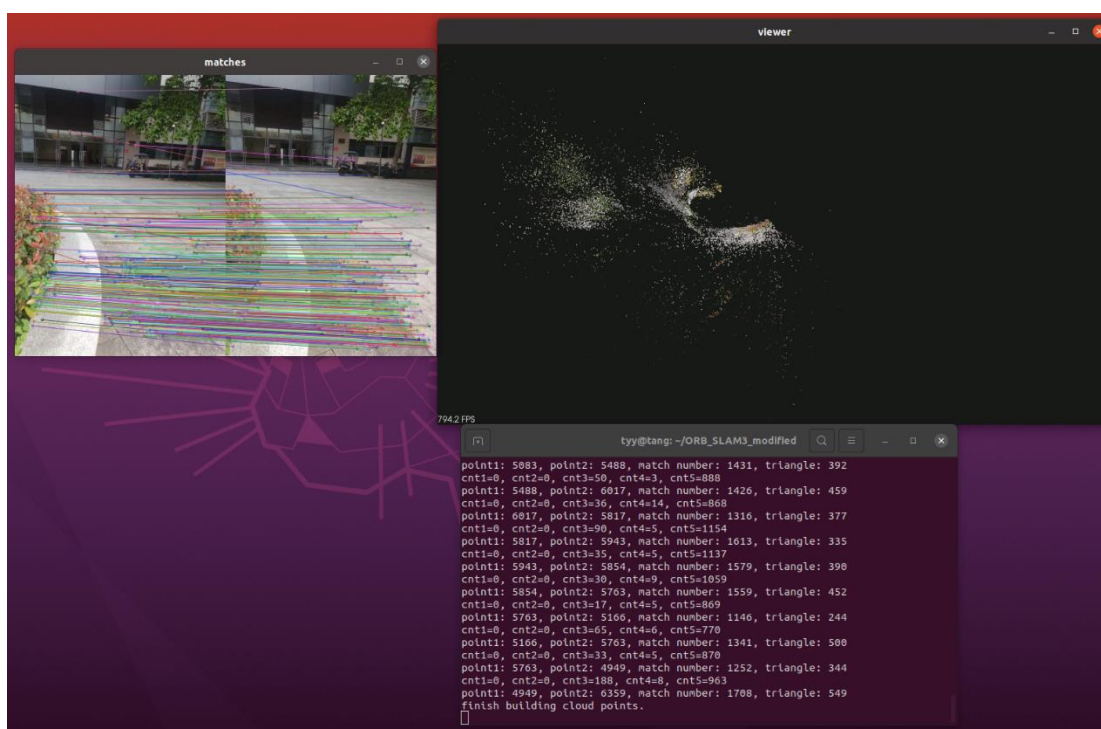


图 5.23 三角化

根据图 5.23 显示,通过在连续帧之间进行特征点匹配,并利用三角化方法估计他们的三维位置,可以计算相机的运动,且点云配准效果效果优良,能够较好的提取到匹配的特征点,从多个视角的数据中恢复出三维点的位置信息,用于重建三维结构、估计相机位姿,但是根据三维重建的效果,未能完全实现所有关键帧匹配,三角化三维重建的效果劣于上文用深度估计实现三维重建的效果,可能的原因和计算机性能以及特征点特征点提取数量有关。



在视觉系统中，AR 可以将虚拟的三维模型与实际场景进行融合，能够直观地观察和理解地图中的物体、结构或场景，增强系统于用户的交互性。本系统设置了 4 种 AR 模型，可以根据需要，额外增加自己的模型，如图 5.24 所示：



图 5.24 AR 模型

## 5.7 本章小结

在本章对系统测试的结果详细呈现。包括下官方数据集下的轨迹和各类误差结果分析，评估系统性能。在各类场景的测试分析系统性能，包括以轨道交通线路为主的大场景，室内明亮和暗淡的场景，室外环形轨迹场景。根据测试的结果，计算系统能够达到建图和定位的精度，展示后端地图优化算法的效果，呈现三维重建的真实性以及 AR 模型的可交互性。

## 第 6 章 总结与展望

### 6.1 本文工作总结

本次毕业设计是基于 ORB 特征点的视觉定位与建图系统，本设计是一个具有挑战性和意义的课题，该课题的设计可以在室内或室外环境下实现实时的三维建图和相机位姿估计，具有广泛的应用前景。在本次设计中，我通过查阅文献、研究方案规划、功能设计和系统实现等多个方面的工作，完成了基于 ORB 特征的视觉定位与建图系统的设计和实现，并取得了一些进展和成果。

