



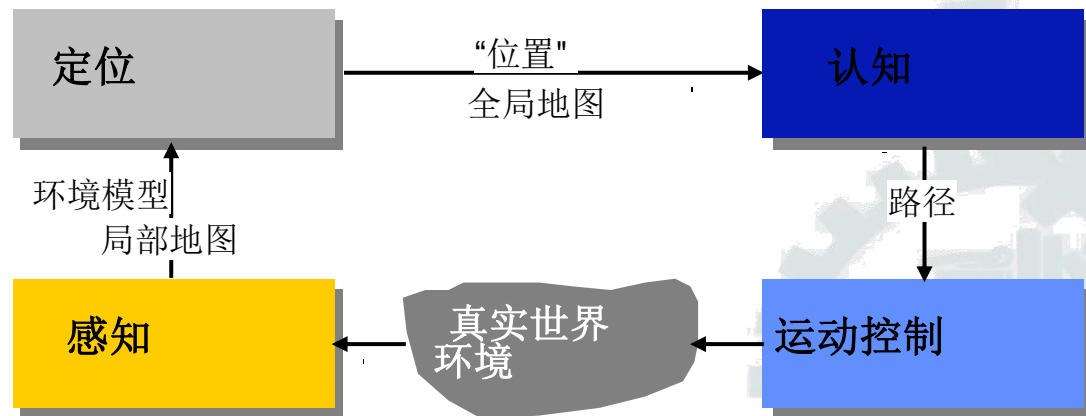
第5章 移动机器人的定位

- ◇ 定位的挑战：误差讨论
- ◇ 信任度：单假设、多假设信任度
- ◇ 地图
- ◇ 马尔可夫定位
- ◇ 卡尔曼滤波器定位



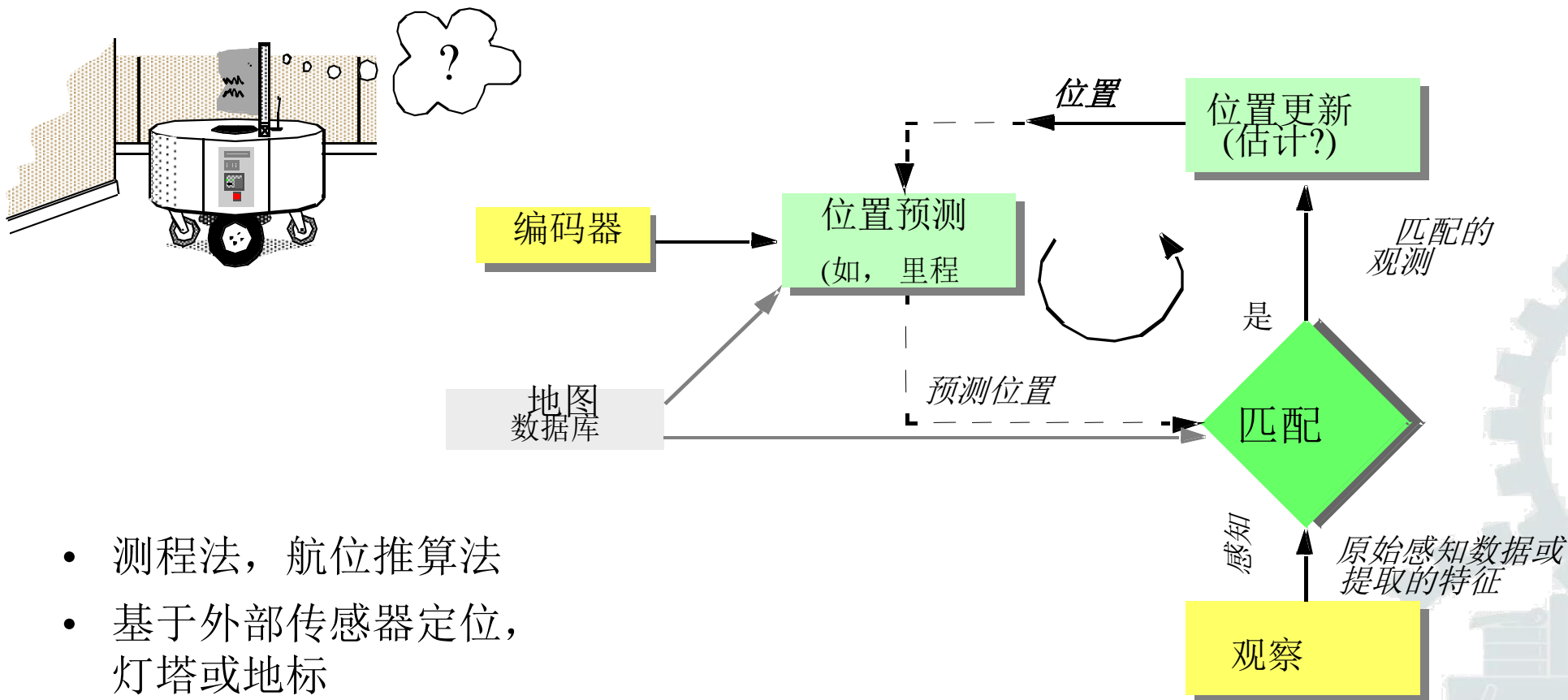
移动机器人的定位

- ◇ 导航是移动机器人应具备的最具挑战性的能力之一
- ◇ 导航需要四个模块的成功运行：
 - ◇ 感知：机器人获取并解释传感器信息，提取有意义特征；
 - ◇ 定位：机器人确定自身在环境中的位置；
 - ◇ 认知：机器人识别并决定如何运动以达到目标；
 - ◇ 运动控制：实现期望的运动轨迹





定位，我在哪里？



- 测程法，航位推算法
- 基于外部传感器定位，灯塔或地标
- 基于概率地图的定位



定位的挑战

- ◇ 仅知道绝对位置 (如 GPS) 是不够的
- ◇ 以人类尺度在环境中的定位
- ◇ 在“认知”环节，需要比位置更多的输入（定位还不够，还要使用地图，规划达到目标的策略）
- ◇ 感知和执行十分重要
 - ◇ 传感器噪声
 - ◇ 传感器混叠（混淆）
 - ◇ 执行机构噪声（导致执行结果不准确）
 - ◇ 测程法位置估计（积分型位置计算）



传感器噪声

- ◇ 传感器噪声主要受环境影响
如，表面，照明 ...

- ◇ 或因测量原理本身
如，超声波测距时的相互干扰

- ◇ 传感器噪声显著减少了传感器获得信息中的有用信息含量。解决方法：
 - ◇ 采取多次读取数据的方法（但会降低采样速率）
