

硕士学位论文

基于 visual SLAM 算法的四旋翼无人机室内 定位研究

**RESEARCH ON INDOOR POSITIONING OF UAV
BASED ON VISUAL SLAM ALGORITHM (学术型)**

徐 铎

哈尔滨工业大学

2017 年 6 月

国内图书分类号：TP242.6

学校代码：10213

国际图书分类号：621.586.4

密级：公开

工学硕士学位论文

基于 visual SLAM 算法的四旋翼无人机室内定位研究

硕士研究生：徐 铎

导师：高云峰 副教授

申请学位：全日制工学硕士

学 科：机械电子工程

所在单位：机电工程学院

答辩日期：2017 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: TP242.6

U.D.C: 621.586.4

Dissertation for the Master Degree in Engineering

**RESEARCH ON INDOOR POSITIONING OF UAV
BASED ON VISUAL SLAM ALGORITHM**

Candidate:	Xu Duo
Supervisor:	Associate Prof. Gao YunFeng
Academic Degree Applied for:	Master of Engineering
Speciality:	Mechatronic Engineering
Affiliation:	School of Mechatronics Engineering
Date of Defence:	June, 2017
Degree-Confering-Institution:	Harbin Institute of Technology

摘 要

当机器人正逐渐从实验室走向广大人民的生活中，无人机是最具有代表性的一种，其在气象探测、公路巡视、勘探测绘、水灾监视等具有广泛的应用。因而无人机在不具有 GPS 信号的情况下的定位就成为急需解决的问题。SLAM (simultaneous localization and mapping) 因其成本低，设备简单适应性强成为一个较好的选择方案。

到目前为止 SLAM 有许多算法已经问世，但是在 3D 空间内，采用摄像机作为传感器的领域仍然有待研究。目前只有较少的基于视觉的 visual SLAM 系统能够在采用较高性能处理器的条件下实时的完成构建地图并对机器人进行定位。本文首先综述了国内外室内定位方法和 visual SLAM 的研究现状，发现了目前的研究所存在的一些问题，并确定本文的主要研究内容。

本文分析了传统算法的优点和不足并在现有研究的基础之上提出了一种新颖的算法，该算法搭载了 Linux 开发平台上，采用华硕出产一种 RGBD 相机的 Xtion pro 作为视觉传感器，以基于优化理论的算法作为核心。前端采用基于 ORB 特征的 Tracking 算法，中间构建地图的过程之中融合惯性测量单元(IMU)得到的信息，后端闭环检测采用基于 ORB 的词袋模型，并进行整体相机的位姿优化。最终实现构建未知空间地图和机器人的定位。

相对于传统算法，本文提出的算法具有以下优点：1.采用改进的相机标定算法，显著的提升了深度相机获取对应像素点的空间点位置的准确性，为后续的 SLAM 算法提供相对精确的深度信息。2.将 IMU 的信息融合进 visual SLAM 的过程中，通过 IMU 可以得到无人机飞行时的位置和姿态，为迭代提供相对精确的初始值，并且提高机器人的定位精度。3.初始化部分抛弃了传统的基于基础矩阵的做法，采用了相对可靠的五点算法 4.采用多线程并行编程，SIMD 向量化编程等高级编程技术提高了运算性能。

该算法可以实时的获取图像并进行机器人的定位和地图构建，仅需无人机及其上 RGBD 相机，无需依赖其它设备就可以完成任务，相对于其它定位算法而言具有简单可靠的优点。实验证明，该方法在室内短时间飞行的定位误差基本在 5%以下，能够满足定位需求。

关键词：机器人定位并建立地图；ORB 特征；Xtion pro；惯性测量单元

Abstract

Robots are moving from the laboratory to our life. Unmanned aerial vehicle (UAV) is the most representative one. It is widely used in meteorological survey, highway inspection, exploration and mapping, flood monitoring and so on. The positioning of UAV without GPS signals becomes an urgent problem. Because of its low cost, simple equipment and strong adaptability, SLAM has become a better choice.

So far, there are many SLAM algorithms that have been published. But the algorithms in the 3D space, using camera remains to be studied. At present, there are fewer visual SLAM systems that can build maps and locate robots in real time under the condition of using high performance processors. Based on the existing research, a novel algorithm is proposed. The algorithm is equipped with the Linux development platform, using Xtion Pro, a RGBD camera produced by ASUS as the visual sensor, and the depth error is calibrated. The algorithm based on optimization theory is taken as the core algorithm. The front end uses Tracking algorithm based on ORB feature. When building the map, IMU is used to fuse into the system. Loop closing in the back end use bag of words based ORB feature. Finally, the construction of unknown space maps and robot localization is achieved.

This algorithm has the following advantages over traditional algorithms: 1. With the improved camera calibration algorithm, the accuracy of the spatial point position of the corresponding pixel is greatly improved. 2. By fusing the information of IMU into visual SLAM, we can provide relatively accurate initial values for the iteration and improve the localization accuracy of the robot. 3. The initialization part abandons the traditional method by using fundamental matrix and use a relatively reliable method by five point algorithm. 4. Advanced programming techniques, such as multi-thread parallel programming and SIMD programming have been used to improve computing performance.

The algorithm can acquire images, locate the robot and construct the map in real time only by UAV and a RGBD camera on it. Compared with other localization algorithms, it has the advantages of simplicity and reliability. Experimental results show that the positioning error of this method is less than 5% within a short time, and it can meet the requirements of positioning.

Keywords: SLAM, ORB feature, Xtion pro, IMU

目 录

摘 要.....	I
Abstract.....	II
目 录.....	III
第 1 章 绪 论.....	1
1.1 课题研究的背景、目的和意义.....	1
1.1.1 课题研究的背景.....	1
1.1.2 课题研究的目的是和意义.....	1
1.2 国内外在该方向的研究现状及分析.....	2
1.2.1 基于滤波器的贝叶斯算法.....	3
1.2.2 基于优化的算法.....	5
1.3 本课题主要研究内容.....	8
第 2 章 Xtion Pro Live 相机标定算法.....	10
2.1 引言.....	10
2.2 Xtion pro live 相机简介.....	10
2.3 Xtion pro live 相机成像模型.....	13
2.4 标定和畸变校正方法.....	15
2.5 误差和畸变补偿实验.....	19
2.6 本章小结.....	20
第 3 章 基于 ORB 特征的前端特征提取与关键帧提取.....	21
3.1 引言.....	21
3.2 ORB 特征简述.....	21
3.3 特征匹配方法.....	23
3.4 融合 IMU 进行位置估算.....	27
3.4.1 IMU 状态模型.....	27
3.4.2 IMU 观测和噪声模型.....	30
3.5 与传统方法对比实验.....	32
3.6 本章小结.....	33
第 4 章 基于李群代数的优化方法.....	35
4.1 引言.....	35
4.2 传统 Bundle Adjustment 方法.....	35

4.3 优化的 Bundle Adjustment 方法.....	36
4.4 算法验证实验.....	38
4.4 本章小结.....	41
第 5 章 无人机室内定位实验.....	42
5.1 引言.....	42
5.2 实验硬件.....	42
5.3 实验软件.....	44
5.4 数据集测试实验.....	46
5.5 无人机室内实际测试.....	50
5.5.1 无人机定距直线飞行实验.....	51
5.5.2 无人机随机飞行实验.....	52
5.6 本章小结.....	53
结 论.....	55
参考文献.....	56
攻读学位期间发表的学术论文.....	60
致 谢.....	62

第 1 章 绪 论

1.1 课题研究的背景、目的和意义

1.1.1 课题研究的背景

无人机又称空中机器人，它由机体内部的自主控制器执行控制及飞行任务，大多数无人机由机架本体，动力系统，制导系统，控制系统，起飞装置等构成，能通过地面人员操控或者通过自身的飞行控制系统进行飞行任务^[1]。由于无人机在天空中飞行，因此其具有隐蔽性好，机动性强等特点，并且其体积小，造价低廉等特点^[2]。

无人机作为一种近现代出现的先进的军事装备，以其得天独厚的“无人”优势，大大的降低了战争的伤亡，十分符合和“零伤亡”的现代战争理念。现代几次局部战争使得无人机崭露头角，得到了飞速的发展。1921 年英国研制出可真正参与实战的第一架实验靶机。此后，无人机技术随着遥测遥控及相关技术的发展而不断进步，最终无人侦察机得以问世。自上世纪 80 年代以来，无人机进入了一个全新的发展时期。据估计，目前全球使用的各种类型的无人机的总数约有 6.8 万架，2020 年将剧增至 12 万架^[3]。无人机在民用上也大有可为，它可应用于场区监控、气象探测、公路巡视、勘探测绘、水灾监视、航空摄影、交通管理、电力线路查巡、森林火灾防救、农业植保等^[4]。

在民用方面，无人机已经和航拍紧紧的结合在一起，普通人完全可以通过手持遥控器获取无人机从高空获取的图像，在出门游玩的过程中，携带一个无人机，就可以将景色收纳与眼底。无人机的应用不仅局限于航拍，还可以起到短程快速运输的作用。在反恐方面，用无人机进入未知室内环境探测成为一项新的应用，在某些未知而又可能对人身造成伤害环境中，需要小型无人机进入探测并记录当前所在位置，从而勾勒出未知环境的地图，侦查危险物品所在位置，从而为后续人员的进入做前期探测，从而保证后续人员的安全^[5]。

1.1.2 课题研究的目的和意义

目前四旋翼无人机越来越向着微型化的方向上发展，其飞行空前从最初的丛林、城市逐步扩展到建筑物内^[6]。然而无人机在上述的复杂环境中飞行时，通常会发生信号遮挡、噪声干扰的问题，上述问题会导致无人机上通常所用的

全球定位系统（GPS）的信号很难被接收到，即使能够接收到也会变得很不稳定。所以四旋翼无人机的位姿参数便会很难准确的获取，这会对无人机的定位和导航提出了很大的难题。因此，如何利用除了 GPS 以外的传感器（例如激光雷达，彩色摄像头，RGBD 摄像头等）对四旋翼无人机在复杂的环境下的定位就成为了各个国家的研究重点。

在军事方面，无人机主要应用于远程的战场中。2001 年，在阿富汗战争中，美军首次使用无人机执行了对地攻击任务；2003 年，在伊拉克战争中，美军同样部署了无人机对战场进行监视、情报侦察以及对地攻击等任务^[7,8]。

在安保或反恐领域当中，不可避免要遇到无法接收到室外 GPS 信号的情况。在此情况下无人机安装的 GPS 接收机便无法读取 GPS 信号，也就失去了自身的定位能力，因此急需一种能够在室内接收不到 GPS 信号的情况下任然能够进行对无人机进行定位的方法，并且该方法能够获得稳定的获取室内图像，并对无人机进行比较准确的定位。

而目前的室内定位方案主要依赖于无线脉冲，或者 2.4G 或 5G，zigbee，蓝牙 WIFI-RFID 等通讯技术^[9]，原理上还是依赖于 3 点定位即同时接受 3 个基站发出的波，计算出该点到 3 个基站的距离，从而实现室内定位。这就需要提前布置好 3 个或者 3 个以上的基站，并对 3 个基站进行精度较高的标定算法，一旦被定位的物体距离 3 个基站较远，或者基站在该物体的同一侧，就会导致精度出现较大误差，而在未知环境中很难在短时间内同时提前布置 3 个及以上基站，以上所提出的方法都是针对已知环境的室内定位解决方案^[10]，因此 visual slam 作为核心的室内定位解决方案便应运而生。

而基于 visual slam 的机器人定位技术即可以实时传回摄像机拍摄所得到的图像，使外部人员在保证安全的前提下得到未知环境的详细信息，又可以对机器人进行定位的同时建立未知环境的 3D 点云地图，得到环境的具体尺寸，使其成为在得不到 GPS 信号的未知环境的情况下的不错的选择。而且无需布置基站，只需在无人机上携带深度相机即可完成上述任务。

1.2 国内外在该方向的研究现状及分析

SLAM 问题最早即源于机器人领域,是指搭载特定传感器的机器人,在没有先验环境信息的情况下,通过机器人的运动过程构建环境的增量式地图^[11],早期的 SLAM 技术一般是离线处理的,后来随着技术的发展出现了实时的 SLAM 技术。SLAM 在理论上按照数学模型不同,大概可以分为两类:一类为基于贝叶斯的滤波器的方法,采用机器人自身里程计的数据作为先验信息,传感器得

到的数据作为后验信息进行滤波；另一类是基于最小二乘的优化方法，将传感器和机器人里程计的数据进行加权并使其代价函数误差最小。其中基于贝叶斯的滤波器的方法可以分为三种：扩展卡尔曼滤波（EKF）^[12]，粒子滤波算法（PF）^[13]和 Fast SLAM^[14]。而在两种数学方法中，我们最常用的还是用基于图优化的方法，在达到相同效果的情况下，图优化的方法的运算量要小于滤波的算法，这主要是因为基于滤波的算法一旦有新的标志点或机器人运动点加入的时候，需要更新整个协方差矩阵，而基于图优化的算法中，只需更新最近的几个机器人的状态和能够被当前状态观察到的标志点，从而减少了运算量^[15]。

1.2.1 基于滤波器的贝叶斯算法

基于滤波器的算法是 SLAM 早期阶段被引入的算法，其中具有代表性的是 EKF-SLAM 和 FAST-SLAM。所有基于滤波器的 SLAM 算法都是依据贝叶斯公式，通过前一步机器人的位置信息和当前机器人的控制信息以及此时传感器返回的路标信息两者相互融合进行滤波，最终得到一个最优的结果。

基于滤波器算法的理论可以将 SLAM 看做如下问题^[16]：

$$\hat{x} = \arg \max p(x|z, u) \quad (1-1)$$

即求解在整个机器人移动的过程中，在通过传感器得到的观测值和机器人控制参数的条件下，机器人位置参数和空间特征点的位置信息的最佳结果。

当前状态是由前一个位置信息加上控制信息作为先验条件，而当前观测值作为后验条件相互融合从而得到最佳的机器人状态变量和空间特征点的位置信息。

1.2.1.1 基于卡尔曼滤波的 SLAM

卡尔曼滤波（Kalman filter）一种基于误差符合正态分布的模型，利用线性系统状态方程，通过观测系统输入和输出数据，在满足系统无后效性的前提下，对线性系统状态进行的最优估计的算法^[17]。

卡尔曼滤波器的关键之处在于其能够进行反馈控制，主要包含两个主要过程，预估和校正。即在线性系统中，先通过状态转移方程对整个系统的状态变量进行估计，同时修正误差协方差估计的值，之后计算卡尔曼滤波的增益，通过增益对先验估计值进行修正，建立起状态变量的后验估计值，这就是预估-校正过程。

预估和校正的两个过程是循环交替进行的，预估过程得到的结果将被用于校正过程的初始值，而校正过程得到的结果将被用于下一个时间阶段的预估过

程的初始值，这样循环往复计算下去，只需计算当前状态的先验和后验值，而之前的状态对现在并不产生影响。

卡尔曼滤波有两个前提条件：1. 必须是线性系统；2. 误差分布必须是正态分布，并且误差项之间必须相互独立，然而往往真实的 SLAM 系统并不是一个线性系统，误差近似为正态分布，移动机器人通过里程计得到的先验信息和通过视觉，激光雷达得到的后验信息都是非线性的，这极大的限制了卡尔曼滤波在 SLAM 中的应用，因此扩展卡尔曼滤波便被引入了 SLAM 领域。

1.2.1.2 基于扩展卡尔曼滤波的 SLAM

一般的卡尔曼滤波有一个很强的假设前提，即只考虑线性系统，现实中，这个假设大大局限了卡尔曼滤波的应用，所以人们又提出了“扩展卡尔曼滤波”（Extended Kalman Filter）来应对非线性系统的情况。当系统为非线性系统时，扩展卡尔曼滤波相对于卡尔曼滤波的优势在于其能够通过计算偏微分导数来近似的将非线性系统线性化，之后便利用卡尔曼滤波器的原理来进行预估-校正的计算。将 EKF 运用到 SLAM 中的算法即为 EKF-SLAM。

扩展卡尔曼滤波和卡尔曼滤波最大的区别就在于卡尔曼滤波当前状态的状态变量可以由前一个状态的状态变量经过线性变换得到，而扩展卡尔曼滤波则是由一个非线性函数得到。同样观测方程也是由观测的非线性函数替代了观测的变换矩阵。

只有当非线性函数的方程接近线性时，同时噪声误差近似满足正态分布时，EKF 的滤波效果才会比较理想。不仅如此，我们还要选取比较好的算法初值，否则会导致整个算法的发散，无法求出正确的最优解。

1.2.1.3 基于粒子滤波的 SLAM

粒子滤波的思想基于蒙特卡洛法，以某事件出现的频率来指代该事件的概率，可以用在任何概率分布的系统模型中。其核心思想是在系统重采样中，以随机抽取的状态粒子的概率分布来近似的模拟系统的状态分布。

粒子滤波大概分为以下五个步骤：

状态初始化：用大量粒子模拟 x_{k-1} ，该粒子的分布满足在状态空间中随机分布。

状态预测：根据状态转移方程，对每一个粒子根据状态转移方程可以计算出一个预测的结果，这些预测的结果在空间中的分布即 x_k 的近似概率分布。

校正阶段：状态预测环节中得到的预测的结果与真实的结果往往存在偏差，根据偏差大小确定粒子的权重，偏差越小，权重越大，最后把粒子的权重进行归一化。

重采样：根据粒子权重对粒子进行重新采样，权值越高的粒子越容易被抽中，不但要保留权重大的粒子，也要保存一部分权重小的粒子。保证样本的丰富性。

滤波：用重采样之后的粒子带入状态转移方法重新进行计算，得到系统状态的近似概率分布。

与传统 EKF 算法相比粒子滤波算法更加高效，尤其对于传递函数非线性，和误差分布并不明确的场合。

1.2.2 基于优化的算法

使用贝叶斯滤波器的 SLAM 方法虽然具有成熟的理论，但是基于贝叶斯滤波器的方法最大的问题就是运算量较大，面对规模较大环境时显得无能为力^[15]。同时也会受限于理论所使用的模型，比如基于卡尔曼滤波的算法只能用在线性系统之中，误差模型一定为正态分布，而扩展卡尔曼滤波则需要传递函数线性度较高，即一阶泰勒展开与原式较为接近。

近些年，SLAM 问题的研究中地图的范围逐渐扩大，基于图优化的方法逐渐占据主流。图优化的核心思想为将空间中的标志点和机器人的位置姿态作为节点，即节点为状态变量和观测变量，而图中的边表示状态变量之间，状态变量和观测变量之间的约束关系^[18]。最终的目标是使整个图处在最佳约束状态。一般来讲基于图优化的 SLAM 一般分为两个部分，前端线程和后端线程。前端线程负责获取传感器的信息，构建图的节点和边，而后端线程则是负责对前端线程构建的节点和边进行优化^[19]。在后端线程中对传感器得到的机器人位置信息和空间中的 3D 点信息进行优化，从而得到最佳的地图信息，该算法的过程框架如图 1-1 所示。

