



北京大学

# 硕士研究生学位论文

题目：具有共面约束的多特征同  
时定位与地图构建方法

姓 名：\_\_\_\_\_ 李鑫

学 号：\_\_\_\_\_ 1801210376

院 系：\_\_\_\_\_ 软件与微电子学院

专 业：\_\_\_\_\_ 电子与通信工程

研究方向：\_\_\_\_\_ 嵌入式系统设计

导 师：\_\_\_\_\_ 林金龙 教授

二〇二一年六月



## 版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以其他方式传播。否则一旦引起有碍作者著作权之问题，将可能承担法律责任。





## 摘要

同时定位和地图构建 (Simultaneous Localization and Mapping, SLAM) 是指基于传感器信息的同时对系统周围的环境信息和系统本身的位姿进行估计, 是移动机器人可以实现自主移动的关键性技术。因为视觉传感器在信息提取和硬件成本方面具有巨大优势, 所以视觉 SLAM 在 SLAM 技术中的发展势头非常强劲。传统视觉 SLAM 大多单纯地依赖点特征, 点特征在纹理信息不丰富的区域中不容易从图像中被提取和匹配。而且在人造的结构化场景下, 点特征不能很好地利用空间中的结构信息。于是本身具有空间结构信息和不依赖细节纹理的直线特征和平面特征这类几何特征逐渐被重视了起来。本文主要研究了如何在 SLAM 系统中加入直线特征以及点线特征的共面约束这些结构化信息。

本文的主要工作包括:

1. 提出了一个融合共面信息的点线视觉惯性里程计 (Visual-Inertial Odometry, VIO)。该系统提出了一种利用 VIO 产生的稀疏空间点云实时构建半稠密的三维空间网格的方法。并结合惯性测量单元 (Inertial Measurement Unit, IMU) 数据在三维空间网格中检测出空间水平和垂直的平面以及空间平面上空间点线特征的共面关联。之后将检测到的共面信息融合到点线 VIO 系统中, 将直线和点线共面约束加入传统只有点特征的 VIO 系统中, 最后构建实验证明了直线和共面信息使得 VIO 系统对位姿的估计和空间地图的重建精度都得到了提升。

2. 对于共面的点线特征, 提出了一种新型的通过空间平面来参数化的方法。首先摆脱了上部分必须通过 IMU 数据来检测空间平面的束缚, 提出了一种只需要利用图像信息就可以直接检测空间平面和共面信息的方法。接着提出了使用空间平面来参数化该平面上的共面点线, 从而将共面信息直接加入到 SLAM 系统中。最后通过实验验证了平面找寻方法以及共面参数化方法的有效性, 并验证了这种共面参数化方法比直接在优化后端添加共面约束要更加高效。

关键词: 同时定位与地图构建, 特征参数化, 多传感器融合



# Multi-feature Simultaneous Localization and Mapping with coplanar constraints

Xin Li (Electronic and Communication Engineering)

Directed by Prof. Jinlong Lin

## ABSTRACT

Simultaneous Localization And Mapping (SLAM) is a key technology for mobile robot to realize autonomous movement, which is to estimate the environment information around the system and the pose of the system based on sensor information. Because vision sensor has great advantages in information extraction and hardware cost, vision slam has a strong development momentum in slam technology. Traditional visual slam mostly relies on point features, but point features are not easy to be extracted and matched from the image in the region with not rich texture information. Moreover, in the artificial structured scene, point features can not make good use of the structural information in space. As a result, geometric features such as line features and plane features, which have spatial structure information and do not depend on detail texture, have been paid more and more attention. This paper mainly studies how to add line feature and co-planar constraint of point line feature into SLAM system.

The main work of this paper includes:

1. a Visual-Inertial Odometry (VIO) is proposed which combines co-planar information. This system proposes a method to construct semi-dense 3D space mesh in real time by using sparse space point cloud generated by VIO. The co-planar correlation of the spatial horizontal and vertical plane and the feature of the spatial point line on the space plane is detected in the 3D space mesh by combining the inertial measurement unit (IMU) data. Then, the detected co-planar information is integrated into the point line vio system, and the co-planar constraints of the line and the point line are added to the traditional point only VIO system. Finally, the construction experiment proves that the line and co-planar information make the pose estimation of VIO system and the reconstruction accuracy of spatial map improved.

2. for the co-planar point line characteristics, a new method of parameterization by space plane is proposed. Firstly, we get rid of the constraint that the upper part must detect the space plane by IMU data. A method that can detect the spatial plane and co-planar information directly is proposed by using the image information. Then, the paper proposes that the co-

planar point line on the plane is parameterized by using the space plane, so that the co-planar information is directly added to the SLAM system. Finally, the effectiveness of the plane searching method and co-planar parameterization method is verified by experiments. It is proved that the co-planar parameterization method is more efficient than adding co-planar constraints directly to the optimization back end.

**KEY WORDS:** SLAM, feature parametrization, sensor fusion



## 目录

<b>第一章 绪论</b>	<b>1</b>
1.1 课题研究背景	1
1.2 研究现状	3
1.2.1 视觉 SLAM	3
1.2.2 视觉和 IMU 融合的 SLAM	5
1.2.3 结构化 SLAM	6
1.3 本文研究内容	6
1.4 本文结构	7
<b>第二章 相关背景知识</b>	<b>9</b>
2.1 相机模型	9
2.2 位姿变换	11
2.2.1 坐标系定义	11
2.2.2 位姿定义	11
2.3 IMU 测量以及预积分	13
2.3.1 IMU 测量模型	14
2.3.2 IMU 预积分	15
2.4 基于优化的 SLAM 方法	16
2.4.1 最大后验概率估计	16
2.4.2 优化问题求解	17
2.5 本章小节	18
<b>第三章 直线特征在 SLAM 系统中的应用</b>	<b>19</b>
3.1 图像中的线特征	19
3.1.1 线段特征的检测	19
3.1.2 线段特征的描述和匹配	20
3.2 空间直线的参数化	23
3.2.1 普吕克坐标	23
3.2.2 正交表示	24
3.3 直线观测模型	25
3.4 本章小结	27

<b>第四章 带有共面约束的点线 VIO 系统设计</b>	<b>29</b>
4.1 生成三维网格 . . . . .	29
4.1.1 从二维网格到三维网格 . . . . .	29
4.1.2 网格滤波 . . . . .	30
4.2 共面信息检测 . . . . .	31
4.2.1 水平平面检测 . . . . .	31
4.2.2 垂直平面检测 . . . . .	31
4.3 带有平面约束的系统构建 . . . . .	32
4.3.1 系统概述 . . . . .	32
4.3.2 优化问题建模 . . . . .	33
4.4 实验以及分析 . . . . .	34
4.4.1 仿真数据验证 . . . . .	35
4.4.2 EuRoC 数据集实验 . . . . .	36
4.5 本章小节 . . . . .	39
<b>第五章 共面点线的平面参数化</b>	<b>41</b>
5.1 空间共面信息找寻方法 . . . . .	41
5.1.1 平面实例分割 . . . . .	41
5.1.2 共面信息提取 . . . . .	42
5.2 共面点线参数化方法 . . . . .	44
5.2.1 共面点特征参数化 . . . . .	44
5.2.2 共面直线特征参数化 . . . . .	45
5.3 实验和分析 . . . . .	45
5.3.1 仿真实验 . . . . .	46
5.3.2 数据集验证以及分析 . . . . .	48
5.4 本章小节 . . . . .	52
<b>第六章 总结与展望</b>	<b>53</b>
6.1 总结 . . . . .	53
6.2 展望 . . . . .	53
<b>参考文献</b>	<b>55</b>
<b>攻读硕士期间发表论文</b>	<b>59</b>
<b>致谢</b>	<b>61</b>
<b>北京大学学位论文原创性声明和使用授权说明</b>	<b>63</b>

# 第一章 绪论

## 1.1 课题研究背景

近些年来，计算机技术以及人工智能技术的不断进步，移动机器人技术也得到了长期的发展并在很多领域得到了应用，出现了很多专业服务机器人替代人工去完成一些可编程实现的重复、危险的工作。与此同时就给移动机器人带来了很多的挑战，使得机器人必须拥有一定的感知以及智能能力才可以在尽量减少人为干预的情况下自主完成任务。

其中对环境的感知能力的基础部分有两个能力：一是对周围环境的感知建模能力；二是对自身位置的估计能力。这两个部分，在知道一个部分的内容的情况下并有传感器信息的时候，另外一个能力的部分是比较容易获得。但是实际情况下，机器人处于未知的环境中，移动机器人往往是不能得到周围环境信息的或者自身位姿的。所以就要求机器人必须具备在未知环境中同时对周围环境的建模能力和自身位置的定位能力，这就大大增加了移动机器人感知能力算法的要求。同时得到这两个问题就应对了即时定位和地图构建 (Simultaneous Localization and Mapping, SLAM)<sup>[1,2]</sup> 技术，SLAM 技术指的是在未知的环境中，利用自身传感器信息同时估计自身位置以及对周边地图进行建模的技术。

SLAM 技术可以使用各种各样传感器来实现，这些传感器可以按照测量信息相较于机器人自身来说分成两类：一是测量系统本体信息的传感器，比如轮速计，陀螺仪和加速度计；二是测量环境信息的传感器，比如摄像头，激光雷达等。这些不同的传感器，每种传感器都有自身的特性和适用的场景。拿激光雷达和深度相机这两个都可以获取周围环境的深度的传感器来举例。在室外无人车上一般就会使用激光雷达而不是使用深度相机，因为深度相机一般会受到环境中光照的影响比较大。而在室内环境中，由于光照环境比较简单，深度相机和激光雷达都能采集到信息。但是由于激光雷达获得的深度信息是扫描获得会受到运动的影响导致有畸变，所以室内场景一般会使用深度相机。同时不同的传感器获得的测量信息的信息量也是不同的，比如相机可以和人类的眼睛一样获得周围环境的颜色和几何等信息，而轮速计只能获得自身移动距离的信息。所以一般 SLAM 技术需要根据使用场景和要求不同来选择合适的传感器。如图1.1所示，不同领域的 SLAM 技术应用都采用了不同类型的传感器。

这些所有的传感器中，相机传感器是受到关注最多的传感器。因为作为传感器相机有如下几方面的优势：

- 1) 相机获取信息的形式和人眼获取信息的形式类似，获得的视觉信息包含的信息



图 1.1 SLAM 技术在不同领域的应用举例

非常丰富；

2) 基于 CCD 和 CMOS 等技术的相机传感器硬件经过这么多年的发展，已经有很多很成熟和廉价的相机硬件设备可以选择；

3) 和激光雷达和深度相机这些需要主动发射信号去感知周围环境信息的传感器相比，相机不需要主动发射信号就可以感知周围环境。

由于相机传感器的这些优势，基于相机的视觉 SLAM 技术的发展非常迅猛。但是单目视觉 SLAM 有个无法解决的问题就是不能获得绝对的尺度信息，使得 SLAM 系统在运行过程中会产生尺度漂移的问题<sup>[3]</sup>。而且只将视觉作为单一的传感器也会导致对视觉信息的依赖较大，在视觉信息不丰富或者失真的情况下就会导致系统估计不准确。为了应对这方面的问题，可以引入其他可以测量真实绝对尺度信息的传感器。一般来说引入惯性测量单元 (Inertial Measurement Unit, IMU) 是比较主流的方法。

而且目前的视觉 SLAM 技术本身也有很多内容还需要去研究。比如 SLAM 系统怎么在纹理特征不丰富的情况下可以长时间鲁棒的工作，或者如何结合深度学习技术感知环境中高阶语义信息。

本文主要关注于室内和结构化信息丰富的场景，所以本文主要关注如何把结构化信息更好的引入现有的视觉 SLAM 系统中。在视觉 SLAM 系统中引入结构化信息一方面可以提升 SLAM 系统定位的精度和鲁棒性，另一方面可以使得环境地图的表达拥有更多的结构化的表达。

## 1.2 研究现状

### 1.2.1 视觉 SLAM

视觉 SLAM 技术的一个标志性事件是 Davison 建立了首个可以实时运行的单目视觉 SLAM 系统—MonoSLAM<sup>[4,5]</sup>。研究人员随后便对视觉 SLAM 技术产生了极大的研究热情，并不断地推动这个邻域向前发展。因为视觉 SLAM 需要实时性的特点，现代视觉 SLAM 系统一般会人为的分割成两个部分：前端 (Front-end)、后端 (Back-end)。前端维持高频率的处理简单任务，后端以较低的频率对系统中的复杂问题进行求解。前端需要视觉信息进行一些预处理：对视觉信息进行提纯，提取和匹配一些视觉特征并做一些简单的位姿跟踪。后端则使用前端提纯的信息对整体系统自身的位姿和周围的环境信息进行估计。依照前后端中使用不同的方法类型，视觉 SLAM 能从多个角度进行分类。

1) 从使用的传感器类型的不同进行分类，可以分成单目、双目、多目以及 RGB-D 的视觉 SLAM。

2) 从视觉 SLAM 中使用图像特征匹配的方法不同可以分为

从视觉 SLAM 中对图像信息的使用方法的不同，则有直接法和间接法。直接法的意思是视觉 SLAM 前端不对图像特征进行描述和匹配的计算，直接追踪像素梯度。间接法的意思是视觉 SLAM 前端对图像特征进行描述和匹配，不同帧图像之间的特征关联在前端就预先求出。

间接法的视觉 SLAM：初期的视觉 SLAM 技术大多都是间接法的视觉 SLAM，在前端就直接建立不同帧特征之间的数据关联。间接法的视觉 SLAM 中比较常用的特征提取方法有：SIFT<sup>[6]</sup>、ORB<sup>[7]</sup> 以及 FAST<sup>[8]</sup>。匹配的方法通常有有建立描述子然后进行计算和直接进行 LK 光流<sup>[9]</sup> 两种。MonoSLAM 就是间接法的视觉 SLAM，它首先从图像中提取 Shi-Tomasi 特征<sup>[10]</sup>，然后使用图像块匹配求得不同帧之间的特征匹配。Klein 等人构建的 PTAM (Parallel Tracking and Mapping)<sup>[11]</sup> 方法也是间接法的视觉 SLAM，它采用的是 FAST 角点来提取图像信息再用图像块匹配算法计算不同帧之间的匹配。近些年大放异彩的 ORB-SLAM<sup>[12]</sup> 也是间接法的视觉 SLAM，它采用的图像特征是 ORB 特征。

直接法的视觉 SLAM：不同于间接法的视觉 SLAM，直接法不需要在前端计算不同帧图像之间的特征数据关联。直接法的视觉 SLAM 基础设定是对同一个物体和区域不同帧图像像素的灰度值是相同的。Newcombe 构建的 DTAM(Dense Tracking and Mapping)<sup>[13]</sup> 就是直接法的经典之作。DTAM 对图像上所有的像素都进行追踪，通过最小化光度误差的方法来求解位姿和地图。但是由于 DTAM 追踪所有像素，DTAM 只能在有 GPU 的设备上才能实时运行。在这之后 Engel 构建了能使用 CPU 运行的直



接法 SLAM 系统 LSD-SLAM(Large-scale Direct SLAM)<sup>[14]</sup>。LSD-SLAM 做出的改进是只追踪和关心图像上梯度较大的像素，这样虽然牺牲了一些地图点的重建，但是使得实时性得到了很大的改善。在 LSD-SLAM 的基础上，Engel 构建了 DSO(Direct Sparse Odometry)<sup>[15]</sup>，将相机的曝光时间、光晕等一系列影响像素灰度值的操作加入系统优化，使得直接法的 SLAM 摆脱了光度不变的假设，极大扩展了实用场景和鲁棒性。

直接法和间接法的视觉 SLAM 之间对比各有优劣。从地图重建的角度来说，直接法一般会构建出更加稠密的地图，如图1.2所示。从实时性的角度出发，在维护同样量级的地图信息的情况下，直接法 SLAM 的运行速度要更快，因为直接法 SLAM 省略了图像特征匹配的过程。但是从实用性角度来说，直接法的 SLAM 对相机的要求比较高，需要对相机的一些固有参数进行标定才能有较好的追踪效果，而间接法的视觉 SLAM 则没有这方面的要求。

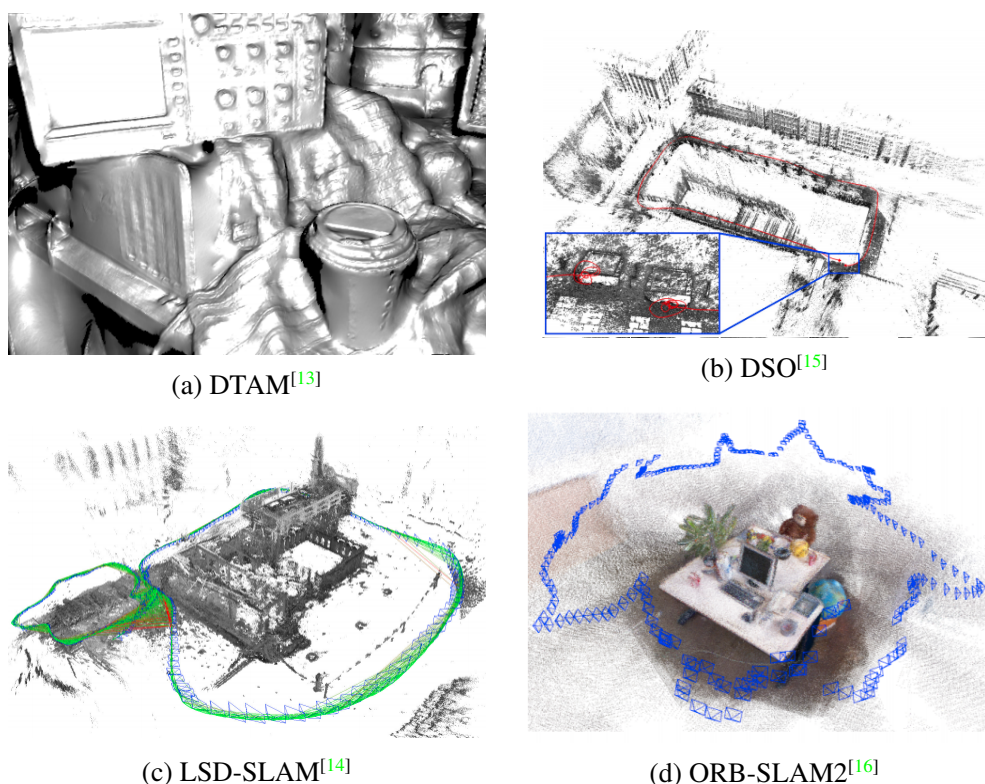


图 1.2 不同 SLAM 方法构建出的地图效果

3) 按照视觉 SLAM 后端优化求解使用的模型不同可以分为基于优化的方法和基于滤波的方法。

最初的视觉 SLAM 一般都是在后段采用滤波的方法来进行优化的,比如 MonoSLAM。是因为当时的硬件计算资源比较有限，并且初期的视觉 SLAM 算法在计算过程方面没有进行优化。首例将 BA 引入 SLAM 的是 Klein 构建的 PTAM。首先 PTAM 中将算法分成了两个独立的线程：追踪和建图。追踪线程实时的处理每一帧的图像数据，建图

线程则通过一些规则挑选一些关键帧融合关键帧上的数据进行优化。这也是现代视觉 SLAM 中前端和后端的雏形。

后来 Strasdat 等人证明：在同等计算资源的情况下，基于优化的方法可以比基于滤波的方法求解出更好的位姿和地图点<sup>[17]</sup>。从这之后基于优化的方法逐渐多了起来。

## 1.2.2 视觉和 IMU 融合的 SLAM

除了只包含视觉传感器的视觉 SLAM，将视觉信息和 IMU 信息融合的多传感器 SLAM 也是近些年来研究的热点问题。

IMU 数据和视觉数据进行融合之所以被广泛的研究是因为它们两者之间的数据可以互相作为补充互作补充：1. 视觉数据缺乏绝对的尺度信息，加上 IMU 的测量数据之后，就可以感知到决定的尺度信息；2. 在快速运动时视觉数据会变得模糊从而导致运动估计不准，但是 IMU 适合测量快速运动所以 IMU 数据可以在快速运动视觉信息不足时作为一个很好的补充；3. IMU 视距在长时间运行时会产生漂移，而视觉数据不会有这个问题，可以利用视觉数据消除 IMU 数据的漂移影响。所以视觉和 IMU 的信息融合的视觉惯性里程计（Visual Inertial Odometry, VIO）组合捕获了研究者的大量目光。

