SLAM 中的几何与学习方法

Geometric Approaches and Neural Networks for SLAM Systems

李言

Federico Tombari

updated 2024 年 7 月 16 日

[7]

说明：文章中的实验结果图片，除特殊说明以外，均来自我们自己的实验。读者可以引用，引用时，请说明出处。

```
@misc{gl-SLAM,
author = {Yanyan Li},
title = {SLAM中的几何与学习方法},
year = {2020},
publisher = {GitHub},
howpublished = {\url{https://github.com/yanyan-li/SLAM-BOOK}
}
```

Part I

SLAM 简史 (Brief history of SLAM)

1 SLAM 问题的提出与定义 (Problem definition of SLAM)

几十年前，站在这一学科的开端，没有人知道如何通过拍摄多张照片同时跟踪相机姿势并重建未知场景。经过社区这么长时间的共同努力，通过不同策略的 SLAM 系统已经达到了最初的目标。然而回过头来看，这棵大树的生长脉络却更加清晰。让我们从这个问题的定义开始，逐步扩展这个领域的复杂性。

对于 SLAM 主题定义的讨论，对于如何呈现 SLAM 问题的核心思想有很多不同的观点，但使用概率表示无疑是最优雅的方式之一。在接下来的章节，我们从状态估计的概论模型出发，通过概论模型中涉及到的变量进行分析。

1.1 状态估计的概论模型与非线性最小二乘

具有 n 相机姿势 ($\mathcal{T} = \{\mathbf{T}_1, \ldots, \mathbf{T}_n\}$) 的图像序列记录在具有 m 个地标的环境中 ($\mathcal{P} = \{\mathbf{P}_1, \ldots, \mathbf{P}_m\}$)。在跟踪过程中，测量值 ($\mathcal{Q} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$) 用于构建条件概率分布模型，如下所示：

$$P(\mathcal{T}, \mathcal{P}|\mathcal{Q}) \tag{1.1}$$

它展示了如何根据新的测量值更新相机姿势和场景表示。通过贝叶斯概率运算，问题可以进一步转化为似然概率模型

$$\begin{aligned} P(\mathcal{T}, \mathcal{P}|\mathcal{Q}) &= \frac{P(\mathcal{Q}|\mathcal{T}, \mathcal{P})P(\mathcal{T}, \mathcal{P})}{P(\mathcal{Q})} \\ &\propto P(\mathcal{Q}|\mathcal{T}, \mathcal{P})P(\mathcal{T}, \mathcal{P}) \end{aligned} \tag{1.2}$$

这里 $P(\mathcal{Q}|\mathcal{T}, \mathcal{P})$ 被称为似然，它可以理解为在当前的环境和位姿下产此测量的概率。$P(\mathcal{T}, \mathcal{P})$ 被称为先验，它可以理解为机器人处于在当前的环境和位姿的概率。与公式 1.1 中引入的后验概率相比，$P(\mathcal{Q}|\mathcal{T}, \mathcal{P})$ 为增量跟踪和映射主题提供了更好的视角，因为它计算新测量值与旧姿势 $\mathcal{T}$ 和表示 $\mathcal{P}$ 的估计值之间的差异。因此，我们可以很直观的感受到，计算最优位姿和 3D 路标的过程就是寻找位姿和路标信息最能产生当前观测的过程，可以通过下面的公式进行描述

$$(\mathcal{T}, \mathcal{P})^* = \arg\max_{\mathcal{T}, \mathcal{P}} P(\mathcal{Q}|\mathcal{T}, \mathcal{P}) \tag{1.3}$$

当我们的观测数据噪声的概论分布满足高斯分布，即 $\mathcal{Q} = f(\mathcal{Q}|\mathcal{T}) + \mathbf{n}$, 其中 $\mathbf{n} \in \mathcal{N}(\mathbf{0}, \mu)$ 表示噪声 $\mathbf{n}$ 服从零均值（zero-mean）正太分布。那么条件概率 $P(\mathcal{Q}|\mathcal{T}, \mathcal{P})$ 的概率分布仍然服从于高斯分布于 $\mathcal{N}(f(\mathcal{Q}|\mathcal{T}), \mu)$。

如方程 1.2所示，该问题主要涉及四个部分，包括**测量**、**相机位姿**、**场景表示**和**误差收敛模型**。

1.2 观测（Measurements）

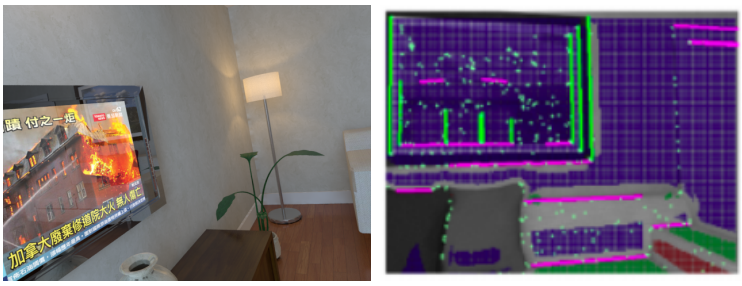

图 1: 从 RGB-D 图像中提取点、线、面特征，并对线面特征进行方向分类。

一般来说，仅提取不同点的方法可以称为特征点检测器，而通过手工规则生成抽象信息以呈现二维测量之间的差异的方法称为描述符。第一个重要任务是自动检测这些测量结果，以实现基于视频的跟踪和绘图性能。如图 1所示，点、线和平面测量是从一对 RGBD 图像中获得的。

Points. point 是 VO/SLAM 系统中最常用的特征，是图像中梯度变化剧烈的像素，如角点、边缘等。对于不同的点检测方法，1987 年提出的 FODPL [16] 和 1988 年提出的 Harris [18] 是打开现代计算机视觉帷幕的重要工具。给定选定的不同点，第一种策略是 Kanade-Lucas-Tomasi (KLT) 方法 [40]，提出通过基于相邻帧的恒定亮度假设定义光度误差来跟踪点。另一种更鲁棒的角点估计方法 FAST [37] 于 2008 年提出。

与只能用于匹配局部区域的颜色强度相比，通过 SIFT [1] (1999 年)、SURF [5]（2006 年）和 ORB [38]（2011 年），其中这些复杂的向量对于图像缩放、平移和旋转、甚至与颜色强度相比的照明变化保持不变，更加稳健。因此，它可以通过计算两个描述符之间的差异来支持全局匹配搜索策

略。特征提取一般包括特征点检测和描述符计算两个过程。描述符是一种衡量特征相似度的手段，用于确定不同图像中对应空间中的同一对象。

SIFT [1] 是最著名的传统局部特征点之一，它通过在图像的高斯差分（DoG）尺度空间表示中寻找局部最大值和最小值来识别候选关键点。然后，它通过拟合对缩放、旋转和仿射失真不变的模型来细化这些候选关键点。最后，它通过获取关键点周围局部邻域中梯度方向的直方图来计算每个关键点的描述符。FAST 是一种专为实时应用程序设计的高速角点检测方法。它的工作原理是检查中心像素周围圆圈上的连续像素，以确定中心像素是否是角点。如果足够数量的连续像素在强度上与中心像素相差很大，则中心像素被分类为角点。Brief 是一个二进制特征描述符，它将检测到的关键点的梯度信息编码为二进制字符串，对关键点周围的局部邻域中的点对的强度值进行采样，并使用二进制测试对它们进行比较。此过程为每个关键点生成描述其局部外观的二进制字符串。

