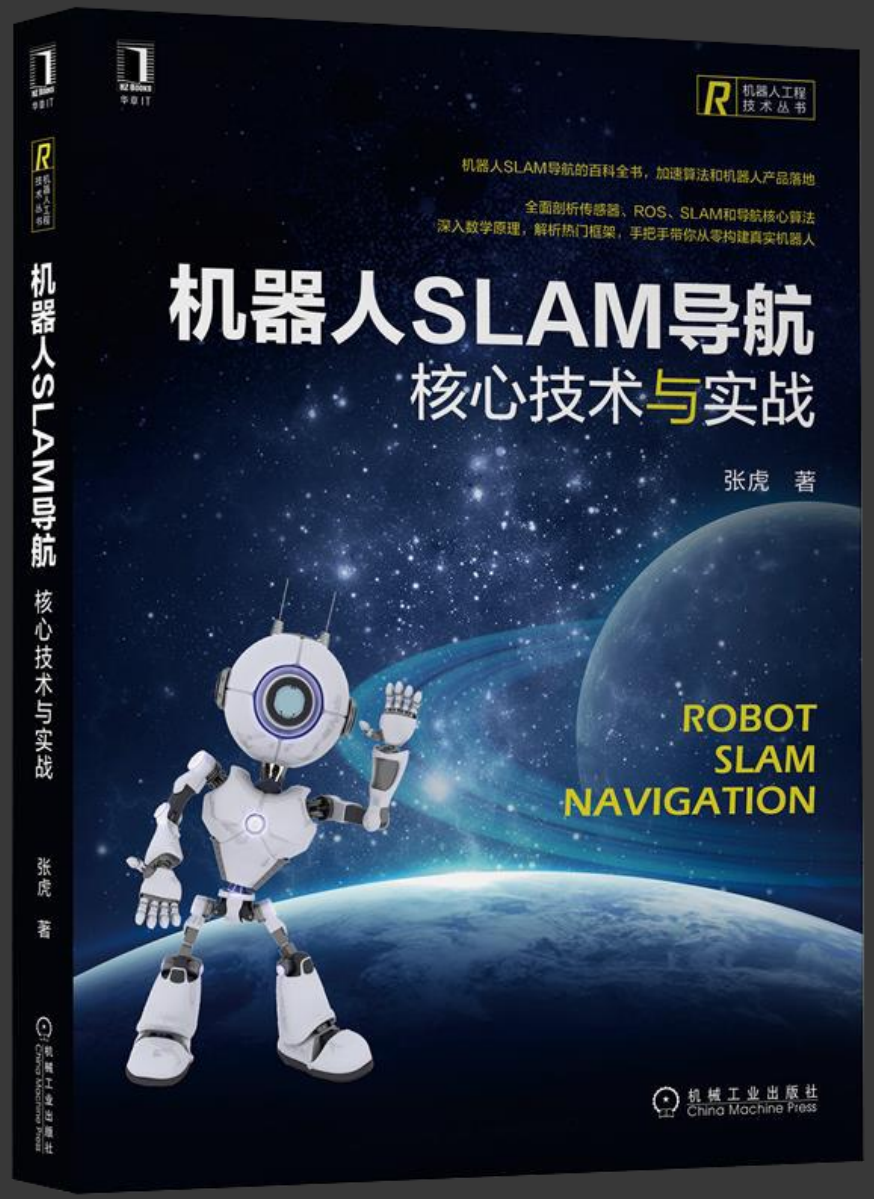


第1季

第11章：自主导航中的数学基础

主讲人：张虎

（小虎哥哥爱学习）

- 先导课
- 第1季：快速梳理知识要点与学习方法 √
- 第2季：详细推导数学公式与代码解析
- 第3季：代码实操以及真实机器人调试
- 答疑课

------（永久免费●系列课程●长期更新）------

本书内容安排

一、编程基础篇

第1章：ROS入门必备知识

第2章：C++编程范式

第3章：OpenCV图像处理

二、硬件基础篇

第4章：机器人传感器

第5章：机器人主机

第6章：机器人底盘

三、SLAM篇

第7章：SLAM中的数学基础

第8章：激光SLAM系统

第9章：视觉SLAM系统

第10章：其他SLAM系统

四、自主导航篇

第11章：自主导航中的数学基础

第12章：典型自主导航系统

第13章：机器人SLAM导航综合实战

自主导航是什么？

讨论范围：**机器人**

① 不受外界指令干预

② 自动与环境发生交互

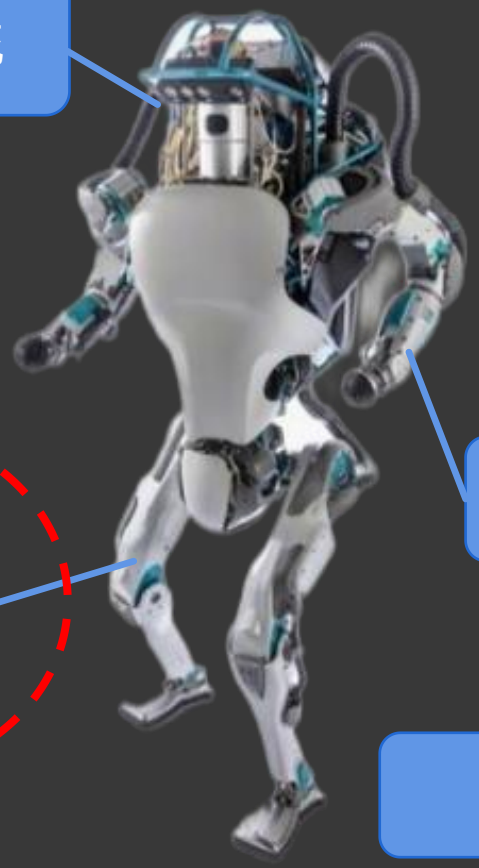
③ 完成特定的任务

导引（智能） + 航行（运动）

- 环境感知
- 路径规划
- 运动控制

- 机械执行

移动（轮式机器人/无人驾驶汽车）
行走（腿足机器人/机器狗）
飞行（无人机/宇宙飞船/星际探测器）
航行（无人船）
潜航（无人潜水艇）



- 自主语言交流
- 自主移动
- 搬运物品
- …

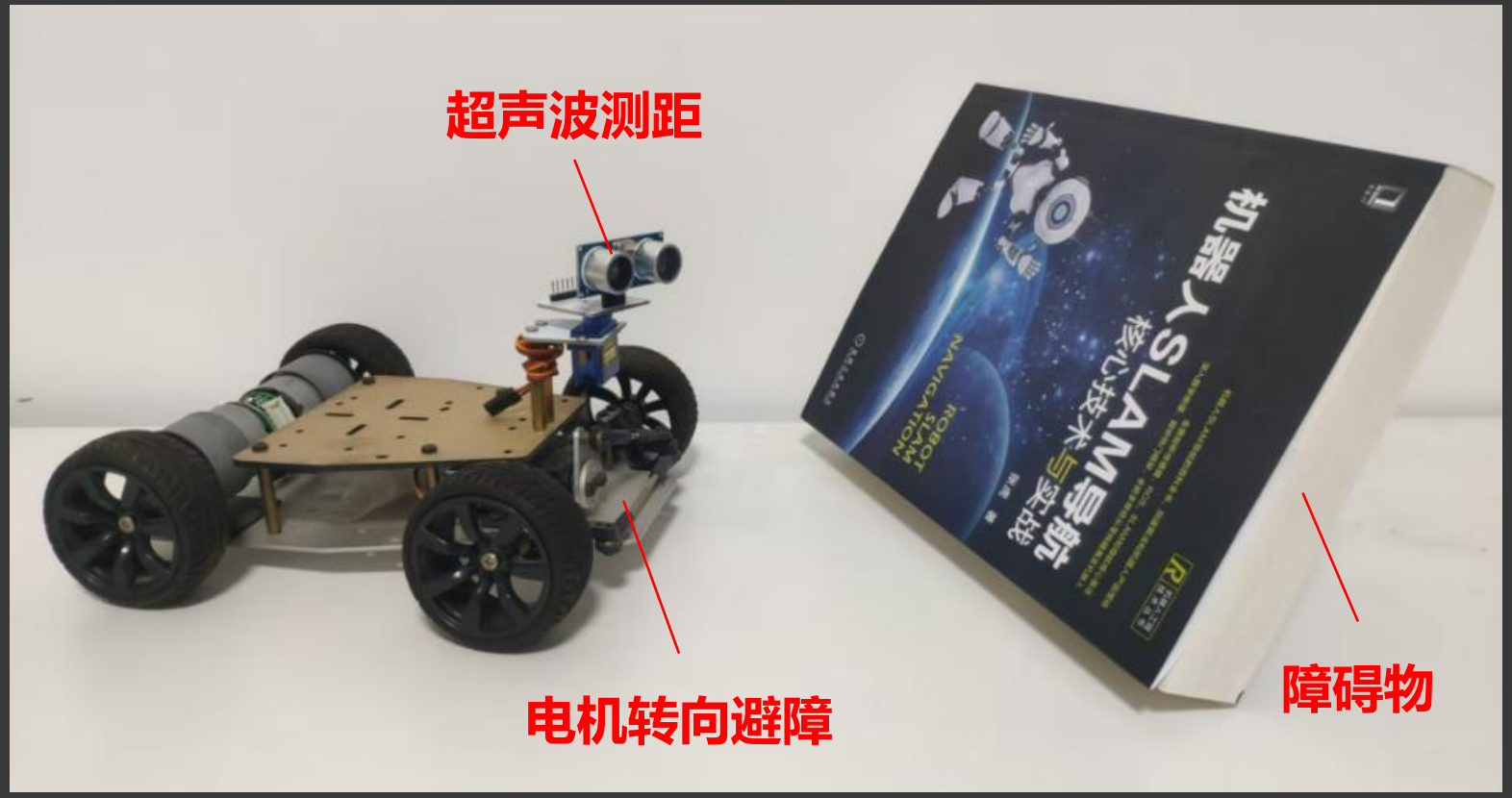
不能自主导航，机器人将失去80%的功能

然而，自主导航非常难？

自主导航，为什么很难？

刚入行的小白：

自主导航很简单呀，我在学校参加科技比赛，做的超声波避障小车还拿过奖了！

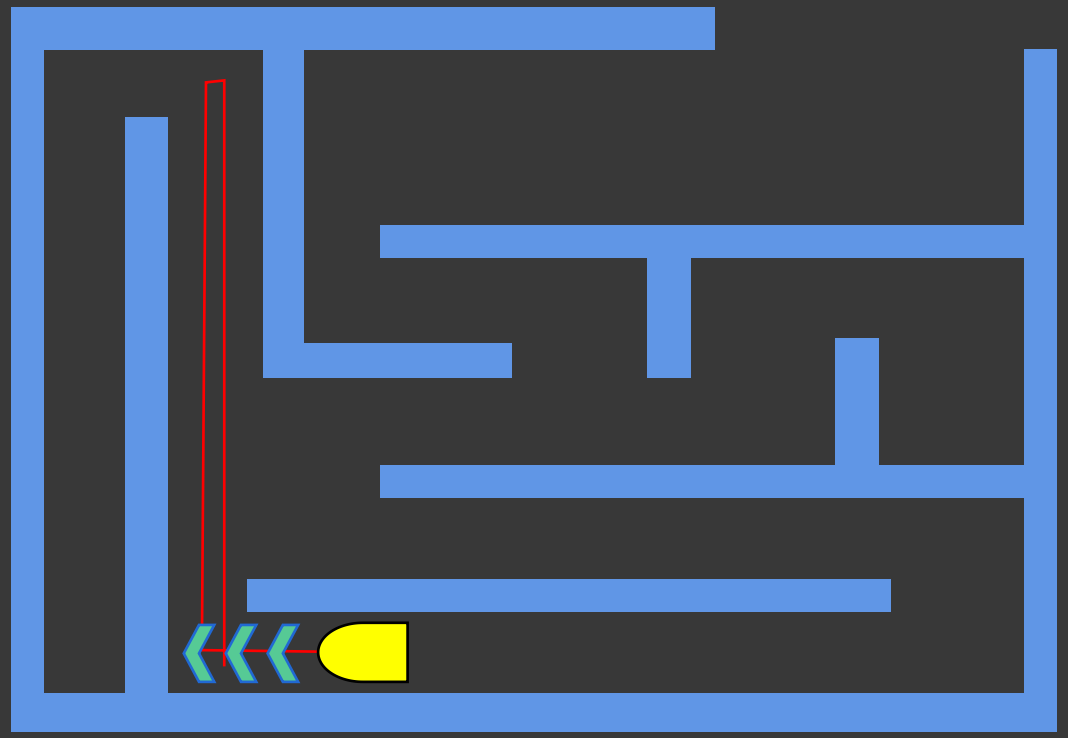



```
while(1)
{
        get(sonar);

        if ( sonar < 0.1 )
        {
                car_stop();
                delay(1);

                car_turn_right();
                delay(1);
        }
        else
        {
                car_go();
        }
}
```