VIO 领域的研究可以通过视觉信息和 IMU 信息之间的耦合利用方法分成松耦合的方法和紧耦合的方法<sup>[18]</sup>。松耦合的 VIO 方法将视觉信息和 IMU 信息首先独立的进行求解系统中分别进行处理，处理成进一步的数据之后在一起联合估计。紧耦合的方法则是把视觉信息和 IMU 信息直接结合在一个求解系统中对所有状态量一起联合求解。Weiss 把 IMU 引入到原版的 PTAM 系统中，构建了一个融合的松耦合 VIO 系统为无人机使用<sup>[19, 20]</sup>。紧耦合系统的一个典型代表是 VINS-Mono<sup>[21]</sup>。松耦合的方法将不同传感器信息分开处理，分解了问题的规模使得问题求解效率更高。但是松耦合的方法本质上不能很好的利用各个传感器之间的互补效应，使得求解问题的精度不能达到最优，所以近些年的 VIO 算法都采用紧耦合的办法。

除了视觉和 IMU 信息之间的耦合关系，不同类型的视觉 SLAM 都可以和 IMU 信息结合。比如间接法的代表 ORB-SLAM 系统最新的 ORB-SLAM3<sup>[22]</sup> 就引入了 IMU 信息。直接法的代表 DSO 也有 IMU 的扩展半 VI-DSO<sup>[23]</sup>。MSCKF(Multi-State Constraint Kalman Filter)<sup>[24]</sup> 是基于滤波的方法的经典 VIO 代表。而前文提到的 VINS-mono 则是优化方法的 VIO 代表。

在 VIO 发展历史上有有个标志性的事件是 Foster 等人发展了 IMU 的预积分方法<sup>[25, 26]</sup>。这使得高频率的 IMU 信息可以和低频率的图像信息可以在保持在同一频率上进行优化，同时也使得在优化利用视觉信息估计 IMU 测量的偏移量提供了便利。

### 1.2.3 结构化 SLAM

对于结构化信息，研究者在 SLAM 的底层技术已经发展到了一定阶段之后都开始关注于将这些高级特征加入 SLAM 系统中来。

对于直线特征在视觉 SLAM 中的利用。J.Neira 在 1997 年就构建了结合垂直直线段的视觉 SLAM 系统<sup>[27]</sup>。但是早期由于空间直线的参数化问题，直线即使加入了 SLAM 系统中也不能得到很好的利用。Sola 对直线特征参数化进行了详尽的分析和实验验证各自的优势<sup>[28]</sup>。Bartoli 为空间直线提出了正交表示<sup>[29]</sup>这种空间直线的最小表示，使得空间直线的参数可以在后端无约束的进行优化。随后直线在 SLAM 中的利用就变得方便了起来。Zhang 将直线利用普吕克坐标<sup>MVG</sup>和正交表示来在同一个系统中参数化空间直线，构建了一个带有点线特征的双目 SLAM 系统<sup>[30]</sup>。He 在 PL-VIO<sup>[31]</sup>中也沿用了同样的参数化方式，将直线加入了带有优化的 VIO 中。除了直线参数化，Yang 对空间直线特征的三角化进行了详尽的分析和提出了一种可以融合多帧观测的直线三角化方案<sup>[32]</sup>。

除了直线特征，平面特征也受到研究者的关注。和直线特征类似，平面特征想要很好的被利用起来，参数化同样是第一个需要被考虑的问题。Lee<sup>[33]</sup>使用球坐标来参数化平面。球坐标虽然是平面的最小表示，但是奇异性也是它不可弥补的缺陷。另外一个常见的方法是用平面的法向量和距离四个参数来表达平面。Trevor<sup>[34]</sup>就使用了这种过参数化的表达方式在它的 SLAM 系统中参数化平面。但是这种过参数化表达方式不能直接在优化中使用无约束的优化，Kaess<sup>[35]</sup>提出了使用四元数的方法来丢平面进行了最小表示的参数化。

除了直线特征和平面特征的单独使用，Lu 构建了一个融合点线面特征的多特征视觉 SLAM 系统<sup>[36]</sup>，这个系统可以检测到空间中的直线和平面特征，同时可以将它们以及它们之间的共线共面约束一起联合优化。

## 1.3 本文研究内容

本文主要研究了在结构化的场景下，将直线特征和空间点线特征的共面约束添加入视觉 SLAM 系统中，使得 SLAM 系统在结构化场景下变得更加鲁棒。本文研究内容主要分为两个部分：

- 1) 设计了一个融合共面信息的点线 VIO 系统，同时该系统可以实时的生成一个半稠密的三维网格地图。系统中加入直线和共面这类几何特征使得系统的定位精度更好，半稠密的地图也可以给机器人实时避障使用。

- 2) 对于空间中共面的点线特征设计一种新型的参数化方法。对于共面信息在系统中的加入，不仅仅可以在优化的后端中直接加入，还可以直接在参数化的层级直接加



入系统。共面特征的参数化实现也可以使得优化后端的优化项减少，使得优化速度变快。

## 1.4 本文结构

本文各章内容安排如下：

第一章是绪论部分。介绍了视觉 SLAM 以及结构化特征在 SLAM 研究中的研究背景和意义，梳理了点线面 SLAM 的发展以及研究现状，最后介绍了全文的安排以及主要内容。

第二章介绍了基于优化的视觉 SLAM 和 VIO 的一些背景知识，包括了单目和双目相机模型，刚体的三维运动变换模型，IMU 的测量模型，预积分的基本原理以及基于优化的 SLAM 方法和基本原理。

第三章介绍了如何将直线特征应用于基于优化的视觉 SLAM 系统中。首先介绍了如何在图像中提取和匹配线段，然后介绍了空间直线的参数化，测量模型。

第四章构建了一个结合平面约束的点线 VIO 系统。该系统首先构建了一个三维网格地图，然后从网格地图中提取到平面以及平面上的共面特征，最后在滑动窗口中联合 IMU 测量，空间点线特征以及它们的共面约束所有信息优化所有状态量。本章最后构建了仿真实验和真实数据集实验来验证提出系统的有效性。

第五章提出了共面点线的平面参数化方法。首先提出了一种基于深度学习平面实例分割结果的平面信息提取方法，然后提出了如何省略共面的点线的空间参数化使用平面信息来间接参数化。最后在优化中直接使用投影误差就可以投建共面约束。本章最后也构建了仿真实验和真实数据集实验来验证提出参数化方法的有效性。

第六章总结了全文的所有工作的内容并对本文的提出的系统的一些未来可以扩展研究的方向进行了介绍。



## 第二章 相关背景知识

本章主要内容是介绍一些基于优化的视觉 SLAM 和耦合 IMU 的 VIO 的一些相关基础知识。

### 2.1 相机模型

相机是视觉 SLAM 来获取场景中信息的传感器，所有场景的光信息经过不同的相机硬件处理然后得到了场景的二维图像信息。二维图像信息包含了场景中几何、亮度、颜色以及场景中物体位置等多种信息。相机模型是表达三维世界的信息是怎么变换到二维图像中的，这个从三维到二维的变换过程被称为投影。而从二维图像上的信息加上深度信息再回到三维真实世界的过程则是反投影。

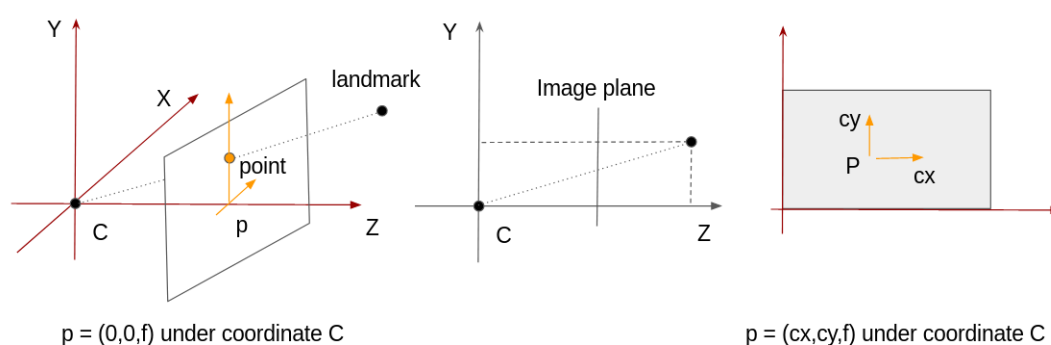


图 2.1 针孔相机模型

相机的投影模型多种多样，其中被普遍使用的是针孔相机模型。针孔相机模型是原理是小孔成像，它的过程是光线经过物体反射后穿过一点后在后方的投影面上投影成像。如图2.1所示，C 是相机模型的光心， $f$  是投影模型的焦距，平面  $z = f$  被是成像平面。

在空间中的路标 (landmark) 点  $(X, Y, Z)^T$  沿着光心方向在成像平面上的投影点 (point) 的坐标是  $(u, v)^T$ ：

$$u = \frac{f_x X}{Z} + c_x \quad (2.1)$$

$$v = \frac{f_y Y}{Z} + c_y \quad (2.2)$$

上式中,  $f_x$  和  $f_y$  分别是相机模型在  $x$  方向和  $y$  方向上焦距,  $c_x$  和  $c_y$  是成像平面的中心在像素单位意义的位置。为了阐述相机模型的主要原理, 本节内容没有将畸变加入。

公式2.1和公式2.2可以合并成矩阵的形式:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} \quad (2.3)$$

矩阵  $\mathbf{K}$  的内部的值的含义是摄像头的内参, 是不同摄像头成像时的不同数值, 可以通过离线的方法标定得到。得到内参矩阵中的参数后可以作为固定的参数也可以作为变化的参数在 SLAM 系统是实时优化。如果公式2.3中的三维特征点  $[X, Y, Z]^T$  不是以相机坐标系为基坐标系的情况下, 那么三维坐标点必须经过一个三维坐标变换后变成以相机坐标系为基坐标系的三维表示才能投影。所以通常情况下, 三维坐标得经过一个旋转和平移的三维坐标变换, 如下式:

$$s\mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_w \quad (2.4)$$

其中,  $\mathbf{X}_w$  是三维坐标点在世界坐标系下表示的坐标,  $\mathbf{R}_{cw}$  和  $\mathbf{t}_{cw}$  分别表示从世界坐标系下到相机坐标系下的旋转矩阵和旋转向量。

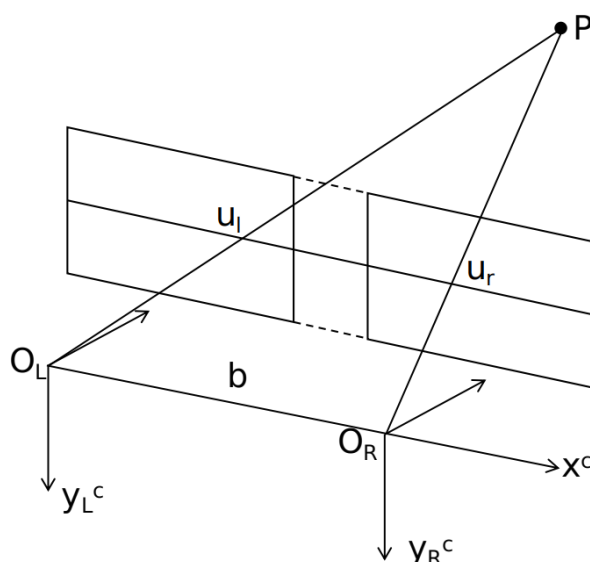


图 2.2 双目相机投影模型

双目相机的成像模型则是通过增加另一个相机形成另一个观测结果。双目相机相较于单目相机的优势是可以通过双目匹配特征得到特征之间的像素坐标差, 然后得到

特征的估计深度值。两个成像平面重合的相同针孔相机模型就可以组成一个双目相机模型，如图2.2所示。

双目相机不同光心间的位移代表基线  $b$ 。一般双目相机的左目相机的光心作为双目相机的坐标原点。在双目视觉里，可以通过极线找到左右目相机图像中对应的特征点的左右目数据关联。如图2.2的右图所示，因为双目相机的成像平面重合，所以左目的特征点  $J_L$  一定可以在右目的极线上找到与之对应匹配的特征点  $J_R$ 。在左右目的极线上寻找匹配的方法有很多，比如 SAD 和 NCC<sup>[37]</sup>。左右目对应特征点的  $x$  方向的像素差被叫做视差，代表了三维空间点距离相机的距离。

由于双目相机的特殊性，找到特征点左右目中对应的匹配关系，就可以计算出特征在相机坐标系下的三维表示。如果已知左右目的对应匹配点  $P_L = [u_l, v]^T$  和  $P_R = [u_r, v]^T$ ，则通过双目相机的基线  $b$  可以求得：

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{b(u_l - c_x)}{u_l - u_r} \\ \frac{b(v - c_x)}{u_l - u_r} \\ \frac{bf}{u_l - u_r} \end{bmatrix} \quad (2.5)$$

一般情况下，双目相机的左右相机之间也会存在一定的三维坐标变换，这被称为双目相机之间的外参。和相机的内参一样，这个外参也可以通过离线标定的方法求得，在系统运行时，既可以选择作为固定参数也可以作为变量进行实时优化。

## 2.2 位姿变换

SLAM 系统要解决的两个问题，是本质可以理解为同时处理相机的姿态求解和维护的地图中的路标点的位置的求解。本部分是为了介绍相机代表刚体在真实世界中运动的位姿表示，和在不同坐标系下的坐标之间的关系。

### 2.2.1 坐标系定义

想要描述待估计物体的位姿，一般需要设定一个全局固定的参考坐标系来说明物体的绝对位态。这个参考坐标系可以被设定成大地，同时用  $\mathbf{W}$  表示。物体自身的坐标一般可以表示为  $\mathbf{B}$ 。除此之外，相机也有一个坐标系表示为  $\mathbf{C}$ 。

### 2.2.2 位姿定义

刚体在三维世界中的运动表示由平移和旋转组成，其中平移可以用一个三维向量表示  $t \in \mathbb{R}^3$ ，旋转的表示则并不如此方便。正常情况下，可以用三维的单位正交矩阵  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  来描述。但是三维世界中的旋转有三个自由度，而上述表达却用了九个参数，

所以旋转矩阵不能做无约束的优化，所以还需要一些其他形式的旋转表达。四元数就是这么一种可以用来表示旋转的紧凑表达。下面本节将分别介绍这两种不同的旋转表达形式。

旋转矩阵本质上是一个李群<sup>[38]</sup>，是三维特殊正交群  $SO(3)$  (Special Orthogonal Group):

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\} \quad (2.6)$$

还有:

$$\mathbf{R}^{-1} = \mathbf{R}^T \quad (2.7)$$

四元数由一位数的实部和三位数的虚部组成:

$$\mathbf{q} = q_w + q_x i + q_y j + q_z k \quad (2.8)$$

把虚部的三个量写成向量的形式，四元数也可以写成

$$\mathbf{q} = \begin{bmatrix} q_w \\ q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \quad (2.9)$$

四元数的虚部元素还满足如下的关系:

$$\begin{aligned} ij &= -ji = k \\ jk &= -kj = i \\ ki &= -ik = j \\ i^2 &= j^2 = k^2 = -1 \end{aligned} \quad (2.10)$$

单位四元数可以用来表示三维空间中的旋转，假设待表示的旋转是围绕单位向量  $n = [n_x, n_y, n_z]^T$  旋转了  $\theta$ ，这个单位四元数的值如下:

$$\mathbf{q} = [\cos \frac{\theta}{2}, n_x \sin \frac{\theta}{2}, n_y \sin \frac{\theta}{2}, n_z \sin \frac{\theta}{2}]^T \quad (2.11)$$

同样能从单位四元数得到原旋转向量和旋转角度的大小:

$$\begin{aligned} \theta &= 2 \arccos q_w \\ [n_x, n_y, n_z]^T &= [q_x, q_y, q_z]^T / \sin \frac{\theta}{2} \end{aligned} \quad (2.12)$$

四元数也有和四元数本身的一系列计算规则，本节简单介绍四元数的加法和乘法：四元数的加法法则如下：

$$\mathbf{p} + \mathbf{q} = \begin{bmatrix} p_w \\ \mathbf{p}_v \end{bmatrix} + \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_w + q_w \\ \mathbf{p}_v + \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_w + q_w \\ p_x + q_x \\ p_y + q_y \\ p_z + q_z \end{bmatrix} \quad (2.13)$$

四元数的乘法规则如下：

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \quad (2.14)$$

在介绍完这两种不同的旋转表达之后，再回到位姿的表达上。对于一个空间坐标点  $\mathbf{P}$ ， $\mathbf{P}$  在世界坐标系  $\mathbf{W}$  和相机坐标系  $\mathbf{C}$  下的表示是  $\mathbf{P}_w$  和  $\mathbf{P}_c$ ，那么可以得到

$$\mathbf{P}_c = \mathbf{R}_{cw} * \mathbf{P}_w + \mathbf{t}_{cw} \quad (2.15)$$

其中  $\mathbf{R}_{cw}$  以及  $\mathbf{t}_{cw}$  分别表示相机坐标系到世界坐标系的旋转和平移。

上式写成矩阵形式如下：

$$\begin{bmatrix} \mathbf{P}_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{P}_w \\ 1 \end{bmatrix} \quad (2.16)$$

其中  $\begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$  可以写成  $\mathbf{T}_{cw}$ 。被认为是由世界参考系到摄像头参考系的三维位姿变换。

### 2.3 IMU 测量以及预积分

一般来说 IMU 测量速度远超相机图像的测量速度，所以两者数据需要进行一些处理才能很好在一个系统中工作。通常 IMU 预积分<sup>[25, 26]</sup> 是不错的预理解决办法。IMU 预积分的思想是把两帧图像帧之间的 IMU 测量数据融合成一个预积分的测量项，如此 IMU 测量和图像测量就达到了同样的测量频率。下面将介绍 IMU 的测量模型以及与积分的基本定义。

### 2.3.1 IMU 测量模型

依照<sup>[25, 26]</sup>中的采用的测量模型, IMU 的数据由两部分组成: 一是陀螺仪提供的角速度  $\hat{\omega}_B \in \mathbb{R}^3$ 、二是加速度计提供的加速度  $\hat{a}_B \in \mathbb{R}^3$  :。

$$\begin{aligned}\hat{\omega}_B &= \omega_B + b_g + \eta_g \\ \hat{a}_B &= (a_B - g_B) + b_a + \eta_a = \mathbf{R}_{BW} (a_W - g_W) + b_a + \eta_a\end{aligned}\quad (2.17)$$

上式中的坐标下标 **B** 和 **W** 分别是体参考系和世界参考系。 $b_a$  以及  $b_g$  是陀螺仪和加速度计的零偏 (bias)。 $\eta_a$  以及  $\eta_g$  是陀螺仪和加速度计的测量高斯噪声。 $\omega_B$  代表体参考系下的角速度,  $\mathbf{R}_{BW}$  代表体坐标和世界坐标之间的旋转,  $a_W$  代表体坐标在世界坐标系下的加速度,  $g_W$  是全局的重力加速度。

依照刚体运动的模型<sup>[39]</sup>, 体坐标 **B** 和世界坐标 **W** 中的旋转、速度以及位移满足下列微分方程:

$$\begin{aligned}\dot{\mathbf{R}}_{WB} &= \mathbf{R}_{WB} \cdot \hat{\omega}_B \\ \dot{v}_{WB} &= a_W \\ \dot{p}_{WB} &= v_{WB}\end{aligned}\quad (2.18)$$

IMU 的采样间隔为  $\Delta t$ , 则有:

$$\begin{aligned}\mathbf{R}_{WB}(t + \Delta t) &= \mathbf{R}_{WB}(t) \cdot \text{Exp}\left(\int_t^{t+\Delta t} \hat{\omega}_B(\tau) d\tau\right) \\ v_{WB}(t + \Delta t) &= v_{WB}(t) + \int_t^{t+\Delta t} a_W(\tau) d\tau \\ p_{WB}(t + \Delta t) &= p_{WB}(t) + \int_t^{t+\Delta t} v_{WB}(\tau) d\tau + \iint_t^{t+\Delta t} a_W(\tau) d\tau^2\end{aligned}\quad (2.19)$$