Lines and Planes. 线路提取任务在早期也引起了社区的兴趣。Burns 等人 [7] 于 1986 年提出了一种检测直线的方法。虽然线性时间直线检测器 LSD [42] 和更快的直线检测方法 EDLines [2] ，分别于 2008 年和 2011 年提出。2013 年提出的描述符 LBD [43] 是一种广泛使用的匹配不同视图之间线段的算法。随着 RGB-D 传感器的发展，基于 RANSAC 的方法（如 [39]）被用于平面检测。此外，利用霍夫变换 [33] 来实现平面分割的预分割步骤。此外，2014 年提出了一种基于凝聚层次聚类算法的更快的平面检测方法 [13]。***Lines*** 广泛存在于人造环境中，它提供了比点特征更多的约束，并且在某些场合比点特征更鲁棒。***Planes*** 也在室内场景中广泛检测，例如建筑物内部和家具表面。与点和线相比，从单个 RGB 图像中提取它们是很困难的。基于 RANSAC 的策略在通过随机选择可能属于同一平面的点子集来从点云拟合平面方面很流行。AHC [14] 从一些初始种子点开始，并通过添加满足某些标准（例如距离和法线角度相似性）的相邻点来扩展每个区域。这些方法计算点云中每个点的法向量，并根据它们的方向将它们累积到箱中。得票最高的垃圾箱代表场景中的主导平面。

1.3 场景表征（Scene Reconstruction）

对于场景重建，稀疏方法将跟踪过程中检测到的对应关系三角化为稀疏地标，而密集系统旨在使用 2D 图像的所有像素。稀疏和稠密重建之间的方

法称为半稠密，其中跟踪点的数量比稀疏方法更重要，但并不是每个像素都可以像稠密重建方法那样用于建模。除了视觉稀疏性之外，另一个基本原则是这些地标在稀疏公式中被视为独立的。尽管借助线和面特征重建的地图可以非常密集和完整，但基于特征的系统的那些地图仍然被视为稀疏方法。

1.3.1 稀疏场景表示（Sparse Scene Parametrization）

XYZ 参数化对于表示点地标是直观的，但在低视差对应中面临挑战。逆深度参数化 [8] 使用射线通过锚定帧位置和观察到的特征。对于线和平面地标的直接表示方法分别是欧几里得 XYZ 端点和 Hessian 参数化，但它们在优化中都存在过度参数化的问题。Bartoli 和 Sturm [4] 提出了 Pl"ucker 和正交表示方法，用于在线的三角测量和捆绑调整。在 [27] 中提出的最小参数化方法使用方位角和俯仰角来表示平面地标的法向量。对于基于特征的 SLAM 系统，通过因子图逐步重建的地标通常通过因子图进行优化。因此，对于地标的优化，需要采用合适的参数化方法。

稀疏点云是指使用一部分特殊几何信息来描述 3D 环境，一般来说这些稀疏点是梯度变化比较明显的部分。我们先讲解如何利用特征法来进行重建，然后再讨论直接法。

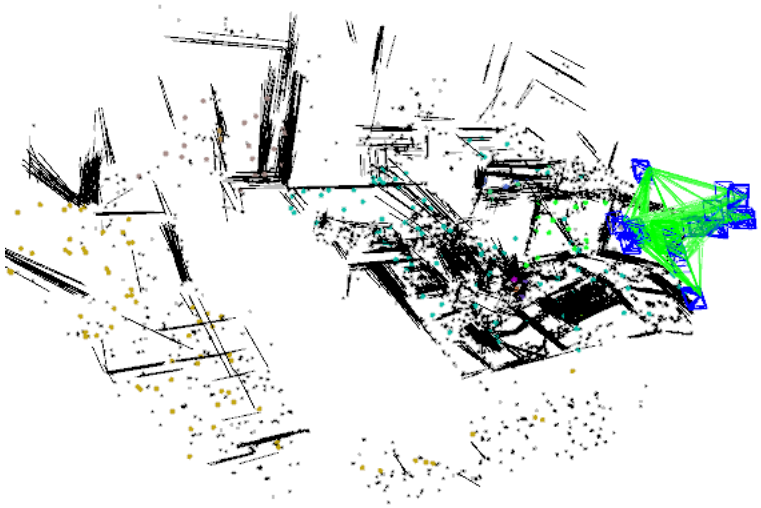

图 2: 点线面构成的稀疏点云。

稀疏点重建 特征法和直接法都可以获得稀疏点特征。特征法是从图片中提取特征点、线和面特征，并利用特征信息来计算相机位姿和重建环境信息。前面章节介绍了基本的特征提取与匹配知识，读者可以根据自己的需求选择相应的特征，因此这里我们直接使用这些特征来完成重建。

三角化 如图3所示，在两幅图像上分别提取同名特征点 x_1 和 x_2，而他们都是对空间路标点 X 的观测信息。我们的目标就是利用 2D 特征点来恢复空间 3D 路标点，众多的路标点则构成了稀疏点云。现在我们已经获得了同

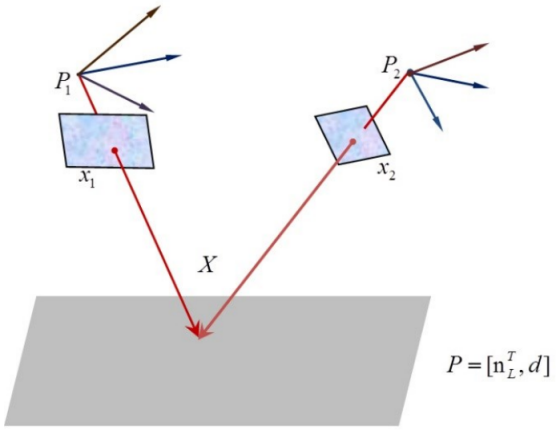

图 3: 特征点的重建

名点 $x_1 = (u_1, v_1)$ 和 $x_2 = (u_2, v_2)$，接下来就是通过简单线性三角化方法来重建稀疏点 X。

$$\begin{aligned} x_1 &= P_1 X_w \\ x_2 &= P_2 X_w \end{aligned} \tag{1.4}$$

这里 P_1 和 P_2 是两个相机对应的矩阵。我们构成一个矩阵 A，并且 $AX_w = 0$。

$$AX_w = \begin{bmatrix} x_1 \times P_1 X_w \\ x_2 \times P_2 X_w \end{bmatrix} \tag{1.5}$$

通过对矩阵 A 进行 SVD 分解，可以获得 3D 点的坐标。

```
cv::Mat A(4,4,CV_32F);
A.row(0) = kp1.pt.x*P1.row(2)-P1.row(0);
```

```
    A.row(1) = kp1.pt.y*P1.row(2)-P1.row(1);
    A.row(2) = kp2.pt.x*P2.row(2)-P2.row(0);
    A.row(3) = kp2.pt.y*P2.row(2)-P2.row(1);
    cv::Mat u,w,vt;
    cv::SVD::compute(A,w,u,vt,cv::SVD::MODIFY_A| cv::
        SVD::FULL_UV);
    x3D = vt.row(3).t();
    x3D = x3D.rowRange(0,3)/x3D.at<float>(3);
```

视差角 前文已经从理论上说明任意同名点都可以通过三角化操作实现 3D 点的重建。事实上，由于特征点和位姿信息都存在误差，当同名点之间视差角太小的时候，误差更容易被放大，因此，我们在重建的地图点的时候，尽量过滤掉这种情况。