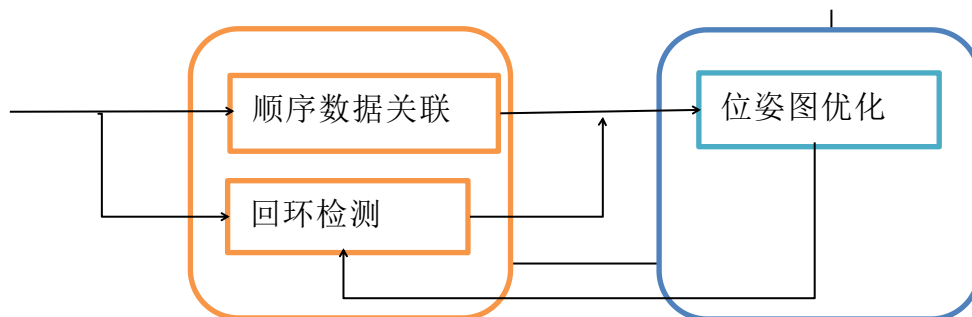


图 1-1 基于优化算法的流程图

1.2.2.1 前端数据关联

前端数据关联是前端线程的主要工作，其主要目的是对传感器输入的数据进行最初的处理，除去噪声，并将传感器所得到的数据抽象化为状态变量和观测变量以及它们之间的约束，并传递到后端线程，一般来讲，这个部分对运算的实时性要求较高，必须能够做到对每一次传感器得到的数据都能够完整的处理，但并不是所有的传感器都会传递到后端线程之中，因为时间上相距很近的几次传感器数据具有很多的重复信息，这样就会加大后端线程的处理负担。往往前端数据关联部分会抽取关键帧，即具有足够代表性的，与之前信息重复性较小的关键信息。

目前数据关联主要是依照时间顺序构建，整个地图记为 m ，在某一时刻的状态变量为 x_k ，该时刻所能观测到的地图上的特征记为 y_k ，而 x_k 可以由上一时刻 x_{k-1} 经过控制变量 u_k 得到，整理结果如图 1-2。

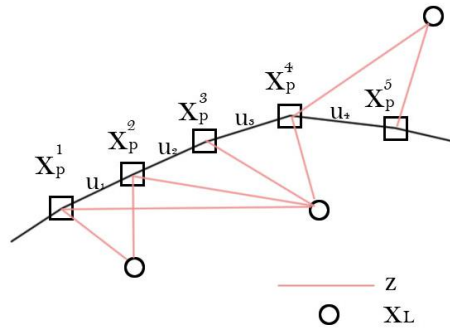


图 1-2 前端数据关联

这样就会按照时间顺序得到一个无向图，相邻的状态变量之间存在边，某个状态变量能够观测到地图上某个特征点，则在该两个节点之间也存在一个边。之后便可以将该图传递到后端线程进行优化。

1.2.2.2 图优化

图优化部分是基于优化的 SLAM 算法的核心部分所在，在前端数据初始处理完成并将图传入后端图优化线程中，这时候便可以启动图优化部分，对前端得到的数据进行高精度的融合。

我们可以把前端线程得到的地图抽象化为一些节点和边的无向图，如图 1-3。

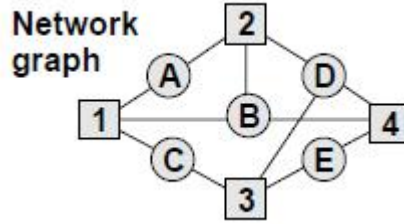


图 1-3 图优化示意图

其中 1 2 3 4 分别为机器人所在的位置，ABCDE 分别代表特征点，如果机器人在某个位置能够看到某个特征点，则在两个节点之间存在一条边。每个边存在一个代价函数。按照传感器的模型，特征点的位置和状态变量，我们可以估计出当前传感器所得到的信息，然而在实际的 SLAM 之中，传感器所测量得到的测量信息和通过传感器的模型，特征点的位置和状态变量预测得到的信息一定会不同，这就存在误差。图 1-5 中每条边上都会存在一个误差函数，而图优化的最终目的就是使整个图的每条边的误差函数之和最小。

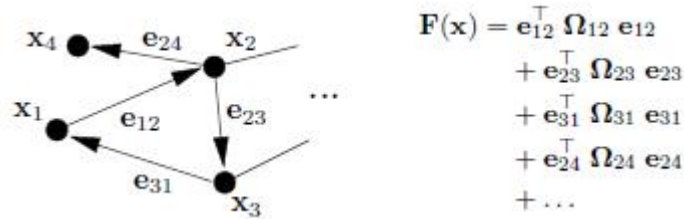


图 1-4 图优化与代价函数

上图为整个图的代价函数，其中 e_{ij} 是某条边的代价函数，而 Ω 是协方差矩阵，而 $F(x)$ 就是最终得到的整个图的代价函数。由于整个代价函数是非凸函数，我们并不能采用凸优化的方法解决这个问题。在学界上，一般会采用一些非线性优化方法来，比如 Levenberg-Marquardt 算法^[20]，高斯-牛顿迭代法^[21]等，通过给定初始值之后进行反复迭代得到局部最优解。

1.2.2.3 回环检测

回环检测是 SLAM 中非常重要的一环，因为 SLAM 算法在原理上也是增量式里程计，即前一步的误差会以某种形式累计到后一步的计算上。虽然具体到某一步的误差较小，但是经过长时间的积累误差就会累计到影响优化结果，甚至出错，无法进行下去的地步，当机器人再一次运行到之前运动过的地方时，理论上讲，两次得到的机器人坐标应该是完全相同的，然而在实际过程中，由于控制噪声和观测噪声的存在，两次得到的结果并不完全一致，甚至会出现较大误差^[22]。这时候就需要回环检测。

回环检测包含两个部分，第一个部分是检测两个位置为同一位置，另一个部分是对整个图进行优化^[23, 24]。

检测同一位置的常用方法是用词袋模型^[25]（Bag-of-Word），即首先从大量图片中提取大量的特征，采用 k-means++ 的方法对其进行聚类，图片中相似的特征被分为同一类，相当于字典中的一个“词”，利用词对图像进行描述。在具体表示某个图像的过程中，首先对图像提取特征，然后将所有特征按照之前得到的词进行分类，计算整个图像中每个词的个数多少，通过每个词的个数组成的向量来对图像进行描述。在比较两个图像的过程中，我们只需对代表两个图像的向量进行比较即可。

而对于回环检测的优化部分则是调整路径上每两个相连接的状态变量之间的边的关系，将最后的误差均匀的分配到参与回环检测的每一个相邻状态变量之中，使误差的和最小。

1.3 本课题主要研究内容

SLAM 发展至今经历了以滤波算法到优化算法，从 2D 以激光雷达作为传感器到 3D 以摄像机作为传感器的转变。发展至今，SLAM 在研究上还有许多点还有待发掘。在 SLAM 中的传感器方向上目前主要集中于如何快速，准确，稳定的提取出空间中的特征。无论是基于 2D 还是 3D 的 SLAM 算法都需要提取稳定的特征点，而传感器的选择，传感器的精度，以及前端特征提取的算法部分都有着很多的解决方案，而采用不同的传感器，不同的特征提取算法会使整个 SLAM 系统无论在运算性能方面还是在构建地图的精度方面都存在着非常显著的差异。就 visual SLAM 而言，传感器一般会有两种选择：RGBD 相机和普通 CCD 相机，两者各有千秋，RGBD 相机能够得到像素点对应的深度信息，能够解决空间 3D 点的问题，但对前端运算能力有较高的要求，很难满足实时性。而普通 CCD 相机信息量相对于 RGBD 相机而言较小，对平台的运算性能要求也相对较低，但是其无法得到真实的尺度信息，仅凭一个单目相机是无法解决真实世界的尺度问题^[26]。

在地图表达部分中目前主要应用的是栅格式地图，但栅格式地图对储存容量要求较高，另一种比较新颖的地图是拓扑式地图，它并不记录图像中有哪些点，而是将点抽象成线段，记录线段中的初始位置和终止位置，这样会极大减少储存容量^[27, 28]。

由于 SLAM 算法目前在精度上和实时性上距离实际应用仍然有一定的距离。随着微小型嵌入式平台的运算性能的提升以及 GPU 的大规模应用。类似于

NVIDIA Jetson TK1, TX1, Intel NUC 等等高性能嵌入式平台的问世, SLAM 算法在实时性方面正在越来越从准实时到实时的迈进。

综合目前 visual SLAM 的研究热点和当前实验条件以及课题需求, 制定了主要研究内容如下:

(1) 在原有开源的代码之上, 综合各个算法的优点, 剔除其不足, 制定本文所使用的 SLAM 算法。在初始化阶段用五点算法代替传统的基于基础矩阵的八点算法和 DLT 算法, 在构建地图的过程之中改进 Bundle Adjustment 算法并提高了精度。

(2) 对 xtion pro 深度相机进行进一步的标定。由华硕官方配套的标定算法只能标定彩色和红外相机的内部参数, 但是对红外相机的扭曲畸变参数并没有提供, 本文对该相机扭曲畸变参数进行了建模并对齐进行了标定。并取得了良好的效果。

(3) 将惯性测量单元 (IMU) 融合进入 SLAM 系统之中。由于 SLAM 算法中构建地图是基于算法, 因此需要一个较好的初始值来进行迭代, 而 IMU 正好可以提供提供一个相对可靠的结果。

(4) 采用多线程, SIMD 等高级编程技术来提高 SLAM 的实时性, 由于 SLAM 算法是需要实时的在机器人上的嵌入式控制器上运行, 因此对算法的实时性提出较高的要求。本文在整体上采用多线程运行技术, 在局部上采用 Intel 公司针对 CPU 的向量化编程技术 SIMD^[29, 30] 进行进一步优化, 降低了算法的运行时间, 提高了实时性。

第 2 章 Xtion Pro Live 相机标定算法

2.1 引言

Xtion 是华硕公司开发的一款 RGBD 相机，相对于传统的摄像机，它能够同时获取场景中的颜色和深度的信息。本章对 xtion pro live 相机做了简单介绍，针对深度图存在畸变的问题提出了一整新的校正方法，相比之前的算法有较大的精度提高。

2.2 Xtion pro live 相机简介

Xtion pro live 是一款 RGB-D 传感器，能够同时获取彩色图像和深度图像的信息。由华硕公司制造，和微软推出的 kinect 传感器相类似，都是针对于体感游戏开发，但其相对于 kinect 传感器的优势在于其质量更轻，功耗更低，IO 接口为 USB2.0，可直接通过 USB 接口供电。可直接放在无人机上作为视觉传感器使用。相比于传统的立体相机或 TOF 相机，它价格更低，对复杂环境的适应性更强^[31]。

Xtion pro Live 相机组件主要有三个镜头构成，中间的镜头为 RGB 相机，左侧为红外线发射器，右侧为红外线摄像机。红外线发射器发射结构光信号，而红外线接收器通过观测光线照射到物体上所衍射的光斑即可得到深度，而左右两个圆形小孔为麦克风，负责接收声音信息。

RGB 彩色相机：它提供两种分辨率的图像分别为 320x240（QVGA），640x480（VGA），其中帧数均为 30 帧每秒。

深度相机：它提供一种分辨率为 640x480 的图像，速度为 30 帧每秒。使用距离为 0.8~3.5m 之间，视野为水平 58 度，垂直 45 度，对角线 70 度^[32]

左右麦克风：用于声音输入，过滤环境噪声。



图 2-1 xtion pro live 结构示意图

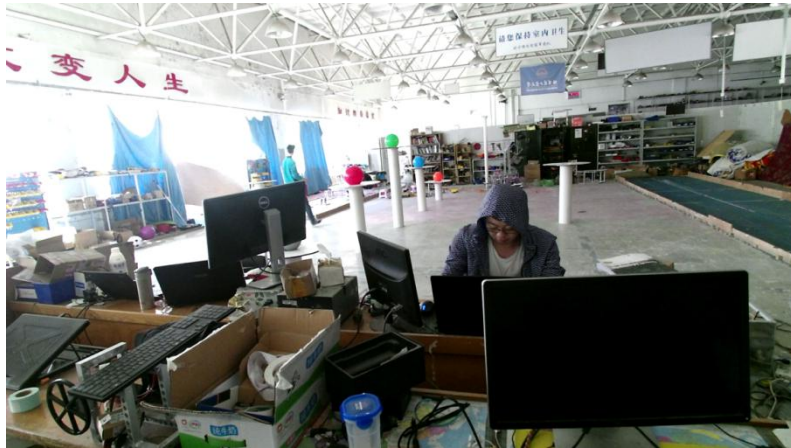


图 2-2 xt看 pro live 获取彩色图



图 2-3 xt看 pro live 获取深度图

如图 2-2,2-3 所示，上侧为普通 RGB 相机获取的颜色信息图，右侧为红外相机获取的经过处理之后的图像，距离相机最近的点显示白色，而在相机能够获取的深度信息之外的点显示灰色乃至黑色，而在这两者之间的颜色为过渡色，反应出场景各点相对于深度相机的距离，深度越深则图像中该点距离我们越远。

Xtion pro Live 体感设备通过结构光原理来实现深度信息的获取。所谓结构光是一种排布有规则的红外线，如图 2-4 所示。整个系统由三部分构成，分别为结构光发射器，红外相机，和待测物体。结构光发射器发射出具有特殊编码的光线，在表面上会形成根据表面形状，距离红外相机的距离不同而形成的二维光条畸变图像，结构光发射器和结构光接收器之间的距离和夹角以及物体表面与上述两者之间的距离决定了光条的畸变程度，光条显示出的位移和物体表面的高度成正比，当结构光发射器和红外相机位置固定时，可以通过畸变的二维光斑来重现物体表面的三维轮廓^[33]。

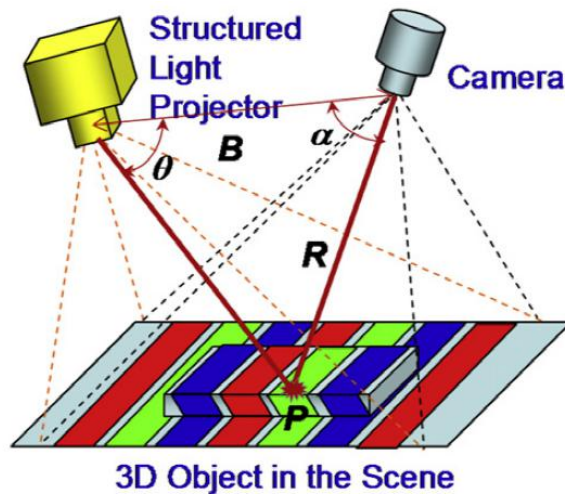


图 2-4 结构光原理示意图

在上图中依据正弦定理，显然有

$$R = B \frac{\sin(\theta)}{\sin(\alpha + \theta)} \quad (2-1)$$

由于在加工制造的过程中， θ ， α 和 B 都是已知变量，即同一个条纹的发射角度是相同的，根据其在红外接收器上的接收角度，和出厂时测量的结构光发射器和红外接收器的距离，就可以通过上式很快的计算出待测量点相对于相机的深度信息。而在实际应用中，可以选用的结构光编码方式有很多，比如条纹，渐变纹，格状纹等等，无论结构光的方式如何，其都能在 XY 两个方向上唯一的确定自身坐标。

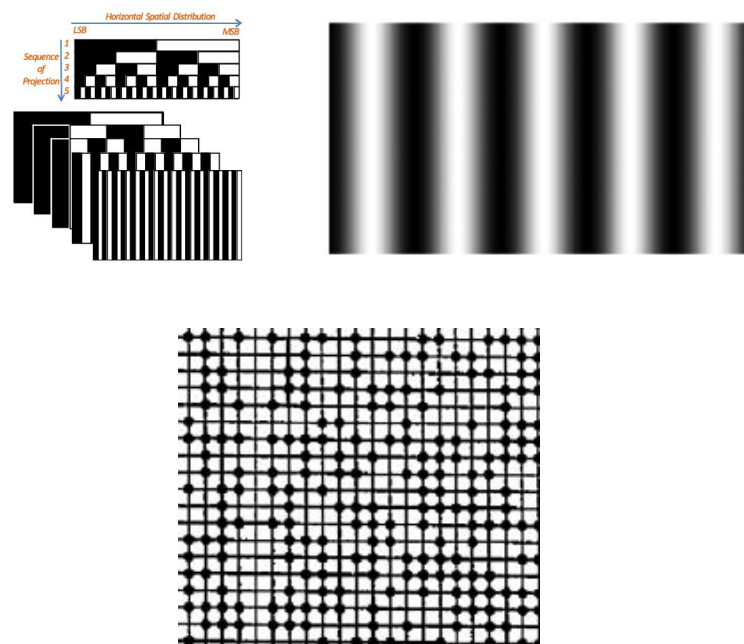


图 2-5 结构光编码图

2.3 Xtion pro live 相机成像模型

和传统相机模型相类似，我们可以用同样的摄像机几何模型来描述，如图 2-6 所示。

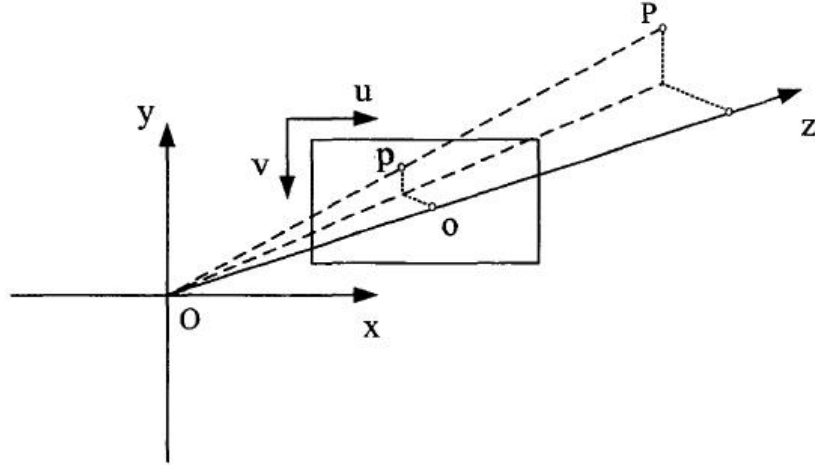


图 2-6 小孔成像模型

以红外相机的摄像头中心为原点，成像平面为 $O-uv$ ，摄像机坐标系为 $O-xyz$ ，其中 x 轴与图像坐标系的 u 轴重合， y 轴与图像坐标系的 v 轴相反， z 轴和光轴重合，深度图像中某像素值对应在该方向上物体到相机的距离。三维点 P 在摄像机上的投影点为 p ，即 p 是三维空间点 P 和相机坐标系原点 O 的连线 and 成像平面的交点。由已知条件深度图像中某点 p 的像素坐标为 d_p ， f 为相机焦距，根据小孔成像原理，可以得到物体在相机坐标系下的坐标为：

$$\begin{cases} x_p = \frac{d_p x_d}{f} \\ y_p = \frac{d_p y_d}{f} \\ z_p = d_p \end{cases} \quad (2-2)$$