上式中  $\text{Exp}(\cdot)$  代表的是  $\text{SO}(3)$  的指数映射。

如果认为 IMU 采样时间内的角速度和加速度是恒定值, 那么则有

$$\begin{aligned}\mathbf{R}_{WB}(t + \Delta t) &= \mathbf{R}_{WB}(t) \cdot \text{Exp}\left(\left(\hat{\omega}_B(t) - b_g(t) - \eta_{gd}(t)\right) \Delta t\right) \\ v_{WB}(t + \Delta t) &= v_{WB}(t) + g_W \Delta t + \mathbf{R}_{WB}(t) \cdot \left(\hat{a}_B(t) - b_a(t) - \eta_{ad}(t)\right) \Delta t \\ p_{WB}(t + \Delta t) &= p_{WB}(t) + v_{WB}(t) \Delta t + \frac{1}{2} g_W \Delta t^2 \\ &\quad + \frac{1}{2} \mathbf{R}_{WB}(t) \cdot \left(\hat{a}_B(t) - b_a(t) - \eta_{ad}(t)\right) \Delta t^2\end{aligned}\quad (2.20)$$



## 2.3.2 IMU 预积分

公式2.3.1描述了一个采样周期前后两帧 IMU 测量值间的相对位移、速度以及旋转。可是如图2.3所示，IMU 的测量数据相对于相机的测量数据来说太高了。所以想要很好的利用好 IMU 测量，Lupton 等人提出了预积分的思想<sup>[26]</sup>，把两个图像关键帧之间的 IMU 测量值都累加到同一坐标系中，然后和图像数据保持同样的频率。

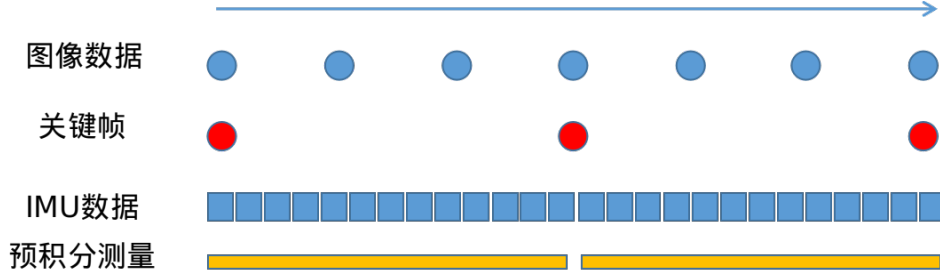


图 2.3 相机和 IMU 的采样示意图

所以根据上文已知的测量模型，将  $i$  时刻和  $j$  时刻之间的 IMU 数据累加起来可以得到：

$$\begin{aligned}
 \mathbf{R}_j &= \mathbf{R}_i \cdot \prod_{k=i}^{j-1} \text{Exp} \left( \left( \hat{\omega}_k - \mathbf{b}_{gk} - \boldsymbol{\eta}_{gd_k} \right) \Delta t \right) \\
 \mathbf{v}_j &= \mathbf{v}_i + \mathbf{g} \Delta t_{ij} + \sum_{k=i}^{j-1} \mathbf{R}_k \left( \hat{\mathbf{a}}_k - \mathbf{b}_{ak} - \boldsymbol{\eta}_{ad_k} \right) \Delta t \\
 \mathbf{p}_j &= \mathbf{p}_i + \sum_{k=i}^{j-1} \left[ \mathbf{v}_k \Delta t + \frac{1}{2} \mathbf{g} \Delta t^2 + \frac{1}{2} \mathbf{R}_k \left( \hat{\mathbf{a}}_k - \mathbf{b}_{ak} - \boldsymbol{\eta}_{ad_k} \right) \Delta t^2 \right] \\
 &= \mathbf{p}_i + \mathbf{v}_i \Delta t_{ij} + \frac{1}{2} \sum_{k=i}^{j-1} \mathbf{g} \Delta t^2 + \sum_{k=i}^{j-1} \frac{1}{2} \mathbf{R}_k \left( \hat{\mathbf{a}}_k - \mathbf{b}_{ak} - \boldsymbol{\eta}_{ad_k} \right) \Delta t^2
 \end{aligned} \tag{2.21}$$

按照 Lupton 等人把 IMU 预积分测量值放在第一帧下，将上式变换可以得到 IMU 测量项如下：

$$\begin{aligned}
 \Delta \mathbf{R}_{ij} &\doteq \mathbf{R}_i^T \mathbf{R}_j = \prod_{k=i}^{j-1} \text{Exp} \left( \left( \hat{\omega}_k - \mathbf{b}_{gk} - \boldsymbol{\eta}_{gd_k} \right) \Delta t \right) \\
 \Delta \mathbf{v}_{ij} &\doteq \mathbf{R}_i^T \left( \mathbf{v}_j - \mathbf{v}_i - \mathbf{g} \Delta t_{ij} \right) = \sum_{k=i}^{j-1} \Delta \mathbf{R}_{ik} \left( \hat{\mathbf{a}}_k - \mathbf{b}_{ak} - \boldsymbol{\eta}_{ad_k} \right) \Delta t \\
 \Delta \mathbf{p}_{ij} &\doteq \mathbf{R}_i^T \left( \mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2 \right) \\
 &= \sum_{k=i}^{j-1} \left[ \Delta \mathbf{v}_{ik} \Delta t + \frac{1}{2} \Delta \mathbf{R}_{ik} \left( \hat{\mathbf{a}}_k - \mathbf{b}_{ak} - \boldsymbol{\eta}_{ad_k} \right) \Delta t^2 \right]
 \end{aligned} \tag{2.22}$$

最后 IMU 预积分模型还包括对零偏的矫正，对测量值的方差分析。本文在此不赘述此方面知识。

## 2.4 基于优化的 SLAM 方法

现代 SLAM 方法一般都会分为两个部分：前端和后端。前端的任务是对传感器的原始数据进行处理。对于图像来说，处理一般是提取出图像上的特征并计算特征之间的数据关联。后端的任务是使用前端处理过的传感器数据来计算出当前系统的位姿以及周围环境的地图。后端的求解方法不同又可以把 SLAM 系统分为基于滤波的方法和基于优化的方法，因为 Strasdat 论证了在一样的资源情况下，优化的方法比滤波的方法得到的精度更高<sup>[17]</sup>，所以现代主流 SLAM 相当一部分通过非线性优化来求解问题。本节会简单介绍基于优化的 SLAM 方法。

### 2.4.1 最大后验概率估计

SLAM 方法中后端要求解的问题是，使用带有噪声的观测数据，求解出所以待估计的量。这些待估计的量包括自身的位姿和 SLAM 地图中的三维特征。也就是一个最大后验概率估计 (Maximun-a-Posteriori,MAP) 问题。

假设 SLAM 问题中的所有位姿变量是  $X = \{x_t\}$ ，地图中的路标变量是  $L = \{l_i\}$ ，传感器提供的测量数据是  $Z = \{z_k\}$ 。其中  $X$  以及  $L$  是待估计变量，两者的集合是  $\mathcal{X}$ 。根据上述信息则有：

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmax}} P(\mathcal{X} | \mathbf{Z}) \quad (2.23)$$

因为  $\mathcal{X}$  的分布一般都是已知的，，所以由贝叶斯法则可知：

$$P(\mathcal{X} | \mathbf{Z}) = \frac{P(\mathbf{Z} | \mathcal{X})P(\mathcal{X})}{P(\mathbf{Z})} \propto P(\mathbf{Z} | \mathcal{X})P(\mathcal{X}) \quad (2.24)$$

上式中左侧一般被叫做后验概率，后侧  $P(\mathbf{Z} | \mathcal{X})$  一般被叫做似然， $P(\mathcal{X})$  一般被叫做先验。一般情况下直接计算后验是不方便的，在 SLAM 中一般会计算最大似然估计问题：

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmax}} P(\mathbf{Z} | \mathcal{X}) \quad (2.25)$$

假设观测数据服从如下模型

$$z_k = h_k(x_k) + \epsilon_k \quad (2.26)$$

其中  $\epsilon_k$  服从高斯分布，那么就有

$$P(\mathbf{Z}_k | \mathbf{X}_k) \propto \exp\left(-\frac{1}{2} \|h_k(\mathbf{X}_k) - \mathbf{z}_k\|_{\Omega_k}^2\right) \quad (2.27)$$

一般会认为观测数据之间是相互独立的，对公式2.4.1求负对数，则有：

$$\begin{aligned} \mathbf{X}^* &= \underset{\mathbf{x}}{\operatorname{argmin}} -\log(\Pi P(\mathbf{Z}_k | \mathbf{X}_k)) \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} \sum \frac{1}{2} \|h_k(\mathbf{X}_k) - \mathbf{z}_k\|_{\Omega_k}^2 \end{aligned} \quad (2.28)$$

如此便得到了一个无约束的最小二成问题，这个问题的最优解就是求解量的最大似然估计。数学中图模型可以很好描述上述问题，常见的图模型有贝叶斯网络和因子图。SLAM 中常用的是因子图来表示。如图2.4所示，圆圈表示待估计的状态量，正方形框表示观测或者先验。

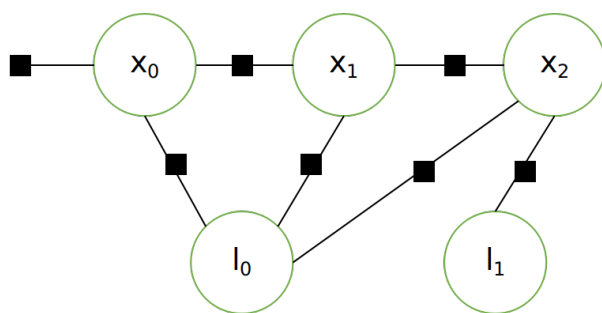


图 2.4 因子图模型

## 2.4.2 优化问题求解

上一节引出的最小二乘问题一般需要通过非线性优化的方式来求解。常用的求解器有高斯牛顿和 LM 方法。这些求解器的中心思想是把非线性问题在局部进行线性化，给当前待估计值  $\hat{\mathcal{X}}$  增加一个微小增量  $\delta\mathcal{X}$ ，使得整体能量函数  $E(\hat{\mathcal{X}} + \delta\mathcal{X}) < E(\hat{\mathcal{X}})$ ，最后通过这种方法找到一个极值点。

用高斯牛顿法来说明，待估计量的初值是  $\hat{\mathcal{X}}$ ，然后在这个初值附近线性近似，则有

$$\begin{aligned}
 E(X) &= (z - f(X))^T \Sigma^{-1} (z - f(X)) \\
 &= e(X)^T \Sigma^{-1} e(X) \\
 &= e(\hat{X} + \delta X)^T \Sigma^{-1} e(\hat{X} + \delta X) \\
 &\approx \left( e(\hat{X}) + \left. \frac{\partial e}{\partial X} \right|_{X=\hat{X}} \cdot (X - \hat{X}) \right)^T \Sigma^{-1} \left( e(\hat{X}) + \left. \frac{\partial e}{\partial X} \right|_{X=\hat{X}} \cdot (X - \hat{X}) \right) \\
 &\doteq (e(\hat{X}) + J\delta X)^T \Sigma^{-1} (e(\hat{X}) + J\delta X)
 \end{aligned} \tag{2.29}$$

上式中  $\hat{X}$  处残差  $e$  关于状态量  $X$  的雅克比是  $J \doteq \left. \frac{\partial e}{\partial X} \right|_{X=\hat{X}}$ ,  $\delta X$  是微小量。通常观测量的数量超过待估计状态量的数量。由梯度为零通过正规方程可以知道：

$$J^T \Sigma^{-1} J \cdot \delta X^* = -J^T \Sigma^{-1} e \tag{2.30}$$

经过上式得到微小扰动的最优解  $\delta X^*$ , 然后一直更新再重复线性化求解  $\hat{X} \leftarrow \hat{X} + \delta X^*$ 。最终就可以得到想要的极值点, 这就是高斯牛顿法的主要过程。

## 2.5 本章小节

本章主要介绍了本文涉及到的一些基础知识。2.1节介绍了单目和双目相机的观测模型。2.2介绍了 SLAM 问题基本的坐标系定义以及位姿的参数化。2.3节介绍了 IMU 单元的测量模型以及怎么用预积分来使得 IMU 测量和图像测量可以保持在同一频率并可以优化。2.4节介绍了基于优化的 SLAM 的基本原理, 并主要讲述了用最大后验估计来描述 SLAM 问题和形成最大后验估计问题后怎么用优化的方法去求解问题。

## 第三章 直线特征在 SLAM 系统中的应用

在图像中，点特征和线段特征描述了两种不同的信息。点特征在图像中比较有代表性的像素点，通常是物体的角点。而线段特征在图像中比较有代表性的像素条，通常是物体的边缘。这两种不同的图像信息在不同的场景中的不同图像的丰富程度是不一样的。对于本文主要研究的室内场景来说，由于室内场景都是人工构建的，会有很多明显的结构化的设计，所以直线特征是非常丰富的。如图3.1所示，点特征只是构建了环境的稀疏点云，只能大概分辨出场景的轮廓。但是由直线特征构建的地图则可以很好描绘出整个环境的场景轮廓信息。除了可以提供结构信息以外，相较于点特征，图像上的线段特征可以在光照和视角等方面提供更好的鲁棒性。接下来本章将介绍如何将直线特征引入视觉 SLAM 系统中。

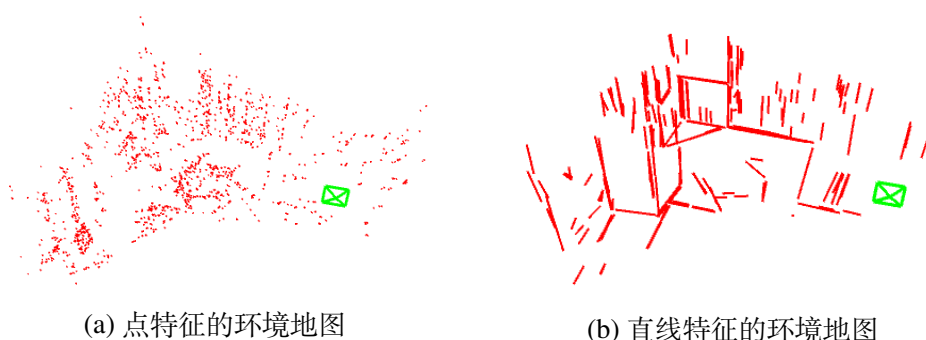


图 3.1 用点特征和直线特征构建的环境地图

### 3.1 图像中的线特征

和利用空间中点特征相同，想要利用空间中的直线特征，首先需要从相机获取的图像上提取和匹配线特征。下面将介绍如何在图像中进行线特征的提取和匹配。

#### 3.1.1 线段特征的检测

从图像中提取线特征一般采用 Canny 这类边缘提取子先得到边缘信息，然后使用霍夫变换来得到直线，最终使用一些手段知道直线的端点信息。但是这种方法来获得图像中线特征耗时很高，还会在图像边缘丰富的地方出现错误检测，所以这种方法不适合于需要实时性的 SLAM 系统。

LSD (Line Segment Detector) 是可以在较短的时间中提取出亚像素级别的局部线段检测算法<sup>[40]</sup>，该算法的中心思想是把图像中梯度方向类似的像素聚类。



图 3.2 LSD 中线段检测过程<sup>[40]</sup>

LSD 算法提取图像中线特征的流程如下：

1、对图像进行降采样和高斯滤波，消弱图像噪点对检测结果的影响。然后计算图像每个像素的梯度大小和方向，使得所有像素点形成一个梯度场，如图3.2中间图所示。

2、将图像中的梯度场方向差不多的邻近像素连通，得到连通域，这些不同的连通域就是线段的支撑域，如图3.2右图所示。

3、最后从图像中所有的线段支撑域中获得最终的线段检测结果。

LSD 检测算法的好处是检测速度快而且不用进行过多的参数设置，同时检测出的线特征带有方向信息可以在匹配中快速的筛去不符合的待匹配线段和错误匹配线段。图3.3是 LSD 检测算法的检测结果，可以看出 LSD 检测算法很好的提取出了图像中的线段信息。

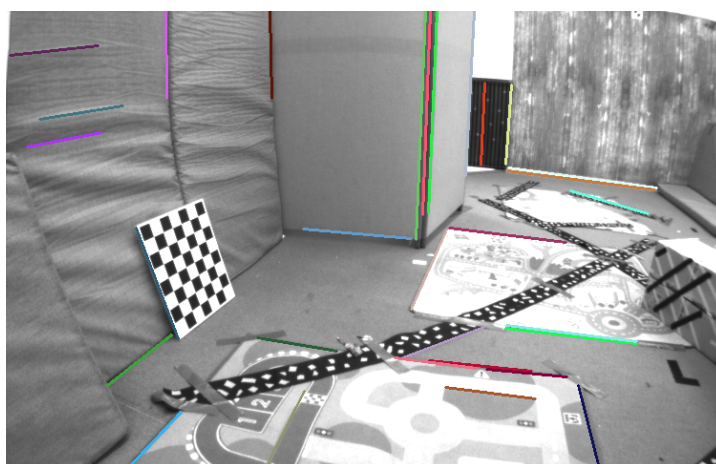


图 3.3 LSD 检测结果

### 3.1.2 线段特征的描述和匹配

和利用点特征类似，完成了线特征提取之后，要得不同图像帧之间的线段的数据关联则需要对线段特征进行描述，然后计算匹配距离完成匹配。但是线段特征的匹配一直是比较难解决的问题，主要原因如下：

1) 由于遮挡的原因，不同视角下观测同一空间直线不能稳定的看到一样的线段端点。

2) 线段的提取往往不是完整的，经常会发生只能提取图像上一条完整直线的一部分线段的情况。

3) 相较于点特征匹配严格的极线约束来说，直线特征并没有图像平面上强制的几何约束。

4) 在低纹理信息不丰富的区域，不同的二维图像线段之间特征区分度过小。

不过虽然有着这些困难，研究者们还是提出了一些比较鲁棒的描述方法。Wang 等人提出了 MSLD (mean-standard deviation line descriptor) 描述子<sup>[41]</sup>，该描述子有抗光照变化和抗旋转等好处。MSLD 描述子和点特征的 SIFT 描述子部分有着相似的思想，MSLD 描述子首先把一条完整线段变成多个小的子区域，然后根据子区域内的像素值的梯度和标准差来描述这条线段。Zhang 等人在 MSLD 描述子之后又进一步提出了 LBD 描述子<sup>[42]</sup>，LBD 描述子由于加上线段全局以及局部的高斯权重系统，相对于 MSLD 描述子，有着更快的描述效率和更好的描述能力。下面将介绍 LBD 描述子描述线段特征的主要流程。

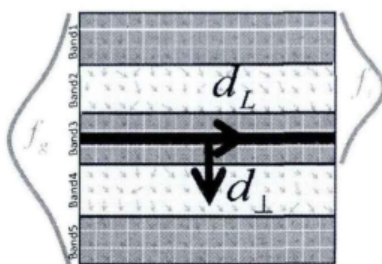


图 3.4 LBD 特征描述中的线段支撑域

最开始，LBD 描述子是在图像的待描述线段上生成一个矩形框，这个框就是线段的支撑域 LSR (Line Support Region)，同时生成了线段的方向向量  $d_L$  和线段的垂直方向向量  $d_{\perp}$ 。如图3.4所示， $m$  表示线段支撑域的条带数量， $w$  表示条带的像素宽度。

之后使用全局的高斯函数  $f_g$  对 LSR 的每一列进行高斯模糊操作，减少远离线段的像素梯度对描述子的影响。使用局部高斯函数  $f_l$  对 LSR 中相邻条带的每一列，减少不同条带的边界效应。

接着就可以利用相邻条带的梯度值数据求得每个条带  $B_j$  对应的特征向量  $BD_j$ ，再把所有不同条带的  $BD_j$  整合到一起，就得到了完整的 LBD 描述：

$$\text{LBD} = (BD_1^T, BD_2^T \dots BD_m^T)^T \quad (3.1)$$

之后把  $B_j$  和与之邻近的  $B_{j-1}$ 、 $B_{j+1}$  的每行的局部的梯度  $\mathbf{g}' = (\mathbf{g}'_{d_{\perp}}, \mathbf{g}'_{d_L})$  分开相加。



其中具体的第  $k$  行，则：

$$\begin{aligned}
 v1_j^k &= \lambda \sum_{g'_{d_{\perp}} > 0} g'_{d_{\perp}} \\
 v2_j^k &= \lambda \sum_{g'_{d_{\perp}} < 0} g'_{d_{\perp}} \\
 v3_j^k &= \lambda \sum_{g'_{d_L} > 0} g'_{d_L} \\
 v4_j^k &= \lambda \sum_{g'_{d_L} < 0} g'_{d_L}
 \end{aligned} \tag{3.2}$$