首先，我们获得两个 2D 点在归一化平面上的坐标 $xn1$ 和 $xn2$, 这两个点可以看成是相机到这个点的向量，通过相机到世界坐标系的旋转矩阵 R_{wc}，可以获得世界坐标系下的两条光线 $ray1$ 和 $ray2$，并计算光线之间的余弦值。3D 点在世界坐标系下的光线

```
    // 世界坐标系下 观察光线
    cv::Mat ray1 = Rwc1*xn1;
    cv::Mat ray2 = Rwc2*xn2;
    // 计算在世界坐标系下，两个坐标向量间的余弦值
    const float cosParallaxRays = ray1.dot(ray2)
    /(cv::norm(ray1)*cv::norm(ray2));

    float cosParallaxStereo = cosParallaxRays+1;
    //ORB-SLAM使用的判断条件
    if(cosParallaxRays<cosParallaxStereo &&
        cosParallaxRays>0 && cosParallaxRays<0.9998)
```

线特征的重建。 对于直线的 3D 重建，我们可以使用类似于点的方式。如图5所示，3D 直线 L 的两个端点 X_s 和 X_e。利用点的方式，公式1.6，我们可以把两个端点投影到图像上，可以获得 $x_s = P_1 X_s$，然后点 x_s 在 l_1 和 l_2

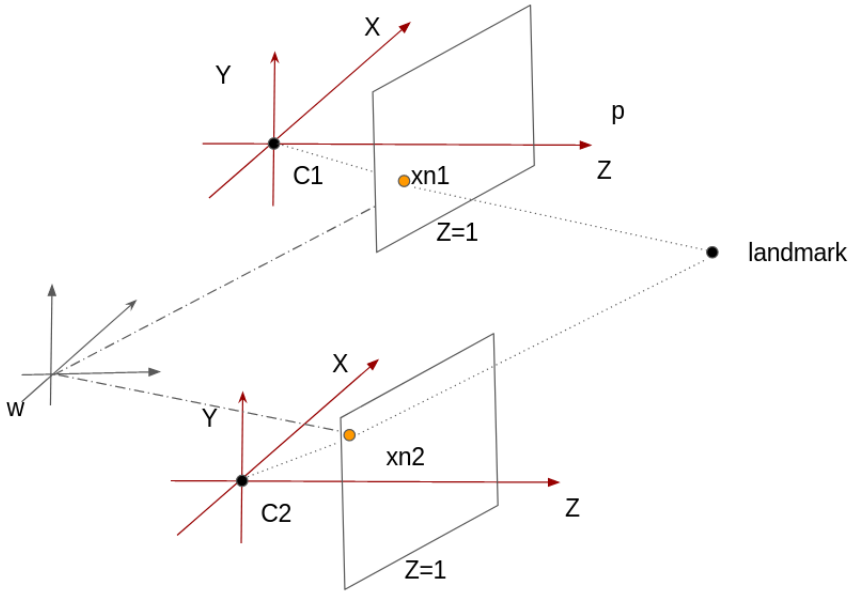

图 4: 在不同坐标系下的路标点三角化。

上，因此我们可以建立

$$\begin{aligned} l_1 x_1 &= l_1 P_1 X_s = 0 \\ l_2 x_2 &= l_2 P_2 X_s = 0 \end{aligned} \tag{1.6}$$

其中 l_1 和 l_2 是 2D 直线的方程，他们可以使用 2D 直线的端点计算出来。可见公式1.6是不足以求解出来 X_s 的，我们再把 l_1(或 l_2) 的端点加入到方程中，就像是特征点在公式1.6中的表现一样，$l_{1n} = P_1 X_n$, 这里 n 表示端点 s 或 e。

```
    // 起始点s
    cv::Mat A(4,4,CV_32F);
    A.row(0) = klF1.t()*M1;
    A.row(1) = klF2.t()*M2;
    A.row(2) = StartC1.at<float>(0)*Tcw1.row(2)-Tcw1.
        row(0);

    A.row(3) = StartC1.at<float>(1)*Tcw1.row(2)-Tcw1.
        row(1);
```

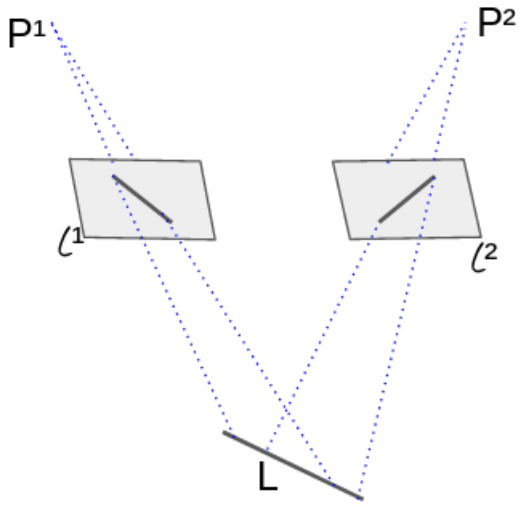

图 5: 特征线的重建

```
    cv::Mat w1, u1, vt1;
    cv::SVD::compute(A, w1, u1, vt1, cv::SVD::
        MODIFY_A| cv::SVD::FULL_UV);

    s3D = vt1.row(3).t();
```

视差角 与点重建过程中的处理方式相同，我们计算直线的中点，并考虑中点与中点之间的视差角。通过这种方式，在视差角太小情况下，我们可以避免 3D 直线的重建。

1.3.2 稠密地图重建 Dense Scene Representation.

与稀疏模型相比，密集重建通常提供更多信息以支持场景理解任务和新视图渲染。在密集重建中，针对不同传感器使用了不同类型的方法。对于单目密集重建，获取多个视图的密集深度图的核心技术是基于极线算法或基于补丁的匹配方法的逐像素匹配 [34]。类似于可以在捆绑调整模型中优化的那些稀疏地标，通过使用高斯分布模型，估计的密集点云也可以基于深度滤波器进行滤波 [15, 41]。

对于 RGB-D 传感器，可以直接获取深度图。与那些参数化地标不同，

下面介绍的方法，包括占据栅格 [11] 和有符号距离函数 (SDF) [9]，构建了非参数环境模型。一个由体素组成的 3D 体积表示环境，每个单元格中记录的值是占据的概率 [11]，当有新的观测时将更新该值。SDF [9] 于 1996 年提出，将表面界面表示为零。从 SDF 表示中，两个主要算法可以渲染表面区域，由 [21] 总结。第一个算法使用 Marching-Cube 算法 [26]。另一种策略是使用光线投射方法 [32]，以避免访问在所需视野之外的函数区域。上面介绍了稀疏点云中点线重建，他们都只考虑到一张图片中一小部分信息。而稠密重建的目标是获得每一个像素的深度信息，然后利用像素坐标和深度表示出来。DTAM 是利用 GPU 的一款单目直接法稠密重建系统，作者 Newcombe Richard 专注与稠密重建，并在 2014 年发表了 KinectFusion, 这两篇文章都是 SLAM 历史上的里程碑。这里我们先简单的分析 DTAM 的

图 6: 从左到右是深度恢复粗糙到精致的过程。图片来源 DTAM [**?**]