自主导航，为什么很难？



小车实际运动，会乱跑

局部避障（短期决策）

全局避障（长期决策）

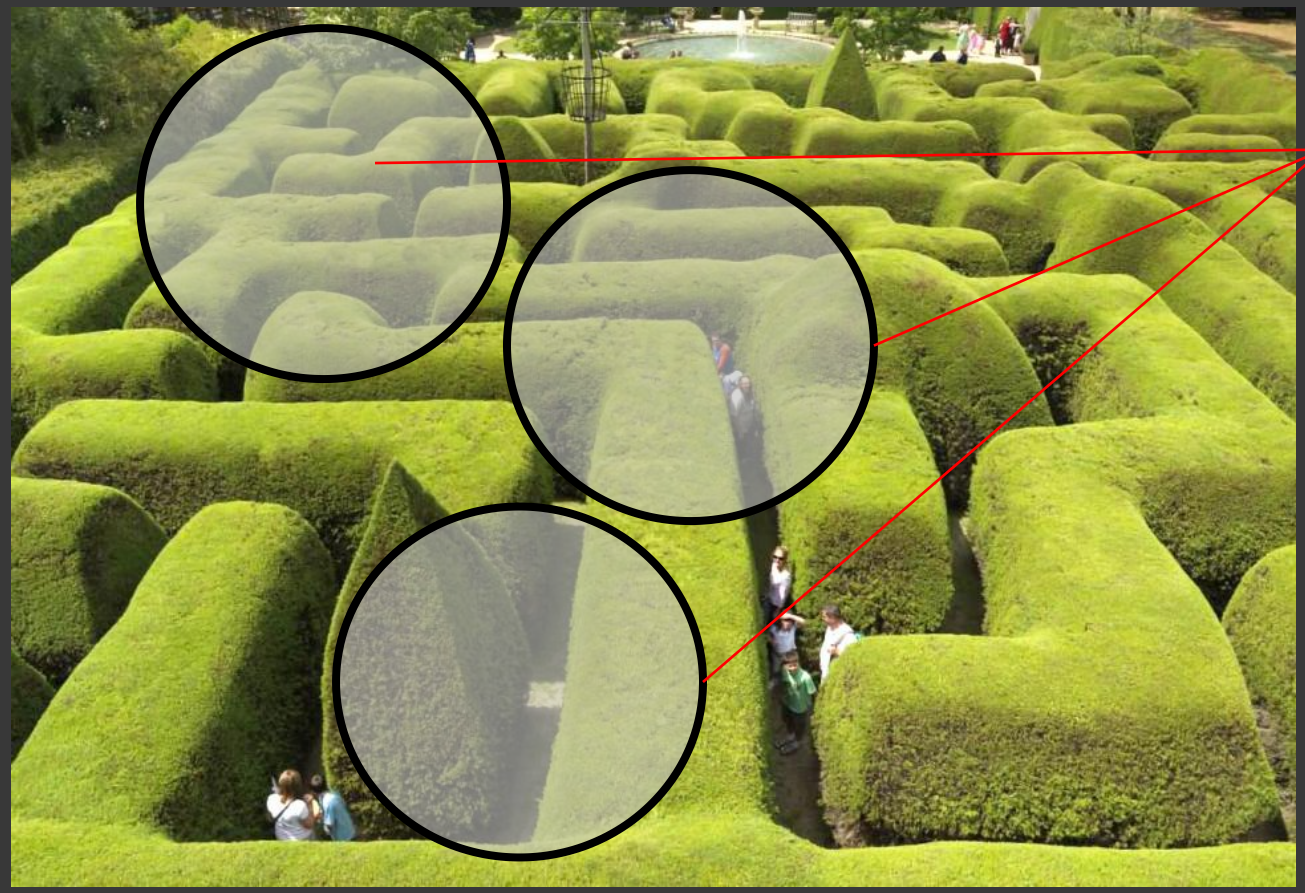
有大聪明会说：

是小车太笨了

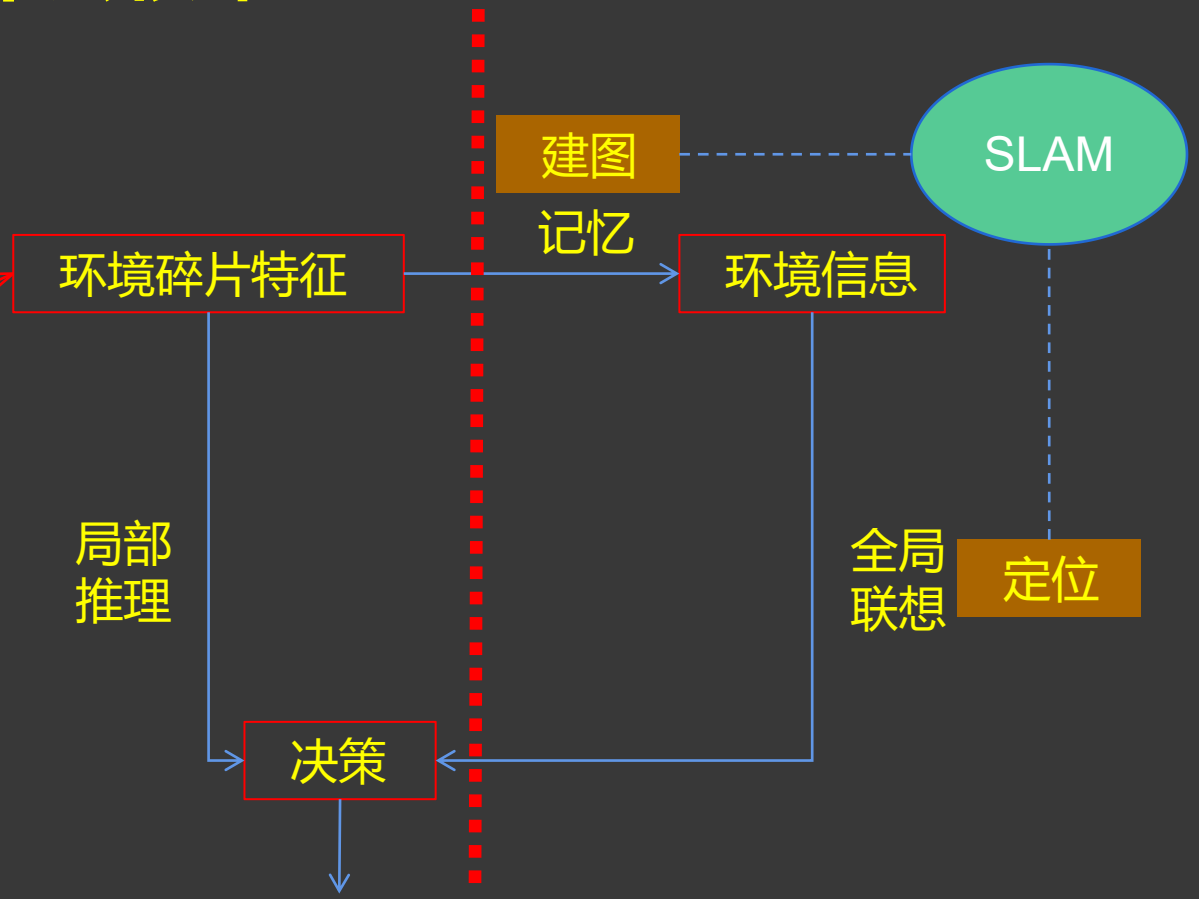
你写的程序太不智能了

没有高级点的算法吗

自主导航，为什么很难？



真人迷宫

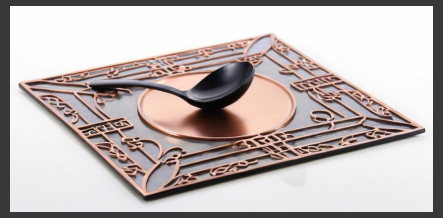


没有SLAM，能自主导航吗？

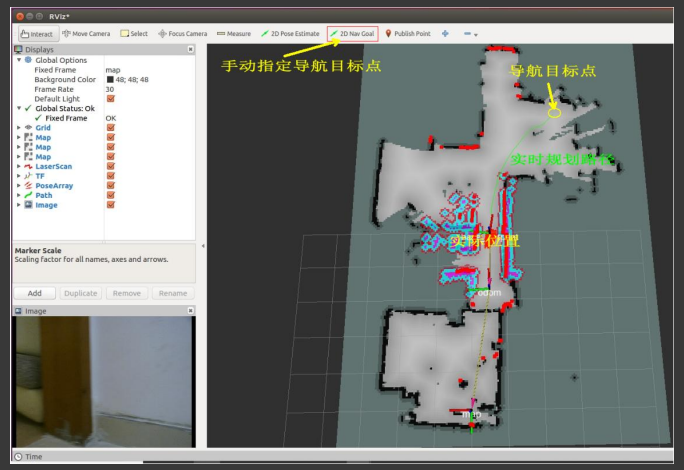
内容概要

11.1 自主导航

- 导航发展简介
- 自主导航问题的本质
- 自主导航工程化体系结构



■ 指南针在航海中的应用



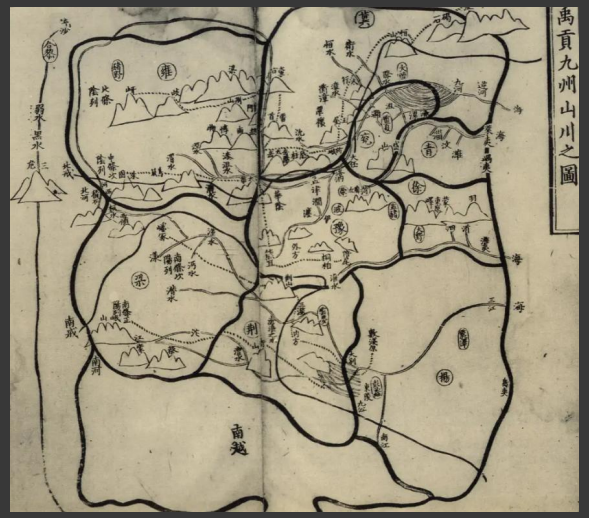
■ SLAM导航

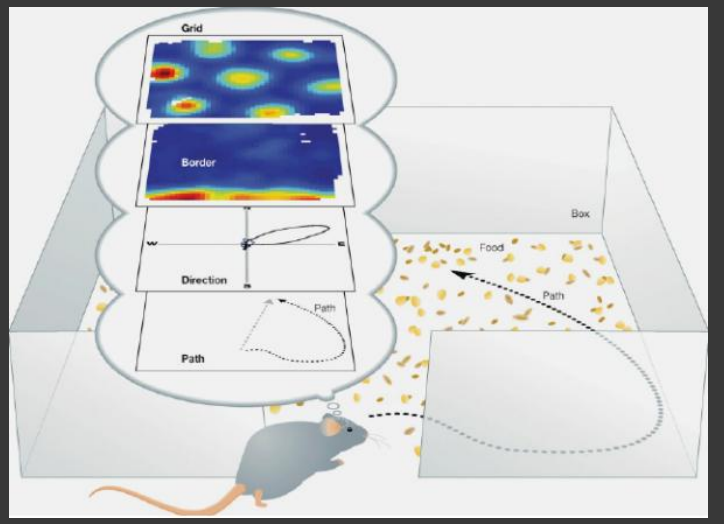
■ 古代手绘军事地图

■ 全球卫星导航

■ 仿生导航

地图
定位
控制
...





11.1 自主导航

- 导航发展简介
- 自主导航问题的本质
- 自主导航工程化体系结构

哺乳动物如何导航

（海马-内嗅皮层）

2014 年诺贝尔生理学或医学奖得主

11.1 自主导航

- 导航发展简介
- 自主导航问题的本质
- 自主导航工程化体系结构

哺乳动物如何导航

（海马-内嗅皮层）

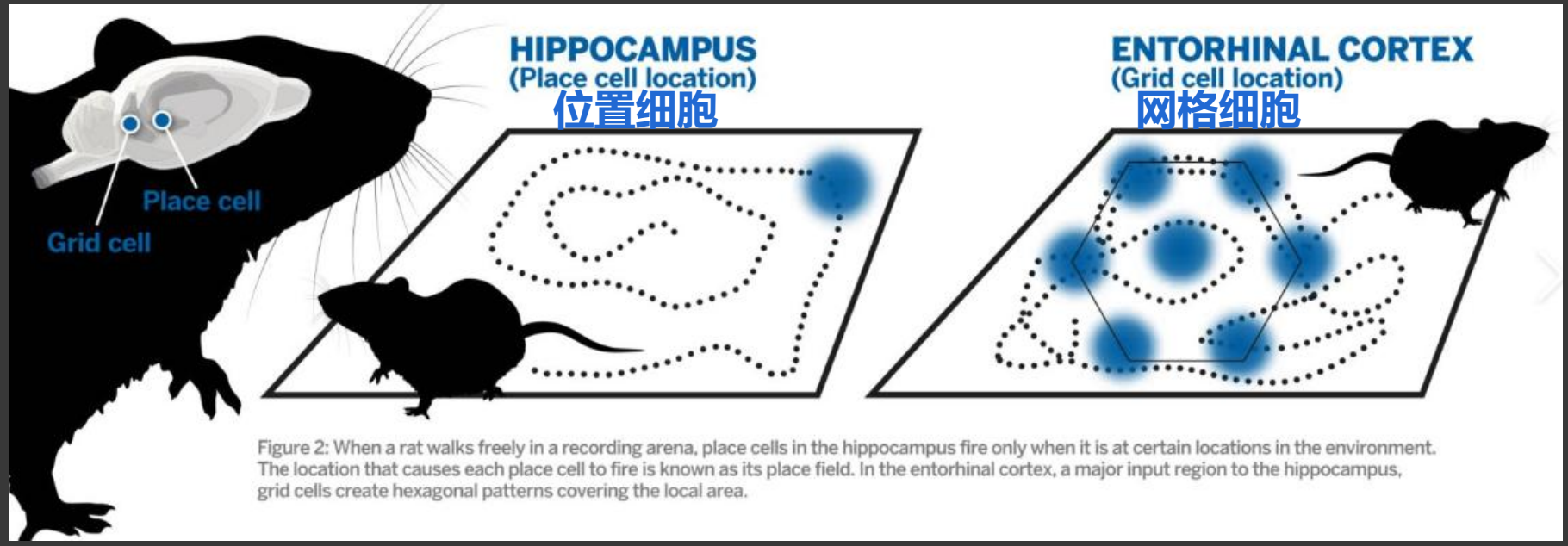
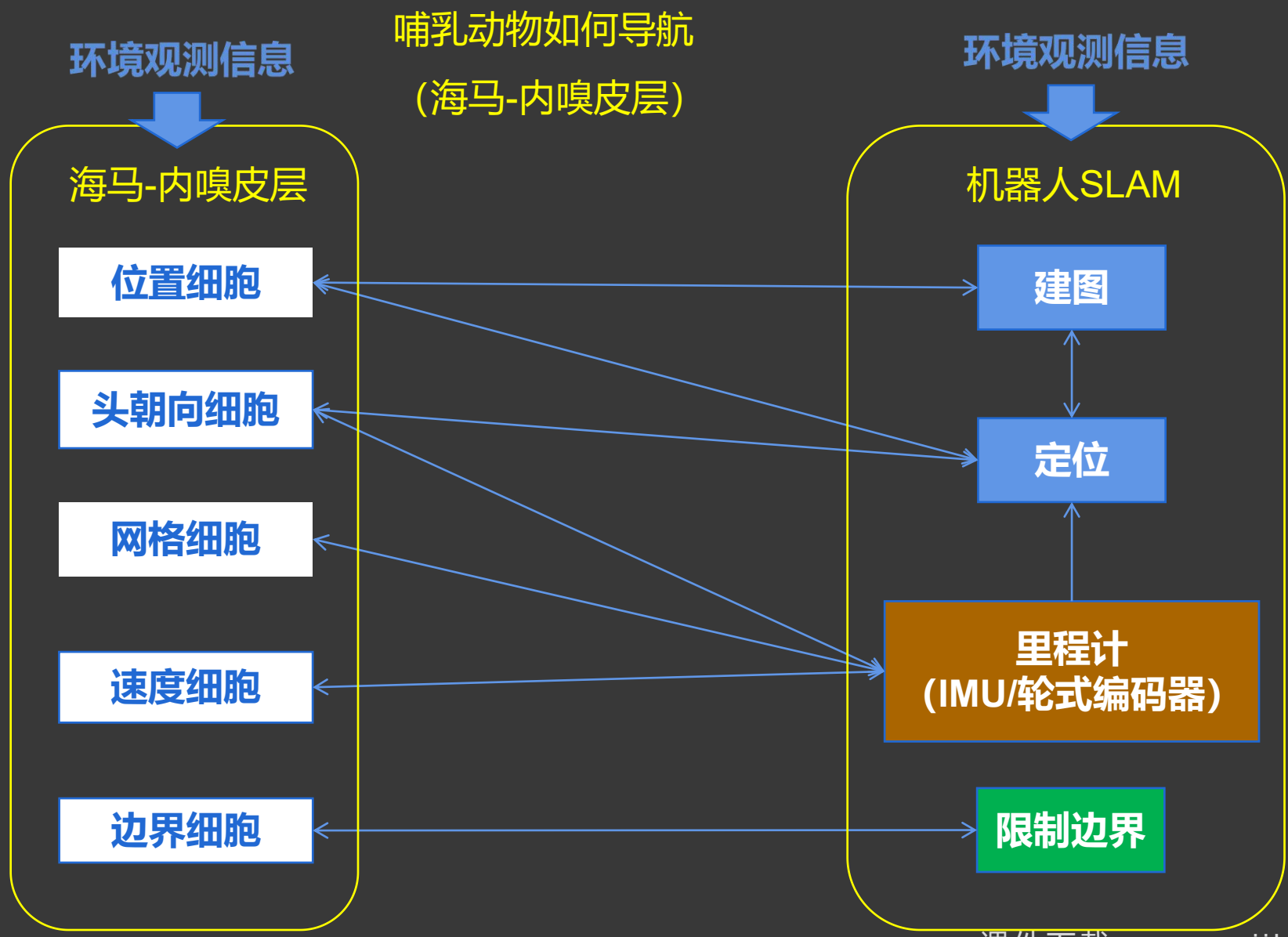


Figure 2: When a rat walks freely in a recording arena, place cells in the hippocampus fire only when it is at certain locations in the environment. The location that causes each place cell to fire is known as its place field. In the entorhinal cortex, a major input region to the hippocampus, grid cells create hexagonal patterns covering the local area.