上式中， $\lambda = f_g(\mathbf{k})f_l(\mathbf{k})$  是高斯系数，接着把每行的和堆在一个矩阵中，就得到了特征向量  $\mathbf{BD}_j$  的特征描述矩阵  $BDM_j$ ：

$$\begin{aligned}
 BDM_j &= \begin{pmatrix} v1_j^1 & v1_j^2 & \dots & v1_j^n \\ v2_j^1 & v2_j^2 & \dots & v2_j^n \\ v3_j^1 & v3_j^2 & \dots & v3_j^n \\ v4_j^1 & v4_j^2 & \dots & v4_j^n \end{pmatrix} \in \mathbb{R}^{4 \times n} \\
 n &= \begin{cases} 2w, j = 1 \text{ or } m \\ 3w, \text{ else} \end{cases}
 \end{aligned} \tag{3.3}$$

最后求出特征描述矩阵  $BDM_j$  的均值方差  $M_j^T$  以及均值向量  $S_j^T$ ，就可以指导最终的特征向量  $\mathbf{BD}_j = (M_j^T, S_j^T)^T \in \mathbb{R}^8$ ，完整的 LBD 特征描述如下：

$$\text{LBD} = (M_1^T, S_1^T, M_2^T, S_2^T \dots M_m^T, S_m^T)^T \in \mathbb{R}^{8m} \tag{3.4}$$

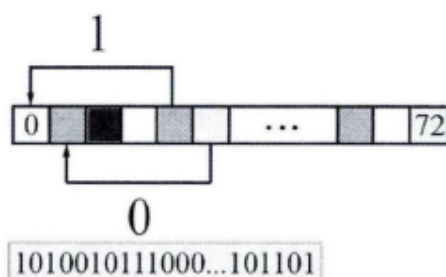


图 3.5 二进制 LBD 描述子

按照 LSD 提出者的实验结果， $m=9$  和  $w=7$  描述结果最优，这样获取的描述子最优。不过描述子还是浮点型的描述子。和点特征类似，在需要实时性的视觉 SLAM 系统中，一般采用二进制的 01 描述而不是采用浮点型的描述。所以一般在视觉 SLAM 的



直线描述中采用二进制 LBD 描述子。如图3.5所示，二进制描述是如果第  $i$  个数据比第  $j$  个数据小就记录下 0，反之则记录下 1。按照固定 256 中数据对比模式，在得到 LBD 描述子后计算得到 256 个 72 维的向量。二进制描述子的优势是可以对两个二进制字符串之间直接进行整体的操作，不需要对每一个数据进行比较，这样极大程度的增加了匹配的速度。如图3.6所示，就是两张图片 LSD 检测直线的 LBD 描述后匹配的效果图。

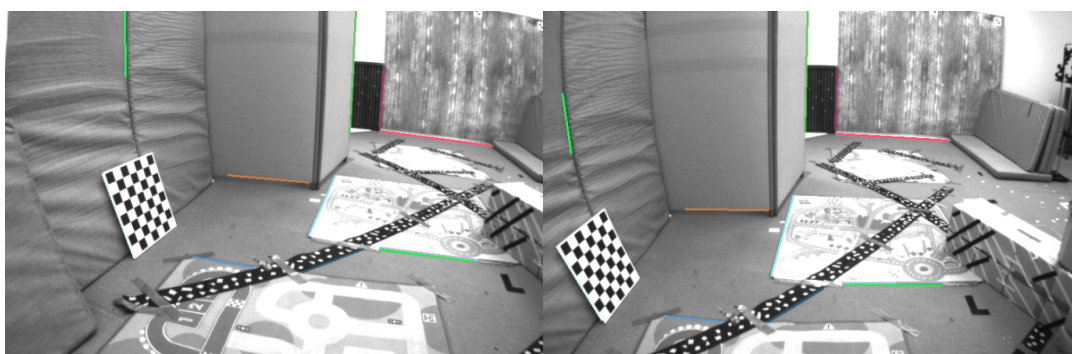


图 3.6 LBD 描述匹配结果

## 3.2 空间直线的参数化

空间中点特征的参数化非常直观，就是  $P = [x, y, z]$ 。而对于空间直线特征来说，参数化就不那么容易了。本文将介绍两种不同的空间无限长直线的参数化方法，普吕克坐标和正交表示。至于为什么需要两种不同的参数化来表示空间直线呢？首先普吕克坐标是比较常用的直线表示方法，因为这种表示可以非常快捷进行初始化而且在不同坐标系之间进行转换。但是空间直线只有 4 个自由度<sup>[30]</sup>，而普吕克坐标有 5 个参数。如果直接在优化后端使用普吕克坐标表示直线进行优化就会导致过参数化问题<sup>[35]</sup>。于是就需要一种只用 4 个参数就可以更新表示直线的参数形式，所以需要引入正交表示<sup>[29]</sup>。

### 3.2.1 普吕克坐标

三维空间中直线  $\mathcal{L}$  的普吕克坐标表示是  $\mathcal{L} = [\mathbf{n}^\top, \mathbf{d}^\top]^\top \in \mathcal{R}^6$ 。其中  $\mathbf{d} \in \mathbf{R}^3$  是空间直线的方向向量， $\mathbf{n} \in \mathbf{R}^3$  是空间直线和相机光心组成平面的法向量。但是普吕克坐标的 6 个参数本身隐含着两个约束：

- 前后两个三维向量必须正交， $\mathbf{n}^\top \mathbf{d} = 0$
- $\|\mathbf{n}\|$  和  $\|\mathbf{d}\|$  两者之间的比值表示了空间直线距离空间坐标原点的最短距离

虽然普吕克坐标有着过参数化的缺陷，但是它有着非常方便的坐标变换和初始化方式。首先介绍普吕克坐标初始化形式。如图3.7b 所示，在已知两个视角的同一空间直线的观测和两个视角相机的位姿之后，就可以通过两个观测直线和视角位置形成的

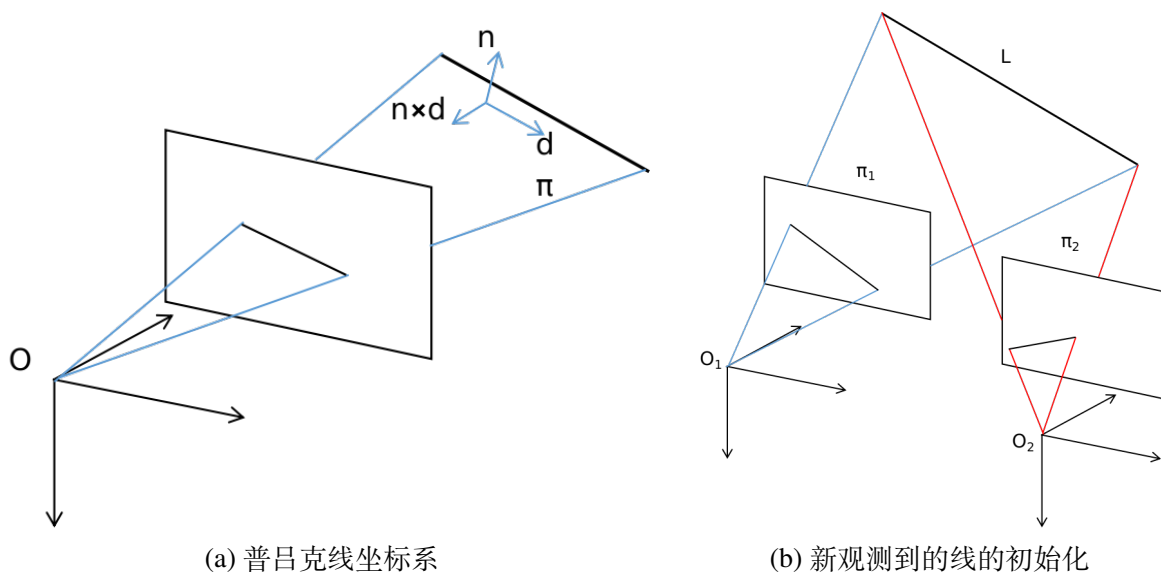


图 3.7 线特征的普吕克坐标

观测平面得到一条交线。这个交线就是想要得到的空间直线。

光心分别组成的观测平面相交的形式得到唯一确定的空间直线。

对于两个不同的直线观测视角，可以分别通过相机光心和观测的线段端点来求得直线观测平面  $\pi = [\pi_x \ \pi_y \ \pi_z \ \pi_w]$ ，然后通过偶普吕克矩阵  $\mathbf{L}^*$ <sup>[43]</sup> 可以由两个直线观测平面得到普吕克坐标。偶普吕克矩阵的性质如下：

$$\begin{aligned} \mathbf{L}^* &= \pi_1 \pi_2^\top - \pi_2 \pi_1^\top \in \mathbb{R}^{4 \times 4} \\ &= \begin{bmatrix} [\mathbf{d}]_{\times} & \mathbf{n} \\ -\mathbf{n}^\top & 0 \end{bmatrix} \end{aligned} \quad (3.5)$$

上式  $[\cdot]_{\times}$  表示三维向量的反对称矩阵。

除了初始化，普吕克坐标在不同坐标之间的变换也同样很容易实现。假设给定世界坐标到相机坐标的变换矩阵  $T_{cw} = \begin{bmatrix} \mathbf{R}_{cw} & t_{cw} \\ 0 & 1 \end{bmatrix}$ ，则有

$$\mathcal{L}^c = \begin{bmatrix} \mathbf{n}^c \\ \mathbf{d}^c \end{bmatrix} = \mathcal{T}_{cw} \mathcal{L}^w = \begin{bmatrix} \mathbf{R}_{cw} & [t_{cw}]_{\times} \mathbf{R}_{cw} \\ \mathbf{0} & \mathbf{R}_{cw} \end{bmatrix} \mathcal{L}^w \quad (3.6)$$

### 3.2.2 正交表示

正交表示的提出是为了使得后端优化中可以有一种可以用 4 个参数表示来进行无约束的优化。正交表示的形式是  $(\mathbf{U}, \mathbf{W}) \in SO(3) \times SO(2)$ ，由一个三维的旋转和一个二维的旋转组成。下面将介绍如何由一个已知的普吕克坐标  $\mathcal{L} = (\mathbf{n}^\top, \mathbf{d}^\top)^\top$  来得到正交表示。

首先对  $\mathcal{L}$  进行 QR 分解，由于普吕克坐标两个三维向量正交，所以可以得到：

$$\begin{bmatrix} \mathbf{n} & \mathbf{d} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{n}}{\|\mathbf{n}\|} & \frac{\mathbf{d}}{\|\mathbf{d}\|} & \frac{\mathbf{n} \times \mathbf{d}}{\|\mathbf{n} \times \mathbf{d}\|} \end{bmatrix} \begin{bmatrix} \|\mathbf{n}\| & 0 \\ 0 & \|\mathbf{d}\| \\ 0 & 0 \end{bmatrix} \quad (3.7)$$

上式右侧第一项是三维单位正交阵  $\mathbf{U}$ ，它表示的是相机坐标系到直线的坐标系的旋转，也就是正交表示的第一项。其中所述的直线坐标系的三个坐标轴定义如下： $\mathbf{X}$  轴是直线的方向向量， $\mathbf{Y}$  轴是直线和光心组成平面的法向量， $\mathbf{Z}$  轴是前面两个正交向量叉乘得到的。于是有

$$\mathbf{U} = \mathbf{R}(\boldsymbol{\psi}) = \begin{bmatrix} \frac{\mathbf{n}}{\|\mathbf{n}\|} & \frac{\mathbf{d}}{\|\mathbf{d}\|} & \frac{\mathbf{n} \times \mathbf{d}}{\|\mathbf{n} \times \mathbf{d}\|} \end{bmatrix} \quad (3.8)$$

其中  $\boldsymbol{\psi} = [\psi_1, \psi_2, \psi_3]^T$  代表的是相机坐标系到直线坐标系在  $x$ ， $y$  和  $z$  轴的旋转角。在得到了正交表示的第一项之后，第二项可以通过如下变换得到：

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \\ &= \frac{1}{\sqrt{(\|\mathbf{n}\|^2 + \|\mathbf{d}\|^2)}} \begin{bmatrix} \|\mathbf{n}\| & -\|\mathbf{d}\| \\ \|\mathbf{d}\| & \|\mathbf{n}\| \end{bmatrix} \end{aligned} \quad (3.9)$$

上式中的  $\phi$  是旋转角。由于坐标原点（相机光心）到 3D 直线的距离是  $d = \frac{\|\mathbf{n}\|}{\|\mathbf{d}\|}$ ，所以  $\mathbf{W}$  包含了距离信息  $d$  的。根据  $\mathbf{U}$  和  $\mathbf{W}$  的定义可以看出，4 个自由度包括旋转的 3 个自由度和距离的一个自由度。在优化的时候，使用  $\mathcal{O} = [\boldsymbol{\psi}, \phi]^T$  作为空间直线更新的最小表示。

在已知直线正交表示的情况下，求出对应的普吕克坐标也很方便。假设有正交表示  $(\mathbf{U}, \mathbf{W})$ ，则有

$$\mathcal{L}' = [w_1 \mathbf{u}_1^T, w_2 \mathbf{u}_2^T]^T = \frac{1}{\sqrt{(\|\mathbf{n}\|^2 + \|\mathbf{d}\|^2)}} \mathcal{L} \quad (3.10)$$

其中  $\mathbf{u}_i$  代表的是  $\mathbf{U}$  矩阵的第  $i$  列。虽然  $\mathcal{L}$  和  $\mathcal{L}'$  有一个尺度的差，但是它们表示的是同一条空间直线。

### 3.3 直线观测模型

介绍直线特征的观测，首先介绍相机坐标系下的直线怎么投影到像素平面上。如图3.8所示， $A$  与  $B$  是空间直线  $\mathcal{L} = (\mathbf{n}^T, \mathbf{d}^T)^T$  上两点，点  $\mathbf{a}$  和  $\mathbf{b}$  是它们在像素平面上的

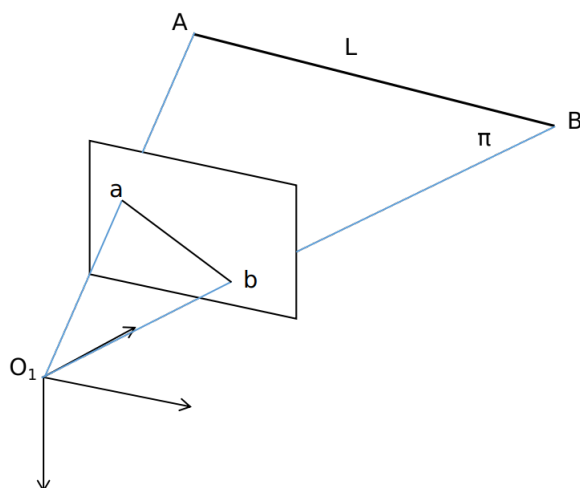


图 3.8 直线投影模型

投影。假设相机的内参矩阵  $\mathbf{K}$  已知，那么

$$\begin{aligned} a &= \mathbf{K}A \\ b &= \mathbf{K}B \end{aligned} \quad (3.11)$$

又有空间直线和光心平面法向量  $\mathbf{n} = [A] \times B$ ，像素平面直线向量  $\ell = [l_1 \ l_2 \ l_3] = [c] \times d$ ，于是整理可以得到

$$\ell = \mathcal{K}\mathbf{n} = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} \mathbf{n} \quad (3.12)$$

由上式直线投影模型可知，直线在投影时直线的方向向量的信息被全部丢弃，只有直线和光心平面法向量的信息得以保留。

对于直线在图像平面的误差测量，本文使用观测线段的端点到预测直线的垂直距离定义。假设已知世界坐标系下空间直线  $\mathcal{L}_{\downarrow}^{\mathbb{R}^3}$  和坐标变换，首先把空间直线经过坐标变换转换到相机坐标系下得到  $\mathcal{L}_{\downarrow}^{\mathbb{R}^3}$ 。然后通过直线的投影模型将直线投影到相机图像中，最后就可以通过计算得到直线的观测误差：

$$\mathbf{r}_l(\mathbf{z}_{\mathcal{L}_l}, \mathcal{X}) = \begin{bmatrix} d(\mathbf{s}_l^c, \mathbf{l}_l^c) \\ d(\mathbf{e}_l^c, \mathbf{l}_l^c) \end{bmatrix} d(\mathbf{s}, 1) = \frac{\mathbf{s}^\top \mathbf{l}}{\sqrt{l_1^2 + l_2^2}} \quad (3.13)$$

上式中  $\mathbf{s}_l^c$  和  $\mathbf{e}_l^c$  表示空间直线对应的 2D 观测的线段端点， $\mathbf{l}_l^c$  表示空间直线重投影到 2D 图像平面的无限长直线。

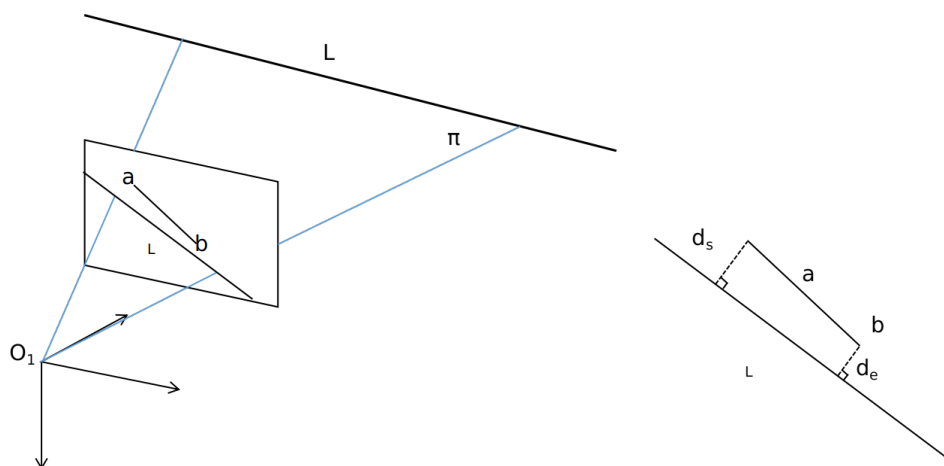


图 3.9 直线误差定义

### 3.4 本章小结

本章介绍了如何将空间中的直线特征应用到视觉 SLAM 系统中。3.1 节介绍了图像层面上的线特征的使用，使用 LSD 算法检测图像上的线特征并使用 LBD 算法对检测的线段特征进行描述。3.2 节介绍了空间直线特征两种不同的参数化形式，一种是方便由图像信息初始化和进行坐标变换的普吕克坐标参数形式，另一种是方便在 SLAM 后端系统中优化的正交表示参数化形式。3.3 节介绍了空间直线的投影模型以及直线特征的误差的构建形式。



## 第四章 带有共面约束的点线 VIO 系统设计

相较于点特征来说，直线特征和平面特征融合了空间结构信息。将直线特征和平面特征加入到只有点特征的视觉 SLAM 系统，这些结构信息可以辅助系统提升位姿估计的精度和空间地图的重建精度。本章的出发点就是构建一个融合空间点线特征以及它们的共面约束的 VIO 系统。对于没有深度信息传感器的 SLAM 系统来说，如何找到空间中的平面信息是首要的问题。本章提出了一个构建三维空间网格然后从空间网格中提取平面信息的解决方法。然后把检测到的平面信息和空间点线特征融合到 VIO 系统中。最后设计了仿真实验和真实数据集实验来验证提出的平面信息检测和共面信息约束后端的有效性。

本章内容构成如下：4.1节介绍了使用空间点线产生三维网格的方法，4.2节介绍了如何利用 IMU 信息在上一节产生的网格中提取平面并检测出共面信息，4.3节介绍了本章构建的 VIO 系统以及该系统中如何利用共面信息，4.4节设计了实验验证本章提出的算法的有效性。

### 4.1 生成三维网格

如果想在 SLAM 系统产生的空间三维点云中直接生成一个稳定的三维网格是很困难的，主要原因如下：

- 1) 直接由 SLAM 系统从观测生成的三维点云充满了噪声，同时有很多外点；
- 2) 点云的分布非常不均匀和规律；
- 3) SLAM 系统中的点云非常不稳定，会经常有点需要加入和删除，而且点的位置也会经常改变。

同时直接从三维点云生成三维网格也会消耗很多计算资源，这对 SLAM 系统来说是不可接受的。于是本章提出了一种融合多帧观测的从二维网格生成三维空间的高效三维网格生成方法。

