

毕业设计（论文）题目：视觉惯性组合定位里程计

学 院：光电学院

专 业：光信息科学与技术

班 级：04131201

姓 名：高文良

指导教师：孟彦彬

摘要

近些年随着惯性导航技术和机器视觉技术的进步，机器人领域导航算法开始愈加成熟。较为代表性的便是单目视觉融合惯性测量系统(VINS)的算法，由于其成本低，体积小，功耗低，因此更多的用在机器人，无人机等依赖定位的设备上面。VINS 可以根据计算摄像头拍摄到的前后帧间特征点的偏移再经过融合 IMU 数据计算出摄像头本身的移动和旋转，它不像 GPS 传感器有信号干扰等限制，因此可以搭载在无人机或者机器人设备上用于灾区搜救，地形勘测等极端场。

本文探索了当前该领域内的主流算法之一，基于 MSCKF 的紧耦合视觉惯性导航算法的数学原理和详细执行步骤，并用 ROS(Robot Operate System) 仿真器对该算法进行了实现。

关键词：惯性导航 机器视觉 非线性优化 多传感器融合 视觉惯性测量系统 ROS

Abstract

With the recent advance in inertial navigation technology and computer vision technology, robotics navigation algorithm becomes more mature. In particular, the monocular visual-inertial system(VINS) that consists of only a camera and an inertial measurement unit (IMU) forms a great minimum sensor suite due to its superior size, weight, and power characteristics. Therefore, it is more used in Robotics, Unmanned Aerial Vehicles (UAV) and other equipment. VINS is based on calculating the offset of feature points between inter-frame and fusion with IMU data, then the movement and rotation of the camera are solved. Unlike GPS sensor limited in pool signal area, it can be mounted on a drone or robot for the disaster searching and rescue, and other extreme scenes. This article explores one of the current mainstream algorithms in the VINS field, the Tightly-Coupled Monocular Visual-Inertial Fusion based on nonlinear optimization. Then it studied its mathematical principles and implementation details, and implemented the algorithm by using the simulator of ROS (Robot Operate System).

Key Words: Inertial Navigation, Computer Vision, Non-Linear Optimization, Multi Sensor Fusion, VINS, ROS

目录

第 1 章 绪论	1
1.1 课题背景与意义	2
1.2 简述	3
1.3 论文课题与内容安排	4
第 2 章 视觉惯性里程计	5
2.1 视觉惯性里程计的方案	5
2.2 基于滤波器的 SLAM 方法	5
2.3 基于关键帧的 SLAM 方法	6
2.4 惯性器件辅助的方法	6
2.5 MSCKF	7
2.6 标识	8
第 3 章 卡尔曼滤波简介	9
3.1 卡尔曼滤波	9
3.2 扩展卡尔曼滤波器	10
第 4 章 惯性导航基础	13
4.1 基础规定	13
4.1.1 坐标系规定	13
4.1.2 位姿表达	15
4.2 惯性测量单元模型	18
4.2.1 加速度计模型	18
4.2.2 陀螺仪模型	20

4.3 惯性更新过程	21
4.3.1 旋转姿态的更新	21
4.3.2 速度更新	22
4.3.3 位移更新	23
第 5 章 视觉基础	25
5.1 视觉系统基础规定	25
5.1.1 坐标系规定	25
5.2 针孔摄像模型	26
5.3 摄像机观测模型	26
5.4 特征处理的算法	29
5.4.1 特征提取算法	29
5.4.2 特征匹配算法	33
第 6 章 MSCKF	37
6.1 状态空间	37
6.2 误差模型	38
6.3 噪声矩阵	40
6.4 状态扩充	40
6.5 更新过程	41
6.6 状态剔除	42
第 7 章 系统初始化与矫正	43
7.1 状态初始化	43
7.1.1 姿态初始化	43
7.1.2 位置与速度初始化	44

7.1.3 特征点初始化.....	45
7.2 里程计的绝对校正.....	47
7.2.1 姿态的绝对校正	47
7.2.2 里程绝对矫正.....	49
第 8 章 系统构建与实验结果	51
8.1 软件环境.....	51
8.2 仿真	52
8.2.1 系统架构.....	52
8.2.2 仿真数据	53
第 9 章 结论与展望	57
致谢	59
参考文献.....	61
附录	1
附录 A $SO(3)$ 数学基础	1
附录 B 四元数	4
B.1 四元数的定义	4
B.2 旋转四元数	4
B.3 四元数微分方程.....	6

第 1 章 绪论

机器人日新月异。

现在的机器人开发者与开发企业越来越多，越来越普遍，前人的工作与新平台的发展使得机器人开发变得越来越容易。随着计算硬件和传感器的发展，使得开发者能够使用便宜，高性能的处理器以及告诉的互联网，让机器人逐渐具备小型化，快速部署和低维护高可靠性的要求，具有一定能力的机器人仅需很少的维护就行长时间工作。无论是在危险环境中的，例如大疆创新的消费级产品精灵 4 无人机（如图 1-1 所示），火星上的探测器（如图 1-2 所示勇气号火星探测器），对于基于视觉的自主定位技术都有需求。



图 1-1 大疆精灵 4



图 1-2 勇气号火星探测器

然而，大多数机器人还处于缺乏自主，仅仅是被控制的执行器的最基本形式。

每个能够自主移动的机器人都需要明确的三个问题：在哪里？如何到达当前位置？如何到达目标？这些问题可以被总结为识别、定位、里程测量和控制：

- 定位，是将机器人与所在的地图进行映射的过程。同时，这里存在有一个迭代的关系：必须通过地图辅助定位，同时必须通过实时定位进行地图创建；
- 里程测量是使用运动传感器的机器人的位姿（位置和姿态）的过程。一个单纯的里程计是独立于地图和定位的，仅仅与相对起始位姿等有关；
- 控制是机器人移动的方法。机器人使用轮子，走路或者飞行等等方式，受控制的进行移动。

1.1 课题背景与意义

本文的研究重点在于里程计。单纯的里程计是在没有地图的情况下，进行位置和姿态的确定。所有的移动机器人都需要里程计，以确定自己的移动情况，以此来进行下一步的定位、建立地图和控制。将里程计与控制分开，尽量抽象的概括里程计算法，使得里程计可以被集成到配备有所需的传感器几乎任何系统。

有很多里程计系统具有特定的硬件设备，或是在特定的环境下使用，例如GPS，或是水下机器人的多普勒里程计，小型机器人的激光雷达等等。准确的里程计通常依赖昂贵的硬件，需要复杂的校准，或者需要十分庞大的计算量处理。为了使里程计更方便，使用的传感器成本低廉，校准方便，算法可以在单核CPU上运行是十分重要的。

本文我们感兴趣的是用低成本传感器和较少的运算量估计的物体位置和姿态，里程计的设计需要完成以下目标：

- 连续：算法必须正确报告估计的不确定性；
- 大尺度：算法必须适用于长距离和长时间的里程测量；
- 易用性：算法必须使用常见的低成本硬件，以及不过于复杂的校准技术；
- 效率：算法需要在实时设备本身上运行；
- 鲁棒性：算法必须足够健壮在不同的环境中工作，并存在环境干扰的时候不出现问题。

1.2 简述

视觉传感器就是相机，能够提供的数据量巨大的时变信息，从视觉信息可以提取出很多信息。在定位中，专注于寻找环境中的特征，并对图像之间的特征进行匹配，通过多个摄像机或者一个移动摄像机的多个图像进行三角测量[1]，进而获取三维的环境信息，进而可以反推照相机本身的位置信息。

双目立体相机能够快速，准确地[2]得到的深度信息[3]，但尺寸大，价格贵，难以校准。双目立体相机对深度的估计取决于基线之间的距离，标定质量。但是对于单目相机，其主要优点是可以在静止的时候进行深度估计。单目相机通过移动进行深度估计，对于深度的估计是动态的，但他们从缺乏尺度信息的观测。如果相机是唯一的传感器，该系统就无法确定尺度信息的方法。这是由于观测的结果是由照相机将三维的世界投影成为二维的图像。

单目摄像机的这个问题可以通过添加一个视觉的初始化或另外的传感器[4]进行解决。例如使用超声传感器或是激光测距仪可以度量物体的距离，气压计可以给出短时间内随高度变化带来的气压变化而推出高度变化，微机电（MEMS）的惯性测量单元（IMU）使用加速度计和陀螺仪进行组合，提供高频的角速度和加速度信息，虽然有较大的噪声，可以在短时间内给出小范围的尺度信息。

单目视觉惯性系统，它仅由低成本的微机电的惯性测量单元和一个摄像头组成。单目的视觉惯性系统是能够进行环境估计的最小传感器套件。使用单目视觉惯性系统，通过图像的对比匹配迭代计算摄像头的姿态改变[5]。从图片中提取出特征，移动到另一个不同的视角拍摄的另一幅图像再次提取特征。然后，匹配相关的特征，并通过匹配结果，以计算相机的位置和姿态变化。同时，在两帧之间，通过惯性测量单元补全位姿测量，并修正图像测量的量纲。

具体的实现流程为：

- 惯性积分 使用惯性测量单元进行位姿的积分解算

- 图像获取 通常利用相机获取图像;
- 图像校正 对相机进行标定, 通过标定参数移除透镜畸变;
- 特征检测 按照需要对图像进行特征提取;
- 特征匹配 在帧帧之间匹配特征点并进行相应;
- 位姿修正 根据帧间特征点的匹配与惯性测量单元的测量, 计算相机的运动, 并向后校正惯性测量单元的误差;

1.3 论文课题与内容安排

本文的大致分配如下:

第一章主要是绪论, 展示作为本文核心的视觉惯性里程计的意义; 第二章对现在的主流方案进行讨论, 按照主流方案的细分内容、实现难度与优缺点, 选取一个合适的研究方向; 第三章主要讲述本文所讨论到的概率论相关基础, 以及对卡尔曼滤波进行讨论, 再之后介绍文本用到的卡尔曼滤波器, 即扩展卡尔曼滤波器; 第四章对惯性测量单元做一些简介; 第五章主要是对本文所使用的视觉处理方法进行简要的叙述; 第六章为核心章节, 详细阐述了视觉惯性里程计的算法处理流程和细节; 第七章讲述视觉惯性里程计的初始化; 第八章讲述系统构建和实验结果; 第九章为基本结论和对未来的展望。同时, 添加两个附录, 分别简述了 $SO(3)$ 流形的性质和四元数的性质。

第 2 章 视觉惯性里程计

2.1 视觉惯性里程计的方案

视觉惯性里程计就是状态估计器。

视觉惯性里程计状态估计的学术流派，无论是单目视觉的方案[6][7]，双目视觉的方案[8]，或者是使用 RGB-D 相机的方案[9]，都可以分类为滤波器方案[6][10]，与图优化方案[8][11]。基于滤波方法可能需要较少的计算资源，其为全局最优的方案，但滤波器方案直接线性化可能导致次优结果；图优化的方案以更高的计算需求为代价，通过迭代提高了线性化的精度。在实际应用中，通常对视觉的特征进行处理，无论滤波器方案，还是图优化方案，都可以保持恒定的计算复杂程度。可以证明，在一定的条件下，基于滤波器方案和基于图优化的方法[12]之间几乎有相同的结果。然而，由于单目视觉的平台不足以扩展特征的追踪，限制了基于图优化方案的效率。

视觉惯性里程计的另一重分类方法是按照耦合程度，分为松耦合和紧耦合两种。松耦合的方案[7][13]通常使用一个独立的模块，仅处理视觉状态，诸如 PTAM[14]或 SVO[15]，另外使用一个单独的 IMU 融合模块[13]来处理状态的融合估计。松耦合的方案不考虑视觉和惯性信息耦合，所以松耦合的方案无法使用视觉估计校正 IMU 的漂移。紧耦合方案[8]进行视觉和 IMU 测量系统的融合，通常能够得到更好的结果，尤其对于单目视觉的视觉惯性里程计，紧密耦合的方案能够将从 IMU 获取的尺度信息转换成视觉特征的深度估计，能够满足需要明确的视觉尺度的建模需要。

2.2 基于滤波器的 SLAM 方法

早起的 SLAM 方法主要基于滤波的思想来实现，即把机器人的姿态位置以及环境中特征点的位置作为系统状态量，并且认为状态和测量是基于概率模型的，应用较多的是扩展卡尔曼滤波器（Extended kalman filter）。它不是传统意义上的滤波

器，而是一种最优估计，把测量值和预测值结合起来，然后根据预测和测量的协方差矩阵来计算两者下一次迭代的权重。从而得到系统状态的期望值，此时的后验估计方差最小，而且还可以随着迭代的次数逐渐收敛。在 **EKF-SLAM** 中，环境特征点和机器人状态的协方差矩阵会随着特征点的增加而平方增长，因此计算量会逐渐增大，导致算法实时性较差。

二十一世纪初出现了基于扩展卡尔曼滤波器的实时单目**SLAM** 算法[16]，在此之前，**SLAM** 只能利用事先采集好的数据离线运行。实时性意味着牺牲一部分结果精度换取计算量。在单目**SLAM** 中，环境依然是较严重的限制因素，因为计算复杂度随着环境特征点增加平方增长。此外，单目**SLAM** 中的环境特征点初始深度未知，即使很小的测量误差也可能导致迭代后的深度出现非线性误差，系统出现局部最优解。为解决此问题，使用特征点的逆深度坐标表示来减小系统对数值的敏感[17]，以此解决局部解的情况。

2.3 基于关键帧的 SLAM 方法

为实现实时性的单目 **SLAM** 系统，在 2007 年介绍了 **PTAM**（**Parallel Tracking and Mapping**）方法[18]，利用了关键帧的方法来实时定位。当机器人在未知环境中探索时，**PTAM** 方法维持一个稀疏的关键图像集合，以及相应环境特征点检测结果和相机的状态。关键帧的优点之一就是它会保留以前的信息而不是像 **EKF-SLAM** 一样将其边缘化。此外，**PTAM** 将地图构建和跟踪分离在两个线程中并行运行。这样非耦合带来的好处就是可以将图优化应用到地图构建中，每个关键帧都可以看作图中被欧式空间约束的节点，两个关键帧之间的关系为边。

2.4 惯性器件辅助的方法

单目视觉的里程计最大的问题在于缺少空间尺度。因此当只在初始化过程中确定尺度的话，很有可能在之后的测量中发生尺度漂移。因为相机的投射变换模型使其在深度方向没有观测，即远处的大物体和近处的小物体在图像中可能没有区别。

因此需要惯性器件辅助 SLAM，有了加速度计以及陀螺仪的测量，可以恢复两帧图像之间相机的平移和旋转，从而可以通过三角测量计算特征点的深度。

通常的利用 IMU 和单目视觉实现了基于卡尔曼滤波的松耦合的里程计[17]，由于状态测量的信息矩阵并没有在异常检测中使用，所以当视觉观测出错或者跟踪到了移动物体上面，里程计就会出错，因为算法假定空间特征点是静止不动的。

2.5 MSCKF

2006 年首先有人提出 MSCKF[19]，多状态约束卡尔曼滤波器（Multi-State Constraint Kalman Filter），作为一种惯性测量单元辅助的 EKF-SLAM 问题。该 MSCKF 是一个单纯的里程计方法。因为它不建立一个地图，它只是进行一定的线性化，并不受到特征性质假设的影响。在估计中，与 EKF-SLAM 不同的是，地标在相机坐标系的信息代替了地标的位置，将地标特征点的相机观测用作相机位姿的滑动窗口上的约束。该滤波器具有比 EKF-SLAM 更好的结果，同时允许在线校准也有可能对系统的许多参数。相机和 IMU 之间的转换并不总是精确地知道，但可以在线的进行估计[20]。

本文选择了 MSCKF 作为解决方案。

MSCKF 其算法流程为：

- 惯性迭代：接收到的每个 IMU 测量，传播滤波器状态和协方差。
- 图像匹配：每当一个新的图像被获取：
 - ◆ 扩大当前的状态和协方差矩阵
 - ◆ 图像处理模块开始处理
- 状态更新：当给定的图像的特征测量有效时，执行 EKF 更新。

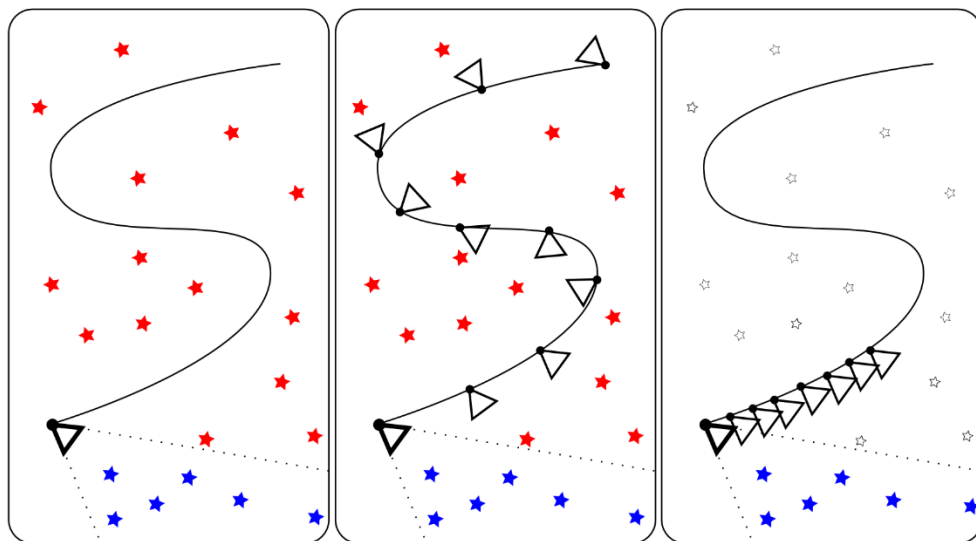


图 2-1 视觉里程计示意图

几种算法的示意图如图 2-1 所示。红色表示旧的特征点，蓝色表示最新的特征点，三角形表示相机状态。左边是 EKF-SLAM 方法，地图中保留了旧的特征点以及最新的相机姿态。中间是关键帧方法，保留了稀疏的相机状态和相应的特征点观测。右边是滑动窗方法，维持一定大小的滑动窗，窗中包括相机状态和特征点观测。

