周翊民老师课题组洞察月报

——专注基于视觉的SLAM 研究

2018 年 12 月　　第 4 期

专题：基于视觉的 SLAM 的研究现状调研

编辑：陈雅兰

指导：周翊民

校对：吴庆甜

目录

一 绪论

1.1 研究背景与意义

近年来，随着科学技术的发展，机器人技术发展迅速并且日趋智能化，除了汽车、机械、化工等应用为主的工业机器人，还有用于科学研究、医疗救援、家庭服务、娱乐教育等方面的服务性机器人。这些机器人需要具备更高级的自动化程度和智能，其广泛应用将不仅丰富了生活，更帮助人类探索环境恶劣或危险的位置环境，推动科技、工业生产的进步。

对于一个想要实现自主运动的机器人，有三个问题摆在它面前：

- ✧ 我在什么地方？——定位

  Where I am?

- ✧ 周围环境是什么样的？（）——构建地图

  What's the environment?

- ✧ 如何到达目标点？ ——路径规划和避障

  How to arrive to destination?

其中，前两个问题是移动机器人自主导航中必须解决的两大关键问题，定位和地图构建，同时也为后续的路径规划和避障问题，提供了可靠信息产生巨大影响。

SLAM 是 Simultaneous Localization and Mapping 的缩写，即同时定位与地图构建，由 Leonard 和 Durrant-Whyte 于 1991 年首次提出，它是指搭载特定传感器的主体，在缺乏环境先验信息的情况下，在运动过程中构建环境的模型，同时估计自身运动。如果使用的传感器主要为相机，则称为视觉 SLAM（visual SLAM）。

传统的 GPS 定位在室内以及其他被遮挡的区域，不能获取 GPS 信号，导致其定位精度不能满足需求，因此需要利用机器人搭载的其他传感器获取环境信息实现定位和地图构建功能。目前用于 SLAM 的传感器主要有激光雷达和相机。其中，激光雷达可以提供精确的环境信息，但是不能用于非常混乱的环境，且价格昂贵，功耗、体积较大，不适合用于无人机和小型移动机器人，而相机具有质量轻、体积小、能获得环境外观和深度信息等优点。采用视觉传感器作为解决 SLAM 问题的主要传感器逐渐成为一种趋势。

应用场景

关于机器人的定位和地图构建问题，最早是在 1986 年 Robotics and Automation Conference 上提出的，经过将近 30 年的发展，SLAM 技术不仅在理论和系统研究上获得巨大发展和成果，同时也已经用于实际生产、生活以及科学研究中。

1） 扫地机器人

扫地机器人是家庭服务机器人应用最为广泛的一种，依据其导航方式不同可分为两类，即随机碰撞型和智能扫描规划型，而智能扫描规划型中包含了北极星定位、无线载波室内定位、RPS 激光定位和图像测算导航四种导航类型。前两种导航技术均是基于信标的定位技术，即由安装在机器人上的接收器接收房间内位置已知的发射器信号，通过计算得到自身位置信息进行导航。

图 1-1　Dyson 360Eye 扫地机器人

RPS 激光定位和图像测算导航采用 SLAM 技术，其中 RPS 激光定位采用激光雷达进行定位和环境建图，而基于图像测算导航型扫地机器人仍比较少，iRobot 公司的 iRobot Roomba 980 利用机器人顶部斜向前的摄像头，通过搭载其公司专利 vSLAM 视觉定位系统和 iAdapt2.0 Naviagtion 算法进行路径规划。此外，英国 Dyson 公司也推出的 360Eye 利用顶部安装的 360°深度图像摄像头，采用视觉 SLAM 技术进行定位、建图和路径规划。

图 1-2　iRobot Roomba980

2） 自动导航车（AGV）

AGV 即 Automated Guided Vehicles 的缩写，是智能工厂的重要组成部分，不仅用于制

造车间和大型物流集散地，也可在危险或军事任务，依托 AGV 的自动驾驶功能结合其他探测或拆卸功能，用于侦察、危险品处理等。采用视觉 SLAM 技术的代表厂商是美国 SEEGRID 公司，其公司首先推出 VGV（Vision Guided Vehicles），并通过实验展示其产品不仅有利于工厂的柔性化生产，同时利用视觉传感器获得障碍物的三维环境信息，减少冲撞产品、装备等造成的损伤。

图 1-3　SEEGRID VGVs 叉车

AGV 还用于无人驾驶汽车中，其中 Alphabet 公司旗下的 Waymo 无人车最初是 Google 于 2009 年开启的一项自动驾驶汽车计划，后于 2016 年底从 Google 独立出来。该无人车搭载激光雷达、GPS、立体视以及惯性导航系统，在没有驾驶者的情况下，在公共道路的路测里程已达 800 万英里。百度的 Apollo 无人车于 2018 年在港珠澳大桥开跑，在无人驾驶模式下完成“8”字交叉跑的高难度动作。

图 1-4 Waymo 无人车

图 1-5 百度 Apollo 无人车

3） AR/VR

AR/VR 即 Augment Reality/Virtual Reality 的缩写，是一种将真实世界信息和虚拟世界信息进行综合和叠加的技术，目标是在屏幕上把虚拟世界套在现实世界并进行互动。AR 技术的应用领域广泛，例如在古迹复原和数字文化遗产保护中，文化古迹以增强现实的方式供参观者欣赏，用户不仅可通过 HMD 看到文字解说，还能看到残缺部分的虚拟重构。在旅游领域，通过 AR 技术构建历史景观或者旅游景点，使得人们可以足不出户就遍览天下美景。LEAPSY 与同程文旅合作的常州淹城春秋乐园与 2018 年 5 月 19 日“中国旅游日”首次向游客推出 AR 体验。在教育领域，可使得书本的文字内容更形象生动地展现出来，提高阅读的趣味性，也能增加课堂的互动性，调动学习的积极性和主动性。在游戏领域，以头戴式设备（HMD）为主的沉浸式游戏模式改变了传统的键鼠/手柄操作模式，其中 Oculus 公司推出了 FPS 游戏。

图 1-6 LEAPSY AR 眼镜观看春秋乐园诸子百家园景点

ARKit 是 Apple 推出的基于 iOS 系统开发增强现实 APP 的开发平台，不仅可以构建虚拟故事书供孩子从任意角度探索故事场景，也可以用于房屋设计或者类似 Pokemon Go 的游戏。类似 ARKit，谷歌推出了搭建增强现实应用程序的软件平台 ARCore，它可以利用云软件和硬件设备进行同步，将数字对象放到现实世界中。

a)　虚拟故事书

b)房屋设计　　c)Pokemon Go

图 1-7　基于 ARKit 开发的 AR 应用

4） 无人机导航

无人机在飞行过程中需要获取周围环境信息，尤其是障碍物的位置，为路径规划和避障提供可靠的依据。而视觉 SLAM 在无人机导航技术中应用较少，大疆无人机的精灵 4 实现了双目立体视觉和惯性测量元件构成的视觉里程计，无人机搭载了向前和向下看两套立体视觉系统，同时增加了 GPS 和地磁指南针等的冗余设计，扩展了飞行器的可活动范围和场景适应性。

图 1-8 DJI PHANTOM 4 Pro v2.0

二 视觉 SLAM 中的相机与图像

针孔相机模型及畸变

1）针孔相机模型

相机的成像过程可以用针孔模型进行几何建模。

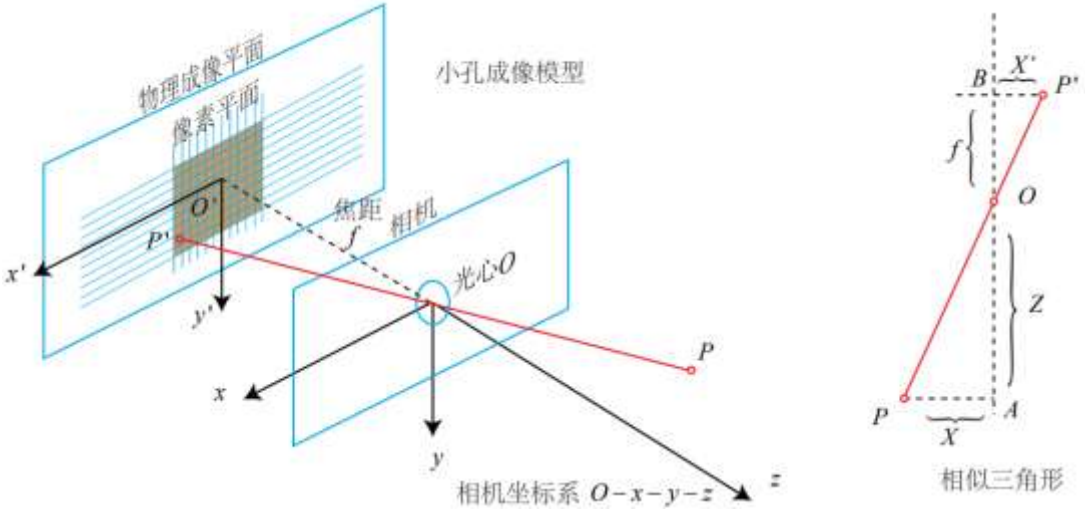


图 2-1 针孔相机模型

设 $O-x-y-z$ 为相机坐标系， z 轴指向相机前方， x 向右， y 向下， O 为相机光心，

即针孔模型中的针孔。现在又一个三维的空间点P，经过小孔O投影后，落在物理成像平面$O^{'}-x^{'}-y^{'}-z^{'}$上得到成像点$P^{'}$。设P的坐标为$[X,Y,Z]^T$，$P^{'}$的坐标为$[X^{'},Y^{'},Z^{'}]^T$，物理成像平面到小孔的距离为f（焦距），根据三角关系可得

$$\frac{Z}{f}=-\frac{X}{X^{'}}=-\frac{Y}{Y^{'}} \qquad (2\text{-}1)$$

其中负号表示成的像是倒立的。为简化模型，将成像平面对称到相机前方，和三维空间点放在相机坐标系同一侧，公式可简化为：

$$\frac{Z}{f}=\frac{X}{X^{'}}=\frac{Y}{Y^{'}} \qquad (2\text{-}2)$$

假设在物理成像平面上固定这一个像素平面$o-u-v$，$P^{'}$的像素坐标为$[u,v]^T$：

其中，像素坐标系的定义为，原点o在图像左上角，u轴向右与x轴平行，v轴向下与y轴平行。由图易知，像素坐标系和成像平面之间，相差了一个缩放和原点的平移。设像素坐标在u轴缩放了α倍，在v上缩放了β倍。同时原点平移了$[c_x,c_y]^T$，则可得$P^{'}$坐标和像素坐标的关系为：

$$\begin{cases}u=\alpha X^{'}+c_x\\ v=\beta Y^{'}+c_y\end{cases} \qquad (2\text{-}3)$$

带入式　并把αf合并成f_x，把βf合并成f_y，得到下式

$$\begin{cases}u=f_x\dfrac{X}{Z}+c_x\\ v=f_y\dfrac{Y}{Z}+c_y\end{cases} \qquad (2\text{-}4)$$

其中，f的单位为米，α，β的单位为像素/米，所以f_x和f_y的单位为像素，

$$\begin{pmatrix}u\\ v\\ 1\end{pmatrix}=\frac{1}{Z}\begin{pmatrix}f_x & 0 & c_x\\ 0 & f_y & c_y\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}X\\ Y\\ Z\end{pmatrix}=\frac{1}{Z}KP \qquad (2\text{-}5)$$

将Z挪到左边可得：

$$Z\begin{pmatrix}u\\ v\\ 1\end{pmatrix}=\begin{pmatrix}f_x & 0 & c_x\\ 0 & f_y & c_y\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}X\\ Y\\ Z\end{pmatrix}=KP \qquad (2\text{-}6)$$

式中的矩阵为相机的内部参数矩阵K，通常认为相机内参在出厂后是固定的，不会再使

用过程中变化。

与内参对应，相机参数中也存在外参，即描述相机的位置和姿态的旋转矩阵 R 和平移向量 t 。相比于不变的内参，外参随相机运动发生改变，同时也是 SLAM 中待估计的目标，代表相机的轨迹。

2）畸变

相机利用透镜获得良好的成像效果，同时透镜的加入对成像过程中光线的传播会产生影响。1）透镜形状对光线传播的影响；2）机械组装过程中，透镜和成像平面不可能完全平行。由透镜形状引起的畸变成为径向畸变。在实际拍摄中，相机的透镜会使得真实环境中的一条直线在成像平面变成曲线，越靠近图像边缘，这种畸变越明显。径向畸变可分为桶形畸变和枕形畸变。

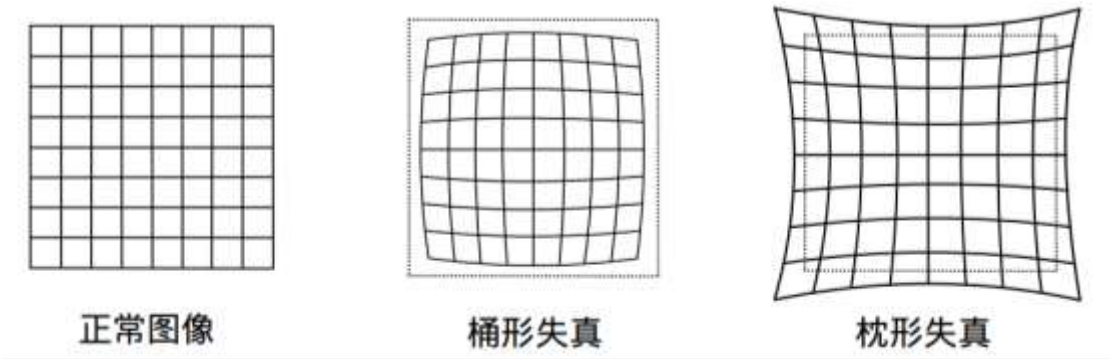


图 2-2 径向畸变的两种类型

对于径向畸变，随着与中心的距离增加而增加，可用一个多项式表达畸变前后的坐标变化：

$$\begin{aligned} x_{distorted} &= x\left(1+k_1r^2+k_2r^4+k_3r^6\right) \\ y_{distorted} &= y\left(1+k_1r^2+k_2r^4+k_3r^6\right) \end{aligned} \qquad (2\text{-}7)$$

在相机组装过程中犹豫透镜和成像平面不能严格平行将引入切向畸变。

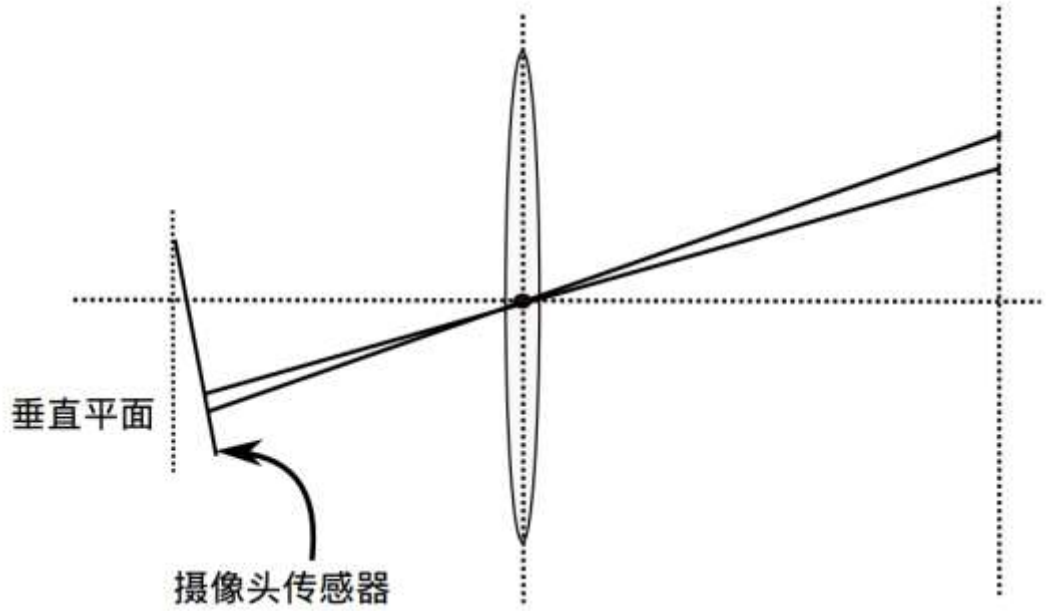


图 2-3 切向畸变来源示意图

对于切向畸变，可通过引入两个参数 p_1, p_2 进行纠正。

$$\begin{aligned} x_{distorted} &= x + 2p_1xy + p_2\left(r^2 + 2x^2\right) \\ y_{distorted} &= y + p_1\left(r^2 + 2y^2\right) + 2p_2xy \end{aligned} \quad (2\text{-}8)$$