#### 4.1.1 从二维网格到三维网格

如图4.2a所示，本章提出的三维网格生成方法大概思想就是在二维图像上生成网格关联，然后在依照已有的二维特征到三维特征的数据关联直接把二维网格投影到三维当中。

首先要做的就是二维图像上保留那些已经有三维深度信息的点特征和直线特征。并不是所有被提取的二维图像特征都可以被保留，是因为后续算法需要把二维的

特征构建的网格信息投影到三维当中，如果保留那些没有三维关联的二维特征，那么就会导致后续投影出现错误。

接着需要做的就是二维图像上做三角剖分，用二维图像特征构建出二维网格。因为本章算法图像特征不仅有点特征还有线段特征，而且线段特征的空间结构信息希望被保留。所以如图4.1本章对图像上的特征进行 CDT (constrained Delaunay triangulation)<sup>[44]</sup>。

最后一步就是保留二维生成的网格信息，将网格整体从二维投影到三维当中。如图4.2a展示。

虽然本章构建的三维网格只从一帧图像中生成，但是整体的三维网格是融合了多帧信息的。因为在 SLAM 系统中，所有三维路标点都是会被融合多帧信息去更新的。当 SLAM 系统在运行的过程中，不断保留生成的三维网格信息，最终就可以得到整个场景的三维稠密网格地图。

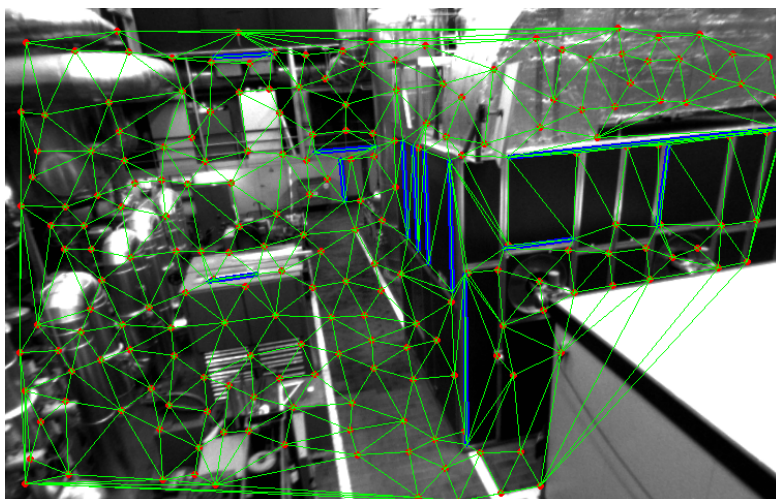


图 4.1 二维图像三角剖分示意图

#### 4.1.2 网格滤波

由二维图像网格重投影到三维空间中生成的三维网格虽然很高效，但是不可避免的会产生一些问题。第一个问题就是，二维图像不包含三维深度信息，使得直接在二维图像上生成的网格不可避免的会生成一些跨越多个平面的不符合空间约束的错误网格。第二个问题就是，三角化得到的二维特征的三维空间坐标，有时候因为错误匹配等问题会使得三维空间位置不对，也会导致一些错误的三维网格。

本章为了处理这些不合理的三维网格，设定了下面几条规则：

1) 去除有两个特别小锐角的三维空间网格。这些瘦小的三维空间网格通常不能提供足够的空间信息，同时也一般是错误的跨平面的网格。



- 2) 去除三角形网格中最长边比最短边的比值过大的三角形。
- 3) 去除那些边长超过固定值的三维网格。如果某个三角形网格有个边长特别长,那么通常不是符合真实世界约束的。

## 4.2 共面信息检测

在空间中生成三维三角网格之后,就可以从网格中提取出一些空间结构信息,比如当前场景存在多少个平面或者当前场景中哪些特征是共面的。本节将使用一种非迭代式的方法(不像 RANSAC<sup>[45]</sup> 这类方法)来从这些网格中提取出空间结构信息。

从网格中提取出空间平面,就可以从这些平面和网格的数据关联中找到生成网格的空间点线的共面信息。于是首先需要从网格中提取出平面。本节提取平面的方法只关注与结构化条件下的平面,只能检测两种类型的平面:垂直的平面(如墙)和水平的平面(如地板、天花板或桌子表面)。所以这种方法需要知道空间中重力的方向,IMU 刚好就可以提供这个方向的估计,所以本节提取平面的方法也依赖于 VIO 方法。

### 4.2.1 水平平面检测

先从水平平面的检测开始描述,水平平面的检测过程如下:

- 1) 将所有的三维三角形网格按照它们的法向量信息分类,只保留法向量和重力方向平行的网格。
- 2) 将所有三角网格按照它们在它们法向量上到空间原点的高统计成一维直方图。
- 3) 从直方图统计数据中提取出局部最大值。为了从直方图中提取出局部最大值,需要先对直方图进行高斯滤波。高斯滤波的目的是为了使得统计数据变得平滑,方便提取出局部最大值。为了滤除一些离群点,提取出局部最大值之后,只保留统计点的数目大于一定数目的结果。提取出来的局部最大值就被认为是空间中的一个平面。
- 4) 从平面中提取出共面信息。因为知道平面只由空间三角网格组成的,而空间三角网格由哪个空间点线组成也是已知的,所以在检测到平面之后就可以从这些数据关联中直接得知哪些空间点线是共面的。

### 4.2.2 垂直平面检测

在知道水平平面是如何从空间中检测出之后,垂直平面的检测过程也与之类似:

- 1) 将所有的三维三角形网格按照它们的法向量信息分类,只保留法向量和重力方向垂直的网格。
- 2) 创建一个二维的直方图统计三角网格信息。这个二维的直方图两个轴分别是这些三角网格平面的法向量和定义初始方向的夹角和这个三角网格在这些法向量上到空

间原点的高。

3) 从二维直方图中提取出局部最大值。和一维直方图统计类似，二维的直方图也需要进行高斯模糊之后提取局部最大值。这个局部最大值就代表了这里有个垂直的大平面。

4) 和水平平面相似，检测到平面之后共面信息的提取也是可以直接获取的。

图4.2是垂直和水平平面的检测结果。

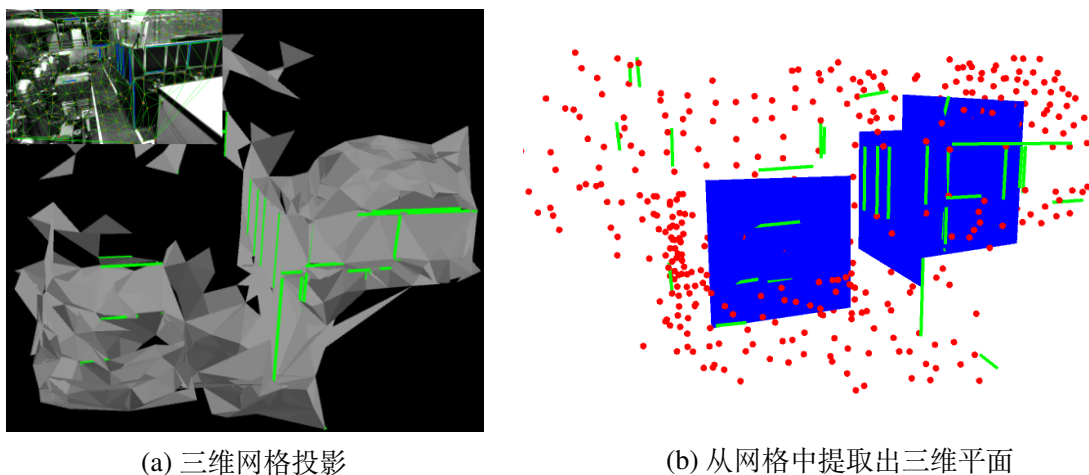


图 4.2 网格提取平面过程

## 4.3 带有平面约束的系统构建

### 4.3.1 系统概述

本章工作是建立在前人 PL-VIO<sup>[31]</sup> 之上的，PL-VIO 它将线段合并到 Vins Mono<sup>[21]</sup>。如图4.3所示系统包括前端和后端两个模块。在前端，对来自 IMU 和摄像机的原始测量数据进行预处理。相关操作包括 IMU 预集成、检测和点线特征匹配。后端是通过前端相机和 IMU 的数据以及它们预处理之后的数据估计出当前系统中相机和 IMU 的位姿还有维护的地图中的三维点和线。

对于前端的图像数据，图像中的点特征的检测和匹配采用的是 FAST 角点和 LK 光流，图像中点特征的检测和匹配采用的是 LSD 检测和 LBD 描述匹配。

对于后端优化，本章系统沿用 VINS mono 中滑动窗口优化的策略，维护一个持续多帧的优化窗口对窗口中的所有数据进行联合优化。滑动窗口中会通过最小化 IMU 预积分误差、视觉部分的点和直线的重投影误差以及空间点线的共勉约束来优化所有状态量。

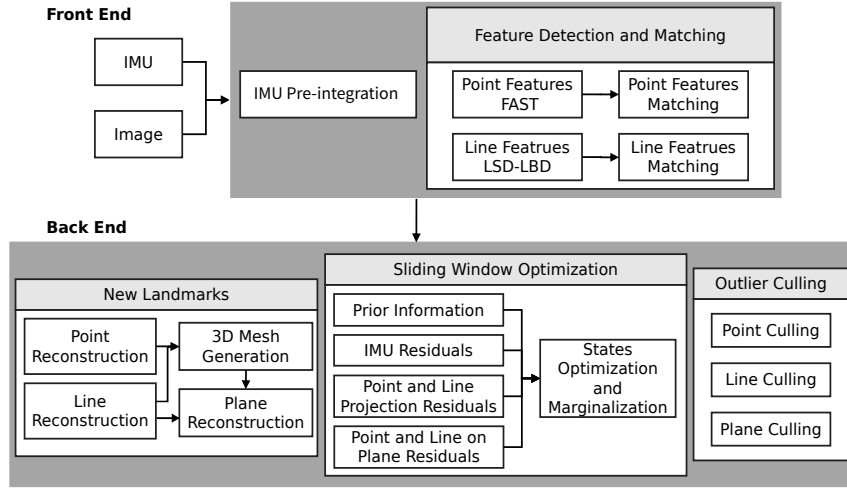


图 4.3 系统架构图

### 4.3.2 优化问题建模

本章提出的 VIO 系统在滑动窗口中维护了 IMU 状态量、点特征的深度信息、直线特征的空间信息以及平面特征的空间信息，所有状态量如下：

$$\begin{aligned} \mathcal{X}_t &\doteq \{\mathbf{x}_i, \lambda_k, \mathbf{O}_l^w, \boldsymbol{\pi}_h^w\}_{i \in \mathcal{B}_t, k \in \mathcal{F}_t, l \in \mathcal{L}_t, h \in \Pi_t} \\ \mathbf{x}_i &\doteq [\mathbf{p}_{wb_i}, \mathbf{q}_{wb_i}, \mathbf{v}_{b_i}^w, \mathbf{b}_a^{b_i}, \mathbf{b}_g^{b_i}] \end{aligned} \quad (4.1)$$

其中集合  $\mathcal{B}_t$  表示的是时刻  $t$  时的 IMU 体位姿状态。集合  $\mathcal{F}_t$ 、 $\mathcal{L}_t$  和  $\Pi_t$  分别表示  $t$  时刻中活跃在在滑动窗口中的所有点特征、直线特征和平面特征。 $\mathbf{x}_i$  由 IMU 在世界坐标  $w$  下的位移  $\mathbf{p}_{wb_i} \in \mathbb{R}^3$ ，方位  $\mathbf{q}_{wb_i}$ ，速度  $\mathbf{v}_{b_i}^w \in \mathbb{R}^3$  组成。 $\mathbf{b}_a^{b_i}, \mathbf{b}_g^{b_i} \in \mathbb{R}^3$  分别是  $b_i$  帧下的加速度计和陀螺仪的偏移量。本章系统使用四元数在状态量中表示旋转，用旋转矩阵  $\mathbf{R} \in SO(3)$  来旋转向量。对于不同的特征，本章使用第一次观测坐标系下的逆深度  $\lambda_k$  来参数化点特征  $k^{th}$ 。 $\mathbf{O}_l^w$  和  $\boldsymbol{\pi}_h^w$  分别是世界坐标  $w$  下直线  $l^{th}$  的正交表示以及平面  $h^{th}$  的空间表达。

本章构建的系统在滑动窗口中通过最小化下面所有对观测信息的误差来优化所有状态量：

$$\begin{aligned} \mathcal{X} &= \arg \min_{\mathcal{X}} \|\mathbf{r}_p - \mathbf{H}_p \mathcal{X}\|^2 + \sum_{i \in \mathcal{B}} \|\mathbf{r}_b\|_{\Sigma_{b_i b_{i+1}}}^2 \\ &+ \sum_{i \in \mathcal{B}} \left( \sum_{k \in \mathcal{F}} \rho \left( \|\mathbf{r}_{f_k}^{c_i}\|_{\Sigma_{\mathcal{F}}}^2 \right) + \sum_{l \in \mathcal{L}} \rho \left( \|\mathbf{r}_{L_l}^{c_i}\|_{\Sigma_{\mathcal{L}}}^2 \right) \right) \\ &+ \sum_{h \in \Pi} \left( \sum_{k \in \mathcal{F}} \rho \left( \|\mathbf{r}_{f_k}^{\pi_h}\|_{\Sigma_{\mathcal{F}}}^2 \right) + \sum_{l \in \mathcal{L}} \rho \left( \|\mathbf{r}_{L_l}^{\pi_h}\|_{\Sigma_{\mathcal{L}}}^2 \right) \right) \end{aligned} \quad (4.2)$$

其中  $\mathbf{r}_b$  是 IMU 的测量误差。 $\mathbf{r}_{fk}^{ci}$  和  $\mathbf{r}_{Ll}^{ci}$  分别是点特征和直线特征的重投影误差。 $\mathbf{r}_{fk}^{\pi h}$  和  $\mathbf{r}_{Ll}^{\pi h}$  分别是点在平面和直线在平面上的误差。图4.4是这些误差的空间描述。

点特征的测量模型和直线特征的测量模型在本节就不重复介绍，下面介绍以下空间平面提供的共面约束的测量模型。

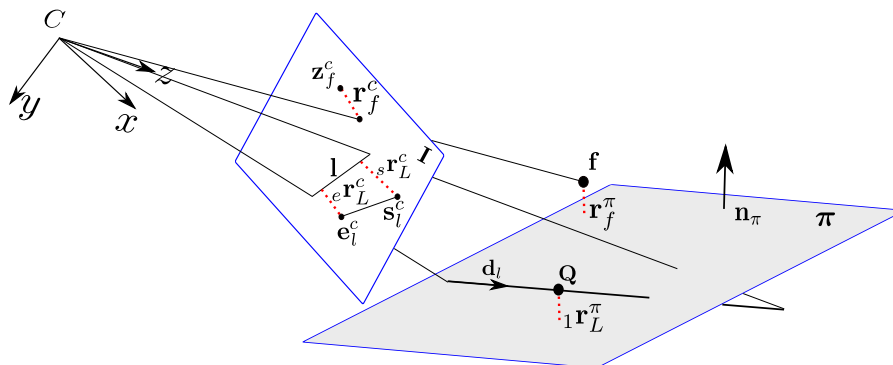


图 4.4 空间点线的共面残差

#### 4.3.2.1 平面特征的测量模型

点到平面的误差：对于空间中点到平面的误差构建比较直观，可以根据空间点到平面的最短距离来构建。假设空间点为  $\mathbf{f}_k = [x, y, z]^T$ ，空间平面坐标为  $\boldsymbol{\pi}_h = [\mathbf{n}_\pi, d_\pi]^T$ ，误差可以表述为：

$$\mathbf{r}_{fk}^{\pi h} = \mathbf{n}_\pi \cdot \mathbf{f}_k - d_\pi \quad (4.3)$$

直线到平面的误差：如果一个直线在平面上，可以提供两个方面的信息<sup>[46]</sup>

- 空间直线上的所有点都在平面上；
- 直线的方向向量和平面的法向量互相正交。

假设当三维空间直线的普吕克坐标是  $\mathcal{L}_l = [\mathbf{n}_l, \mathbf{d}_l]^T$ ，空间平面为  $\boldsymbol{\pi}_h = [\mathbf{n}_\pi, d_\pi]^T$ ，误差可以表述为

$$\begin{aligned} {}_1\mathbf{r}_{Ll}^{\pi h} &= \mathbf{n}_\pi \cdot \mathbf{Q} - d_\pi \\ {}_2\mathbf{r}_{Ll}^{\pi h} &= \mathbf{n}_\pi \cdot \mathbf{d}_l \end{aligned} \quad (4.4)$$

其中  $\mathbf{Q} = \frac{\mathbf{n}_l \times \mathbf{d}_l}{\|\mathbf{d}_l\|}$ ，表示三维空间直线上的一个点。

## 4.4 实验以及分析

为了验证本章提出的为点线 VIO 系统添加共面约束的算法是有效的，本节在仿真数据和真实数据集上做了相关的实验。本节实验的所有结果都是在搭载英特尔 Core

i7-8550U @ 1.8GHz 的 CPU 和 16GB 运行内存的笔记本上产生的。

为了对比本章提出的 VIO 系统的定位精度达到了先进的水平, 本部分将和 OKVIS, VINS-Mono, PL-VIO 以及 Kimera-VIO 进行精度的对比。为了下文行文方便, 本文将不同的算法组合分成:

P: 只有点特征的算法序列, 此时系统和 VINS-Mono 的系统完全相同

PP: 带有点特征和点特征的共面约束的算法序列, 此时系统和 Kimera-VIO 系统类似。

PL: 带有点特征和直线特征的算法序列, 此时系统和 PL-VIO 系统相似。

PLP: 带有点线特征和它们的共面约束的算法序列, 此时系统是本章提出算法的最终系统。

本节将会在定位精度, 地图精度以及实时性三个方面验证所提出算法的性能。本文采用绝对位置误差 (absolute pose error, APE) 作为主要的轨迹计算精度评判标准, 用相对位置误差 (relative pose error, RPE) 补充分析比较。

#### 4.4.1 仿真数据验证

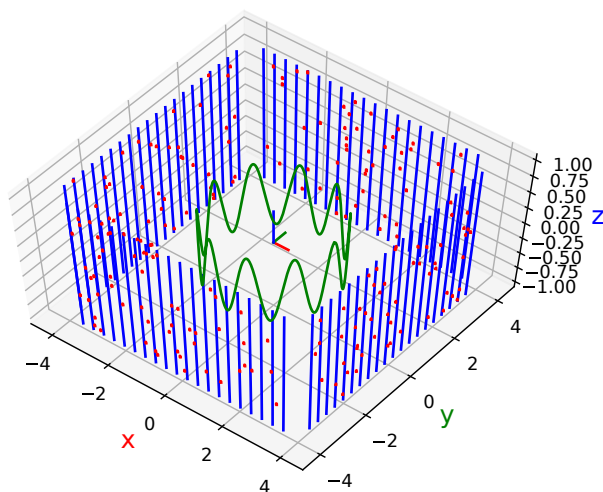


图 4.5 仿真场景示意图

本小节仿真实验的目的是为了验证本章提出的点线共面约束在后端优化中的有效性, 所以仿真实验默认了所有空间点线图像观测的数据关联都是已知的, 空间点线的共面信息也是已知的, 并且提供了带有噪声的初始相机位姿。在保证这些和后端优化部分无关的变量都一致的情况下, 再使用同样的数据运行本章开始提出的四种不同的算法组合。为了使得实验结果不包含随机性, 所有仿真实验的结果都是在多次蒙特卡洛实验后产生的。仿真场景的示意图如4.5所示。整个仿真实验生成的相机序列大概有 300 帧图像, 图像分辨率为  $640 \times 480$ , 每一帧图像大概可以观测到 15 个特征点和 8

个直线。每个观测到的特征点和直线的端点都会在精确的图像投影上被添加上方差为 1 个像素的高斯白噪声。仿真实验提供的相机初始位姿也是在正确位姿的基础上带有 1 度的旋转和 10cm 的高斯白噪声的结果。

表 4.1 所有不同方法在仿真数据上的结果

	P	PP	PL	PLP
RMSE APE (cm)	8.94	8.06	7.97	<b>7.34</b>
地图误差 (cm)	3.21	1.57	3.44	<b>1.32</b>