稠密重建思路，详细的优化过程推荐阅读论文原文。在直接法中，我们依靠光度误差来实现参考帧 I_r 和当前帧 I_m 的位姿估计，也就是文中的亮度一致性假设（brightness constancy assumption）。

$$\rho_r(I_m, u, d) = I_r(u) - I_m(\pi(KT_{mr}\pi^{-1}(u, d))) \tag{1.7}$$

公式中 $\pi()$ 是将 3D 投影成 2D。这样就建立起了 d 和 T_{mr} 的光度误差方程。

$$C_r(u, d) = \frac{1}{|I(r)|}\sum_m ||\rho_r(I_m, u, d)|| \tag{1.8}$$

将所有的关键帧的光度误差联合起来构建造优化方程，就可与优化出合适的地图和位姿信息。

根据地图的表现形式来分，稠密重建可以分为 mesh、SDF 和 surfel。下面我们一一介绍。

TSDF TSDF（truncated signed distance function）是一种密集重建中计算隐势面常用的表示方式，其中文全称为“基于截断的带符号距离函数”。

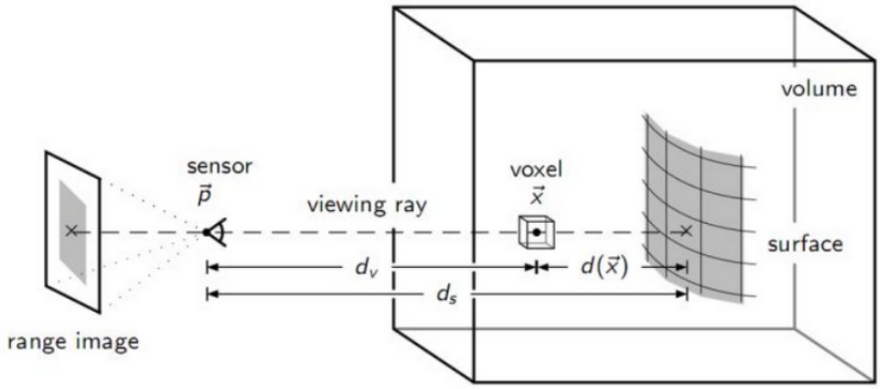

图 7: TSDF 算法示意图

TSDF 是 SDF 的一种改进方法，通过在 SDF 中增加表面到体素的截断距离来实现计算量的下降。

TSDF 算法把即将重建的空间看成是一个大的立方体区域 (volume)，并将大的立方体划分成许多小的立方体，我们称之为体素（voxel, 可以看出这个词就是 volume pixel 的意思)。

在 SLAM 或 SfM 问题中，我们通过不同的形式来获得空间信息，如空间三维点 P_w，然后将该点初始成一个体素，很多体素堆放在一起就是我们的重建目标。

1. 如何将当前帧融入到模型中 在讨论这个问题之前，我们先介绍 TSDF 算法中需要用到哪些信息来定义一个 voxel.

- 该 voxel 到最近表面的带符号距离，记作 tsdf(x)
- 体素涉及到更新问题，以此需要记录每次更新的权重，记作 w 。

我们抛开计算 3D 点的不同情况，从比较简单的 RGB-D 数据出发，深度信息由第 i 帧深度数据提供，并获得了相机位姿矩阵 T_{wc}，然后通过 $T_{wc}^T P_w$ 我们可以物体表面信息 P 在世界坐标系下的坐标转化到相机坐标系下 P_c。然后利用相机内参矩阵，找到该点对应在当前帧中的位置信息，一遍求取当前帧新观测信息到已知表面的距离信息。

具体来说，深度图可以知道当前帧给出的深度值 d，对因表面上的点距

离当前帧相机原点 O 的距离通过下式计算。

$$d_s = distance(O - P_c) \tag{1.9}$$

由相机内参矩阵，反投影 P_c 点求深度图像中的对应像素点 v，我们就可以获得改像素点对应的深度值 d_v。因此我们就获得了当前帧观测点的 sdf 距离。

$$sdf(p) = d_v - d_s \tag{1.10}$$

与 sdf 不同，TSDF 通过对距离设置一个截断阈值，构造了新的截断函数，

$$tsdf(p) = \begin{cases} sdf(d)/|u| & d < u \\ s \end{cases} \tag{1.11}$$

在计算 sdf 距离之后，我们还需要更新权重 $w(p)$ 的计算公式：

$$w(p) = cos(\theta)/d_s$$

其中 θ 是投影光线与表面法向量的夹角。

TSDF 的全局更新地图中的体素 x 的信息记为 $TSDF(x)$ 和 $W(x)$，这个点对应在当前帧中的信息是 $tsdf(x)$ 和 $w(x)$，新的帧进入之后，我们对观测到的信息进行更新：

$$\begin{aligned} TSDF(x) = & \quad \frac{W(x)TSDF(x)+w(x)tsdfx}{W(x)+w(x)} \\ W(x) = & \quad W(x) + w(x) \end{aligned} \tag{1.12}$$

2. 物体表面 物体表面在计算 tsdf 中有很重的作用，用来计算新加入信息的截断距离。一般使用 marching cubes 算法在去寻找 distance 加权和为 0 的等值表面。

1.4 误差收敛模型（Error Convergence Model）

如第 1.2节所介绍的，基于获得的姿态和地标，计算了测量值与估计值之间的差异。在本节中，我们通过两个方向介绍了通过不同的路线进行残差计算和初始细化：直接 vs. 间接和基于滤波 vs. 基于优化。

直接法 vs. 非直接法。 在方程 1.2中，相机姿态是我们想要用不同方法估计的目标。基于对环境的约束，可以将跟踪和映射系统分为直接和间接方法。具体而言，直接策略利用原始传感器值作为概率模型中的测量值，而间

接方法则利用具有已知匹配关系的部分像素。简而言之，**直接方法将基于光度学约束同时估计每个像素的相对姿态和深度**。然而，**间接方法首先执行检测对应关系，提供相机姿态和特征深度信息中更强的几何约束**。

直接方法。 这些直接方法不需要知道图像之间的强健对应关系。因此，用于跟踪的点可以是关键点和角点，而无需计算描述符。一些早期系统 [6, 20] 提出通过迭代重新加权方法检测异常值以实现鲁棒性。与此同时，当场景中出现光线和暗区变化时，直接方法还可以提供稀疏、半稀疏和密集模型。

*非直接方法。*在由 KLT 特征跟踪算法或描述符生成的对应关系的基础上，可以在极线几何中估计相对相机姿态和深度信息。在这些理论的帮助下，从 2000 年的 FastSLAM-2.0 [29] 和 2007 年的 MonoSLAM [10] 开始，提出了基于滤波的算法，以实现实时跟踪能力。第一个使用非线性最小二乘优化处理从关键帧获取的关键点的多线程系统是 2007 年提出的 PTAM [23]。

滤波与优化。 滤波和优化是 SLAM 系统用于细化初始估计的两个方向。基于滤波的 SLAM 通常更具计算效率，而基于优化的 SLAM 通常更准确但计算成本较高。

基于滤波的 SLAM 方法 [3, 19] 使用递归贝叶斯滤波技术，如卡尔曼滤波器 [22] 或扩展卡尔曼滤波器 [19]，来估计相机的姿态并映射环境。这些方法通过将概率状态估计向前传播，然后根据新的测量进行更新来工作。由于基于滤波的 SLAM 方法计算效率高，能够实时操作，因此它们在计算资源有限的应用中非常理想。

基于优化的 SLAM 方法 [30, 31, 36] 使用非线性最小二乘优化技术，如捆绑调整，来优化相机的姿态并重建环境。这些方法通过最小化预测图像测量与基于光度或几何关系的传感器实际测量之间的差异来工作。基于优化的 SLAM 可能比基于滤波的方法更准确，但在全局优化方面计算成本较高。