而摄像机空间坐标系的单位是 mm ，而实际上算法中常用的是以像素单位，以 u_d 和 v_d 来表示这个距离，则他们与 x_d 和 y_d 之间的换算关系为：

$$\begin{cases} x_d = \frac{u_d}{c_x} \\ y_d = \frac{v_d}{c_y} \end{cases} \quad (2-3)$$

式中 c_x —— 横向像素大小与相机坐标系长度比值（像素/毫米）

c_y —— 纵向像素大小与相机坐标系长度比值（像素/毫米）

而摄像机坐标系原点和像素坐标系原点并不在同一个位置上，因此需要原点进行平移。则有

$$\begin{cases} u_d = u_p - u_0 \\ v_d = v_p - v_0 \end{cases} \quad (2-4)$$

式中 u_0 —— 光轴和成像平面的交点的像素坐标值的横坐标值（pixel）

v_0 —— 光轴和成像平面的交点的像素坐标值的纵坐标值（pixel）

综合上式，可以得到

$$\begin{cases} x_p = \frac{d_p(u_p - u_0)}{fc_x} \\ y_p = \frac{d_p(v_p - v_0)}{fc_y} \\ z_p = d_p \end{cases} \quad (2-5)$$

综合以上三式，将图像坐标系和在摄像机下坐标系用矩阵的方式关联起来可以得到

$$z_p \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \quad (2-6)$$

其中 $f_x = fc_x$ ， $f_y = fc_y$ ，这两者分别被叫做横纵方向上的等效焦距，这样就可以建立空间三维点和图像上坐标之间的联系。

在实际应用中往往不会以相机坐标系作为世界坐标系，通常意义下世界坐标系是固定在世界中某一点，在整个讨论问题的过程中都不会变。因此需要世界坐标系和摄像机坐标系的转换关系。

在我们生活的三维空间中，任意两个物体的相对空间关系可以通过六个参数来描述，分别为三个旋转参数和三个平移参数^[34]，通常会把三个旋转参数写成旋转矩阵的形式，平移关系写成向量形式：

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (2-7)$$

则某三维空间点在世界坐标系下的表示为 $P_w = [X_w, Y_w, Z_w]^T$ ，在摄像机坐标

系下的表示为 $P_c = [X_c, Y_c, Z_c]^T$ ，则两者有如下关系：

$$P_c = RP_w + T \quad (2-8)$$

为了方便做矩阵乘法运算，学界一般会采用齐次坐标的形式来表示某三维空间点，表示方法也很直观易懂，在某三维空间点后加 1，将其扩展成 4x1 的矩阵，则世界坐标系下的三维空间点 P 的齐次坐标表示为 $P'_w = [X_w, Y_w, Z_w, 1]^T$ ，摄像机坐标系下三维点 P 的齐次坐标为 $P'_c = [X_c, Y_c, Z_c, 1]^T$ ，两者关系如下：

$$P'_c = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} P'_w \quad (2-9)$$

融入上文提到的摄像机坐标系到像素坐标系的变换矩阵，可以得到：

$$z_p \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} P'_w \quad (2-10)$$

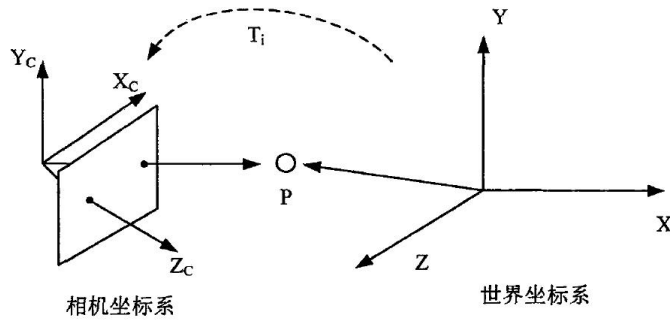


图 2-7 坐标系变换

其中 u_0 , v_0 , f_x , f_y 叫做摄像机的内参数， $\begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix}$ 叫做相机的外参数，

令

$$P = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \quad (2-11)$$

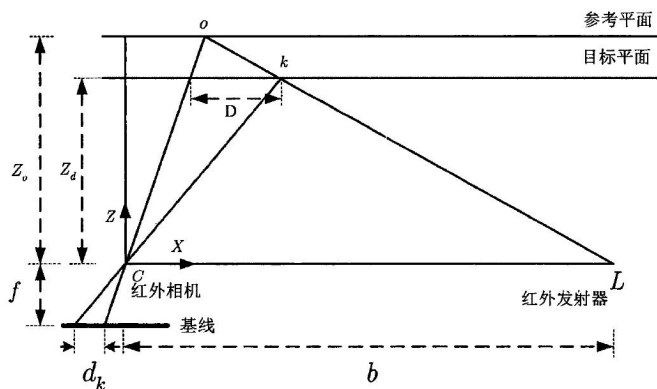
则 P 为摄像机的投影矩阵。

2.4 标定和畸变校正方法

对于 xtion pro 的一种标准的标定方式是分别独立的标定彩色和深度相机的信息，然后标定它们之间的相对位置。但是这并不是最好的方法，因为两个相机的标定是相对独立的，并没有充分利用它们之间的相互对应的信息。而且独

对于这个问题 Fuchs 和 Hirzings^[35]提出了一种针对于 TOF 相机的基于多项式运算的标定方法，但是这种方法需要很多的参数并且要求有一个高精度的机械臂来确定相机的位置。但是高精度的机械臂的获取相对而言比较困难，Lindner 和 Kolb^[36]为了去除掉机械臂使用了高精度的彩色相机来确定深度相机的位置，Lichti^[37]提出了一种采用 3D 激光雷达的标定方法，但是仍然需要使用一种高精度的标定物体。Zhu 等人提出了一种融合立体相机和 TOF 相机的标定方法。他们的方法将经过双目立体视觉的三角测量方法的结果作为真实结果来标定深度相机，但是这个方法最大的问题是忽略了三角测量方法的误差，直接将三角测量的结果直接作为真实结果。

为了解决深度畸变问题，本文主要针对深度畸变信息进行校正。在讨论该问题之前，我们需要详细的讨论 Xtion Pro Live 计算深度的物理模型。



图中 k ——物体实际位置 (m)

d_k ——视差 (pixel)

依据相似三角形，有以下两个方程：

$$\frac{D}{b} = \frac{Z_0 - Z_d}{Z_0} \quad (2-12)$$

$$\frac{d_k}{f} = \frac{D}{Z_d} \quad (2-13)$$

整理可得：

$$Z_d = \frac{Z_0}{1 + \frac{Z_0}{fb} d_k} \quad (2-14)$$

上式为 xtion pro Live 相机的基本数学模型，其中 Z_0 ， b 和 f 都是红外相机的内部参数，在出厂的时候可以确定，这样通过计算视差 d_k 即可计算出深度 Z_d 。

为了简化计算，合并多余参数，将公式 2-14 简化为：

$$Z_d = \frac{1}{m_1 d_k + m_0} \quad (2-15)$$

式中 $m_1 = \frac{1}{fb}$ ， $m_0 = \frac{1}{Z_0}$ 。

由于该模型仅仅是一个理论模型，如果仅用该模型则会得到一个非常奇怪的结果，然而由于 xtion pro live 内部畸变参数并不是开源的，因此我们不没有办法计算像素畸变，所以采用如下模型来解决畸变问题：

$$Z_{dd} = Z_d + Z_\delta(u, v, d) \quad (2-16)$$

即加入一个以 u ， v 和 xtion pro 返回的视差 d 作为自变量的为自变量的函数作为误差补偿。通过观测可以发现误差大概和 xtion pro 返回的视差成指数关系，

实验测量结果如图 2-9 所示，其中横坐标轴是大小，以像素为单位。纵坐标轴是深度误差大小，以毫米为单位。针对以上实验结果，设计如下函数作为误差补偿函数：

$$d_k = d + D(u, v) * \exp(\gamma_0 - \gamma_1 d) \quad (2-17)$$

式中 d ——xtion pro live 返回的视差 (pixel)

γ_0 ， γ_1 ——畸变参数

$D(u, v)$ ——该点到光心原点距离

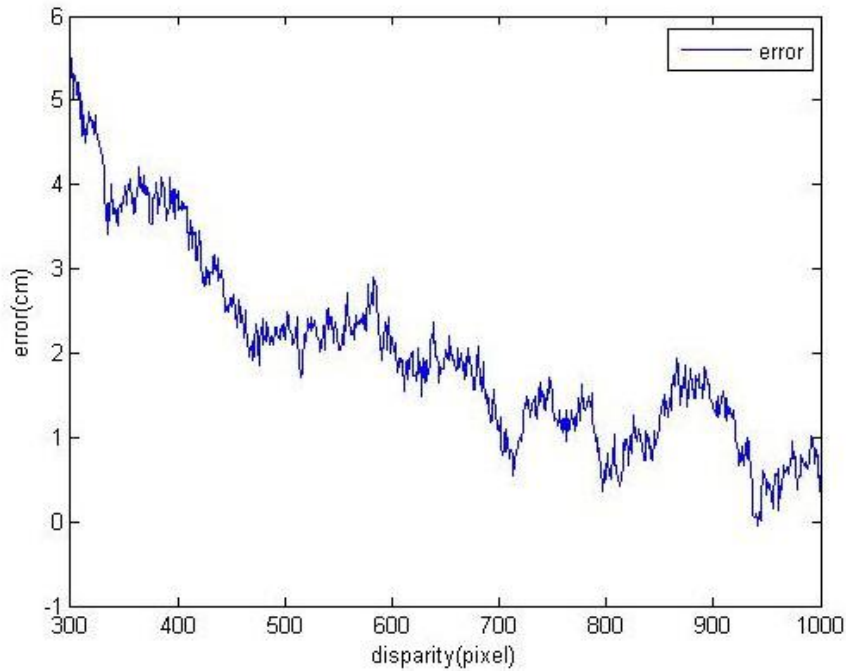


图 2-9 误差与视差关系

在实验中，选用一块平面放置在相机正前方，调节相机位置，并测量得到相机距离整个平面的距离作为真实结果 Z_{dd} ，使相机整个视野都落在该平面上，对图像中每一点进行逐点标定。并将标定结果和 $xtion$ 自身计算出的深度值进行比较，得出误差，通过上述模型进而计算出视差的误差，并拟合出 γ_0 和 γ_1 ，标定结果如图 2-10。

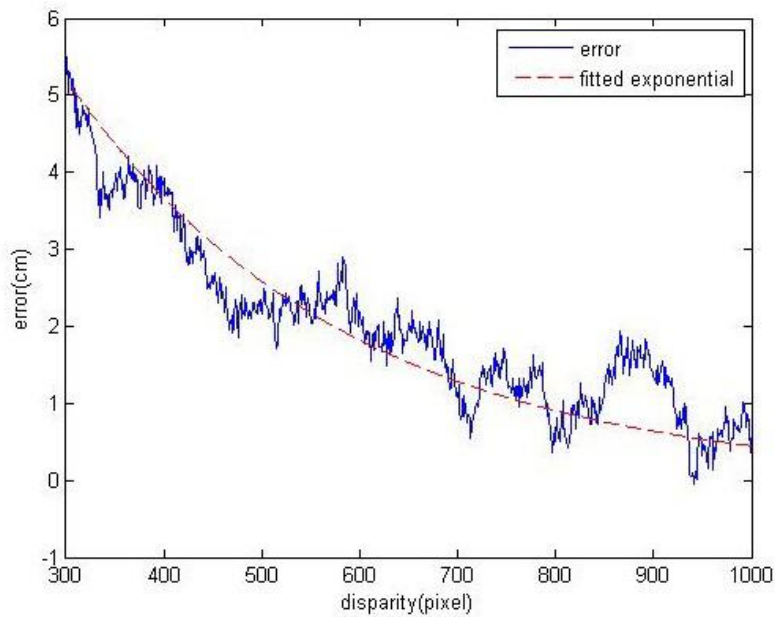


图 2-10 误差拟合结果

2.5 误差和畸变补偿实验

实验采用一个 150cm x 90cm 的平面放在 xtion 前面 2m 处，测定 xtion 得到的点云图的原始数据，得到的点云深度图如 2-11 所示。



图 2-11 原始数据图

可以发现图像的四个角明显的有向内翻转的情况，这正是由于红外相机的畸变参数造成的。下图是经过标定算法得到的平面点图，可以发现四个角落的翘起得到明显的缓解。

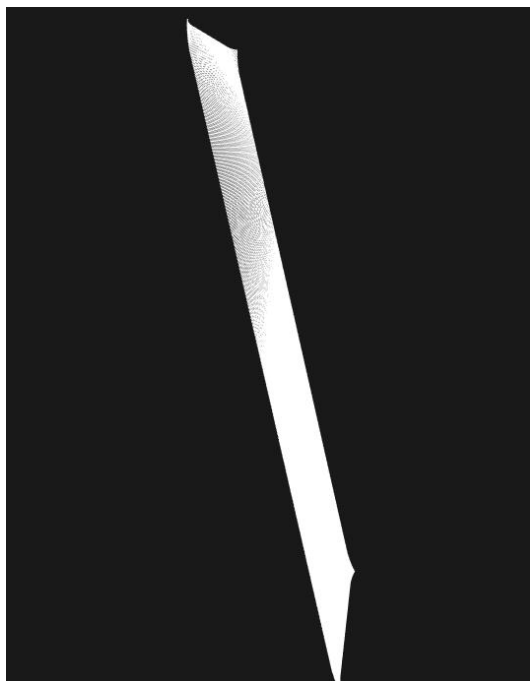


图 2-12 畸变补偿之后结果

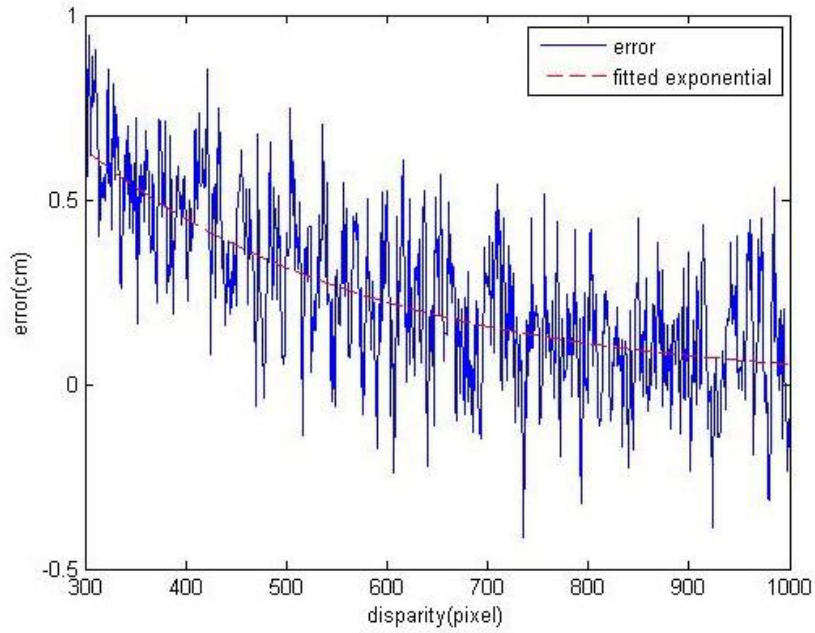


图 2-13 畸变补偿之后误差统计

如图 2-13 所示，经过补偿之后可以发现误差已经大大缩小，误差已经基本集中于正负 0.5cm 范围之内，基本满足了 SLAM 的定位需求。

2.6 本章小结

本章针对 xtion pro 相机的功能，参数以及成像原理进行了详尽的论述，针对红外相机的畸变问题建立了误差模型，对误差进行了分析，得到了误差的拟合函数并进行了补偿，实验证明该误差补偿算法能够满足 SLAM 系统对点云精度的整体需求。

第 3 章 基于 ORB 特征的前端特征提取与关键帧提取

3.1 引言

近年来局部不变特征已成为国内外的研究热点，目前在图像处理领域应用较多的局部不变特征大致有以下几种：SIFT^[39]，SURF^[40]，FAST^[41]，AKAZE^[42]，ORB^[38]等。其中 SIFT 和 SURF 是最著名的两种特征描述子，而之后绝度大多特征描述子的思想和计算方法都是依据这两者的思路。虽然这两种特征匹配效果准确，具有旋转和尺度不变形，但其运算代价较大，不适宜应用于 visual SLAM 领域。针对这个问题，本章提出了一种基于先验信息的 ORB 特征提取及匹配方法并比较了该方法与传统方法在效率和准确度上的优劣。

3.2 ORB 特征简述

ORB 特征是一种局部不变特征，它是一种描绘图片中某一处或某一点的特征的描述方法^[38]。ORB 特征是 E Rublee 等人于 2011 年的 ICCV (IEEE International Conference on Computer Vision)上提出的，该特征是建立在 FAST 特征的基础之上，具有极高的运算速度，约为 SIFT 特征的 100 倍，SURF 特征的 10 倍，在提出之后便被广大的计算机视觉研究者所使用。

特征提取是该 tracking 线程的第一部分，也是整个算法的第一部分，考虑到无人机上所搭载的处理器一般性能不会很高，同时要考虑特征的鲁邦性，最终采用 FAST-9 作为所提取的特征。即周围连续 16 个候选点中至少有 9 个点的像素值均高于或低于中心点超过阈值。将周围点与中心点作差值得 $S = \{s_1 s_2 s_3 s_4 s_5 \dots s_n\}$ 到的绝对值累加作为该点的得分。

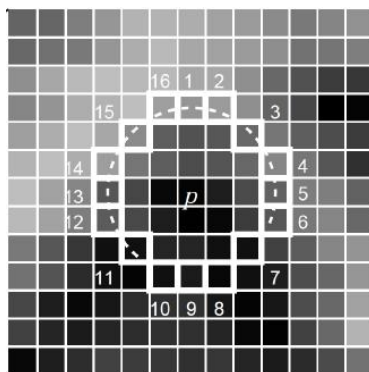


图 3-1 FAST-9

$$S = \max \left\{ \sum I_{i,j} - p \text{ if } (I_{i,j} - p > \text{threshold}) \right. \\ \left. \sum p - I_{i,j} \text{ if } (p - I_{i,j} > \text{threshold}) \right\} \quad (3-1)$$

不同于以往的特征提取算法，本文采用了一种启发式的自适应阈值的特征提取算法。首先每 32x32 个方格划分为一个 grid，在每个 grid 之中提取二个最佳的特征，即得分最高这样可以使特征均匀分散在整个图像上，减少某一部分的特征过于集中导致位姿优化对该部分过拟合。而提取特征的阈值则采用启发式的算法，即将阈值设为在 max threshold 和 min threshold 之间，阈值从 max threshold 开始，如果能提取到 2 个特征则使用该阈值，取当前区域得分最高的两个特征点。如果提取不到则降低阈值直到能提取到至少两个为止，如果降低到 min threshold 还没有提取到，则放弃该 grid。即取当前格子中所有特征集合为 S，

$$S = \{s_1 s_2 s_3 s_4 s_5 \dots s_n\} \quad (3-2)$$

$$S = \begin{cases} s_1 s_2 & \text{if } \min \text{thresh} < s_1 s_2 < \max \text{thresh} \\ \emptyset & \text{else} \end{cases} \quad (3-3)$$