2.6 标识

本论文使用如下的表示方法：

用小写字母 $\mathbf{v}$ 表示向量；用大写字母 $\mathbf{R}$ 表示矩阵；用字母上带点 $\dot{\mathbf{v}}$ 表示导数；

用字母上带帽子 $\hat{\mathbf{v}}$ 表示估计的量；用字母上带横线 $\bar{\mathbf{v}}$ 表示单位向量或是单位四元数；用字母上带波浪线 $\tilde{\mathbf{v}}$ 表示误差的量；

用小写字母 $\mathbf{b}$ 表示坐标系；用 $\mathbf{v}^e$ 表示向量在 $\mathbf{e}$ 坐标系中；用 $\mathbf{R}_b^e$ 表示从体坐标系 $\mathbf{e}$ 到世界坐标系 $\mathbf{b}$ 的旋转矩阵；

用 $\mathbf{R}_b^e$ 表示从体坐标系 $\mathbf{e}$ 到世界坐标系 $\mathbf{b}$ 的旋转四元数。

第 3 章 卡尔曼滤波简介

3.1 卡尔曼滤波

卡尔曼滤波器是一个线性系统的最优估计。假设状态符合线性估计。可以通过给定的当前状态推测下一个状态的概率密度分布，如图 3-1。对于状态 X 的变化，如果一个测量是独立于当前状态的，我们也可以使用该信息来修正与概率密度相关的信息。

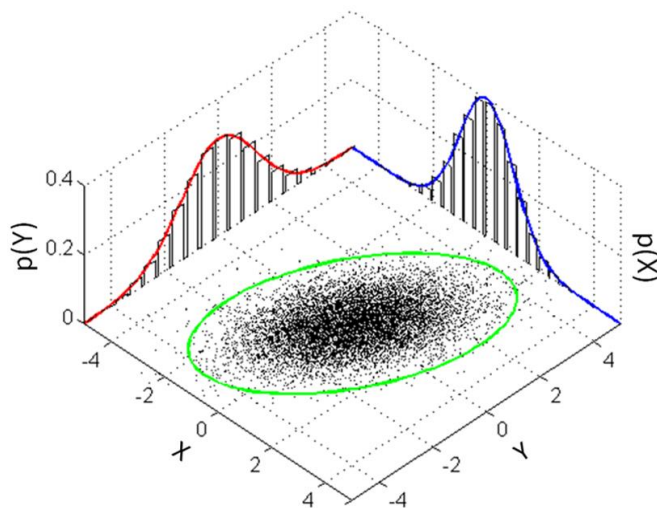


图 3-1 概率分布示意图

对系统的状态和观测模型进行构建，如式(3-1)所示：

$$\begin{aligned} X_k &= \phi_{k,k-1} X_{k-1} + \Gamma_{k-1} W_{k-1} \\ Z_k &= H_k X_k + V_k \end{aligned} \quad (3-1)$$

X_k 是系统的 n 维的状态向量， Z_k 是系统的 m 维的状态向量； W_{k-1} 是 p 维的系统过程噪声向量； V_k 是 m 维观测噪声向量序列， W_k 是系统的 $n \times n$ 维的状态观测矩阵， Γ_{k-1} 是 $n \times p$ 维噪声输入矩阵， H_k 是 $m \times n$ 维观测矩阵。

状态预测：

$$X_{k,k-1} = \varphi_{k,k-1} \hat{X}_{k-1} \quad (3-2)$$

状态修正：

$$\hat{X}_k = \hat{X}_{k,k-1} + K_k (Z_k - H_k X_{k,k-1}) \quad (3-3)$$

卡尔曼增益矩阵：

$$K_k = P_{k,k-1} H_k^T (H_k P_{k,k-1} H_k^T + R_k)^{-1} \quad (3-4)$$

误差方差矩阵预测：

$$P_{k,k-1} = \varphi_{k,k-1} P_{k,k-1} \varphi_{k,k-1}^T + \Gamma_{k,k-1} Q_{k,k-1} \Gamma_{k,k-1}^T \quad (3-5)$$

估计方差矩阵：

$$P_k = (I - K_k H_k) P_{k,k-1} \quad (3-6)$$

式(3-1)至(3-6)构成了基本的随机线性卡尔曼滤波方程[21]。由方程可知，在卡尔曼滤波器执行时，系统采用递推运算实现状态和观测的更新，由前一时刻的结果递推得到当前时刻的结果。在滤波器开始运算时，需要对 $\hat{X}_0$ 和 P_0 进行初始化。在之后 k 时刻的运算中，通过 $k-1$ 时刻的 $\hat{X}_{k-1}$ 和 P_{k-1} ，结合 k 时刻的观测值 Z_k ，实现 $\hat{X}_k$ 和 P_k 求取。同时，通过递推的方式进行滤波器的更新，可以有效的减小对控制器计算性能和数据存储量的要求，使卡尔曼滤波能够在运算能力和存储能力有限的微控制器上实现。

3.2 扩展卡尔曼滤波器

在系统模型为线性情况的条件下，可以采用随机线性离散系统的卡尔曼滤波器作为数据融合方案。在工程应用中很多系统都为非线性模型，例如飞行器、船舶、水下航行器的导航系统，卫星导航系统，工业上的控制系统等。针对传统的卡尔曼滤波器在模型上存在的限制，可以对系统模型中存在变量和过程惯性非线性的部分

进行线性化，应用卡尔曼滤波器进行数据融合，这种改进的滤波方式称为扩展卡尔曼滤波器[22]。

对于系统中存在非线性关系的状态和观测方程，可以求取偏导数，进行了低阶的泰勒级数展开，实现非线性数学关系的线性化。应用卡尔曼滤波器计算经过线性化后的系统的估计。

系统的状态、观测方程如式(3-7)所示：

$$\begin{aligned} X_k &= f(X_{k-1}, U_{k-1}, W_{k-1}) \\ Z_k &= h(X_k, V_k) \end{aligned} \quad (3-7)$$

式中， W_{k-1} 为状态过程噪声， V_k 为观测过程噪声；状态方程 $X_k = f(X_{k-1}, U_{k-1}, W_{k-1})$ 为非线性函数，将 $k-1$ 时刻的 X_{k-1} 映射至当前的时刻 k 的状态 X_k ；观测方程 $Z_k = h(X_k, V_k)$ 为非线性函数，表示了当前的时刻 k 的状态序列 X_k 与观测序列 Z_k 之间的关系。

对系统的状态和观测方程进行线性化，如式(3-8)所示：

$$\begin{aligned} X_k &= \tilde{X}_k + \varphi(X_{k-1} - \hat{X}_{k-1}) + W_{k-1} \\ Z_k &= \tilde{Z}_k + H_k(X_k - \tilde{X}_k) + V_k \end{aligned} \quad (3-8)$$

式中， $\tilde{X}_k$ 和 $\tilde{Z}_k$ 分别为系统状态和观测变量的预测值， X_k 和 Z_k 分别为系统状态和观测变量的预测值， $\hat{X}_{k-1}$ 为系统 $k-1$ 时刻的状态变量的后验估计， W_{k-1} 和 V_k 分别为系统的状态和观测噪声， φ 、 H 、 W 、 V 为对每一项求取偏导数，获得雅克比矩阵，如式(3-9)所示：

$$\begin{aligned} \varphi_{[i,j]} &= \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{X}_{k-1}, U_{k-1}, 0), H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{X}_k, 0) \\ W_{[i,j]} &= \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{X}_{k-1}, U_{k-1}, 0), V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{X}_k, 0) \end{aligned} \quad (3-9)$$

令状态误差为 $\tilde{e}_{xk}$ ，观测误差为 $\tilde{e}_{zk}$ ，误差表达式如式(3-10)所示：

$$\begin{aligned}\tilde{e}_{xk} &\approx \varphi(X_{k-1} - \hat{X}_{k-1}) + \varepsilon_k \\ \tilde{e}_{zk} &\approx H\tilde{e}_{xk} + \eta_k\end{aligned}\tag{3-10}$$

式(3-10)具有线性关系，可通过 $\tilde{e}_{zk}$ 去估计 $\tilde{e}_{xk}$ 。估计得到的结果为 $\hat{e}_k$ ， $\hat{X}_k$ 与 $\hat{e}_k$ 的关系如式(3-11)所示：

$$\hat{X}_k = \tilde{X}_k + \hat{e}_k\tag{3-11}$$

$\hat{e}_k$ 与 $\tilde{e}_{zk}$ 的关系如式(3-12)所示：

$$\hat{e}_k = K_k \tilde{e}_{zk}\tag{3-12}$$

将式(3-12)代入式(3-11)，可得 EKF 的观测更新方程：

$$\hat{X}_k = \tilde{X}_k + K_k \tilde{e}_{zk} = \tilde{X}_k + K_k (Z_k - \tilde{Z}_k)\tag{3-13}$$

综上所述，可以得到 EKF 的方程组，如式(3-14)所示：

$$\begin{aligned}X_k &= f(\hat{X}_{k-1}, k-1) \\ \hat{X}_k &= \hat{X}_{k-1} + K_k (Z_k - h(\hat{X}_{k-1}, k)) \\ K_k &= P_{k,k-1} H_k^T [H_k P_{k,k-1} H_k^T + R_k]^{-1} \\ P_{k,k-1} &= \varphi_{k,k-1} P_{k-1} \varphi_{k,k-1}^T + Q_{k-1} \\ P_k &= [I - K_k H_k] P_{k,k-1}\end{aligned}\tag{3-14}$$

EKF 的初值和误差协方差矩阵的初值如式(3-15)所示：

$$\begin{aligned}\hat{X}_0 &= E[X_0] \\ P_0 &= E[(X_0 - \hat{X}_0)(X_0 - \hat{X}_0)^T]\end{aligned}\tag{3-15}$$

第 4 章 惯性导航基础

惯性导航是利用惯性测量的设备，在平台内部感测平台运动的加速度，以及平台本身的旋转角速度，以牛顿力学为基础进行积分运算，以获取载体的速度和位置等导航信息的技术。由于惯性导航不依赖外部的辐射信息与外部的能量介入，故是一个可以孤立工作的自主导航系统。特别的，将惯性测量的设备直接装在运动平台上，用计算机实现姿态结算的做法，被称为捷联惯导。

惯性测量的设备被称为惯性测量单元（IMU），通常表示由陀螺仪和加速度计组成的设备。随着现代芯片工业的发展，微机电（MEMS）传感器的产量十分巨大，生产成本降低，而且 MEMS 传感器的尺寸变得越来越小，其中能够集成的功能越来越多，已经普遍应用在智能手机、虚拟现实设备、小型机器人和小型无人机等等。

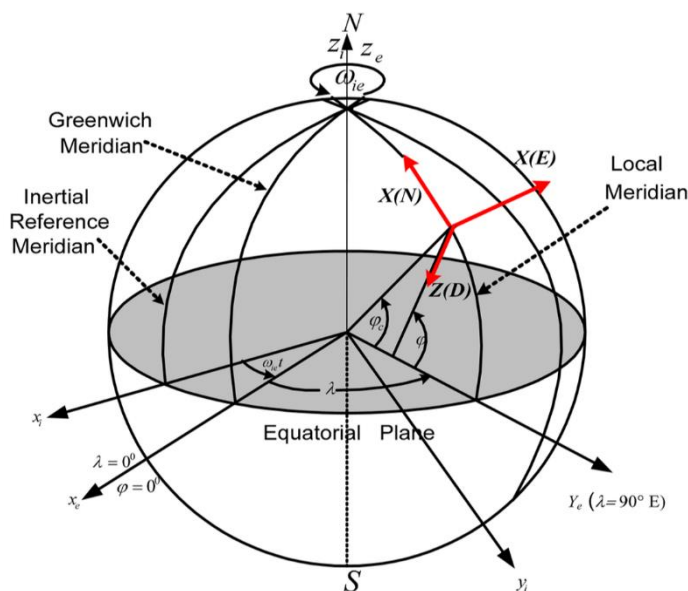
4.1 基础规定

4.1.1 坐标系规定

4.1.1.1 地球坐标系 e :

定义坐标系 e 为导航坐标系，或地球坐标系（earth frame）。本文使用的导航坐标系为标准的北东地坐标系（NED），即坐标系的 x 轴指向地理北方， y 轴指向地理东方， z 轴沿重力指向地心， x 轴与 y 轴所确定的平面与地球相切，。

在不考虑由于不同纬度地区的重力受地球自转向心力的影响而偏转，同时假定移动平台的移动距离相于地球半径很小，满足平板地球假设，即随着移动平台的移动，三个坐标轴的方向始终保持一致， x 轴与 y 轴所确定的平面唯一，地球表面为不随移动而改变的平面。



在导航坐标系中的点 P^e 表达为: $P^e = [x^e \quad y^e \quad z^e]^T$ 。

定义坐标系 **b** 为机体坐标系(**body frame**)。机体坐标系坐标原点通常定义为移动平台的质心，但是为了避免科里奥利力对加速度计测量的影响，也可定义机体坐标系坐标原点为加速度计的测量中心，即加速度计的坐标系原点。机体坐标系的 **x** 轴由原点指向机头方向，当机头指北时，机体坐标系 **b** 的各坐标轴与导航坐标系 **e** 的各坐标轴方向一致，即 **x** 轴向前方，**y** 轴向右方，**z** 轴指向下方，如图 4-2 所示。在解算过程中，陀螺仪和加速度计的各个坐标轴的方向应该与机体坐标系的坐标轴指向相同。

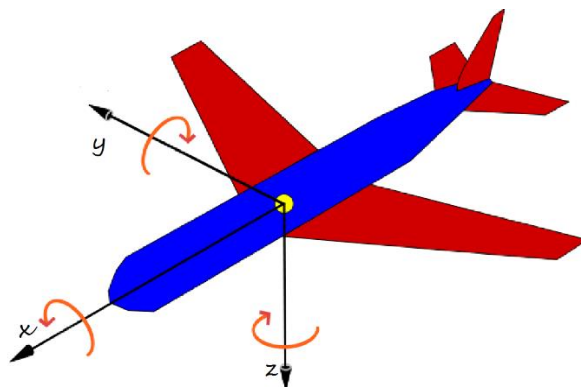


图 4-2 机体坐标系示意图

在机体坐标系中的点 P^b 表达为： $P^b = [x^b \ y^b \ z^b]^T$ 。

4.1.2 位姿表达

本文所讨论的视觉惯性里程计是用以确定移动平台的移动位置和姿态的，故需要确立唯一的位置和姿态的表达方式。

4.1.2.1 位置表征

由于本文研究的里程计应用条件满足平板地球假设，故导航坐标系为可以唯一确立的固定三维笛卡尔坐标系。在这个坐标系下，仅仅使用一个三维的平移向量即可表达里程计中点的位置。即向量 $v^e = [x^e \ y^e \ z^e]^T$ 即可表示 P_e 点的位置。

4.1.2.2 旋转表征

旋转的表征通常使用欧拉角，方向余弦矩阵和四元数。

4.1.2.2.1 欧拉角：

欧拉角是表述旋转最为直观的方式。要表示三维空间中刚体的旋转，最少需要三个自由度。可以用从参考系按照一定的旋转次序，依次绕三个坐标轴旋转的三个角度来表达。通常的 z-y-x 顺序旋转中，规定绕参考坐标系 x 轴旋转的角为滚转角（roll），绕 y 轴旋转为俯仰角（pitch），绕 z 轴旋转为偏航角（yaw）。如图 4-3 所示。

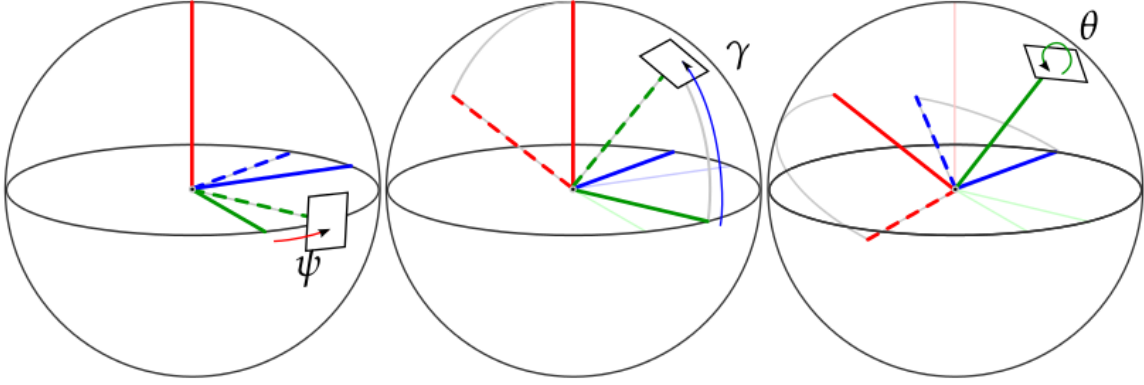


图 4-3 欧拉角表示姿态

对于一个旋转，使用 $[\theta \ \gamma \ \psi]^T$ 进行表达。

由于旋转次序不唯一，对于任意一个旋转，欧拉角可以有二十四种方式进行表达。同时，由于欧拉角再表达旋转时有奇点的影响，存在万向节锁(*gimbal lock*)，故不使用欧拉角作为迭代姿态的表示方法。

4.1.2.2.2 方向余弦矩阵：

方向余弦矩阵即为旋转矩阵。方向余弦矩阵是 $SO(3)$ 的元素，对于任意一个旋转，方向余弦矩阵是唯一的一一对应的表达方式。同时，方向余弦矩阵具有一些特性，详见附录 A $SO(3)$ 基础。对于图 4-3 所示的三次旋转，使用方向余弦矩阵分别表达如下：