通过式（2-7）和（2-8），对于相机坐标系中的一个空间点 $P(X,Y,Z)$，可通过畸变系数获得其在像素坐标系下的对应坐标。

相机标定原理及方法

通过介绍针孔相机模型及其畸变影响，可知要获得空间物体某个点的三维空间位置及其在像素平面中对应点之间的关系，就需要通过实验和计算确定相机的内参和畸变系数，而求解参数的过程即相机标定。相机的参数标定是非常关键的环节，标定结果的精度和算法稳定性直接影响相机工作产生结果的准确性。

虽然相机的标定方法有许多，例如传统相机标定利用尺寸已知的物体作为标定物，通过建立标定物上坐标已知的点和其图像点之间的对应，利用一定算法获得相机模型的内外参数。标定精度虽然比较高但是过程复杂，现在最常用的标定方法是张正友教授于 1998 年提出的单平面棋盘格的相机标定方法，介于传统标定法和自标定法之间，克服了传统标定法需要高精度标定物的缺点，只需要打印一张棋盘格即可。

MATLAB 和 ROS 均已提供了完整的基于张正友标定法的标定工具，因为我们只需要确保获得较为精确的相机参数模型，对于其标定原理可以不必详细了解，只需知道如何使用这些现成的标定工具。

在使用标定工具前，需打印一张棋盘格，将其贴在平整的平面上作为标定物。

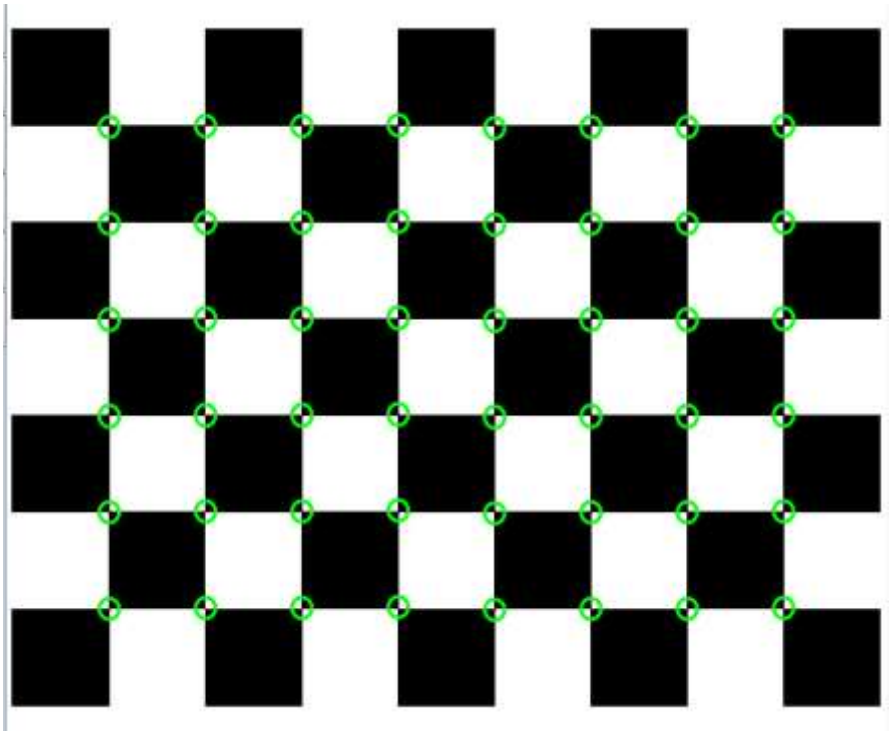


图 2-4 标定用棋盘格 6*8

下面以使用 MATLAB 的标定工具为例：

1. 首先打开 MATLAB 应用程序中的“Camera Calibrator”；
2. 在新窗口选择添加图片“Add Image”，再依据弹出的提示框填写棋盘格真实尺寸；
3. 工作空间出现已检测出的棋盘格，然后点击 Calibration 开始标定；
4. 使标定板尽量出现在相机视野的哥哥位置，通过不断平移、转动或倾斜标定板进行标定，平均误差小于 0.5 即认为标定结果准确，
5. 标定结束后，通过 Export Camera Parameters 导出标定参数，并且可在 MATLAB 工作空间看到相应的参数。

而在 ROS 环境下，只需要在终端用一行命令启动标定程序：

Rosrun camera_calibration cameracalibrator.py –size 11x8 –square 0.03 image:={image path} camera:={camera path}

其中，size 是标定板的棋盘格内部角点个数，square 后的数字是棋盘格的真实尺寸，单位

为米。

然后按照 MATLAB 中标定步骤的第四点移动标定板，待图形界面的右侧的 CALIBRATE 按钮变亮，点击按钮即可进行标定，标定过程将持续一两分钟，并且标定界面会变成灰色无法操作。标定完成后，窗口中的图像为标定后的结果，并且各个参数会出现在标定程序的终端窗口，若对标定结果满意则点击 COMMIT 按钮将结果保存到默认文件夹中，下次启动此相机时将自动调用标定文件。

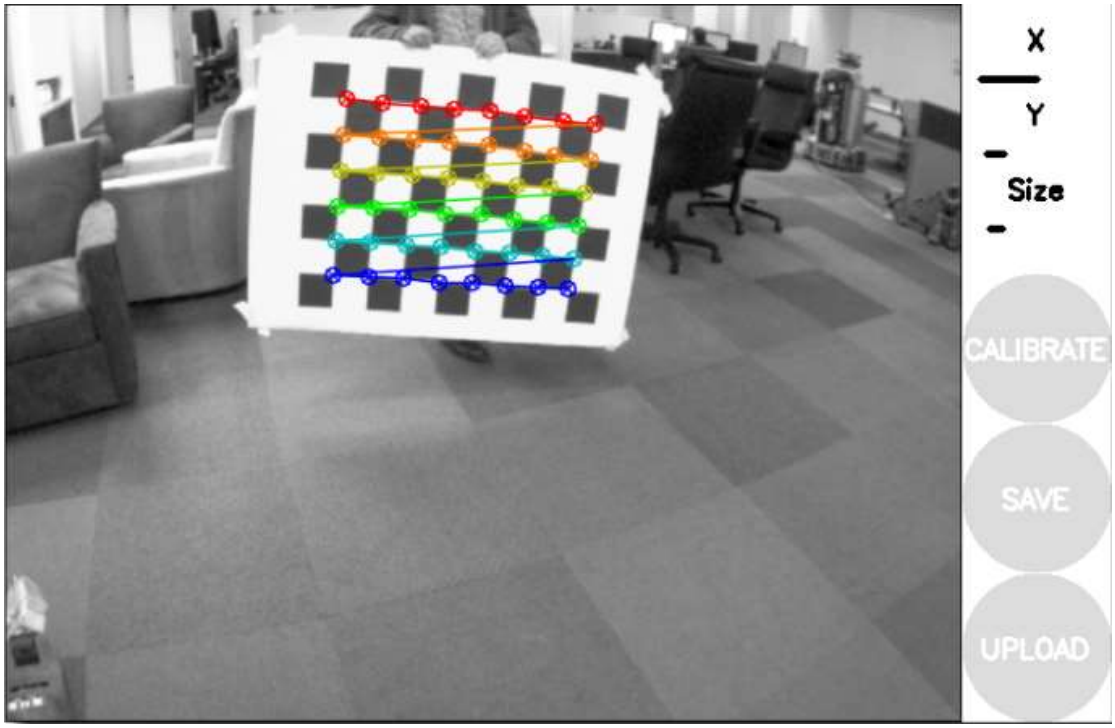


图 2-5 ROS 环境中的相机标定程序界面

单目相机

单目相机顾名思义就是只有一个摄像头的相机。单目相机结构特别简单，成本特别低，其镜头按焦距能分为标准镜头、广角镜头、鱼眼镜头等。照片本质是三维空间物体在相机平面的投影，以二维的形式反映三维的世界。在一幅照片中无法确定一个物体的真实尺度，成为尺度不确定性。

单目相机既有弊也有利，其缺点是失去尺度信息，在 SLAM 的运动估计和恢复中无法确定真实的尺度；其优点也是因为尺度不确定性，使其在 AR 中能让虚拟世界和真实世界无缝融合。

双目相机

双目相机一般由左右两个水平放置的相机组成，当然也可以做成上下两目的，不过主流双目都是做成左右形状的。左右双目相机中的两个摄像头都可以看成是针孔相机，其光心位于 x 轴，两个光心的距离称为双目相机的基线，是双目相机的重要参数，基线越长时，双目能测量的最大距离也会越大。

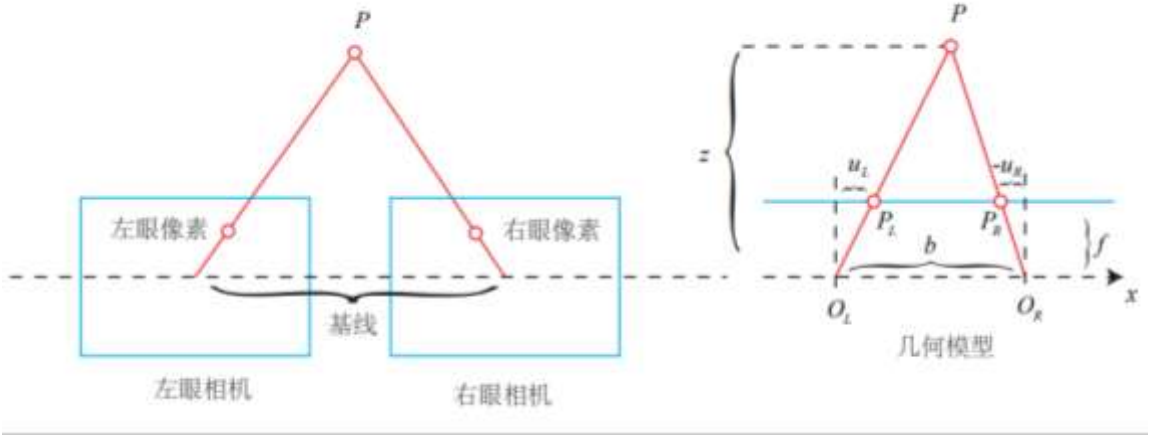


图 2-6　双目相机成像模型

其获得像素深度信息的原理：对一个三维空间点 P，会在左右相机中各成一像分别记为 P_L 和 P_R。由于基线的存在，两个成像位置是不同的，理想情况下，由于左右相机只在 x 轴上有位移，因此 P 在左右两眼相机的成像只在 x 轴上有差异，记其左侧图像的坐标 u_L，右侧坐标 u_R，根据 ΔPP_LP_R 和 ΔPO_LO_R 的相似关系，其中 O_L 和 O_R 是左右光圈中心：

$$\frac{z-f}{z}=\frac{b-u_L+u_R}{b} \qquad (2\text{-}9)$$

整理之后，可得：

$$z=\frac{fb}{d},\quad d=u_L-u_R$$

其中，d 为左右图的横坐标之差，即视差。视差越大，距离越近。

双目相机的缺点也比较明显，配置和标定较为复杂且视差计算非常消耗计算机资源，需要使用 GPU 和 FPGA 设备加速后，才能实时输出整张图的深度信息。

常用的双目相机有 ZED 双目相机，小觅双目摄像头。

图 2-7　ZED 双目相机

图 2-8 小觅双目摄像头（深度版）

深度相机

深度相机也称为 RGB-D 相机，是 2010 年左右开始兴起的一种相机，其最大特点是可以通过红外结构光或者 Time-of-Flight 原理，通过主动向物体发射光并接收返回的光，测出物体与相机之间的距离。测量深度后，RGB-D 相机通常按照生产时的各相机摆放位置，自动完成深度和彩色图像之间的匹配，输出一一对应的彩色图和深度图。然后通过读取色彩信息和距离信息，计算像素的三维坐标，生成点云。对 RGB-D 数据，既可以在图像层面处理也可以在点云层面处理。

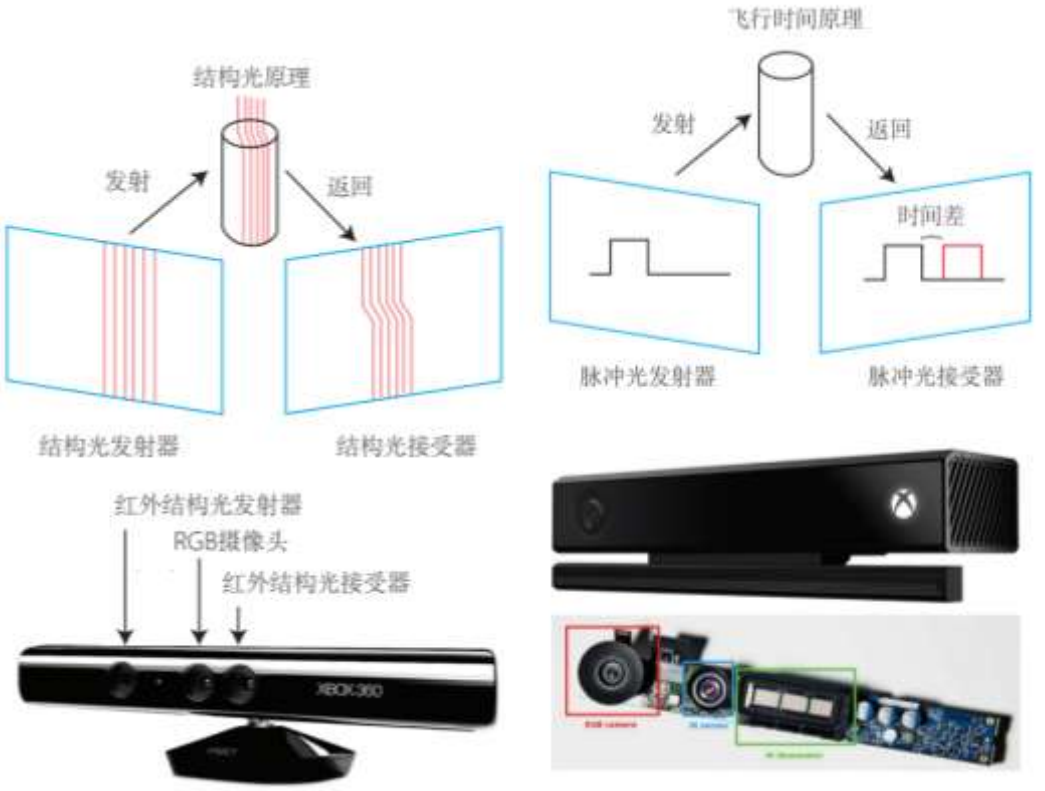


图 2-9 Kinect 深度测量原理

RGB-D 相机还存在着测量范围小，噪声大、视野小、易受光照影响、无法测量透射材质等诸多问题，主要应用于室内，室外则较难应用。目前常用的 RGB-D 相机包括 Kinect/Kinect v2、Xtion Pro Live、RealSense。