确定特征点位置之后下一步计算特征点的方向。由于特征点需要具有旋转不变性，即无论从任何角度来观看同一个特征，算法都能够保证每次观测到的特征均为同一个。而计算特征的方向就能够保证特征具有旋转不变形。特征的方向的计算方法为以前一步所计算的关键点为圆心，以关键点和取点区域的形心的连线为 X 轴建立二维坐标系。而这个坐标系和标准直角坐标系所成的旋转角度即为该特征的旋转角度。由于圆心是固定的，重心是和图像一同旋转的，所以在不同的旋转角度下，重心和特征点中心的连线是与图像之间的关系是固定的。首先定义图像块的矩为：

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (3-4)$$

记为图像的 p,q 阶矩，则图像中心坐标可以用矩的形式来表示：

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3-5)$$

而特征的角度为

$$\theta = a \tan 2(m_{01}, m_{10}) \quad (3-6)$$

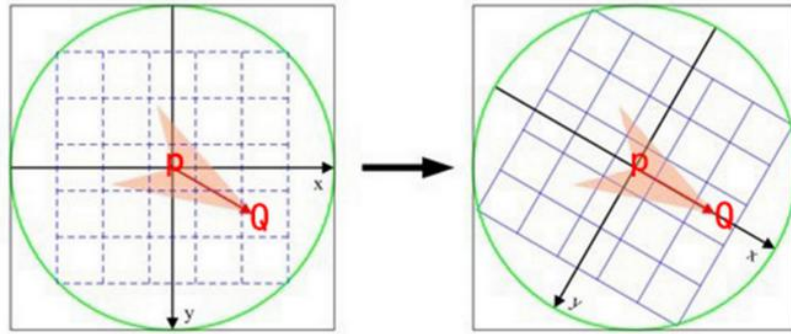


图 3-2 特征方向性

特征描述子部分采用的是 BRIEF 描述子^[43]。以特征点为中心，16x16 像素范围内随机取 64 个像素点对 (p_i, q_i) ，同时建立一个由 64 位字节构成的二进制串 $S = \{s_1 s_2 \dots s_{64}\}$ ，对每一个点对儿比较两个点对儿亮度值，如果 $I_p > I_q$ 则对应的二进制串该位记为 1，否则记为 0。这样便会得到每个特征的描述子。即

$$s_i = \begin{cases} 1 & \text{if } I_p > I_q \\ 0 & \text{if } I_p \leq I_q \end{cases} \quad (3-7)$$

为了应对同一个特征距离相机远近不同而导致的图像大小不同的问题，需要解决特征尺度的问题，解决方法是将图片放大到不同的倍数，在同一个位置提取特征，此时会得到同一个特征在不同尺度上的描述子，一旦摄像机距离图像的远近不同得到的图像在尺度上有差异。在匹配的时候取相同尺度的描述子进行匹配即可^[42]。

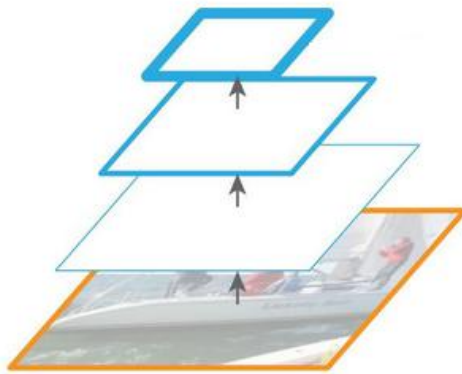


图 3-3 图像金字塔

3.3 特征匹配方法

ORB 特征的匹配方法是基于特征描述子之间的汉明距离设计的，汉明距离表示的是两个（相同长度）字对应位不同的数量。ORB 特征为 64 位，每一位

都被表示为 0 或 1。而特征的匹配方法则是计算两个特征汉明距离。具体计算方法则是对两个 64 位字符串做异或操作，即如果同一个位置的两个数值相同，则结果为 1，否则为 0。由于 CPU 的内部是基于门电路设计，对 0,1 的与非操作极快，ORB 特征匹配正是利用了该点极大的提高了算法的匹配速度。

传统的特征匹配方法是基于 Brute Force 的匹配方法。所谓 Brutal Force 其实就是暴力算法。枚举所有可能的排列组合，然后找到最优化的或者理想的结果。Brute Force 算法的结果一般来讲是最优的，但是匹配的过程却极为简单粗暴，耗时较长。尤其在特征相对较多时，两张图片匹配的时间大小一般在秒级，而特征匹配线程则是需要放置在前端，对实时性要求较高。针对这个问题，本文提出了一种基于 ICP 的相机位姿估计算法^[44,45]，并在此先验信息基础之上进行特征匹配。

该算法首先根据之前得到的数据预测相机位置，之后以该处为起点进行 ICP 迭代，得到精度较高的位置结果，之后将空间中的 3D 点按照该相机姿态进行投影，在投影得到的位置周围进行特征点匹配。

但是由于 ICP 计算代价较大，因此并不对所有空间点云进行迭代，而是对特征点对应的空间 3D 点进行较少次数的迭代，目的是在既提高精度的基础之上，保证运算性能。

如图 3-4 所示，我们首先建立相机运动模型，由于无人机在空中的运动并不是毫无规矩可循的，而是按照一定的运动规律进行运动，因此我们可以对相机的运动进行建模分析。

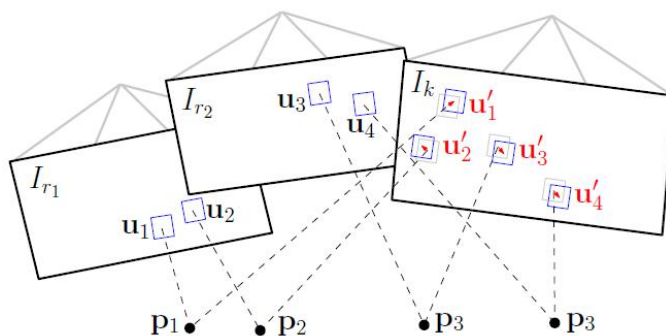


图 3-4 相机运动分析

由于相机帧数为 30，即每秒拍摄 30 张图片，因此，每两张图片之间的时间间隔为 33ms。在这段时间相机移动的距离除以时间即可得到相机的运动速度。然而由于采样时间相对较短，每两帧之间的速度误差相对较大，实测数据如图 3-5 所示。

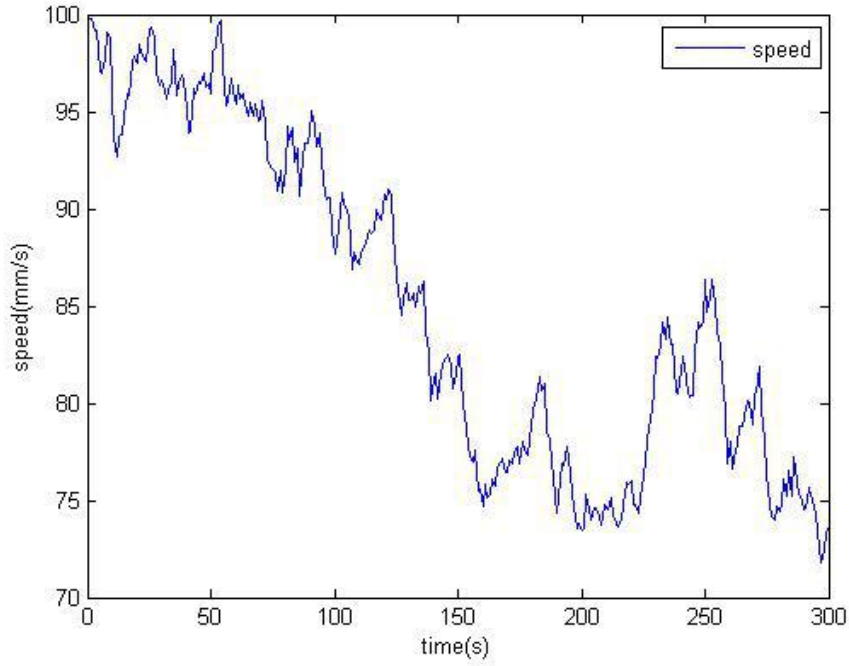


图 3-5 相机速度分析

本文采用滑动均值滤波对每两个相机之间的姿态进行处理，得到如下图所示设某一时刻相机运动的位置为 p_i ，则此时得到的速度原始值为 $\vec{v}_i = p_i - p_{i-1}$ ，对齐进行大小为 5 的滑动均值滤波处理可以得到：

$$\vec{v}_m = \frac{1}{5} \sum_{i=m-4}^m \vec{v}_i \quad (3-8)$$

即当前速度为最后 5 帧的速度平均值。下一帧速度的估计值如下：

$$\vec{v}_{m+1} = \frac{1}{9} \left(\vec{v}_m + \vec{v}_{m-1} + \vec{v}_{m-2} - \vec{v}_{m-3} - \vec{v}_{m-4} - \vec{v}_{m-5} \right) + \vec{v}_m \quad (3-9)$$

即将之前 6 帧的加速度的平均值作为该帧的加速度，来预测下一帧的速度。，则此时相机的预测位置为

$$p_{m+1} = p_m + \vec{v}_{m+1} \quad (3-10)$$

经过平滑之后的速度如下图：

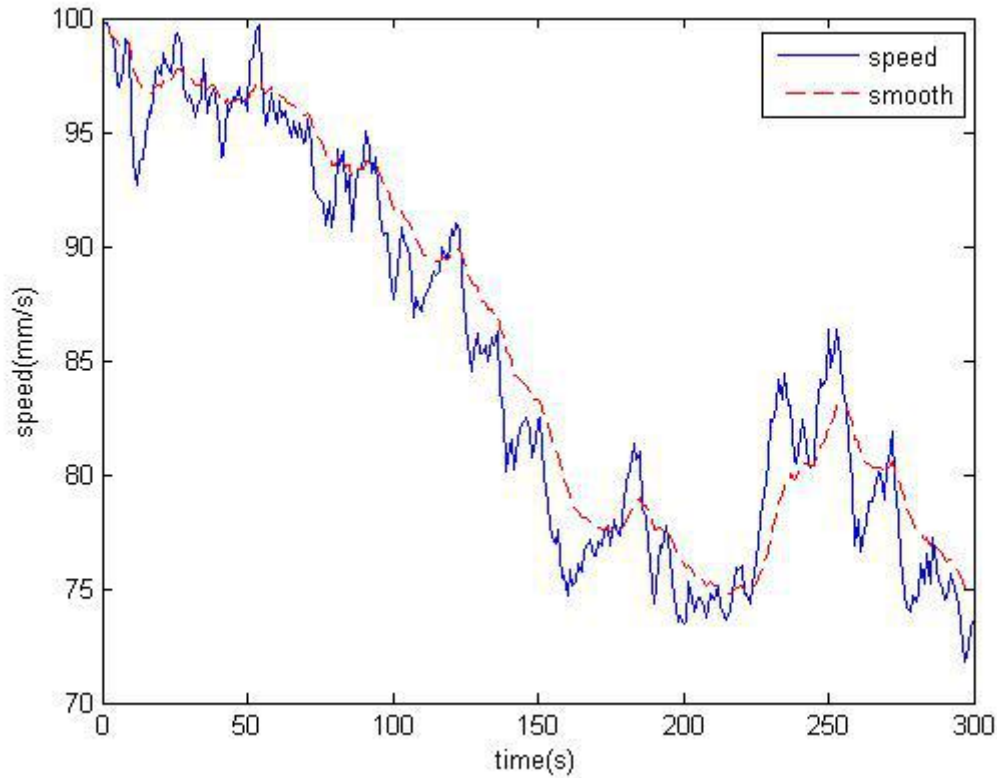


图 3-6 相机速度平滑结果

得到相机预测点位置之后, 由于此时预测精度较低, 仍然会出现错误匹配位置的问题。在此基础之上借助于深度相机获取的点云信息进行 ICP 点云迭代算法。Besl 等于 1992 年提出了 ICP 算法, 对两个点集建立对应匹配点, 之后在给定初始旋转和平移矩阵的基础之上, 计算出两个点集对应点的误差之和。之后不断进行迭代, 直至误差达到某个精度或迭代次数达到上限。

设第一个点集为 $\{P | P_i \in R^3, i=1,2...N\}$, 第二个点集为 $\{Q | Q_i \in R^3, i=1,2...N\}$, 计算的代价函数是使两个点集对应点的误差之和最小, 即

$$R, T = \arg \min \sum_{i=1}^m \|Q_i - (RP_i + T)\|_2^2 \quad (3-11)$$

公式 3-11 中 R 和 T 分别为两个空间点集之间的旋转和平移矩阵

ICP 算法的目的是得到两个点集之间的旋转和平移参数, 其计算步骤为:

1. 计算 P_i 中每一个点在 Q 中对应的最近点, 记为 Q_i
2. 按照公式 3-11 求取在此时所有点集距离之和最小刚体变换矩阵 R 和 T
3. 利用 R 和 T 对点集 P 进行刚体变换操作, 得到新的点集
4. 测试新得到的点集与原始点集是否满足目标函数, 即误差是否小于阈值

或者迭代次数在某个特定次数范围之内，如果满足则停止迭代，如果不满足则进行下一次迭代。

经过 ICP 迭代之后便可以得到精度较高的结果，将空间中已有 3D 点投影到图像上，以该点为中心取 10x10 像素的正方形小块，将 3D 点对应的特征与图像中该特征进行匹配，这样可以极大的减少特征匹配时间。

3.4 融合 IMU 进行位置估算

IMU 是移动机器人、移动智能设备上常见的传感器。常见的 IMU 为六轴传感器，配备输出三轴加速度的加速度计和输出三轴角速度的陀螺仪。加速度计检测物体在自身坐标系下沿三个坐标轴上的加速度，而陀螺检测载体相对于自身坐标系下的角速度信号，测量物体在三维空间中的角速度和加速度，并用这些信息通过积分等方法解算出当前物体的姿态。九轴 IMU 还会配备输出三轴姿态角的磁力计。而在该算法中采用的是六轴 IMU。

3.4.1 IMU 状态模型

通常来讲 IMU 的状态量表示为

$$X_{IMU} = \begin{bmatrix} {}^I\bar{q}^T & b_g^T & {}^Gv_I^T & b_a^T & {}^Gp_I^T \end{bmatrix} \quad (3-12)$$

其中用 I 表示 IMU 坐标系，G 表示参考坐标系，IMU 的姿态由旋转量 ${}^I\bar{q}^T$ 和平移量 ${}^Gp_I^T$ 表示^[46]。具体来说，前者将任意量从参考坐标系映射到旋转坐标系，用单位四元数表示；后者为 IMU 在参考坐标系下的三维位置。 ${}^Gv_I^T$ 表示 IMU 在参考坐标系下的平移速度，另外两个量 b_g^T 和 b_a^T 分别表示陀螺仪和加速度计的偏移。

对于 IMU 状态估计问题，需要提供运动模型，观测（噪声）模型，估计误差模型：

$$\dot{x} = f(x) \quad (3-13)$$

$$z = g(x) + n \quad (3-14)$$

$$\delta(x) = e(\hat{x}, x) \quad (3-15)$$

这是一个通用模型，本文用 x 表示真实状态量，用 z 表示观测量， n 表示观测噪声， \hat{x} 表示当前的状态估计值。下面对其进行详细的分析：

一个刚体在同一惯性坐标系下进行平移运动，其平移量对时间的一阶导和二阶导即速度和加速度：

$$\dot{p} = v \quad (3-16)$$

$$\dot{v} = a \quad (3-17)$$

对于旋转量以及非惯性系参考系，情况要复杂很多。

首先，考虑一个从原点出发的向量 r 绕单位轴 u 旋转，角速度大小为 $\dot{\theta}$ ：

角速度矢量可以表示为 $\omega = \dot{\theta}u$ ，易得向量 r 末端点 P 的速度矢量，即 r 的时间一阶导数为

$$\frac{dr}{dt} = \omega \times r \quad (3-18)$$

坐标系 B 绕单位轴 u 旋转，如上所述，其三个轴的时间一阶导数为：

$$\frac{di_B}{dt} = \omega \times i_B \quad (3-19)$$

$$\frac{dj_B}{dt} = \omega \times j_B \quad (3-20)$$

$$\frac{dk_B}{dt} = \omega \times k_B \quad (3-21)$$

由于 $[i_B \ j_B \ k_B]$ 是坐标系 B 相对于参考坐标系的旋转矩阵 R ，所以 R 的时间一阶导数为：

$$\dot{R} = [\omega \times i_B \ \omega \times j_B \ \omega \times k_B] = \omega \times R \quad (3-22)$$

在数学计算中叉乘运算可以转化为反对称矩阵的乘法：

$$\dot{R} = \omega \times R = \omega^\wedge R \quad (3-23)$$

其中，反对称矩阵为：

$$\omega^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (3-24)$$

此时角速度 ω 是在参考坐标系下表达的。当角速度表达在体坐标系 B 下时，记为：

$${}^B\omega = R^T \omega \quad (3-25)$$

也可以表达为 $\omega = R^B \omega$ ，于是可以写为：

$$\dot{R} = (R^B \omega)^\wedge R \quad (3-26)$$

反对称矩阵具有一个非常好的性质：对任意旋转矩阵 R 和三维向量 v ，都有 $(Rv)^\wedge = Rv^\wedge R^T$ ，于是上式可以写为：

$$\dot{R} = R({}^B\omega)^\wedge \quad (3-27)$$

用旋转矩阵来表示旋转虽然很直观，但是过于冗余，因为旋转仅仅有三个自由度，而旋转矩阵存在九个量。表示旋转还可以用欧拉角，但是由于存在万向锁的问题，而且计算也不方便。旋转矩阵和四元数是更常用的表征方法，在惯性导航的过程中更多的是运用四元数。

一个四元数由一个实部和三个虚部构成，书写顺序分为两种，一种是虚部在前，实部在后，另一种是实部在前，虚部在后。本文采用虚部在前，实部在后的写法：

$$\bar{q} = q_1 i + q_2 j + q_3 k + q_4 = [v^T \quad q_4] \quad (3-28)$$

虚部 $v = [q_1 \quad q_2 \quad q_3]^T$ ，虚部三个基 i, j, k 满足 $i^2 = j^2 = k^2 = ijk = -1$ 。尽管如此，四元数仍然是一种冗余的表达方法，四元数有四个变量，而旋转具有三个自由度，因此为了消除一个冗余的变量，通常采用单元四元数来进行表示，即四元数的模的值为 1，这样正好符合三个自由度。