头朝向细胞 **速度细胞** **边界细胞**

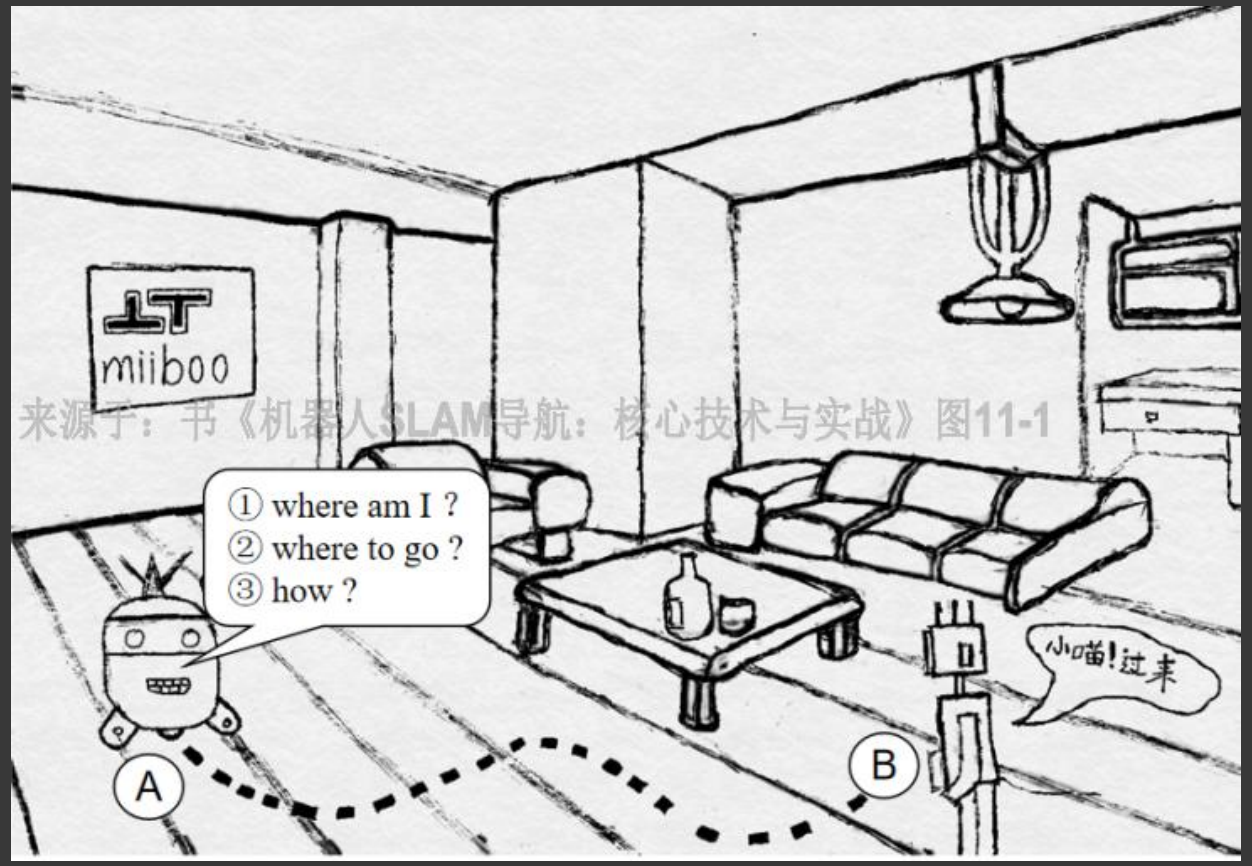
11.1 自主导航

- 导航发展简介
- 自主导航问题的本质
- 自主导航工程化体系结构



11.1 自主导航

- 导航发展简介
- 自主导航问题的本质
- 自主导航工程化体系结构

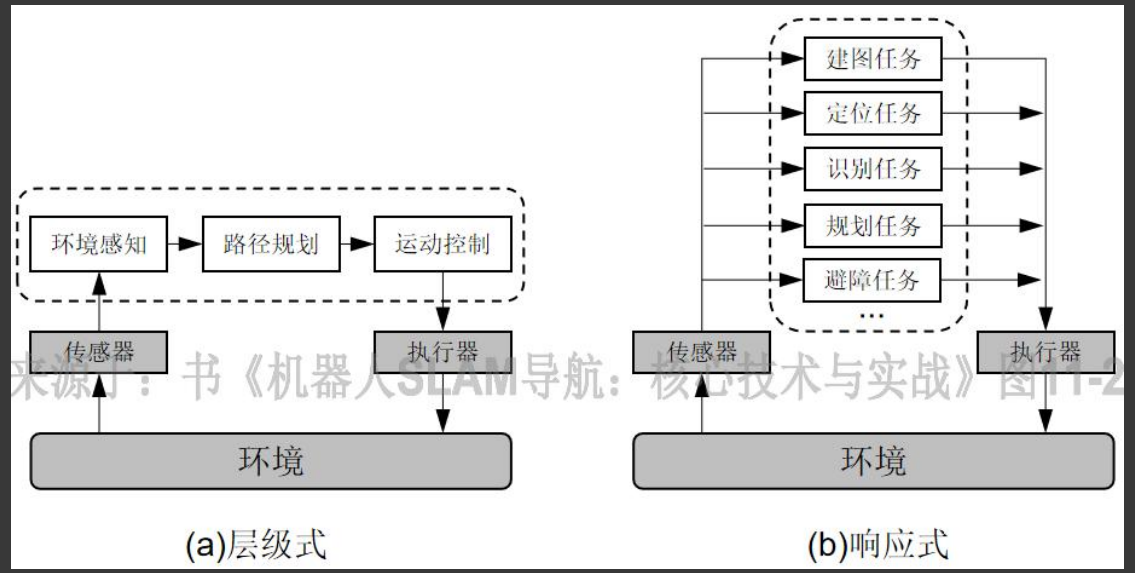


来源于：书《机器人SLAM导航：核心技术与实战》图11-1

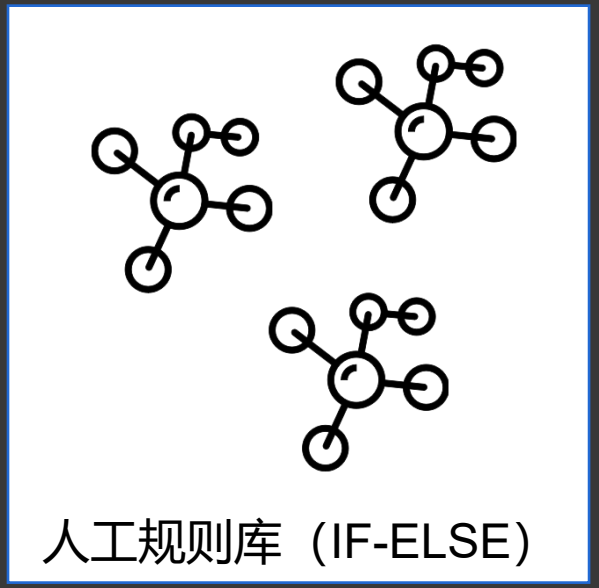
- 我在哪?
- 我将到何处去?
- 我该如何去?

11.1 自主导航

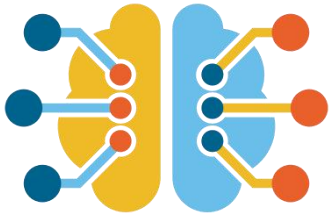
- 导航发展简介
- 自主导航问题的本质
- 自主导航工程化体系结构



来源：书《机器人SLAM导航：核心技术与实战》图11-2



人工规则库（IF-ELSE）

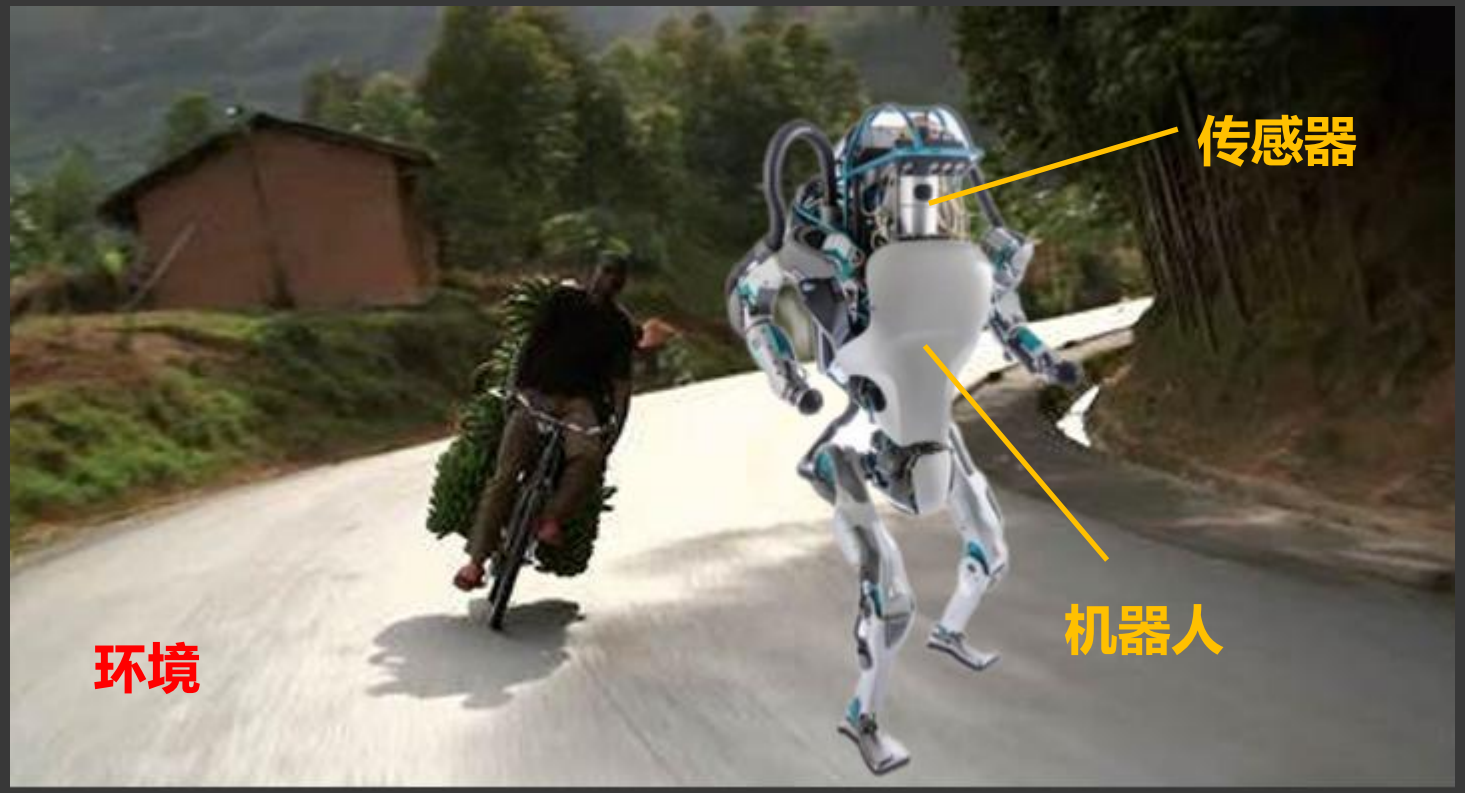


人工神经网络端到端

内容概要

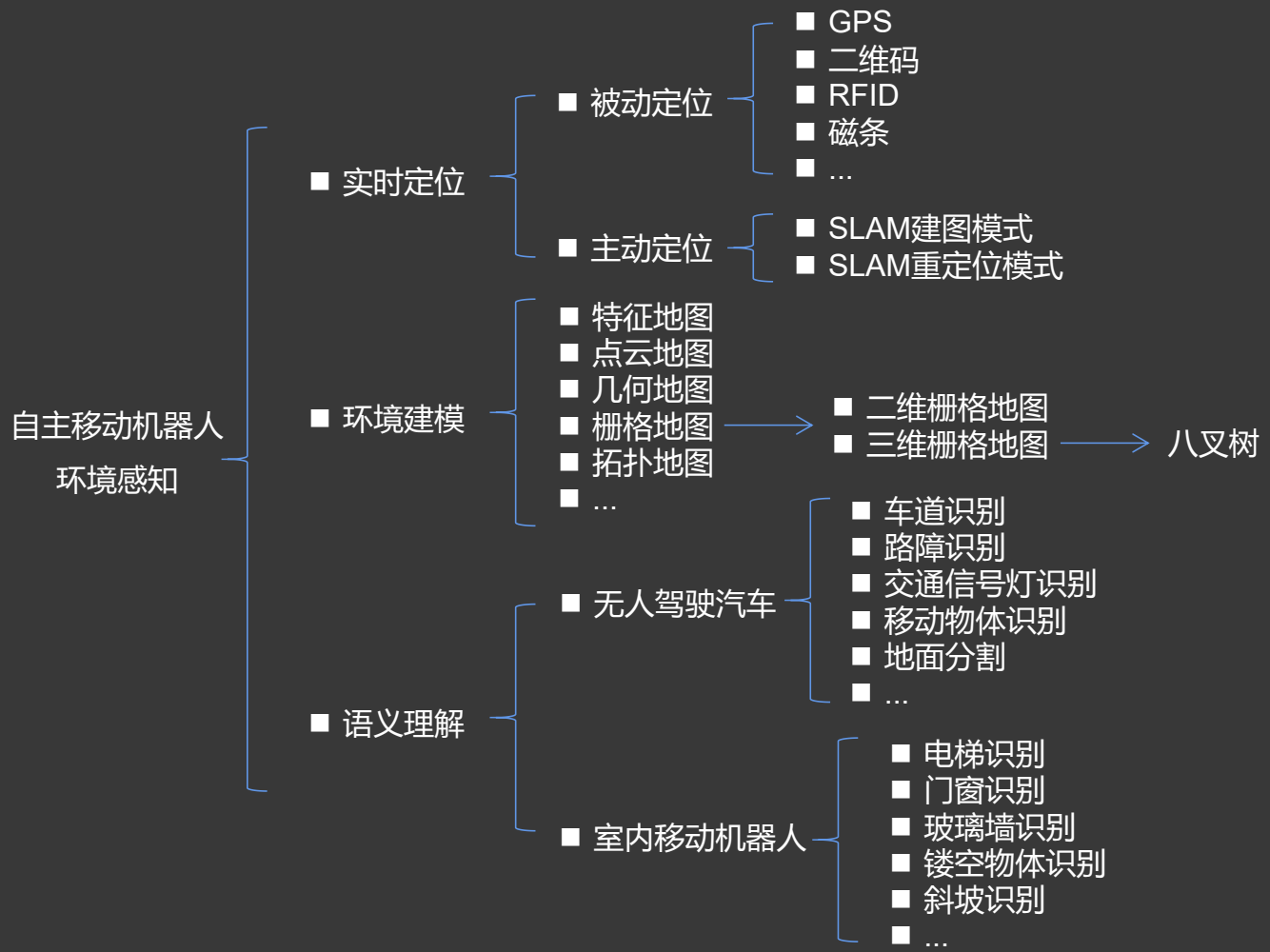
11.2 环境感知

环境感知就是**机器人**利用**传感器**获取**自身及环境状态信息**的过程，自主导航机器人的环境感知主要包括实时定位、环境建模、语义理解等。



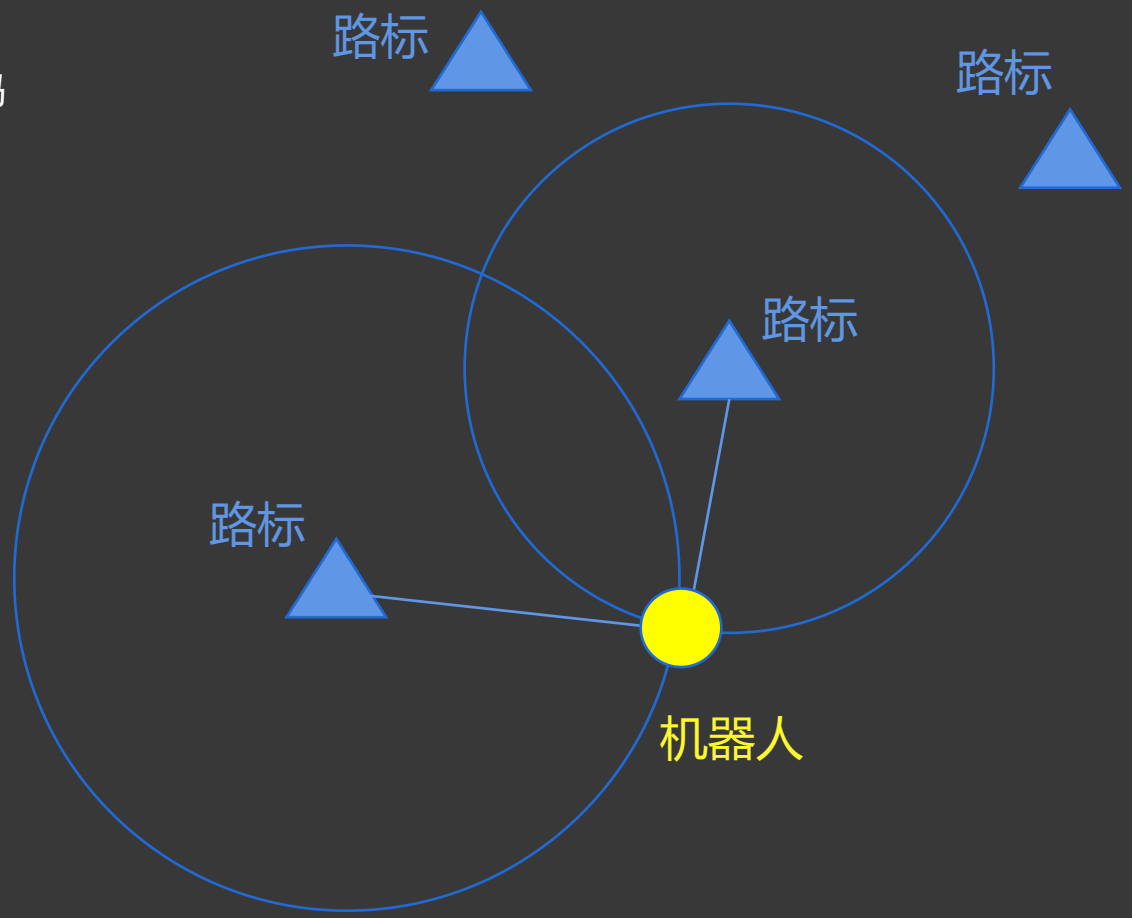
11.2 环境感知

环境感知就是**机器人**利用**传感器**获取**自身及环境状态信息**的过程，自主导航机器人的环境感知主要包括实时定位、环境建模、语义理解等。



11.2 环境感知

- 实时定位
  - 被动定位
    - GPS
    - 二维码
    - RFID
    - 磁条
    - ...
  - 主动定位
- 环境建模
- 语义理解