 - ◇ 运用时间融和、多传感器融和方法



传感器混叠

- ◇ 机器人传感器读数的非惟一性是正常的
- ◇ 即便有多个传感器，仍然存在从环境状态到机器人感知输入之间的多对一的映射关系
- ◇ 因此，机器人根据单次传感器接收到的信息量不足以确定自身的位置
 - ◇ 机器人定位通常基于传感器得到的数据序列
 - ◇ 随着时间的过去，恢复出足够的信息（如，滤波方法）



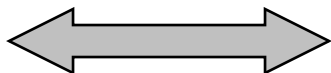
执行器噪声：里程表，航位推算法

- ◇ 里程表和航位推算法：
位置的更新依据本体感受传感器
 - ◇ 里程表：车轮轴连编码器（测路程）
 - ◇ 航位推算法：还可能要用到航向传感器（如陀螺仪）
- ◇ 将车轮编码器（和航向传感器）数据积分得到机器人的移动。
 - ◇ 优点：直观，简单
 - ◇ 缺点：误差被累积 -> 无界
- ◇ 使用额外的航向传感器（如陀螺）可能有助于减小累积误差，但问题的本质仍然存在。
- ◇ 执行的误差，也被看做是里程表偏差



里程表法：误差源

确定性的
(系统的)



非确定性的
(非系统的)

- ◇ 确定性的误差可采用系统校正（标定）的方法消除。
- ◇ 非确定性的误差必须用误差模型描述，必然导致位置估计的不确定性。
- ◇ 主要误差源：
 - ◇ 积分中的有限分辨率（时间增量、测量分辨率 ...）
 - ◇ 轮子的不正确装配 (确定性的)
 - ◇ 轮子直径不等(确定性的)
 - ◇ 轮子触地点的变化
 - ◇ 不同的地面接触 (打滑、不平坦 ...)
 - ◇ ...