前面章节介绍了传感器位姿的估计，这里我们介绍 SLAM 另外一个目标-地图重建。对于未知环境的重建，是机器和环境进行交互的基础。在常见的 SLAM 系统中，地图主要分为稀疏点云、半稠密点云和稠密点云。

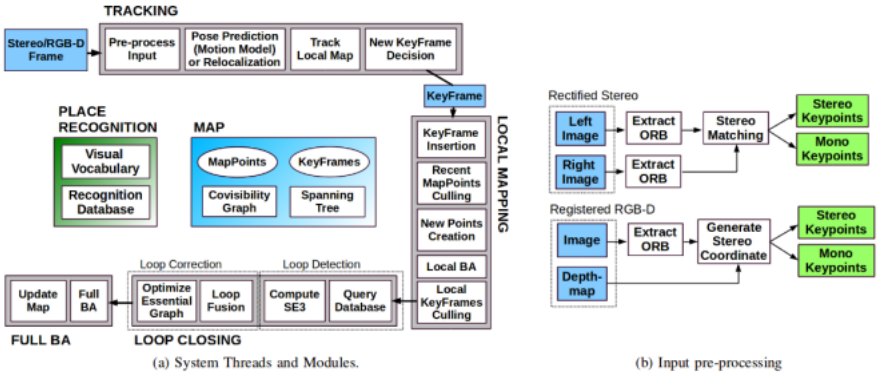

图 8: 特征法 SLAM（ORB-SLAM2）的经典框架。

2 SLAM 中的经典解决方案（Classic solutions in SLAM systems）

通过第二章的介绍，我们已经对 SLAM 系统的功能、输入和输出有了一定的了解。在我们进一步介绍更多基础理论之前，我们希望为那些处于早期学习阶段的读者提供更有针对性的内容。首先，我们按照不同的传感器对 SLAM 方案和基础知识进行了分类。这样，读者可以根据自己的研究或项目情况，直接深入学习与其相关的知识。随着学习的深入，读者将逐渐掌握更广泛的知识，从而完善个人的 SLAM 体系。

2.1 视觉 SLAM

2.1.1 基于特征和优化的 ORB-SLAM 系列

对于基于特征法的 SLAM 系统而言，2007 年发表的 **PTAM** 可被视为一个重要的里程碑。该论文提出了基于多线程的 tracking 和 mapping 策略，同时描绘了关键帧的发展历程。在 Mapping 部分，该系统仅处理关键帧的特征信息，并采用了局部光束法平差（Bundle Adjustment）以优化路标点和相机位姿。相较于 2007 年之前的 SLAM 或视觉里程计系统，PTAM 能够实时维护几千个点的地图。

同样基于点特征的系统，2017 年发表的 **ORB-SLAM2** [31] 是一个集

大成之作。由 Raúl Mur-Artal 博士等人提出，该系统是一个完备的基于特征点的纯视觉实时 SLAM 系统，适用于单目、双目和 RGB-D 相机。作者在 2015 年首次在 IEEE Transactions on Robotics（T-RO）期刊上发表了单目版本（ORB-SLAM [30]），这是一个基于纯特征点的单目实时 SLAM 框架。两篇文章所涉及的系统均非常完整，通过多线程实现了稀疏地图、回环和优化环节，为 SLAM 在工业界和学术界的发展做出了巨大贡献。

在 ORB-SLAM（V1/V2）工程的基础上，图 8中的各个模块吸引了研究人员的关注。其中，为系统增加其他特征信息是最为常见的改进之一。众所周知，点特征对于光照纹理较为敏感，尤其是在纹理较弱的区域，提取特征点的数量相对较少。为了应对低纹理场景中特征点能力的限制，一些系统引入了点线或点线面的结构，以适应单目 [35]、双目 [17] 和 RGB-D 相机 [25]。随着新特征的引入，关键帧策略、地图和优化环节都经过相应的调整。

2.1.2 基于光度误差和优化的 DSO 系列

LSD-SLAM（大规模直接单目 SLAM） [12] 是一种直接法 SLAM，于 ECCV2014 发表。与 ORB-SLAM 等使用特征法不同，直接法视觉里程计（VO）直接利用图像像素点的灰度信息进行建图与定位，克服了特征点提取方法的局限性，可以利用图像上的全部信息。在特征点稀缺的环境下，该方法仍能实现高定位精度与鲁棒性，并提供更丰富的环境几何信息，对机器人和增强现实应用具有重要意义。

该论文的第一作者 Jakob Engel 在直接法的发展方面做出了重要贡献。所提出的方法能够构建大尺度、全局一致的环境地图。除了能够通过直接图像配准获得高精度的姿态估计外，还能够实时重构三维环境地图，生成关键帧的姿态图和相应的半稠密深度图。这些结果是通过对大量像素点对之间的基线进行立体配准并进行滤波得到的。该算法提出了计算尺度漂移的公式，即使在图像序列的场景尺度发生较大变化时仍能适用。

整体流程 该算法主要包括三个主要组成部分：图像跟踪、深度图估计和地图优化，如图 9所示。

- **图像跟踪：** 连续跟踪从相机获取的新“图像帧”。即使用前一帧图像帧作为初始姿态，估算当前参考关键帧和新图像帧之间的刚体变换 $\xi \in se(3)$。

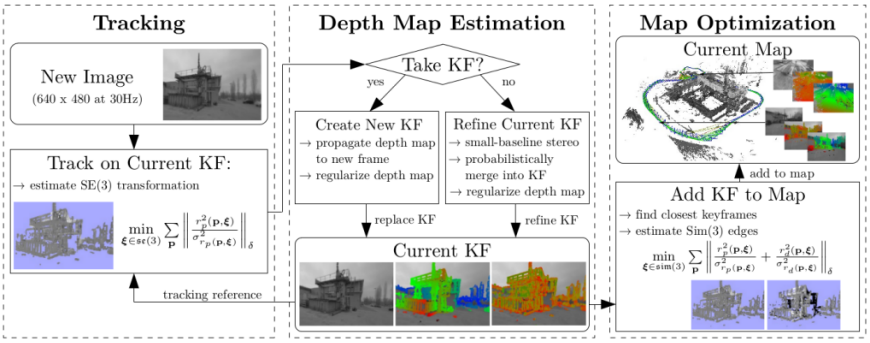

图 9: Pipeline of LSD-SLAM [12]

- **深度图估计：**利用被跟踪的“图像帧”，通过对当前关键帧进行深度更新或替换当前关键帧来进行深度图估计。深度更新基于像素小基线立体配准的滤波方式，并与深度地图的正则化相耦合。如果相机移动足够远，就初始化新的关键帧，并将现存相邻关键帧的图像点投影到新建立的关键帧上。

- **地图优化模块：**一旦关键帧被当前图像替代，其深度信息将不再进一步优化，而是通过地图优化模块插入到全局地图中。为了检测闭环和尺度漂移，采用尺度感知的直接图像配准方法来估计当前帧与现有邻近关键帧之间的相似性变换。

光度误差与深度误差 关键帧在 LSD-SLAM 系统中扮演着关键的角色。启动 LSD-SLAM 系统时，只需初始化首帧关键帧即可，而关键帧的深度信息最初被设定为一个方差很大的随机变量。在算法运行的最初几秒钟，一旦摄像头运动了足够的平移量，LSD-SLAM 算法就会“锁定”到某个特定的深度配置。通过几个关键帧的传递，系统将逐渐收敛到正确的深度配置。