本系统主要是研究单目视觉 SLAM。单目相机有很多的优点诸如价格低廉、安装简单、数据量小等优点被广泛的应用于视觉 SLAM 领域。本系统详细论述了 ORB 特征单目 SLAM 系统的整个系统原理。此外根据 ORB 特征提取法建图存在的不足，本文提出一种基于单目深度估计的三维重建 SLAM 系统。本论文主要工作内容如下：

（1）介绍了基于 ORB 特征的 SLAM 算法原理，对 ORB 特征点法做了详细分析，论述一种新的单目深度估计模型以及在三维重建方面应用；

（2）具体分析了计算机的双系统模式以及 Ubuntu 系统的文件组成，对实现本系统功能的主要三方库做了系统论述，用以搭建实现本设计所需的系统环境。

（3）详细介绍了本系统采集数据所使用的软硬件，并完成对软硬件参数的配置，根据相机成像原理完成对相机参数的标定以及对数据采集软件的开发；

（4）结合 SLAM 算法的基本原理，根据系统稀疏建图的缺点，实现了三维重建的开发工作，同时为了增加系统与用户的可交互性，在系统中设计了 AR 模型；

（5）对系统使用官方数据集做精度测试分析，计算本课题设计的绝对轨迹误差和相对轨迹误差，并将误差数据可视化分析。同时，用另一种相机模式作为对照组分析，为系统性能改善提供一种新的方向；

（6）通过各种类型的自制数据集测试，不断增强本系统定位与建图的准确性，囊括各种生活中的场景，实现高精度稀疏建图和三维建图，并在系统中拓展增强现实的应用。

## 6.2 展望

随着机器人技术和 SLAM 技术的不断发展,ORB-SLAM 系统也在不断地进行升级和改进。对 ORB-SLAM 系统的展望主要包括以下几个方面:

(1)精度和鲁棒性的进一步提高:ORB-SLAM 系统在精度和鲁棒性方面已经有了很大的提高,但在复杂和变化的环境中仍然存在一定的挑战。因此,未来的研究可以继续探索新的特征提取算法、深度学习模型、优化算法等,进一步提高 ORB-SLAM 系统的精度和鲁棒性。

(2)多传感器融合的应用:随着科学技术在传感器方面取得的发展,除了单目相机外,ORB-SLAM 系统还可以与其他传感器进行融合,例如双目相机、激光雷达、深度相机等,以获得更加全面和准确的定位和建图信息。本系统已测试了双目相机,精度得到明显提高。未来的研究可以探索多传感器融合的应用,将 ORB-SLAM 系统应用于更广泛的场景和应用领域。

(3)扩展应用领域范围:ORB-SLAM 系统已经在机器人导航、自动驾驶、虚拟现实等领域得到了广泛的应用。未来可以探索更多的应用场景和领域,例如智能制造、医疗健康等,将 ORB-SLAM 系统应用于更广泛的领域和行业,为实现人工智能提供更加有效可靠的技术支持。

(4)与深度学习的结合:目前也有不少的研究者提出并付出实践将 ORB-SLAM 与深度学习结合,但是在研究成果方面并没有取得特别大的进展。要么是因为使用机器学习法但因为没有较好的硬件条件导致深度计算量不足,要么因为非深度学习法导致精度低下。本系统简单尝试了将之与深度学习模型结合,用于实现三维重建功能,取得一定的成果,但是过于依赖计算机性能。在将来可以探索出一种介于二者之间的方法,加大单目视觉 SLAM 的实际应用性。



## 致 谢

执笔至此，意味着一段人生旅程即将结束，心中顿时颇有感慨。弹指一挥间，我在交大已走过四个春秋，回望过去在学习和生活上的方方面面，不仅有教室里的埋头苦学，也有操场上的汗如雨下。有在学习上收获时的喜悦，亦有在成长中失败时的沮丧。这些点滴汇成了我人生长河中弥足珍贵的一部分。在此分别之际，以最诚挚的心表达我的感激之情。

首先，我要感谢我的导师和指导老师们。他们不仅为我提供了全方位的指导和帮助，还在我的论文设计方向上给了我深刻的启示和指引，他们的辛勤工作和严谨态度让我在学习和研究中受益匪浅。在我遇到挫折和困难的时候，老师总是给予我耐心的指导和鼓励，让我直面困难，重新振作起来；每当我想放弃时，老师总是能悉心与我探讨问题所在，重拾对学术的信心。在老师的指导下，我不仅仅是提高了自己对科研的兴趣，更是学会了如何以积极的心态面对困难。