里程表法: 积分误差的分类

- ◇ 路程误差: 机器人运动的路程积分 (距离)
 - ◇ 轮子移动的总和
- ◇ 转动误差: 与路程误差类似, 但转动的产生
 - ◇ 差动驱动时, 轮子移动之差
- ◇ 漂移误差: 轮子误差的差异, 导致产生机器人转角方向的误差

很长时间运行后, 转动和漂移误差
会远远比路程误差严重 (影响朝向) !

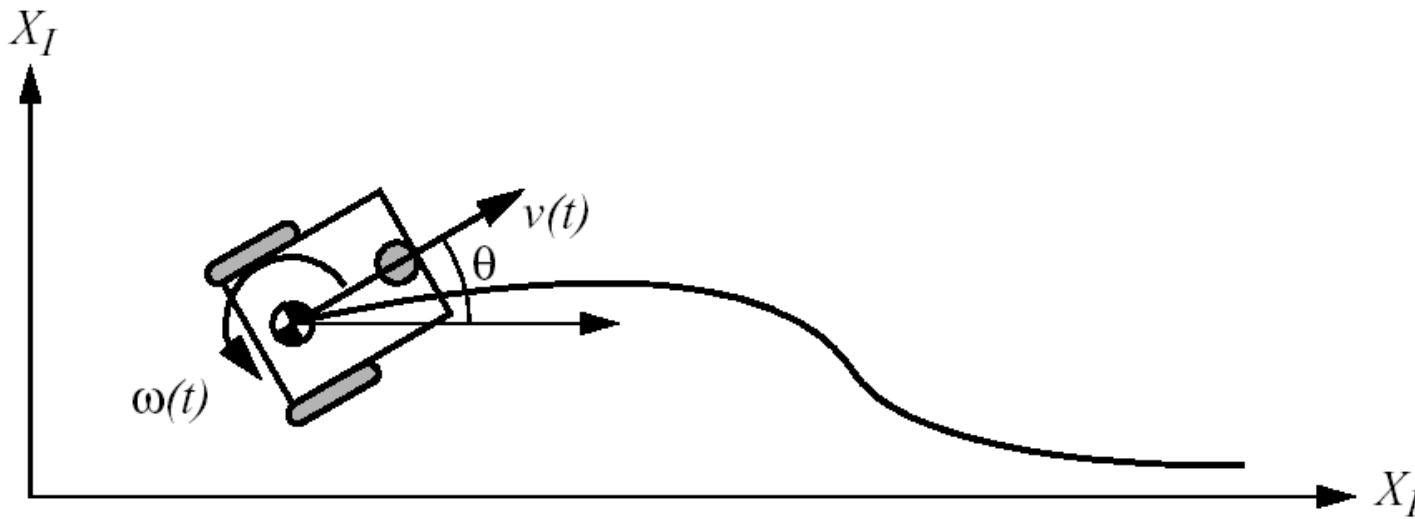
- ◇ 考虑沿 x 轴向前直线移动。移动 d 米后, y 分量的误差为 $d\sin\Delta\theta$, 当角度误差 $\Delta\theta$ 增大, 此分量会很大。



里程表法：差动驱动机器人 (1)

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

$$p' = p + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix}$$





里程表法：差动驱动机器人 (2)

◇ 运动学

$$\Delta x = \Delta s \cos(\theta + \Delta\theta/2)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta/2)$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2}$$

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$



里程表法：差动驱动机器人 (3)

$$\Sigma_{\Delta} = \text{covar}(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r |\Delta s_r| & 0 \\ 0 & k_l |\Delta s_l| \end{bmatrix}$$

$$\Sigma_{p'} = \nabla_p f \cdot \Sigma_p \cdot \nabla_p f^T + \nabla_{\Delta_{rl}} f \cdot \Sigma_{\Delta} \cdot \nabla_{\Delta_{rl}} f^T$$