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad \mathbf{R}_y = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \quad \mathbf{R}_z = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-1)$$

综合的 z-y-x 顺序旋转的旋转矩阵表达为：

$$\mathbf{R}_{xyz} = \mathbf{R}_{x'}\mathbf{R}_{y'}\mathbf{R}_z \quad (4-2)$$

方向余弦矩阵是在三维空间中表达旋转时最准确的方式，但其具有九个元素，在迭代计算时开销较大，所以通常不使用方向余弦矩阵进行迭代计算。

4.1.2.2.3 四元数：

四元数使用四个维度表示旋转，奇点的影响。对于一个旋转，可以将其转化为轴角的形式，并用四元数表示。将图 4-3 的旋转转化为轴角的形式，如图 4-4 所示。

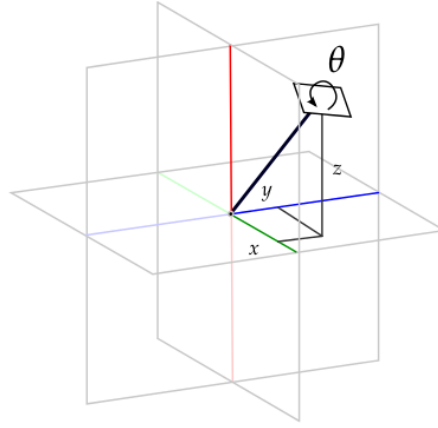


图 4-4 四元数姿态表示

在轴角形式的表达，其中三个数值是表示旋转在三维空间的轴线的单位向量，而第四个数值表示绕该轴线的旋转，最终如下表示四元数：

$$\bar{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{k}} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} = \begin{bmatrix} k_x \sin(\theta/2) \\ k_y \sin(\theta/2) \\ k_z \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \quad (4-3)$$

由于表示轴角的向量是单位向量，故满足上式的旋转四元数为单位四元数。四元数的代数计算与常规的向量计算不同，具体可以参见附录 B 四元数。由于本文中，算法迭代不使用欧拉角和方向余弦矩阵，而是使用的四元数，仅在最终的结果表达时使用欧拉角作为表示。

4.2 惯性测量单元模型

4.2.1 加速度计模型

加速度计是用来测量线性加速度的传感器。测量单位为 g 或是 m/s^2 。

刚体相对于地球的静止状态时，线性加速度为 0 。此时，由于受到重力影响，使用加速度计测量刚体的线性加速度时，在竖直向下指向地心的轴线上会有读数为 g ，约为 $9.81m/s^2$ 。同理，在重力的作用下，当刚体上的加速度计的读数为 0 时，刚体处于自由落体状态，实际上有竖直向下 g 的实际加速度。

MEMS 的加速度计的内部为弹簧-质量块的微结构，即在微弹簧-质量块的形变轴线上有加速度时，会使得微弹簧产生形变。使用微电子的方式测量微弹簧的形变，即可取得测量出轴线上的加速度。由于这样的结构，**MEMS** 加速度计不能测量刚体的实际加速度，只能给出沿其测量轴的加速度测量值。在实际使用中，通常使用三套 **MEMS** 测量系统，共同组成正交的三轴测量系统，分别测量实际加速度在三个正交测量轴上的加速度分量。通过三个正交测量轴上的加速度分量，可以解算出实际的加速度。当三轴加速度计的测量轴与机体坐标系的坐标轴相重合时，三轴加速度计的测量值与实际加速度的关系如下：

$$a_m^b = R_e^b(a^e + g^e + 2[\omega^e \times]v^e + [\omega^e \times]^2 p^e) \quad (4-4)$$

其中， a_m^b 为加速度计实际测量值，底标 m 表示是测量值，为机体坐标系下的表达； R_e^b 为机体坐标系与导航坐标系的旋转矩阵，用于转换导航坐标系到机体坐标系； a^e 为刚体的实际运动加速度，为导航坐标系下的表达； g^e 为重力加速度，在导航坐标系下仅有 z 轴分量，可以表达为 $g^e = [0 \ 0 \ 1]^T$ ， ω^e 为地理系下的角速度， v^e 为地理系下的运动速度， p^e 为重心偏置在地理系下的表达。

假设刚体的重心在加速度计的坐标系原点，即可不考虑由于运动带来的科里奥利力等加速度测量值，如下式所示：

$$a_m^b = R_e^b(a^e + g^e) \quad (4-5)$$

由于 MEMS 传感器的高噪声特性，故其本身的测量值应该表达为：

$$a_m^b = R_e^b(a^e - g^e) + b_a + n_a \quad (4-6)$$

其中， b_a 为加速度计的零点偏移，通常为时变的随机游走噪声，其本身的噪声可以描述为零均值的正态分布的变量 $n_{wa} \sim \mathcal{N}(0, N_{wa})$ ； n_a 为加速度计的噪声，对于噪声较大的 MEMS 传感器， n_a 可直接表达为零均值的正态分布噪声 $n_a \sim \mathcal{N}(0, N_a)$ 。

对其进行进一步扩展，写为

$$a_m^b = R_{axis} T_{scale} R_e^b(a^e - g^e) + b_a + n_a \quad (4-7)$$

式中， R_{axis} 为表示正交测量轴旋转误差的旋转矩阵，用以表达三轴加速度计三个测量轴的正交情况； T_{scale} 表示三轴加速度计各个坐标轴的比例误差，用以表达三轴加速度计三个测量轴测量值与真实值的放大比例。理想情况下，三轴加速度计三个测量轴两两正交，且测量值能够真实反应真实值，无比例放大情况，则：

$$R_{axis} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T_{scale} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

实际上，考虑 MEMS 陀螺仪的制造误差，三轴加速度计三个测量轴的正交情况不一定完美，且由于各种原因三个轴存在比例放大，则实际误差矩阵有：

$$R_{axis} \neq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T_{scale} = \begin{bmatrix} scale_x & 0 & 0 \\ 0 & scale_y & 0 \\ 0 & 0 & scale_z \end{bmatrix} \quad (4-8)$$

对于每个芯片， R_{axis} 的值不变，仅仅是芯片个体不同而不同。 T_{scale} 会随着静电和温度等等因素发生微小的改变。写为 $R_{axis} T_{scale} = T_a$ ，仅需要确定 3×3 的误差增益矩阵 T_a 即可进行制造误差的校正。最终确定的加速度计模型为：

$$a_m^b = T_a R_e^b (a^e - g^e) + b_a + n_a \quad (4-9)$$

可在标定过程或自标定过程中确定加速度计的 T_a 。如果使用环境变化不大，可以一次性标定 T_a ，如果在大动态的恶劣环境中使用，应当通过自标定功能的算法动态估计 T_a 。

4.2.2 陀螺仪模型

陀螺仪是用来测量刚体旋转角速度的传感器。测量单位为 rad/s 。

与 MEMS 加速度计类似的，MEMS 陀螺仪也只能测量绕单个测量轴旋转的角速度分量，故使用时也是集成封装为具有三个正交测量轴的三轴陀螺仪，分别测量刚体旋转角速度在三个测量轴上的旋转分量，最终合成刚体的实际旋转角速度。

与 MEMS 加速度计类似的，MEMS 陀螺仪的数学模型可表达为：

$$\omega_m^b = \omega^b + b_g + n_g \quad (4-10)$$

其中， ${}^m\omega_b$ 为陀螺仪的实际测量值，为机体坐标系下的表达； ω_b 为刚体的实际旋转角速度，为机体坐标系下的表达； b_g 为陀螺仪的零点偏移，为时变的随机游走噪声，其本身的噪声可以描述为零均值的正态分布的变量 $n_{wg} \sim \mathcal{N}(0, N_{wg})$ ； n_g 为加速度计的噪声，对于噪声较大的 MEMS 传感器， n_g 可直接表达为零均值的正态分布噪声 $n_g \sim \mathcal{N}(0, N_g)$ 。

对于陀螺仪，特殊的，有加性的加速度敏感误差。即各个测量轴的测量误差中，有一项加性的误差，仅与当前测量轴上的线性加速度相关。事实上，这个误差与加速度是非线性关系，对于一般噪声较大的 MEMS 陀螺仪，由于可以认为加速度敏感误差的高阶分量已经淹没在噪声当中，故支取其与加速度相关的一阶分量，将陀螺仪模型改写为：

$$\omega_m^b = \omega^b + T_s a^b + b_g + n_g \quad (4-11)$$

式中， $T_s a^b$ 即为陀螺仪加速度敏感误差项，表达为加速度相关的一阶形式， a^b 即为作用在各个测量轴的线性加速度， T_s 为加速度敏感系数矩阵。

同样的考虑陀螺仪制造误差与放大误差，并且忽略加速度计敏感性，将陀螺仪表模型该写为：

$$\omega_m^b = T_g \omega^b + b_g + n_g \quad (4-12)$$

此式中， $T_g = R_{axis_g} T_{scale_g}$ ， R_{axis_g} 表示衡量三轴陀螺仪的三个测量轴正交程度的定常的旋转误差， T_{scale_g} 表示三轴陀螺仪的比例放大误差，其定义与表现与加速度计中的定义类似，故可写为一个 3×3 的误差矩阵 T_g 。

4.3 惯性更新过程

本节将分别叙述通过惯性测量单元的陀螺仪和加速度计更新姿态，速度和位移的过程。

4.3.1 旋转姿态的更新

由于在惯性更新的开始过程中，无法进行绝对姿态的初始化，故从系统开始工作的时候为姿态零起点进行初始化。

首先根据陀螺仪的测量模型，将陀螺仪的测量值换算为载体旋转角速度的估计值：

$$\hat{\omega}^b(t) = \hat{T}_g^{-1} (\omega_m^b(t) - \hat{b}_g(t)) \quad (4-13)$$

由于使用姿态四元数表达姿态，所以不能使用三轴的角速度进行直接线性积分。更新过程的四元数微分方程在附录 B 中进行了说明。在连续旋转过程中，有：

$$\dot{R}_b^e = R_b^e \hat{\omega}^b \quad (4-14)$$

将其按照四元数进行连续旋转展开，有：

$$\bar{q}_{b(t+\Delta t)}^e = \bar{q}_{b(t)}^e \otimes \bar{q}_{b(t+\Delta t)}^{b(t)} \quad (4-15)$$

那么，使用四元数微分方程表达连续的小旋转：

$$\dot{\bar{q}}_{b(t)}^{b(l)} = \frac{1}{2} \Omega(\hat{\omega}^b) \bar{q}_{b(t)}^{b(l)} \quad t \in [t_l, t_{l+1}] \quad (4-16)$$

四元数更新是一个非线性的过程，同时传感器的读数是一个离散时间的采样过程。在离散采样的非线性更新过程中，为了保证四元数姿态的连续和光滑，同时在一定程度上节约计算量，在两次采样之间的 Δt 时间内，使用四阶龙格-库塔方法更新姿态。对陀螺仪的测量值进行简要改写： $\hat{\omega}^b(t_l) = \hat{\omega}_l$ ，将更新过程写为：

$$\bar{q}_0 = [0 \quad 0 \quad 0 \quad 1]^T \quad (4-17)$$

$$\hat{q}_{b(t+\Delta t)}^{b(t)} = \bar{q}_0 + \frac{\Delta t}{6} (k_1 + 2k_2 + 2k_3 + k_4) \quad (4-18)$$

其中， k_1 ， k_2 ， k_3 与 k_4 按照如下的表达式进行定义：

$$\begin{aligned} k_1 &= \frac{1}{2} \Omega(\hat{\omega}_l) \bar{q}_0 \\ k_2 &= \frac{1}{2} \Omega\left(\frac{\hat{\omega}_l + \hat{\omega}_{l+1}}{2}\right) \left(\bar{q}_0 + \frac{\Delta t}{2} k_1\right) \\ k_3 &= \frac{1}{2} \Omega\left(\frac{\hat{\omega}_l + \hat{\omega}_{l+1}}{2}\right) \left(\bar{q}_0 + \frac{\Delta t}{2} k_2\right) \\ k_4 &= \frac{1}{2} \Omega(\hat{\omega}_{l+1}) (\bar{q}_0 + \Delta t k_3) \end{aligned} \quad (4-19)$$

四阶龙格-库塔更新是通过对单位四元数的加和进行更新，会导致更新后增量四元数为非单位四元数，所以需要进行四元数的归一化：

$$\hat{q}_{b(t+\Delta t)}^{b(t)} = \frac{\hat{q}_{b(t+\Delta t)}^{b(t)}}{\|\hat{q}_{b(t+\Delta t)}^{b(t)}\|} \quad (4-20)$$

4.3.2 速度更新

与陀螺仪相同的方式，首先根据加速度计的测量模型，将加速度计的测量值换算为载体移动的加速度估计值：

$$\begin{aligned}\hat{a}^b(t) &= T_a^{-1}(a_m^b(t) - \hat{b}_a) \\ \hat{a}^e(t) &= \hat{R}_b^e(t)\hat{a}^b(t) - g^e\end{aligned}\quad (4-21)$$

使用加速度计的估计值进行迭代，计算速度更新。

$$\begin{aligned}\hat{v}^e(t_{l+1}) &= \hat{v}^e(t_l) + \int_{t_l}^{t_{l+1}} \hat{a}^e(\tau) d\tau \\ &= \hat{v}^e(t_l) + \int_{t_l}^{t_{l+1}} (\hat{R}_{b(\tau)}^e \hat{a}^b(\tau) - g^e) d\tau \\ &= \hat{v}^e(t_l) + \int_{t_l}^{t_{l+1}} \hat{R}_{b(\tau)}^e \hat{a}^b(\tau) d\tau - g^e \Delta t \\ &= \hat{v}^e(t_l) + \hat{R}_{b(t_l)}^e \int_{t_l}^{t_{l+1}} \hat{R}_{b(\tau)}^{b(t_l)} \hat{a}^b(\tau) d\tau - g^e \Delta t \\ &= \hat{v}^e(t_l) + \hat{R}_{b(t_l)}^e \hat{s}_l - g^e \Delta t\end{aligned}\quad (4-22)$$

其中， $\Delta t = t_{l+1} - t_l$ 。 $\hat{s}_l$ 定义为：

$$\hat{s}_l = \int_{t_l}^{t_{l+1}} \hat{R}_{b(\tau)}^{b(t_l)} \hat{a}^b(\tau) d\tau \quad (4-23)$$

$\hat{s}_l$ 为加速度关于时间的连续积分。由于速度变化为线性变化，故连续的积分过程可以使用离散的梯形积分进行代替。此处的梯形积分定义如下：

$$\hat{s}_l \approx \frac{\Delta t}{2} (\hat{R}_{b(t_{l+1})}^{b(t_l)} T_a^{-1}(a_m^b(t_{l+1}) - \hat{b}_a) + T_a^{-1}(a_m^b(t_l) - \hat{b}_a)) \quad (4-24)$$

4.3.3 位移更新

在速度更新的基础上进行位移更新。根据加速度计的测量模型，将加速度计的测量值换算为载体移动的加速度估计值，再用加速度估计值转化为速度估计值，进而推导位移估计值。

$$\begin{aligned}\hat{p}^e(t_{l+1}) &= \hat{p}^e(t_l) + \int_{t_l}^{t_{l+1}} \hat{v}^e(\tau) d\tau \\ &= \hat{p}^e(t_l) + \int_{t_l}^{t_{l+1}} (\hat{v}^e(t_l) + \int_{t_l}^{\tau} \hat{a}^e(\zeta) d\zeta) d\tau \\ &= \hat{p}^e(t_l) + \hat{v}^e(t_l) \Delta t + \hat{R}_{b(t_l)}^e \int_{t_l}^{t_{l+1}} \int_{t_l}^{\tau} \hat{R}_{b(\zeta)}^{b(t_l)} \hat{a}^b(\zeta) d\zeta d\tau - \frac{1}{2} g^e \Delta t^2 \\ &= \hat{p}^e(t_l) + \hat{v}^e(t_l) \Delta t + \hat{R}_{b(t_l)}^e \hat{y}_l - \frac{1}{2} g^e \Delta t^2\end{aligned}\quad (4-25)$$

$\hat{y}_l$ 也为加速度关于时间的连续积分。与 $\hat{s}_l$ 类似，在离散采样下使用梯形积分进行更新，定义如下：

$$\hat{y}_l = \int_{t_l}^{t_{l+1}} \int_{t_l}^{\tau} \hat{R}_{b(t_\zeta)}^{b(t_l)} \hat{a}^b(\zeta) d\zeta d\tau \quad (4-26)$$

$$\hat{y}_l \simeq \frac{\Delta t}{2} \hat{s}_l \quad (4-27)$$

第 5 章 视觉基础

随着计算机科学和机器人学的进步，计算机视觉开始迅速发展。像人眼一样，视觉算法开始赋予摄像头识别，检测，跟踪三维世界中的物体等能力。同时，摄像头器件成本低廉，体积小，可视范围广，比激光雷达有不可比拟的优势，这也是为什么机器人领域越来越多选择摄像头作为里程计的重要传感器。本章我们会介绍通用的相机模型，以及三维空间与二维图像之间的映射关系等。

5.1 视觉系统基础规定

5.1.1 坐标系规定

5.1.1.1 图像坐标系 i :

定义坐标系 i 为图像坐标系。图像坐标系用以表示像素点在图像上的位置，故图像坐标系的坐标轴量纲为像素。由于图像为二维信息，故图像坐标系仅有两个坐标轴。按照开源视觉库 **OpenCV** 对图片坐标的规定，以图片的左上角为图像坐标系的原点，沿图像宽度方向向右为 u 坐标轴，沿图像高度方向向下为 v 坐标轴，如图 5-1 所示。

