

视觉 SLAM 笔记^{*}

易鹏

中山大学

内部版本号：V1.12.028 (内测版)

2022 年 9 月 28 日

^{*}本笔记已经开源，可以免费下载，github 地址：<https://github.com/Lovely-XPP/Notebook>
gitee 分流地址：https://gitee.com/sysu_xpp/Notebook

目录

第 1 章 SLAM 简介	1
1.1 SLAM 的目标	1
1.2 视觉 SLAM	1
1.2.1 SLAM 传感器	1
1.2.2 视觉 SLAM 的传感器	1
1.3 经典的视觉 SLAM 框架	2
1.3.1 视觉里程计	3
1.3.2 后端优化	3
1.3.3 回环检测	3
1.3.4 建图	3
符号说明	1
附录	5
a. 插图目录	5
b. 表格目录	5
c. 索引	9

第 1 章 SLAM 简介

1.1 SLAM 的目标

SLAM 的全称为 Simultaneous Localization and Mapping，中文翻译为即时定位与建图，所以 SLAM 的目标就是两个：

- (1) 我在什么地方？——**定位**
- (2) 周围环境怎么样？——**建图**

1.2 视觉 SLAM

1.2.1 SLAM 传感器

机器的定位有两类传感器。

- (1) **机器本身携带**：轮式编码器、相机、激光传感器、IMU 等
- (2) **安装于环境**：导轨、二维码标识等

由于环境中的传感器受限于环境的条件，而 GPS 在室内没有信号，SLAM 就是为了解决在任意未知环境中进行定位。这里主要讲视觉 SLAM 的工作。

1.2.2 视觉 SLAM 的传感器

视觉 SLAM 主要的传感器就是**相机**，照片本质上是场景在相机的成像平面上的**投影**。它以**二维**的形式记录了**三维**的世界，在这个过程中丢掉了场景的一个维度：**深度**（距离）。相机主要有三类：

(1) **单目相机**

- 组成：只使用一个摄像头。
- 原理：在单张图像中，无法确定一个物体的真实大小。它可能是一个**很大但很远**的物体，也可能是一个**很近但很小**的物体。所以如果要恢复三维结构，只能改变相机的视角。所以在单目 SLAM 中，我们必须移动相机才能估计它的**运动**（Motion），同时估计场景中物体的远近和大小，称为**结构**（Structure）。从图像的变化可以得到相机的相对运动状态，同时我们还知道：**近处的物体移动快，远处的物体移动慢，极远处（无穷远处）的物体（如太阳、月亮）看上去是不动的**。所以物体在图像上的运动就形成了**视差**（Disparity），通过视差就可以定量判断物体的距离远近。
- 缺点：由于单张图像无法确定深度，所以得到的物体远近仅仅是一个相对值。也就是说，如果把相机的运动和场景放大同样的倍数，单目相机得到的像是一样的。所以，单目 SLAM 估计的轨迹和地图将与真实的

轨迹和地图相差一个因子，即**尺度** (Scale)。由于单目 SLAM 无法仅仅通过图像确定真实尺度，所以又称为**尺度不确定性** (Scale Ambiguity)。

(2) 双目相机

- 目的：通过某种手段测量物体与相机之间的距离，克服单目相机无法知道距离的缺点。
- 组成：两个单目相机，且两个相机之间的距离（**基线**）已知。
- 原理：和人眼相似，通过左右眼图像的差异判断物体的远近，具体定量用基线确定。
- 缺点：需要大量计算才能近似（不太可靠）估计每一个像素点的深度，且深度的量程和精度受基线和分辨率所限，视差的计算需要消耗大量的计算资源。
- 应用：室内室外

(3) 深度相机

- 目的：通过某种手段测量物体与相机之间的距离，克服单目相机无法知道距离的缺点。
- 组成：深度相机又称 **RGB-D 相机**，由激光传感器等组成。
- 原理：通过红外结构光或 Time-of-Flight (ToF) 原理，像激光传感器那样，主动向物体发射并接受返回的光，测出物体与相机的距离，此为通过物理测量手段进行测量，可以节省大量的计算资源。
- 缺点：测量范围窄、噪声大、视野小、易受日光干扰、无法测量透射物质。
- 应用：室内