11.2 环境感知

- 实时定位 → 被动定位、主动定位 → SLAM建图模式、SLAM重定位模式
- 环境建模
- 语义理解

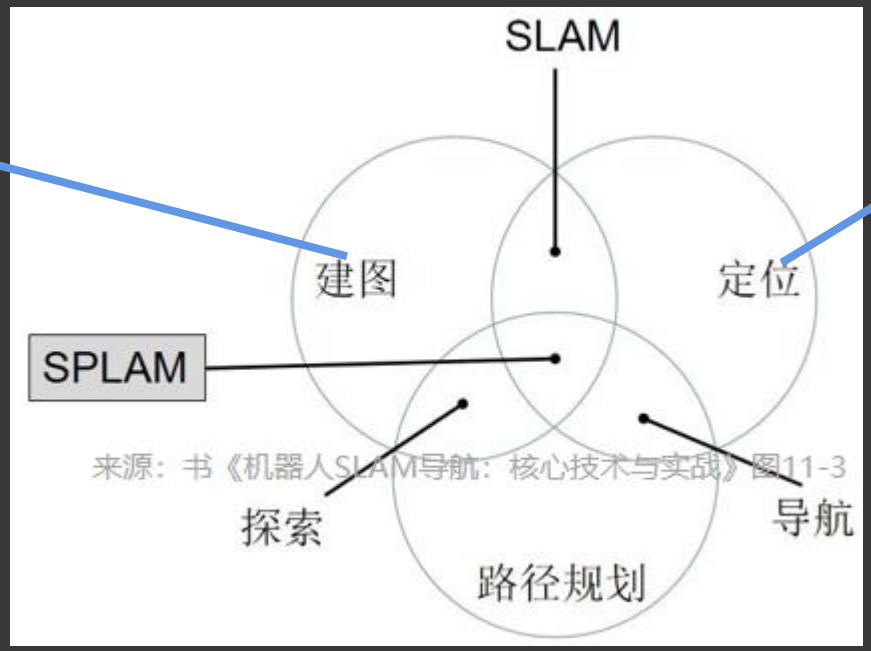
被动定位搭建路标基站成本高，且很多场景不具备搭建条件。

➢ 可解释地图

栅格地图、点云地图、拓扑地图

➢ 不可解释地图

高维数据库、神经网络压缩感知



来源：书《机器人SLAM导航：核心技术与实战》图11-3

➢ 可解释定位

三维正交空间坐标、极坐标、相对位移坐标、模糊关系坐标

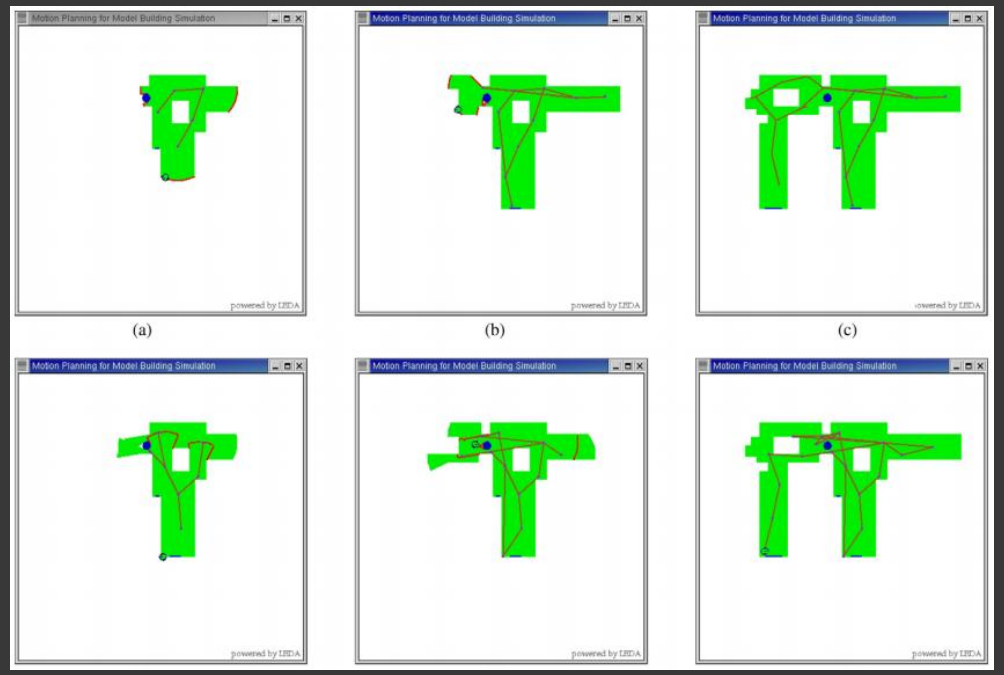
➢ 不可解释定位

非线性映射、端到端定位

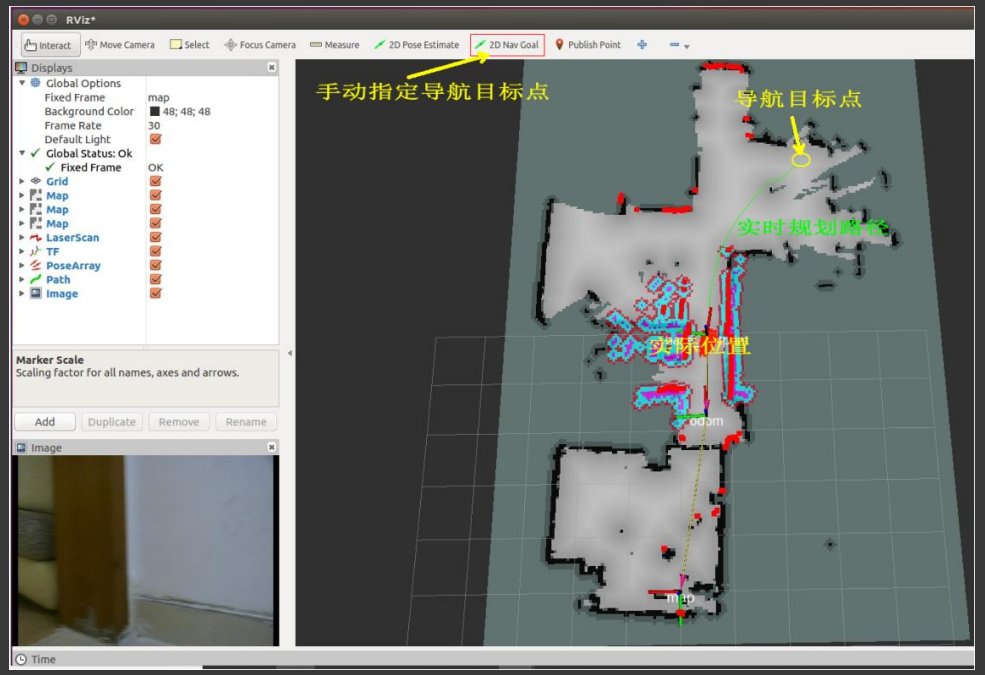
有很多人说现在的自动驾驶已经不用SLAM（比如高精地图定位、端到端），这样的观点对吗？

11.2 环境感知

- 实时定位
  - 被动定位
  - 主动定位
    - SLAM建图模式
    - SLAM重定位模式
- 环境建模
- 语义理解



① SLAM建图模式（地图未知，探索建图）



② SLAM重定位模式（地图已知）

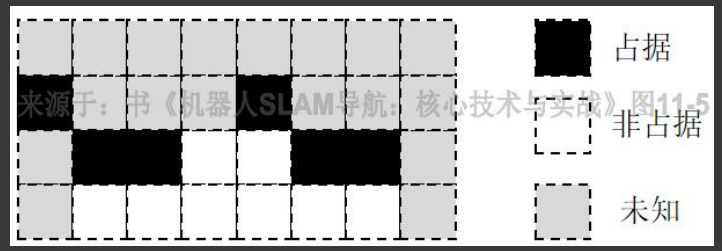
11.2 环境感知

- 实时定位
- 环境建模
- 语义理解

- 环境建模其实就是在对环境状态进行描述，也就是构建环境地图。
- 地图的一方面可以用于定位，另一方面可以用于避障，因此**定位用到的地图**与**避障的地图**并不一定相同。
- 环境地图的表示方法有很多种，比如特征地图、点云地图、几何地图、栅格地图、拓扑地图等。
- **视觉SLAM**通常以构建特征地图和点云地图为主，而**激光SLAM**则以构建栅格地图为主。
- 由于导航过程中需要避开障碍物，所以特征地图或点云地图必须被转换成栅格地图后才能导航，下面主要讨论一下二维栅格地图和三维栅格地图。

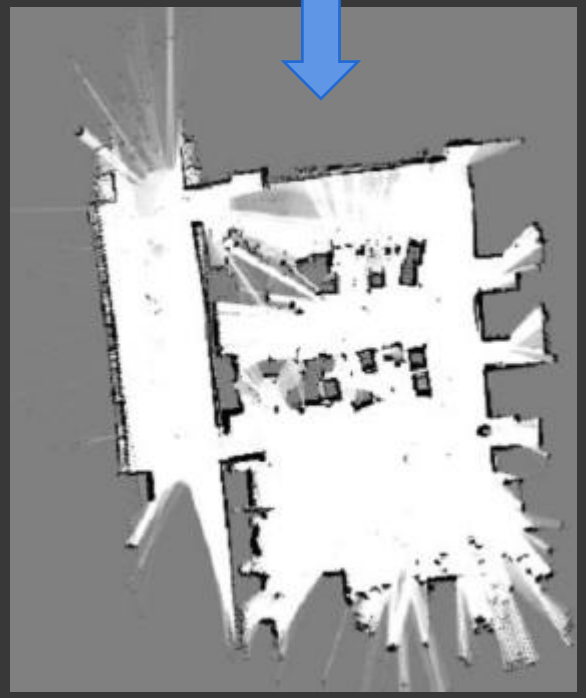
11.2 环境感知

- 实时定位
- 环境建模 → 二维栅格地图、三维栅格地图
- 语义理解

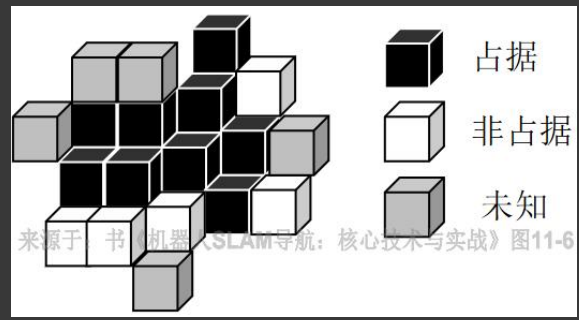


二维栅格地图

描述

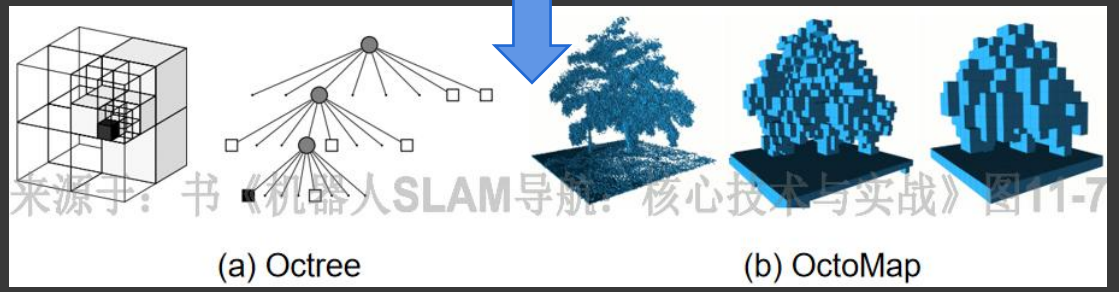


灰度图像



三维栅格地图

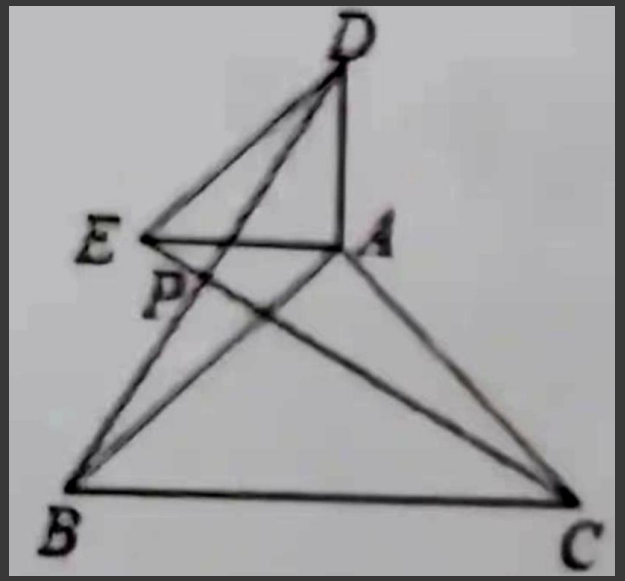
描述



八叉树与八叉树地图

11.2 环境感知

- 实时定位
- 环境建模
- 语义理解



你会联想到什么？

图片语义理解

11.2 环境感知

- 实时定位
- 环境建模
- 语义理解

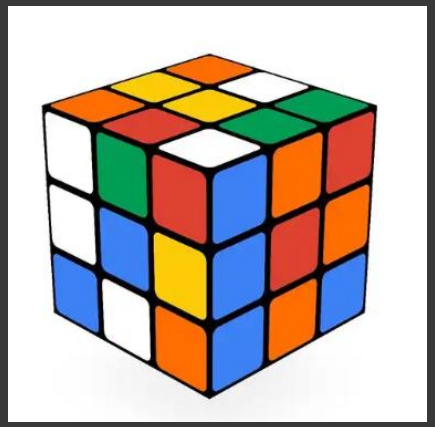
点云

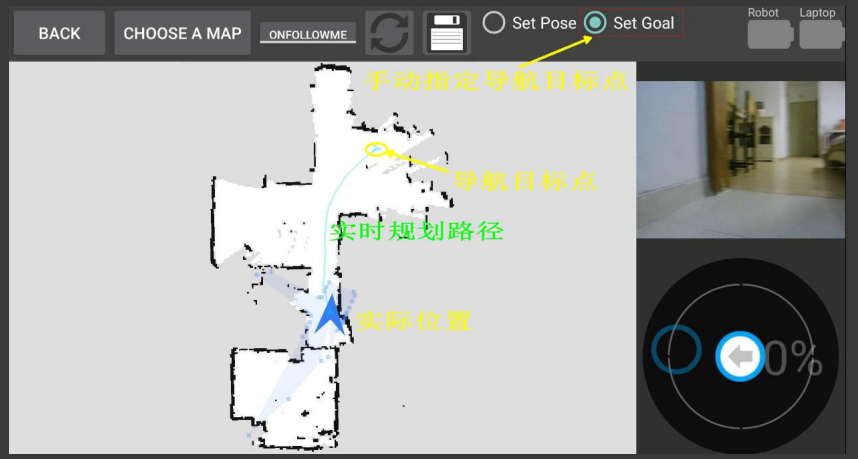
- 对环境状态的理解是多维度的，比如对于定位问题来说环境状态被机器人理解为特征点或点云，对于导航避障问题来说环境状态被机器人理解为二维或三维占据栅格。
- 站在更高层次去理解，会得到环境状态数据之间的各种复杂关系，即语义理解。
- 比如对于无人驾驶汽车而言，包括车道识别、路障识别、交通信号灯识别、移动物体识别、地面分割等。对于室内机器人的话，包括电梯识别、门窗识别、玻璃墙识别、镂空物体识别、斜坡识别等。
- 机器人要在环境中运动自如的话，离不开语义理解这项重要能力。

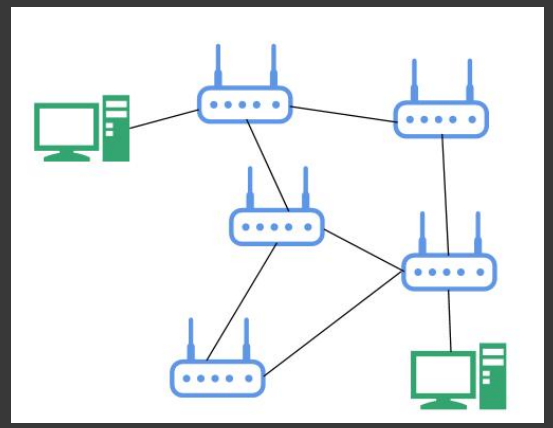
内容概要

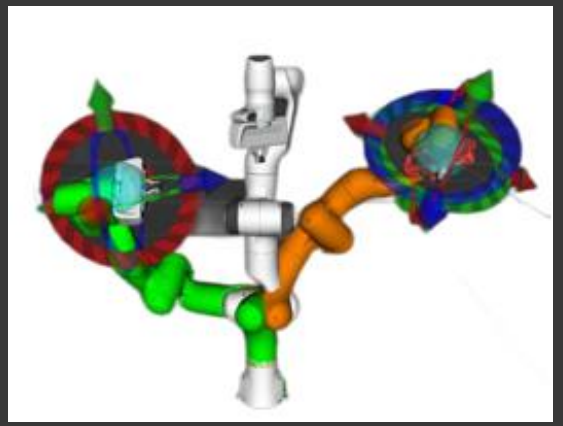
11.3 路径规划

- 广义的路径规划就是一种问题求解策略，比如魔方还原、积木拼图、计算机网络通信中的路由通路选择、数控机床车刀动作线、机械臂抓取动作、机器人自主导航等都蕴含着路径规划思想。
- 狭义的路径规划就是在度量地图上寻找到一条从起点到目标点可行通路的问题。



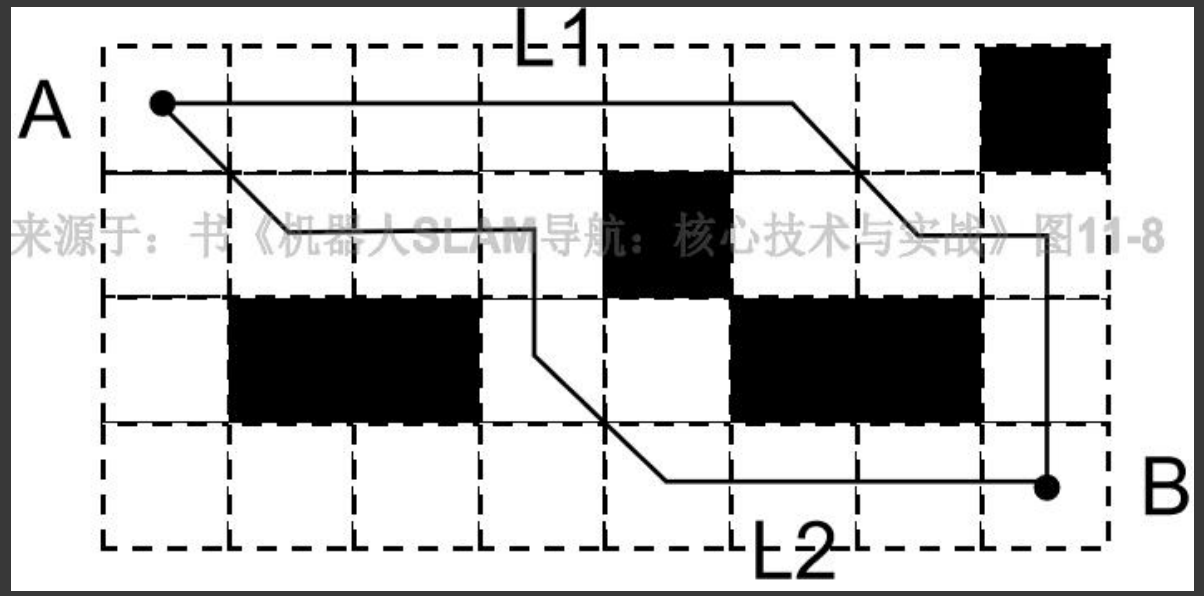






11.3 路径规划

- 广义的路径规划就是一种问题求解策略，比如魔方还原、积木拼图、计算机网络通信中的路由通路选择、数控机床车刀动作线、机械臂抓取动作、机器人自主导航等都蕴含着路径规划思想。
- 狭义的路径规划就是在度量地图上寻找到一条从起点到目标点可行通路的问题。