注：粗体表示最佳结果

总结了方法 P、PP、PL 和建议的 PLP 对仿真数据的测试结果。用 APE 的均方根误差 (root mean squared error, RMSE) 评价其准确性。地图误差是指真实三维位置和估计的位置之间距离的均方根误差。结果表明，共平面约束可以提高轨迹精度，本章提出的 PLP 方法优于比较方法。对于地图质量，可以看出，PL 的最大地图误差是由直线端点的不精确三角化造成的。而平面约束可以有效地提高直线的精度和地图结果。

#### 4.4.2 EuRoC 数据集实验

##### 4.4.2.1 数据集介绍

本章测试使用的数据集是经常用来评测系统的室内 SLAM 和 VIO 系统的数据集——EuRoC 数据集。EuRoC 数据集原始数据是由无人机设备采集的，数据集提供了双目相机图像原始数据。双目相机型号是 MT9V034，采用全局快门，采集的图像是单色图像，采集频率是 20Hz。数据集的位姿真值是由高精度的位姿采集机器获得的。EuRoC 数据集一共提供了 11 组在两个不同场景中记录的数据。

其中一个场景是 Machine Hall，用字母 MH 表示，是一个工厂的内部。这个场景的平面信息不是特别丰富，因为数据集记录时并没有很多的记录上天花板和地板的数据。但是由于工厂也是人工建筑，所以直线结构在场景中还是比较丰富的。

另一个场景是 Vicon 房间，用字母 V 表示，是典型的室内场景。这个场景地板天花板墙面都记录的很全面，同时房间中还堆叠了一些箱子和床垫，所以这个场景的直线和平面信息非常丰富。其中本场景数据共有两种不同的记录类型 V1 和 V2，这两个不同的记录类型只是改变了房间中堆叠物体的位置。

除此之外，每一组数据序列都根据所代表的 SLAM 算法的难度级别进行了标记，使用了“easy”，“medium”和“difficult”来形容。一般来说，提高无人机的飞行速度就可以是记录的序列运行难度增加，因为运动会使得图像上产生运动模糊和光照变化。例如数据集 MH03\_medium 就代表无人机用中速飞行在 MH 工厂内记录的数据序列。



室内的 Vicon 房间场景还提供了由高精度雷达采集的环境真实点云信息，可以用来检测算法生成的环境稠密地图的精度。

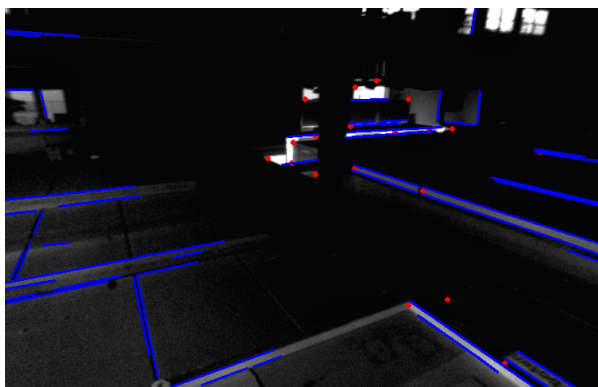


图 4.6 平面上具有丰富线条特征和少量点特征的场景示例。检测到的点和线分别用红色和蓝色显示

#### 4.4.2.2 定位精度

表 4.2 所有不同方法在 EuRoc 数据集上的 RMSE 结果

Seq.	OKVIS	P	PP	PL	PLP
	trans.rot.	trans.rot.	trans.rot.	trans.rot.	trans.rot.
MH02	36.03.5	15.32.1	14.42.0	11.71.8	<b>10.91.7</b>
MH03	21.51.5	18.51.3	18.61.3	15.41.1	<b>14.91.1</b>
MH04	24.01.1	24.71.7	24.51.7	21.61.5	<b>20.61.4</b>
MH05	39.61.1	22.21.3	22.31.3	18.31.2	<b>16.61.2</b>
V101	8.6 5.5	6.1 <b>5.2</b>	5.7 <b>5.2</b>	6.4 5.7	<b>5.4 5.2</b>
V102	12.22.3	7.5 <b>1.8</b>	<b>7.4 1.8</b>	7.6 <b>1.8</b>	7.5 <b>1.8</b>
V103	19.63.8	12.84.4	12.34.0	11.43.8	<b>11.03.3</b>
V202	18.22.7	15.73.2	13.72.6	15.02.4	<b>12.52.4</b>
V203	30.54.3	20.03.4	16.83.4	16.63.2	<b>15.42.4</b>

注：平移误差单位为厘米，旋转误差单位为角度，**粗体**表示序列中的最佳结果

我们在 EuRoc 数据集上评估了 OKVIS, P, PP, PL 和 PLP 不同算法的 RMSE APE。表4.2展示了不同方法的结果。对于 P、PP、PL 和 PLP 的实验细节，除了特征数之外，我们使用了<sup>[21]</sup>中的默认参数。为了提高三维网格检测生成的有效性，本章系统增加了点特征的数目，每个新帧图像将检测 200 个点特征和长度超过 50 像素的所有线特征。

从表4.2中可以看出，PLP 可以提高 PL 在所有序列上的性能，但 PP 不一定对 P 有效，这说明仅使用稀疏点特征检测平面有时会带来不正确的结构信息，直线的引入可以提高平面检测的精度。此外，除了 V102 序列，使用所有点、线和平面特征的方法

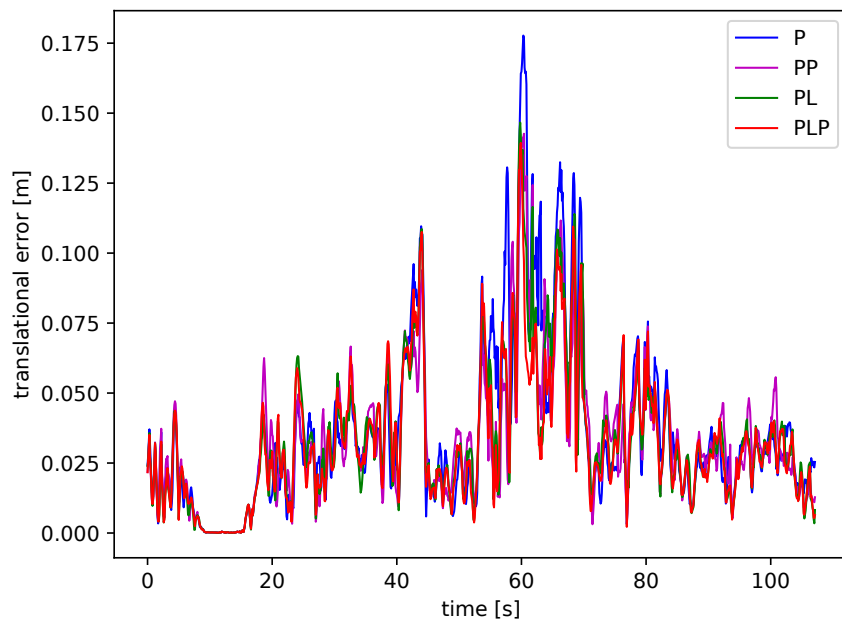


图 4.7 P, PP, PL, PLP 方法在序列 MH\_05 上 RPE 的平移误差

在所有序列上实现了最小的平移误差，V1 的序列被收集在一个正方形的房间里，房间的墙壁和地板上都有丰富的纹理。进一步分析 V101 和 V102 的结果表明，纹理非常丰富，只有点特征的 P 获得了很高的精度，PL 中不匹配的直线特征会降低精度。但是，由于与 V101 和 V102 相比，V103 中的相机移动速度非常快，高速运动引起的模糊和光照变化会降低 P 的精度，而直线特征匹配和检测更稳定，因此 PL 的精度高于 P。结果还可以说明平面结构规律对所有序列的 P 和 PL 都是有效的。

为了更深入地研究的系统的有效性，使用 RPE 来研究轨迹的局部精度。图 4.7 显示 MH05 上的结果。PLP 呈现出比其他方法更低的 RPE 转换误差，尤其是在 57 ~ 67s 的时间范围内。在该时间范围内记录的图像示例如图 4.6 所示，展示了一个平面上点特征少、结构线多的场景。当点特征不能有效地用于定位时，利用线特征或平面结构规则可以有效地提高定位精度。此外，PP 的精度有时比 P 低，例如  $t = 20s$  和  $t = 90 \sim 100s$  左右。相比之下，直线使平面检测更准确，PLP 误差小于或接近 PL，从而验证了直线在保留空间结构信息方面的有效性。

#### 4.4.2.3 地图质量比较

为了有效的评估地图质量，本文采用 Schops 定义点云评估标准<sup>[47]</sup>，先对三维网格进行均匀密度为  $10^3$  点/ $m^2$  的采样预处理。然后使用迭代最近点 (ICP) 算法将生成的点云与地面真值点云对齐。对于每个点，地图误差是它到最近的地面真值点的距离。为了进一步验证直线的几何结构对平面检测和稠密重建的有效性，验证平面结构可以提高地图的精度，比较了 PLP 和 P 方法，对于 P 方法，利用稀疏点特征构造网格进行稠密



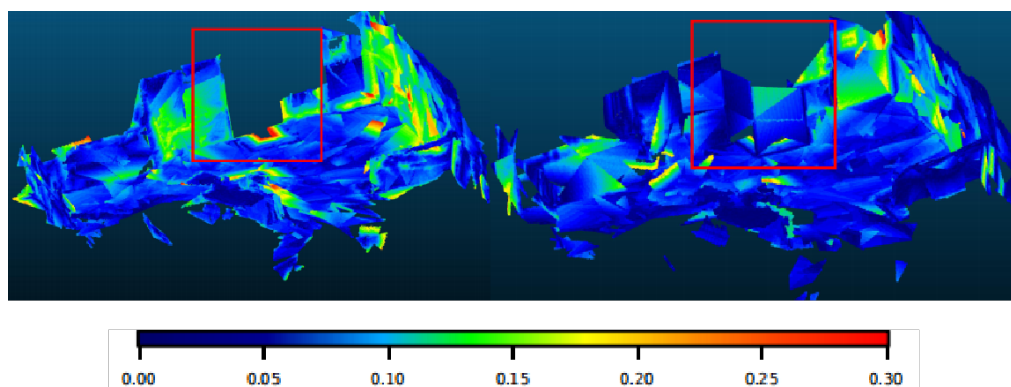


图 4.8 地图生成质量对比，左图为 P 序列生成结果，右图为 PLP 序列生成结果，误差单位：厘米

重建，但不要在优化中使用结构约束。P 和 PLP 在序列 V101 上给出的结果如图4.8所示。PLP 比 P 给出了更完整的稠密映射，因为直线在网格重建过程中提供了更好的结构信息，特别是对于具有无纹理表面的对象，例如图4.8中用红线标记的大长方体对象。此外，PLP 的平均距离误差（2.8cm）低于 P（3.6cm）。

#### 4.4.2.4 实时性

由于在滑动窗口优化中加入了直线和平面，增加了状态向量的大小和问题的复杂度。这里主要将后端模块的平均执行时间与其他方法进行比较，这些方法是在 V102 序列上计算的。如表4.3，可以观察到非迭代方法有效地检测平面并创建三维网格，运行时间在 2ms 以下，优化和边缘化的时间开销仅增加了几毫秒。其原因是采用 Schur 补法加速求解，而且滑动窗口中的平面数（ $\approx 5$ ）和线数（ $\approx 30$ ）比点数（ $\approx 200$ ）少。

表 4.3 不同方法在序列 V1\_02 上的平均运行时间（单位：毫秒）

Module	P	PP	PL	PLP
平面检测	0	0.39	0	0.44
网格生成	0	0.65	0	1.42
后端优化	31.61	33.63	30.59	35.88
边缘化	6.05	6.27	7.29	8.34

## 4.5 本章小节

本章针对室内场景结构化信息较多的特点设计了一个结合共面信息的点线 VIO 系统。该系统可以结合空间点线特征实时的构建出一个半稠密的空间三维网格地图，并且可以从网格地图中提取出平面特征和点线特征共面信息。该系统在后端优化也加入

了点线的共面约束，使得系统很好的利用了空间结构信息。最后本章通过仿真数据和真实数据集数据对算法进行了验证。实验验证了本章提出算法可以很好的利用空间中直线平面这些结构化信息提升 VIO 系统精度和鲁棒性，并可以利用共面约束提高对地图点的估计精度。

## 第五章 共面点线的平面参数化

对于在 SLAM 系统中加入共面的约束，第四章直接在优化后端加入点线在平面的共面上的优化项是一种解决办法。这种方法加入约束非常直接，不过也不可避免的增加了优化问题的规模。本章从参数化的角度给出了另一种添加共面约束的解决办法。同时第四章检测平面的方法必须借助 IMU 的测量信息，使得没有 IMU 数据的系统不能借鉴。本章也给出了另一种只利用图像信息来检测空间平面和共面约束的方法。最后本章在仿真数据和真实数据集上都进行了实验来验证本章提出的平面找寻方法和参数化方法的有效性。

本章内容构成如下:5.1节介绍了提取空间中点和直线特征的共面约束的方法,5.2节介绍了利用平面参数化共面点线的方法，5.3节设计了实验验证本章提出的算法的有效性。

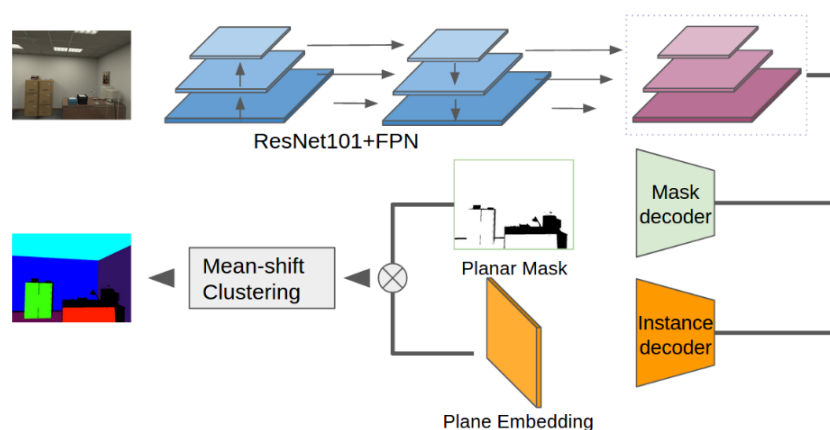


图 5.1 平面实例分割网络的架构图

### 5.1 空间共面信息找寻方法

本文找寻共面特征的方法没有使用带有深度信息的传感器，同时想要实时的检测出空间中的共面约束。

#### 5.1.1 平面实例分割

本文使用的平面实例分割网络是简化版的 PlaneReconstruction 网络。图5.1是本文所使用平面实例分割网络的架构图。

图5.2是本文平面实例分割网络在室内场景的分割结果。图中不同颜色代表了网络

分割出来的不同平面区域，可以看出本文网络虽然不能每个像素都可以正确的识别出来。但是本文后续算法会使用视觉 SLAM 的结果然后使用滤波器去滤除分割错误的部分。

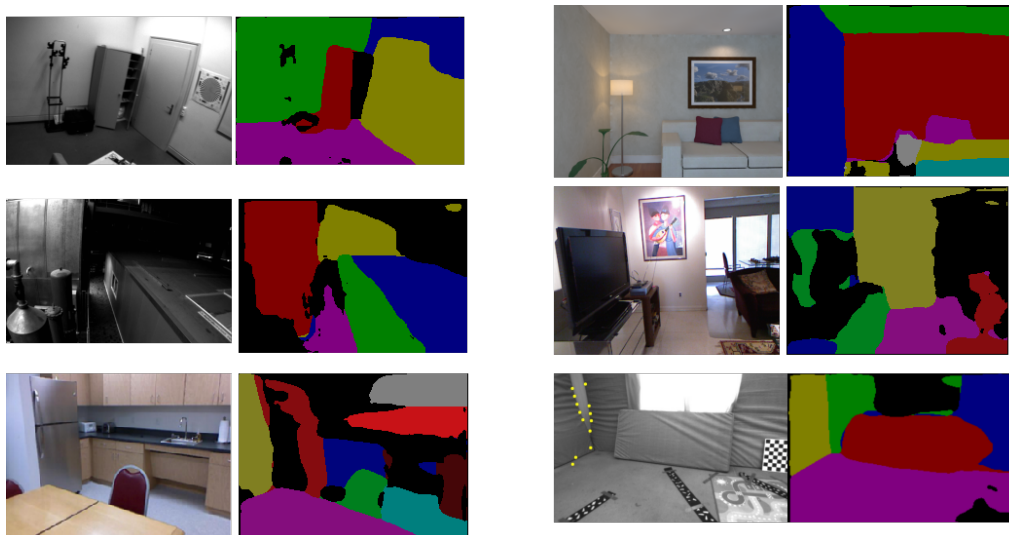


图 5.2 平面分割网络的实例图

### 5.1.2 共面信息提取

由于神经网络在图像信息上提取的平面有时不够准确，所以本章使用深度学习检测结果和基于 RANSAC 的滤波器来稳定的提取正确的共面信息的，本节就来介绍这种基于 SLAM 结果的 RANSAC 滤波器。

和第三章在图像平面利用带有深度信息的二维特征类似，需要在所有二维图像提取的二维特征点线中保留带有三维深度信息的特征。如图5.3所示，有了二维图像特征和前文用深度神经网络提取的平面区域之后，就可以得到一些潜在的特征共面集合。

得到潜在的特征共面集合之后，需要做的就是如何在这些集合中找到正确的共面信息。本文提出的找到正确共面信息的方法步骤如下：

1) 潜在特征共面集合的滤波。在得到这些特征共面集合之后，只保留集合内特征数量足够多的集合。因为在神经网络做图像平面区域分割的时候有一定概率会分割出来一些跨平面的小区域，或者在不容易提取特征的区域，这一步就是滤除这些不容易判断的区域。

2) 平面集合内部信息滤波。在得到一些大的带有深度的特征共面集合之后，需要对集合内的元素进行滤波。因为在神经网络做平面区域提取时有一定概率会使得提出的平面区域超出真实的平面区域一些像素。这一步就是为了滤除这些超出真实平面区域的像素上的特征。

所有在同一区域的空间特征点  $C_x$  和线  $C_l$  都会被加入一个 RANSAC 滤波器中。这个 RANSAC 滤波器的离群点计算准则如下：

$$f(c, \Gamma) = \begin{cases} \delta_{\perp}(P_x, \Gamma), z \in C_x \\ \max(\delta_{\perp}(c_{ls}, \Gamma), \delta_{\perp}(c_{le}, \Gamma)), z \in C_l \end{cases} \quad (5.1)$$

其中， $\delta_{\perp}(\cdot, \cdot)$  表示的是三维空间特征到空间平面的距离。对于点特征来说，该距离就是点到平面的最短距离。对于直线特征来说则是空间线段两个端点到平面的距离的最大值。需要注意的是，一条空间直线必须两个端点同时处于一个空间区域才会被计算。在通过滤波器之后，就可以得到空间点线的共面信息，同时在空间中可以得到一个潜在空间平面区域和平面信息。找到空间平面之后，每当 SLAM 系统生成新的三维点线之后都使用 5.1 标准检测是否以及在现有的平面之上。