1.3 经典的视觉 SLAM 框架

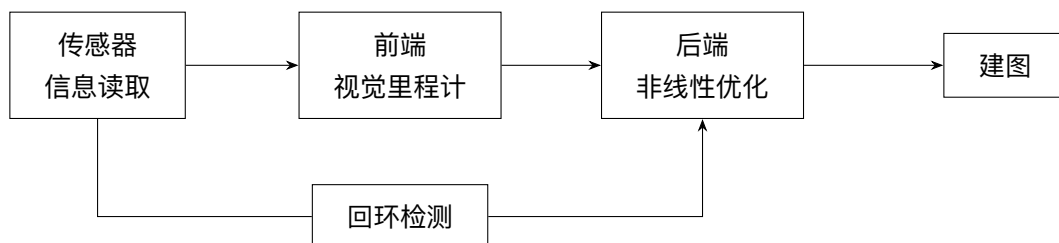


图 1.1: 经典的视觉 SLAM 框架

如图 1.1 所示，经典的视觉 SLAM 框架包括以下几个流程。

1. **传感器信息读取**：相机图像信息的读取和预处理。在机器人中，还可能包括码盘、惯性传感器等信息的读取和同步。
2. **前端视觉里程计** (Visual Odometry, **VO**)：估算相邻图像间相机运动，以及局部地图的样子。VO 又称为**前端** (Front End)。
3. **后端非线性优化** (Optimization)：接受不同时刻视觉里程计测量的相机位姿，以及回环检测的信息，对它们进行优化，得到全局一致的轨迹和地图。由于接在 VO 之后，又称为**后端** (Back End)。
4. **回环检测** (Loop Closure Detection)：回环检测判断机器人是否到达过先前的位置。如果检测到回环，它会把信息提供给后端进行处理。
5. **建图**：根据估计的轨迹，建立与任务要求对应的地图。

在静态、刚体、光照变化不明显、没有人为干扰的场景下，视觉 SLAM 技术已经相当成熟。

1.3.1 视觉里程计

视觉里程计是通过相邻帧间的图像估计相机运动，再通过相机与空间点的几何关系来估计当前时刻的相机运动，然后将相邻时刻的运动“串起来”，可以估计任意时刻的相机运动并恢复整个场景的空间结构。它称为“里程计”是因为它和实际的里程计一样，只计算相邻时刻的运动（不限于 2 帧，可以是 5 ~ 10 帧），而和过去的信息没有关联。

但是，由于视觉里程计在最简单的情况下只估计极少量图像间的运动，每个时刻的估计都产生一定的误差，导致先前的误差会传递到下一时刻，误差会随时间不断累积，这就是**累积漂移**（Accumulating Drift）。这将导致无法建立一致的地图。为了解决漂移问题，我们还需要后端优化和回环检测。回环检测负责吧“机器人回到原始位置”的事情检测出来，而后端优化则根据该信息，校正整个轨迹的形状。

1.3.2 后端优化

后端优化主要指助理 SLAM 过程中的**噪声**问题。任何传感器都会带一定的噪声，为此我们需要估计传感器带有多大的噪声，这些噪声是如何从上一时刻传递到下一时刻的，而我们又对当前的估计有多大的自信。后端优化要考虑的问题就是如何从这些带有噪声的数据中估计整个系统的状态，以及这个状态轨迹的不确定性有多大——这称为**最大后验概率估计**（Maximum-a-Posteriori, MAP）。这里的状态既包括机器人自身的轨迹，也包含地图。

前端给后端提供待优化的数据，后端只关心数据而不关心数据来自哪里。在视觉 SLAM 中，前端和计算机视觉研究更为相关，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算法。

后端：借助状态估计理论，把定位和建图的不确定性表达出来，然后采用滤波器或非线性优化，估计状态的均值和不确定性（方差）。

1.3.3 回环检测

回环检测又称闭环检测，主要解决**位置估计随时间漂移**的问题。这需要让机器人具有**识别到过的场景**的能力。视觉 SLAM 一般采用**判断图像间的相似性**的方法来完成回环检测。所以视觉回环检测实质上是一种计算图像数据相似性的算法。机器人判断回到原点后，将数据重新更新为当时不含累积漂移的数据，同时后端根据这些新的信息，把轨迹和地图调整到符合回环检测结果的样子。所以，如果我们又充分而且正确的回环检测，则可以消除累积误差，得到全局一致的轨迹和地图。

1.3.4 建图

建图是指构件**地图**的过程。地图是对环境的描述，但这个描述并不是固定的，需要视 SLAM 的应用而定。大体上讲，地图可以分为**度量地图**和**拓扑地图**两种。

(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，通常用稀疏（Sparse）与稠密（Dense）对其分类。

- **稀疏地图**：稀疏地图进行了一定程度的抽象，并不需要表达所有物体。例如，我们选择一部分具有代表性的东西，称之为**路标**（Landmark），那么一张稀疏地图就是由路标组成的地图，而不是路标的部分就可以忽略。

- **稠密地图**：稠密地图着重于建模所有看到的東西。稠密地图通常按照某种分辨率，由许多个小块组成，在二维度量地图中体现为许多个一般导航就是使用稠密地图。

相对地，

插图目录

1.1 经典的视觉 SLAM 框架 2

表格目录

索引

D

定位, 1

J

建图, 1

S

SLAM, 1