① 对于机器人自主导航问题，通常在给定的栅格地图上进行路径规划。

② 通过对整个栅格地图遍历很容易找到一条从A到B的路径，比如路径L1或L2，而且这样的路径并不唯一。

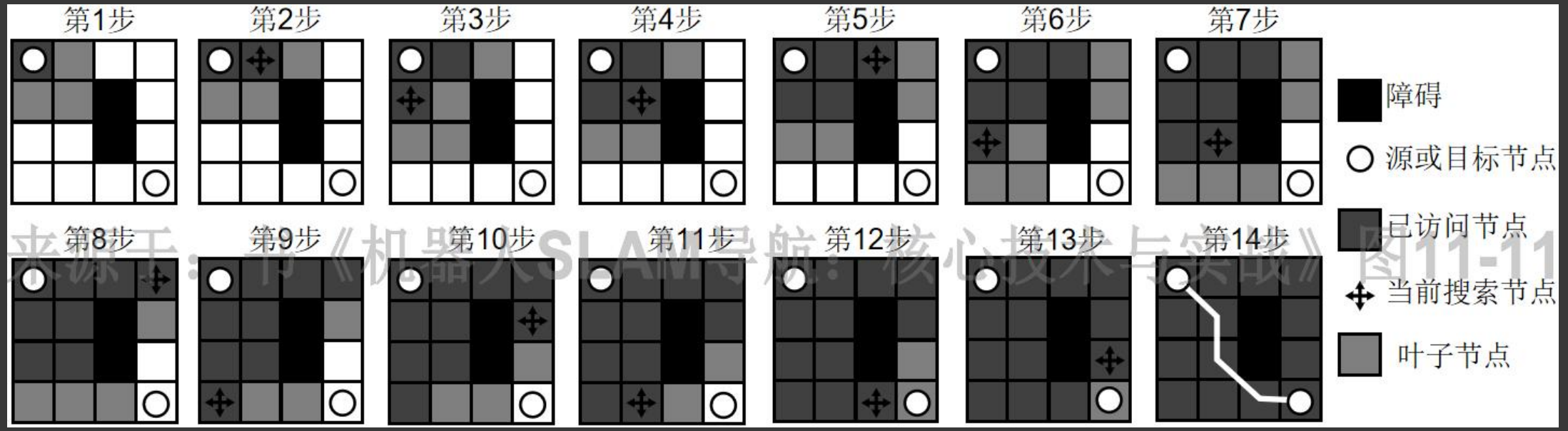
③ 在实际机器人导航中，不仅仅是找到一条可通行路径就完了，还必须考虑路径的各项性能（比如长度、平滑性、碰撞风险、各种附加约束等）。

11.3 路径规划

- 常见的路径规划算法
- 带约束的路径规划算法
- 覆盖的路径规划算法

常见的路径规划算法：
- 基于图结构的路径搜索
  - Dijkstra
  - A*
  - D*
  - AD*
  - D*-Lite
  - ...
- 基于采样的路径搜索
- 遗传算法
- 蚁群算法
- 模糊算法
- ...

$$\underbrace{P(A,F)}_{\arg\min} = \underbrace{P(A,D)}_{\arg\min} + P(D,F)$$



栅格地图上的Dijkstra搜索过程

11.3 路径规划

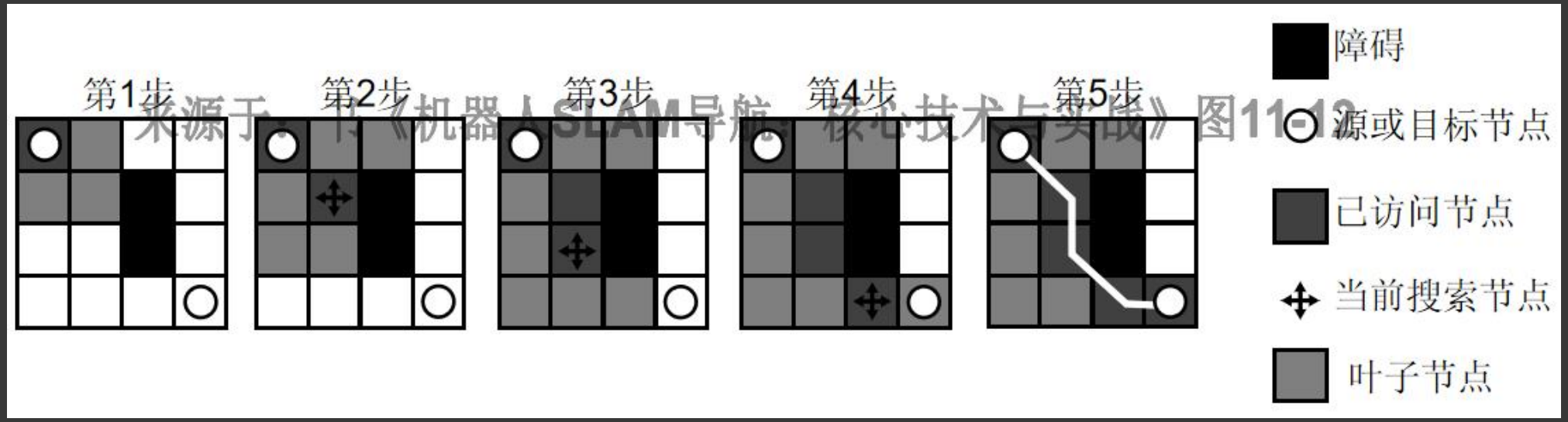
- 常见的路径规划算法
- 带约束的路径规划算法
- 覆盖的路径规划算法

常见的路径规划算法：
- 基于图结构的路径搜索
  - Dijkstra
  - A*
  - D*
  - AD*
  - D*-Lite
  - …
- 基于采样的路径搜索
- 遗传算法
- 蚁群算法
- 模糊算法
- …

$$\underbrace{P(A,F)}_{\arg\min} = \underbrace{P(A,D)}_{\arg\min} + P(D,F)$$

① $\underbrace{f(x)}_{\text{代价函数}} = \underbrace{g(x)}_{\text{实际代价}}$

② $\underbrace{f(x)}_{\text{代价函数}} = \underbrace{g(x)}_{\text{实际代价}} + \underbrace{h(x)}_{\text{估计代价（启发函数）}}$

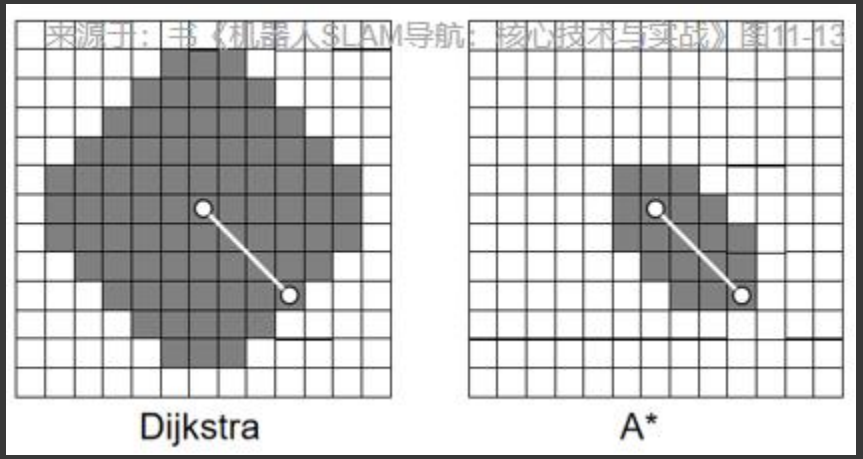


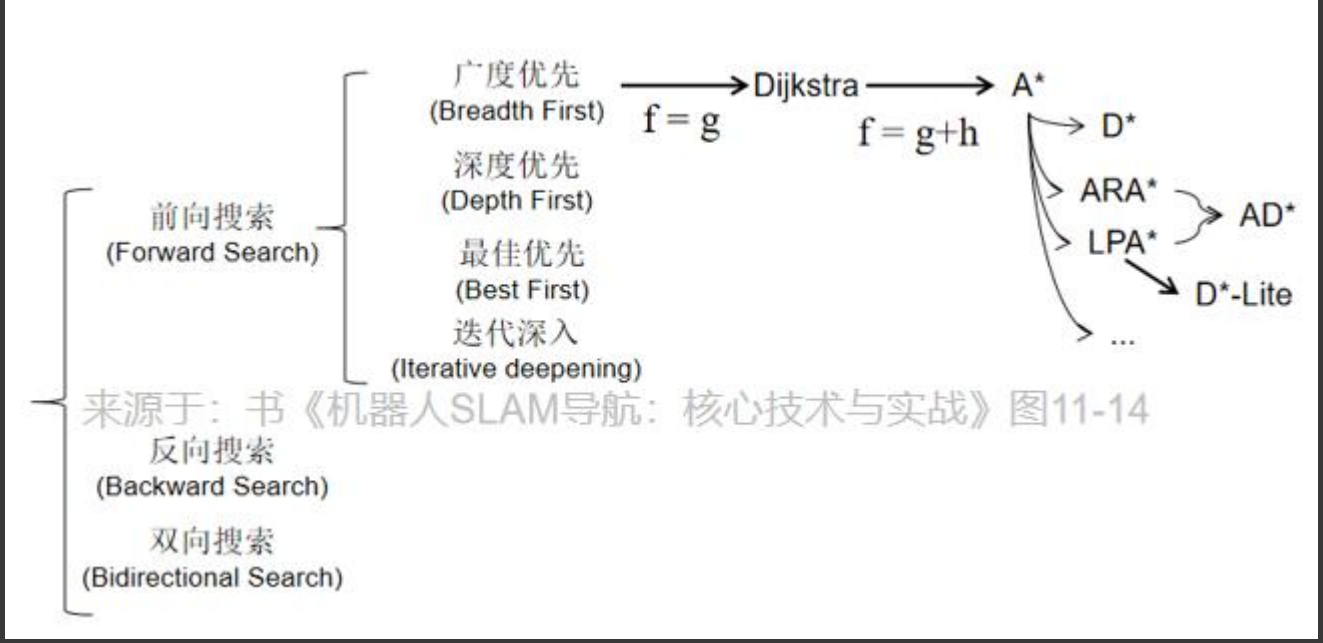
栅格地图上的A*搜索过程

11.3 路径规划

- 常见的路径规划算法
- 带约束的路径规划算法
- 覆盖的路径规划算法

常见的路径规划算法：
- 基于图结构的路径搜索
  - Dijkstra
  - A*
  - D*
  - AD*
  - D*-Lite
  - …
- 基于采样的路径搜索
- 遗传算法
- 蚁群算法
- 模糊算法
- …

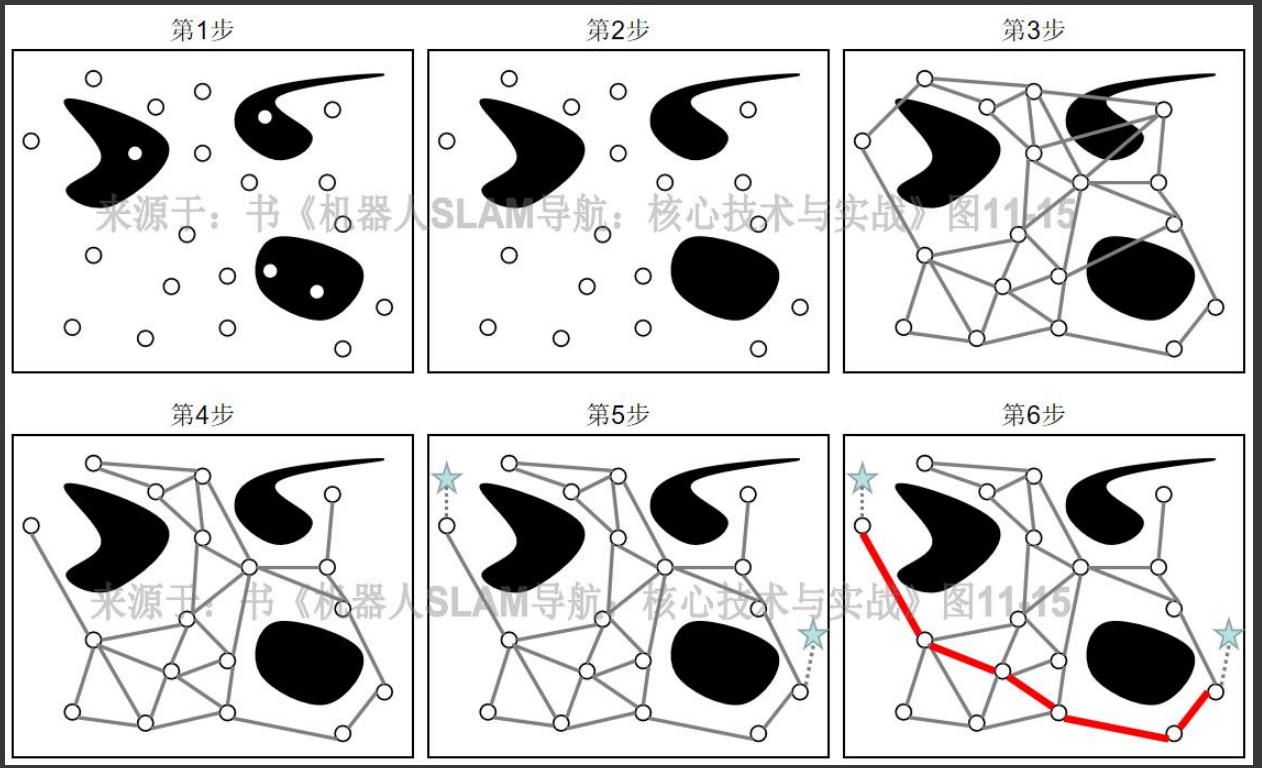




11.3 路径规划

- 常见的路径规划算法
- 带约束的路径规划算法
- 覆盖的路径规划算法

常见的路径规划算法：
- 基于图结构的路径搜索
- 基于采样的路径搜索
  - PRM
  - PRM*
  - RRT
  - Goal-Bias-RRT
  - Bi-RRT
  - Dynamic-RRT
  - RRT*
  - B-RRT*
  - SRRT*
  - ...
- 遗传算法
- 蚁群算法
- 模糊算法
- ...

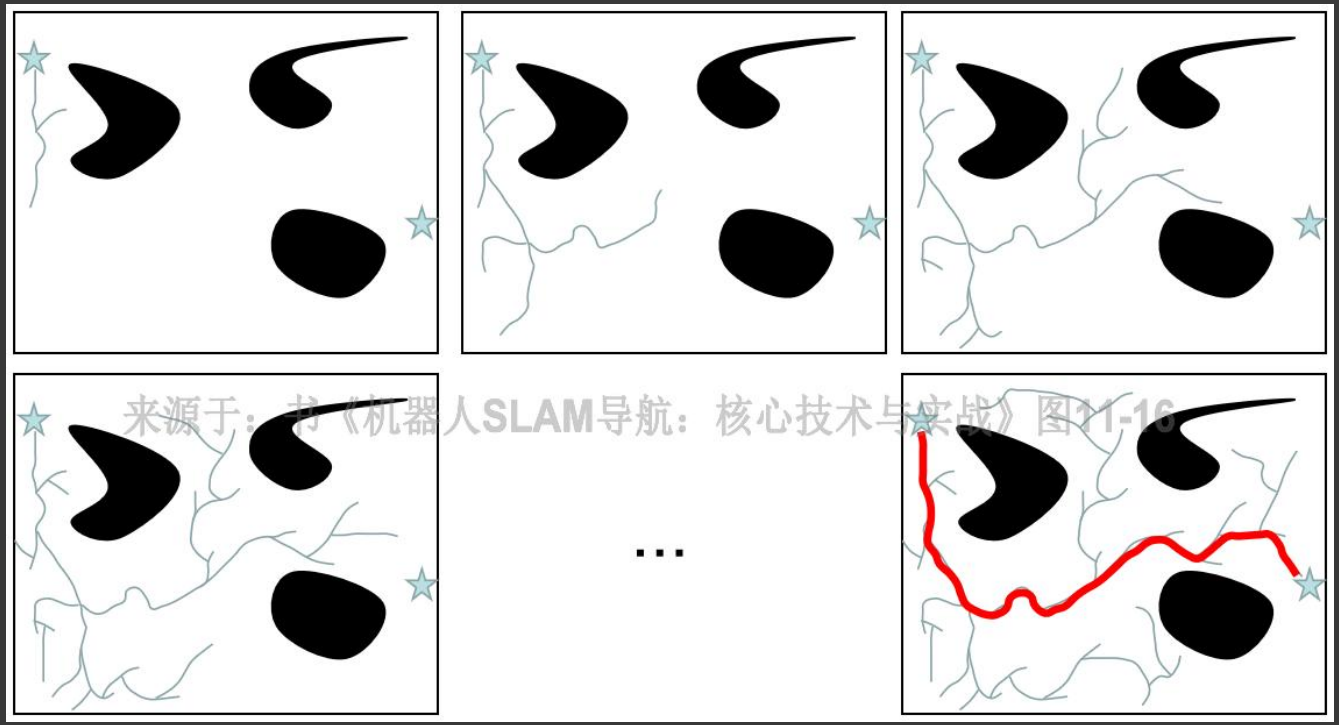


PRM工作流程

11.3 路径规划

- 常见的路径规划算法
- 带约束的路径规划算法
- 覆盖的路径规划算法

常见的路径规划算法：
- 基于图结构的路径搜索
- 基于采样的路径搜索
  - PRM
  - PRM*
  - RRT
  - Goal-Bias-RRT
  - Bi-RRT
  - Dynamic-RRT
  - RRT*
  - B-RRT*
  - SRRT*
  - …
- 遗传算法
- 蚁群算法
- 模糊算法
- …



来源于：书《机器人SLAM导航：核心技术与实战》图11-16

RRT工作流程

11.3 路径规划

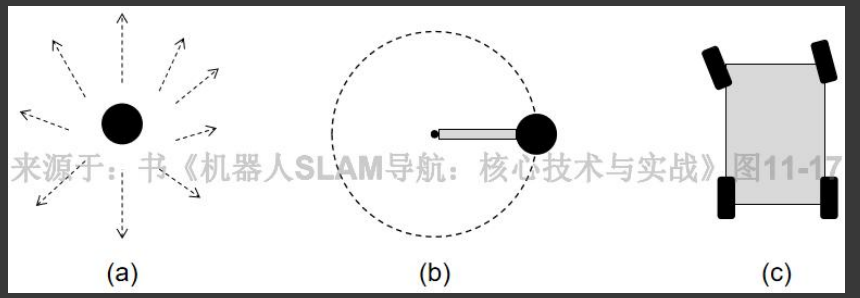
- 常见的路径规划算法
- 带约束的路径规划算法
  - 什么是约束
  - 几何约束
  - 微分约束
- 覆盖的路径规划算法