四元数和旋转向量具有很直接的转换关系，绕单位轴 u 旋转了 θ 角度，用四元数的表达为：

$$q = \left[u \sin \frac{\theta}{2} \quad \cos \frac{\theta}{2} \right] \quad (3-29)$$

而四元数的乘法 \otimes 类似于多项式乘法的逐项相乘：

$$\begin{aligned} q \otimes p &= (q_1 i + q_2 j + q_3 k + q_4)(p_1 i + p_2 j + p_3 k + p_4) = \\ &= (q_4 p_1 + q_1 p_4 - q_2 p_3 + q_3 p_2)i + (q_4 p_2 + q_2 p_4 - q_3 p_1 + q_1 p_3)j \\ &+ (q_4 p_3 + q_3 p_4 - q_1 p_2 + q_2 p_1)k + (q_4 p_4 + q_1 p_1 + q_2 p_2 + q_3 p_3) \end{aligned} \quad (3-30)$$

而这个结果可以表达为矩阵相乘的形式：

$$q \otimes p = \begin{bmatrix} q_4 I_3 - v_q^\wedge & v_q \\ -v_q^T & q_4 \end{bmatrix} \begin{bmatrix} v_p \\ p_4 \end{bmatrix} = \begin{bmatrix} p_4 I_3 + v_p^\wedge & v_p \\ -v_p^T & p_4 \end{bmatrix} \begin{bmatrix} v_q \\ q_4 \end{bmatrix} \quad (3-31)$$

四元数乘法的物理意义和旋转矩阵相乘是一样的，即：

$$R(q \otimes p) = R(q)R(p) \quad (3-32)$$

四元数对应矩阵为：

$$R(q) = (2q_4^2 - 1)I_3 - 2q_4v^\wedge + 2vv^T \quad (3-33)$$

由此可以得出四元数对时间的一阶导数为：

$$\dot{\bar{q}} = \frac{1}{2} \begin{bmatrix} -\omega^\wedge & \omega \\ -\omega^T & 0 \end{bmatrix} \bar{q} = \frac{1}{2} \Omega(\omega) \bar{q} \quad (3-34)$$

由此，可以得到 IMU 的运动模型，由四元数表达如下：

$$\begin{aligned} {}^I_G \dot{\bar{q}} &= \frac{1}{2} \Omega(\omega) {}^I_G \bar{q} \\ \dot{b}_g &= n_{wg} \\ {}^G \dot{v}_I &= {}^G a \\ \dot{b}_a &= n_{wa} \\ {}^G \dot{p}_I &= {}^G v_I \end{aligned} \quad (3-35)$$

3.4.2 IMU 观测和噪声模型

IMU 的噪声模型是基于高斯白噪声的随机游走模型，在具体使用中，需将其进行离散化。高斯白噪声是一种概率分布是正态函数，而二阶矩不相关，一阶矩为常数，是一种较为理想的模型。其瞬时值服从高斯分布，其功率谱密度是均匀分布的。常见的噪声中热噪声和散粒噪声都属于高斯白噪声。而随机游走则是指基于过去的 IMU 状态，并不能预测将来的发展步骤和方向，即之前的所有状态对当前的状态并不产生影响。

一个时间连续的高斯白噪声 $n(t)$ ，满足以下两个条件：

$$E[n(t)] = 0 \quad (3-36)$$

$$E[n(t_1)n(t_2)] = \sigma_g^2 \delta(t_1 - t_2) \quad (3-37)$$

其中 $\delta(\cdot)$ 表示狄克拉函数。可以看出，不同时刻的高斯白噪声是相互独立的。 σ_g^2 为方差，值越大表示噪声程度越大。将其离散化可以得到

$$n_d[k] = \sigma_{gd} w[k] \quad (3-38)$$

其中 $w[k] \sim N(0,1)$ ， $\sigma_{gd} = \frac{\sigma_g}{\sqrt{\Delta t}}$ ，其中 Δt 为采样时间，并且假定在一个采样

周期内 $n(t)$ 为常数，则：

$$n_d[k] = n(t_0 + \Delta t) \approx \frac{1}{\Delta t} \int_{t_0}^{t_0 + \Delta t} n(\tau) d\tau \quad (3-39)$$

$$\begin{aligned} E(n_d[k]^2) &= E\left(\frac{1}{\Delta t^2} \int_{t_0}^{t_0 + \Delta t} \int_{t_0}^{t_0 + \Delta t} n(\tau) n(t) d\tau dt\right) \\ &= E\left(\frac{\sigma_g^2}{\Delta t^2} \int_{t_0}^{t_0 + \Delta t} \int_{t_0}^{t_0 + \Delta t} \delta(t - \tau) d\tau dt\right) \\ &= E\left(\frac{\sigma_g^2}{\Delta t}\right) \end{aligned} \quad (3-40)$$

$$\text{即 } \sigma_{gd} = \frac{\sigma_g}{\sqrt{\Delta t}}$$

随机游走是一个离散模型，其对应的连续过程为维纳过程。维纳模型是高斯随机游走的积分：

$$\dot{b}_g(t) = n(t) = \sigma_{bg} w(t) \quad (3-41)$$

其中 w 为单位高斯白噪声，其离散化可以得到随机游走的模型：

$$b_d[k] = b_d[k-1] + \sigma_{bgd} w[k] \quad (3-42)$$

$$\text{其中： } w[k] \sim N(0,1), \quad \sigma_{gd} = \sigma_{bg} \sqrt{\Delta t}$$

IMU 观测模型在不同坐标系下有不同的表达方式，目前惯性导航中经常用到的有三个坐标系其分别为地心地固坐标系，地心惯性坐标系，体坐标系。

Earth-Centered-Earth-Fixed (ECEF) Frame: 地心地固坐标系 ECEF，该坐标系以地心为坐标原点，向北为 z 轴，赤道平面为 x - y 平面， x 轴正方形为经度 $(0,0)$ 点，ECEF 固定在地球上，跟随地球自转，非惯性坐标系。

Earth-Centered-Inertial (ECI) Frame: 地心惯性坐标系 ECI。以地心为坐标原点，向北为 z 轴， x - y 平面为赤道平面，春分点（每年春分时日心-地心连线与赤道的交点）为 x 轴的正方形指向。ECI 不跟随地球自转，其在惯性导航算法中被视为惯性坐标系。

Body Frame: 体坐标系。原点在导航体质心，与导航体固连，用来表示导航体的姿态。

当观测模型以 ECEF 为参考坐标系时，由于 ECEF 不是惯性系，需要把地球自转加入其中考虑，于是在加速度模型中会引入科氏加速度。记 ω_G 为地球自转角速度， G_g 为重力加速度， ω_m 为陀螺仪的观测量， a_m 为加速度计的观测量，

观测模型可以表达为下式：

$$\omega_m = \omega + R\left({}^I_G \bar{q}\right) \omega_G + b_g + n_g \quad (3-43)$$

$$a_m = R\left({}^I_G \bar{q}\right) \left({}^G a - {}^G g + 2\omega_G^{\wedge G} v_l + \left(\omega_G^{\wedge} \right)^{2G} p_l \right) + b_a + n_a \quad (3-44)$$

上述观测量都是在体坐标系下的表达，所以当用在参考坐标系上时都需要左乘一个旋转矩阵来进行转化。在上式中每个观测量的不确定量都可以用一个高斯白噪声之和和一个随机游走来表达。

当观测模型以 ECI 为参考坐标系时，由于 ECI 为惯性系，不需要考虑地球自转，于是观测模型与 ECEF 相比要简单许多：

$$\omega_m = \omega + R\left({}^I_G \bar{q}\right) \omega_G + b_g + n_g \quad (3-45)$$

$$a_m = R\left({}^I_G \bar{q}\right) \left({}^G a - {}^G g \right) + b_a + n_a \quad (3-46)$$

3.5 与传统方法对比实验

针对匹配算法的效率和准确度问题，本文对比了常用的局部不变特征包括 SIFT，GPU-SIFT(SIFT 特征的 GPU 运算形式)，SURF 和传统 ORB 特征的提取和匹配方法和本文提出的 ORB 特征提取和匹配方法。

测试采用分辨率大小为 640x480 的两张图，其上具有正确的匹配特征点数目为 158 对，匹配的结果如图 3-7 所示。

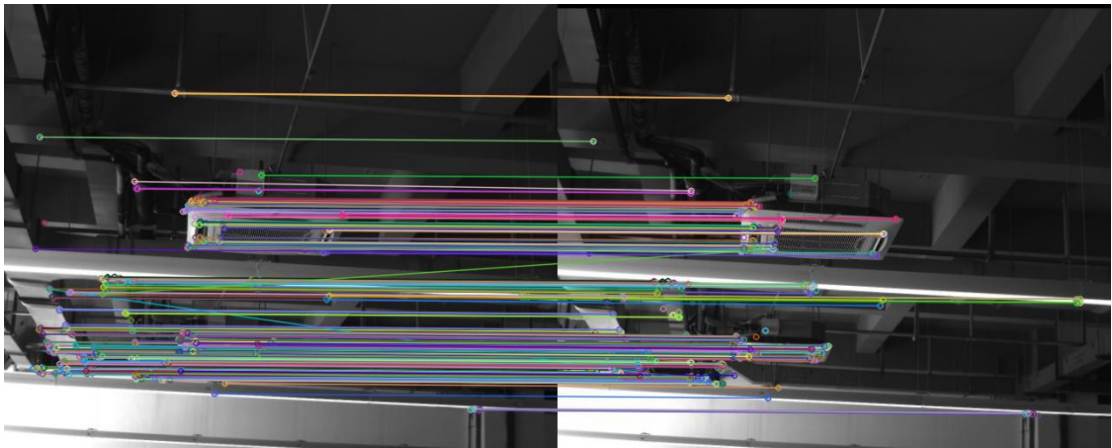


图 3-7 特征匹配结果

不同种特征提取和匹配算法的耗时如图 3-8 所示。

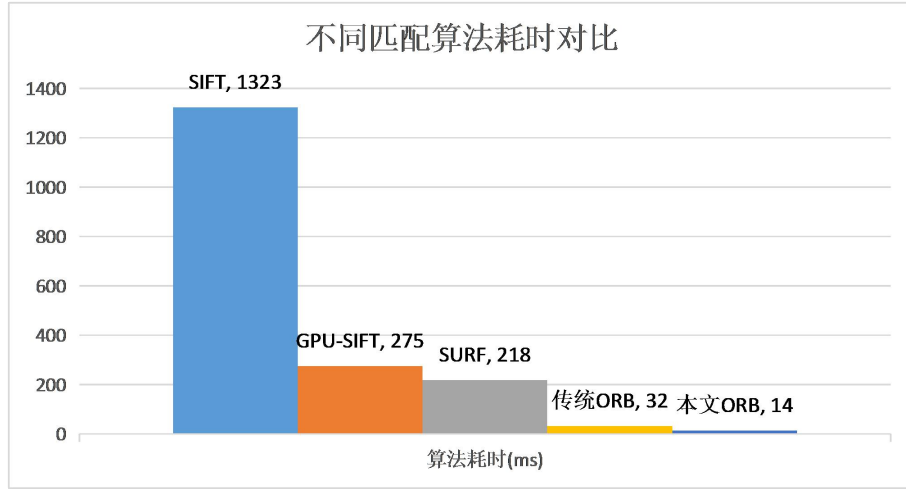


图 3-8 不同特征匹配算法时间

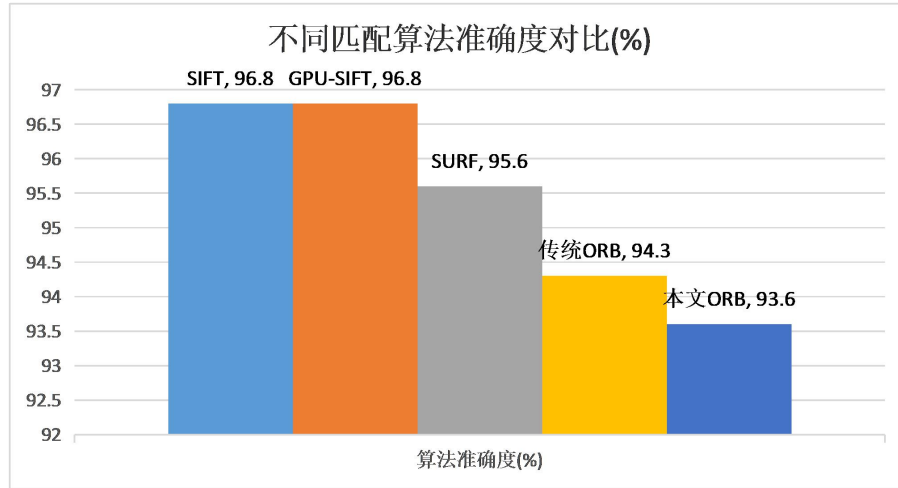


图 3-9 不同特征匹配算法准确度

如图 3-8 所示,本文提出的特征匹配方法相对于传统的 ORB 特征匹配方法,以及之前的 SIFT 特征等方法在运算时间上具有极大的优势,相对于传统的 ORB 特征匹配方法能够提高近一半的速度,而在准确度方面 SIFT 仍然占据最高的精度,在 158 对特征中准确的找到了 153 对,正确匹配率达到了 96.8%,而本文提出的 ORB 匹配算法也达到了 93.6%的水平,已经能够保证为后端优化提供足够数量的特征。

3.6 本章小结

本章详尽的叙述了 visual SLAM 中的前端特征提取部分,针对传统特征匹配方法运行速度较慢,计算量较大的问题进行了深入的研究。采用 ORB 特征作为特征描述方法,该特征具有运算速度快,错误匹配概率低等特点。在特征匹配的过程中采用了对运行相机进行速度建模,采用 IMU 得到的无人机位移数

据作为相机位置的初始参考点，并采用速度滑动窗口滤波的方法减弱噪声对算法带来的影响。实验证明本文提出的算法相对于传统的算法而言运算时间上具有明显优势，能够满足后端优化部分所需。

第 4 章 基于李群代数的优化方法

4.1 引言

目前绝大多数的 SLAM 算法在理论上都是基于优化算法，而传统的优化算法一般指 Bundle Adjustment 算法，即同时优化特征点位置信息和相机姿态信息。而传统 Bundle Adjustment 算法仅仅利用了图像的特征点的位置信息并没有利用图像的像素信息，这丢失了许多重要的特征信息，本章在传统 Bundle Adjustment 算法的基础之上提出了一种综合考虑特征点的位置信息和特征点周围图像信息的优化算法。

4.2 传统 Bundle Adjustment 方法

Bundle Adjustment 最初起源于大地测量学^[47]，后来很快的被应用于计算机图形学上来。Bundle Adjustment 可以将所观测的图像位置和预测的图像位置点进行最小 error 的映射（匹配），由很多非线性函数的平方和表示（error）。因此，最小化 error 由非线性最小二乘法实现，而目前最好的迭代算法为 Levenberg-Marquardt 提出的 Levenberg-Marquardt 算法^[20]（下简称 LM 算法），由其简单实现性和阻尼策略，能使其从初始假设值很快收敛到稳定状态，因而成为最好的方法。

通常意义上的 Bundle Adjustment 算法的代价函数是 3D 点在相机平面的重投影误差的和，即：

$$\arg \min_{C_1 \dots C_m, X_1 \dots X_n} \sum_{i=1}^m \sum_{j=1}^n \|h(C_i, X_j) - \hat{x}_{ij}\| \quad (4-1)$$

其中 C_i 代表相机的内参， X_j 代表 3D 点的坐标， $h(C_i, X_j)$ 是 3D 点投影到该相机上的像素坐标，而 \hat{x}_{ij} 则是提取的特征点位置，也就是说最小化重投影误差。而这种传统的方法最大的问题就是一定要利用特征点的位置，将图像的信息抽象画为许多特征点的位置，这样就会损失了所有的图像信息，一旦特征点的位置提取的不准确，或者错误的匹配较多，就会导致优化不收敛。针对以上问题，本文提出了一种新的优化方法。

4.3 优化的 Bundle Adjustment 方法

该方法主要分为两个步骤，第一步的算法仍然是最常见的 Bundle Adjustment 算法，以 3D 点重投影的误差之和作为代价函数，计算最优的变换矩阵。

该方法第一步的代价函数如下：

$$T_{LR} = \arg \min \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \left\| \pi(KK, T_{LR}, P_j) - x_{ij} \right\|^2 \quad (4-2)$$

即第一步仍然用较常规的计算 3D 点在左右两个相机的重投影误差。第二步是加入图像融合信息，将第一步计算的结果作为初始值带入第二步的优化。第二步主要应用像素信息，计算在当前变换矩阵的形式下，同一个 3D 点在左右图像上呈现的图像的像素差，求和之后作为代价函数

$$T_{LR} = \arg \min \frac{1}{2} \sum_{i=1}^m \left\| \delta I(T_{LR}, x_i) \right\| \quad (4-3)$$

因为上式是一个关于 T_{LR} 的一个非线性优化，因此采用了高斯-牛顿迭代法[21]，由上一步优化得到的 T_{LR} 作为该次迭代的初始值，然后通过 twist 矩阵 ξ [48] 更新 T_{LR} ，通过逆向更新的算法[49]来计算每次迭代更新的步长 ξ 和残差 e 。其中

$$\delta I(\xi, u_i) = I_R \left(\pi \left(\overset{\vee}{T}_{LR}, KK_R, P_i \right) \right) - I_L \left(\pi(T(\xi), KK_L, P_i) \right) \quad (4-4)$$

$I(u, v)$ 代表以该像素为中心，3x3 大小的像素块， $\pi(\cdot)$ 表示投影函数：

$$z_c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = (KK \quad 0_{3 \times 1}) T \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (4-5)$$

即从 3D 点经过投影方程得到该 3D 点在当前图像上的坐标 u, v 。

而变换矩阵更新方程为

$$\overset{\vee}{T}_{LR} \leftarrow \overset{\vee}{T}_{LR} \cdot T(\xi)^{-1} \quad (4-6)$$

为了寻找最佳的更新步长 $T(\xi)^{-1}$ ，计算了上式的偏导，并把它置为 0，得到如下方程

$$\sum_{i=1}^n \nabla \delta I(\xi, u_i)^T \delta I(\xi, u_i) = 0 \quad (4-7)$$

为了求解上式，把 $\delta I(\xi, u_i)$ 在零点泰勒展开，得到如下方程

$$\delta I(\xi, u_i) = \delta I(0, u_i) + \nabla \delta I(0, u_i) \xi \quad (4-8)$$

而雅格比方程 $J = \nabla \delta I(0, u_i)$ 是一个 9×6 的矩阵，因为在取图像块的过程中，取 3×3 大小的图像块而变换矩阵具有 6 个自由度。应用链式求导法，可以得到如下方程：

$$\frac{\partial \delta I(\xi, u_i)}{\partial \xi} = \frac{\partial I(a)}{\partial a} \Big|_{a=u_i} \cdot \frac{\partial \pi(b)}{\partial b} \Big|_{b=p_i} \cdot \frac{\partial T(\xi)}{\partial \xi} \Big|_{\xi=0} \cdot P_i \quad (4-9)$$