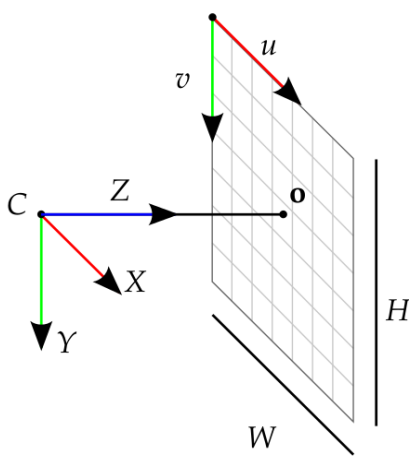


图 5-1 相机坐标系和图像坐标系

5.1.1.2 相机坐标系 C:

定义坐标系 C 为相机标系。相机坐标系的原点为相机光学系统的光心，x 轴与 y 轴分别与图像坐标系的 u 轴与 v 轴互相平行，z 轴由原点指向物体，相机方向，表示景深。如图 5-1 所示。

5.2 针孔摄像模型

针孔相机是最常用的相机模型，如图 5-2。和小孔成像过程类似，把三维世界中的实体映射到像平面上。由于是三维到二维，因此会失去一个维度的信息。即在沿相机焦点射线方向上的所有的点会映射到图像中的同个点。针孔相机只是最简单的模型，实际应用中由于透镜镜头的特点，成像会有各种各样的畸变，可能会导致图像变形失真。

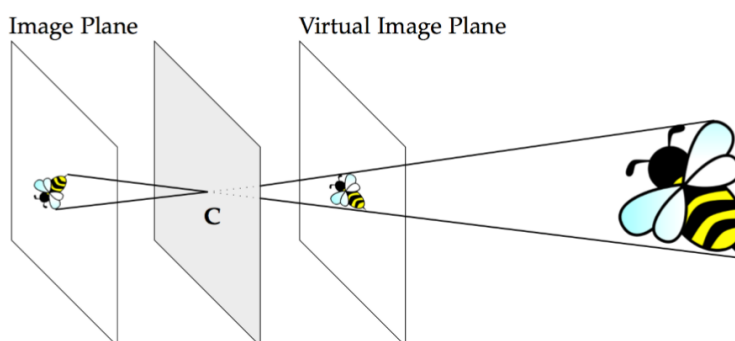


图 5-2 针孔摄像机示意图

5.3 摄像机观测模型

按照前文所述的坐标系，图像坐标 $x^i = [u \ v]^T$ 与相机系的坐标 $x^c = [X \ Y \ Z]^T$ 有如下关系：

$$\begin{aligned} u &= f \frac{X}{Z} \\ v &= f \frac{Y}{Z} \end{aligned} \quad (5-1)$$

f 为相机焦距。

由于图像坐标系和相机坐标系的原点不同，则：

$$\begin{aligned} u &= f \frac{x}{z} + o_x \\ v &= f \frac{y}{z} + o_y \end{aligned} \quad (5-2)$$

此时， f 的单位为像素。关系写为测量方程为：

$$h \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} o_x \\ o_y \end{bmatrix} + \begin{bmatrix} f & 0 \\ 0 & f \end{bmatrix} \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \end{bmatrix} \quad (5-3)$$

由于存在各种镜头的畸变，所以真实的相机要比针孔模型复杂的多，如图 5-3 所示，左边是无畸变的模型，中间是径向畸变，右边是切向畸变。

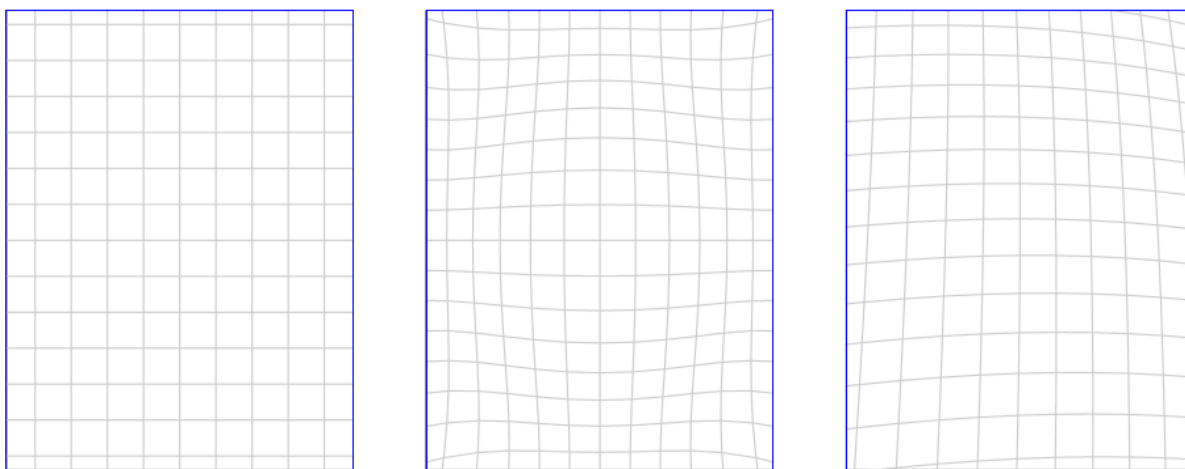


图 5-3 畸变示意图

在引入各种非线性的畸变修正之后，就形成复杂的非线性成像模型。镜头的畸变主要分为径向畸变、切向畸变等。径向畸变来自于透镜的形状，即常说的“鱼眼效应”，在图像中心畸变为 0，越到边缘畸变越大。切向畸变来自于摄像机的组装

过程，由于制造工艺的限制，使得透镜本身产生缺陷与图像的平面不平行。考虑着两个效应之后，相机的投射函数可以改为如下表达：

$$h \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} o_x \\ o_y \end{bmatrix} + \begin{bmatrix} f & 0 \\ 0 & f \end{bmatrix} (d_r \begin{bmatrix} u \\ v \end{bmatrix} + d_t) \quad (5-4)$$

$$d_r = (1 + k_1 r + k_2 r^2 + k_3 r^3) \quad (5-5)$$

$$d_t = \begin{bmatrix} 2uv t_1 + (r + 2u^2) t_2 \\ 2uv t_2 + (r + 2v^2) t_1 \end{bmatrix} \quad (5-6)$$

此处， $r = u^2 + v^2$ 。

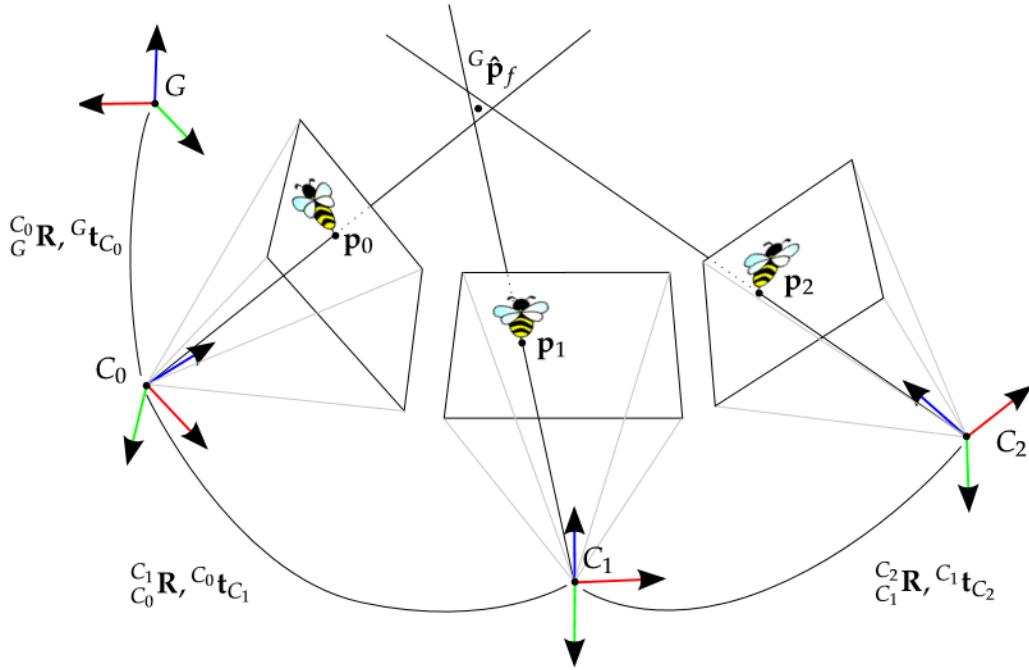


图 5-4 连续测量示意图

实际情况中我们感兴趣的通常是点在三维空间的世界坐标而不是相机坐标系中的坐标。因此通常知道点在像相机坐标系中的坐标到像素坐标的映射关系之后，还

需要知道相机在全局坐标系中的平移和旋转。规定一个三维空间中的点 $\mathbf{x}$ ，在世界坐标系中的坐标可以转换到相机坐标系。空间点的 3D 坐标到像素坐标的转换关系为：

$$\begin{bmatrix} u \\ v \end{bmatrix} = h(R_c^e(\mathbf{x}^e - \mathbf{p}_c^e)) \quad (5-7)$$

其中 $\mathbf{p}_c^e$ 是相机中心在全局坐标系中的坐标。 R_c^e 是相机相对于全局坐标系的旋转。

5.4 特征处理的算法

在前面的叙述中，从针孔摄像机模型展开，讨论三维的场景中的点与二维图像坐标之间的关系，接着讨论了视觉定位中，特征点的测量值与相机位置之间的关系。在本节中，我们需要对“特征点”进行一定的说明，定义特征点这个概念，以及如何使用算法快速和准确的识别特征点，并在连续的图像流中实现对某些关键的特征点进行连续的追踪。

对特征点的整个处理算法流程大致分为三个步骤：

- 特征点的检测与提取；
- 特征点的描述，计算特征向量；
- 通过特征向量进行特征点匹配；

由于本文的关注重点在于通过对特征点测量进行移动平台的定位，所以特征点的表达、生成和匹配方法不在本文的研究范畴之内。本节仅对现有的较流行的特征处理方法进行对比，并选出合适的特征处理方案。

5.4.1 特征提取算法

要在连续的图像流中实现对图像的局部特征进行连续的追踪，首先需要讨论的问题是，什么是局部特征。由于使用的目的和领域不同，图像中的特征没有准确的定义，只要是图像中可以重复使用的特性即可称之为特征，一般有点、边缘、颜

色、纹理或者频谱等等较为模糊的划分，通常通过一定的算法将图像中的特征表达为一个具有一定数据结构的高维特征向量。而将特征从图像中抽取出来，表达为高维特征向量的过程被称为特征提取。

由于本文的摄像机测量定位模型是建立在以点测量为基础上的，所以本文的图像处理过程关心的是图像中的点特征。对于一般定位环境，用“角点”即可较好的描述定位需要的点特征。

角点，顾名思义即为角的顶点，在图像上可以直观地用边缘的交点来描述，在数值上，可以表达为二维图像的极值点，边缘线段的终点和交点，边缘曲线曲率突变或最大的点，或者图像二维两个方向上具有较大梯度的点。特征点的检测方法，主要分为三个大的类别，即基于灰度图的特征点点检测、基于二值图像的特征点检测和基于轮廓的特征点检测。基于灰度图的特征点检测通常计算图像的梯度，通过灰度图像数值一阶导数衡量图像的局部变化情况，找到特征点，称为基于灰度图梯度的特征点检测方法；或者考虑像素领域点的灰度变化，与相邻点进行强度比较，将对比度足够大的点定义为特征点，称为基于灰度图模板的特征点检测方法。

本文选取了三种最为流行的特征点提取算法进行简述与对比，分别是 Harris 特征点检测算法、Shi-Tomasi 改进的 Harris 特征点检测算法和 FAST 特征点检测算法。

Harris 特征点检测算法是最为经典的检测方法，在二维图像的两个维度上取梯度后，梯度较大的点是边缘点，既认为是特征点。算法实现时，在两个方向计算梯度，根据设定的阈值判断是否是满足要求的特征点。

Shi-Tomasi 改进的 Harris 特征点检测算法在 Harris 特征点检测算法的基础上，改善了特征值生成的方法[23]，具有较好的特征点识别效果。

FAST 特征点检测算法将角点定义为在该像素点的周围邻域内有足够多的像素点值与该点处不同的区域 [24]。如图 5-5 所示为 FAST 算法的原理，对于中心点，

一部分周围的点灰度差大于阈值，则确定该点为特征点。在小的区域中选出一系列特征点后，利用非极大值进行抑制，从一系列特征点中取出最佳特征点。**FAST** 算法利用机器学习的方法，通过决策树指定识别方向，进行 2-3 次判定即可确定该点是否为角点。

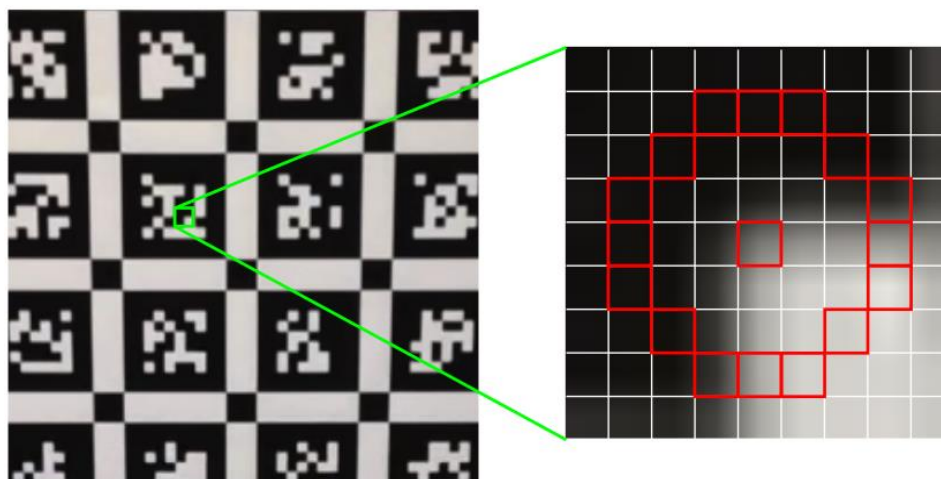


图 5-5 FAST 原理简述

选取使用 VIO 的一般的室内环境照片作为测试样例，选取一定数量的图片，以每张图片提取 200 个特征点作为标准，对特征点提取算法进行特征提取的对比测试。特征提取的结果如 图 5-6 等四张样例图所示，可以，在提取相同数量的特征点的过程中，Harris 算法提取到的特征点较为集中，出现了点簇，而另一些较为明显的特征点却没有被提取出来；Shi-Tomasi 改进的 Harris 算法提取到的特征点较为均匀的分布在图片中，识别的也较为准确；FAST 算法提取到的特征点散布程度介于 Harris 算法与 Shi-Tomasi 改进的 Harris 算法之间，在局部较为均匀。



图 5-6 室内场景图



图 5-7 Harris 算法结果



图 5-8 Shi-Tomasi 改进的 Harris 法结果



图 5-9 FAST 算法结果

表 5-1 特征点提取算法对比

特征点提取方式	Harris	改进的 Harris	FAST
特征点个数	202	200	205
耗时（ms）	14.4721	12.0665	1.23222

在另一方面，从计算开销的角度考虑，Harris 算法与 Shi-Tomasi 改进的 Harris 算法的运行需要的计算开销相似，而 FAST 算法的计算开销却远远小于前两个算法。综合考虑特征点的质量和效率，选择 FAST 方案作为视觉部分的特征的提取算法。

5.4.2 特征匹配算法

由于本文的系统要求做到实时定位，所以特征点描述的核心在于快速准确的描述特征点。

特征点描述算法基本的思路都是按照一定的规律，通过特征点以及其周围的像素，计算出高维的特征向量，用高维的特征向量作为特征点在高维空间的描述。目前比较流行的特征点描述算法有：Lowe D G.提出的 SIFT [25]算法，Bay H 等人提出的 SURF [26]算法，Calonder M 等人提出的 BRIEF [27]算法，Leutenegger S 等人提出的 BRISK [28]算法，Rublee E 等人提出的 ORB [29]算法，Alahi A 等人提出的 FREAK [30]等一系列的算法以及其改进算法。