- 通常路径规划问题的解并不唯一，对于导航中的路径规划来说就是存在多条可行的路径。
- 在实际应用中，需要依据某些准则从多条可行的路径中挑选出最合适的一条路径，这就是带约束的路径规划。
- 要注意，很多时候最短路径，不一定是最优路径。

约束（Constraint）一词从字面意思理解就是限制条件，
从代数的角度看就是变量应该满足的等式或不等式方程，
从物理的角度看就是物体自由度受到限制。

$$x^2 + y^2 = 4, x > 0, y > 0$$

代数约束条件



来源于：书《机器人SLAM导航：核心技术与实战》图11-17

运动物体的自由度受限

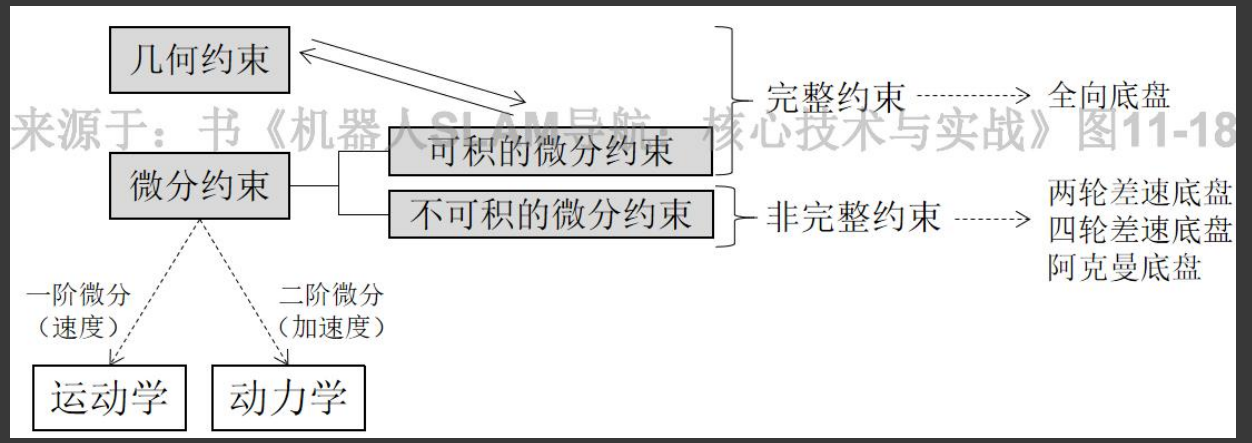
11.3 路径规划

- 常见的路径规划算法
- 带约束的路径规划算法
- 覆盖的路径规划算法

带约束的路径规划算法 → 什么是约束 → 完整约束与非完整约束；硬约束与软约束；定常约束与非定常约束

几何约束

微分约束



来源于：书《机器人SLAM导航：核心技术与实战》图11-18

硬约束：$\arg\min f(x),\ s.t.\begin{cases} g_1(x)=a_1, g_2(x)=a_2, \ldots, g_m(x)=a_m \\ h_1(x)>b_1, h_2(x)>b_2, \ldots, h_n(x)>b_n \end{cases}$

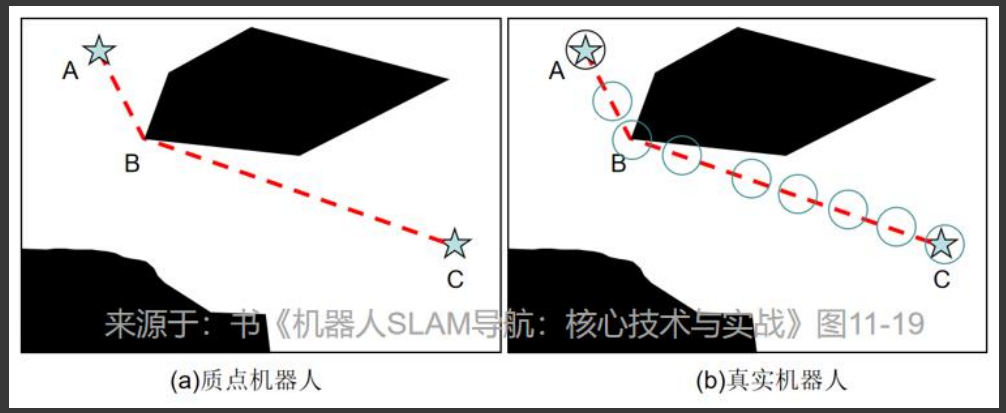
软约束：$\arg\min\{f(x)+\lambda_1 g_1(x)+\lambda_2 g_2(x)+\ldots\}$

定常约束：$x^2+y^2-R^2=0$

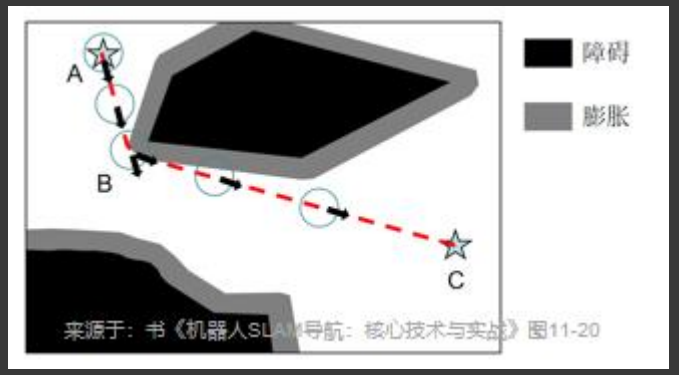
非定常约束：$x^2+y^2-(R(t))^2=0$

11.3 路径规划

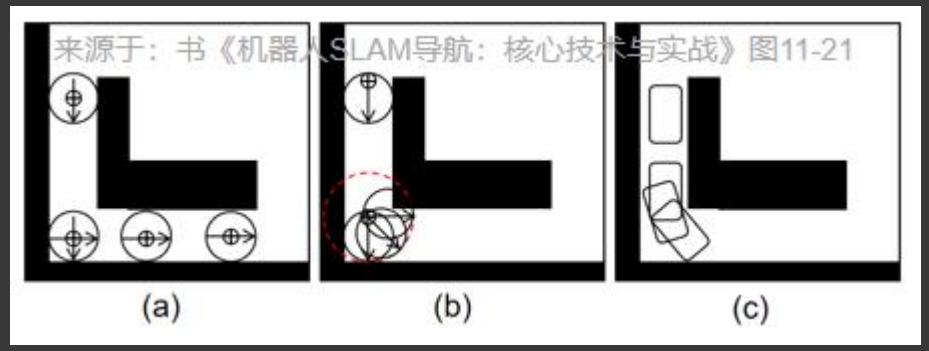
- 常见的路径规划算法
- 带约束的路径规划算法
  - 什么是约束
  - 几何约束
  - 微分约束
- 覆盖的路径规划算法



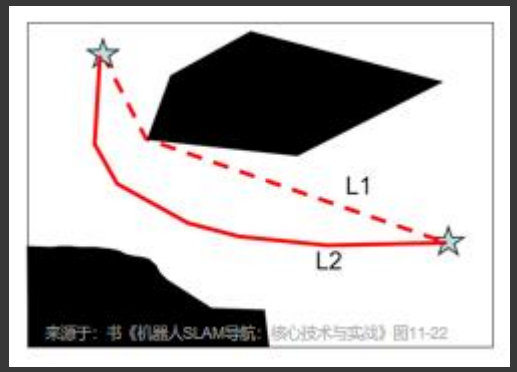
① 路径紧贴障碍物



② 给障碍物添加膨胀



③ 不同种类的机器人



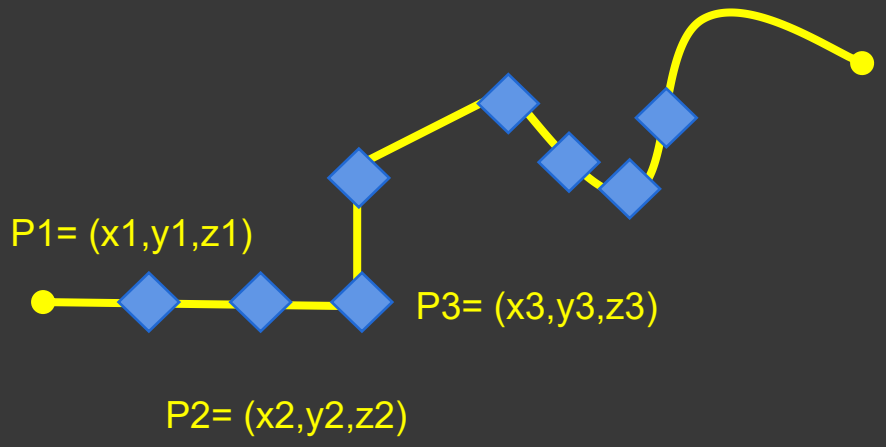
④ 路径尽可能避开障碍物

路径长度并不是判断最优路径的唯一准则，离障碍物距离、路径平滑性、路径执行效率等也是重要评价准则。

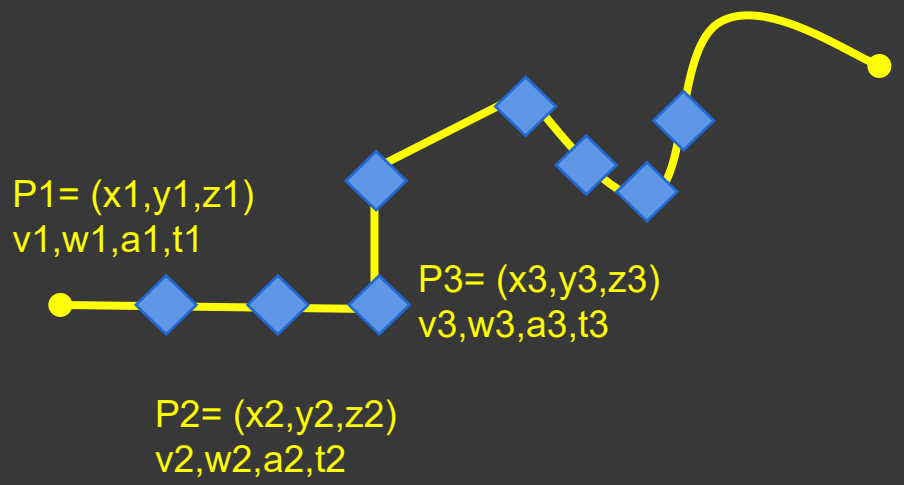
$$\arg\min_{L}\{\lambda_1 f_1(L)+\lambda_2 f_2(L)+\lambda_3 f_3(L)+\ldots\}$$

11.3 路径规划

- 常见的路径规划算法
- 带约束的路径规划算法
  - 什么是约束
  - 几何约束
  - 微分约束
- 覆盖的路径规划算法



几何约束的路径



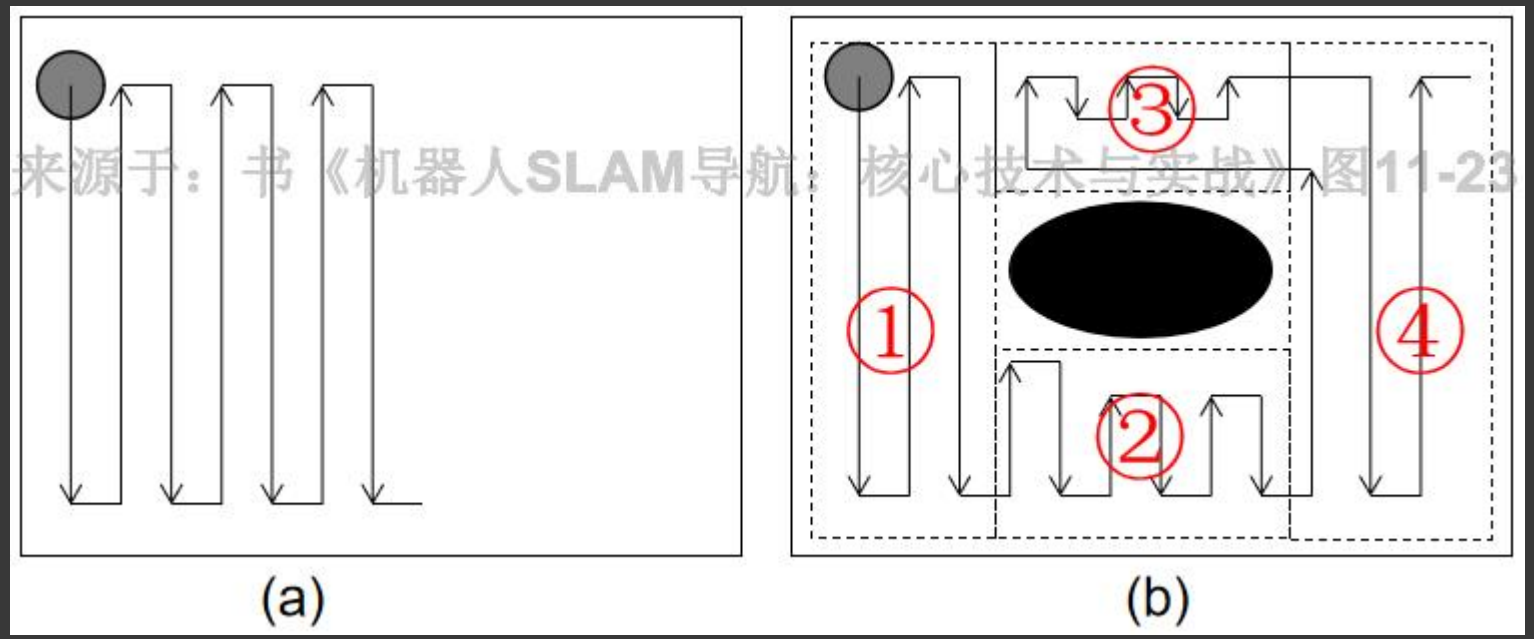
微分约束的路径

11.3 路径规划

- 常见的路径规划算法
- 带约束的路径规划算法
- 覆盖的路径规划算法

几何约束让路径不与障碍物发生碰撞，微分约束让路径能适于机器人实际执行。

结合具体应用场景，路径规划还有很多地方值得讨论，比如路径覆盖、路径调度、路径探索等。



路径覆盖

路径覆盖的一些算法实现思路：

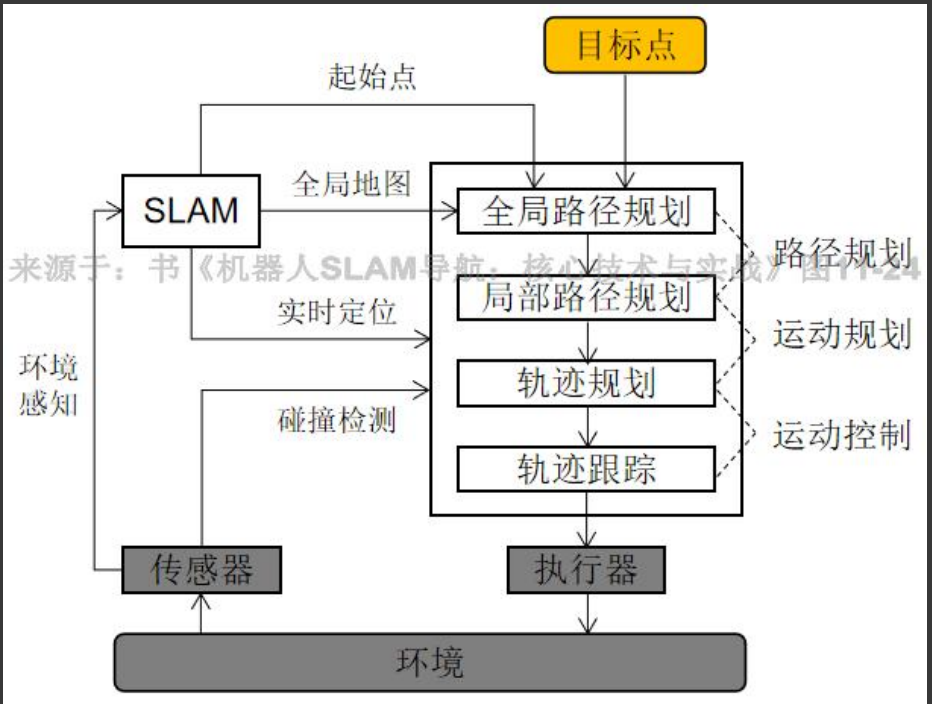
- **凸多边形分割**
- **搜索区域连接路线（TSP问题）**
- **Z字形路径覆盖**