图 2-10　RealSense 深度相机和华硕 Xtion pro live 深度相机

三 视觉 SLAM 的结构及模块

结构概述

经典的视觉 SLAM 框架如图所示：

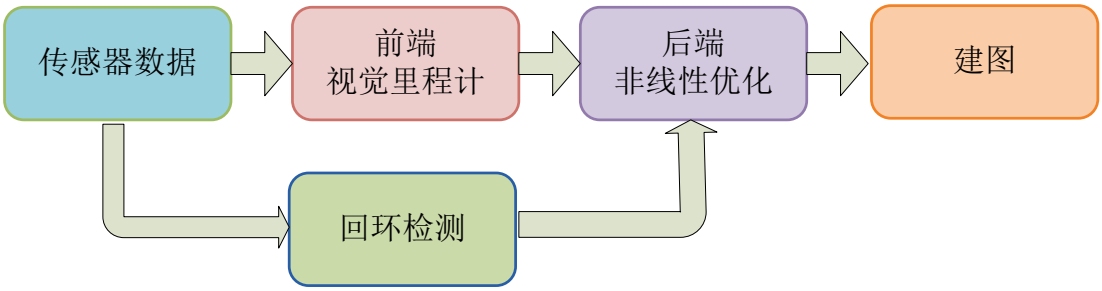


图 3-1　视觉 SLAM 整体流程图[23]

视觉 SLAM 的主要可分为视觉里程计前端和优化后端，回环检测也是对视觉里程计的优化。

1） 获取传感器信息：在视觉里程计中主要读取相机采集的图像信息并进行预处理，在机器人或无人机平台上，还可能需要读取惯性传感器等信息。

2） 视觉里程计（Visual Odometry，VO）。即通过估计相邻图像之间的相机运动，获得相机的运动轨迹和局部地图。

3） 后端优化。后端接收由视觉里程计获得的相机位姿以及回环检测信息，通过优化算法得到全局一致的轨迹和地图。

4） 回环检测。通过回环检测判断机器人是否到达曾经访问过的位置，如果检测到回环则将信息传递给后端进行位姿和地图优化。

5） 建图。即根据估计的位姿和轨迹，建立与任务要求对应的地图。

视觉 SLAM 的框架经过十多年的发展逐渐建立起来的，这个框架本身及其所包含的算法基本定型。下面将对此框架中的几个模块进行详细介绍。

视觉前端 I

视觉里程计的主要问题是根据相邻图像信息估计出相机的位姿变化，为后端优化提供较好的初始值。而视觉 SLAM 根据视觉里程计所用算法的不同可以分为特征点法和直接法。

因为图像本身是一个由亮度和色彩组成的矩阵，直接对矩阵进行运动估计，不仅非常困难

而且计算量大，特征点法即通过提取图像中具有代表性，且在视角发生少量变化后保持不变的点称之为特征点，在对特征点进行数据关联基础上进行位姿估计。

一、特征点

在计算机视觉领域，研究者设计了许多稳定的局部图像特征，其中点特征的使用最广泛，常用点特征有 SIFT、SURF、ORB、AKAZE 等点特征，相比于朴素的角点，这些人工设计的特征点具有以下性质：

1. 可重复性：即相同的特征区域能在不同的图像中找出；
2. 可区别性：即不同的特征区域具有不同的表达方式；
3. 高效率性：即在同一图像中特征点的数量应远小于像素的数量；
4. 本地性：特征仅与一小片图像区域相关。

特征点是由两部分组成，关键点（Key-point）和描述子（Descriptor）。关键点是指该特征点在图像中的位置，部分特征还具有方向和大小等信息；描述子是按照“外观相似的特征应具有相似的描述子”的原则设计的。下面对几种较为常用的特征进行介绍：

1、SIFT 特征算法[1]

1999 年，加拿大教授 David G.Lowe 总结了现有的基于不变量技术的特征检测方法，提出**SIFT**(Scale Invariant Feature Transfor)尺度不变特征变换，并在 2004 年完善总结。SIFT 特征对旋转、尺度缩放、亮度变化等保持不变性，是一种非常稳定的局部特征。其算法还具有独特性好、多量性和可扩展性的优点，即适用于在海量特征数据库中进行快速、准确匹配，及时图像中只有少量物体也能产生大量 SIFT 特征向量，同时可以与其他形式的特征向量联合。

SIFT 特征提取步骤可分为四步：

1）尺度空间的极值检测。通过高斯函数对图像模糊处理以及降采样得到图像高斯金字塔，再使用高斯金字塔每组中相邻上下两侧图像相减得到高斯查分图像，进行极值检测。

2）定位关键点。为了寻找极值点，每一个像素点要和它所有的相邻点比较，看其是否比它的图像域和尺度域的相邻点大或者小。中间的检测点和它同尺度的 8 个相邻点和上下相邻尺度对应的2×9个点共 26 个点比较，以确保在尺度空间和二维图像空间都检测到极值点，达到良好的尺度不变性。

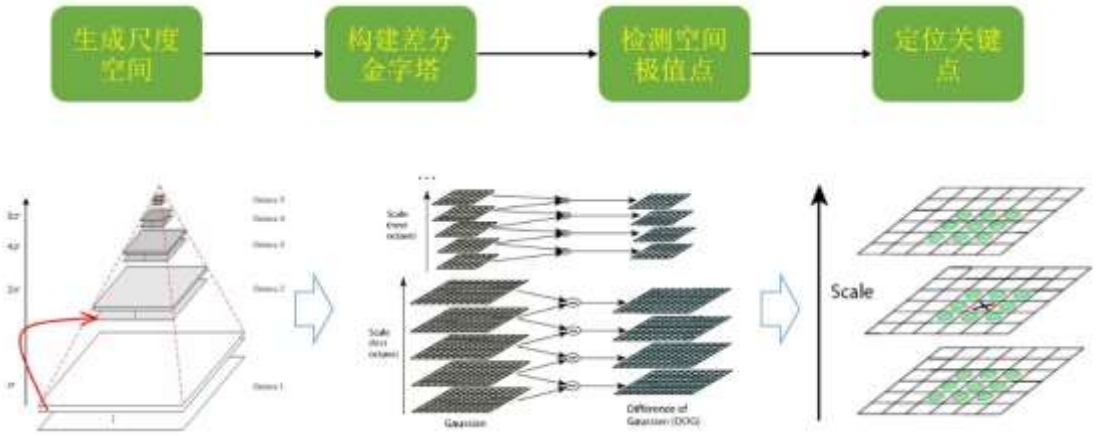


图 3-2　SIFT 算法构建尺度空间和定位关键点

3）确定关键点方向。为了使描述符具有旋转不变性，需要利用图像的局部特征为给每一个关键点分配一个基准方向。使用图像梯度的方法求取局部结构的稳定方向。在完成关键点的梯度计算后，使用直方图统计邻域内像素的梯度和方向。梯度直方图将 0~360° 的方向范围分为 36 个柱(bins)，其中每柱 10 度。直方图的峰值方向代表了关键点的主方向。

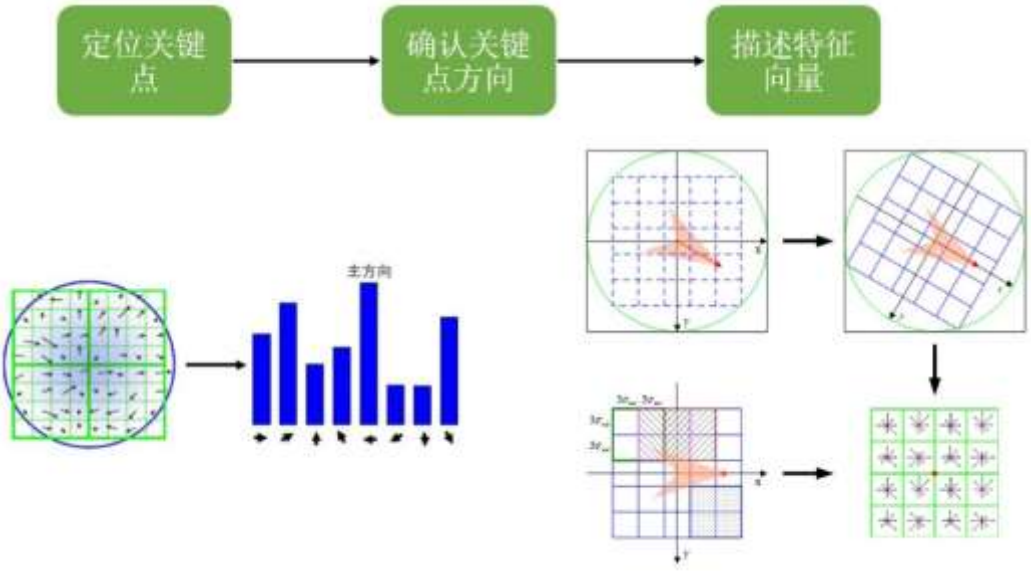


图 3-3　确定 SIFT 特征关键点方向

4）生成特征描述子。SIFT 描述子是关键点邻域高斯图像梯度统计结果的一种表示。为了保证特征矢量的旋转不变性，要以特征点为中心，在附近邻域内将坐标轴旋转 θ（特征点的主方向）角度。旋转后以主方向为中心取 8×8 的窗口。每个小格代表为关键点邻域所在尺度空间的一个像素，求取每个像素的梯度幅值与梯度方向，箭头方向代表该像素的梯度方向，长度代表梯度幅值，然后利用高斯窗口对其进行加权运算。最后在每个 4×4 的小块上绘制

8 个方向的梯度直方图，计算每个梯度方向的累加值，即可形成一个种子点。每个特征点由 4 个种子点组成，每个种子点有 8 个方向的向量信息。在实际的计算过程中，为了增强匹配的稳健性，Lowe 建议对每个关键点使用 4×4 共 16 个种子点来描述，这样一个关键点就可以产生 128 维的 SIFT 特征向量。

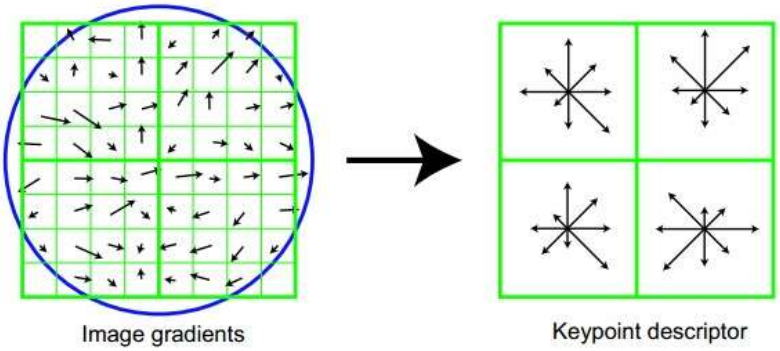


图 3-4　生成 SIFT 描述子

2、SURF 特征算法[2]

Speeded Up Robust Features（SURF，加速稳健特征），是一种稳健的局部特征点检测和描述算法。最初由 Herbert Bay 发表在 2006 年的欧洲计算机视觉国际会议（Europen Conference on Computer Vision，ECCV）上，并在 2008 年正式发表在 Computer Vision and Image Understanding 期刊上。Surf 是对 Sift 算法的改进，提升了算法的执行效率，为算法在实时计算机视觉系统中应用提供了可能。

SURF 特征的计算流程分为以下步骤：

1） 构建 Hessian（黑塞矩阵），生成所有的兴趣点，用于特征的提取。构建 Hessian 矩阵的过程对应于 Sift 算法中的高斯卷积过程。

2） 构建尺度空间。Sift 中下一组图像的尺寸是上一组的一半，同一组间图像尺寸一样，但是所使用的高斯模糊系数逐渐增大；而在 Surf 中，不同组间图像的尺寸都是一致的，不同的是不同组间使用的盒式滤波器的模板尺寸逐渐增大，同一组间不同层间使用相同尺寸的滤波器，但是滤波器的模糊系数逐渐增大

3） 特征点的定位过程 Surf 和 Sift 保持一致，将经过 Hessian 矩阵处理的每个像素点与二维图像空间和尺度空间邻域内的 26 个点进行比较，初步定位出关键点，再经过滤除能量比较弱的关键点以及错误定位的关键点，筛选出最终的稳定的特征点。

4） 特征点主方向分配，在 Surf 中，采用的是统计特征点圆形邻域内的 harr 小波特征。即在特征点的圆形邻域内，统计 60 度扇形内所有点的水平、垂直 harr 小波特征的

总和，然后扇形以 0.2 弧度大小的间隔进行旋转并再次统计该区域内 harr 小波特征值之后，最后将值最大的那个扇形的方向作为该特征点的主方向。

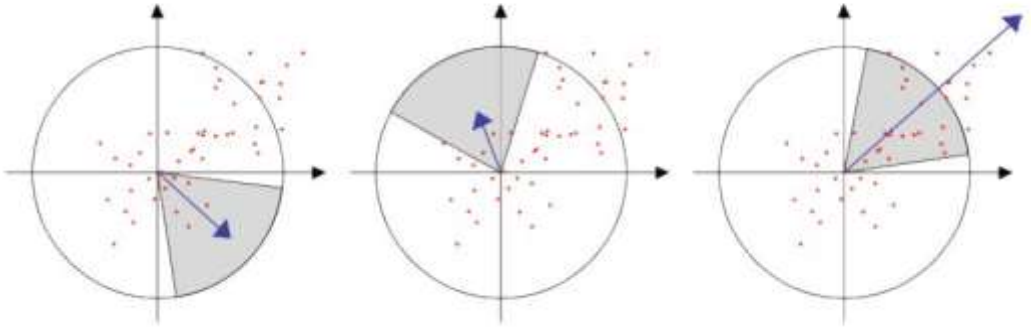


图 3-5　小波响应确定特征点方向

5） Surf 算法中，也是在特征点周围取一个4×4的矩形区域块，但是所取得矩形区域方向是沿着特征点的主方向。每个子区域统计 25 个像素的水平方向和垂直方向的 haar 小波特征，然后使用提取结果统计$\sum dx$，$\sum dy$，$\sum|dx|$，$\sum|dy|$四个值作为该子区域的特征，一个关键点就可以使用 16 个子区域的特征联合表示，即 64 维向量。

3、ORB 特征算法[3]

ORB（Oriented FAST and Rotated BRIEF）是一种快速特征点提取和描述的算法。此算法分为两部分，特征点提取和特征点描述。特征提取是由 FAST（Features from Accelerated Segment Test）算法发展来的，特征点描述是根据 BRIEF（Binary Robust Independent Elementary Features）特征描述算法改进的。ORB 特征是将 FAST 特征点的检测方法与 BRIEF 特征描述子结合起来，并在它们原来的基础上做了改进与优化。据说，ORB 算法的速度是 sift 的 100 倍，是 surf 的 10 倍。

其算法流程分为以下步骤：

1） 提取 FAST 角点。在图像中选取像素 p，假设它的亮度为I_p，然后设置一个阈值 T (比如I_p 20%)。以像素 p 为中心，选取半径为 3 的圆上的 16 个像素点，假如选取的圆上，有连续的N（通常取 12）个点的亮度大于I_p+T或小于I_p-T，那么像素 p 可以被