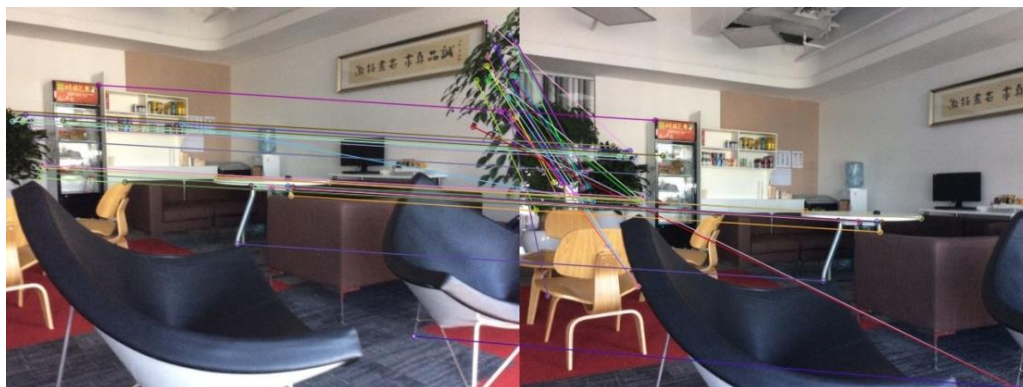


图 5-10 SIFT 匹配结果

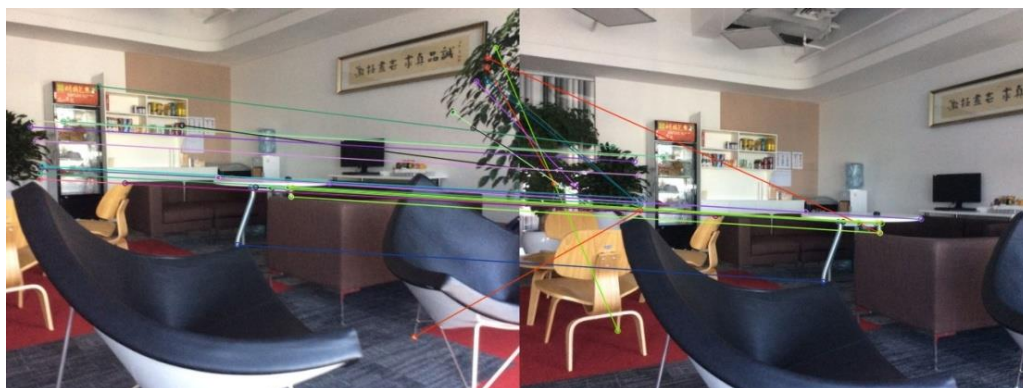


图 5-11 BRIEF 匹配结果



图 5-12 ORB 匹配结果

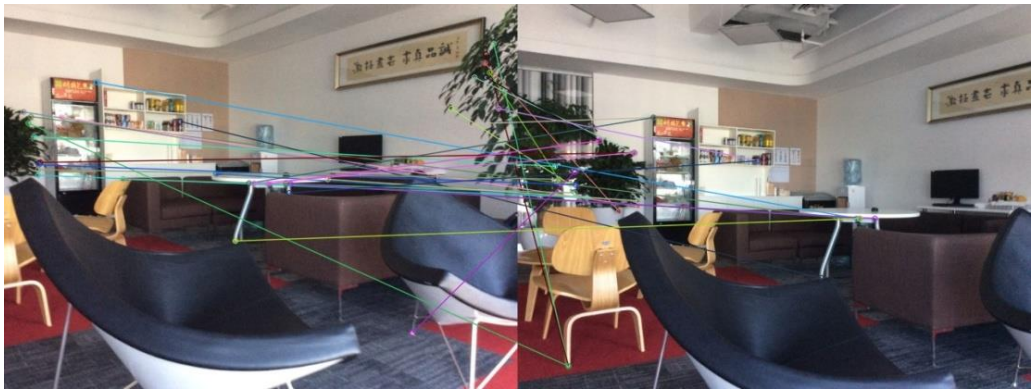


图 5-13 FREAK 匹配结果

选取使用 VIO 的一般的室内环境照片作为测试样例，选取一定数量的图片，以 FAST 算法提取图片中的特征点，对特征点描述算法进行特征描述的对比测试。

分别采取密集特征点和稀疏特征点进行描述来对比算法的性能。分别调节 FAST 算法的阈值，表格样例中两张图片分别有 463 和 450 个特征点，示例图片样例中两张图片分别有 47 和 86 个特征点，然后使用暴力的直接匹配来匹配特征向量。进行特征的描述和匹配后，结果如表 5-2 所示。

表 5-2 匹配算法消耗示例表

匹配算法	SIFT	SURF	BRIEF	BRISK	ORB	FREAK
匹配数	463	463	352	415	333	369

北京理工大学本科生毕业设计

描述耗时	139.42	39.755	6.9095	455.001	8.0527	66.65
匹配耗时	9.2981	6.0034	7.8403	17.535	7.1807	15.335

按照各个算法的消耗与效率来看，最终选择 **ORB** 算法来进行特征点的特征提取与匹配。

第 6 章 MSCKF

MSCKF（多态约束卡尔曼滤波器）可以被认为是 EKF-SLAM 和 SWF 的混合体。MSCKF 的核心思想是保持平台位姿态向量的窗口，并同时更新使用的是整个窗口可见的特征，来优化估计窗口中的每个位姿。通常当一个新的特征被追踪时，当前的状态量被加入到滑动窗口状态中，而当滑动窗口中某一帧的所有特征都丢失时，则将这一帧从滑动窗口中移除。

6.1 状态空间

要维持状态传递，我们定义 IMU 的状态向量为 16 维，如下：

$$x^I = [\bar{q}_b^{eT} \quad p^{eT} \quad v^{eT} \quad b_g^T \quad b_a^T]^T \quad (6-1)$$

状态量的微分方程为：

$$\begin{aligned} \dot{\bar{q}}_{b(t)}^e &= \frac{1}{2} \Omega(\omega^b) \bar{q}_{b(t)}^e \\ \dot{p}^e(t) &= v^e(t) \\ \dot{v}^e(t) &= a^e(t) \\ \dot{b}_g(t) &= n_{wg}(t) \\ \dot{b}_a(t) &= n_{wa}(t) \end{aligned} \quad (6-2)$$

n_{wg} 和 n_{wa} 即为高斯噪声。

按照（6-1）所示，状态的误差状态为 15 维，如下：

$$\tilde{x}^I = [\tilde{\theta}^T \quad \tilde{p}^{eT} \quad \tilde{v}^{eT} \quad \tilde{b}_g^T \quad \tilde{b}_a^T]^T \quad (6-3)$$

旋转的误差 $\tilde{\theta}$ 由以下公式推导：

$$R_b^e \simeq (I_3 - [\tilde{\theta} \times]) \hat{R}_b^e \quad (6-4)$$

历史的状态为：

$$x_i = [\bar{q}_{b_i}^{eT} \quad p_i^{eT} \quad v_i^{eT}]^T \quad (6-5)$$

完整的状态即为 $16 + 10i$ 维，即为：

$$x_l = [\bar{q}_b^{eT} \quad p^{eT} \quad v^{eT} \quad b_g^T \quad b_a^T \quad x_1 \quad \dots \quad x_i]^T \quad (6-6)$$

6.2 误差模型

现在，我们推导出描述了 IMU 状态估计误差是如何传播的过程中进化状态转移矩阵 Φ 。

在 l 时刻我们使用 IMU 状态估计 $\hat{x}_{l|l}$ 和 IMU 测量来计算传播状态估计 $\hat{x}_{l+1|l}$ 。我们的目标是得到 IMU 误差状态转移矩阵 $\Phi_{l|l}$ 使得 $\hat{x}_{l+1|l}$ 满足表达式

$$\hat{x}_{l+1|l} \simeq \Phi_{l|l} \hat{x}_{l|l} + w_l。$$

旋转误差：

$$\hat{R}_{l+1|l} = \hat{R}_{l|l} \hat{R}_{b(l+1)}^{b(l)} \quad (6-7)$$

其中， $\hat{R}_{l|l} = R(\bar{q}_{b(t)}^e)$ ， $\hat{R}_{l+1|l}$ 表示通过 $\hat{R}_{l|l}$ 得到的下一时刻估计。

$$R_{b(l)}^{b(l+1)} \simeq (I_3 - [\tilde{\theta}_{\Delta l} \times]) \hat{R}_{b(l)}^{b(l)} \quad (6-8)$$

另一方面，在 l 和 $l+1$ 满足 $R_{b(l+1)}^e = R_{b(l)}^e R_{b(l+1)}^{b(l)}$ 中的真正的旋转矩阵。在这个等式中，得到用于线性误差传播下列表达式：

$$\tilde{\theta}_{l+1|l} \simeq \hat{R}_{l+1|l} \hat{R}_{l|l}^T \tilde{\theta}_{l|l} + \tilde{\theta}_{\Delta l} \quad (6-9)$$

根据第四章惯性传递的推导，可以得到速度和位置的线性传递表达式，与上式组合，可以得到：

$$\begin{bmatrix} \tilde{\theta}_{l+1|l} \\ \tilde{p}_{l+1|l} \\ \tilde{v}_{l+1|l} \end{bmatrix} = \begin{bmatrix} \hat{R}_{l+1|l} \hat{R}_{l|l}^T & 0_3 & 0_3 \\ -\hat{R}_{l|l}^T [\hat{y}_l \times] & I_3 & \Delta t I_3 \\ -\hat{R}_{l|l}^T [\hat{s}_l \times] & 0_3 & I_3 \end{bmatrix} \begin{bmatrix} \tilde{\theta}_{l|l} \\ \tilde{p}_{l|l} \\ \tilde{v}_{l|l} \end{bmatrix} + \begin{bmatrix} \tilde{\theta}_{\Delta l} \\ \hat{R}_{l|l}^T \hat{y}_l \\ \hat{R}_{l|l}^T \hat{s}_l \end{bmatrix} \quad (6-10)$$

其中，按照第四章的推导， $\hat{y}_l$ 与 $\hat{s}_l$ 有：

$$\begin{aligned}\hat{s}_l &= \hat{R}_{I_{l+1|l}}(\hat{v}_{l+1|l} - \hat{v}_{l|l} - g\Delta t) \\ \hat{y}_l &= \hat{R}_{I_{l|l}}(\hat{p}_{l+1|l} - p_{l|l} - \hat{v}_{l|l}\Delta t - \frac{1}{2}g\Delta t^2)\end{aligned}\quad (6-11)$$

因此 IMU 的状态传递为：

$$\Phi_{I_l} = \begin{bmatrix} \hat{R}_{I_{l+1|l}}\hat{R}_{I_{l|l}}^T & 0_3 & 0_3 \\ -\hat{R}_{I_{l|l}}^T[\hat{y}_l \times] & I_3 & \Delta t I_3 \\ -\hat{R}_{I_{l|l}}^T[\hat{s}_l \times] & 0_3 & I_3 \end{bmatrix}\quad (6-12)$$

完整的状态传递矩阵为：

$$\Phi_{I_l} = \begin{bmatrix} \Phi_{qq} & 0_3 & 0_3 & \Phi_{qb_g} & 0_3 \\ \Phi_{pq} & I_3 & \Delta t I_3 & \Phi_{pb_g} & \Phi_{pa} \\ \Phi_{vq} & 0_3 & I_3 & \Phi_{vb_g} & \Phi_{va} \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 \end{bmatrix}\quad (6-13)$$

上式中的各项为：

$$\begin{aligned}\Phi_{qq} &= \hat{R}_{I_{l+1|l}}\hat{R}_{I_{l|l}}^T \\ \Phi_{pq} &= -\hat{R}_{I_{l|l}}^T \left[(\hat{p}_{l+1|l} - p_{l|l} - \hat{v}_{l|l}\Delta t - \frac{1}{2}g\Delta t^2) \times \right] \\ \Phi_{vq} &= -\hat{R}_{I_{l|l}}^T [(\hat{v}_{l+1|l} - \hat{v}_{l|l} - g\Delta t) \times] \\ \Phi_{qb_g} &= -\hat{R}_{I_{l+1|l}}\hat{R}_{I_{l|l}}^T \int_{t_l}^{t_{l+1}} \hat{R}_\tau^l d\tau \\ \Phi_{pb_g} &= \int_{t_l}^{t_{l+1}} \int_{t_l}^w [(\dot{\hat{v}}_\tau - g) \times] \hat{R}_{I_{l|l}}^T \int_{t_l}^\tau \hat{R}_s^l ds d\tau dw \\ \Phi_{vb_g} &= \int_{t_l}^{t_{l+1}} [(\dot{\hat{v}}_\tau - g) \times] \hat{R}_{I_{l|l}}^T \int_{t_l}^\tau \hat{R}_s^l ds d\tau \\ \Phi_{pa} &= -\hat{R}_{I_{l|l}}^T \int_{t_l}^{t_{l+1}} \int_{t_l}^\tau \hat{R}_\tau^l ds d\tau \\ \Phi_{va} &= -\hat{R}_{I_{l|l}}^T \int_{t_l}^{t_{l+1}} \hat{R}_\tau^l d\tau\end{aligned}\quad (6-14)$$

协方差矩阵传递为：

$$\Sigma_{l+1} = \Phi_{l_l} \Sigma_l \Phi_{l_l}^T + Q_l \quad (6-15)$$

6.3 噪声矩阵

噪声矩阵为：

$$\begin{aligned} Q_l &= \int_{t_l}^{t_{l+1}} \Phi(t_{l+1}, \tau) G_c Q_c G_c^T \Phi(t_{l+1}, \tau)^T d\tau \\ &\simeq \frac{\Delta t}{2} \Phi(t + \Delta t, t) N_c \Phi(t + \Delta t, t)^T \end{aligned} \quad (6-16)$$

其中：

$$\begin{aligned} N_c &= G_c Q_c G_c^T \\ G_c &= \begin{bmatrix} -I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & -\hat{R}_l^T & 0_3 & 0_3 \\ 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 \end{bmatrix} \\ Q_c &= \begin{bmatrix} \sigma_{g_c}^2 I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & \sigma_{a_c}^2 I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & \sigma_{w_{g_c}}^2 I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & \sigma_{w_{a_c}}^2 I_3 \end{bmatrix} \end{aligned} \quad (6-17)$$

6.4 状态扩充

状态向量增加后，协方差矩阵也需要扩充：

$$\Sigma_{l+1|l} \leftarrow \begin{bmatrix} \Sigma_{l+1|l} & \Sigma_{l+1|l} J^T \\ \Sigma_{l+1|l} J & J \Sigma_{l+1|l} J^T \end{bmatrix}^T \quad (6-18)$$

此处：

$$J = \begin{bmatrix} I_3 & 0_3 & 0_3 & 0_3 & \dots \\ 0_3 & I_3 & 0_3 & 0_3 & \dots \\ 0_3 & 0_3 & I_3 & 0_3 & \dots \end{bmatrix} \quad (6-19)$$

6.5 更新过程

更新过程同卡尔曼滤波的更新，那么测量方程也同普通的卡尔曼滤波：

$$r = H\tilde{x} + n \quad (6-20)$$

计算卡尔曼增益：

$$K = \Sigma H^T (H \Sigma H^T + R)^{-1} \quad (6-21)$$

更新协方差矩阵：

$$\Sigma_{l+1|l+1} = (I - KH)\Sigma_{l+1|l}(I - KH)^T + K R K^T \quad (6-22)$$

更新状态：

$$\Delta x = K r \quad (6-23)$$

$$\Delta x = [\delta\theta^T \quad \Delta p^T \quad \Delta v^T \quad \Delta b_g^T \quad \Delta b_a^T \quad \dots \quad \delta\theta_i^T \quad \Delta p_i^T \quad \Delta v_i^T \quad \dots]^T \quad (6-24)$$

$$x_{l+1|l+1} = \begin{bmatrix} q \otimes \delta q \\ p + \Delta p \\ v + \Delta v \\ b_g + \Delta b_g \\ b_a + \Delta b_a \\ \vdots \\ q_i \otimes \delta q_i \\ p_i + \Delta p_i \\ v_i + \Delta v_i \end{bmatrix} \quad (6-25)$$

此处 $\delta q = \begin{bmatrix} \frac{1}{2} \delta\theta \\ 1 \end{bmatrix}$ 。

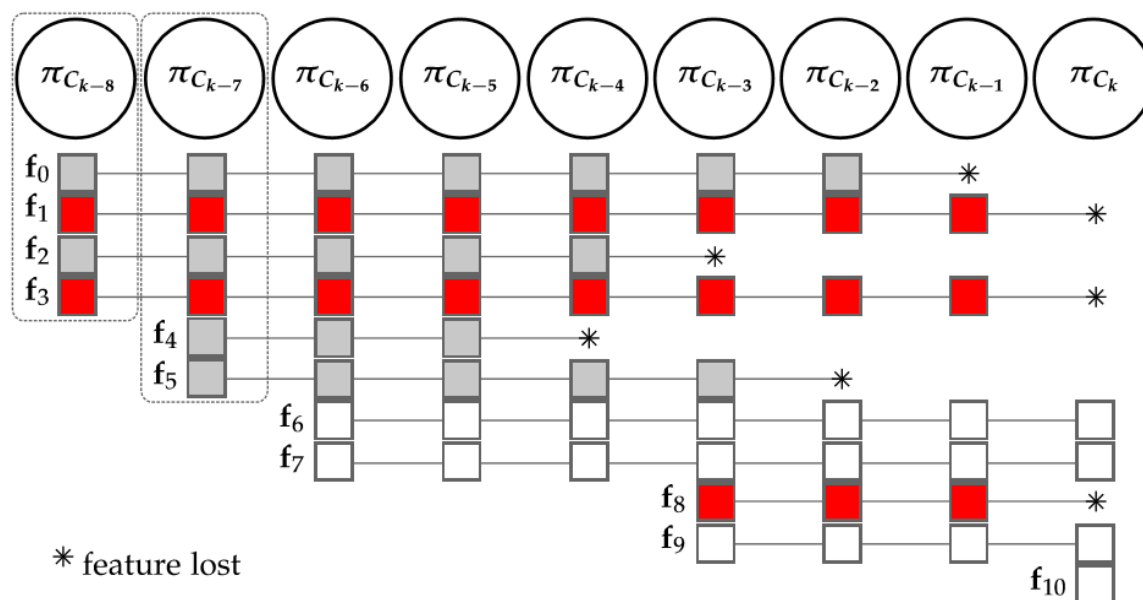


图 6-1 状态保持示意图

6.6 状态剔除

更新发生后，如果有在状态向量有太多的状态时，最早的位姿状态被去除，相应的行和协方差矩阵的列被切出。遗留下来的位姿也不再被跟踪，可以在此时移除。如图 6-1 所示。

第 7 章 系统初始化与矫正

7.1 状态初始化

由于 MSCKF 等等的一些基于滤波器的视觉惯性导航系统的基础是卡尔曼滤波器，导致他们同卡尔曼滤波器一样，在开始工作的时候需要初始化。好的初始化状态会使得滤波器能够较快的迭代收敛到最优估计附近，而差的初始化状态会导致迭代慢、收敛慢，或是收敛到错误结果上等等后果。

7.1.1 姿态初始化

使用加速度计与指南针进行系统姿态的初始化。初始化过程中，由于仅仅需要一个迭代开始，所以初始化状态应当尽可能的接近最优估计值，但是不必严格的与最优估计值相同。所以在初始化姿态时，可以采用将加速度计和指南针读数转化为欧拉角，使用欧拉角计算初始化四元数进行姿态初始化。