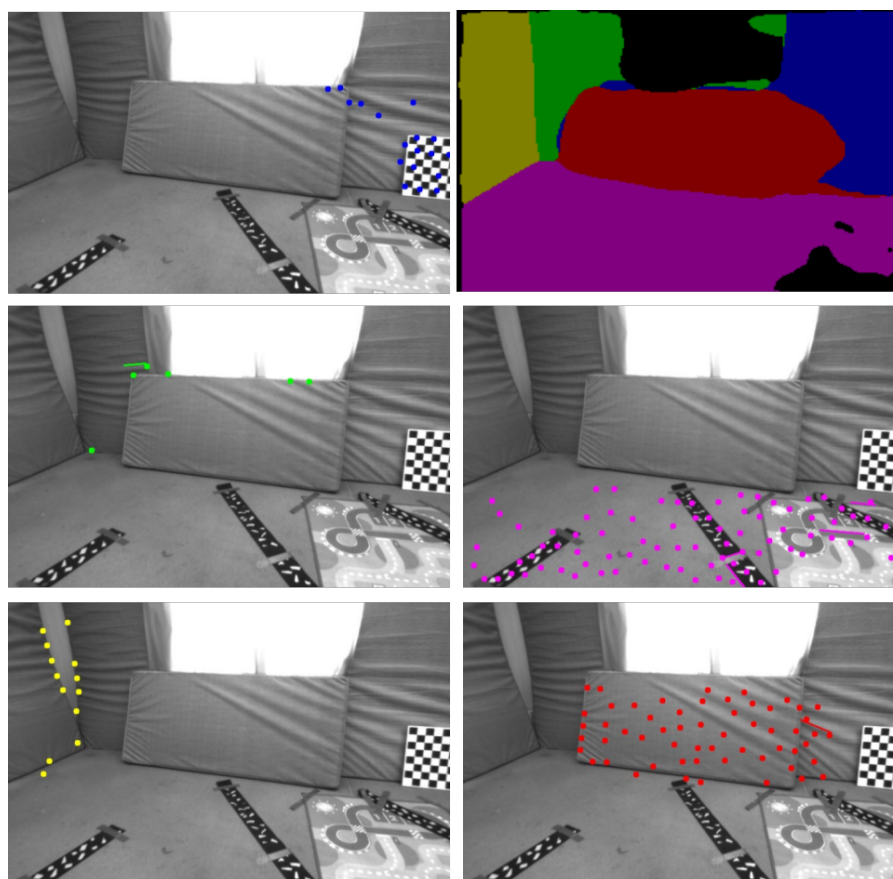


图 5.3 二维点线分类

## 5.2 共面点线参数化方法

上文中实现了在优化中直接加入点线特征的共面约束，但是直接在优化中添加共面约束会使得优化问题相较于原来没有共面约束的优化问题更加复杂。首先是优化问题的待优化参数会增加，除了直线和平面参数化之外，需要把平面特征也参数化才可以在优化问题中表示共面约束。然后是优化问题中的优化项会变多，显然的，需要把共面约束项添加到优化问题中才能使得优化问题中包含共面约束。但是如果可以在参数化的层面就把那些已经检测出来共面的点线就用一种新型参数化方法，并且这种参数化方法可以把共面约束融合进参数化本身，那么就可以在不改变优化问题的同时把共面约束添加进系统中，同时如果这种参数化方法的参数比原来的参数量要少的话还会使得优化问题变得更加简单。所以本文就提出了这种共面约束的点线的参数化方法。

### 5.2.1 共面点特征的参数化

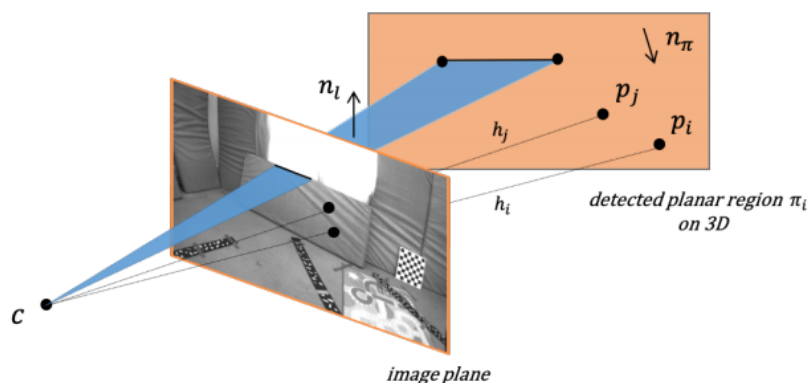


图 5.4 空间点线观测信息和平面相交示意图

在得到点特征的共面信息之后，就可以很方便的把这些共面的点特征用平面信息进行二次参数化。如5.4所示，平面上的三维空间的特征点可以由相机的观测射线（即由相机光心出发经点的二维观测发出的射线）和平面的空间位置相交得到。

由逆深度参数化<sup>[48]</sup>可知，对于已知三维特征点的图像测量信息的情况下，知道三维特征点的深度信息就可以重建出空间三维特征。

假设有在平面  $\pi$  上的某一点  $P$  在相机坐标系下的坐标为  $\mathbf{P} = [x^c, y^c, z^c]$ 。相机对该点的观测为  $(u, v)$ ，则该点观测的在归一化平面上的坐标是

$$(\hat{x}, \hat{y}, 1)^T = \mathbf{K}^{-1}(u, v, 1) \quad (5.2)$$

其中  $\mathbf{K}$  是相机内参矩阵。

在知道  $(x^c, y^c, z^c) = (\hat{x}, \hat{y}, 1) \cdot h_i$  和平面参数是  $\mathbf{n}_\pi^T P_x^c + d_\pi = 0$  的情况下

$$h_i \cdot \mathbf{n}_\pi^T K^{-1}(u, v, 1)^T + d_\pi = 0 \quad (5.3)$$

通过上述计算方法，可以直接从平面参数方程中计算出在平面上的点特征观测的深度信息。

### 5.2.2 共面直线特征参数化

共面直线特征参数化的方法比共面点特征参数化要更为直观。本文在第三章对空间直线特征参数化进行了介绍，对于直线特征来说，可以使用两个平面相交的办法使用对偶普吕克矩阵来得到空间表达。

$$\begin{aligned} \mathbf{L}^* &= \pi_1 \pi_2^\top - \pi_2 \pi_1^\top \in \mathbb{R}^{4 \times 4} \\ &= \begin{bmatrix} [\mathbf{d}]_\times & \mathbf{n} \\ -\mathbf{n}^\top & 0 \end{bmatrix} \end{aligned} \quad (5.4)$$

如图5.4所示，当已知空间中的平面  $\pi$  的参数，再由直线的观测平面（即由直线的观测线段端点和相机光心组成的平面）的参数，就可以通过公式5.4直接计算出直线的空间表达。

## 5.3 实验和分析

本章评估了本文提出的结合平面约束的点线视觉 SLAM 系统的表现，本章在仿真数据集和无人机采集的数据集上分别对本文提出的算法设计了实验。因为 SLAM 系统的精度主要由系统估计出的相机轨迹和真实的相机轨迹之间的偏差来判断。所以从系统定位的精度着手验证本章提出的共面参数化方法是可以将共面约束引入 SLAM 系统提高定位精度的，并且还可以和上一章提出的在优化中直接加入共面约束要更加高效。为了可以验证本文提出的各个模块的算法，本章提供了不同的算法组合来进行对比实验：

P(-wo): 系统中只利用点特征，同时没有共面约束

P(-r): 系统中只利用点特征，并带有共面约束，并使用第三章的方法在优化项中直接添加共面约束的共面约束优化方法一

P(-w): 系统中只利用点特征，并带有共面约束，并使用本章提出的方法使用共面参数化所有共面参数的共面约束优化方法二

PL(-wo): 系统利用了点特征和直线特征，同时没有共面约束



PL(-r): 系统利用了点特征和直线特征, 并带有共面约束, 并使用第三章的方法在优化项中直接添加共面约束的共面约束优化方法一

PL(-w): 系统利用了点特征和直线特征, 并带有共面约束, 并使用本章提出的新方法使用共面参数参数化所有共面参数的共面约束优化方法二

### 5.3.1 仿真实验

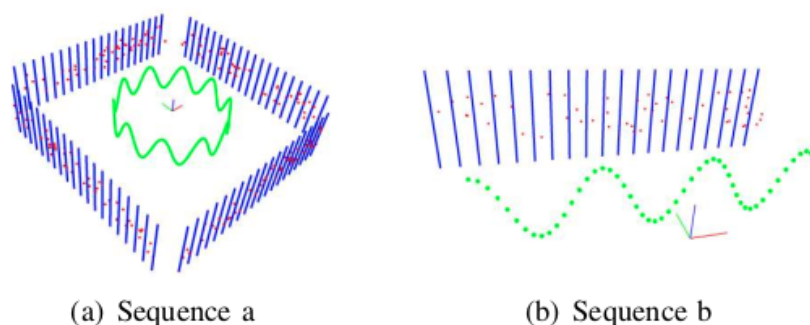


图 5.5 仿真实验的场景示意图 (直线特征-蓝色, 点特征-红色, 相机轨迹 (绿色))

设计仿真数据实验是为了验证前文所提出的不同模块在 SLAM 系统的后端优化中的作用, 所以仿真实验默认了所有空间点线图像观测的数据关联都是已知的, 空间点线的共面信息也是已知的, 并且提供了带有噪声的初始相机位姿。在保证这些和后端优化部分无关的变量都一致的情况下, 再使用同样的数据运行本章开始提出的六种不同的算法组合。为了使得实验结果不包含随机性, 所有仿真实验的结果都是进行 100 次蒙特卡洛实验产生的。

仿真实验生成的两个不同场景如图5.5所示。仿真实验场景 (a) 是一个边长为 8 米的正方形的房间, 房间中有分布在墙上的空间直线和点。相机在场景 (a) 中围着房间中心做圆周正弦运动。仿真实验场景 (b) 是一面长 8 米的墙, 墙上分布了空间直线和点, 相机在墙前方做正弦运动。

整个仿真实验生成的相机序列有 300 帧图像, 图像分辨率为  $640 \times 480$ , 每一帧图像大概可以观测到 15 个特征点和 8 个直线。每个观测到的特征点和直线的端点都会在精确的图像投影上被添加上方差为 1 个像素的高斯白噪声。仿真实验提供的相机初始位姿也是在正确位姿的基础上带有 1 度的旋转和 10cm 的高斯白噪声的结果。

仿真实验先由上文仿真实验场景生成的仿真实验数据序列, 然后再通过本章开始提出的不同算法组合优化得到实验结果。优化部分被看成是非线性最小二成问题, 采用高斯-牛顿优化算法进行求解。所有仿真实验中高斯-牛顿优化每次求解最多只经过 10 次迭代。



图5.7和图5.6是仿真实验在不同算法组合下经过蒙特卡洛实验得到的优化时间对比图表和优化结果 RMSE 对比图。

### 5.3.1.1 优化精度评测

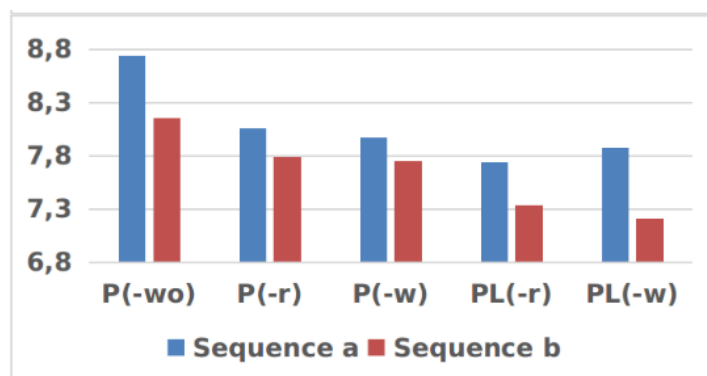


图 5.6 不同方法的仿真实验 RMSE 对比

从图5.6中可以看出如下结果：

1) 可以看出所有带有直线特征优化的 PL 方法精度要优于只有点特征的 P 方法。因为引入了直线的观测和优化，所以 PL 方法比 P 方法的精度要更好，这是符合预期的。

2) 不管是只有点特征的 P 方法和带有直线特征观测的 PL 方法在带有共面信息约束的 (-w) 和 (-r) 方法都比不带有共面约束的比 (-wo) 方法精度要更好。这也是符合预期的，因为加入了共面约束，最小二乘优化中的优化项更多了，优化最后优化结果更好了。

3) P 方法和 PL 方法在带有共面约束的 (-w) 和 (-r) 方法上都有着差不多同等的精度表现。这是因为两种添加平面约束的方法虽然途径不同，但是两者都是引入了相同的共面约束信息到整体的优化中。

### 5.3.1.2 优化时间评测

从图5.7中可以得出如下结果：

1) 不管是只有点特征的 P 方法和带有直线特征观测的 PL 方法中使用加入平面约束方法一的 (-r) 方法相较不添加共面约束的方法 (-wo) 的优化时间也都要更多。原因和结果 1) 类似，同样是因为在优化中添加了平面的参数使得待优化参数和优化项增加了，所以优化时间增加了。

2) 不管是只有点特征的 P 方法和带有直线特征观测的 PL 方法中的两种不同的添加共面约束的方法的优化时间要相差很多。因为 (-r) 方法使用了共面特征参数化参数的点和直线，使得待优化的参数量大幅度减少，所以优化时间大规模减少了。

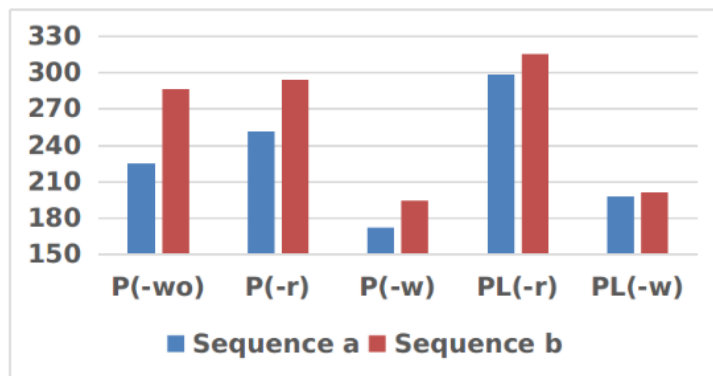


图 5.7 不同方法的仿真实验优化时间对比

对于结果 2) 中两种不同共面约束添加方法造成的待优化参数数量的差别, 使用仿真场景 (b) 来仔细分析待优化参数数量的区别。

表 5.1 仿真场景 (b) 中不同算法在优化中的待优化参数数量

	P (-wo)	P (-r)	P (-w)	PL (-wo)	PL (-r)	PL (-w)
待优化参数项	100	101	<b>51</b>	120	121	<b>51</b>
待优化参数规模	350	353	<b>303</b>	430	433	<b>303</b>

### 5.3.2 数据集验证以及分析

本章测试使用的数据集是第三章介绍过的 EuRoC 数据集。由于本章提出的共面参数化和平面提取方法有一定的普适性, 所以本章的测试将在双目视觉 SLAM 中和 VIO 中共同进行。

#### 5.3.2.1 双目视觉 SLAM 序列

对于双目视觉 SLAM 序列, 本章选择在 ORB-SLAM2 的算法框架上进行本章算法的实现。但是 ORB-SLAM2 只包含点特征, 所以本章还会使用 LSD 和 LBD 在前端进行直线特征的提取和匹配, 以及运用第三章的知识对直线特征进行三角化和加入优化。最终本章构建的双目 SLAM 的整体优化代价函数如下:

$$E = \sum_{k,i} \rho_p(\mathbf{r}_{ik}^p \top \Lambda_{ik} \mathbf{r}_{ik}^p) + \sum_{k,j} \rho_l(\mathbf{r}_{jk}^l \top \Lambda_{jk} \mathbf{r}_{jk}^l) \quad (5.5)$$

其中  $\rho_p$  和  $\rho_l$  代表鲁棒 Cauchy 核函数。  $\Lambda_{ik}$  和  $\Lambda_{jk}$  分别是点特征和直线特征的信息矩阵。  $\mathbf{r}_{ik}^p$  和  $\mathbf{r}_{jk}^l$  分别是点特征的重投影误差和直线特征的重投影误差。

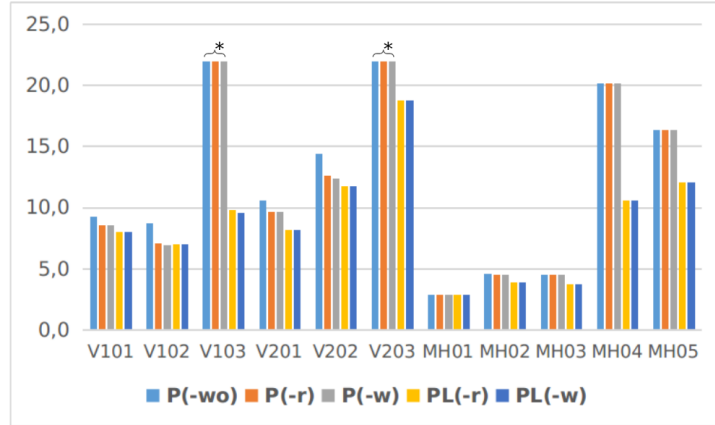


图 5.8 双目序列不同方法的仿真实验优化时间对比

### 5.3.2.2 VIO 序列

对于 VIO 序列，本章选择 PL-VIO 作为算法的框架基础。由于 PL-VIO 本身以及有了点线特征的提取匹配以及优化，所以本文只需要将平面检测和参数化加入就可以完成算法。本节介绍使用共面约束的 VIO 系统的整体优化代价函数如下：

$$\begin{aligned}
 E = & \sum_{k,i} \rho_p(\mathbf{r}_{ik}^p \top \Lambda_{ik} \mathbf{r}_{ik}^p) + \sum_{k,j} \rho_l(\mathbf{r}_{jk}^l \top \Lambda_{jk} \mathbf{r}_{jk}^l) \\
 & + \sum_b \rho_l(\mathbf{r}^b \top \Lambda_b \mathbf{r}^b) + E_m
 \end{aligned} \tag{5.6}$$

其中  $\mathbf{r}^b$  是 IMU 预积分测量误差， $E_m$  是边缘化操作产生的先验误差。和 5.5 中的字母代表含义一样， $\rho_p$  和  $\rho_l$  代表鲁棒 Cauchy 核函数。 $\Lambda_{ik}$  和  $\Lambda_{jk}$  分别是点特征和直线特征的信息矩阵， $\mathbf{r}_{ik}^p$  和  $\mathbf{r}_{jk}^l$  分别是点特征的重投影误差和直线特征的重投影误差。

### 5.3.2.3 定位精度

本部分对算法的定位精度的评测分成了两个部分：本章提出算法的不同部分的对比实验；和其他先进算法的对比实验。对于定位精度，本章使用绝对轨迹精度 (Absolute Trajectory Error, ATE) 来进行比较。ATE 是指轨迹真值和估计轨迹之间的绝对平移距离。

首先进行本章提出算法的不同部分的对比实验。

EuRoC 数据集上双目和 VIO 版本的不同方法的结果如图 5.8 和 5.9 所示。可以从图中看出，和使用在优化中平面约束的方法、普通没有平面约束的点特征方法相比，使用本章提出的参数化方法引入共面约束的方法能得到更低的 RMSE。在 MH 序列中，直线特征比点特征要更容易找到平面，使得 MH 序列带有直线特征的平面约束带来的降低要更多。

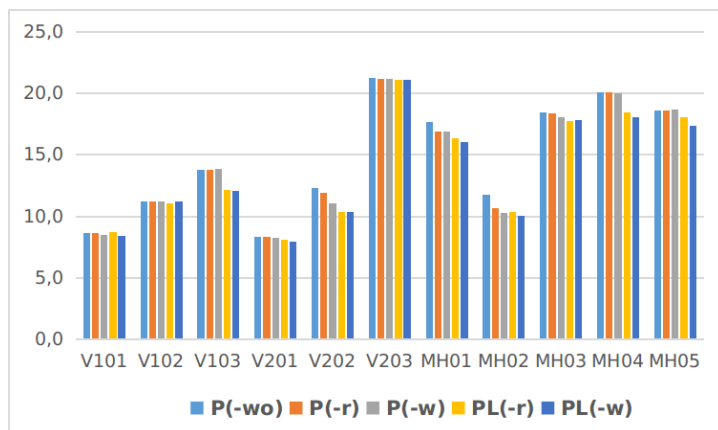


图 5.9 VIO 序列不同方法的仿真实验优化时间对比

表 5.2 不同 VIO 算法的 RMSE (cm) 结果

	MSCKF	ROVIO	VINS-Mono	Kimera-VIO	PL(-wo)	PL(-r)	PL(-w)
V101	34	10	7	<b>6</b>	8.4	8.5	8.4
V102	20	10	10	<b>7</b>	11.0	10.9	11.0
V103	67	14	13	17	<b>11.9</b>	<b>11.9</b>	<b>11.9</b>
V201	10	12	<b>8</b>	<b>8</b>	8.1	8.1	<b>8.0</b>
V202	16	14	<b>8</b>	10	12.0	10.5	10.5
V203	113	<b>14</b>	21	27	20.9	20.9	20.9
MH01	42	21	27	<b>14</b>	17.1	16.3	16.2
MH02	45	25	12	13	11.0	10.2	<b>10.0</b>
MH03	23	25	<b>13</b>	21	17.6	17.6	17.6
MH04	37	49	23	22	18.5	18.4	<b>18.2</b>
MH05	48	52	35	23	18.2	18.0	<b>17.7</b>
平均	41.3	22.3	16.0	15.2	14.1	13.8	<b>13.7</b>