认为是特征点。通过对图像的每个像素点进行检测提取出所有 FAST 角点。

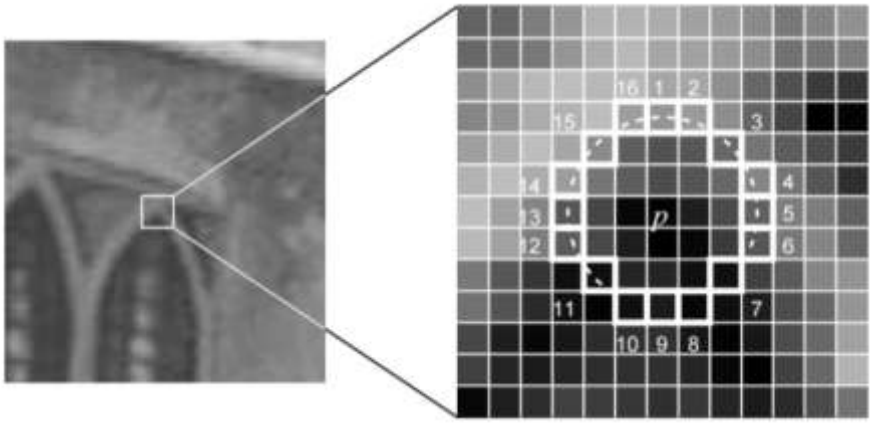


图 3-6　提取 FAST 角点

2） 而特征的旋转是由灰度质心法（Intensity Centroid）实现的。我们稍微介绍一下。 质心是指以图像块灰度值作为权重的中心。其具体操作步骤如下：

1. 在一个小的图像块 B 中，定义图像块的矩为：

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y) \quad p,q = \{0,1\}$$

2. 通过矩可以找到图像块的质心：

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

3. 连接图像块的几何中心O与质心C，得到一个方向向量，于是特征点的方向可以定义为：

$$\theta = \arctan(m_{01}/m_{10})$$

通过以上方法，FAST 角点便具有了尺度与旋转的描述，大大提升了它们在不同图像之间 表述的鲁棒性。所以在 ORB 中，把这种改进后的 FAST 称为 Oriented FAST。

BRIEF[19]是一种二进制描述子，它的描述向量由许多个 0 和 1 组成，这里的 0 和 1 编码了关键点附近两个像素（比如说 p 和 q）的大小关系：如果 p 比 q 大，则取 1，反之就取 0。如果我们取了 128 个这样的 p 和 q，最后就得到 128 维 0 和 1 组成的向量

利用 ORB 在 FAST 特征点提取阶段计算了关键 点的方向，所以可以利用方向信息，计算了旋转之后的“Steer BRIEF”特征，使 ORB 的 描述子具有较好的旋转不变性。

4、AKAZE 特征算法[4]

SIFT、SURF 算法用金字塔策略构建高斯尺度空间，但是此构造方法存在一个重要的缺点：高斯模糊不保留对象边界信息并且在所有尺度上平滑到相同程度的细节与噪声，影响定位的准确性和独特性。 针对高斯核函数构建尺度空间的缺陷，有学者提出了非线性滤波构建尺度空间：双边滤波、非线性扩散滤波方式。

AKAZE 特征算法利用非线性扩散滤波的优势获取计算量低的特征，再引入快速显示扩散数学框架 FED 来快速求解偏微分方程。采用 FED 来建立尺度空间要比当下其它的非线性模式建立尺度空间都要快，同时比 AOS 更加准确。然后引入一个高效的改进局部差分二进制描述符(M-LDB)，较原始 LDB 增加了旋转与尺度不变的鲁棒性，结合 FED 构建的尺度空间梯度信息增加了独特性。与 SIFT、SURF 算法相比，AKAZE 算法更快同时与 ORB、BRISK 算法相比，可重复性与鲁棒性提升很大。

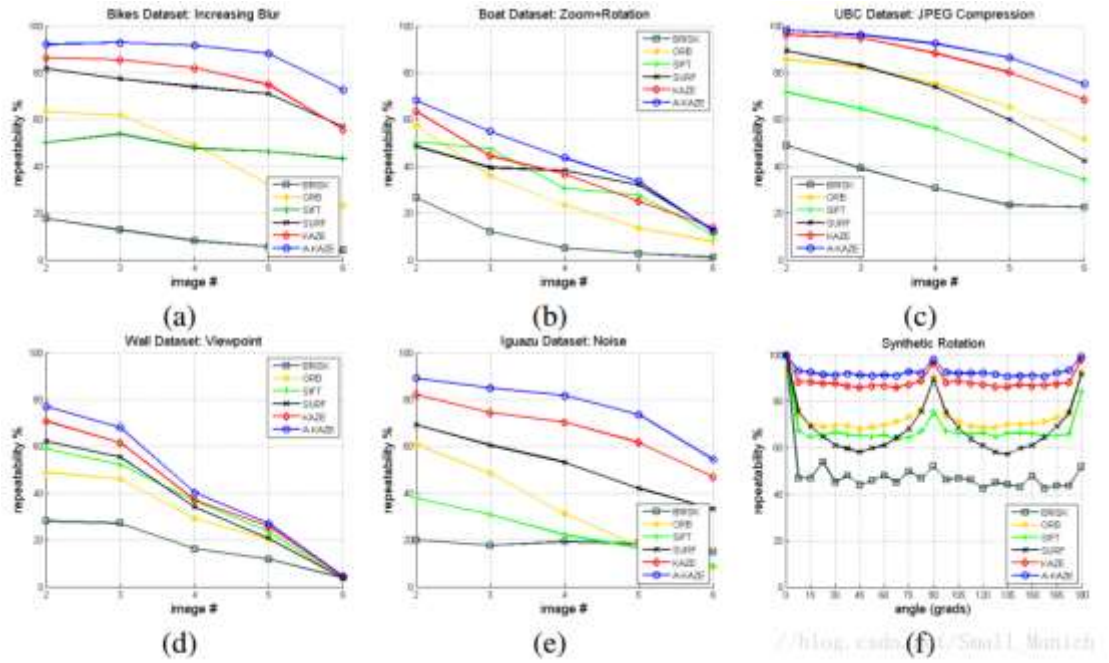


图 3-7 牛津标准数据集进行尺度缩放旋转等算法性能对比

这四种点特征提取算法各有优缺点，可依据计算条件和应用要求进行选择，其中 ORB 特征因其计算速度快且具有旋转不变性的优点，使用最为广泛。

二、特征匹配

特征匹配是视觉 SLAM 中极为关键的环节，解决了数据关联的问题，即利用描述子之间的距离确定当前图像帧检测的特征点与之前的图像帧的特征点之间的对应关系，距离越小，表

示特征点之间越相似。同时，因为场景中经常存在大量重复的纹理使得特征描述非常相似，将造成特征点的错误匹配。

目前常用的特征匹配算法有两种：暴力匹配（Brute-Force Mather）和快速近似最近邻（FLANN）。

暴力匹配其实是一种遍历算法，若在图 1 中提取了 M 个特征点，在图 2 提取了 N 个特征点，计算图 1 中特征点与图 2 中 N 个特征点之间的相似程度，然后排序，选择相似程度最高的一个特征点作为匹配点。特征之间的相似程度经常用描述子的距离来描述，对于 SIFT 和 SURF 特征，常用欧式距离来衡量，而 ORB 特征则是采用汉明距离（Hamming distance）作为度量。

当特征点数量很大时，暴力匹配的计算量将变得很大，为实现特征点的快速匹配采用 FLANN 算法进行匹配，其算法可分为建立 KD-tree 搜索树和在 KD-tree 中搜索最近邻两个环节。

由于这些匹配算法理论已经成熟，而且已集成到 OpenCV，可直接调用 OpenCV 中的函数进行特征匹配，而无需详细了解其匹配原理。

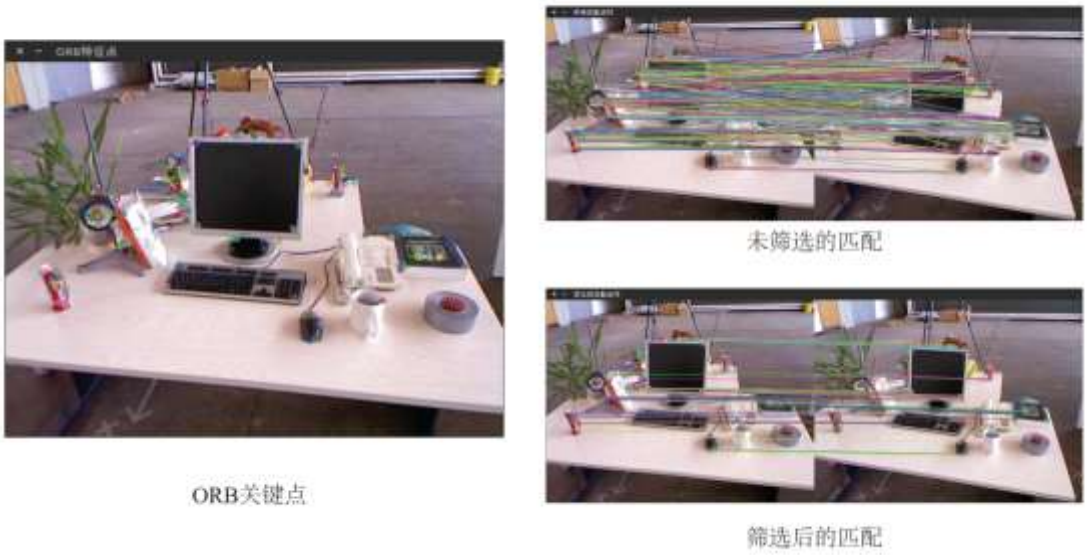


图 3-8 特征提取与匹配结果

三、位姿估计

由匹配的特征点进行位姿估计一般有 2D-2D，3D-2D 和 3D-3D 三种方式。

2D-2D 的方法是通过计算两帧图像之间的本质矩阵，然后通过 SVD 分解法得到旋转矩阵和平移向量，加上计算平移向量的相对尺度，可得到相机的运动估计。

3D-2D 是通过采用最小化重投影误差的方法计算位姿变换。最小化重投影误差的一般形式如下所示，其目的是找出使得该式最小的T_k：

$$\arg\min_{T_k} \sum_i \left\| p(k,i) - p(k-1,i) \right\|^2$$

其中，$p(k,i)$是第i个特征点在第k帧图像中的位置，$p(k-1,i)$是第i个特征点由$k-1$帧图像推算的三维点通过估计的变换矩阵重投影到第k帧图像中的位置。此类问题一般称之为 PnP（perspective from n points）问题，PnP 问题至少需要 3 组匹配点进行求解，也称为 P3P。

3D-3D 的方法则是从三维对应位姿误差估计运动，其一般方式是通过最小化两个三维特征点集的距离，如下式：

$$\arg\min_{T_k} \sum_i \left\| X_k^i - T_k X_{k-1}^i \right\|$$

其中，i表示第i个特征点，T_k为所要求的位姿变换，X_k与X_{k-1}分别表示三维点的齐次坐标，如$X = [x, y, z, 1]^T$，3D-3D 方法的代表即 ICP 算法。

视觉前端 II

在上一小节，简要介绍了视觉 SLAM 中特征点法中涉及的特征提取和匹配算法以及根据特征匹配进行位姿估计的方法。特征点法在视觉里程计中占据着主流地位，但是也存在几个缺点：

1、 关键点的提取和描述子的计算非常耗时。

2、 只利用了图像的特征点，而舍弃了大部分可能可能有用的信息；

3、 在缺少明显纹理信息、特征缺乏的情况下特征点数量明显减少，可能找不到足够多的匹配点进行位姿估计；

4、 在人工环境中往往存在许多重复或相似的纹理，增加了错误匹配的几率，导致运动估计的精度不可靠。

为避免提取关键点和计算描述子，直接跟踪像素的亮度信息。此方法还能充分利用环境中的几何信息，对特征点匮乏的环境有更高的准确性和鲁棒性。

直接法是从光流法演变而来，具有相同的假设条件——灰度不变假设，即同一个空间点的像素灰度值，在各个图像中是固定不变的。

光流是一种描述像素随时间在图像之间运动的方法，计算部分像素的运动称为稀疏光流，

计算所有像素则称为稠密光流，其中稀疏光流以 Lucas-Kanade 光流为代表，并可以在 SLAM 中用于跟踪特征点的位置。

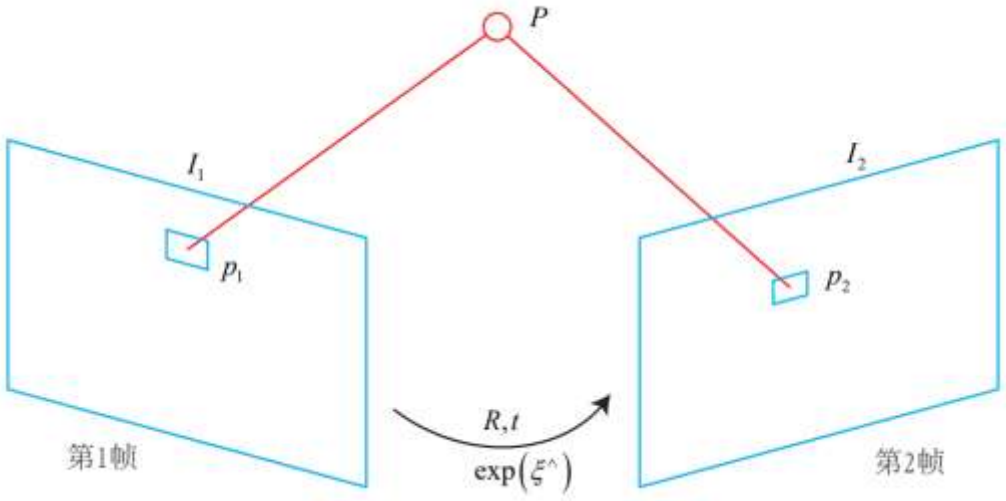


图 3-9　直接法示意图

而直接法和光流具有一定相似性。考虑两个时刻，不同位置获取的图片 I 和 2，P 的世界坐标为$[X,Y,Z]$。

在两个时刻的图像上的像素坐标记为：p_1 和 p_2。

以图像 1 位参照系，图像 2 的旋转矩阵和平移向量分别为：R 和 t。两个图像对应同一个相机，相机内参相同都是 K。

直接法根据当前位姿估计 R 和 t 寻找 p_2 的位置，但是 p_2 和 p_1 的外观可能因位姿不准确而存在明显差别，通过最小化光度误差对估计的位姿进行优化，从而得到较为准确的位姿估计。

直接法可根据跟踪像素点的不同分为稀疏直接法、半稠密直接法、稠密直接法。其中，稀疏直接法不计算描述子，仅提取关键点；半稠密直接法只考虑带有梯度的像素点，而舍弃像素梯度不明显的点；稠密直接法需计算所有像素，需要 GPU 加速才能满足实时性的要求。