内容概要

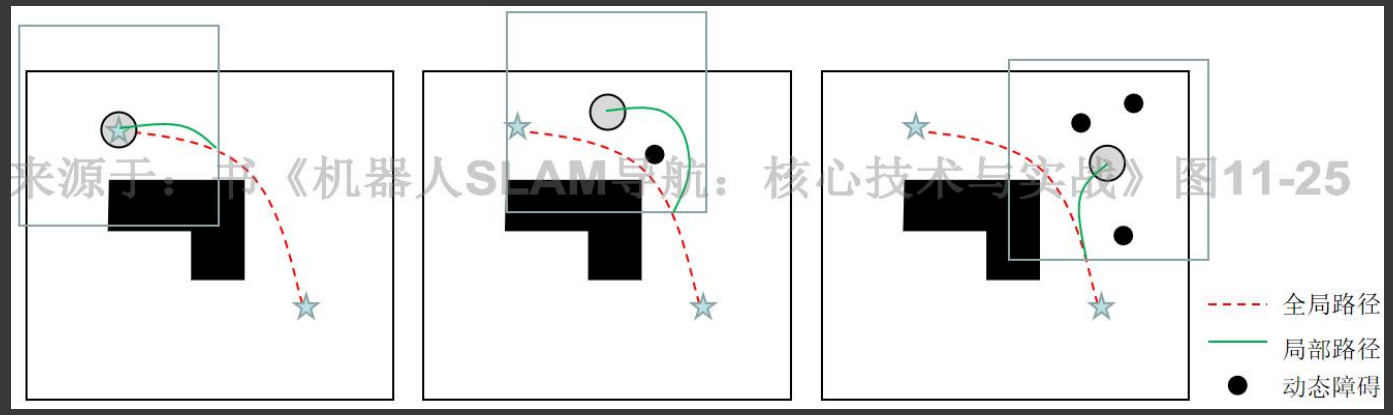
11.4 运动控制

从全局路径到作用在执行器的动作量就是一个**逐步细化**的过程：

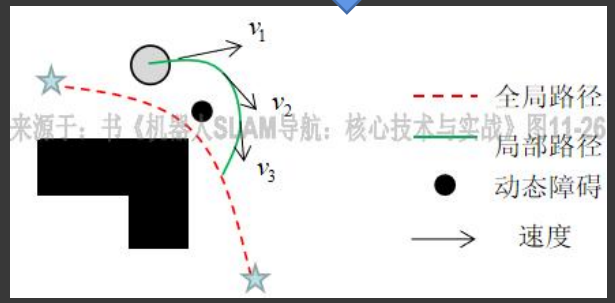
- 全局路径规划
- 局部路径规划
- 轨迹规划
- 轨迹跟踪



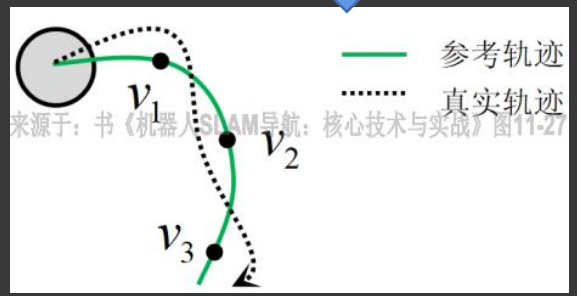
局部路径规划



轨迹规划

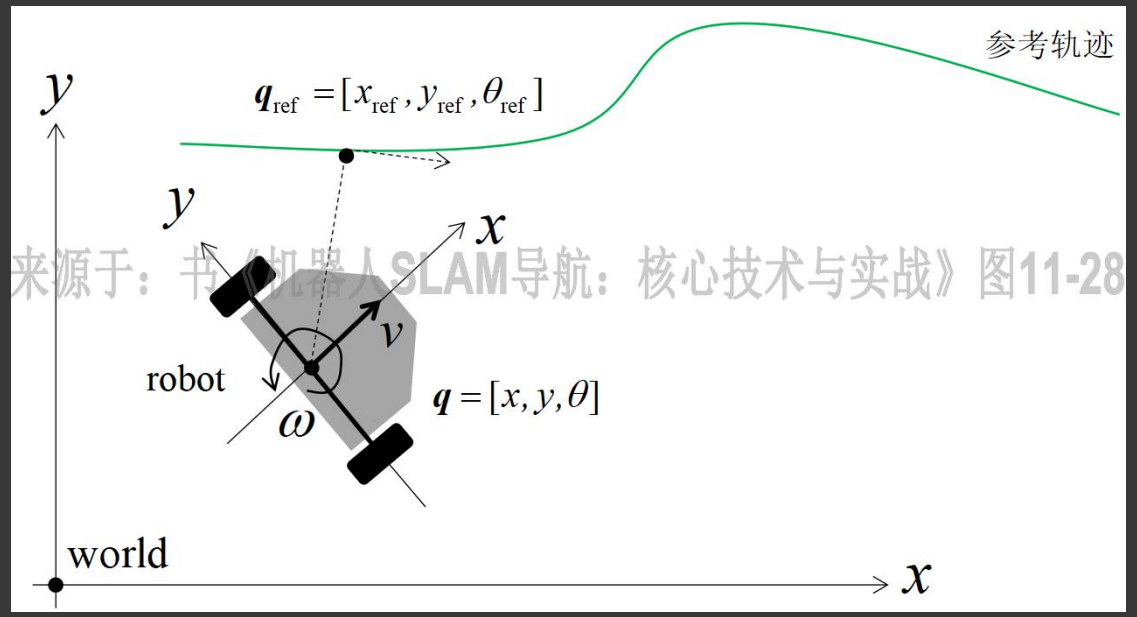


轨迹跟踪



11.4 运动控制

- 基于PID的运动控制
- 基于MPC的运动控制
- 基于强化学习的运动控制



难点

操控性：

- 两轮差速底盘
- 四轮差速底盘
- 阿克曼底盘
- 全向底盘

控制目标选择：

- 最近邻点
- 前进方向投射点
- ...

$$v(t)=k_p\cdot D_e(t)+k_i\cdot\sum_t D_e(t)+k_d\cdot(D_e(t)-D_e(t-1))$$

其中：$D_e(t)=\sqrt{(x-x_{ref})^2+(y-y_{ref})^2}$

$$\omega(t)=k'_p\cdot\theta_e(t)+k'_i\cdot\sum_t\theta_e(t)+k'_d\cdot(\theta_e(t)-\theta_e(t-1))$$

其中：$\theta_e(t)=\theta-\theta_{ref}$

修正

$$v(t)=k_p\cdot D_e(t)+k_i\cdot\sum_t D_e(t)+k_d\cdot(D_e(t)-D_e(t-1))+v_{ref}$$

其中：$D_e(t)=\sqrt{(x-x_{ref})^2+(y-y_{ref})^2}$

$$\omega(t)=k'_p\cdot\theta_e(t)+k'_i\cdot\sum_t\theta_e(t)+k'_d\cdot(\theta_e(t)-\theta_e(t-1))+\omega_{ref}$$

其中：$\theta_e(t)=\theta-\theta_{ref}$

非线性映射

$$\begin{pmatrix}v\\ \omega\end{pmatrix}=\begin{pmatrix}f(D_e,\theta_e,v_e,\omega_e)\\ g(D_e,\theta_e,v_e,\omega_e)\end{pmatrix}$$

11.4 运动控制

- 基于PID的运动控制
- 基于MPC的运动控制
- 基于强化学习的运动控制

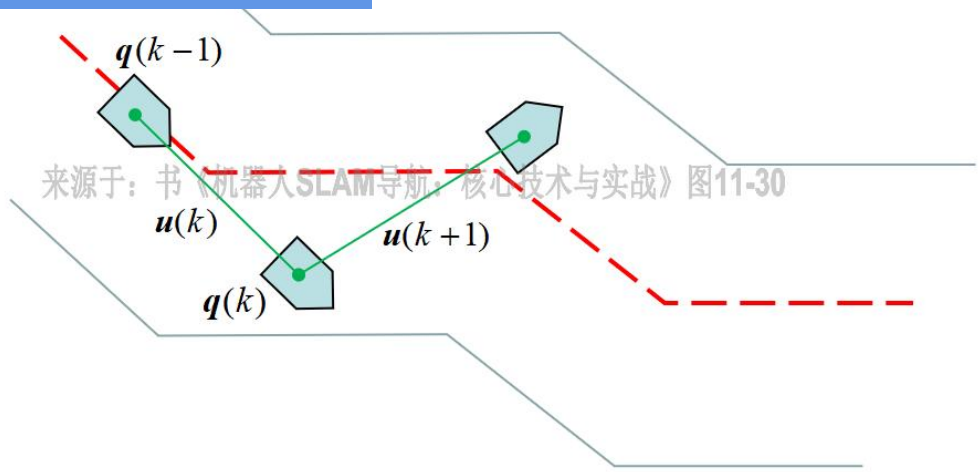
传统PID控制存在控制**滞后**的问题。

这种滞后性对**高速运动**的机器人或无人车产生极大的安全隐患。

MPC（Model Predictive Control，模型预测控制）：

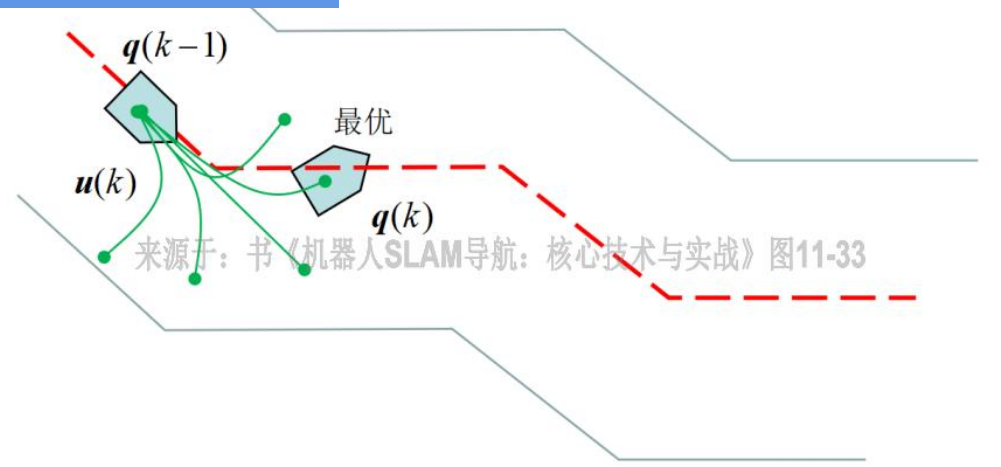
- 模型构建
- 预测轨迹偏离度评价
- MPC优化求解

控制滞后带来的问题



来源于：书《机器人SLAM导航：核心技术与实战》图11-30

MPC优化求解过程

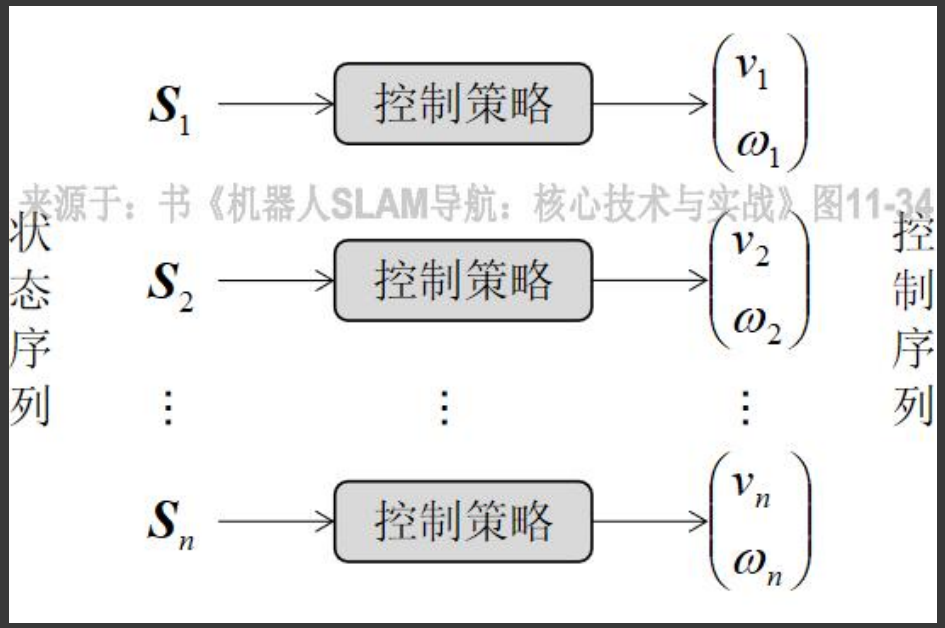


来源于：书《机器人SLAM导航：核心技术与实战》图11-33

11.4 运动控制

- 基于PID的运动控制
- 基于MPC的运动控制
- 基于强化学习的运动控制

- 自主导航给人的直观感受就是机器人由外界状态触发产生的一系列运动。
- 机器人的输入就是各种状态信息，比如描述全局障碍的地图信息、描述动态障碍的实时传感器扫描信息、机器人定位信息、目标信息等。
- 机器人的输出就是执行以线速度v和角速度w为实际控制量的运动过程，当然输出是由一系列运动控制量组成的控制序列。
- 那么自主导航问题就是在寻找**输入状态**与**输出控制序列**之间的映射关系，这种映射关系也就是**控制策略**。
- 输入状态S是一个不断变化的量，比如环境中动态障碍物的变动、机器人的移动、定位偏差等都会产生新输入状态。这个变化的状态量经过控制策略映射出对应控制量，控制量自然也在不断变化。这样来看的话，控制策略不仅仅是解决单个输入状态到单个控制量的映射，而是解决整个状态序列到控制序列的映射。



来源于：书《机器人SLAM导航：核心技术与实战》图11-34

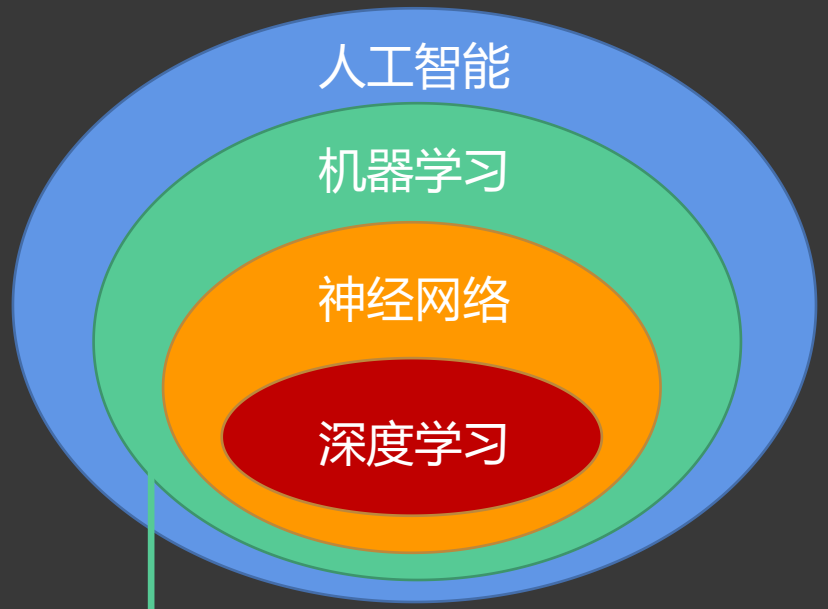
将控制策略分解成不同环节逐一建模是目前主流的做法，比如按照层级方式将求解过程分解为路径规划、运动规划、运动控制等环节。在每个环节中构相应的建数学模型，这些数学模型都具有确切的物理意义。

对于处在复杂环境中的机器人，建立精确的数学模型一点都不简单。因为要同时考虑全局障碍、动态障碍、机器人自身运动约束限制、操控的舒适性、轨迹偏差等因素，在数学模型上的表现就是各种约束条件。从上面讨论的路径规划和运动控制内容来看，无论是路径搜索和带约束的路径规划，还是PID和MPC运动控制，都相当困难。

那么有没有更容易的控制策略求解方法呢？我想强化学与自主导航相结合可以回答这个问题。

内容概要

11.5 强化学习与自主导航

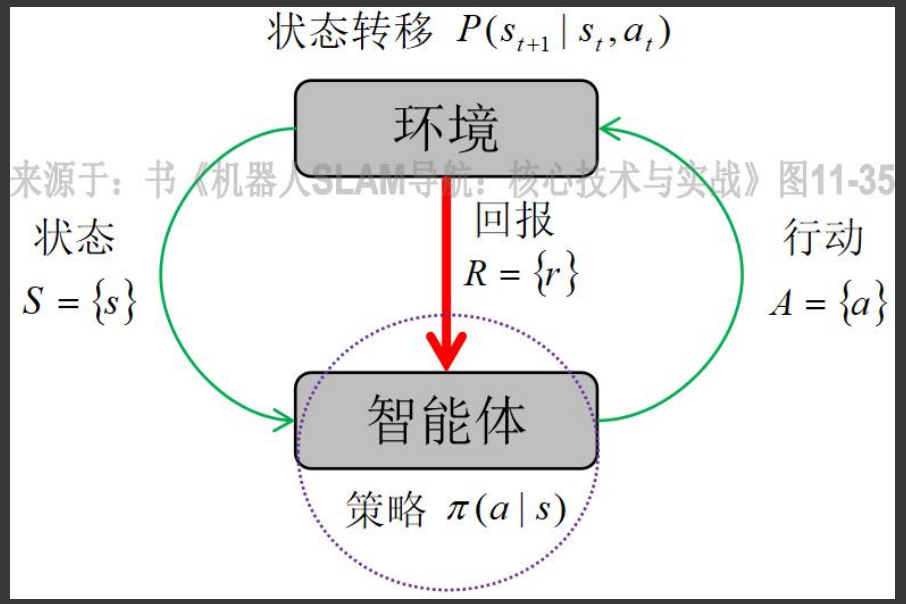


机器学习
- 数据
- 模型：神经网络、决策树、概率图等
- 学习策略：有监督学习、无监督学习、强化学习等
- 学习算法：学习策略在程序上的具体实现（比如误差反向传播算法）

学习策略：
- 有监督学习：以最小化误差为学习目标，解决分类、回归等问题 → 车牌识别
- 无监督学习：以特定评价准则（比如让系统趋于平衡态、让系统组织最简化等）为学习目标，解决聚类、数据压缩等问题 → 自动编解码器
- 强化学习：以最大化回报为学习目标，解决序贯决策等问题 → 围棋比赛