注：粗体表示最佳结果

对于双目序列，从图5.8可以看出，直线特征使得系统整体变得更加鲁棒，特别实在 V103 和 V203 序列中，因为在这两个序列中发生了严重的运动模糊使得点特征不能很好的跟踪。在某些序列中使用本章的共面参数化的方法性能要稍微优于使用第三章在优化中添加共面约束的方法，这可能是因为在本章提出的共面参数化使得共面特征和平面直接的距离直接去除了。在纹理比较丰富的 MH01、MH02 和 MH03 序列中，所有方法都得到了相似的结果。

从图5.9中，可以看出在 VIO 序列的结果中，MH03、MH04 和 MH05 序列上共面约束都没有起到很有效的作用，是因为没有检测到足够的共面信息。当可以检测到共面信息的时候，比如在 V202、MH01 和 MH02 序列上，共面信息为算法精度提供了一些帮助。

在做完本章算法自身模块的对比实验之后，本章把本章提出的算法和 ORB-SLAM2<sup>[16]</sup> 和 FMD-SLAM<sup>[49]</sup> 这两个双目 SLAM，MSCKF<sup>[24]</sup>、ROVIO<sup>[50]</sup>、VINS-MONO<sup>[21]</sup> 以及

表 5.3 不同双目 SLAM 算法的 RMSE (cm) 结果

	ORB-SLAM2	FMD-SLAM	PL(-wo)	PL(-r)	PL(-w)
V101	9	9	8.4	<b>8.0</b>	<b>8.0</b>
V102	8	20	7.5	<b>7.0</b>	<b>7.0</b>
V103	20	53	10.6	9.8	<b>9.6</b>
V201	<b>7</b>	9	9.0	8.2	8.2
V202	10	<b>8</b>	12.4	11.8	11.8
V203	×	×	19.8	<b>18.8</b>	<b>18.8</b>
MH01	4	4	<b>2.9</b>	<b>2.9</b>	<b>2.9</b>
MH02	5	4	4.1	<b>3.9</b>	<b>3.9</b>
MH03	4	5	4.0	<b>3.7</b>	<b>3.7</b>
MH04	16	<b>9</b>	10.6	10.6	10.6
MH05	20	<b>9</b>	12.1	12.1	12.1
平均	10.6*	12.9*	8.1*	<b>7.8*</b>	<b>7.8*</b>

注：粗体表示最佳结果，带有 \* 号的平均结果不包括序列 V203

Kimera-VIO<sup>[51]</sup> 这些 VIO 算法进行了比较。

表5.2展示的是 VIO 序列部分和其他先进的比较。结果表明  $PL-w$  是一种可以和先进算法媲美的鲁棒而且有高精度的算法。和 Kimera-VIO<sup>[51]</sup>（该方法在优化过程中也使用平面信息建立共面规则）相比，本章的方法在大多数序列上表现更好，其中 Kimera-VIO 和本文第四章类似在平面检测过程中使用重力辅助，从三维网格中获得了更多的垂直平面。当在 V103 和某些 MH 序列中水平面和垂直面难以检测时，Kimera-VIO 容易退化成普通的 VIO，所以无法建立共面约束。在序列 MH05 中，可以看出本章算法比性能第二好的算法（Mesh-VIO）相比提高了 26%，在序列 V103 中，与 VINS-MONO 和 Kimera-VIO 相比分别提高了 15% 和 35%。可以看出，VINS-MONO、Kimera-VIO 和  $PL-w$  的优化方法比基于滤波器的 MSCKF 具有更强的鲁棒性。同时，本章提出的方法对于具有大量共面规律的室内环境更具鲁棒性。

双目 SLAM 和其他先进算法的比较在表5.3中。V203 是一个困难的序列，因为快速运动和强烈的光照变化，使得 ORB-SLAM2 和 FMD-SLAM 跟踪都失败。但受益于同时使用了点和线特征，本章提出的方法可以处理这个序列。在不考虑序列 V203 的情况下计算的公平性平均 RMSE 值表明，与 ORB-SLAM2 和 FMD-SLAM 相比，本章提出的方法分别获得了 25.7% 和 38.7% 的改进。

#### 5.3.2.4 实时性评测

表5.4展示了不同算法序列的 VIO 在 V101 上不同模块的运行时间比较。可以看出在后端加入优化的 (-r) 序列在 P 和 PL 方法上都比 (-wo) 要耗时更多。而本章提出的 (-w) 方法的优化时间相较于 (-r) 方法要更少。可以证明本章提出的平面参数化方

法的确在引入共面约束的情况下降低了优化时间。

表 5.4 EuRoc V101 序列中不同操作的计算时间 (单位: 毫秒)

	P(-wo)	P(-r)	P(-w)	PL(-wo)	PL(-r)	PL(-w)
点特征提取匹配	4	4	4	4	4	4
直线特征提取匹配	-	-	-	96	96	96
平面实例生成	-	29	29	-	29	29
平面检测	-	10	10	-	10	10
后端优化	43	44	40	36	46	42
总计	56	60	55	57	58	54

注: \* 表示该操作作用于每个帧, 否则仅对关键帧执行。-表示不使用该操作。

## 5.4 本章小节

本章提出了直接使用平面参数来参数化平面上的点线特征的方法。首先介绍了仅使用相机观测结合深度学习的平面实例分割结果的空间平面特征和共面信息提取方法。接着介绍了本章提出的仅仅使用空间平面参数化平面上的共面点线的参数化方法。最后本章设计了仿真实验和真实数据集实验来验证本章提出的参数化方法的有效性。实验验证了本章提出的参数化方法可以有效的将共面信息加入 SLAM 系统中, 并相较于第三章的方法可以减少优化时间。

## 第六章 总结与展望

### 6.1 总结

SLAM 技术是机器人实现自动移动的基础技术，本文主要关注于将空间结构化信息引入 SLAM 系统中。现有的 SLAM 技术大多只利用了点特征或者少量的直线特征，没有把结构信息最多的平面特征和共面约束添加进系统中。本文主要工作是将空间中直线特征以及空间点线特征的共面信息加入 SLAM 系统中。本文的工作主要体现在两个方面：1. 使用视觉传感器以及结合 IMU 传感器来检测空间中的结构化信息；2. 将检测到的结构化信息加入到现有的 SLAM 系统中。

本文主要工作有：

#### 1. 融合共面信息的点线视觉惯性里程计

- 提出了一种基于 VIO 产生的稀疏空间点线实时进行半稠密网格重建的算法，并且该算法可以检测出环境中的空间平面以及平面上点线的共面约束。
- 针对室内结构化信息丰富的场景，构建了一个同时利用空间点线以及它们共面信息的 VIO 系统。这个系统可以结合 IMU 观测、点线特征的视觉观测以及点线特征的空间共面约束来一同估计系统的位姿和空间地图。并且通过实验证明由于直线和共面信息的加入，系统对于位姿估计和空间地图的精度都得到了提升。

#### 2. 共面点线的平面参数化

- 为 SLAM 系统提出了一种基于图像上的平面实例分割结果和简单的 RANSAC 滤波器鲁棒的提取共面约束的方法，使得室内视觉 SLAM 系统可以便捷地把共面这种结构化信息加入 SLAM 系统中。
- 提出了一种新的点线共面信息的参数化方法，将共面约束在参数化的层面上就加入 SLAM 系统中，并且该种参数化方法可以便捷地引入已经检测出共面信息的系统中。最后通过实验证明了相较于直接将共面信息放在优化中，本文提出的共面参数化方法可以在保持估计精度不下降的前提下，降低系统的优化时间。

### 6.2 展望

本文的研究内容主要是针对空间结构信息与 SLAM 系统进行融合的初步试探性工作。但在空间信息的应用以及 SLAM 系统的改进上还有很多方面需要继续探索。未来还可以考虑从下面几个方面进行改进：

1. 将 SLAM 与深度学习在更多方面进行结合。基于传统点和直线的图像检测算法

存在一定的局限性，伴随着深度学习特征检测的高速发展，深度学习的点和直线的检测匹配的成功率慢慢地超过了传统算法。所以在图像前后变化剧烈的情况下，未来可以考虑利用深度学习检测来匹配点和直线特征，SLAM 系统的鲁棒性和精度会进一步提升。

2. 将更多的结构化特征加入 SLAM 系统之中。因为本文提出的算法是考虑到室内的结构化信息，除了直线和平面这种结构化特征，还有其他一些常见的室内结构化信息，比如相交直线等。随这近些年这些高阶特征的检测和匹配算法的成熟，如果可以把这些特征也添加入 SLAM 系统中，SLAM 系统在室内结构化信息丰富的情况下，系统的鲁棒性和精度还会进一步地提升。

3. 将结构化信息在回环检测中进行应用。回环检测是 SLAM 系统构建全局一致的地图和抑制大尺度场景下的漂移的首选。现有的回环检测算法还没有很好地利用空间结构信息，大多只是使用普通的图像点线检测来进行回环，如何充分利用这些空间结构信息也是一个值得思考的方向。



## 参考文献

- [1] BAILEY T, DURRANT-WHYTE H. Simultaneous localization and mapping (SLAM): Part II[J]. IEEE robotics & automation magazine, 2006, 13(3): 108-117.
- [2] DURRANT-WHYTE H, BAILEY T. Simultaneous localization and mapping: part I[J]. IEEE robotics & automation magazine, 2006, 13(2): 99-110.
- [3] STRASDAT H, MONTIEL J, DAVISON A J. Scale drift-aware large scale monocular SLAM[J]. Robotics: Science and Systems VI, 2010, 2(3): 7.
- [4] DAVISON A J. Real-time simultaneous localisation and mapping with a single camera[C]//Computer Vision, IEEE International Conference on: vol. 3. [S.l. : s.n.], 2003: 1403-1403.
- [5] DAVISON A J, REID I D, MOLTON N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(6): 1052-1067.
- [6] LOWE D G. Object recognition from local scale-invariant features[C]//Proceedings of the seventh IEEE international conference on computer vision: vol. 2. [S.l. : s.n.], 1999: 1150-1157.
- [7] RUBLEE E, RABAU D V, KONOLIGE K, et al. ORB: An efficient alternative to SIFT or SURF[C]//2011 International conference on computer vision. [S.l. : s.n.], 2011: 2564-2571.
- [8] ROSTEN E, DRUMMOND T. Machine learning for high-speed corner detection[C]//European conference on computer vision. [S.l. : s.n.], 2006: 430-443.
- [9] BAKER S, MATTHEWS I. Lucas-kanade 20 years on: A unifying framework[J]. International journal of computer vision, 2004, 56(3): 221-255.
- [10] SHI J, et al. Good features to track[C]//1994 Proceedings of IEEE conference on computer vision and pattern recognition. [S.l. : s.n.], 1994: 593-600.
- [11] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces[C]//2007 6th IEEE and ACM international symposium on mixed and augmented reality. [S.l. : s.n.], 2007: 225-234.
- [12] MUR-ARTAL R, MONTIEL J M M, TARDOS J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE transactions on robotics, 2015, 31(5): 1147-1163.
- [13] NEWCOMBE R A, LOVEGROVE S J, DAVISON A J. DTAM: Dense tracking and mapping in real-time[C]//2011 international conference on computer vision. [S.l. : s.n.], 2011: 2320-2327.
- [14] ENGEL J, SCHÖPS T, CREMERS D. LSD-SLAM: Large-scale direct monocular SLAM[C]//European conference on computer vision. [S.l. : s.n.], 2014: 834-849.
- [15] ENGEL J, KOLTUN V, CREMERS D. Direct sparse odometry[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(3): 611-625.
- [16] MUR-ARTAL R, TARDÓS J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.
- [17] STRASDAT H, MONTIEL J M, DAVISON A J. Visual SLAM: why filter?[J]. Image and Vision Computing, 2012, 30(2): 65-77.
- [18] CORKE P, LOBO J, DIAS J. An introduction to inertial and visual sensing[Z]. 2007.

- [19] SCARAMUZZA D, ACHELNIK M C, DOITSIDIS L, et al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments[J]. *IEEE Robotics & Automation Magazine*, 2014, 21(3): 26-40.
- [20] WEISS S, SIEGWART R. Real-time metric state estimation for modular vision-inertial systems[C]// 2011 IEEE international conference on robotics and automation. [S.l. : s.n.], 2011: 4531-4537.
- [21] QIN T, LI P, SHEN S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. *IEEE Transactions on Robotics*, 2018, 34(4): 1004-1020.
- [22] CAMPOS C, ELVIRA R, RODRIGUEZ J J G, et al. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM[J]. *ArXiv preprint arXiv:2007.11898*, 2020.
- [23] VON STUMBERG L, USENKO V, CREMERS D. Direct sparse visual-inertial odometry using dynamic marginalization[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). [S.l. : s.n.], 2018: 2510-2517.
- [24] MOURIKIS A I, ROUMELIOTIS S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]//Proceedings 2007 IEEE International Conference on Robotics and Automation. [S.l. : s.n.], 2007: 3565-3572.
- [25] LUPTON T, SUKKARIEH S. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions[J]. *IEEE Transactions on Robotics*, 2011, 28(1): 61-76.
- [26] FORSTER C, CARLONE L, DELLAERT F, et al. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry[J]. *IEEE Transactions on Robotics*, 2016, 33(1): 1-21.
- [27] NEIRA J, RIBEIRO M I, TARDÓS J D, et al. Mobile robot localization and map building using monocular vision[C]//In The 5th Symposium for Intelligent Robotics Systems. [S.l. : s.n.], 1997.
- [28] SOLA J, VIDAL-CALLEJA T, CIVERA J, et al. Impact of landmark parametrization on monocular EKF-SLAM with points and lines[J]. *International journal of computer vision*, 2012, 97(3): 339-368.
- [29] BARTOLI A, STURM P. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment[J]. *Computer vision and image understanding*, 2005, 100(3): 416-441.
- [30] ZHANG G, LEE J H, LIM J, et al. Building a 3-D line-based map using stereo SLAM[J]. *IEEE Transactions on Robotics*, 2015, 31(6): 1364-1377.
- [31] HE Y, ZHAO J, GUO Y, et al. PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features[J]. *Sensors*, 2018, 18(4): 1159.
- [32] YANG Y, GENEVA P, ECKENHOFF K, et al. Visual-inertial odometry with point and line features[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.l. : s.n.], 2019: 2447-2454.
- [33] LEE T K, LIM S, LEE S, et al. Indoor mapping using planes extracted from noisy RGB-D sensors[C]//2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. [S.l. : s.n.], 2012: 1727-1733.
- [34] TREVOR A J, ROGERS J G, CHRISTENSEN H I. Planar surface SLAM with 3D and 2D sensors[C]//2012 IEEE International Conference on Robotics and Automation. [S.l. : s.n.], 2012: 3041-3048.

- 
- [35] KAESS M. Simultaneous localization and mapping with infinite planes[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). [S.l. : s.n.], 2015: 4605-4611.
- [36] LU Y, SONG D. Visual navigation using heterogeneous landmarks and unsupervised geometric constraints[J]. IEEE Transactions on Robotics, 2015, 31(3): 736-749.
- [37] KUMARID, KAUR K. A survey on stereo matching techniques for 3D vision in image processing[J]. Int. J. Eng. Manuf, 2016, 4: 40-49.
- [38] BARFOOT T D. State estimation for robotics: A matrix lie group approach[J]. Draft in preparation for publication by Cambridge University Press, Cambridge, 2016.
- [39] MURRAY R M, LI Z, SASTRY S S, et al. A mathematical introduction to robotic manipulation[M]. [S.l.]: CRC press, 1994.
- [40] VON GIOI R G, JAKUBOWICZ J, MOREL J M, et al. LSD: A fast line segment detector with a false detection control[J]. IEEE transactions on pattern analysis and machine intelligence, 2008, 32(4): 722-732.
- [41] WANG Z, WU F, HU Z. MSLD: A robust descriptor for line matching[J]. Pattern Recognition, 2009, 42(5): 941-953.
- [42] ZHANG L, KOCH R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency[J]. Journal of Visual Communication and Image Representation, 2013, 24(7): 794-805.
- [43] ANDREW A M. Multiple view geometry in computer vision[J]. Kybernetes, 2001.
- [44] CHEW L P. Constrained delaunay triangulations[J]. Algorithmica, 1989, 4(1): 97-108.
- [45] FISHLER M A. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography[J]. Commun, ACM, 1981, 24(6): 381-395.
- [46] NARDIF, DELLA CORTE B, GRISETTI G. Unified representation and registration of heterogeneous sets of geometric primitives[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 625-632.
- [47] SCHOPS T, SCHONBERGER J L, GALLIANI S, et al. A multi-view stereo benchmark with high-resolution images and multi-camera videos[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l. : s.n.], 2017: 3260-3269.
- [48] CIVERA J, DAVISON A J, MONTIEL J M. Inverse depth parametrization for monocular SLAM[J]. IEEE transactions on robotics, 2008, 24(5): 932-945.
- [49] TANG F, LI H, WU Y. Fmd stereo slam: Fusing mvg and direct formulation towards accurate and fast stereo slam[C]//2019 International Conference on Robotics and Automation (ICRA). [S.l. : s.n.], 2019: 133-139.
- [50] BLOESCH M, OMARI S, HUTTER M, et al. Robust visual inertial odometry using a direct EKF-based approach[C]//2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). [S.l. : s.n.], 2015: 298-304.
- [51] ROSINOL A, SATTLER T, POLLEFEYS M, et al. Incremental visual-inertial 3d mesh generation with structural regularities[C]//2019 International Conference on Robotics and Automation (ICRA). [S.l. : s.n.], 2019: 8220-8226.



## 攻读硕士期间发表论文

**Xin Li**, Yijia He, Jinlong Lin, and Xiao Liu. "Leveraging planar regularities for point line visual-inertial odometry." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020.

**Xin Li**, Yanyan Li, Evin Pnar Örneke, Jinlong Lin, and Federico Tombari. "Co-Planar Parametrization for Stereo-SLAM and Visual-Inertial Odometry." IEEE Robotics and Automation Letters 5, no. 4 (2020): 6972-6979.



## 致谢

行文至此，心中只有感恩。

感谢我的导师林金龙教授对我的精心教导。从刚入学开始对我研究方向的探讨我就指导林老师是一位真心为学生着想的好老师。后来林老师也经常在组会上分享一些他的人生经验和感悟，让我收获良多。林老师也对我的论文和写作技巧做出过非常多帮助和指导。我深深为我研究生阶段可以遇到林老师这样的导师感到幸运。

感谢我们导师组的冯泽邦，王沛峰，闫妍，何人天，周子鹏，张岩。虽然大家因为只在学校住宿一年后续接触不是很多，但是大家一起开组会讨论一起嬉闹的日子我会一直铭记。

感谢在旷视 SLAM 组的 mentor 和同事们。贺一家博士在我刚入门 SLAM 不知所以时带领我进入 SLAM 研究领域的大门，让我可以在 SLAM 的研究领域中找到属于自己的一片天地。也感谢在 SLAM 组中教会我很多知识的刘晓、柴政、肖锡臻。还有一起工作的小伙伴们。

感谢泡泡机器人的一众小伙伴们。从我加入泡泡机器人就对我进行一系列指导的蔡纪源博士，到后续认识一直一起搭档工作的万应才博士他们都一直对我照顾。感谢泡泡机器人的图灵智库板块，从组员到组长，这个板块伴随了我的研究生生涯。还要感谢泡泡机器人很多平时一起沟通讨论的小伙伴们。

感谢在北大遇到的一众伙伴。从和我一起入学一起住一个宿舍的康欣欣，到后来同住一个宿舍的王源和张尧，后续因羊汤结缘的舒展，陈晨，耿泽浩，徐博。以及最后一年在一起租房的刘家怿。感谢大家在研究生三年对我的照顾，我会一直怀恋大家一起吃饭喝羊汤的日子。

感谢一路带我科研的李言言师兄，师兄不仅在学术上给我进行一系列的指导还在生活上传授了我很多生活的感悟，这一些都让我收货颇多。

感谢我相恋 6 年的女朋友刘梦冰。是你一路陪伴我走过来，尽管我身上有着各种各样的小毛病，你还是不断包容我理解我。也感谢你在我心情低迷的时候给我的鼓励和帮助，让我有了直面苦难的勇气。还要感谢你对我学业的支持，本来我们有机会可以不需要天南海北的相隔，感谢你支持我来北大学习。

最后感谢我的父母和家人们。感谢我的父亲和母亲，你们一直在我背后支持我，我犯下什么错误时也都包容我。我这辈子最大的财富就是你们教会给我的善良。感谢你们这么多年来对我的养育之恩，一直为我提供优良的学习和生活环境。