在系统开始工作的时候，在基本静止的状态下，一段时间内，读取加速度计均值 $\hat{a}^b$ 数据与指南针数据 $\hat{m}^b$ 。首先通过加速度计的读数计算初始化的俯仰角和滚转角：

俯仰角 θ_{init} ：

$$\theta_{init} = -\arcsin\left(\frac{\hat{a}_x}{\hat{g}}\right) \quad (7-1)$$

滚转角 ϕ_{init} ：

$$\phi_{init} = -\arcsin\left(\frac{\hat{a}_y}{\hat{g}}\right) \quad (7-2)$$

对于偏航角，需要用指南针数据进行处理，首先需要对指南针的读数进行倾斜补偿：

$$\begin{aligned} \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} &= R(\theta_{init})^T R(\phi_{init})^T \begin{bmatrix} m_{bx} \\ m_{by} \\ m_{bz} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}^T \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}^T \begin{bmatrix} m_{bx} \\ m_{by} \\ m_{bz} \end{bmatrix} \end{aligned} \quad (7-3)$$

即可得到初始偏航角：

$$\psi_{init} = -\arctan\left(\frac{h_y}{h_x}\right) \quad (7-4)$$

通过初始化欧拉角 $[\phi_{init} \theta_{init} \psi_{init}]^T$ 转化为姿态四元数：

$$\begin{aligned} q_1 &= \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_2 &= \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_3 &= \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) \\ q_4 &= \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \end{aligned} \quad (7-5)$$

那么，初始化姿态即为：

$$q_{init} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T \quad (7-6)$$

初始姿态 q_{init} 在几乎静止的状态下，经过后文所述的 IEKF 迭代后，即可得到初始的姿态，用于 MSCKF 迭代。

7.1.2 位置与速度初始化

相对姿态初始化而言，位置的初始化较为简单。在开始的时候，直接将初始化位置状态定为：

$$p_{init} = p_{e0} = [0 \quad 0 \quad 0]^T \quad (7-7)$$

而在初始化时，由于没有传感器能够给出绝对的速度信息，以及在随后的迭代过程中速度信息偏差会造成一系列的错误。故本套基于视觉与惯性的定位里程计要求在静止状态下开始初始化，初始速度状态为：

$$\boldsymbol{v}_{init} = \boldsymbol{v}_{e0} = [0 \quad 0 \quad 0]^T \quad (7-8)$$

7.1.3 特征点初始化

在初始时，由于仅有摄像机对特征点视觉观测，并且初始状态时 IMU 无法给出系统的移动尺度信息，视野内所有被追踪的特征点没有深度信息，会造成在从初始状态开始，视觉的观测都无效。

在本系统中，使用外部辅助标志进行视觉观测的初始化。通过一个已知尺寸的视觉标识物，在视野内通过已知尺寸的物体与已知的相机内参数求解三维对象的位置和姿态[32]，即为 PnP 的方法。

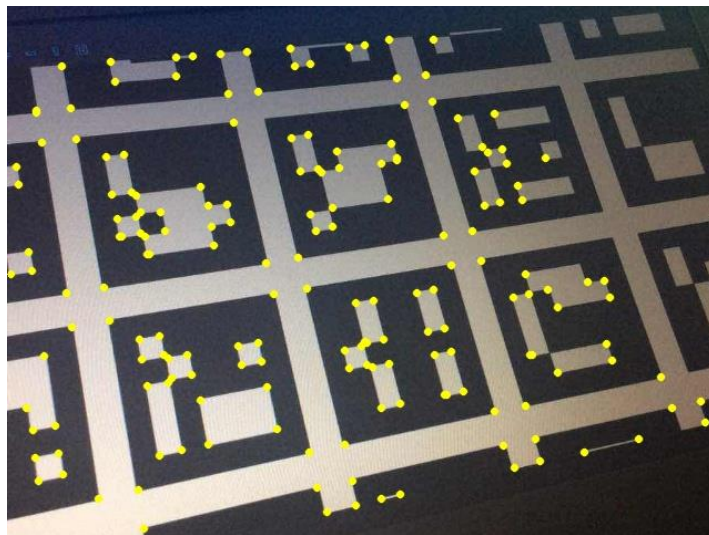


图 7-1 初始化板的 FAST 特征点

本系统中，通过快速的对简单的黑白二维码进行识别，并解算和初始化事业中的特征点的深度信息[33]，加入观测进行初始化。在系统开始工作并进行初始化的

时候，使摄像机正拍摄由一系列简单二维码构成的视觉识别图标平板。由于是正拍摄，所以平板上所有图标的所有特征点关于摄像机具有相同，并且可以由二维码测量出来的景深。通过对这些二维码上的特征点进行追踪，并通过二维码计算特征点深度，就可以在系统开始工作的时候，在静止状态下完成视觉测量的初始化工作。

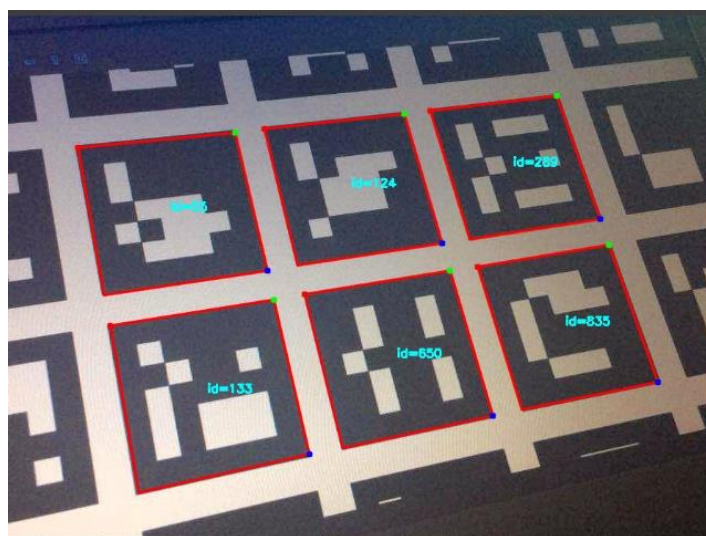


图 7-2 初始化板的二维码识别与定位

图 7-1 与图 7-2 即为初始化时，摄像机拍摄二维码平板的画面中通过 FAST 算法找到的特征点与二维码识别算法找到的可以计算平板景深的二维码。注意，图 7-1 与图 7-2 为了表示方便，是倾斜拍摄的，实际在初始化过程中，应该采取正拍摄的方式保证初始化时的特征点深度尽量一致。在相机进行了简单标定的条件下，可以计算出较为准确的特征点景深。使用的二维码识别库为开源库 ArUco[34]，在已经标定相机，并且二维码尺寸已知的前提下，可以给出二维码标志与相机的相对位置、相对姿态。

使用 PnP 的方法解算出来的景深具有一定的误差，且存在连续跳变等等情况，会导致识别二维码标志的时候，会出现数据跳变。但由于初始化的目的是给出一个尽量靠近真实值的初始值，之后在 MSCKF 的迭代过程中就可以给出持续的深度值

测量。故可以简单地取前 N 帧的图片进行 PnP 解算，做简单的中值滤波后即可作为初始的特征点景深，然后进行迭代。

取前 N 帧图像， N 为奇数，若为系统开始工作时，由 ArUco 二维码给出的景深序列，按大小顺序排列为：

$$z_0 \leq z_1 \leq \dots \leq z_{N-1}$$

则景深的初始化值为：

$$z_{init} = h_i \text{Med}\{z_0, z_1, \dots, z_{N-1}\} = h_i \frac{z_{N-1}}{2} \quad (7-9)$$

h_i 为第 i 个特征点的相对深度。相对深度通过 bundle adjustment 计算得到。Bundle adjustment 是一种通过最小二乘法等等方法，将二维的图像点反推三维实际点的误差进行最小化的方法[35]。通过这个方法，可以在多次逐帧的计算之后，解算出视野中特征点的无尺度相对深度，这个深度可以用于初始化 MSCKF 中的滑动窗口的特征点估计。

7.2 里程计的绝对校正

7.2.1 姿态的绝对校正

在移动平台静止状态下，例如地面机器人停车，或是空中机器人悬停的时候，摄像机获取的图像几乎没有帧差，会造成视觉观测失效，这时候由于陀螺仪漂移等，长时间会产生较大的姿态误差。为保证系统的稳定性，在判断出现移动平台静止的时刻进行绝对姿态的校正。

设计一个简单的滤波器在 MSCKF 的视觉观测失效的时候，通过加速度计和指南针的读数进行姿态更新，保持较好的状态传递，并持续观测陀螺仪零偏。

$$X = \begin{bmatrix} q_{eb} \\ b_g \end{bmatrix} \rightarrow \tilde{X} = \begin{bmatrix} \delta\theta \\ \delta b_g \end{bmatrix} \quad (7-10)$$

按照 $SO(3)$ 的连续旋转的传递方式，在四元数下表示姿态，表达为：

$$q_b^e = q_b^e \otimes \hat{q}_b^b \quad (7-11)$$

即是：

$$\begin{aligned} q &= \hat{q} \otimes \delta q \\ b_g &= \hat{b}_g + \delta b_g \end{aligned} \quad (7-12)$$

状态传递可以写为：

$$\begin{aligned} \begin{bmatrix} \delta\theta \\ \delta\dot{b}_g \end{bmatrix} &= \begin{bmatrix} -[\hat{\omega} \times] \delta\theta - \delta b_g - n_r \\ n_w \end{bmatrix} \\ &= \begin{bmatrix} -[\hat{\omega} \times] & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta\theta \\ \delta b_g \end{bmatrix} + \begin{bmatrix} -I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} n_r \\ n_w \end{bmatrix} \end{aligned} \quad (7-13)$$

这个就是姿态初始化过程中的 IEKF 传递模型。

那么，按照这样的过程，有：

$$X = \begin{bmatrix} \delta\theta \\ \delta\dot{b}_g \end{bmatrix} \quad F = \begin{bmatrix} -[\hat{\omega} \times] & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad G = \begin{bmatrix} -I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (7-14)$$

卡尔曼滤波状态中，过程协方差矩阵为：

$$\begin{aligned} \Phi &= \exp(F\Delta t) \approx I + F\Delta t = \begin{bmatrix} I - [\hat{\omega} \times]\Delta t & -I_{3 \times 3}\Delta t \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \\ Q &= \int_t^{t+1} \Phi(t+1, \tau) G(\tau) Q_c G(\tau)^T \Phi(t+1, \tau)^T d\tau \approx \begin{bmatrix} \sigma_r^2 \Delta t & -\sigma_w^2 \Delta t^2 \\ -\sigma_w^2 \Delta t^2 & \sigma_w^2 \Delta t \end{bmatrix} \end{aligned} \quad (7-15)$$

系统开始工作的时候，在静止状态进行姿态状态的初始化。

按照静止状态下加速度计和指南针测量模型：

$$v^b = R_e^b v^e = R(q_e^b) v^e \quad (7-16)$$

此方程即为加速度计或是指南针的测量模型。

那么，以误差为状态的 IEKF 的测量模型可以改写为：

$$\begin{aligned}\tilde{v}^b &= R_e^b v^e - \tilde{R}_e^b v^e \\ &= -[\delta\theta \times] R(q_e^b) v^e \\ &= [R(q_e^b) v^e \times] \delta\theta\end{aligned}\quad (7-17)$$

其中，对加速度计有 $v^e = g^e = [0 \ 0 \ 1]^T$ ；在不知道经纬度时，对指南针有 $v^e = m^e = [1 \ 0 \ 0]^T$ 。此方程即为加速度计的测量模型。

$$\tilde{v}^b = [[R(q_e^b) v^e \times] \ 0_{3 \times 3}] \begin{bmatrix} \delta\theta \\ \delta b_g \end{bmatrix} + n_m \quad (7-18)$$

那么，有：

$$H = [[R(q_e^b) v^e \times] \ 0_{3 \times 3}] \quad (7-19)$$

7.2.2 里程绝对校正

里程合速度的校准十分困难，因为本文所设计的系统是没有携带 GPS，声呐或是多普勒雷达等能够测量里程或是速度的设备。本文给出一般的里程校准方式。

在有 GPS 等等其他能够给出绝对定为辅的辅助传感器数据时，根据给出的绝对定位信息进行绝对位置的校正。以 p_0^e 进行移动平台起始位置初始化后，迭代计算，在 i 时刻位置为 p_i^e ，而在 i 时刻接收到绝对位置信息为 p_i^g ，那么，校正位移差为：

$$p_i^e = p_i^g + \Delta p^e \quad (7-20)$$

对于之前的每一点位移 p_n^e ($n \in [0, i]$)，都有：

$$p_n^e = \hat{p}_n^e + \Delta p^e \quad n \in [0, i] \quad (7-21)$$

第 8 章 系统构建与实验结果

本章节讲述建立仿真系统，在本地使用模拟器进行仿真。展示了仿真的架构，以及仿真的结果。

8.1 软件环境

运行环境为机器人操作系统（Robot Operating System，缩写 ROS）。ROS 是一款被广泛应用于机器人领域的各个分支机器人操作系统框架。作为一款机器人中间件系统，它构架在 Linux 平台之上，提供了一系列便捷接口。其以话题（Topic）为核心的消息总线（Message Bus）架构，使得机器人软件系统中的各个节点（Node）之间可以通过消息（Message）进行数据交换。各层级各模块使用该消息接口协同工作，大大减少了整个软件系统的耦合度，增强软件的可扩展性，使得架构的层次更加清晰。

本文所设计的仿真系统采用了以需求进化为核心的敏捷开发方式，系统应用了 ROS 框架，每个模块独立开发，每个模块同时也是 ROS 框架下的包，各个模块协同工作。

使用的 C++ 计算库为 Eigen。Eigen 是一个线性代数的开源的 C++ 模板库，以头文件的形式，完备了的向量和矩阵函数及相关算法。使用十分方便，功能强大，高效快速。

使用的 C++ 计算机视觉库为 openCV。openCV（全称为 Open Source Computer Vision Library）是一个主要采用 C/C++ 编写的可移植开源计算机视觉库。由于其执行效率高，封装完善，维护迅速，已经广泛地被全世界的视觉研究者所使用。

8.2 仿真

8.2.1 系统架构

为了使仿真程序足够的能够反映真实环境下算法的执行情况，本文的 VINS 仿真仅在数据源上构造仿真数据。获取数据之后对数据的处理及其他的架构和代码都一致。

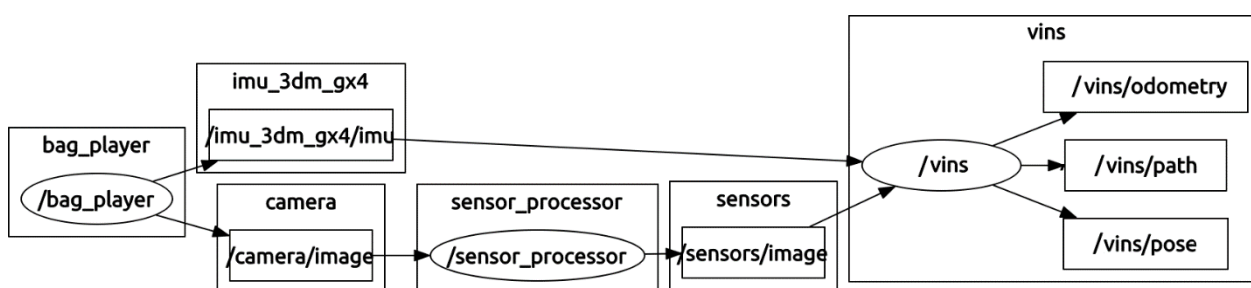


图 8-1 算法仿真架构图

如图 8-1 所示即为本文的算法仿真架构图。一共有三个主要节点，其中两个作为核心节点，一个节点作为生成仿真数据的数据包节点：

- **/sensor_processor**：核心节点，对摄像头获取的视频流进行特征获取、匹配等处理，输出具有上下文匹配关系的特征点的二维图像观测；
- **/vins**：核心节点，运行 MSCKF 算法，输入 IMU 数据和特征点的二维图像观测，生成里程、位姿和轨迹；
- **/bag_player**：仿真数据包节点包，这个包是用来生成仿真的模拟数据的，包括 IMU 数据和视觉的数据。

相机的仿真数据按照德国产的 mvBlueFOX-MLC200w 灰度相机生成，使用 90 度的广角镜头作为视觉输入源，以 30Hz 的 752×480 的视频流进行输入；IMU 的仿真数据按照 Microstrain 生产的 3DM-GX4 型微惯性测量单元生成，IMU 的输出频率为 500Hz。

8.2.2 仿真数据

在仿真中，设计质点摄像机以一定的轨迹移动，并设定移动的速度和移动的位置，即可得到一个周期的仿真轨迹。通过对这个周期轨迹的真值添加噪声，即可生成仿真的 IMU 数据。仿真轨迹由图 8-2 所示，即为有噪声的仿真输入数据。

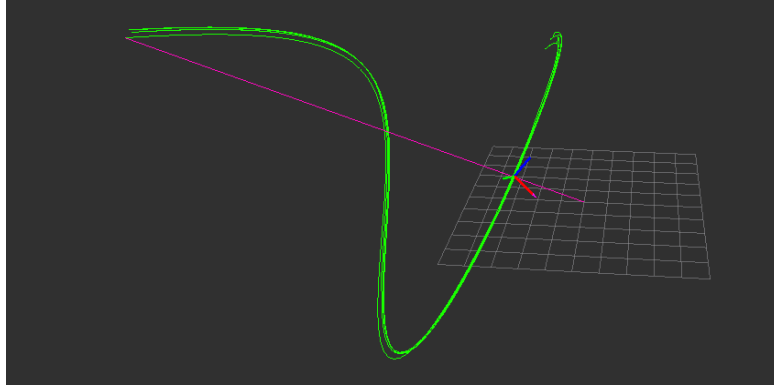


图 8-2 仿真轨迹示意图

仿真的轨迹方程如下， d_{max} 为轨迹最大范围， t_{max} 为循环周期时间：

位置 $[x \ y \ z]^T$ ：

$$\begin{cases} x = \frac{d_{max}}{2} + \frac{d_{max}}{2} \cos\left(\frac{\pi t}{t_{max}}\right) \\ y = \frac{d_{max}}{2} + \frac{d_{max}}{2} \cos\left(\frac{4\pi t}{t_{max}}\right) \\ z = \frac{d_{max}}{2} + \frac{d_{max}}{2} \cos\left(\frac{2\pi t}{t_{max}}\right) \end{cases} \quad (8-1)$$