下面将总结一下直接法的优缺点：

优点：1）节省了关键点提取和描述子计算的时间；

2）只要求具有像素梯度，不需要提取特征点，在纹理或特征缺失的场合具有较好的鲁棒性和准确性

3）可以构建半稠密或稠密地图，这是特征点法无法做到的。

缺点：1）优化过程中容易陷入局部最小值，对初始位姿估计和图像质量的要求较高；

2）在运动速度较快，拍摄频率不高的情况下，容易跟踪丢失；

3）灰度不变假设是一个强假设，对环境光照要求较高，而实际环境中往往难以满足；

4)直接法不使用特征点,无法像特征点法一样通过特征点进行重定位和回环检测。

优化后端

视觉里程计虽然能提供较为准确的位姿估计，但是再精确的传感器也存在误差，而且基于状态估计理论的视觉里程计也不可避免算法误差，后端优化环节就是为处理 SLAM 中的噪声和误差问题。在视觉 SLAM 中，前端和计算机视觉研 究领域更为相关，比如图像的特征提取与匹配等，后端则主要是滤波与非线性优化算法。

目前，视觉 SLAM 中常用的优化算法分为两类：滤波器方法和非线性优化方法。

其中，滤波器方法以扩展卡尔曼滤波（EKF）为主，由于假设了马尔可夫性质，只利用前一状态来估计当前状态的值，很难做到全局的优化。现在常用的非线性优化方法，则是把所有数据都考虑进来，放在一起优化，虽然会增大计算量，但效果好得多。非线性优化方法以图优化为代表，常用的图优化库包括 g2o，ceres 等。

Strasdat[22]等人详细地总结了基于滤波器法、马尔可夫随机场法和最优化方法在立体视觉和单目视觉传感器方案上的应用，并指出最优化方法较滤波器方法和其他方法有一定优势。

回环检测

SLAM 采用增量式位姿估计和建图，其位姿估计将不可避免地累计形成漂移。现在假设有一机器人经过一段时间运动后回到了原点，但是由于漂移，SLAM 估计的位置却不是原点。此时通过某种方式让机器人认识到已回到原点,然后把轨迹和地图调整到符合回环检测结果，从而消除漂移，这就是回环检测。

目前，在视觉 SLAM 通过判断图像之间的相似性完成回环检测，常用的回环检测方法是词袋模型（Bag-of-Words）[5]。

词袋模型把特征当成一个个单词，通过比较两张图片中出现的单词是否一致，来判断两张

图片是否是同一场景。

为了能够把特征归类为单词，我们需要训练一个字典。所谓的字典，就是包含了所有可能的单词的集合，为了提高通用性，需要使用海量的数据训练。

字典的训练其实就是一个聚类的过程，假设所有图片中共提供了 10,000,000 个特征，可以使用 K-Means 方法把他们聚成 1000,000 个单词。但是，如果只是用这 100,000 个单词来匹配的话，效率还是太低，因为每个特征需要比较 100,000 次才能找到自己对应的单词。为了提高效率，字典在训练的过程中，构建了一个 k 个分支，深度为 d 的树，如下图所示。只管上看，上层节点提供了粗分类，下层结点提供了细分类，直到叶子结点，利用这个树，就可以将时间复杂度降低到对数级别，大大加速了特征匹配。

建图

SlAM 顾名思义包含了两个任务，即定位与建图。经过以上三个环节，通过回环检测和后端优化对视觉里程计估计的位姿估计进行优化，可得到较为准确的相机位姿估计结果；而建图是指构建环境地图，即对环境进行描述，描述方式随 SLAM 的具体应用而定。例如，对于家用扫地机器人指在低矮平面中运动，二维地图已经能够满足需求；而无人机具有六个运动自由度，需要三维地图才能实现在空间的路径规划和避障；对于古建筑等的三维重建，则需要精密的点云地图来表现。

地图大体上可分为两类，度量地图和拓扑地图。

1、 度量地图

度量地图强调精确地表示地图中物体的位置关系，通常可以分为稀疏与稠密两类。稀疏地图只对特征点或路标进行表示，而忽略其他部分，稀疏地图已足以用于定位；而稠密地图着重于对传感器捕捉的环境信息进行建模，通常按照某种分辨率，由许多个小块组成。二维度量地图是许多个小格子（Grid），三维则是许多小方块（Voxel）。一般情况，一个小块含有占据、空闲、未知三种状态，以表达该格内是否有物体。当我们查询某个空间位置时，地图能够给出该位置是否可以通过的信息，这样的地图可以用于各种导航算法。但是稠密地图需要存储每一个格点的状态，耗费大量的存储空间，而且多数情况下地图的许多细节部分是无用的。

2、 拓扑地图

相比于度量地图的精确性，拓扑地图则更强调地图元素之间的关系。拓扑地图是由节

点和边组成的图，只考虑节点间的连通性。它放松了地图对精确位置的需要，去掉地图的细节问题，是一种更为紧凑的表达方式。但是，拓扑地图不擅长表达具有复杂结构的地图。

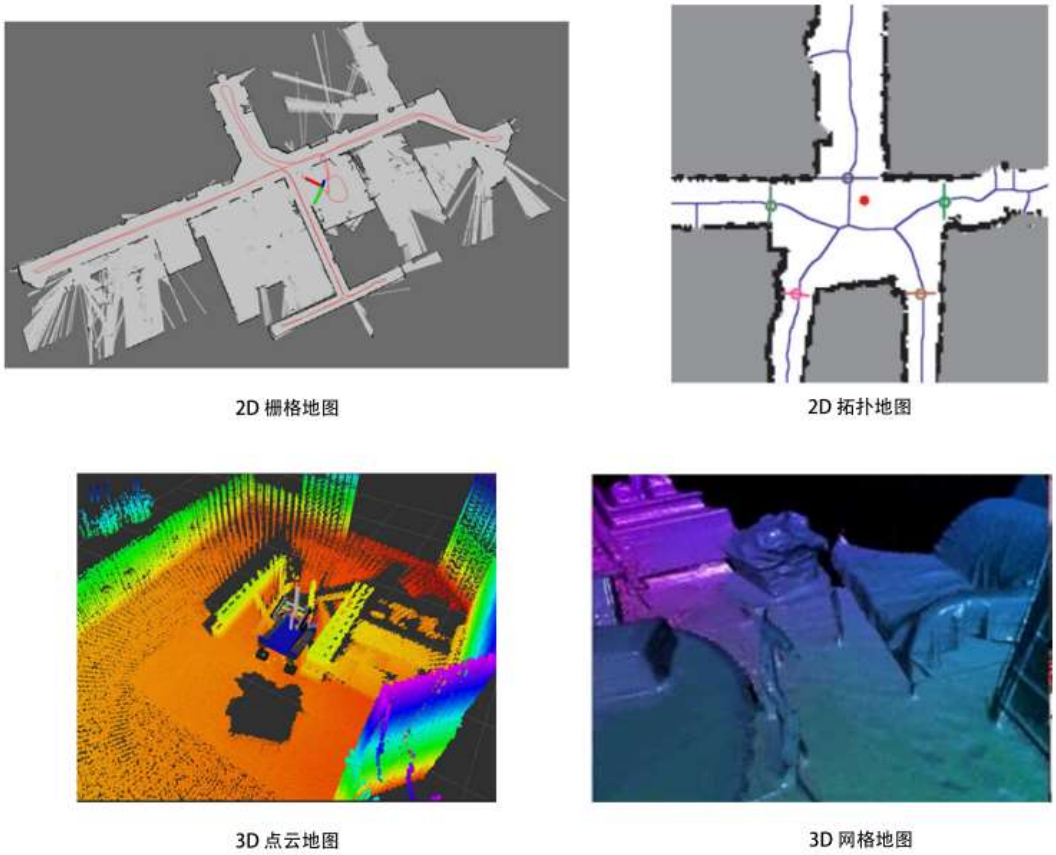


图 3-10　度量地图和拓扑地图

四 开源视觉 SLAM 方案

概述

自 2002 年起，视觉 SLAM 算法的研究开始引起关注， 经过十多年的研究发展，视觉 SLAM 已经取得许多突破性的进展，在机器人、无人机甚至手机等移动端都取得了实时运行的效果，其应用方式也逐渐丰富。表格按照传感器的不同整理了自 2007 年起一些重要的视觉同时定位与建图算法和视觉里程计算法。

常用开源视觉 SLAM 方案

方案名称	传感器形式	地址链接
MonoSLAM[6]	单目	https://github.com/hanmekim/SceneLib2
PTAM[7]	单目	http://www.robots.ox.ac.uk/~gk/PTAM/

ORB-SLAM[8]	单目	https://github.com/raulmur/ORB_SLAM
ORB-SLAM2[9]	单目/双目/RGB-D	https://github.com/raulmur/ORB_SLAM2
LSD-SLAM[10]	单目为主	https://github.com/tum-vision/lsd_slam
DSO[11]	单目	https://github.com/JakobEngel/dso
SVO[12]	单目	https://github.com/uzh-rpg/rpg_svo
DTAM[13]	RGB-D	https://github.com/anuranbaka/OpenDTAM
RGBD-SLAM-V2 [14]	RGB-D	https://github.com/felixendres/rgbdslam_v2
Elastic Fusion [15]	RGB-D	https://github.com/mp3guy/ElasticFusion
RTAB-MAP[16]	双目/RGB-D	https://github.com/introlab/rtabmap
OKVIS[17]	多目+IMU	https://github.com/ethz-asl/okvis
VINS-Mono[18]	单目+IMU	https://github.com/HKUST-Aerial-Robotics/VINS-Mono

基于单目相机的方案

目前，仅将单目相机作为传感器获取环境信息的视觉 SLAM 方案有 MonoSLAM，PTAM，ORB-SLAM/ORB-SLAM2，视觉里程计方案有 DSO 和 SVO。下面将对各种算法做一个简要介绍，着重介绍 PTAM、ORB-SLAM 和 SVO。

1、 MonoSLAM

2007 年，A. J. Davison 提出第一个实时的单目视觉 SLAM 系统，通过基于扩展卡尔曼滤波的后端实现对稀疏特征点（即 Shi-Tomasi 角点）的跟踪。在 EKF 中，每个特征点的位置服从高斯分布，因此可以用一个椭球表示其均值和不确定性，椭球在某个方向越长，表示在该方向的位置越不确定。MonoSLAM 现在看来存在许多弊端，例如应用场景很窄、路标数量有限、稀疏特征点容易跟踪失败等问题，但是在当时已经是一座里程碑。

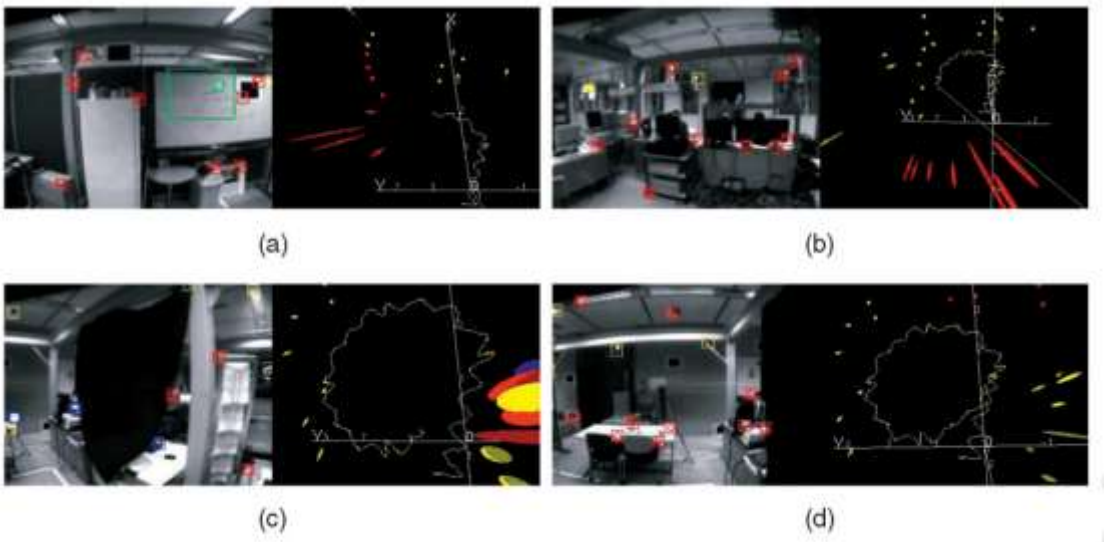


图 4-1　MonoSLAM 捕捉特征点及轨迹估计（（a）位姿估计初期；（b）继续运行不确定性增大；（c）回环前不确定性达到最大；（d）回环检测后消除漂移）

2、 PTAM

在 2007 年提出 PTAM，是一种基于关键帧的单目视觉 SLAM 算法，开创性地将位姿跟踪（Tracking）和建图（Mapping）作为两个并行处理的线程，并且也是第一个使用非线性优化的方案，通过引入关键帧机制，将几个关键帧串联起来然后优化其轨迹和地图。PTAM 的开源对于 V-SLAM 的发展来说意义深远，目前市面上很多 V-SLAM 系统都是基于 PTAM 的算法框架改进而来

Tracking 线程包括特征点提取、地图初始化、跟踪定位、关键帧提取及重定位；而 Mapping 线程则包括局部和全局 Bundle Adjustment、通过极线搜索加入地图点。

PTAM 通过对每个 FAST 角点计算其 Shi-Tomas 得分然后通过得分高低筛选候选特征点；再通过人工指定第一和第二关键帧，进行特征匹配，然后通过随机采样极大似然估计（MLESAC）进行迭代并求取两个关键帧的位姿变化，从而实现地图初始化；跟踪定位环节通过运动模型和基于 ESM 的视觉跟踪算法对当前的相机位姿进行预测，然后对特征点为中心选取 8x8 的像素块基于 SSD 计算其相似度，选择 SSD 值最小的特征点作为匹配点，然后通过最小化重投影误差的方式对前面预测的位姿进行优化。

PTAM 中通过四个指标对图像进行衡量，进入判断是否选为关键帧：1）跟踪质量，即跟踪过程中搜索到的点和搜索的点数比例；2）距离上一个关键帧是否有足够空间位移；3）距离上一个关键帧的提取是否间隔足够多图像帧或时间；4）关键帧缓存队列是否已满。而重定位环节是基于 SSD 算法对比当前帧和关键帧的高斯模糊小图相似度，选择相似度最高的关键帧，根据基于 ESM 的视觉跟踪算法计算出相机位姿作为当前帧的相机位姿。

地图构建的优化主要通过 Bundle Adjustment（BA），即平束光差法。本质是一个优化模型。目的是最小化重投影误差，用于最后一步优化，优化相机位姿和世界点。BA 是一个图优化模型，一般选择 Levenberg-Marquardt（LM）算法进行解算。

PTAM 同时也是一个增强现实软件，是早期的结合 AR 的 SLAM 工作之一，演示了酷炫的 AR 效果。和许多早期 SLAM 工作相似，存在明显的缺陷：应用场景小，容易跟踪丢失等。