在这个过程中，初始时的高方差深度信息允许系统对深度进行较大范围的探索。随着摄像头的运动，通过像素小基线立体配准的滤波方式，LSD-SLAM 能够逐渐减小深度估计的不确定性，使其趋于稳定。一旦算法锁定到特定深度配置，关键帧的传递和深度信息的更新将更加稳定，系统将在全局地图中插入新的关键帧，同时实时维护和优化地图的一致性。这一策略使得 LSD-SLAM 系统能够在运行初期适应不同深度配置的场景，并在系统稳

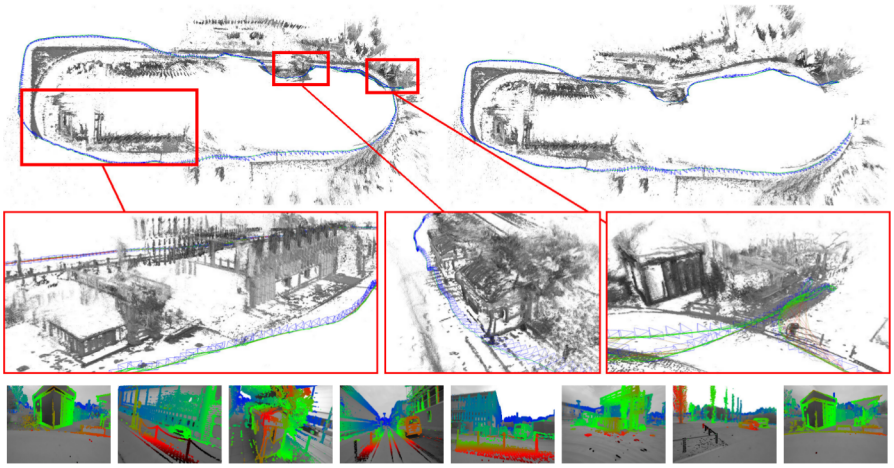

图 10: LSD-SLAM 中的闭环检测模块。左图是闭环检测的效果，右图是没有闭环检测。[12]

定后提供高精度和鲁棒性的深度估计。

$$E(\xi) = \sum_i (I_{ref}(p_i) - I(w(p_i, D_{ref}(p_i), \xi)))^2 \tag{2.1}$$

公式 2.1是当前帧 I 与参考帧 I_{ref} 之间的光度误差。其中 $w(p_i, D_{ref}(p_i), \xi)$ 是将这些像素重投影到参考帧 I_{ref}。

一旦新图像帧被选为关键帧，将上一帧关键帧的兴趣点投影到新创建的关键帧上，从而得到这一帧的兴趣点。随后，深度图被平均至逆深度为 1，便可以使用相似变换 $sim(3)$ 来计算关键帧之间的边，因为相似变换 $sim(3)$ 能够较好地考虑关键帧之间的尺度缩放差异。

除了包含光度测量残差 r_p 外，该论文还引入深度残差（depth residual）r_d 来惩罚关键帧之间的逆深度偏差，以直接估计帧间的相似变换。LSD-SLAM 论文中使用的误差函数 $E_{\xi_{ji}}$，

$$E(\xi_{ji}) = \sum || \frac{r_p^2(p, \xi_{ji})}{\sigma^2_{r_p(p,\xi_{ji})}} + \frac{r_d^2(p, \xi_{ji})}{\sigma^2_{r_d(p,\xi_{ji})}} || \tag{2.2}$$

其中 r_d 是深度惨差，$\sigma_{r_p}^{-2}$ 和 $\sigma_{r_d}^{-2}$ 分别是光度残差和深度惨差的逆方差。

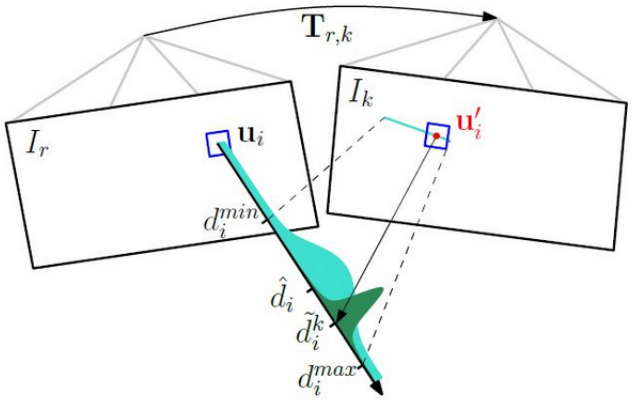

图 11: 深度的高斯-均匀分布。

2.1.3 SVO

SVO 的全称为 Fast Semi-direct Monocular Visual Odometry（半直接视觉里程计），是苏黎世大学机器人感知组 Christian Forster 等人于 2014 年在 ICRA 会议上发表的工作。在 2016 年扩展至多相机和 IMU 后，被写入期刊论文，并被称为 SVO 2.0。与基于特征法的 ORB-SLAM 和直接法的 LSD-SLAM 不同，SVO 提取了稀疏点，但使用直接法进行图像之间的对齐，因此被称为半直接法。在 LDSO 中，高翔博士同样使用了特征点和光度，其中特征点的主要作用是实现闭环检测，尽管半稠密的概念目前已经不如以前流行。SVO 将直接法和特征点法结合在一起，但特征点仅从关键帧中提取，以确保系统运行速度。SVO 算法可分为两部分：位姿估计和深度估计。下文详细介绍这两个模块。

位姿估计 通过对稀疏的特征块使用直接法进行配准，以获取相机位姿。通过获取的位姿来预测参考帧中特征块在当前帧中的位置。由于深度估计的不准确性可能导致位姿偏差，从而使得对特征块位置的预测不准确。由于预测的特征块位置和真实位置很接近，因此可以使用牛顿迭代法对这个特征块的预测位置进行优化。优化后的特征块位置的修正表明之前使用直接法进行的预测存在问题。利用这个经过优化的特征块预测位置，再次使用直接法进行相机位姿（pose）和特征点位置（structure）的优化。而特征点值从关键帧中提取，文章使用重投影误差进行位姿的细化。

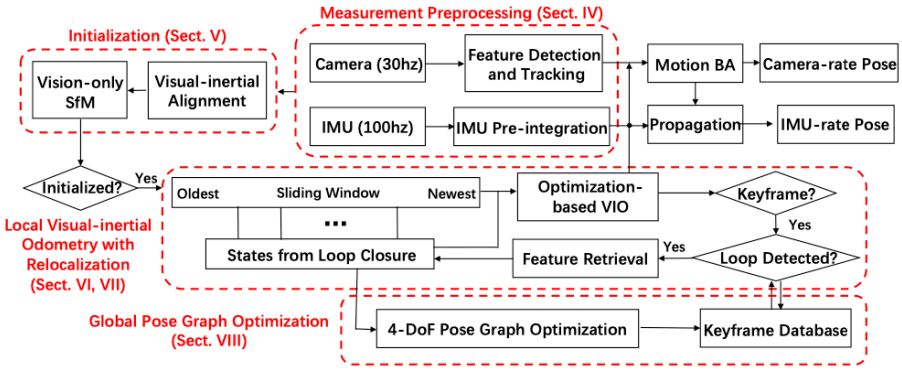

图 12: VINS-Mono 的系统框架。[36]

深度估计 除了追踪线程，SVO 还包含一个映射线程，其主要任务是完成对 2D 特征的深度估计。SVO 采用高斯-均匀分布对逆深度信息进行建模。当当前帧被选为关键帧时，系统会提取关键帧上的新特征点，这些特征点并没有对应的深度信息，因此需要进行深度计算。