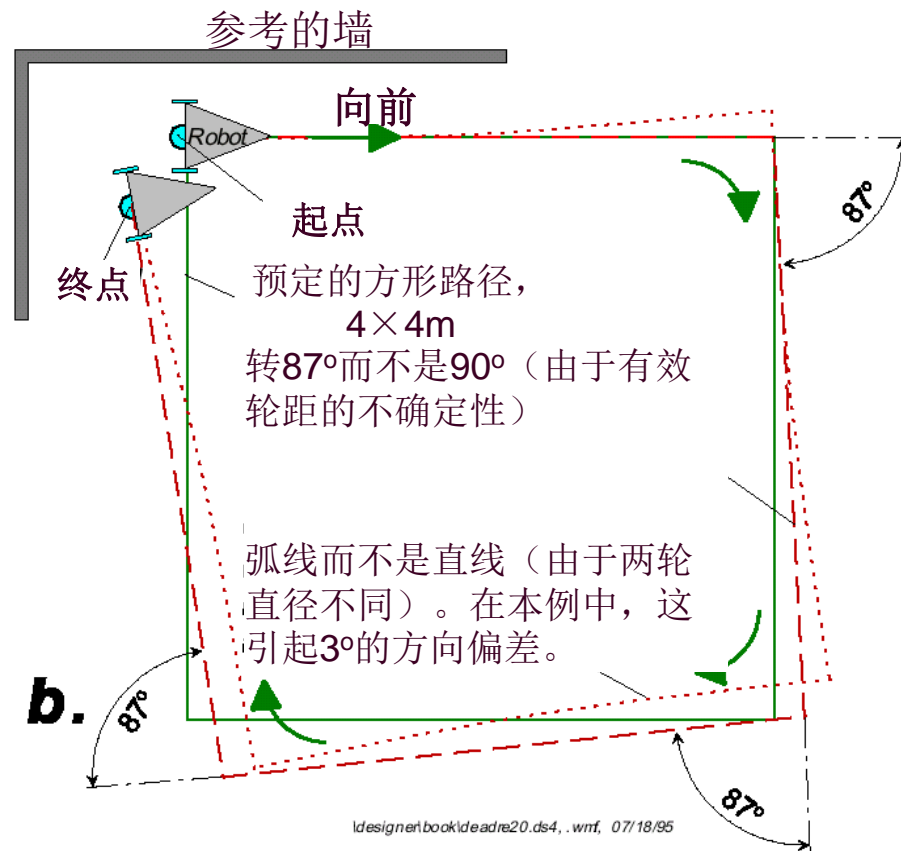
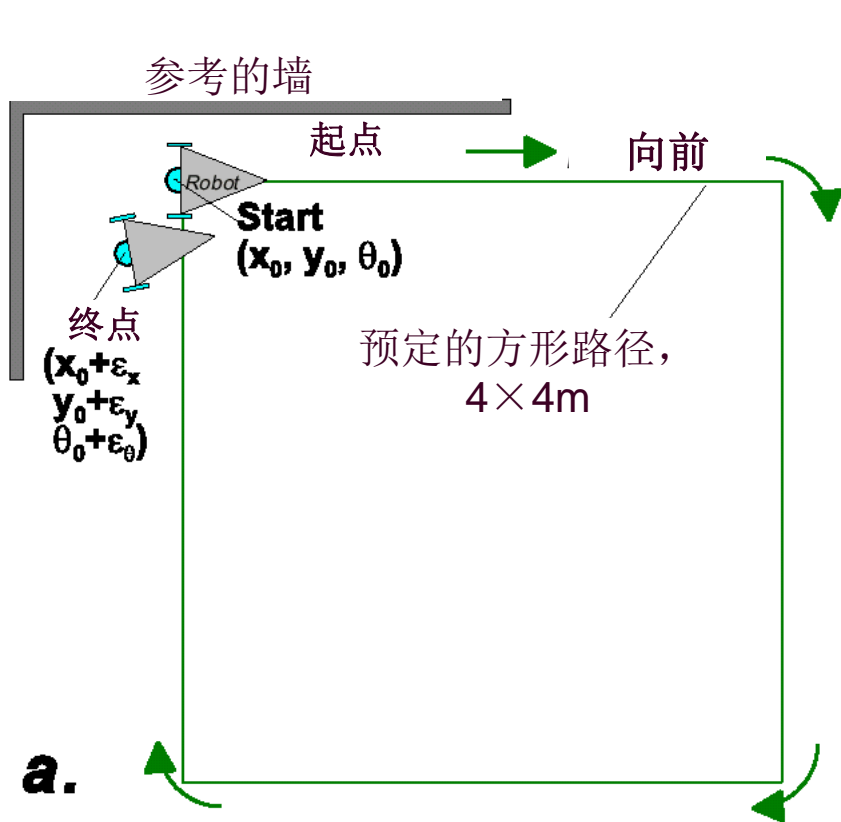
$$F_p = \nabla_p f = \nabla_p (f^T) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \Delta\theta/2) \\ 0 & 1 & \Delta s \cos(\theta + \Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix}$$