11.5 强化学习与自主导航

- 强化学习
- 基于强化学习的自主导航



强化学习的过程

- 智能体就是承载强化学习算法的主体，比如机器人。
- 智能体与环境之间通过状态s和行动a实现交互，同时环境会对智能体的每次行动给予回报r。
- 假设智能体的任务是完成在地图中自主导航，那么状态s就代表机器人在地图中的位置以及周围障碍情况，行动a就代表机器人的线速度和角速度。
- 回报r则是对机器人当前行动a表现好坏的评价，比如行动a执行后使得机器人处于不利状态（靠近障碍物、与障碍物发生碰撞、远离导航目标点等）时回报r为负数值，而行动a执行后使得机器人处于有利状态（远离障碍物、靠近导航目标点等）时回报r为正数值。当然定义回报的形式并不唯一，可以根据实际任务及需求来定义。
- 连接状态与行动关系的就是策略$\pi(a|s)$，连接行动与状态关系的则是状态转移$P(s_{t+1}|s_t,a_t)$。
- 强化学习过程其实就是利用试探行动获得的回报不断调整策略$\pi(a|s)$，直到策略$\pi(a|s)$最优为止。

$$s_0, a_0, s_1, a_1, s_2, a_2, s_3$$
$$\quad\ -1 \quad\ +1 \quad\ -1$$

思考：什么样的策略$\pi(a|s)$才算最优策略呢？

11.5 强化学习与自主导航

- 强化学习
- 基于强化学习的自主导航

马尔可夫决策过程
价值函数
贝尔曼方程
贝尔曼最优方程
马尔可夫决策过程求解方法

强化学习主要用来解决序贯决策问题，而序贯决策问题通常用马尔可夫决策过程（Markov Decision Process，MDP）来描述，下面对马尔可夫决策过程的数学形式进行介绍。

$\pi(a_t \mid s_t)$

$P(s_{t+1} \mid s_t, a_t)$

- 马尔可夫性：

系统当前状态仅与前一时刻状态有依赖关系，而与更早的状态无关

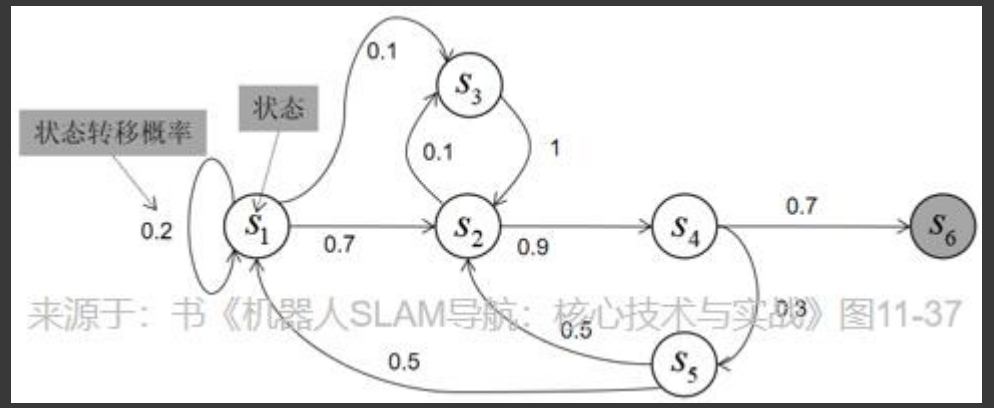
- 马尔可夫过程：

具有马尔可夫性的随机过程

1. 时间离散、状态离散的马尔可夫过程，称为马尔可夫链。
2. 时间连续、状态离散的马尔可夫过程，称为连续时间的马尔可夫链。
3. 时间连续、状态连续的马尔可夫过程，称为一般的马尔可夫过程。

- 马尔可夫决策过程：

考虑了决策行动A的马尔可夫过程就是马尔可夫决策过程，马尔可夫决策过程由元组(S,A,P,R,γ)表示



时间、状态均离散的马尔可夫过程（或马尔可夫链）

11.5 强化学习与自主导航

- 强化学习
  - 马尔可夫决策过程
  - 价值函数
  - 贝尔曼方程
  - 贝尔曼最优方程
  - 马尔可夫决策过程求解方法
- 基于强化学习的自主导航

已知当前状态 s_t 时，按照某种策略 π 进行交互产生的长期回报 G_t 的期望，就定义为**状态的价值函数** $v_\pi(s_t)$，$v_\pi(s_t)$ 可以理解为策略 π 在状态 s_t 时的价值，s_t 可以取状态空间 S 中的任意状态值，所以 $v_\pi(s_t)$ 也是关于 s_t 的函数。

已知当前状态 s_t 和行动 a_t 时，按照某种策略 π 进行交互产生的长期回报 G_t 的期望，就定义为**状态-行动的价值函数** $q_\pi(s_t, a_t)$ 。

状态的价值函数 $v_\pi(s_t) = E_\tau[G_t \mid s_t]$

状态-行动的价值函数 $q_\pi(s_t, a_t) = E_\tau[G_t \mid s_t, a_t]$

11.5 强化学习与自主导航

- 强化学习
  - 马尔可夫决策过程
  - 价值函数
  - 贝尔曼方程
  - 贝尔曼最优方程
  - 马尔可夫决策过程求解方法
- 基于强化学习的自主导航

贝尔曼方程描述了价值函数t与t+1时刻的递推关系

$$\underline{v_\pi(s_t)} = \sum_{a_t} \pi(a_t \mid s_t) \sum_{s_{t+1}} P(s_{t+1} \mid s_t, a_t) \left(r_{t+1} + \gamma \bullet \underline{v_\pi(s_{t+1})} \right)$$

$$\underline{q_\pi(s_t, a_t)} = \sum_{s_{t+1}} P(s_{t+1} \mid s_t, a_t) \sum_{a_{t+1}} \pi(a_{t+1} \mid s_{t+1}) \left(r_{t+1} + \gamma \bullet \underline{q_\pi(s_{t+1}, a_{t+1})} \right)$$

11.5 强化学习与自主导航

- 强化学习
- 基于强化学习的自主导航

马尔可夫决策过程

价值函数

贝尔曼方程

贝尔曼最优方程

马尔可夫决策过程求解方法

将贝尔曼方程代入最优问题中就得到了贝尔曼最优方程，贝尔曼最优方程为判断某个策略是否达到最优提供了理论依据。

$$\forall s_t \in S, \pi \geq \pi' \Leftrightarrow v_\pi(s_t) \geq v_{\pi'}(s_t)$$

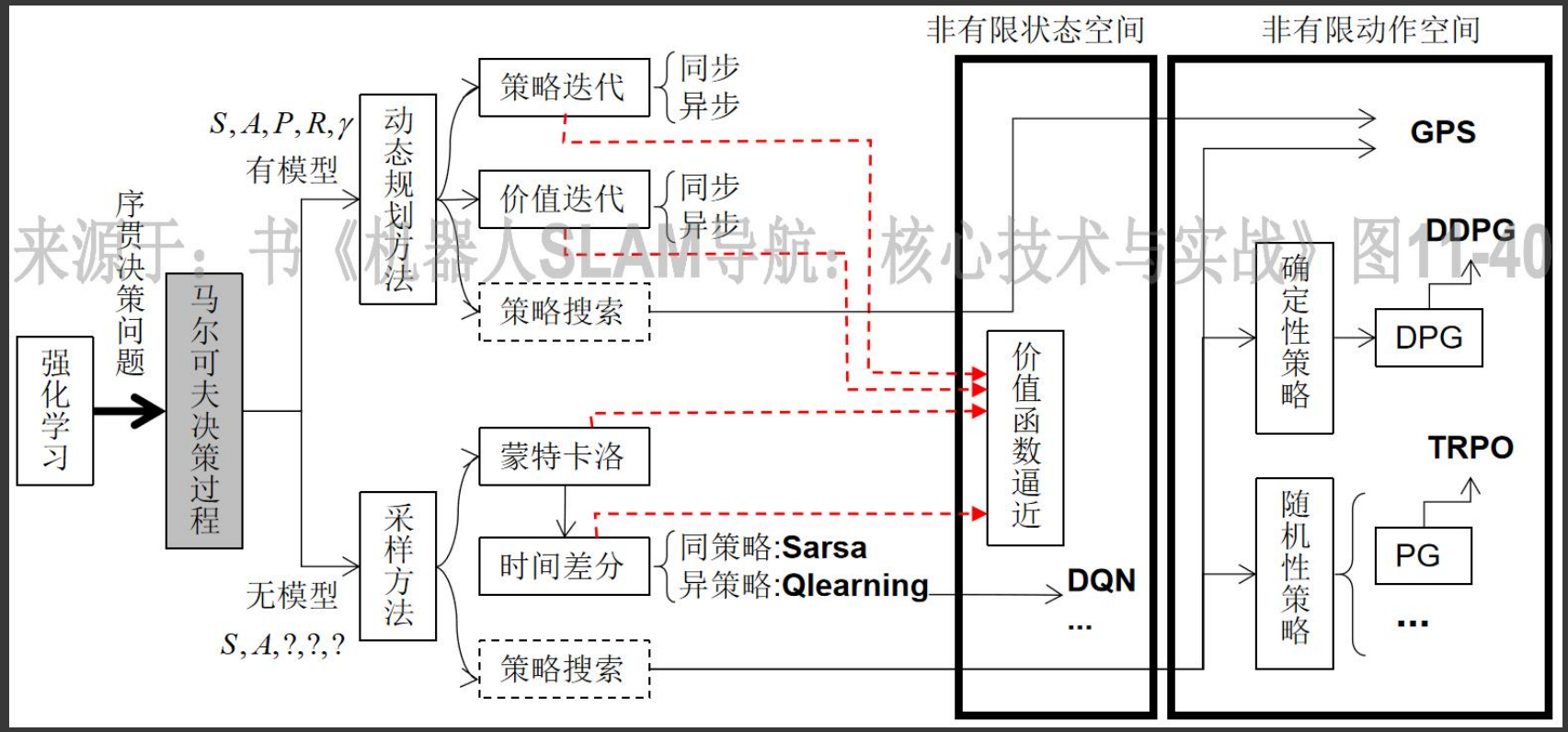
$$v^*(s_t) = \arg\max_\pi v_\pi(s_t)$$

$$q^*(s_t, a_t) = \arg\max_\pi q_\pi(s_t, a_t)$$

11.5 强化学习与自主导航

- 强化学习
- 基于强化学习的自主导航

马尔可夫决策过程
价值函数
贝尔曼方程
贝尔曼最优方程
马尔可夫决策过程求解方法

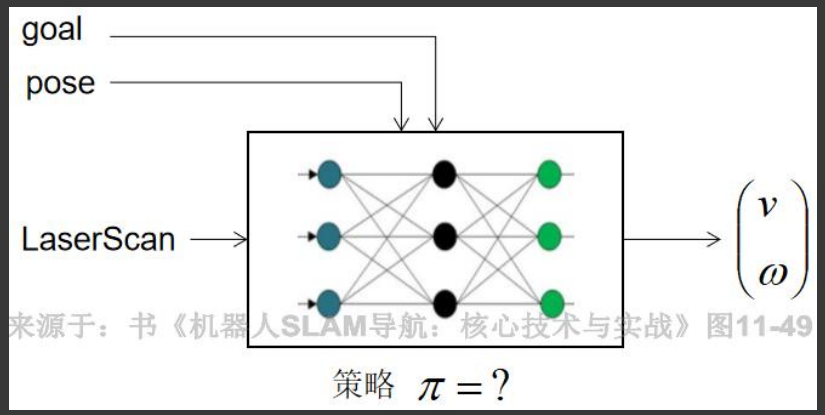


关于马尔可夫决策过程求解方法的具体讲解，将在第2季课程中展开讨论

11.5 强化学习与自主导航

- 强化学习
- 基于强化学习的自主导航

端到端的自主导航问题：



用强化学习实现自主导航的思路很多，这里以具体例子来说明：

➢ AutoRL

1. 借助强化学习可以将非常复杂的传统自主导航问题变成简单的端到端问题。
2. 以导航目标点(goal)、机器人定位（pose）和雷达扫描数据（LaserScan）直接为输入，以机器人的线速度和角速度控制量直接为输出。
3. 传统的强化学习算法（比如DDPG）基于人为给定的回报函数对策略π进行学习，而自动强化学习（AutoRL）直接对回报函数的形式以及策略π同时进行学习。

AutoRL将策略、价值函数和回报函数同时进行参数化：

$$\pi(s \mid \theta_\pi) = FF(\theta_\pi)$$

$$Q(s,a \mid \theta_Q) = FF(\theta_Q)$$

$$R(s,a \mid \theta_r) = \sum_i r(s,a,\theta_{r_i})$$

AutoRL具体学习迭代过程：

$$\theta_r' = \arg\max_i J(\theta_\pi, \theta_Q, \theta_r^i)$$

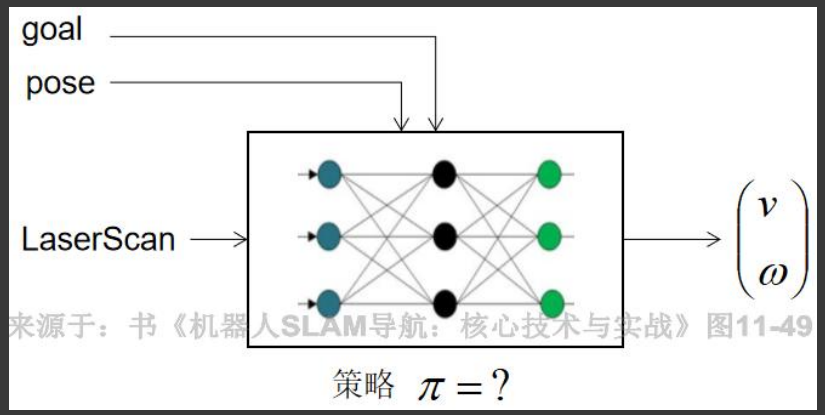
$$\theta_\pi', \theta_Q' = \arg\max_j J(\theta_\pi^j, \theta_Q^j, \theta_r')$$

$$\pi'(s \mid \theta_\pi') = AutoRL\left(Actor(\theta_\pi'), Critic(\theta_Q'), R(\theta_r')\right)$$

11.5 强化学习与自主导航

- 强化学习
- 基于强化学习的自主导航

端到端的自主导航问题：



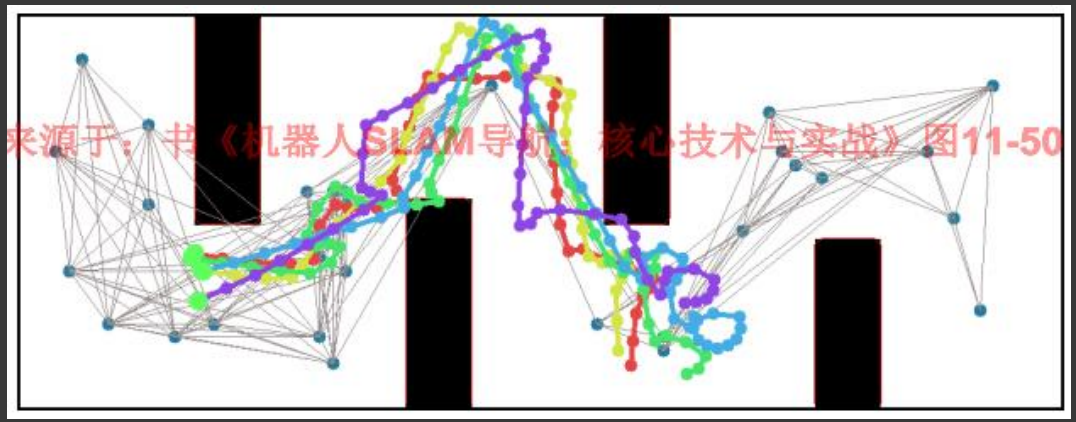
来源于：书《机器人SLAM导航：核心技术与实战》图11-49

用强化学习实现自主导航的思路很多，这里以具体例子来说明：

➢ PRM-RL

AutoRL在小范围静态环境中进行训练，所以AutoRL其实只相当于实现了局部地图自主导航。而PRM-RL是传统路径规划（PRM）与强化学习（RL）的结合，PRM负责从起始点到目标点采样可行的路线图，RL则通过所学习策略从这些路线图中挑选出一条最合适的。

PRM-RL：

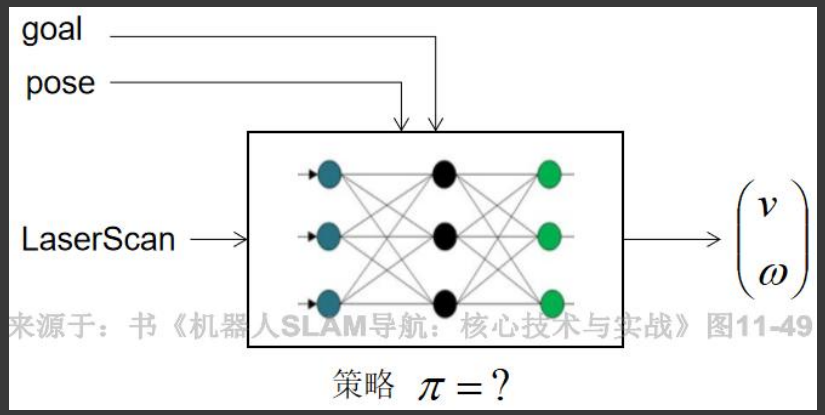


来源于：书《机器人SLAM导航：核心技术与实战》图11-50

11.5 强化学习与自主导航

- 强化学习
- 基于强化学习的自主导航

端到端的自主导航问题：



来源于：书《机器人SLAM导航：核心技术与实战》图11-49

用强化学习实现自主导航的思路很多，这里以具体例子来说明：

➢ AutoRL+PRM-RL

将PRM-RL中的传统强化学习方法替换成AutoRL，那么导航效果会更加稳健，

这就是AutoRL+PRM-RL。

- 例程源码下载： https://github.com/xiihoo/Books_Robot_SLAM_Navigation
- 课件PPT下载： www.xiihoo.com

敬请关注,长期更新...

下集预告