第1讲 预备知识

SLAM 是 Simultaneous Localization and Mapping 的缩写，中文译作“**同时定位与地图构建**”[1]。它是指搭载特定**传感器**的主体，在**没有环境先验信息**的情况下，**于运动过程中**建立**环境**的模型，同时估计自己的**运动**[2]。如果这里的传感器主要为相机，那就称为“视觉 SLAM”。

本书的主题就是视觉 SLAM。这里我们刻意把许多定义放到一句话中，希望帮助读者建立一个较明确的概念。首先，SLAM 的目的是解决“定位”与“地图构建”这两个问题。也就是说，一边要估计传感器自身的位置，一边要建立周围环境的模型。那么怎么解决呢？这需要用到传感器的信息。传感器以一定形式观察外部世界，但不同传感器观察的方式不同。之所以要花一本书的篇幅讨论这个问题，是因为它很难——特别是我们希望**实时地**、在**没有先验知识**的情况下进行 SLAM。当用相机作为传感器时，我们要做的就是根据一张张连续运动的图像（它们形成了一段视频），从中推断相机的运动，以及周围环境的情况。

第2讲

2.1

两个问题

视觉SLAM

单目相机

- 只有一个摄像头
- 以二维的形式记录了三维的世界，但丢失了一个维度，无法计算相机与物体之间的距离
- 尺度不确定性

双目相机

- 有两个摄像头
- 两个相机之间的距离(称为基线)是已知的，可以通过基线估计每个像素的空间位置
- 需要大量的计算才能(不太可靠地)估计每一个像素的深度

- 基线距离越大，能测量到的物体越远
- 双目或多目相机的缺点是配置与标定较为复杂，其深度量程和精度受双目的基线和分辨率所限，目前计算是主要问题

深度(RGB-D)相机

- 多个摄像头、能采集到彩色图片、还可以读出每个像素与摄像头之间的距离
- 最大的特点是可以通过红外结构光或Time-of-Flight原理，像激光传感器那样，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。它并不像双目相机那样通过软件计算来解决，而是通过物理的测量手段，所以相比于双目相机可节省大量的计算资源
- 不过，现在多数RGB-D相机还存在测量范围窄、噪声大
  视野小、易受日光干扰、无法测量透射材质等诸多问题，在SLAM方面，主要用于室内，室外则较难应用。

2.2 经典视觉SLAM框架

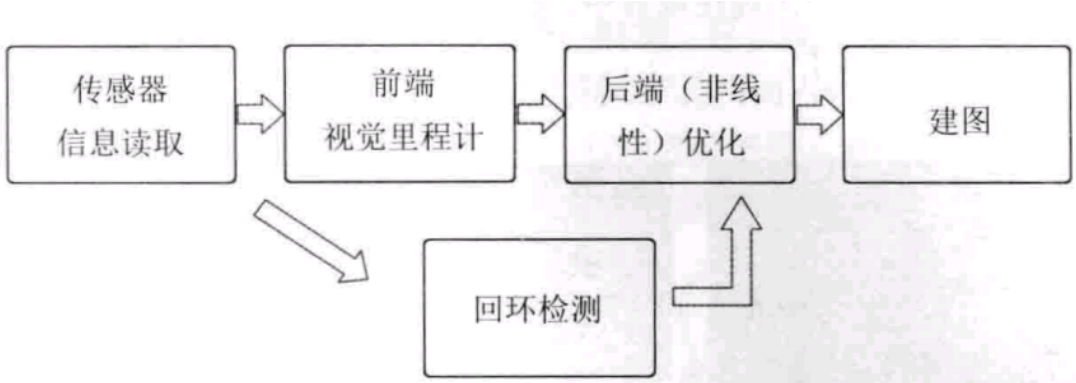

图 2-7　经典的视觉 SLAM 框架

整个视觉SLAM流程包括以下步骤:

1. 传感器信息读取。在视觉SLAM中主要为相机图像信息的读取和预处理。如果是在机器人中，还可能有码盘、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）。视觉里程计的任务是估算相邻图像间相机的运动，以及局部地图的样子。VO又称为前端（FrontEnd）。
3. 后端（非线性）优化（Optimization）。后端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（BackEnd）。

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（Mapping）。它根据估计的轨迹，建立与任务要求对应的地图。

2.2.1 视觉里程计

视觉里程计关心相邻图像之间的相机运动.

视觉里程计能够通过相邻帧间的图像估计相机运动，并恢复场景的空间结构。称它为“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，而和过去的信息没有关联。在这一点上，视觉里程计就像一种只有短时记忆的物种（不过可以不限于两帧，数量可以更多一些，例如5~10帧）。

现在，假定我们已有了一个视觉里程计，估计了两张图像间的相机运动。那么，一方面，只要把相邻时刻的运动“串”起来，就构成了机器人的运动轨迹，从而解决了定位问题。另一方面，我们根据每个时刻的相机位置，计算出各像素对应的空间点的位置，就得到了地图。这么说来，有了视觉里程计，是不是就解决了SLAM问题呢？

然而，仅通过视觉里程计来估计轨迹，将不可避免地出现累积漂移（Accumulating Drift）。这是由于视觉里程计在最简单的情况下只估计两个图像间的运动造成的。我们知道，每次估计都带有一定的误差，而由于里程计的工作方式，先前时刻的误差将会传递到下一时刻，导致经过一段时间之后，估计的轨迹将不再准确。

这也就是所谓的漂移（Drift）。它将导致我们无法建立一致的地图。为了解决漂移问题，我们还需要两种技术：后端优化和回环检测。回环检测负责把“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.