速度 $[v_x \ v_y \ v_z]^T$ ：

$$\begin{cases} v_x = -\frac{d_{max}}{2} \frac{\pi}{t_{max}} \sin\left(\frac{\pi t}{t_{max}}\right) \\ v_y = -\frac{d_{max}}{2} \frac{4\pi}{t_{max}} \sin\left(\frac{4\pi t}{t_{max}}\right) \\ v_z = -\frac{d_{max}}{2} \frac{2\pi}{t_{max}} \sin\left(\frac{2\pi t}{t_{max}}\right) \end{cases} \quad (8-3)$$

加速度 $[a_x \ a_y \ a_z]^T$ ：

$$\begin{cases} a_x = -\frac{d_{max}}{2} \left(\frac{\pi}{t_{max}} \right)^2 \cos\left(\frac{\pi t}{t_{max}}\right) \\ a_y = -\frac{d_{max}}{2} \left(\frac{4\pi}{t_{max}} \right)^2 \cos\left(\frac{4\pi t}{t_{max}}\right) \\ a_z = -\frac{d_{max}}{2} \left(\frac{2\pi}{t_{max}} \right)^2 \cos\left(\frac{2\pi t}{t_{max}}\right) \end{cases} \quad (8-4)$$

仿真的图像数据来自于模拟器中，随机生成的固定点块，按照指定的摄像机内外参数成像生成的图片构成的视频流。随机生成的固定点块如图 8-3 所示。

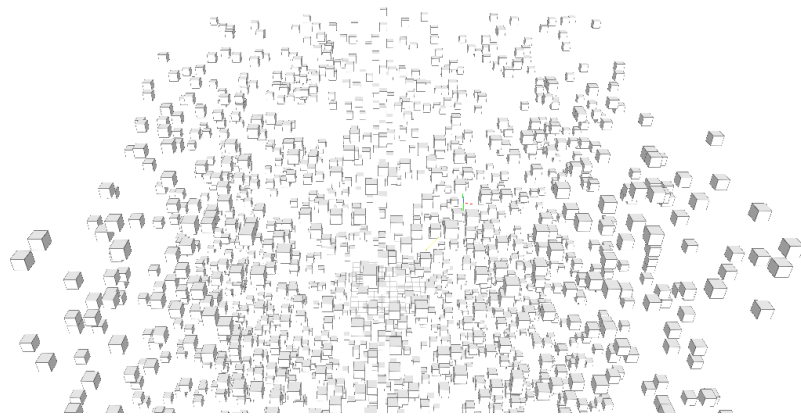


图 8-3 随机点块

添加噪声之后，真值相对于标准方程的轨迹有如图 8-4 的误差。

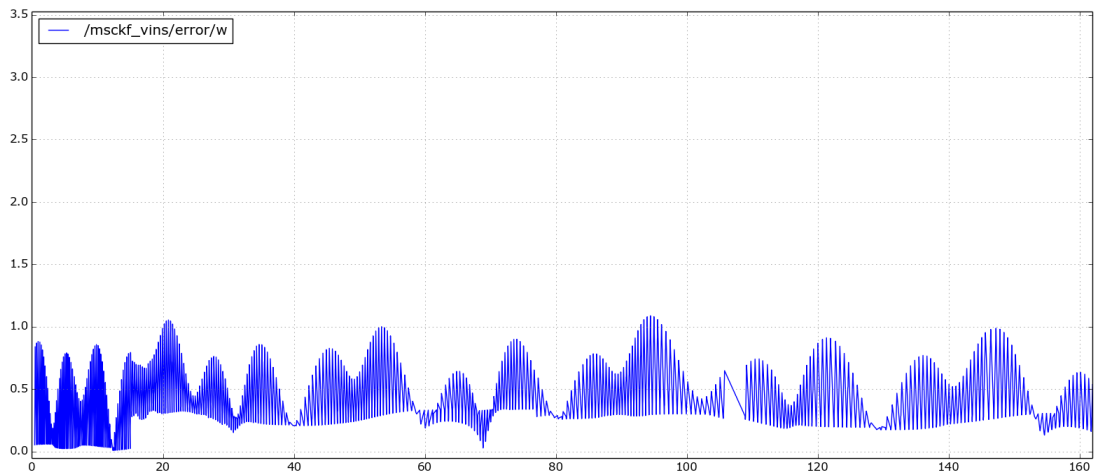


图 8-4 输入数据误差

仿真生成的轨迹如图 8-5 所示，由于没有较好初始化，故仿真中会出现一定程度上的尺度偏差，紫色为真值轨迹，绿色为输出轨迹。输出数据误差如图 8-6 所示，相对于图 8-4 的真值数据，基本保持了相应的变化趋势，同时误差的保持也较为稳定。

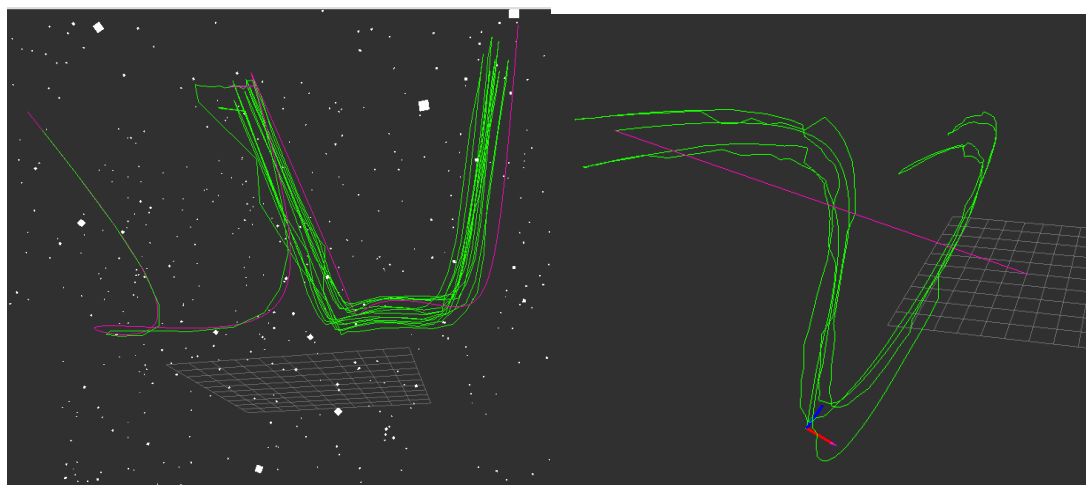


图 8-5 仿真结果轨迹

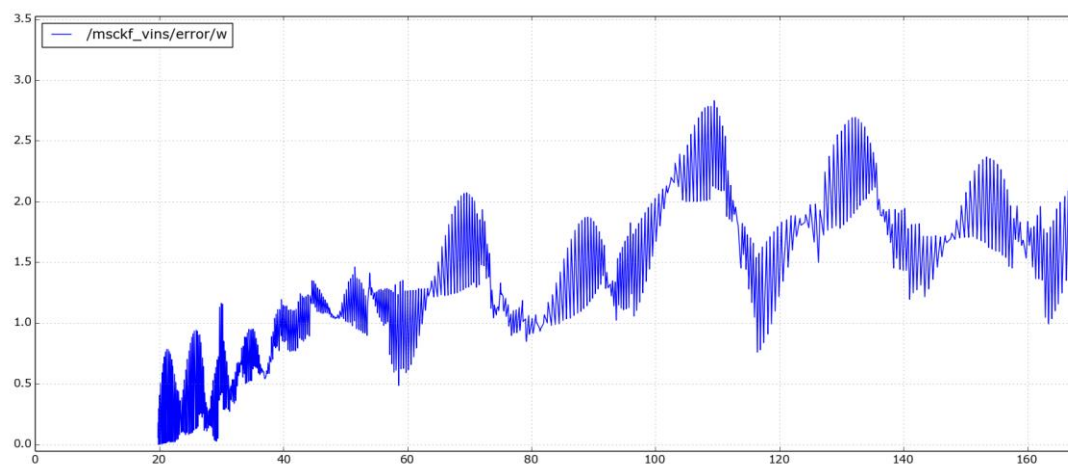


图 8-6 输出数据误差

对于一些轨迹，在具有一定的初始化的情况下，会具有较好的跟随效果。

对于普通的圆轨迹，追踪情况如图 8-7 所示；如图 8-6 的轨迹，在初始化较好的情况下，会表现的十分优秀，如图 8-8 所示，但在没有静止检测时，任然存在长距离移动后的偏移。

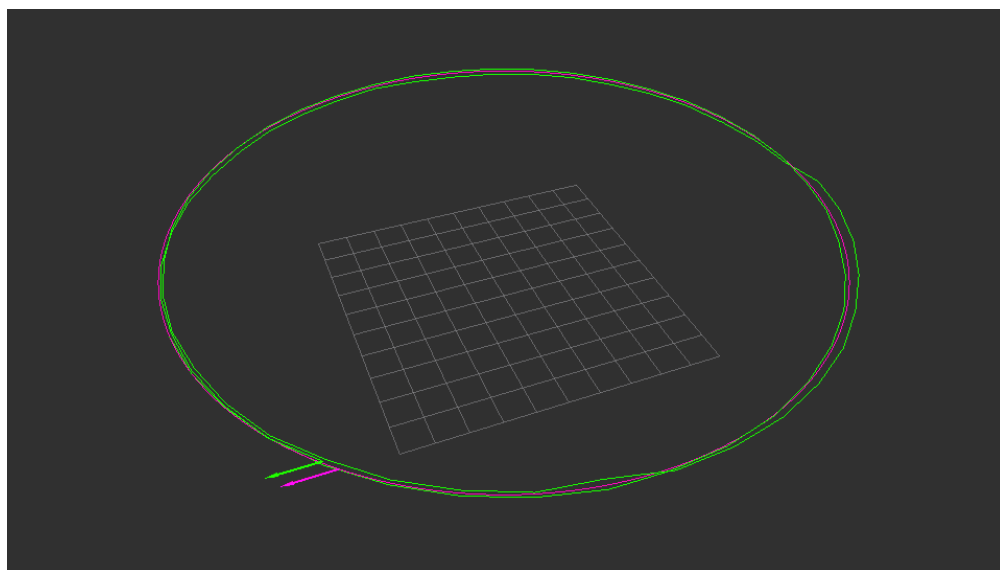


图 8-7 圆轨迹跟随情况

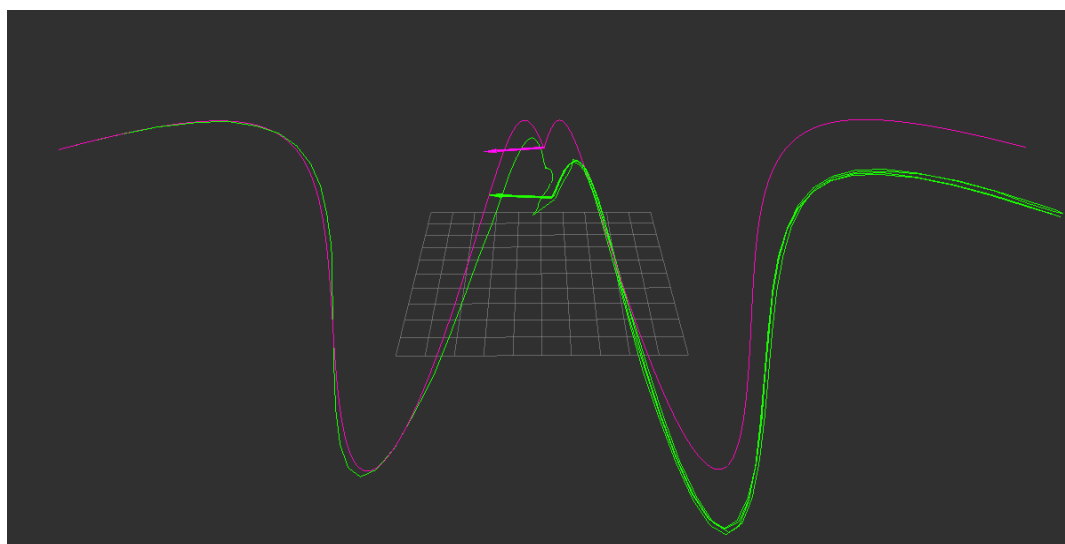


图 8-8 有初始化时的轨迹

第 9 章 结论与展望

在本文中，对于基于 EKF 的 MSCKF 算法，使用视觉辅助惯性导航进行了理论分析。

本文主要是完善了视觉的测量模型，该模型能够表达多个不同位姿的相机观察中出现的相同的静态特征点带来的几何约束。相机的位姿按照约束，以滑动窗口的方式维持在状态估计中。这种测量模型不像 EKF-SLAM 需要将三维特征点作为状态量加入估计中，算法仅在线性化时有一定特征数量相关的复杂度，之后的计算成本基本相同。算法能够非常准确的姿势在大规模实际环境估计。

本文集中介绍单目相机融合视觉测量与惯性测量。作为一个通用的一般性方法，可以应用于具有相应的传感器的移动平台，用于移动的测量。

同时，本文还介绍了以加速度计和指南针为核心的姿态初始化方法和以特殊标志物开始的视觉初始化方法。

致谢

时光就这样跨越了四年，让我从一个懵懂的高中生成长成为一个懂事的青年。在这四年间，经过学习，外出访问，参加比赛，公司实习等一系列的活动，我收获了很多。同时，和很多人成为朋友，在前进的道路上互相帮助，也让我倍感欣慰。尤其是毕业设计完成期间，受到了很多帮助。在此感谢所有帮助过我的人。

首先要感谢毕业设计的课题指导老师孟彦彬。孟老师在毕业设计过程中，给了我自由和空间，让我在毕业设计期间有时间有机会能够接触到香港科技大学的学者以及企业上的工程师等能够提供帮助的人。在孟老师的鼓励和支持下，我能够大胆的放开思路，从多方面考虑问题，尽量多的接触外界的环境，最终完成课题。

尤其要感谢来自香港科技大学的沈邵劼教授。沈教授在这个方向上有造诣精深，可以代表相关领域全世界最高学术水平。在课题上，沈教授帮助我指明了探索与实现的方向，带领我构建了理论框架，梳理了知识，最终指导我完成课题。

感谢深圳市大疆创新科技有限公司的工程师杨硕和光流科技的创始人胡博涛，在他们的指引下我补充了理论知识，较为深入的理解了课题在工程上的应用，以及让我有了对工业界的接触。尤其是杨硕带领我们在“脑洞中心”小组进行了很多有意义的讨论，丰富了各个邻近领域知识的认知。感谢中国科技大学的同学栗培梁，为这一样的目标做出了不同的思考；华中科技大学的刘毅博士，深夜的讨论总是让人思维开阔。

自然，还要感谢父母二十余年来的养育和支持。在困难的时候，家是最平静的港湾，父母是最真诚的陪伴。

在此，向所有为我提供过帮助的人说一声谢谢。

高文良

2016 年 5 月 1 日

参考文献

- [1] Hartley R, Zisserman A. Multiple view geometry in computer vision[M]. Cambridge university press, 2003.
- [2] Izadi S, Kim D, Hilliges O, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera[C]//Proceedings of the 24th annual ACM symposium on User interface software and technology. ACM, 2011: 559-568.
- [3] Bachrach A, Prentice S, He R, et al. RANGE – Robust autonomous navigation in GPS - denied environments[J]. Journal of Field Robotics, 2011, 28(5): 644-666.
- [4] Nützi G, Weiss S, Scaramuzza D, et al. Fusion of IMU and vision for absolute scale estimation in monocular SLAM[J]. Journal of intelligent & robotic systems, 2011, 61(1-4): 287-299.
- [5] Scaramuzza D, Fraundorfer F. Visual odometry [tutorial][J]. Robotics & Automation Magazine, IEEE, 2011, 18(4): 80-92.
- [6] Hesch J A, Kottas D G, Bowman S L, et al. Consistency analysis and improvement of vision-aided inertial navigation[J]. Robotics, IEEE Transactions on, 2014, 30(1): 158-176.
- [7] Scaramuzza D, Achtelik M C, Doitsidis L, et al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments[J]. Robotics & Automation Magazine, IEEE, 2014, 21(3): 26-40.
- [8] Leutenegger S, Lynen S, Bosse M, et al. Keyframe-based visual–inertial odometry using nonlinear optimization[J]. The International Journal of Robotics Research, 2015, 34(3): 314-334.
- [9] Huang A S, Bachrach A, Henry P, et al. Visual odometry and mapping for autonomous flight using an RGB-D camera[C]//International Symposium on Robotics Research (ISRR). 2011, 2.

- [10] Scaramuzza D, Achtelik M C, Doitsidis L, et al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments[J]. Robotics & Automation Magazine, IEEE, 2014, 21(3): 26-40.
- [11] Shen S, Michael N, Kumar V. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs[C]//Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015: 5303-5310.
- [12] Lange S, Sunderhauf N, Protzel P. Incremental smoothing vs. filtering for sensor fusion on an indoor UAV[C]//Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013: 1773-1778.
- [13] Weiss S, Achtelik M W, Lynen S, et al. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments[C]//Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012: 957-964.
- [14] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]//Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, 2007: 225-234.
- [15] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]//Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014: 15-22.
- [16] Davison A J. Real-time simultaneous localisation and mapping with a single camera[C]//Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003: 1403-1410.
- [17] Weiss S, Achtelik M W, Lynen S, et al. Monocular Vision for Long - term Micro Aerial Vehicle State Estimation: A Compendium[J]. Journal of Field Robotics, 2013, 30(5): 803-831.