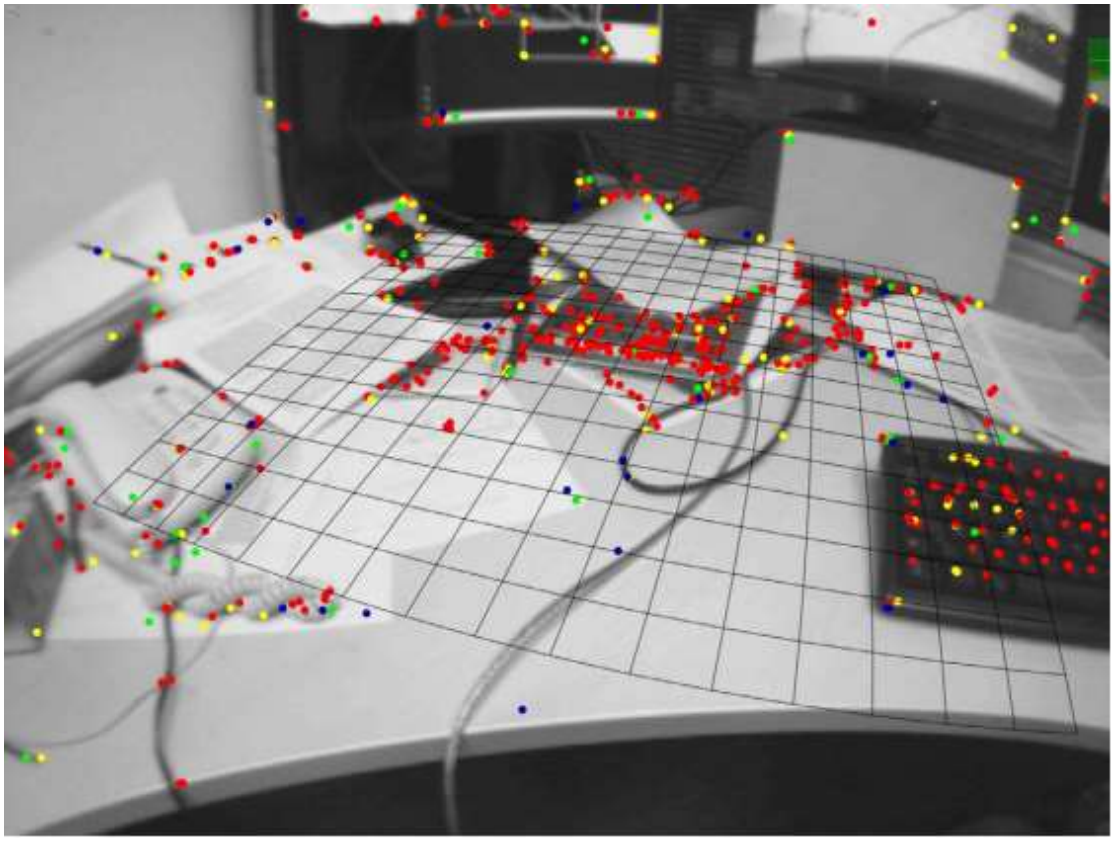


图 4-2 PTAM 的演示截图。即可以提供实时的定位和建图，也可以在虚拟平面叠加虚拟物体

图 4-3　PTAM 在 AR 中应用实例（左：Darth Vader 的激光枪是由相机的光轴瞄准防御狂热的 ewok 部落；右：用户使用虚拟的以摄像机为中心的放大镜和虚拟太阳的热量将图像映到光盘上）

3、 ORB-SLAM

ORB-SLAM 是 Raul Mur-Artal 在 2015 年发表的 SLAM 算法，并在 2017 年扩展了双目相机和深度相机接口，是一个基于特征点的实时单目 SLAM 系统，适合在大规模、小规模、室内外环境中运行。

ORB-SLAM 具有以下特点：

1） 创新使用三个线程完成 SLAM，实时跟踪特征点的 Tracking 线程、局部 Bundle Adjustment 的优化线程、全局位姿图回环检测和优化。Tracking 通过提取 ORB 特征并进行匹配，利用匀速运动模型计算最小化重投影误差，并进行图像帧的定位；局部建图：通过执行局部 BA 管理局部地图并优化；回环检测通过检测回环环并通过执行位姿图优化更正漂移误差；全局游湖是在在位姿图局部优化之后，计算整个系统最优结构和运动结果。

2） 整个系统围绕 ORB 特征进行计算，包括视觉里程计与回环检测的 ORB 字典。ORB 在 CPU 可进行实时计算，同时相比 Harris 角点等简单特征点，又具有良好的旋转和缩放不变性。ORB 提供描述子使其能够在大范围运动时能够进行回环检测和重定位。

3） 基于 ORB 特征的回环检测。ORB-SLAM 通过词袋模型完成全局位置识别和回环检测，优秀的回环检测算法保证 ORB-SLAM 能有效地防止累计误差，并且在丢失之后还能迅速找回。

4） ORB-SLAM 中建立 convisibility graph 保存了关键帧之间的连接，即以关键帧为节点，能观测到相同地图点的节点之间具有连线，其权值是相连关键帧观测到的形同地图点数量。而 Essential Graph 是通过一定阈值筛选过后的 Covisibility graph 的子集，利用最少的边将关键帧节点连接起来的子图成为 spanning graph。

ORB-SLAM 的三线程结构取得了非常好的跟踪和建图效果，能保证轨迹与地图的全局一致性，许多研究工作都是以 ORB-SLAM 作为标准，或在其基础上进行后续开发。同时 ORB-SLAM 也存在一些不足。首先，整个 ORB-SLAM 采用特征点进行计算，需对每一帧图像进行处理，非常耗时；其次，三线程结构也给 CPU 带来较大负担，在 PC 架构的 CPU 上才能是实时运行，移植到嵌入式设备仍有困难；最后，ORB-SLAM 的建图是稀疏特征点，只能满足定位需求而不能用于导航、避障等功能。

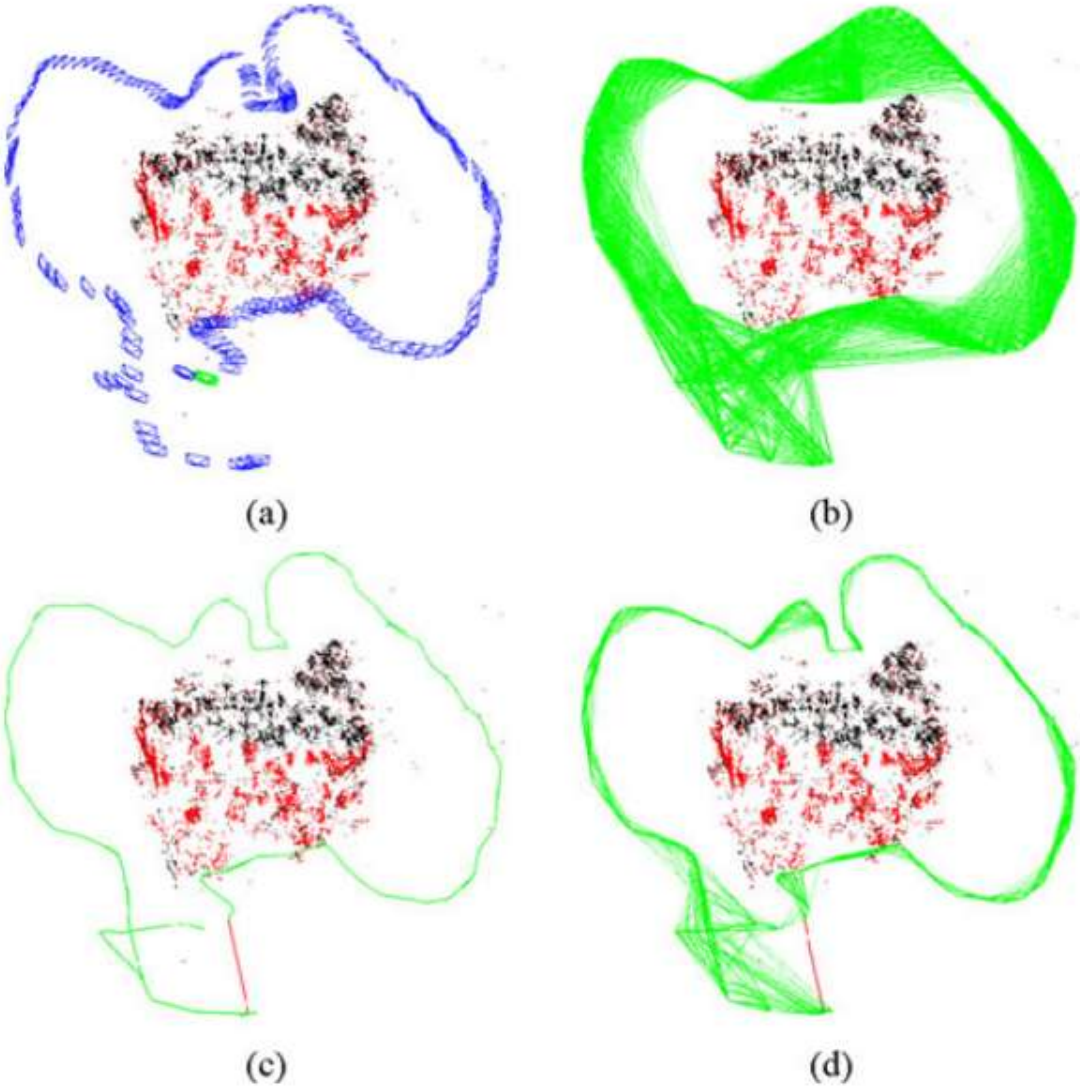


图 4-4　（a）ORB-SLAM 运行中关键帧（蓝色）、当前位置（绿色）、地图点（红色和黑色）；（b）Covisibility Graph；（c）Spanning tree；（d）Essential Graph

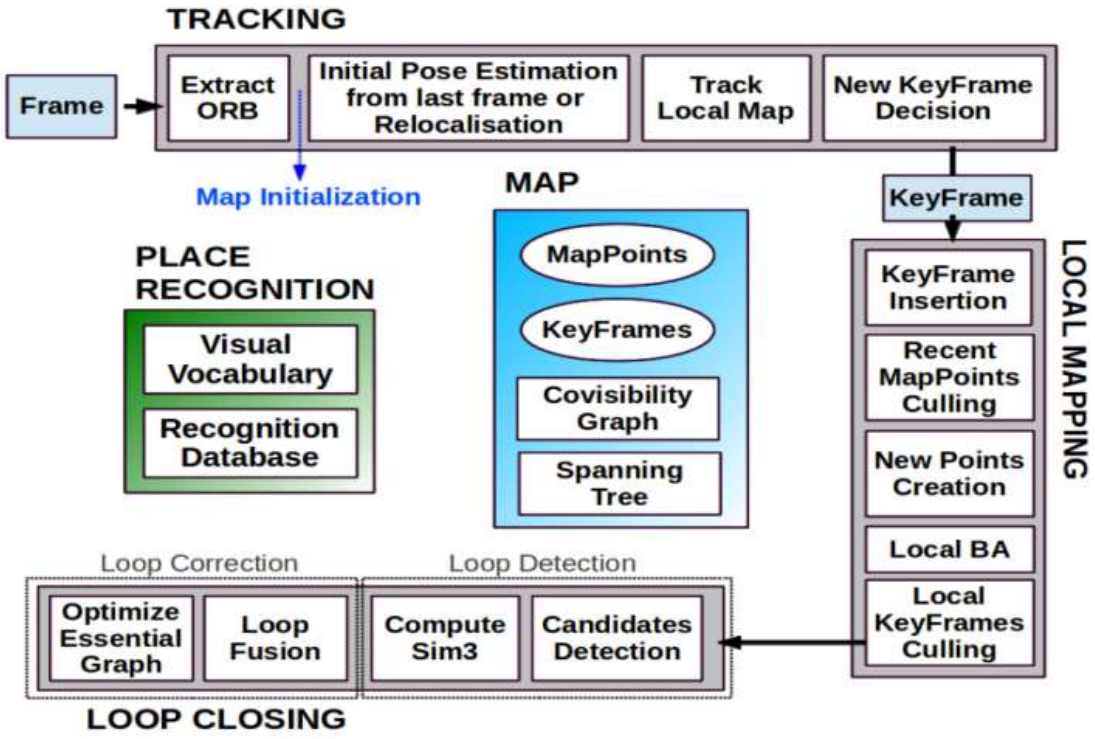


图 4-5　ORB-SLAM 的三线程结构图

4、 LSD-SLAM

Engel 于 2014 年首先提出单目 LSD-SLAM（Large-scale Direct SLAM）系统，标志着单目直接法在 SLAM 汇总的成功应用，其核心贡献是将直接法应用到半稠密单目 SLAM 中，无需计算特征点还能构建半稠密地图。此后 LSD-SLAM 扩展到双目和大视角相机，实现了手机端的 AR 应用等其他功能。

LSD-SLAM 的主要优点有：

1、 LSD-SLAM 的直接法是针对像素进行的。

2、 LSD-SLAM 在 CPU 上实现了半稠密场景的重建，这在之前的方案中很少见。基于特征点的方法只能是稀疏的，而进行稠密重建的方案大多要使用 RGB-D 传感器，或者使用 GPU 构建稠密地图。

3、 LSD-SLAM 的半稠密追踪使用了一些精妙的手段保证追踪的实时性与稳定性。在深度估计时，LSD-SLAM 首先用随机数初始化深度，在估计完后又把深度均值归一化，以调整尺度；在度量深度不确定时，不仅考虑三角化的几何关系，而且考虑了极限与深度的夹角，归纳成一个光度不确定性项；关键帧之间的约束使用了相似变换群，在后端优化中可以将不同尺度的场景考虑进来，减小了尺度漂移现象。

由于 LSD-SLAM 使用了直接法进行跟踪，所以它既有直接法的优点（对特征缺失区 域不敏感），也继承了直接法的缺点。例如，LSD-SLAM 对相机内参和曝光非常敏感，并且在相机快速运动时容易丢失。在回环检测部分，由于目前没有基于直接法的回环检测方式，因此 LSD-SLAM 必须依赖于特征点方法进行回环检测，尚未完全摆脱特征点的计算。

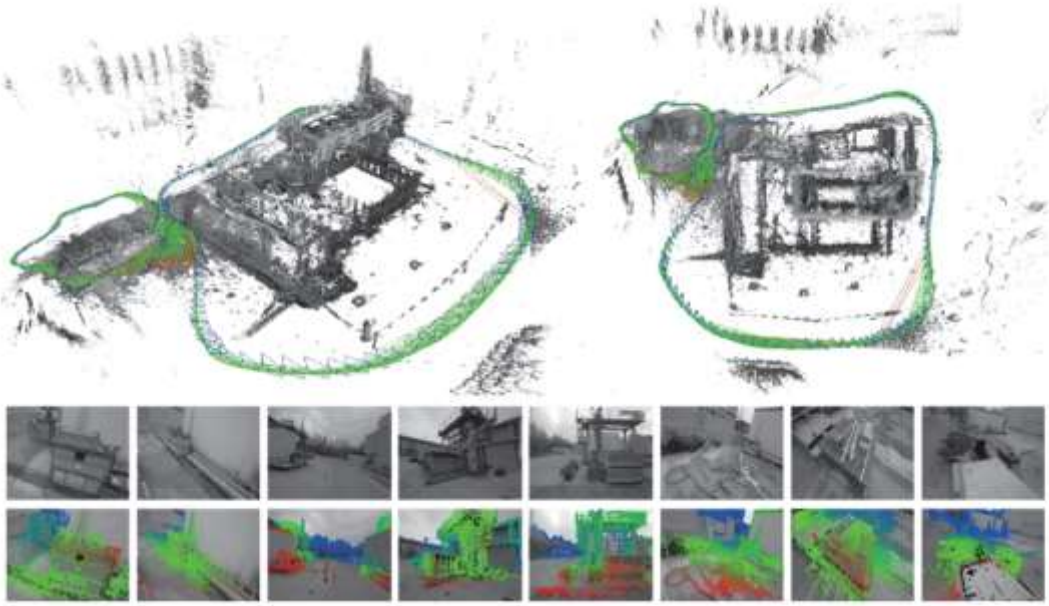


图 4-6　　LSD-SLAM 运行图片。上半部分为估计的轨迹与地图，下半部分为图像中被建模的部分，即具有较好的像素梯度的部分。

5、 DSO

DSO（Direct Sparse Odometry）是由慕尼黑工业大学 Engel 博士于 2016 年发布的一种单目稀疏直接法视觉里程计，在 2017 年扩展可双目功能。不同于特征点法需要通过匹配特征点进行数据关联，DSO 将数据关联与位姿跟踪放在一个统一的非线性优化框架中求解，是一个复杂的优化问题。其后端使用一个包含 5 到 7 个关键帧组成的窗口，通过设置一套方法管理关键帧的插入和删除。后端除了维护这个窗口中的关键帧和地图点，还维护与优化相关的结构。