对于关键帧上的新特征点，映射线程通过沿着极线搜索的方式获取其深度信息。在这个过程中，从极线的一个端点移动到另一个端点，对每个位置的图像块与参考帧进行比较，从而找到正确的匹配。如果某个种子点的深度分布已经收敛，那么将其加入地图中，以供追踪线程使用。

当处理普通帧时，映射线程利用普通帧的信息来更新所有种子点的概率分布。通过沿着极线从一个端点移动到另一个端点，将每个位置的图像块与参考帧进行比较，以找到正确的匹配。如果某个种子点的深度分布已经足够收敛，那么将其添加到地图中，以供后续追踪线程使用。这样，映射线程和追踪线程协同工作，实现了对 2D 特征深度的有效估计和地图的动态更新。

2.2 视觉惯导、激光雷达及其传感器融合 SLAM

2.2.1 基于光流和优化的 VINS 系列

VINS 系列主要指的是港科沈绍杰教授团队提出的 VIO 相关论文和开源系统，其中以 VINS-Mono 为代表。这篇文章的引用量在短时间内迅速上升至 1000（根据 Google Scholar 截至 2021 年 6 月），成为 VIO 领域不可忽视的重要论文。让我们一起回顾一下 VINS-Mono 提出的经典框架。

系统概述 如图12所示，系统首先分别以 30Hz 和 100Hz 的帧率输入单目图片和 IMU 信息。使用 KLT 光流算法跟踪每帧图像提取的 Harris 角点，而 IMU 部分则采用预积分方法进行处理。系统运行后的第一步是初始化程序。不论是 VINS 还是后续的 ORB-SLAM3，系统都会先进行纯视觉的初始化，成功后再进行视觉和 IMU 的共同优化。

在追踪过程中，该系统通过使用优化模型构建优化方程，其中包含了特征点约束、IMU 预积分约束和闭环检测约束，从而求解滑窗内所有帧的位置、速度、旋转和 IMU 的 bias。值得注意的是，在闭环检测中使用的图像帧是从滑窗中选取的关键帧。一旦通过 DBoW 方法检测到闭环，系统会对涉及闭环的关键帧进行全局优化。

2.2.2 激光雷达 SLAM 及其传感器融合方案

SLAM（Simultaneous Localization and Mapping）关注传感器在环境中的定位与建图问题。在其中，定位部分研究的是传感器的 6D 位姿估计问题，而建图则旨在恢复环境的 3D 信息。SLAM 系统提供的位姿和环境信息可应用于增强/虚拟现实以及无人驾驶系统，为机器与环境的交互提供服务。

根据不同的传感器类型，SLAM 系统在位姿估计和重建方面采用各种不同的方法。对于相机（包括 RGB-D），位姿估计方法可以分为 $2D-2D$、$2D-3D$ 和 $3D-3D$ 几种方式；对于激光雷达，有 $scan-scan$ 和 $scan-map$ 的位姿估计策略；而对于 IMU 和 GPS，则使用预积分和差分法等方法进行位姿估计。在重建方面，有稀疏、半稠密和稠密几种不同的方式。

2.3 SLAM 研究方向的变迁

SLAM 领域在过去的二三十年里经历了高速的发展，逐渐变得日益成熟。每一篇划时代的论文都获得了巨大的引用量，通过阅读这些经典作品，我们可以深切感受整个 SLAM 研究方向的演进和变迁。这些论文记录了 SLAM 领域的重要里程碑，反映了研究者们在传感器融合、位姿估计、地图构建等方面所取得的突破性成果。这些经典作品为 SLAM 技术的不断创新和进步奠定了坚实的基础。

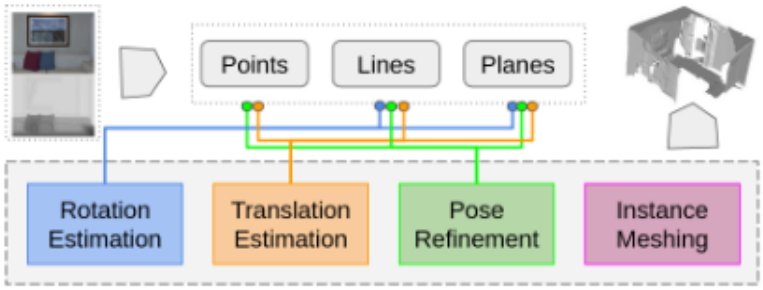

图 13: 多特征 SLAM 系统框架。

2.3.1 多几何特征与结构约束

ORB-SLAM 系列利用的特征信息主要是点特征，然而在后续的工作中，研究者们纷纷尝试增加新的特征类型。对于单目/双目传感器，早期的工作如 PL-SLAM 引入了直线信息；而对于 RGB-D 传感器，则有工作尝试增加平面信息。在这些早期的研究中，引入直线和平面信息主要是为了应对在纹理较弱区域难以提取点特征的问题。然而，由于没有充分利用这些信息的特性，即便是系统增加了特征的种类，SLAM 系统的性能提升并不明显。

不同于传统的点-线-面系统，RGB-D SLAM [25] 进一步探索直线与直线，平面与平面之间的平行/垂直关系。由于，这种几何关系满足 Manhattan World 约束，这种约束带来了零漂移旋转估计和 3D 平移估计的策略。同时，在提取平面特征之后，我们对环境重建的能力也增强了，可以在 CPU 基础上对环境进行实时 mesh 重建。

2.3.2 多传感器

由于运动模糊和位姿剧烈变化，快速运动情况对基于特征法的 SLAM 系统提出了较大的挑战。为了克服这些挑战，研究人员通过增加传感器来引入更多的信息源。其中，惯性测量单元（IMU）是一款成本较低的设备，可以提供高帧率的位姿估计结果。VINS-Mono [36] 是一种经典的相机和 IMU 紧密耦合的系统。由于 IMU 能够以高频率记录角速率和加速度信号，通过对这些高频信号进行预积分操作，系统能够稳定地获取两帧图像之间的相对位姿。然后，通过相机带来的视觉特征的约束，实现了位姿信息的优化。此外，也有研究结合 IMU 与 ORB-SLAM 或 DSO 的工作，它们在处理快速运动情况下取得了显著的效果。采用与增加线特征的策略相类似，PL-VIO

在 VINS-Mono 的基础上引入了线特征，以应对特征提取在快速运动情况下的挑战。

2.3.3 几何与学习方法在 pose 上的合作

深度学习的引入拓展了传统 SLAM 系统的传感器边界，主要聚焦在提升重建和位姿估计方面。早期的结合工作包括 DVSO 和 CNN-SLAM，这两篇工作均于 2018 年发表，都使用深度学习来估计深度信息以实现场景的重建。

除了利用深度学习估计的深度信息用于单目 SLAM 的密集重建外，还有一些新的工作直接应用深度学习来提高位姿估计的效果。这一趋势表明深度学习在 SLAM 领域的应用逐渐多样化，不仅仅局限于重建方面，还在位姿估计等关键任务上发挥着重要作用。值得注意的是，深度学习提供的特

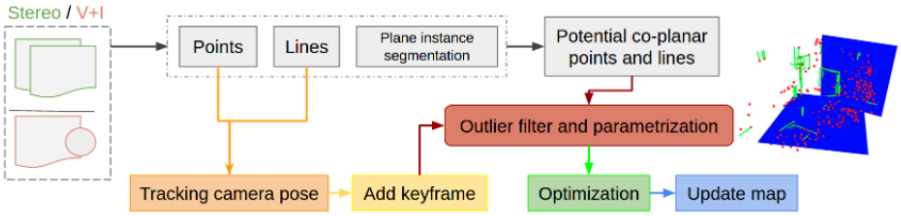

图 14: 基于学习平面的共面参数化方法（CP-Param [24]）。

征并不像传统的几何特征那样稳定。因此，在为 SLAM 引入学习信息的同时，也需要考虑引入一个去伪存真的策略，以应对深度学习特征的不稳定性。这种策略的目的是确保通过深度学习获得的信息能够在 SLAM 系统中得到可靠的应用，从而提高系统的稳定性和鲁棒性。