$$F_{\Delta_{rl}} = \begin{bmatrix} \frac{1}{2} \cos\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b} \sin\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2} \cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b} \sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{2} \sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b} \cos\left(\theta + \frac{\Delta\theta}{2}\right) & \frac{1}{2} \sin\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b} \cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}$$



里程表法：误差校正 I (Borenstein [5])

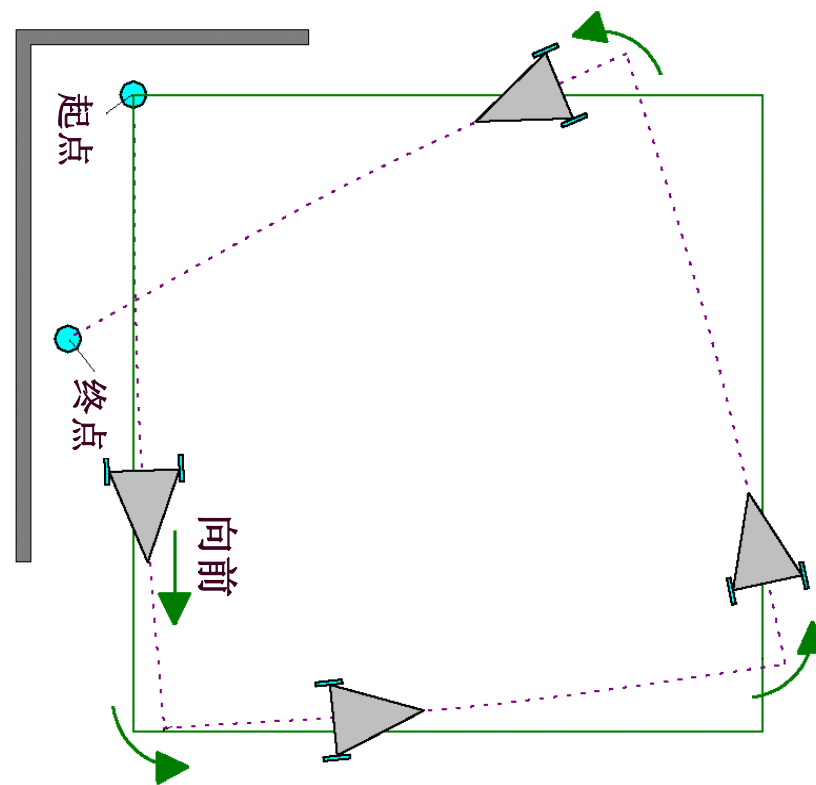
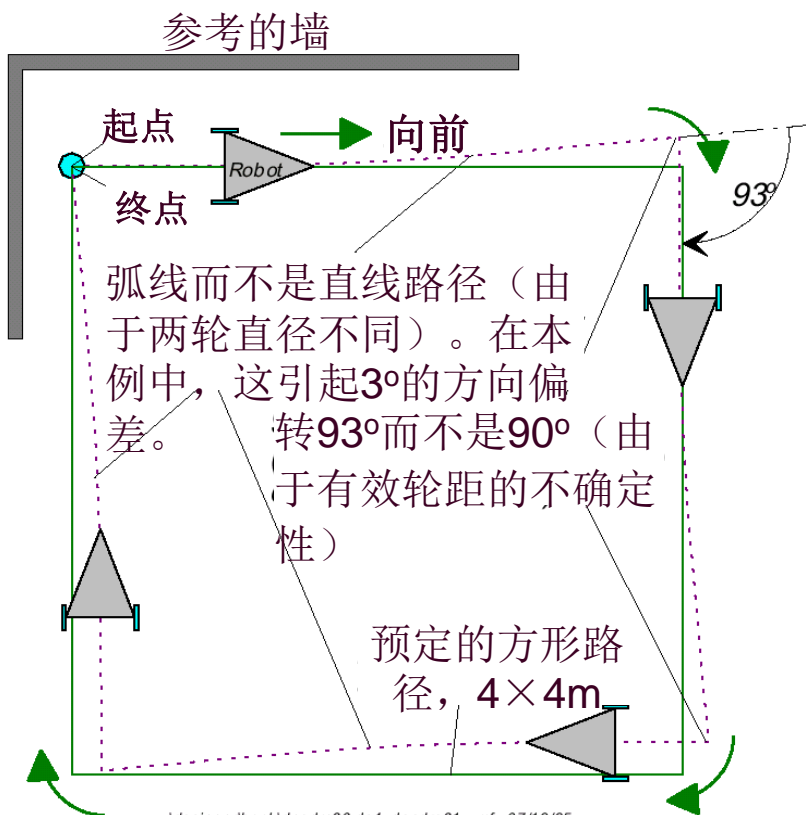
◇ 单向正方形路径实验