其中 $\frac{\partial I(a)}{\partial a}$ 为像素差对像素求导，这里和光流法有些类似。是图像的梯度，在这里，采用双线性插值来得到重投影点准确的像素值

其中 (u, v) 是要求取的图像坐标平面上的点，参与计算的包括其周围的最邻近的四个像素点，则该点的像素值可以由如下公式来计算：

$$I(u, v) = \frac{1}{4} \sum_{i=0}^3 I(u_i, v_i) (u - u_{3-i})(v - v_{3-i}) \quad (4-10)$$

即该点的像素值由周围的四个像素决定，与四个像素的距离成反比。 $\frac{\partial \pi(b)}{\partial b}$ 为像素对 3D 的相机坐标求导，为 2×3 矩阵由图像投影关系

$$\begin{aligned} u &= \frac{1}{z_c} (f_x x_c + z_c u_0) \\ v &= \frac{1}{z_c} (f_y y_c + z_c v_0) \end{aligned} \quad (4-11)$$

U, v 分别对 x_c, y_c, z_c 求导可得

$$\frac{\partial \pi(b)}{\partial b} = \begin{pmatrix} \frac{f_x}{z_c} & 0 & -\frac{f_x x_c}{z_c^2} \\ 0 & \frac{f_y}{z_c} & -\frac{f_y y_c}{z_c^2} \end{pmatrix} \quad (4-12)$$

$\frac{\partial T(\xi)}{\partial \xi}$ 为相机坐标对 3D 点所在的世界坐标系求导。由于是坐标对相机位姿求导，而相机位姿具有 6 个自由度，包括三个旋转自由度和三个平移自由度，所以得到的是 3×6 的一个矩阵。

平时常用的坐标变换矩阵为特殊欧式群 $SE(n)$ ，是李群^[50]中的一种，。而李群是指具有连续性质的群。并且，一般连续群上的运算还是无限可微，乃至解析的

$$SE(3) = \left\{ T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid R \in SO(3), t \in \mathbb{R}^3 \right\} \quad (4-13)$$

每个变换矩阵有六个自由度。其中旋转矩阵属于另一种李群叫做特殊正交群 $SO(3)$ ：常见的记法为

$$SO(n) = \{ R \in \mathbb{R}^{n \times n} \mid RR^T = I, \det(R) = 1 \} \quad (4-14)$$

故对应的李代数位于 \mathbb{R}^6 中：

$$se(3) = \{ \Xi = \xi^\wedge \in \mathbb{R}^{4 \times 4} \mid \xi \in \mathbb{R}^6 \} \quad (4-15)$$

而李群和李代数之间的关系是一个指数映射关系，即对李代数进行指数映射可以得到对应的李群，而对李群取以 e 为底的对数可以得到对应的李代数，本文选择的是用李代数进行求导，因为李群对加减并不封闭，而李代数对加减是封闭的。而在这里 ξ^\wedge 并不是上文所提到的反对称矩阵，而是

$$\xi^\wedge = \begin{bmatrix} \rho \\ \phi \end{bmatrix}^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ 0^T & 0 \end{bmatrix} \quad (4-16)$$

其中 ϕ^\wedge 指的是 $[\phi_1 \ \phi_2 \ \phi_3]^T$ 的反对称矩阵

令 $q = \exp(\xi^\wedge)P$ ，则最后一项可以写为 $\frac{\partial q}{\partial \xi}$ 按照矩阵求导法则，可得

$$\frac{\partial q}{\partial \xi} = \frac{\exp(\delta \xi^\wedge) \exp(\xi^\wedge) P - \exp(\xi^\wedge) P}{\delta \xi} = \begin{bmatrix} I_{3 \times 3} & -q^\wedge \end{bmatrix} \quad (4-17)$$

把 4-17 式放入 4-9 式中，并且把雅可比矩阵带入，可以得到的得到

$$\left(\sum_{i=1}^n J^T J \right) \xi = - \sum_{i=1}^n J^T \delta I(0) \quad (4-18)$$

通过求解上面的方程，可以得到 twist 矩阵 ξ ，之后按照公式 4-6 进行迭代即可算出最优解。

4.4 算法验证实验

为了比较传统 Bundle Adjustment 算法和本文提出的改进的 Bundle Adjustment 算法在性能，准确度与运算时间上的区别，本文设计了如下实验。来验证算法的精度和计算时间，步骤如下：

1. 采用张氏标定法来标定彩色相机的内部参数

2.将 xtion pro 相机固定到某一特定位置，并采用张氏标定法标定相机外部参数，并将其作为真实值。

3.给出某个随机值作为 Bundle Adjustment 的初始值。提取棋盘格角点作为特征点。

4.采用传统 Bundle Adjustment 算法来标定相机的外部参数。

5.采用改进的 Bundle Adjustment 算法来标定相机的外部参数

6.将 4 和 5 两步得到的参数与步骤 3 给出的参数进行比较。

标定所得相机内参如下：

$$KK = \begin{bmatrix} 533.0037 & 0 & 341.586 \\ 0 & 533.152 & 234.259 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-19)$$

扭曲参数：

$$kc = [-0.28947 \quad 0.10326 \quad 0.00103 \quad -0.00029 \quad 0] \quad (4-20)$$

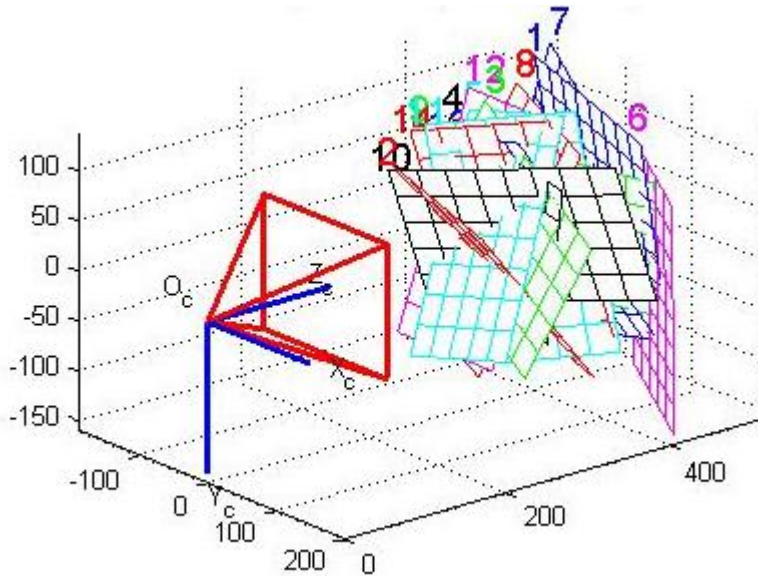


图 4-1 相机内参标定

图 4-1 为相机内参标定结果。相机相对于标定板的外参结果如下：

$$R = \begin{bmatrix} 0.0099 & 0.9622 & -0.2721 \\ 0.9861 & 0.0358 & 0.1623 \\ 0.1659 & -0.2699 & -0.9485 \end{bmatrix} \quad (4-21)$$

$$T = [-89.6507 \quad -129.5371 \quad 477.0779] \quad (4-22)$$

分别采用传统的 Bundle Adjustment 算法和改进版 Bundle Adjustment 算法标定，结果如表 4-1

表 4-1 标定数据及结果

Original	Calibrated by Algorithm	Ground truth
$R = \begin{bmatrix} 0.0034 & 0.9567 & -0.2475 \\ 0.9782 & 0.1087 & 0.1934 \\ 0.1693 & -0.3485 & -0.9647 \end{bmatrix}$	$R = \begin{bmatrix} 0.0123 & 0.9872 & -0.3026 \\ 0.9712 & 0.0475 & 0.1632 \\ 0.1845 & -0.2469 & -0.9367 \end{bmatrix}$	$R = \begin{bmatrix} 0.0099 & 0.9622 & -0.2721 \\ 0.9861 & 0.0358 & 0.1623 \\ 0.1659 & -0.2699 & -0.9485 \end{bmatrix}$
$T = [-93.2837 \quad -134.6732 \quad 484.0723]$	$T = [-88.6790 \quad -123.6543 \quad 475.7692]$	$T = [-89.6507 \quad -129.5371 \quad 477.0779]$

以真实结果作为世界坐标系原点，将传统 Bundle Adjustment 算法和改进之后的 Bundle Adjustment 算法的所标定的标定板投影到该坐标系下，得到结果如下图：

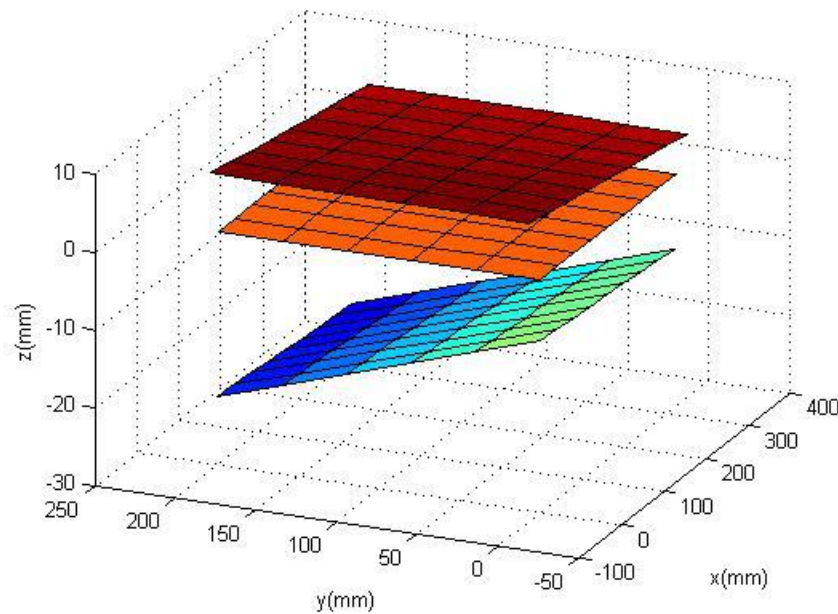


图 4-2 3D 点重投影结果

最上端为改进的 Bundle Adjustment 算法，中间为真实值结果，下端为传统 Bundle Adjustment 算法。计算传统算法和改进算法的精度平均值整理如图 4-4 所示。由此可见改进的 Bundle Adjustment 算法相对于传统算法而言精度有一定提升。

在运算时间上改进 Bundle Adjustment 算法要高于传统 Bundle Adjustment 算法，采用 300 个左右特征点，传统 BA 算法运算时间约为 320ms，而改进 BA 算法的平均运算时间约为 380ms（测试机 CPU 为 i5-5200U）。

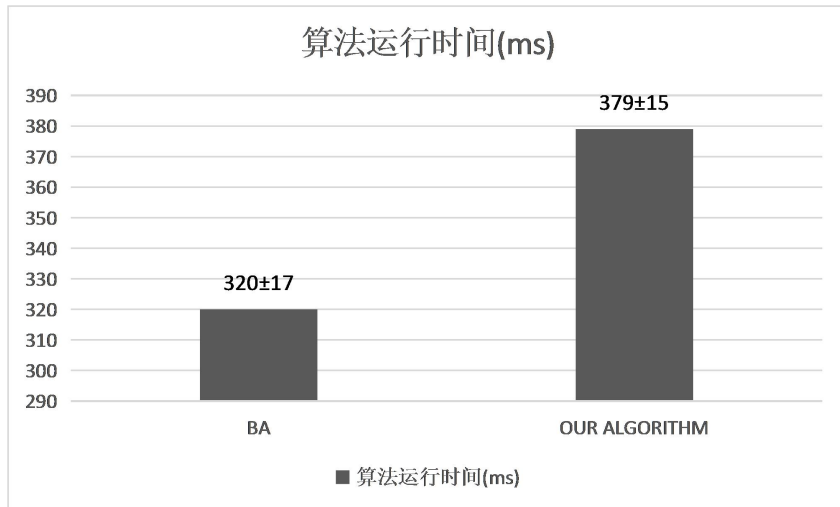
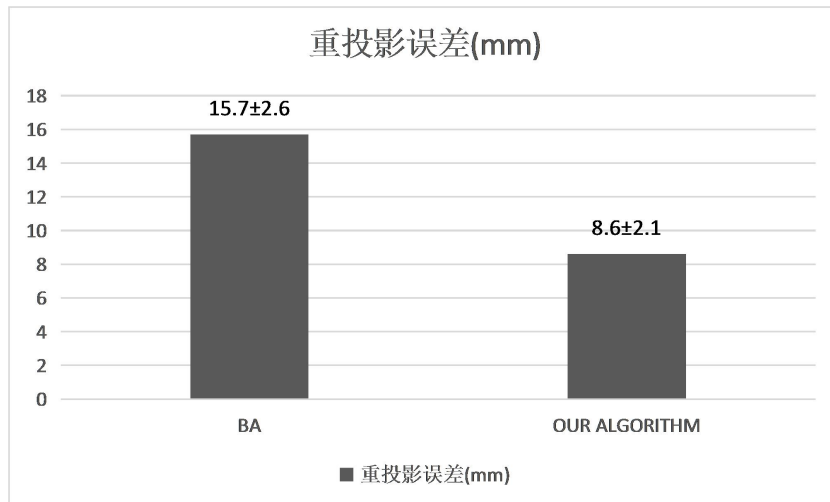


图 4-3 算法运行时间比较



4.5 本章小结

本章针对传统 Bundle Adjustment 算法并没有利用图像信息，只是简单将图像抽象成特征点的信息的特点，提出了一种既应用特征点位置信息，又利用特征点周围的图像信息的改进 Bundle Adjustment 算法，该算法应用李群的连续性和李群李代数的相互转换关系进行求导，并应用高斯-牛顿法进行迭代直至找到局部最优解。实验证明，该算法相对于传统 Bundle Adjustment 算法精度有一定提升，但运算时间相对而言要延长许多，该算法不失为一种在运算性能充足的条件下的高精度算法。

第 5 章 无人机室内定位实验

5.1 引言

本章将对前文提出的主要算法进行实验验证与分析，在无人机上搭建了实验平台，编写了 PC 端上位机软件和无人机端图像采集程序，以及上位机和下位机之间的通信程序。进行了基于数据集和实际场景的测试。

5.2 实验硬件

在 visual SLAM 系统中，硬件部分是程序运行的基础。硬件的好坏与否直接影响系统的整体性能。整个 SLAM 系统硬件架构大致分为六大部分：无人机平台，处理器，xtion pro 相机，2.4G 通信模块，图传模块和处理器。组织结构如图 5-1 所示。

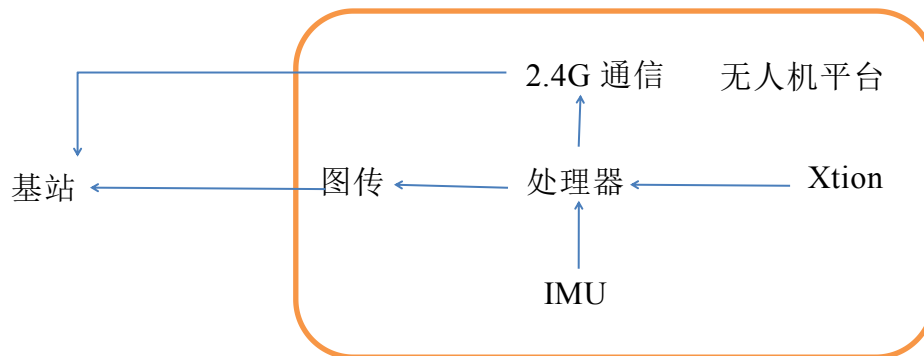


图 5-1 硬件架构

无人机采用的是大疆 M100 无人机开发平台，如图 5-2 所示。M100 是大疆公司于 2014 年推出的针对用户二次开发而设计的无人机，其可以将很多飞行数据通过串口反馈给用户，并且用户可以通过大疆公司开发的 SDK 来控制无人机的飞行，这款平台非常适合当前系统开发。

由于 SLAM 算法运算量较大，为了保证每秒 30 帧的实时性，我们最终采用了 Intel 公司出品的 NUC 迷你智能电脑，其 CPU 为 i7-6770HQ，最高能够满足八个线程同时运行，具有很高的运算速度，而且其有 4 个 USB 口，这样既能够满足 SLAM 的多线程运算需求，又能够满足接入 Xtion Pro，IMU，图传等传感器输入信息，以及输出的图像信息。



图 5-2 大疆 M100 无人机

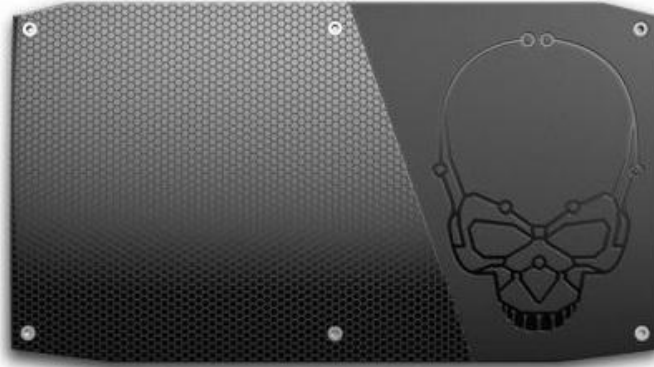


图 5-3 NUC

图像输入部分采用的是华硕出品的 Xtion pro live，类似于微软出品的 kinect，但是比 kinect 体积更小，重量更轻，价钱更低廉，有效距离为 0.8-4m 之间，USB 可以直接供电，比 kinect 供电更加方便，可以得到类似于 kinect 的 RGB 图像以及深度图像，支持 Windows，Linux 等各种开发平台，可以由 C++ 语言进行相应的应用级别开发。可以挂载到无人机上面采集 RGB 图像信息与深度图像信息，从而得到空间点云信息，进而构建空间 3D 点云模型。

IMU 部分采用的是 MPU-6050，其角速度测量范围最高可达 $\pm 2000^{\circ}/\text{sec}$ ，可以对快速运行进行准确的跟踪，在加速度测量方面最高可达 $\pm 16g$ 。该产品与主控可以通过 SPI 接口进行快速通信，最高可达 20MHz，完全能够满足通信带宽需求。其内部还包含温度传感器，能够提供该芯片的温度信息，以便进行温度补偿。以其较高的性价比被广泛的应用于智能手机，平板设备，可穿戴设备，游戏机之中。在 SLAM 算法中其可以为算法提供无人的初始化位姿，为特征提取和后端优化部分提供初始边界条件



图 5-4 Xtion pro live



图 5-5 MPU-6050

在通信部分采用的是 2.4G 模块。硬件架构抛弃了传统的蓝牙和 ZigBee 模块，2.4G 的传输性能较普通的蓝牙要高出很多，并且延迟降低到了 us 级别，完全能够满足高实时性传输的需求。