2.3.4 学习方法在模型重建和渲染上的表现

深度学习方法在模型重建和渲染方面取得了显著的进展，其中两个重要的领域分别是 NeRF（Neural Radiance Fields）和高斯光滑（Gaussian Splatting）。首先，NeRF 是一种基于神经辐射场的模型，通过训练深度神经网络学习场景中每个点的辐射率。该模型不仅能够高度准确地还原场景的几何形状，而且能够捕捉光照和颜色等细致的光学效果。NeRF 的优势在于其对场景的高保真还原，特别在重建真实世界中的复杂光学现象时表现出色。另一方面，高斯光滑是一种基于高斯函数的点云渲染方法。通过将场

图 15: 通过单个单目摄像机，我们以每秒 3 帧的速度实时重建高保真的 3D 场景。对于每个传入的 RGB 帧，我们逐步形成 3D 高斯并与相机姿态一同进行优化。我们展示了光栅化的高斯（左侧）和用于突显几何形状的高斯阴影（右侧）。请注意捕捉到的细节和复杂的材质特性（例如透明度）。诸如电线等细小结构由众多小而纵向的高斯准确表示，并且透明物体通过沿边缘放置高斯来有效表示。我们的系统显著提高了单目 SLAM 系统能够捕捉的保真度。 [28]

景中的点云投影到图像平面，并利用高斯函数对每个投影点进行权重分布，高斯光滑能够在图像中生成平滑的效果。这种方法在处理点云数据时能够有效地提高渲染效果，使得渲染结果更加真实、连续。这两个成果展示了深度学习在模型重建和渲染方面的强大能力，为计算机图形学和计算机视觉领域带来了新的可能性，尤其是在真实感图像生成和三维场景建模方面。

在光线场重建和视图合成领域，随着神经辐射场（NeRFs）的崛起，取得了显著的进展。训练 NeRF 的一个重要初始化步骤是为每个输入图像准备相应的相机姿态，通常通过运行 Structure-from-Motion（SfM）库 COLMAP 来实现。然而，这种预处理不仅耗时，而且由于其对特征提取误差的敏感性以及难以处理无纹理或重复区域而可能失败。一部分研究希望通过在 NeRF 框架内直接集成姿态估计来减少对 SfM 的依赖。在 NeRF 及其隐式表示的背景下，由于优化过程通常涉及额外的约束，这一挑战变得更加严峻。例如，要求初始姿态接近其地面实际位置，而 NeRFmm 在面向前的场景中受到较大限制。最近提出的 Nope-NeRF 训练时间较长（30 小时），在相机姿态发生较大变化（例如，360 度）时效果不佳，最新引入的 3D Gaussian Splatting 展现出来很多优于 NeRFs 的特性。与 NeRF 相比，3DGS 执行可微光栅化。类似于常规图形光栅化，通过迭代要光栅化的原语而不是沿着射线前进，3DGS 利用了 3D 场景的自然稀疏性，并实现了一种富有表现力的表示，能够捕捉高保真的 3D 场景，同时提供显著更快的渲染速度。一些研

490 究已经应用了 3D 高斯和可微渲染来进行静态场景捕捉，特别是近期的一些
491 工作利用了 3DGS，并在视觉任务（如动态场景捕捉和 3D 生成）中展示了
492 卓越的结果。

参考文献

[1] Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[2] Cuneyt Akinlar and Cihan Topal. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognition Letters*, 32(13):1633–1642, 2011.

[3] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the ekf-slam algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568. IEEE, 2006.

[4] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer vision and image understanding*, 100(3):416–441, 2005.

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[6] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

[7] J. Brian Burns, Allen R. Hanson, and Edward M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):425–455, 1986.

[8] Javier Civera, Andrew J Davison, and JM Martinez Montiel. Inverse depth parametrization for monocular slam. *IEEE transactions on robotics*, 24(5):932–945, 2008.

[9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[10] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[11] Alberto Elfes and Larry Matthies. Sensor integration for robot navigation: Combining sonar and stereo range data in a grid-based representataion. In *26th IEEE conference on decision and control*, volume 26, pages 1802–1807. IEEE, 1987.

[12] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[13] Chen Feng, Yuichi Taguchi, and Vineet R Kamat. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6218–6225. IEEE, 2014.

[14] Chen Feng, Yuichi Taguchi, and Vineet R. Kamat. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6218–6225, 2014.

[15] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[16] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, volume 6, pages 281–305. Interlaken, 1987.

[17] Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuniga-Noël, Davide Scaramuzza, and Javier Gonzalez-Jimenez. Pl-slam: A stereo slam system through the combination of points and line segments. *IEEE Transactions on Robotics*, 35(3):734–746, 2019.

[18] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[19] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. Observability-based rules for designing consistent ekf slam estimators. *The International Journal of Robotics Research*, 29(5):502–528, 2010.

[20] Michal Irani, Benny Rousso, and Shmuel Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12:5–16, 1994.

[21] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[22] Farrokh Janabi-Sharifi and Mohammed Marey. A kalman-filter-based method for pose estimation in visual servoing. *IEEE transactions on Robotics*, 26(5):939–947, 2010.

[23] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[24] Xin Li, Yanyan Li, Evin Pınar Örnek, Jinlong Lin, and Federico Tombari. Co-planar parametrization for stereo-slam and visual-inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):6972–6979, 2020.

[25] Yanyan Li, Raza Yunus, Nikolas Brasch, Nassir Navab, and Federico Tombari. Rgb-d slam with structural regularities. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11581–11587. IEEE, 2021.

[26] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[27] Lingni Ma, Christian Kerl, Jörg Stückler, and Daniel Cremers. Cpa-slam: Consistent plane-model alignment for direct rgb-d slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1285–1291. IEEE, 2016.

[28] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. *arXiv preprint arXiv:2312.06741*, 2023.

[29] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI*, volume 3, pages 1151–1156, 2003.

[30] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[31] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[32] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[33] Bastian Oehler, Joerg Stueckler, Jochen Welle, Dirk Schulz, and Sven Behnke. Efficient multi-resolution plane segmentation of 3d point clouds. In *Intelligent Robotics and Applications: 4th International Conference, ICIRA 2011, Aachen, Germany, December 6-8, 2011, Proceedings, Part II 4*, pages 145–156. Springer, 2011.

[34] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. Remode: Probabilistic, monocular dense reconstruction in real time. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 2609–2616. IEEE, 2014.

[35] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. Pl-slam: Real-time monocular visual slam with points and lines. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 4503–4508. IEEE, 2017.

[36] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[37] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2008.

[38] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[39] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[40] Carlo Tomasi and Takeo Kanade. Detection and tracking of point. *Int J Comput Vis*, 9:137–154, 1991.

[41] George Vogiatzis and Carlos Hernández. Video-based, real-time multi-view stereo. *Image and Vision Computing*, 29(7):434–441, 2011.

[42] Rafael Grompone Von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):722–732, 2008.

[43] Lilian Zhang and Reinhard Koch. An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7):794–805, 2013.