其次，我要感谢我的父母。感谢他们在我学习的道路上给予我的关心、支持和鼓励，他们的支持和鼓励是我前进的动力和勇气。正是有了他们对我几十年如一日的无私奉献，我才能有机会无忧无虑的在重交完成四年的学习生活。

此外，我还要感谢我的同学和朋友。他们不仅是我的好同学，还是我在学习和生活上的参与者。他们的帮助让我在学习和生活中有了更多的思考和收获，毕业设计能顺利完成同样离不开他们的悉心解惑。

最后，我要向母校致以最诚挚的感谢和敬意。在这里度过的四年时光，不仅为我提供了学习的机会和平台，更重要的是培养了我的思维能力和创新精神。感谢母校为我提供的各种资源，让我能够在学习中不断探索和发现新的知识和技能。在母校的培养下，我不仅收获了专业知识，还学会了如何成为一个不负社会，不负国家的人。在这里衷心祝愿交大在未来的发展中秉承优良的传统，不断创新和进取，为社会培养更多的优秀人才，也能够蓬勃发展、繁荣昌盛，为广大师生营造更美好的学习和生活环境。

## 参 考 文 献

- [1] 刘美生.全球定位系统及其应用综述(一).定位技术发展的沿革[J].中国测试技术,2006(05):1-7.
- [2] 王铁胜.计算机视觉技术的发展及应用[J].信息系统工程,2022,No.340(04):63-66.
- [3] 张海,吴克强,张晓鸥.视觉导航技术的发展[J].导航定位与授时,2017,4(02):1-8.
- [4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel et al, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," in IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874-1890, Dec. 2021.
- [5] 何香玲,张跃,郑钢等.GPS 全球卫星定位技术的发展现状、动态及应用[J].微计算机信息,2002(05).
- [6] 夏禹.北斗卫星导航系统发展与应用[J].电子技术与软件工程,2021(16):25-26.
- [7] Roberts, Lawrence G . Machine perception of three-dimensional solids. Massachusetts Institute of Technology, 1965.
- [8] Marr D. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information[J]. The Quarterly Review of Biology, 1983, 58(2).
- [9] 管叙军,王新龙.视觉导航技术发展综述[J].航空兵器,2014,No.283(05):3-8.
- [10] Sim R , Little J . Autonomous vision-based robotic exploration and mapping using hybrid maps and particle filters[J]. Image and Vision Computing, 2009, 27.
- [11] 陈文佑,章伟,史晓帆等.一种改进 ORB 特征匹配的半稠密三维重建 ORB-SLAM 算法[J].电子科技,2021,34(12):62-67.
- [12] 王宏勇. Ubuntu Linux 基础教程[专著][M]. 清华大学出版社, 2015.
- [13] 姜子鹏,卜凡亮,秦静.PCL 环境搭建与在三维重建中的应用[J].软件导刊,2018,17(11):123-129.
- [14] 王敏.VTK 可视化类库及其应用进展[J].中国科技信息,2010(20):114-115.
- [15] 许宏科,秦严严,陈会茹. 基于改进 ORB 的图像特征点匹配[J].科学技术与工程,2014,14(18).
- [16] 邱笑晨,赵晨旭,张海,许辉.ORB-SLAM 系统优化框架分析概述[J].导航定位与授时,2019,6(03).
- [17] B. Wang, H. Wang, Y. Yu and L. Zong, "ORB-SLAM based semi-dense mapping with monocular camera," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 2017, pp. 1-5.

- [18] 廖泓真,王亮,孙宏伟,刘云清.一种改进的 ORB 特征匹配算法[J].北京航空航天大学学报,2021,47(10):2149-2154.
- [19] Rosten E, Drummond T. Machine learning for high-speed corner detection[C],European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2006.
- [20] 罗会兰,周逸风.深度学习单目深度估计研究进展[J].中国图象图形学报,2022,27(02):390-403.
- [21] 江俊君,李震宇,刘贤明.基于深度学习的单目深度估计方法综述[J].计算机学报,2022,45(06):1276-1307.
- [22] 陈加,张玉麒,宋鹏,魏艳涛,王煜.深度学习在基于单幅图像的物体三维重建中的应用[J].自动化学报,2019,45(04):657-668.
- [23] 王加康.基于双目视觉的三维重建技术研究[D]. 辽宁大学.
- [24] 李京燕.AR 增强现实技术的原理及现实应用[J].艺术科技,2018,31(05):92.