图 5-6 2.4G 通信模块

5.3 实验软件

实验软件主要架构在 ubuntu 操作系统上，其实 linux 的发行版之一，也是 linux 最受欢迎的发行版。linux 系统相对于常用的 windows 操作系统具有系统占用空间小，开发快捷，第三方库齐全等特点，因此深受广大程序员的喜爱，

成为开源软件开发的主要平台。

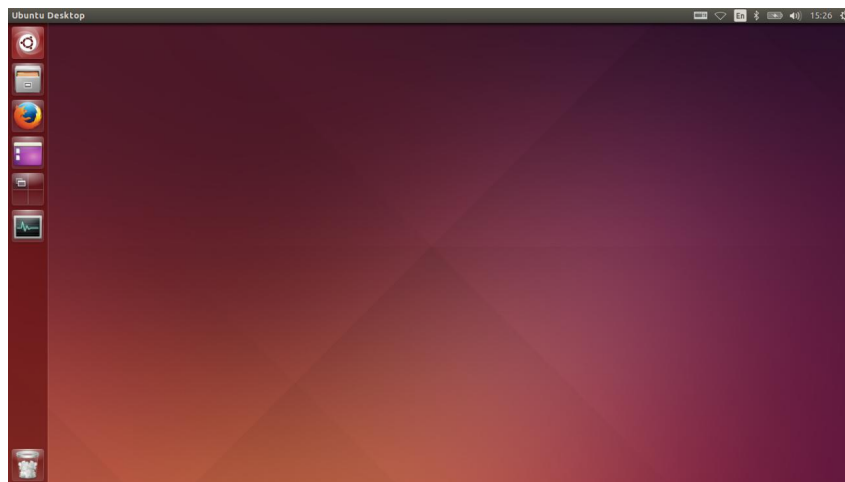


图 5-7 ubuntu 操作系统界面

实验界面开发主要基于 QT，QT 是一个跨平台 C++ 图形用户界面应用程序开发框架，支持包括 linux 在内的几乎所有常见的操作系统，如 Microsoft Windows，Linux，Solaris，SunOS，HP-UX，Digital UNIX 等等。并且其基于面向对象的开发方式和对 2D/3D 图形渲染的良好支持，使其具有大量的开发者和绚丽的图形界面。

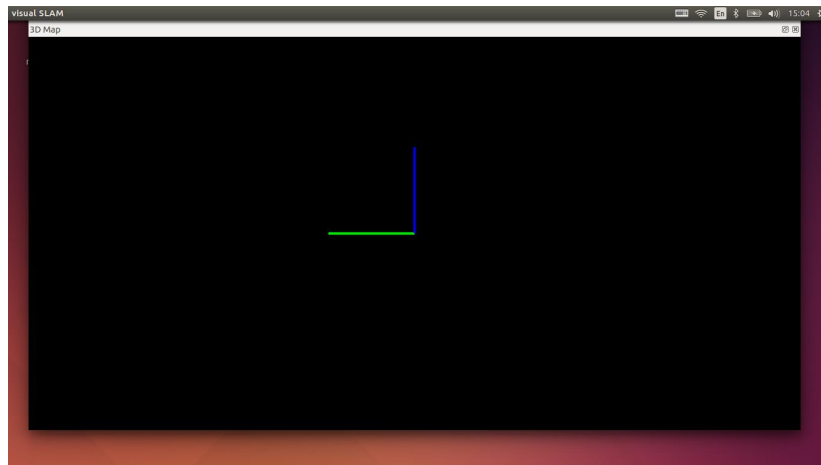


图 5-8 实验开发界面

在软件编写的过程中采用了一些非常优秀的开源第三方库如 OpenNI，OpenCV^[51]，Ceres^[52]，g2o^[53]，Eigen，CSparse，Pangolin 等等。这些第三方库被应用到程序的各个方面。图像采集采用 OpenNI 开源库，其支持 Kinect，Kinect2，xtion pro 等等深度相机，能够快捷的获取深度相机的彩色和深度图像。图像的通用处理采用 OpenCV 开源库，OpenCV 是一个跨平台计算机视觉库，实现了计算机视觉和图像处理方面的很多通用算法。在优化过程中采用 Ceres，g2o，Eigen，CSparse，其中 g2o 和 ceres 是两种被广泛应用在凸优化领域的数

学运算库，将模型构建，迭代方法都集成进系统中，方便用户开发。

5.4 数据集测试实验

在进行正式实验之前首先采用 SLAM 常用的数据集进行精度和运行时间等详细参数的测试，由于数据集有真实结果，可以对算法的准确性进行验证，因此算法在真实发布之前都会进行测试。

本次测试采用的是德国慕尼黑工业大学计算机视觉组公开的 TUM 测试集，该测试集具有左右相机和对应的深度图信息以及此时的以四元数表示 IMU 信息以及对应的相机运行的真实结果。



图 5-9 TUM 数据集

将 TUM 数据集作为输入放入 SLAM 系统之中，经过计算得到了如图 5-10 红色所示的路径，其中显示结果是将真实相机运动路径结果投影到 x-y 平面下得到的，而蓝色的结果是 TUM 数据集提供的真实结果投影的到 x-y 平面下得到的真实的结果，对比如图 5-10 所示。

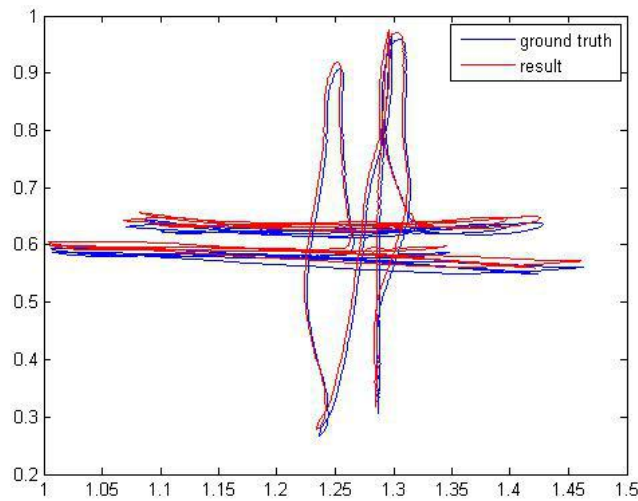


图 5-10 真实结果与算法得出结果对比图

可以发现算法运行结果和真实结果在整体趋势上趋于一致，但是在局部细节上仍然有细微差距。但已经基本满足定位需求，记录每一帧误差与时间的变化趋势图，结果如下图 5-11。

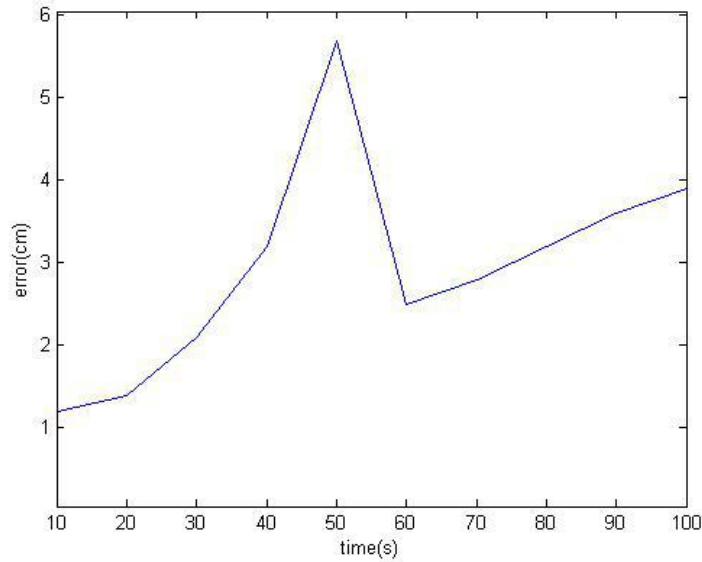


图 5-11 误差与时间关系

通过图 5-11 可以发现随着时间的增加，系统的误差也在增加，这是由于 SLAM 系统同样也是一种基于增量式的定位方法，前一帧的误差同样也会累计到后一帧上，所以误差的整体趋势一定会不断增加。并且增加的速度会随着图像帧数的增多变得越来越快，最终有可能导致结果不再收敛，出现明显的错误与真实结果相差非常远。由图 5-11 所示，在第 60 帧左右，误差并没有随着时间的增加而不断增加，而是出现了一次断崖式的下降，由 5.7cm 左右直接降到了 2.3cm 左右，原因是由于图像中存在闭环检测。在第 60 帧左右，相机重新回到了之前来到过的位置，由于闭环检测的作用不仅仅能够判断该位置来到过，而且能够对之前构建的地图进行修正，同时能够对之前运动的路径进行修正。闭环检测算法是减小 slam 算法累计误差的强有力的手段之一。由于闭环检测修正了累计误差，因此大大降低了平均每一帧误差。

在上述测试的基础之上本文对比了目前开源的主流算法与当前算法的精度比较。测试方法同样是采用多个数据集运行，对比通过 SLAM 计算得到的关键帧的位置和官方给出的关键帧的位置的距离，并提取平均值，得到如图 5-12 所示的结果。



图 5-12 单帧平均误差

可以发现该算法运行精度可以与大多数主流算法相媲美，已经超过了之前 SLAM 广为流行的 PTAM 和 SVO^[54]等算法，但是仍然距离目前精度最高的 ORB-SLAM 算法有一定差距。原因可能是由于构建地图的线程运行时间相对于 ORBSLAM 较长，在同样的运行时间内能够添加进地图的关键帧数目较少，影响了精度。

在运行时间上，本文比较了主流开源算法和当前算法的单帧运行时间。测试采用 640x480 分辨率，测试机为 i5-5200U 结果如下表：

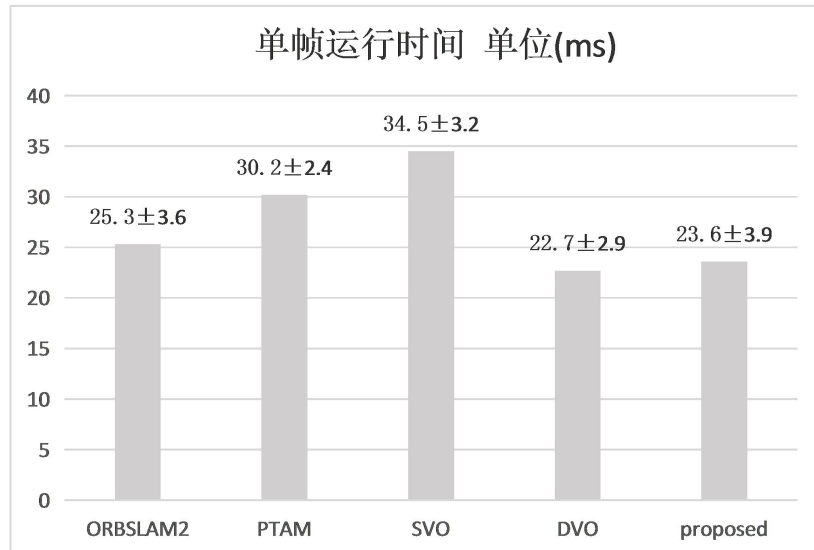


图 5-13 单帧运行时间

可以发现算法在运行时间上相对于主流算法而言较短，实时性较高，这得益于算法在特征值匹配过程中的优化。使得前端特征提取线程在到达同样运行

结果的条件下，能够更快的计算出结果。

‘由于优化过程运行时间较长因此 LocalMapping 线程大约为 Tracking 线程的 10 倍左右,但是由于多线程的存在使得算法在前端处理和后端建立地图的过程中得以同时开展，因此可以在实时的条件下解决 SLAM 问题。

在 SLAM 问题的逐渐从实验室中走到实际应用场合的过程中，随着地图的慢慢变大，关键帧的逐渐增多，优化的时间逐渐变长，最后可能出现优化时间过长导致无法实时的构建地图最终导致程序崩溃的情形，在本文提出的算法之中，采用的 ORB-SLAM 相同的构建地图时的优化方法。即只对与当前帧有共同特征点的帧进行优化，而其它帧保持固定不动的状态。由于添加当前帧对其它帧的影响并不是很大，因此不将其纳入优化范围是一个相对优化的策略。下图反应了随着时间增加总关键帧的多少。

如图 5-14 所示，随着时间的增加参与建立地图优化的关键帧的边的数目并不是线性增加的，而是随着时间的增加增加的趋势逐渐减缓，采用上述建立地图的策略能够保证在建立地图并优化的过程中，优化时间能够在可以接受的范围之内，在关键帧边数为 1400 左右时，后端优化的参数为 200ms 左右，这完全在能够接受的范围之内，之后的关键帧边数基本上在 1400 个之内，因此能够保证实时的建立地图信息。

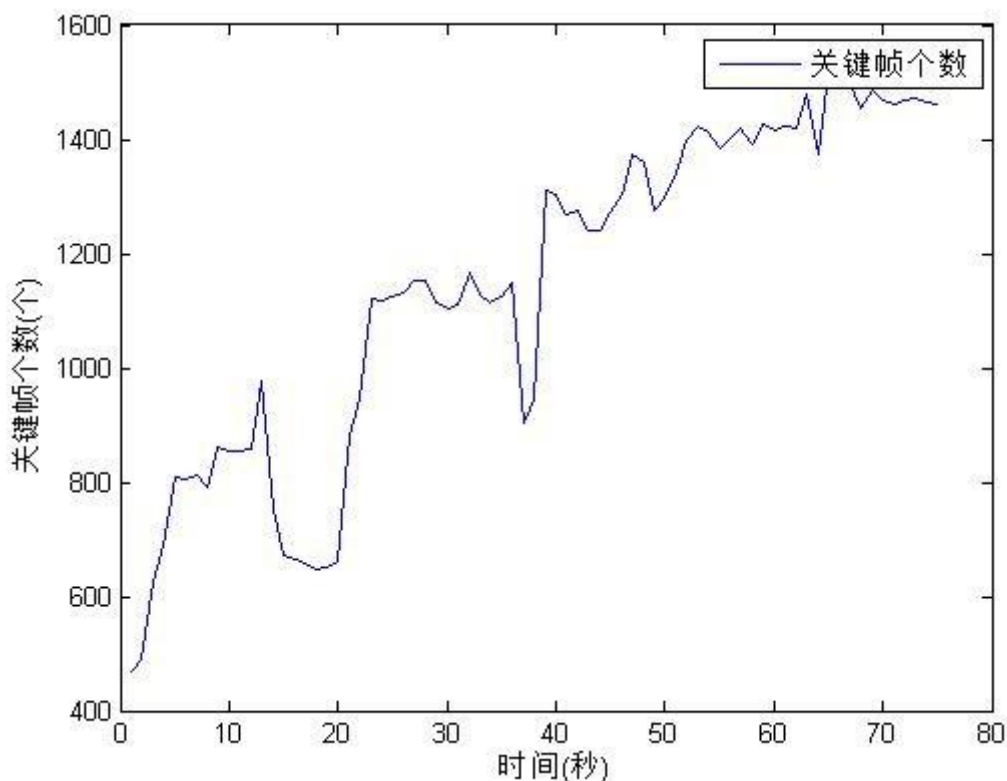


图 5-14 关键帧与时间变化关系图

5.5 无人机室内实际测试

在完成数据集测试实验之后，采用实际场景进行测试，无人机从中间平台起飞之后开始移动并运行 SLAM 算法。

飞行过程中部分结果如图 5-16 所示，左上角为特征检测部分，绿色表示非常温和的匹配结果，黄色表示可能的匹配结果，左下角是建立的空间 3D 模型，右边是点云图，并且在其中用粉色的线标明无人机在空间中的移动路径，粉色的点标明该点为关键帧。红色，绿色和蓝色的坐标轴分别为世界坐标系的 x 轴， y 轴和 z 轴。



图 5-15 无人机实际定位飞行

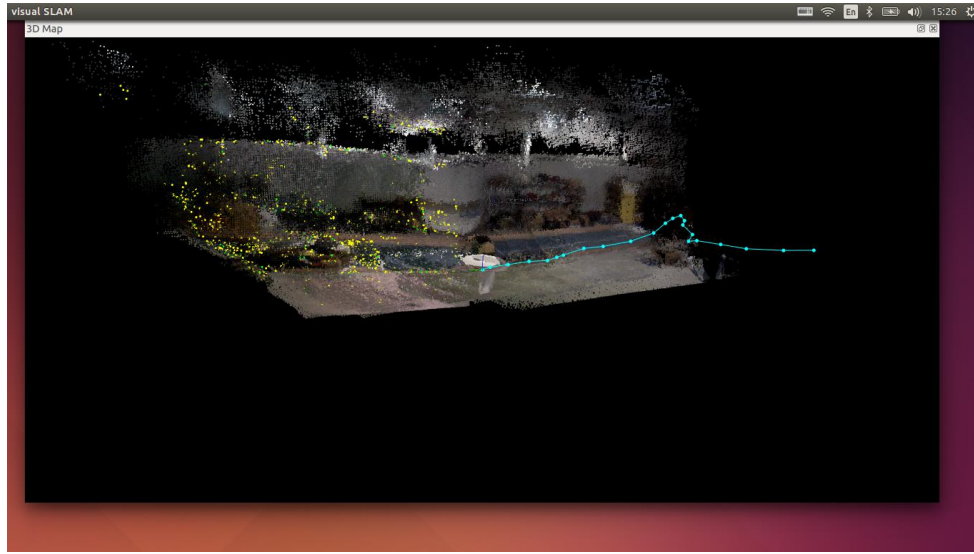


图 5-16 实际场景测试

考虑到数据集测试相对于真实场景测试仍然存在差异，虽然数据集具有真实结果和相机采集每张图片时的姿态，能够对每一帧算法运行的结果和数据集真实的结果进行对比，但数据集采集的环境相对固定，相机的移动距离也相对较短，因此需要在真实的，移动相对较大的环境中进行测试。

5.5.1 无人机定距直线飞行实验

将无人机置于初始位置，分别将其于地面 1m 高度处用遥控器操作使其飞行约 1-10m，分别测定算法得到的无人机运动距离和真实距离之间的差距如下表 5-1。

表 5-1 直线飞行数据

飞行距离(m)	姿态四元数	坐标
1.032	[0.002 -0.002 0.002 1.000]	[0.0209 1.0607 0.0306]
1.987	[-0.019 -0.026 -0.014 0.999]	[0.0550 2.0485 0.0718]
3.123	[0.015 -0.036 0.051 0.998]	[0.1112 3.2630 0.1283]
4.073	[-0.001 -0.003 0.005 1.000]	[0.1304 4.2648 0.0865]
4.985	[0.040 -0.023 0.012 0.999]	[0.1369 5.0936 0.1495]
5.897	[0.025 -0.018 0.007 0.999]	[0.1232 6.0428 0.1950]
6.938	[0.005 0.010 0.001 0.999]	[0.3305 7.2755 0.3310]
8.102	[0.010 -0.001 -0.009 0.999]	[0.2093 8.4789 0.3857]
9.053	[-0.007 -0.015 -0.015 0.999]	[0.2808 9.4655 0.3186]
10.140	[-0.004 -0.043 -0.045 0.998]	[0.3878 10.4893 0.3919]