2.2.2 后端优化

笼统地说，后端优化主要指处理SLAM过程中的噪声问题。除了解决“如何从图像估计出相机运动”，我们还要关心这个估计带有多大的噪声，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，以及这个状态估计的不确定性有多大一—这称为最大后验概率估计(Maximum-a-Posterior i，MAP)这里的状态既包括机器人自身的轨迹，也包含地图

相对地，视觉里程计部分有时被称为“前端”。在SLAM框架中，前端给后端提供待优化的数据，以及这些数据的初始值。而后端负责整体的优化过程，它往往面对的只有数据，不必关心这些数据到底来自什么传感器。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。

SLAM问题的本质：对运动主体自身和周围环境空间不确定性的估计。为了解决SLAM问题，我们需要状态估计理论，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）。

2.2.3 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。怎么解决呢？假设实际情况下机器人经过一段时间的运动后回到了原点，但是由于漂移，它的位置估计值却没有回到原点。怎么办呢？如果有某种手段，让机器人知道“回到了原点”这件事，或者把“原点”识别出来，我们再把位置估计值“拉”过去，就可以消除漂移了。这就是所谓的回环检测

回环检测与“定位”和“建图”二者都有密切的关系。事实上，我们认为，地图存在的主要意义是让机器人知晓自己到过的地方。为了实现回环检测，我们需要让机器人具有识别到过的场景的能力。例如，可以判断图像间的相似性来完成回环检测。这一点和人是相似的。当我们看到两张相似的图片时，容易辨认它们来自同一个地方如果回环检测成功，则可以显著地减小累积误差。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。由于图像的信息非常丰富，使得正确检测回环的难度降低了不少

在检测到回环之后，我们会把“A与B是同一个点”这样的信息告诉后端优化算法。然后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。这样，如果我们有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。

2.2.4 建图

建图是指构建地图的过程。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。

地图的形式随SLAM的应用场合而定。大体上讲，可以分为度量地图和拓扑地图两种。

度量地图（Metric Map）

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。

- 稀疏地图进行了一定程度的抽象，并不需要表达所有的物体。例如，我们选择一部分具有代表意义的东西，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。相对地，稠密地图着重干建模所有看到的东西，定位时用稀疏路标地图就足够了。

- 而用于导航时，则往往需要稠密地图（否则撞上两个路标之间的墙怎么办？）。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小格子（Grid）,而在三维度量地图中则体现为许多小方块（Voxel）。通常，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，如A*、D*等，为机器人研究者所重视。但是我们也看到，一方面，这种地图需要存储每一个格点的状态，会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。另一方面，大规模度量地图有时会出现一致性问题。很小的一点转向误差，可能会导致两间屋子的墙出现重叠，使地图失效

拓扑地图（Topological Map）

相比于度量地图的精确性，拓扑地图更强调地图元素之间的关系。拓扑地图是一个图（Graph），由节点和边组成，只考虑节点间的连通性，例如只关注A、B点是连通的，而不考虑如何从A点到达B点。

它放松了地图对精确位置的需要，去掉了地图的细节，是一种更为紧凑的表达方式.然而，拓扑地图不擅长表达具有复杂结构的地图。如何对地图进行分割，形成节点与边，又如何使用拓扑地图进行导航与路径规划，仍是有待研究的问题.

2.3 SLAM问题的数学表述

运动

自身各个时刻的位置记为x_1，x_2,……，x_K。

运动方程:x_k = f(x_{k-1},u_k,w_k)

(u_k是运动传感器的读数或输入，,w_k是该过程中加入的噪声。)

观测

在x_k位置上看到一个路标点y_j，产生一个观测数据$z_{k,j}$

观测方程:$z_{k,j}$ = h(y_j,x_k,$v_{k,j}$)

($v_{k,j}$是这次观测里的噪声)

参数化

$x_k = \begin{bmatrix} x_1 & x_2 & \theta \end{bmatrix}_k^T$

$u_k = \begin{bmatrix} \Delta x_1 & \Delta x_2 & \Delta\theta \end{bmatrix}_k^T$

运动方程可以具体化为:

$$\begin{bmatrix} x_1 \\ x_2 \\ \theta \end{bmatrix}_k = \begin{bmatrix} x_1 \\ x_2 \\ \theta \end{bmatrix}_{k-1} + \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta\theta \end{bmatrix}_k + \boldsymbol{w}_k.$$

激光传感器观测到一个2D路标点时,能测到两个量，路标点与本体之间的距离r和夹角ϕ

记路标点 $y_j = \begin{bmatrix} y_1 & y_2 \end{bmatrix}_j^T$

位姿 $x_k = \begin{bmatrix} x_1 & x_2 \end{bmatrix}_k^T$

观测数据$z_{k,j} = \begin{bmatrix} r_{k,j} & \phi_{k,j} \end{bmatrix}_j^T$

观测方程可写为:

$$\begin{bmatrix} r_{k,j} \\ \phi_{k,j} \end{bmatrix} = \begin{bmatrix} \sqrt{(y_{1,j} - x_{1,k})^2 + (y_{2,j} - x_{2,k})^2} \\ \arctan\left(\dfrac{y_{2,j} - x_{2,k}}{y_{1,j} - x_{1,k}}\right) \end{bmatrix} + \boldsymbol{v}.$$

SLAM过程可总结为两个基本方程:

$$\begin{cases} \boldsymbol{x}_k = f\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k\right), & k = 1, \cdots, K \\ \boldsymbol{z}_{k,j} = h\left(\boldsymbol{y}_j, \boldsymbol{x}_k, \boldsymbol{v}_{k,j}\right), & (k,j) \in \mathcal{O} \end{cases}.$$