- [18] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]//Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, 2007: 225-234.
- [19] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]//Robotics and Automation, 2007 IEEE International Conference on. IEEE, 2007: 3565-3572.
- [20] Li M, Mourikis A I. 3-D motion estimation and online temporal calibration for camera-IMU systems[C]//Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013: 5709-5716.
- [21] 秦永元, 张洪钺, 汪叔华. 卡尔曼滤波与组合导航原理 (第 2 版) [M]. 西北工业大学出版社, 2011.
- [22] 冯刘中. 基于多传感器信息融合的移动机器人导航定位技术研究[D]. 西南交通大学, 2011.
- [23] Shi J, Tomasi C. Good features to track[C]//Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. IEEE, 1994: 593-600.
- [24] Rosten E, Drummond T. Machine learning for high-speed corner detection[M]//Computer Vision—ECCV 2006. Springer Berlin Heidelberg, 2006: 430-443.
- [25] Lowe D G. Object recognition from local scale-invariant features[C]//Computer vision, 1999. The proceedings of the seventh IEEE international conference on. IEEE, 1999, 2: 1150-1157.
- [26] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features[M]//Computer vision—ECCV 2006. Springer Berlin Heidelberg, 2006: 404-417.
- [27] Calonder M, Lepetit V, Strecha C, et al. Brief: Binary robust independent elementary features[J]. Computer Vision—ECCV 2010, 2010: 778-792.

- [28] Leutenegger S, Chli M, Siegwart R Y. BRISK: Binary robust invariant scalable keypoints[C]//Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011: 2548-2555.
- [29] Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF[C]//Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011: 2564-2571.
- [30] Alahi A, Ortiz R, Vandergheynst P. Freak: Fast retina keypoint[C]//Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012: 510-517.
- [31] Breckenridge W G. Quaternions proposed standard conventions[J]. Jet Propulsion Laboratory, Pasadena, CA, Interoffice Memorandum IOM, 1999: 343-79.
- [32] Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate o (n) solution to the pnp problem[J]. International journal of computer vision, 2009, 81(2): 155-166.
- [33] Garrido-Jurado S, Muñoz-Salinas R, Madrid-Cuevas F J, et al. Automatic generation and detection of highly reliable fiducial markers under occlusion[J]. Pattern Recognition, 2014, 47(6): 2280-2292.
- [34] ArUco: a minimal library for Augmented Reality applications based on OpenCV, <http://www.uco.es/investiga/grupos/ava/node/26>
- [35] Triggs B, McLauchlan P F, Hartley R I, et al. Bundle adjustment—a modern synthesis[M]//Vision algorithms: theory and practice. Springer Berlin Heidelberg, 1999: 298-372.

附录

附录 A $SO(3)$ 数学基础

接下来，按照[43] [44]描述流形 $SO(3)$ 的定义与性质

群是一种集合加上一种运算的代数结构，记作 $(A, \cdot)$ 。具有连续性质的群被称为李群，其群上的运算无限可微。

特殊正交群 $SO(n)$ 即为旋转矩阵群，表达为：

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

特殊的，在 $n = 3$ 的情况下，特殊正交群仅表示三维空间下的旋转，即为三维旋转群 $SO(3)$ ：

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} | RR^T = I, \det(R) = 1\} \quad (A-2)$$

三维旋转群的元素为表示三维空间任意旋转的唯一旋转矩阵，为 3×3 的正交矩阵，其逆矩阵就是其转置矩阵，且行列式值为 1。 $SO(3)$ 群可以表示光滑的流形，其流形的本地空间表示为 $\mathfrak{so}(3)$ ，即 $SO(3)$ 的李代数，由 3×3 的斜对称矩阵构成。

在所有在 $\mathbb{R}^3$ 下的斜对称矩阵和向量之间，定义运算：

$$\omega^\wedge = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^T = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3) \quad (A-3)$$

同样，可以定义 $(\cdot)^\vee$ 为其逆运算，即如果 $\omega^\wedge = S$ ，那么， $\omega = S^\vee$ 。可以得到下面这个有用的性质：

$$a^\wedge b = -b^\wedge a, \forall a, b \in \mathbb{R}^3 \quad (A-4)$$

定义指数映射 $\exp$ 运算表示 $SO(3) \rightarrow \mathfrak{so}(3)$ 即李代数与旋转之间的关系：

$$\exp(\phi^\wedge) = I + \frac{\sin(\|\phi\|)}{\|\phi\|} \phi^\wedge + \frac{1 - \cos(\|\phi\|)}{\|\phi\|^2} (\phi^\wedge)^2 \quad (A-5)$$

对其做一阶近似，得：

$$\exp(\phi^\wedge) \approx \mathbf{I} + \phi^\wedge \quad (\text{A- 6})$$

定义对数映射 $\log$ 运算表示将 $SO(3)$ 上的旋转矩阵转化为李代数下的斜对称矩阵：

$$\log(R) = \frac{\varphi(R-R^T)}{2 \sin(\varphi)} \quad (\text{A- 7})$$

其中， φ 表示为：

$$\varphi = \cos^{-1}\left(\frac{\text{tr}(R)-1}{2}\right) \quad (\text{A- 8})$$

需要注意的就是，通过 $\log(R)^\vee = \mathbf{a}\varphi$ 的转化，可以将一个 $SO(3)$ 上的旋转矩阵转化为 $\mathbf{a}$ 和 φ ，即旋转轴和旋转角来表示。

如果将旋转限制在开放球，即 $\|\phi\| < \pi$ ，那么 $\exp$ 映射是双射，其逆就是 $\log$ 映射。但是，由于不限制旋转域，那么 $\exp$ 映射变得满映射，即对于 $SO(3)$ 上的一个唯一旋转矩阵 R ，有 $\phi = (\varphi + 2k\pi)\mathbf{a}$ 的矢量与之对应。但是 $\mathfrak{so}(3)$ 中的元素对应到同一个 $SO(3)$ 元素上。对于一个旋转角 φ ，其旋转有周期性。

将 $\exp$ 映射与 $\log$ 映射表达为如下：

$$\begin{aligned} \exp: \mathbb{R}^3 \ni \phi &\rightarrow \exp(\phi^\wedge) \in SO(3) \\ \log: SO(3) \ni R &\rightarrow \log(R)^\vee \in \mathbb{R}^3 \end{aligned} \quad (\text{A- 9})$$

也就是跳过 $\mathfrak{so}(3)$ 直接用向量表示。

接下来，将其进行一阶近似展开：

$$\exp(\phi + \delta\phi) \approx \exp(\phi)\exp(J_r(\phi)\delta\phi) \quad (\text{A- 10})$$

其中， $J_r(\phi)$ 表示的是 $SO(3)$ 下的右雅可比矩阵，表示旋转空间的正切空间的右乘增量，如图 A-1 所示。

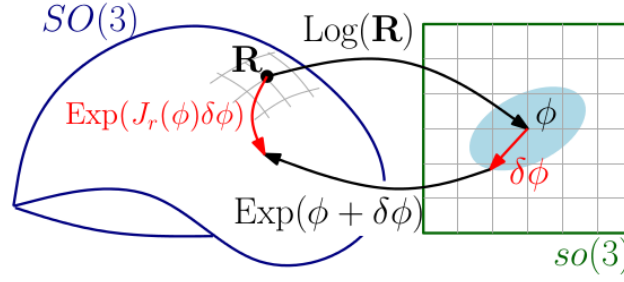


图 A-1 $SO(3)$ 与 $so(3)$ 关系表示

$$J_r(\phi) = \mathbf{I} - \frac{1 - \cos(\|\phi\|)}{\|\phi\|^2} \phi^\wedge + \frac{\|\phi\| - \sin(\|\phi\|)}{\|\phi\|^3} (\phi^\wedge)^2 \quad (\text{A- 11})$$

相似的， $\log$ 映射也可以写为一阶近似：

$$\log(\exp(\phi) \exp(\delta\phi)) \approx \phi + J_r^{-1}(\phi) \delta\phi \quad (\text{A- 12})$$

雅格比矩阵的逆表达为：

$$J_r^{-1}(\phi) = \mathbf{I} + \frac{1}{2} \phi^\wedge + \left(\frac{1}{\|\phi\|^2} + \frac{1 + \cos(\|\phi\|)}{2\|\phi\| \sin(\|\phi\|)} \right) (\phi^\wedge)^2 \quad (\text{A- 13})$$

右雅可比矩阵 $J_r(\phi)$ 和其逆 $J_r^{-1}(\phi)$ 最终还原到单位矩阵。

由此，可以推导出另一个有用的表达：

$$R \exp(\phi) R^T = \exp(R \phi^\wedge R^T) = \exp(R\phi) \quad (\text{A- 14})$$

$$\Leftrightarrow \exp(\phi) R = R \exp(R^T \phi) \quad (\text{A- 15})$$

附录 B 四元数

B.1 四元数的定义

四元数通常定义为：

$$q = \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 + q_4 \quad (\text{B- 1})$$

其中， $\mathbf{i}, \mathbf{j}, \mathbf{k}$ 为超越虚数，满足：

$$\begin{aligned} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i}\mathbf{j}\mathbf{k} = -1 \\ \mathbf{i}\mathbf{j} = -\mathbf{j}\mathbf{i} = \mathbf{k} \quad \mathbf{j}\mathbf{k} = -\mathbf{k}\mathbf{j} = \mathbf{i} \quad \mathbf{k}\mathbf{i} = -\mathbf{i}\mathbf{k} = \mathbf{j} \end{aligned} \quad (\text{B- 2})$$

需要特别注意的是，这个表达为汉密尔顿表达，是 1843 年由爱尔兰数学家汉密尔顿（William Rowan Hamilton）提出的。与汉密尔顿表达不同，1979 年 NASA 的喷气推进实验室为了让四元数具有直观和自然表述提出另一种表达[31]。本文规定四元数表达为汉密尔顿表达。

四元数 q 中， q_4 称为实部或者数值， $\mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3$ 为虚部或者向量部分。四元数可以表达为一个四列的矩阵 $\bar{q}$ ：

$$\bar{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = [q_1 \quad q_2 \quad q_3 \quad q_4]^T \quad (\text{B- 3})$$

四元数的乘法定义为：

$$\begin{aligned} \bar{q} \otimes \bar{p} &= (\mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 + q_4)(\mathbf{i}p_1 + \mathbf{j}p_2 + \mathbf{k}p_3 + p_4) \\ &= \begin{bmatrix} q_4p_1 - q_3p_2 + q_2p_3 + q_1p_4 \\ q_3p_1 + q_4p_2 - q_1p_3 + q_2p_4 \\ -q_2p_1 + q_1p_2 + q_4p_3 + q_3p_4 \\ -q_1p_1 - q_2p_2 - q_3p_3 + q_4p_4 \end{bmatrix} = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \end{aligned} \quad (\text{B- 4})$$

B.2 旋转四元数

当四元数 q 满足：

$$\bar{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{k}} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} = \begin{bmatrix} k_x \sin(\theta/2) \\ k_y \sin(\theta/2) \\ k_z \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \quad (\text{B- 5})$$

的时候，四元数表示一个旋转，被称为旋转四元数。其中， $\hat{\mathbf{k}}$ 为一个单位向量，表示旋转轴， θ 表示旋转的角度。此时，四元数 $\bar{q}$ 满足：

$$\bar{q} = \sqrt{\bar{q}^T \bar{q}} = \sqrt{|\mathbf{q}|^2 + q_4^2} = 1 \quad (\text{B- 6})$$

自此，即可以用四元数 q 表示任意三维空间中的旋转。

同时，可以定义单位四元数表示乘积为本身四元数，定义如下：

$$\begin{aligned} \bar{q}_I &= [0 \quad 0 \quad 0 \quad 1]^T \\ \bar{q} \otimes \bar{q}_I &= \bar{q}_I \otimes \bar{q} = \bar{q} \end{aligned} \quad (\text{B- 7})$$

逆旋转可以表达为旋转四元数的逆。对于单位四元数，其逆就是其共轭四元数：

$$\bar{q}^{-1} = \begin{bmatrix} -\mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} -\hat{\mathbf{k}} \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{k}} \sin(-\theta/2) \\ \cos(-\theta/2) \end{bmatrix} \quad (\text{B- 8})$$

同时四元数的逆满足：

$$\begin{aligned} \bar{q} \otimes \bar{q}^{-1} &= \bar{q}^{-1} \otimes \bar{q} = \bar{q}_I \\ (\bar{q} \otimes \bar{p})^{-1} &= \bar{q}^{-1} \otimes (\bar{p})^{-1} \end{aligned} \quad (\text{B- 9})$$

四元数同样可以表达旋转的连续传递：

$$\bar{q}_{b(t+\Delta t)}^e = \bar{q}_{b(t)}^e \otimes \bar{q}_{b(t+\Delta t)}^{b(t)} \quad (\text{B- 10})$$

两个坐标系下的向量的关系可以表达为：

$$\begin{aligned} \bar{p}^a &= \bar{q}_b^a \otimes \bar{p}^b \otimes \bar{q}_b^{a-1} \\ \bar{p}^a &= R_b^a * \bar{p}^b \end{aligned} \quad (\text{B- 11})$$

对于四元数 q_b^a ，其与旋转矩阵之间的关系表达为：

$$R_b^a = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1^2 q_2^2 - q_3^2 q_4^2) & 2(q_1^2 q_3^2 + q_2^2 q_4^2) \\ 2(q_1^2 q_2^2 + q_3^2 q_4^2) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2^2 q_3^2 - q_1^2 q_4^2) \\ 2(q_1^2 q_3^2 - q_2^2 q_4^2) & 2(q_2^2 q_3^2 + q_1^2 q_4^2) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (\text{B-12})$$

此式中， $q = q_b^a$ 。

例如，对于如图B-1所示的绕Z轴旋转 θ ，则这个旋转用四元数表达和用旋转矩阵表达分别如下：

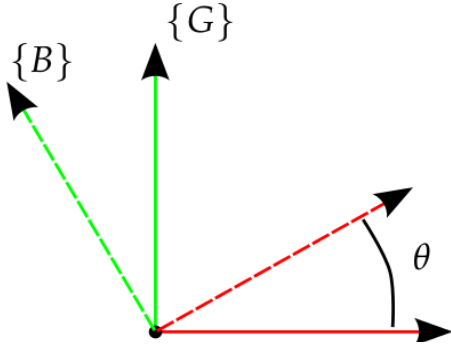


图 B-1 坐标系旋转

$$R_B^G = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B-13})$$

$$q_B^G = \begin{bmatrix} 0 \\ 0 \\ \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \quad (\text{B-14})$$

B.3 四元数微分方程

定义一个四元数 $\delta\bar{q}$ ，表示从时刻 t 到时刻 $t + \Delta t$ 发生一个非常小的旋转 $\delta\theta$ 。按照公式的表达,同时依据三角函数小角度近似， $\delta\bar{q}$ 可以写为：

$$\delta\bar{q} = \bar{q}_{b(t+\Delta t)}^{b(t)} = \begin{bmatrix} \hat{k} \sin(\delta\theta/2) \\ \cos(\delta\theta/2) \end{bmatrix} \approx \begin{bmatrix} \hat{k} \delta\theta/2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \delta\theta \\ 1 \end{bmatrix} \quad (\text{B-15})$$

向量 $\delta\theta$ 表达了这个小旋转的旋转轴和旋转量，当这个向量对时间取导，即除以 Δt 并取极限后，可以表达为角速度：

$$\omega = \lim_{\Delta t \rightarrow 0} \frac{\delta\theta}{\Delta t} \quad (\text{B-16})$$

用一个矩阵 Ω 来表示向量与四元数的乘积，对于角速度，表示为 $\Omega(\omega)$:

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} = \begin{bmatrix} -[\omega \times] & \omega \\ \omega^T & 0 \end{bmatrix} \quad (\text{B- 17})$$

对于在本地坐标系 L 下表示旋转的四元数 $\bar{q}$ ，取其在 t 时刻的导数，依据四元数连续旋转公式，可以表达为：

$$\bar{q}_{b(t+\Delta t)}^e = \bar{q}_{b(t)}^e \otimes \bar{q}_{b(t+\Delta t)}^{b(t)} \quad (\text{B- 18})$$

既有：

$$\begin{aligned} \dot{\bar{q}}_{b(t)}^e &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} (\bar{q}_{b(t+\Delta t)}^e - \bar{q}_{b(t)}^e) \\ &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} (\bar{q}_{b(t)}^e \otimes \bar{q}_{b(t+\Delta t)}^{b(t)} - \bar{q}_{b(t)}^e) \\ &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} (\bar{q}_{b(t)}^e \otimes \bar{q}_{b(t+\Delta t)}^{b(t)} - \bar{q}_{b(t)}^e \otimes \bar{q}_I) \\ &\approx \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left(\bar{q}_{b(t)}^e \otimes \left(\begin{bmatrix} \frac{1}{2} \delta \theta \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \right) \\ &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left(\bar{q}_{b(t)}^e \otimes \begin{bmatrix} \frac{1}{2} \delta \theta \\ 0 \end{bmatrix} \right) \\ &= \frac{1}{2} \bar{q}_{b(t)}^e \otimes \begin{bmatrix} \omega \\ 0 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} -[\omega \times] & \omega \\ -\omega^T & 0 \end{bmatrix} \bar{q}_{b(t)}^e \\ &= \frac{1}{2} \Omega(\omega) \bar{q}_{b(t)}^e \end{aligned} \quad (\text{B- 19})$$

以上即为小角度近似下的四元数微分方程。尤其注意，由于 $\omega = {}^L\omega$ ，所以此处表达的角速度为本地角速度，或者描述为机体角速度。