直接法通过跟踪图像灰度来确定相机位姿，易受光照影响，DSO 提出广度标定，通过对相机的晕影、曝光时间、伽马响应进行标定以补偿其影响。统一的非线性优化框架和光度标定法使 DSO 在实时性和跟踪精度都具有良好的表现。

同时，DSO 也存在明显的缺陷，即它只是视觉里程计而非完整的 SLAM 系统，缺少回环检测、重定位、地图重用等在实际场景中必不可少的功能，而且 DSO 的代码清晰度和可

读性明显弱于 ORB-SLAM、SVO 等，使研究者很难以它为基础，展开后续的研究工作。

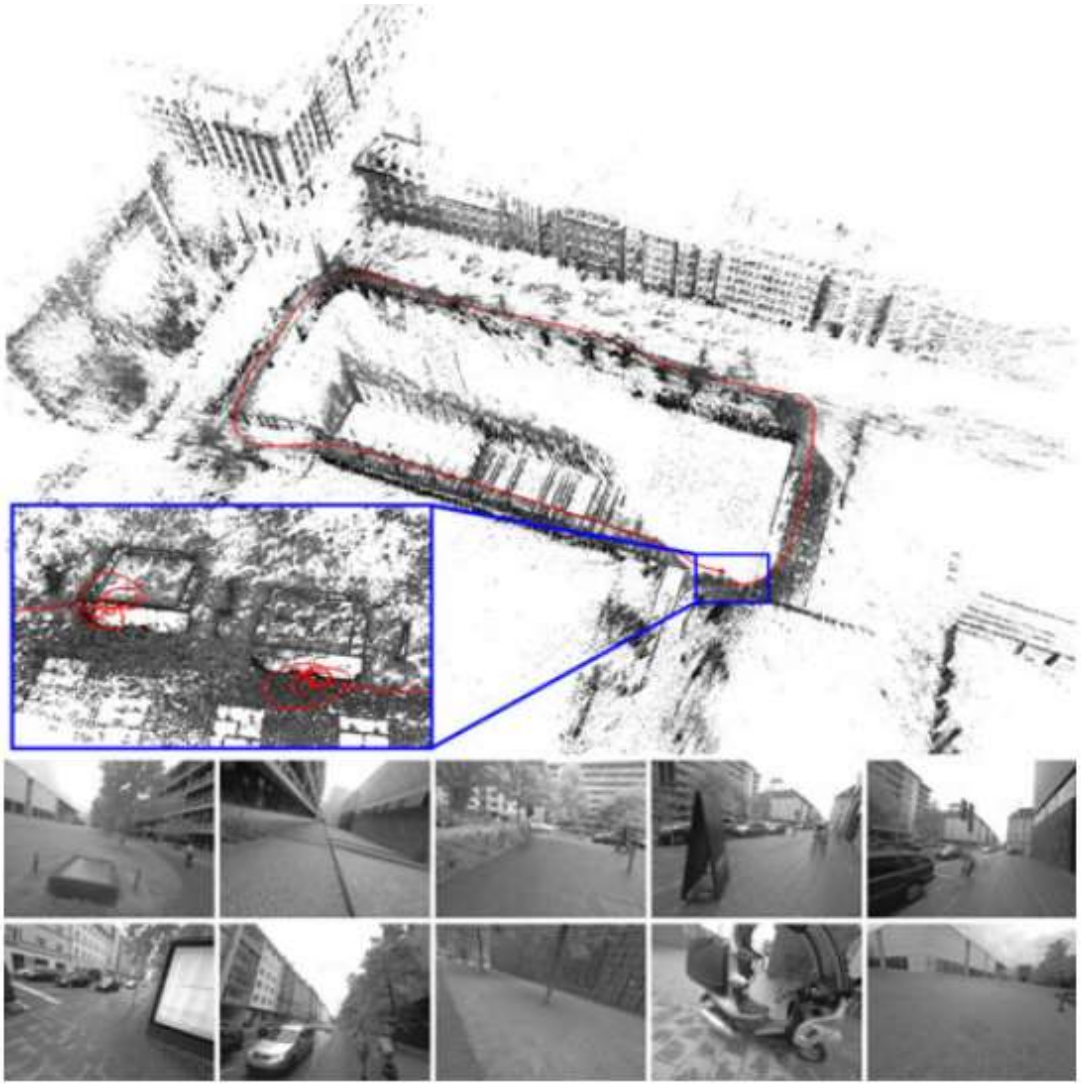


图 4-7　DSO-围绕建筑物骑行一圈得到的轨迹估计和建图效果

6、 SVO

SVO（Semi-direct Visual Odometry）是 2014 年由 Foster 等人提出的一种基于稀疏直接法的视觉里程计，正如其名，是一种半直接法，即将特征点法和直接法结合使用，是指通过对图像中提取特征点的图像块进行直接匹配来计算相机的位姿，因此 SVO 不必耗时去计算描述子也不必处理稠密或半稠密直接法那么多信息。

相比其他方案，SVO 最大的优势是速度极快，在 PC 平台其处理速度可达每秒 100 多帧，在其后续的 SVO2.0 中，速度更达到每秒 400 帧。这使得它非常适合计算条件受限的场合，如无人机、手持 AR/VR 设备的定位。无人机也是 Foster 等人开发 SVO 的目标应用平台。

SVO 另一个创新之处是提出深度滤波器的概念，并推到基于均匀-高斯混合分布的深度滤波器。其原理较为复杂，在此不做详细解释。SVO 将这种滤波器用于关键点的位置估计，

并使用逆深度作为参数化形式，使之能够更好地计算特征点位置。

SVO 的开源代码清晰易读，适合初学者作为第一个 SLAM 实例进行分析。

同时，像 DSO 一样作为不完整的 SLAM，因缺少后端优化和回环检测，也基本没有地图构建功能，虽然提升了速度和轻量化，但是也意味着 SVO 的位姿估计必然存在累计误差，而且没有重定位。

基于深度相机的方案

RGB-D 相机能够同时获取环境的 RGB 图像和每个像素的深度信息，相比单目相机和双目相机利用算法得到空间点的三维坐标，RGB-D 相机能更方便地获取空间点的三维坐标。现在基于 RGB-D 相机的视觉 SLAM 方案主要包括：DTAM、DVO、RGB-D v2、Elastic Fusion 和 RTAB-MAP。下面将对这几种 SLAM 方案作一个介绍。

1、 DTAM

DTAM（Dense Tracking and Mapping）由 Newcombe 等人于 2011 年提出，是基于直接法的 SLAM 方案。DTAM 直接对每个像素的深度数据进行反深度计算和优化，从而建立稠密地图并且稳定跟踪。DTAM 在稳定性和准确性上具有良好表现，但是因为需要对每个像素进行计算，使其难以满足实时性要求，需通过 GPU 加速。

2、 RGB-D v2

RGB-D v2 是由 Endres 等人于 2014 年提出的基于 RGB-D 相机的 SLAM 算法，是最早为 Kinect 设计的的 SLAM 系统之一。RGB-D v2 系统分为前端视觉里程计和后端图优化两部分组成。其系统流程图如下图所示：

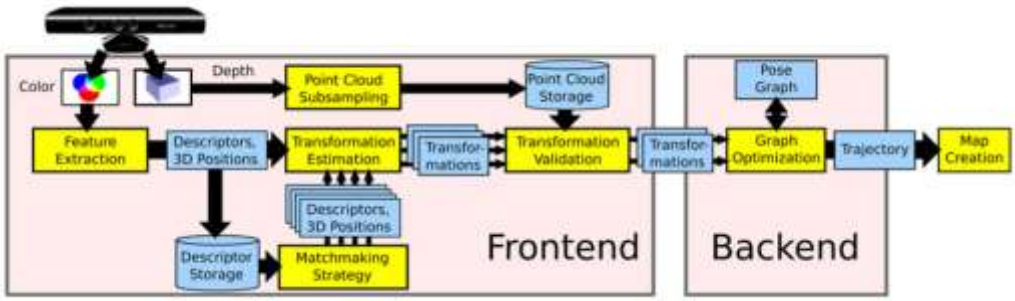


图 4-8　　RGB-D v2 的算法流程图

视觉里程计前端通过对每一帧 RGB 图像提取特征点并计算描述子，通过 RANSAC 算法和

ICP 算法进行帧间位姿估计，然后通过 EMM（Environment Measurement Model）模型判断位姿估计是否能否接受。RGB-D v2 的后端采用基于 g2o 图优化库的位子图优化，而回环检测是采用基于随机森林的随机回环。

它整合了 SLAM 领域里的各种技术：图像特征、回环检测、点云、图优化等等，是一个非常全面且优秀的程序。而且其用户界面（UI）也做得很漂亮，不仅能导出位姿估计结果，也能导出所构建的点云地图和八叉树地图，可以在其源代码上继续开发。其缺点也比较明显，特征提取、点云渲染都会消耗大量计算资源，导致算法的实时性差，如果相机运动过快容易跟踪丢失，且关键帧采集频率高，不适合作为长时间运行的 SLAM 方案。

3、 Elastic Fusion

Elastic Fusion 是 Whelan 等人在 2015 年提出的一种基于直接法的 SLAM 方案。Elastic Fusion 能够建立基于 Surfel 模型的全局一致性稠密地图，而且不需要进行位姿图优化。这种算法使用稠密的帧到模型的相机跟踪方法，尽可能频繁地使用模型到模型的表面闭环优化来接近地图分布，而全局闭环则用于减小随机漂移误差和获得全局一致性。

Elastic Fusion 通过使用 Surfel 模型能够实现效果不错的三维重建，三维重建不是像点云地图般由许多点云构成，而是像“面片”般相互重叠构成平面。Elastic Fusion 只适合于重建房间大小室内场景，不适合较大的场景。同时也存在一般基于 RGB-D 相机的直接法 SLAM 一样的问题，即不适合用于室外三维重建，而且计算量大，需要用 GPU 才能满足实时性需求。

Elastic Fusion 的算法流程图如下图所示：

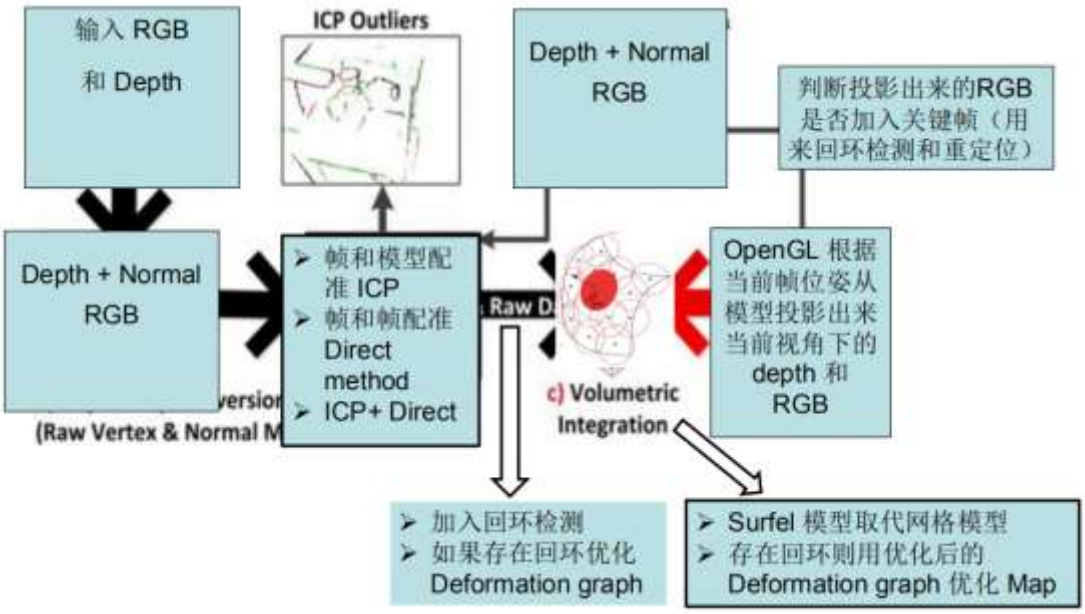


图 4-9　Elastic Fusion 的流程图

4、 RTAB-MAP

RTAB-MAP （Real Time Appearance-Based Mapping）是 RGB-D SLAM 中比较经典的方案。它实现了 RGB-D SLAM 中所以应该有的东西：基于特征的视觉里程计、基于词袋的回环检测、后端的位姿图优化，以及点云和三角网格地图。RTAB-MAP 支持一些常见的 RGB-D 和双目传感器，像 kinect、Xtion 等，且提供实时的定位和建图功能。因此，RTAB-MAP 给出了一套完整的 RGB-D SLAM 方案。目前我们已经可以直接从 ROS 中获得它二进制程序，此外，在 Google Project Tango 上也可以获取其 App。不过由于集成度较高，更适合作为 SLAM 应用而非研究。

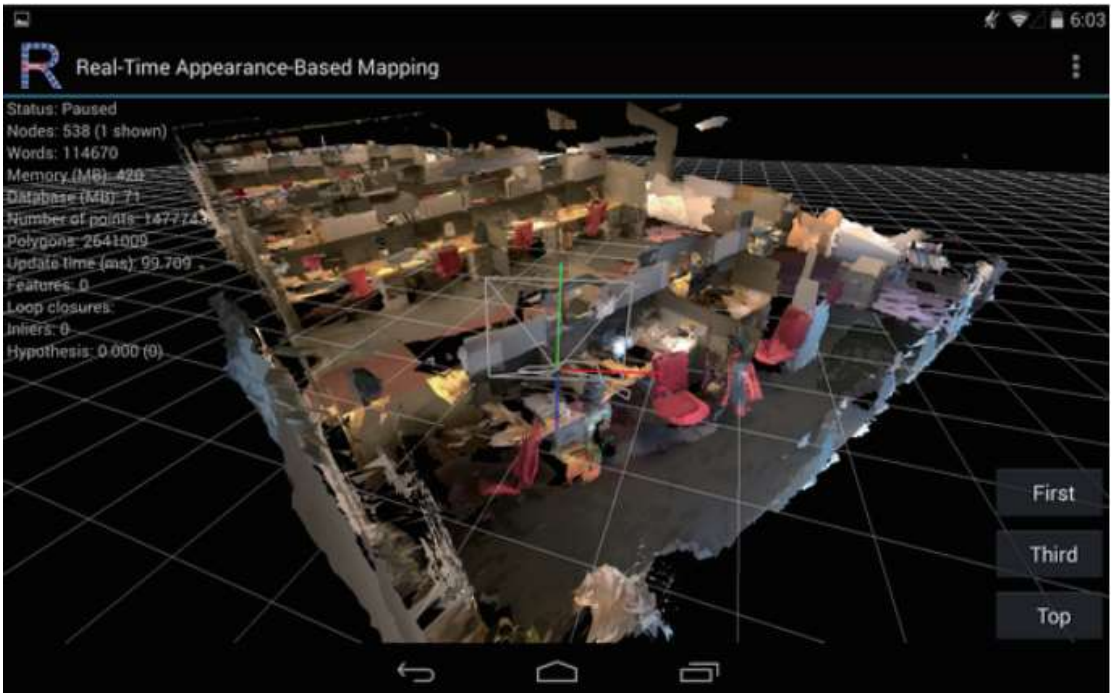


图 4-10　　RTAB-MAP 在 Google Project Tango 上的运行样例

多传感器融合方案

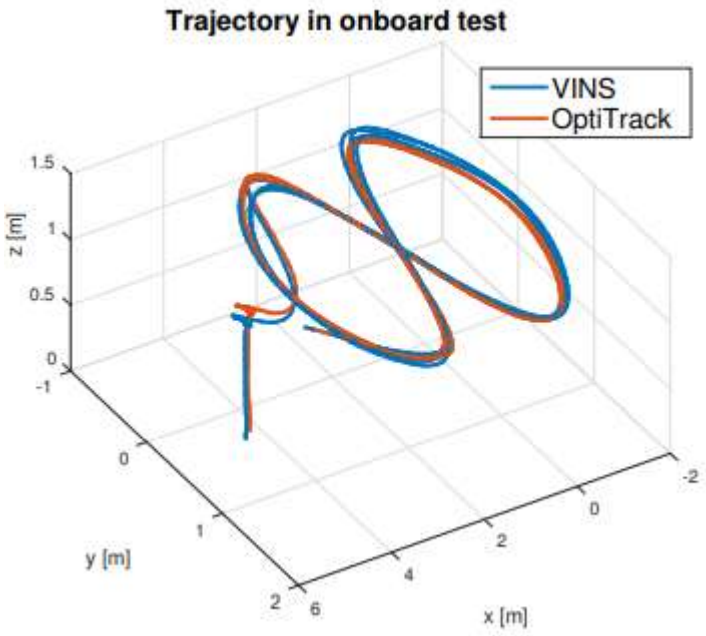
目前，多传感器融合的视觉 SLAM 方案主要通过与惯性测量元件 IMU 或者激光雷达进行配合，下面将对惯性视觉 SLAM 方案，OKVIS 和 Vins-mono 进行简要介绍。