将表 5-1 的数据结果整理如图 5-17,5-18 所示。

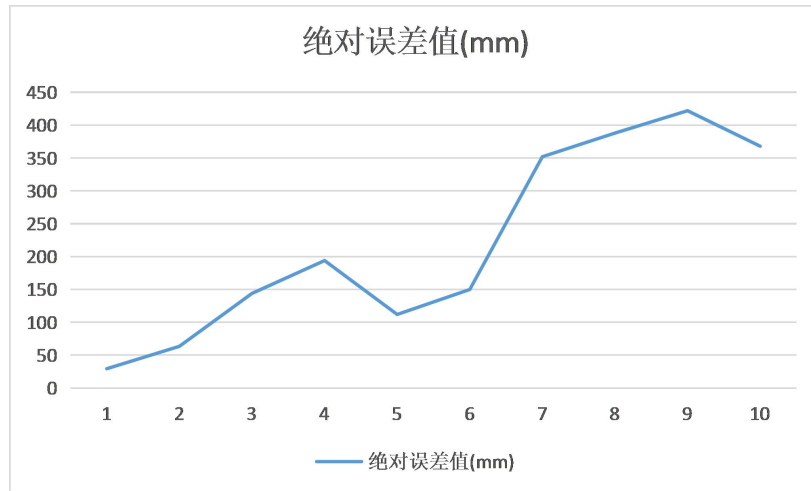


图 5-17 无人机定位绝对误差值

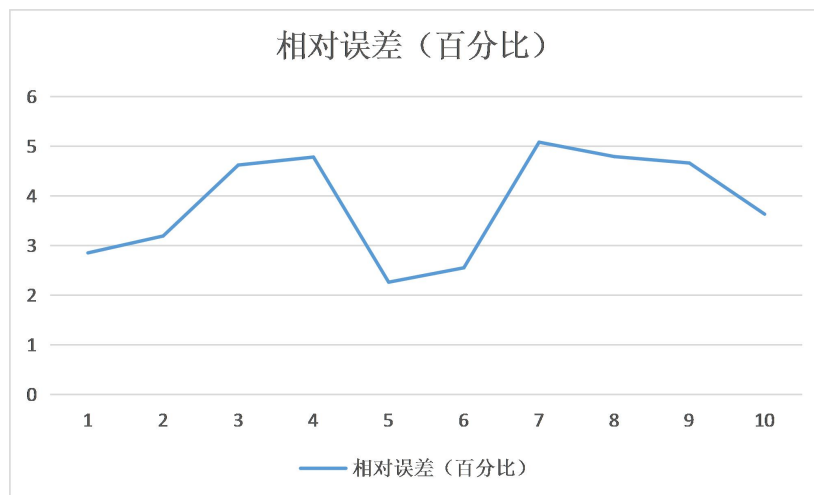


图 5-18 无人机定位相对误差值

由图可知，无人机定位的误差随着飞行距离的增大而增大，这也正是由于基于里程计原理的定位方式所致，但相对误差并没有明显的增多，基本维持在5%以下，考虑到室内环境的大小，该算法在没有闭环检测的条件下的精度能够大体满足室内的局部定位要求，但是精度仍然有待提升。

5.5.2 无人机随机飞行实验

为了进一步模拟无人机在未知环境的定位情况，将无人机在室内以随机的运动方式飞行，并最终回到出发点，测定此时算法得到的无人机的位置与真实无人机位置之间的偏差，得到的实验数据如表 5-2 所示。

表 5-2 无人机原点飞行实验误差

无人机飞行距离(m)	误差(m)	相对误差(百分比)
1.54	0.06 ± 0.02	3.97 ± 1.27
3.48	0.15 ± 0.04	4.35 ± 1.15
5.78	0.19 ± 0.04	3.46 ± 0.69
6.47	0.21 ± 0.07	3.20 ± 1.08
8.35	0.24 ± 0.06	2.89 ± 0.72
10.45	0.32 ± 0.11	3.13 ± 1.05
12.68	0.46 ± 0.09	3.67 ± 0.71
15.89	0.34 ± 0.11	2.13 ± 0.69
18.80	0.50 ± 0.15	2.67 ± 0.80

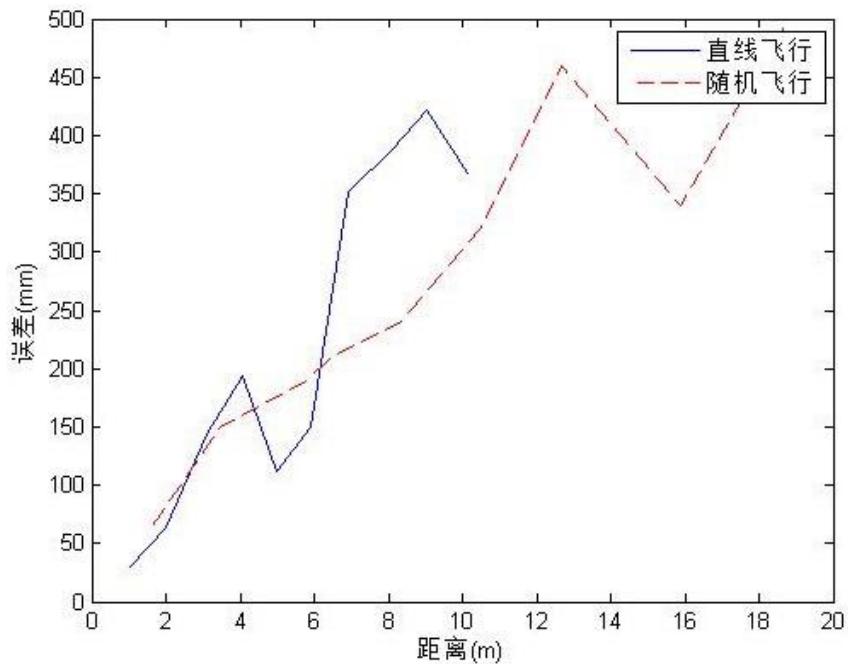


图 5-19 直线飞行与随机飞行误差绝对值对比图

实验表明由于随机飞行会在算法中进行闭环检测的步骤，在相同的运动距离的条件下，无人机原定定位实验中无人机回到原点的定位精度要略高于无人机直线飞行的定位精度。随机飞行原地定位实验的相对误差一般可以在 3% 左右，基本上满足了短程室内的定位需求。

5.6 本章小结

本章对整个 SLAM 系统进行了基于数据集和基于场景的测试，在基于数据

集的测试中分别对 SLAM 运行结果和真实值进行比较, 算法运行结果和真实结果在整体趋势上趋于一致, 但是在局部细节上仍然有细微差距。在每帧的平均误差方面上, 可以与大部分 SLAM 算法相媲美, 但是距较先进的算法任然存在一定差距。在运算时间上与大多数算法区别不大。在实际场景的测试中, 无人机随机飞行的轨迹误差要小于无人机在单方向的飞行轨迹误差, 在随机飞行 18 米左右时, 相对误差约在 3%, 基本上能够满足室内定位的需求。

结 论

尽管在在学术研究领域 visual SLAM 已经取得了相对不错的效果，但其在定位精度和运算速度上仍然有较多可以优化之处。针对该问题，本文分别从特征提取方法，Bundle Adjustment 优化，以及 Xtion pro 相机标定的方法方面进行研究，根据已完成的研究内容，将本文的研究成果总结如下：

(1) 采用了基于 ORB 特征为基础的特征匹配方法，针对之前的算法中的匹配方法之中具有很多的无效匹配的问题，提出了一种基于相机运动模型的解决办法，并在此基础之上对相机的运动速率及运动方向进行平滑，得到一个相对理想的相机位置，最后将空间中的 3D 点反向投影到相机的成像平面上，最终可以非常有效的进行匹配。

(2) 本文深入的剖析了深度相机的成像原理，针对畸变产生的原因进行了深入的分析并采用平面的实验得到相机的误差大小，针对误差的数据提出了一种基于指数的误差拟合方法，并应用该方法对误差进行了针对性的补偿，应用该方法基本上消除了 xtion pro live 深度相机的畸变。基本上满足了 SLAM 算法中点云深度的精度需求。

(3) 针对 Bundle Adjustment 算法没有充分利用图像信息的问题，提出了一种改进的 Bundle Adjustment 算法。该算法利用链式求导和梯度下降的方法求取像素点信息与空间 3D 点，相机姿态坐标的导数，对空间中的相机位姿进行优化，得到最佳的相机姿态坐标。经过实验验证，该算法相对于传统 Bundle Adjustment 算法而言精度略有提升，运算时间稍长。

总结本文研究的内容，本人认为还有以下几个方面需要进行深入的研究。

1. 在闭环检测的过程中，仍然采用的是较为传统的基于词袋模型的算法，如今已经有基于深度学习的闭环检测的论文发表，在日后的工作中，进行将深度学习融入闭环检测的试验。

2. 在 IMU 融入部分，仅仅将 IMU 信息作为一种先验条件加入算法之中，优化算法以 IMU 提供的点为初始信息进行优化，并没有将完整的 IMU 信息与特征点的信息相互融合，因此需要在此进行进一步优化。

参考文献

- [1] 秦博, 王蕾. 无人机发展综述[J]. 飞航导弹, 2002(8):4-10.
- [2] 淳于江民, 张珩. 无人机的发展现状与展望[J]. 飞行导弹, 2005(2):23-27.
- [3] 冯琦, 周德云. 军用无人机发展趋势[J]. 电光与控制, 2003, 10 (1) :9-13
- [4] 高新华. 无人机在农业精准化方向中的作用[J]. 资源节约与环保 2015 (10) :171-171
- [5] 高劲松, 朱荣刚, 陈哨东, 郭涛, 孙隆和. 航空工业的一个新发展机遇--用无人机反恐[C]. 中国科协学术年会, 2005.
- [6] K.Nonami, 宪三野波, 肖阳. 自主飞行机器人:无人机和微型无人机[M]. 国防工业出版社, 2014 :25-36.
- [7] 房建成, 陶冶, 于歌. 空战新兵[M]. 花城出版社, 2010:128-130
- [8] 王念, 刘真志, 陈俊. 无人机在救灾中的应用综述[J]. 中国化工贸易, 2016, 8(8)
- [9] Subrata, Goswami, 杜利东. 室内定位技术[M]. 国外科技新书评介, 2014 (3) :19-19
- [10] 王小辉, 汪云甲, 张伟. 基于 RFID 的室内定位技术评述[J]. 传感器与微系统, 2009, 28 (2) :1-3
- [11] 权美香, 朴松昊, 李国. 视觉 SLAM 综述[J]. 智能系统学报, 2016, 11(6):768-776.
- [12] JA Castellanos, R Martinez-Cantin, JD Tardós, J Neira Robocentric map joining: Improving the consistency of EKF-SLAM.[J] Robotics & Autonomous Systems, 2007, 55 (1) :21-29
- [13] 祝继华, 郑南宁, 袁泽剑, 张强. 基于中心差分粒子滤波的 SLAM 算法[J]. Acta Automatica Sinica, 2010, 36 (2) :249-257.
- [14] Montemerlo M, Thrun S, Koller D, et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem[C]// Arch Environ Contam Toxicol, 2002:593-598.
- [15] H Strasdat, JMM Montiel, AJ Davison, Visual SLAM: Why filter?[J] Image & Vision Computing, 2012, 30 (2) :65-77.
- [16] S Thrun Probabilistic robotics[M] Communications of the Acm, 2002, 45 (3) :52-57
- [17] G Welch, G Bishop An Introduction to the Kalman Filter[M] University of

- North Carolina at Chapel Hill , 1995 , 8 (7) :127-132
- [18] Liu H, Zhang G, Bao H. A Survey of Monocular Simultaneous Localization and Mapping[J]. Journal of Computer-Aided Design & Computer Graphics, 2016.vol.28,no.6:855-868
- [19] G Klein , D Murray. Parallel Tracking and Mapping for Small AR Workspaces[J]. IEEE & Acm International Symposium on Mixed & Augmented Reality.2008 :1-10.
- [20] Wikipedia.Levenberg – Marquardt algorithm[EB/OL]. [https://en.wikipedia.org/wiki/Levenberg – Marquardt_algorithm](https://en.wikipedia.org/wiki/Levenberg_-_Marquardt_algorithm),2017-6-19/2017-6-20.
- [21] Wikipedia.Gaussian-Newton method[EB/OL].https://en.wikipedia.org/wiki/Newton%27s_method,2017-6-19/2017-6-20.
- [22] P. Newman and K. Ho, Slam-loop closing with visually salient features[C]// in proceedings of the 2005 IEEE International Conference on Robotics and Automation,IEEE, 2005:635-642.
- [23] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[M] Autonomous Robot Vehicles. Springer New York, 1990:435-461.
- [24] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[M] Computer Vision – ECCV 2014. Springer International Publishing, 2014:834-849..
- [25] Galvez-López D, Tardos J D. Bags of Binary Words for Fast Place Recognition in Image Sequences[J]. IEEE Transactions on Robotics, 2012, 28(5):1188-1197.
- [26] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]// IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012:573-580.
- [27] Beeson P, Modayil J, Kuipers B. Factoring the Mapping Problem: Mobile Robot Map-building in the Hybrid Spatial Semantic Hierarchy[J]. International Journal of Robotics Research, 2008, 29(4):428-459.
- [28] Liang M, Min H, Luo R. Graph-based SLAM: A Survey[J]. Robot, 2013, 35(4):500-512.
- [29] Wikipedia.SIMD[EB/OL].<https://en.wikipedia.org/wiki/SIMD>.2017-6-15/2017-6-20.
- [30] 高伟, 赵荣彩, 韩林, 庞建民, 丁锐. SIMD 自动向量化编译优化概述[J]. 软件学报 , 2015 , 26 (6) :1265-1284
- [31] Łukasz Żmudzinski , A Augustyniak , P Artiemjew. CONTROL OF MINDSTORMS NXT ROBOT USING XTION PRO CAMERA SKELETAL

- TRACKING[J]. Technical Sciences , 2016 , 19 (1) :71-81
- [32] ASUS. Xtion pro live.[EB/OL] https://www.asus.com/us/3D-Sensor/Xtion_PRO_LIVE/.2016-5-20/2017-6-20.
- [33] J Geng Structured-light 3D surface imaging: a tutorial[J] Advances in Optics & Photonics , 2011 , 3 (2) :128-160
- [34] Stuelpnagel J. On the Parametrization of the Three-Dimensional Rotation Group[J]. Siam Review, 1964, 6(4):422-430.
- [35] Fuchs S, Hirzinger G. Extrinsic and Depth Calibration of ToF-cameras[J]. Proc. IEEE Conf. Comput. Vision Pattern Recog, 2008:1-6.
- [36] M Lindner , A Kolb Lateral and Depth Calibration of PMD-Distance Sensors[J]. Springer Berlin Heidelberg , 2006 , 4292 (9) :524-533.
- [37] Chow J C K, Lichti D D, Glennie C, et al. Improvements to and Comparison of Static Terrestrial LiDAR Self-Calibration Methods[J]. Sensors, 2013, 13(6):7224.
- [38] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]// IEEE International Conference on Computer Vision. IEEE, 2012:2564-2571.
- [39] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [40] Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[J]. Computer Vision & Image Understanding, 2006, 110(3):404-417.
- [41] Rosten E, Drummond T. Fusing Points and Lines for High Performance Tracking[C]// Tenth IEEE International Conference on Computer Vision. IEEE, 2005:1508-1515 Vol. 2.
- [42] Brown M, Lowe D G. Invariant Features from Interest Point Groups[C]// British Machine Vision Conference 2002, BMVC 2002, Cardiff, UK, 2-5 September. DBLP, 2002:656--665.
- [43] Calonder M, Lepetit V, Strecha C, et al. BRIEF: Binary Robust Independent Elementary Features[C]European Conference on Computer Vision. Springer-Verlag, 2010:778-792.
- [44] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point set[C], Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1987:Vol 5,698-700
- [45] Sharp G C, Lee S W, Wehe D K. ICP registration using invariant features[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002, 24(1):90-102.
- [46] Chang L, Zha F, Qin F. Indirect Kalman Filtering based Attitude Estimation

- for Low-cost Attitude and Heading Reference Systems[J]. IEEE/ASME Transactions on Mechatronics, PP(99):1-1.
- [47] Triggs B, Mclauchlan P F, Hartley R I, et al. Bundle Adjustment — A Modern Synthesis[C]//International Workshop on Vision Algorithms: Theory and Practice. Springer-Verlag, 1999:298-372.
- [48] Wikipedia.Euclidean group[EB/OL].https://en.wikipedia.org/wiki/Euclidean_group.2017-6-12/2017-6-20.
- [49] S Baker, I Matthews Lucas-Kanade 20 Years On: A Unifying Framework[M]. Kluwer Academic Publishers , 2004 , 56 (3) :221-255
- [50] Varadarajan V S. Lie groups, Lie algebras, and their representations[J]. 1974, 659(5):351.
- [51] OpenCV. OpenCV[EB/OL]<http://opencv.org/>.2017-5-31/2017-6-20.
- [52] S. Agarwal, K. Mierle.Ceres solver[EB/OL].<http://ceres-solver.org>.2016-12-23/ 2017-6-20.
- [53] Kümmerle R, Grisetti G, Strasdat H, et al. G2o: A general framework for graph optimization[C]IEEE International Conference on Robotics and Automation. IEEE, 2011:3607-3613.
- [54] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]//IEEE International Conference on Robotics and Automation. IEEE, 2014:15-22.

攻读学位期间发表的学术论文

- 1 Duo Xu, YunFeng Gao,ZhengHua Hou Self calibration of binocular vision based on Bundle Adjustment algorithm[C]. International Conference on Intelligent Robotics Applications. (2017) (EI 检索)

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于 visual SLAM 算法的四旋翼无人机室内定位研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：

日期： 2017 年 7 月 6 日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：

日期：2017 年 7 月 6 日

导师签名：

日期：2017 年 7 月 6 日

致 谢

值此论文完成之际，谨向曾经给予我关心和帮助的老师 and 同学们致以诚挚的感谢！

衷心感谢导师高云峰教授和王珂老师对本人的悉心指导。由于老师的指导，我在工程实践中得到了很好的锻炼。在课题的方向上得到了深入了解。老师严谨的治学态度、求实的工作作风、前沿的思考理念是我的典范。他们的言传身教将使我终生受益，感谢导师在学习、工作和生活等方面给予我的关心和帮助。

感谢郭本生师兄，张明明师兄在课题进展、方法研究、论文撰写和生活上给予的帮助感谢实验室硕士生同学龙学雄、朱景泉在学习以及生活中给予的帮助；感谢实验室侯政华在课题上给予的无私支持和帮助；感谢本科生师弟仲星光在共同工作过程中的支持。感谢京东公司配送员许少军为本人配送了各种实验器材。当遇到困难时通过与他们的交流可以找到解决的方法，与他们共同生活学习使我受益匪浅！