OKVIS（Open Keyframe-based Visual Inertial SLaM）是 Stefan 等人提出的一种基于紧耦合、非线性优化的 IMU 与多目视觉的实时融合方法。紧耦合即表示 OKVIS 将视觉和 IMU 的误差项和状态量一起放入系统优化的能量函数中一起优化，这种做法无疑将增加其计算复杂性，但是其精度也将得到大幅提升。

Vins-Mono 是 2017 年香港科技大学沈邵劼课题组开源的单目视觉惯性状态估计器，提出

紧耦合滑窗 VINS 状态估计算法，能够在未知状态下稳健初始化，具有在线外参标定、统一定义在球面上的重投影误差、闭环检测和 4 自由度位子图优化的特点。

(a) (b)



(c)

图 4-11　室内无人机测试（（a）为用于测试的无人机；（b）中黄线为预设的无人机飞行轨迹；（c）为 VINS-Mono 估计的轨迹和真实轨迹）

图 4-12　VINS-Mono 在大尺度环境中的位姿估计结果，黄色部分为估计的位姿，红线表示回环检测

开源数据集及工具介绍

公开数据集和评价标准为算法的科学量化评估和目标对比提供了重要支持，并推动相关科技发展的水平。对于和机器人领域密切相关的 SLAM 问题，同样也有不少研究者提供了开源数据集，供 SLAM 研究者使用，下面就对这些重要数据集进行介绍。

1） TUM 数据集[20]

TUM 数据集是慕尼黑工业大学（Technische Universität München，TUM）的 Sturm 等人利用 Kinect 深度相机采集了 39 个图像序列及其相机的真实轨迹，采集方式包括利用手持相机采集以及利用搭载 Kinect 的 Pioneer 移动机器人采集图像，而图像序列涵盖了办公室和工厂环境，此数据集对算法可靠性要求较高。此数据集可通过下载.zip 和.bag 两种格式，可以根据需要自行选择。Sturm 等人还在此基础上提供了测评工具，这些工具是用 Python 写的脚本，通过运行脚本即可得到 SLAM 算法估计的轨迹和真实轨迹之间的绝对误差、相对误差以及用图形形象表示轨迹。

数据集的下载地址为：https://vision.in.tum.de/data/datasets/rgbd-dataset

评测工具下载地址为：https://svncvpr.in.tum.de/cvpr-ros-pkg/trunk/rgbd_benchmark/rgbd_benchmark_tools/

图 4-13　TUM 数据集中 fr1_desk2 中 RGB 图像及其匹配的深度图

2） KITTI 数据集

KITTI 数据集是由德国卡尔斯鲁厄工业学院和丰田美国技术研究院联合创办，是目前国际上最大的自动驾驶场景下的计算机视觉算法评测数据集，该数据集用于评测立体图像(stereo)，光流(optical flow)，视觉测距(visual odometry)，3D 物体检测(object detection)和 3D 跟踪(tracking)等计算机视觉技术在车载环境下的性能。KITTI 包含市区、乡村和高速公路等场景采集的真实图像数据，每张图像中最多达 15 辆车和 30 个行人，还有各种程度的遮挡与截断。整个数据集由 389 对立体图像和光流图，39.2 km 视觉测距序列以及超过 200k 3D 标注物体的图像组成。

KITTI 数据集针对不同的任务采用不同的评价准则，算法评测标准在 README 文件中有详细记载。

数据集下载地址：

http://www.cvlibs.net/datasets/kitti/eval_odometry.php

图 4-14　KITTI 数据集样本

3） EuRoC 数据集

EuRoC 数据集是苏黎世联邦理工学院（Eidgenössische Technische Hochschule Zürich，简称 ETH Zürich）的 Burri 等人为了给视觉-惯性联合定位算法提供可靠的数据集，利用小型无人机采集的视觉-惯性数据集。此数据集一共包含 11 个序列，每个图像序列包括同步的双目图像和 IMU 测量数据以及真实的相机轨迹，既有在慢速飞行情况下获得的清晰图像，也有运动模糊导致质量较差的图像，真实反映了无人机在飞行状态下采集图像的情况。

数据集的下载地址为：

https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets

图 4-15　ETH Machine hall

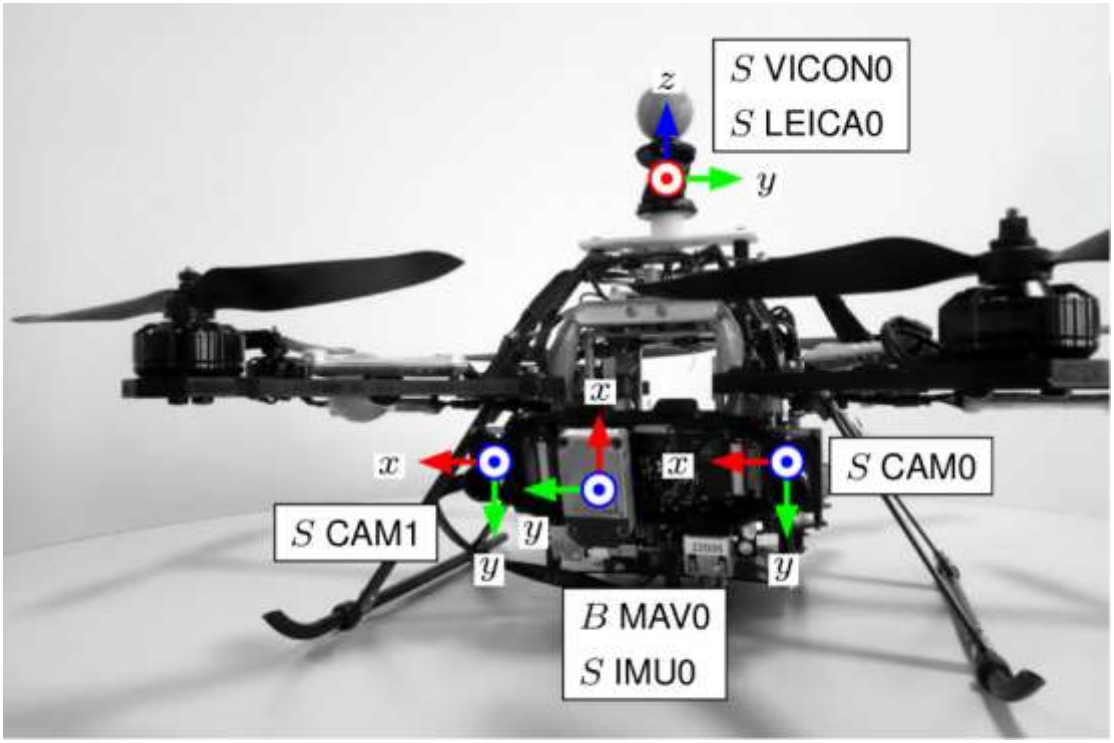


图 4-16　用于 EuRoc 数据采集的 Asctec Firefly hex-rotor helicopter

五 发展趋势

随着视觉 SLAM 在 AR 等移动端应用需求的扩展，以及三维重建、无人驾驶汽车等需求的增加，SLAM 的未来发展趋势可分为两类：其一是向轻量化、小型化方向发展；其二是利用高性能设备实现精密的三维重建和场景理解。

而视觉 SLAM 在算法方面的发展趋势主要有两个：多传感器融合和与深度学习结合。

一、多传感器融合

在视觉应用中，机器人或无人机通常不会只携带一种传感器，往往是多种传感器相互结合共同完成任务，这种冗余设计能够提高系统对不同环境的适应能力和减少故障发生的概率。惯性传感器 IMU 能够测量传感器本体的角速度和加速度，被认为与视觉传感器具有明显的互补性：

首先，IMU 虽然具有短时精确性，但是其测得的角速度和加速度都存在漂移，长期运行将产生较大误差，使得两侧积分得到的位姿数据非常不可靠；而相机在运动速度过快时容易出现运动模糊，或者前后两帧图像角度变化太大，重叠区域太少无法完成特征匹配。而 IMU 正好能补足相机数据无效时的位姿估计。其次，相机数据不存在漂移，能够有效估计并修正 IMU 漂移。最后，纯视觉 SLAM 不擅长处理动态场景，IMU 能够获得自身的运动信息，减少动态物体的影响。

除了 IMU，激光雷达也是与视觉 SLAM 相辅相成的传感器。激光 SLAM 系统通过对不同时刻两片点云的匹配与比对，计算激光雷达相对运动的距离和姿态的改变，也就完成了对机器人自身的定位。激光雷达距离测量比较准确，误差模型简单，在强光直射以外的环境中运行稳定，点云信息本身包含直接的几何关系。而视觉传感器具有能获取环境丰富的纹理信息，基于激光点云无法分辨的场景，通过图像能偶轻易分辨，同时容易受到光线变化的影响，而基于光度不变假设的直接法对光照变化更为敏感。

激光与相机的联合不仅能够保证 SLAM 应用于更大的场景，同时也能保证在遮挡或者视觉 SLAM 失效情况下，SLAM 系统能够继续完成定位与建图的功能，提高系统鲁棒性。

二、结合深度学习的 SLAM

到目前为止，SLAM 的方案都处 于特征点或者像素的层级。而关于这些特征点或像素的来源却无从得知。而随着深度学习在计算机视觉领域的成功，大家对深度学习在机器人领域的应用有很大兴趣，其中，深度学习与视觉 SLAM 的结合可分为三种：

其一是将深度学习用于帧间估计。相较于传统的基于特征法或直接法的帧间运动估计，基

于深度学习的方法无需提取特征、计算描述子，也无需特征匹配和复杂的几何计算，通过搭建端到端的深度神经网络架构可以快速提取图像序列的帧间运动。然而，不同学习算法之间的神经网络架构设计差异较大，对训练数据库依赖较强。基于深度神经网络的帧间估计仍处于起步阶段，有待进一步发展。

其二是与回环检测相结合。回环检测本质上是场景识别问题，不同于基于特征匹配的方式，通过神经网络学习图像的深层特征[21]，识别率更高，增加了回环检测的成功率。

其三是与语义 SLAM 结合。语义 SLAM 是指 SLAM 系统在建图过程中不仅获得环境中的几何结构信息，同时可以识别环境中独立个体的标签，以应对复杂场景及完成更智能的服务任务。

图 5-1　RGB-D 图像语义分割

图 5-2 语义 SLAM 结果

综合来说，SLAM 与语义的结合点有以下两个方面：一方面语义辅助 SLAM。传统的语义分割和物体识别往往只考虑一幅图片，而在 SLAM 中利用一台移动的相机，如果把语义分割应用到 SLAM 中，将得到一个带有标签的语义地图。另外，语义信息也可以为回环检测和

BA 优化提供更多的信息。另一方面 SLAM 辅助语义。物体识别和语义分割都需要大量的训练数据，并且需要人工从不同视角采集该物体的图片，输人分类器进行识别。利用 SLAM，可以自动地计算物体在图像中的位置，节省人力成本，并且能加快分类器的训练。

六 参考文献

[1] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.

[2]BAY H, ESS A, TUYTELAARS T, et al. Speeded-up robust features (SURF)[J]. Computer vision and image understanding, 2008, 110(3): 346-359.

[3] RUBLEE E, RABAUD V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF[C]//Proceedings of 2011 IEEE International Conference on Computer Vision. Barcelona, Spain, 2011: 2564-571.

[4] Alcantarilla P F. Fast explicit diffusion for accelerated features in nonlinear scale spaces[C]//British Machine Vision Conference. 2013(13): 1-11.

[5] Gálvez-López, D., & Tardos, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, *28*(5), 1188-1197.

[6] Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 1052-1067.

[7] Klein, G., & Murray, D. (2007, November). Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on* (pp. 225-234). IEEE.

[8] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, *31*(5), 1147-1163.

[9] Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, *33*(5), 1255-1262.

[10] Engel, J., Schöps, T., & Cremers, D. (2014, September). LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision* (pp. 834-849). Springer, Cham.

[11] Engel, J., Koltun, V., & Cremers, D. (2018). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, *40*(3), 611-625.

[12] Forster, C., Pizzoli, M., & Scaramuzza, D. (2014, May). SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*(pp. 15-22). IEEE.

[13] Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011, November). DTAM: Dense tracking and mapping in real-time. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2320-2327). IEEE.

[14] Endres, F., Hess, J., Sturm, J., Cremers, D., & Burgard, W. (2014). 3-D mapping with an RGB-D camera. IEEE Transactions on Robotics, 30(1), 177-187.

[15] Whelan, T. , Salas-Moreno, R. F. , Glocker, B. , Davison, A. J. , & Leutenegger, S. . (2016). Elasticfusion. International Journal of Robotics Research, 35(14), 1697-1716.

[16] Labbe, M., & Michaud, F. (2014, September). Online global loop closure detection for large-scale multi-session graph-based SLAM. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on (pp. 2661-2666). IEEE.

[17] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, *34*(3), 314-334.

[18] Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, *34*(4), 1004-1020.
[19] CALONDER M, LEPETIT V, STRECHA C, et al. BRIEF: binary robust independent elementary features[C]//Proceedings of the 11th European Conference on Computer Vision. Heraklion, Crete, Greece, 2010: 778-792.
[20]Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012, October). A benchmark for the evaluation of RGB-D SLAM systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (pp. 573-580). IEEE.
[21] Costante, G., Mancini, M., Valigi, P., & Ciarfuglia, T. A. (2016). Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation. *IEEE robotics and automation letters*, *1*(1), 18-25.
[22] Strasdat, H., Montiel, J. M., & Davison, A. J. (2012). Visual SLAM: why filter?. *Image and Vision Computing*, *30*(2), 65-77.
[23] 高翔, 张涛, & 刘毅. (2017). 视觉 SLAM 十四讲-从理论到实践.