里程表法：误差校正 II (Borenstein [5])

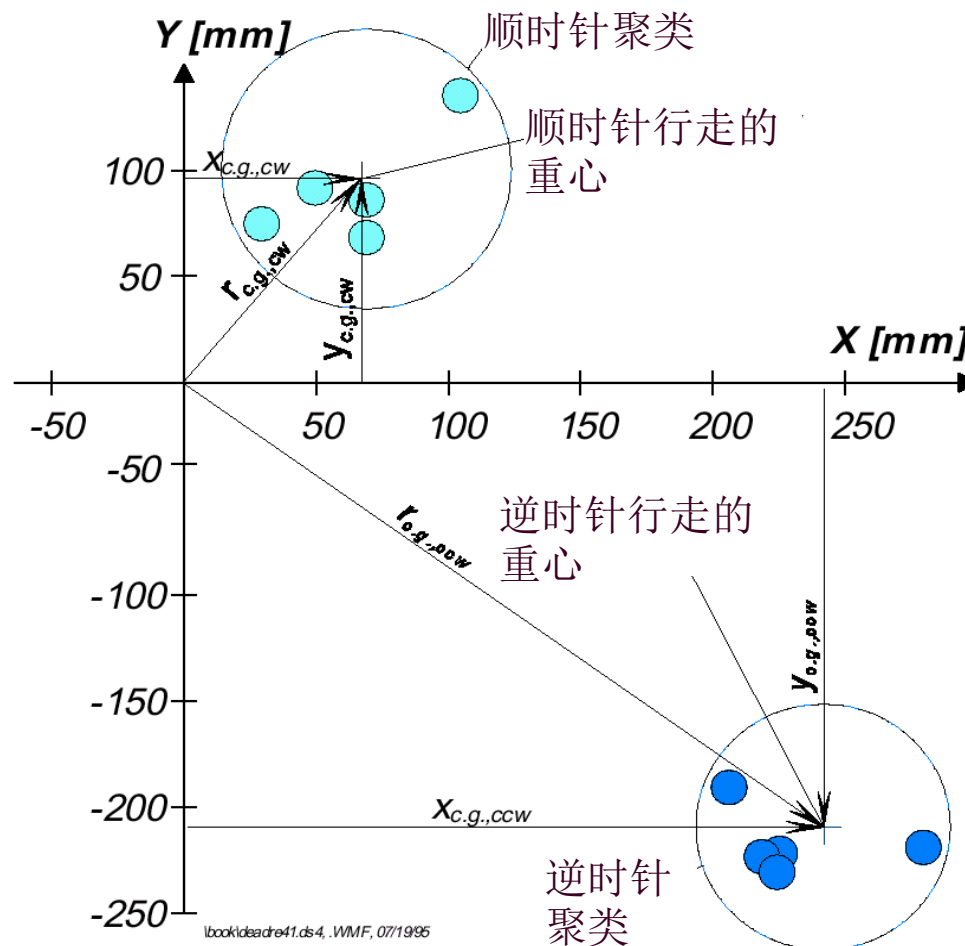
◇ 双向正方形路径实验





里程表法：误差校正 III (Borenstein [5])

◇ 确定性和非确定性误差





定位或不定位?

- 如何在 A 和 B 两个房间之间导航
 - 避障
 - 探测目标位置
- 总是沿着左（右）墙就可以
 - 但是，如何发现目标已经到达?



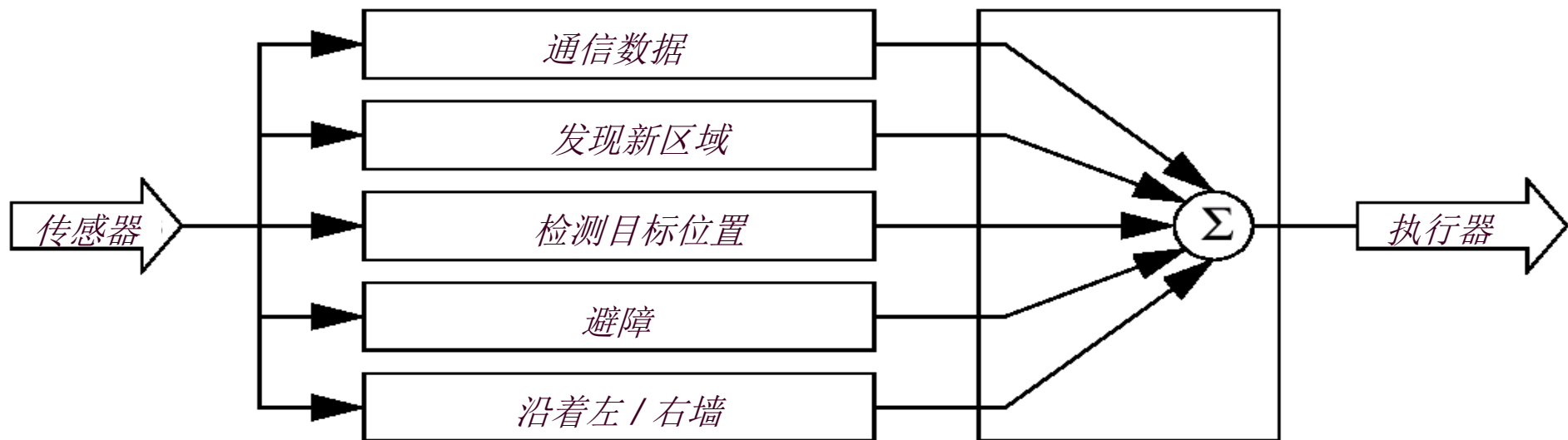


定位或不定位?

- 移动机器人是否一定需要定位系统？考虑人如何在自如地行走在房间之间的？
- 通过在地图上定位，到达目标时再检测确认（必须的，否则会有偏差）。
- 另一种方法：基于行为。取消建立地图、定位、路径规划的步骤，原因是传感器和执行器有噪声，且信息是有限的（不那么可靠的）。
 - 如要行走到一个房间，基于行为的导航方法是：设计一个左墙壁的跟踪行为和房间 B 的检测器。
 - 该检测器由某独特特征检测所触发，如地毯颜色改变；
 - 沿着左墙壁走，直到检测到房间 B，结束这一段程序。



基于行为的导航



协调 / 融合
如，通过向量求和的融合



基于行为的导航

- 基于行为的导航方法的特点

- 在具有少数目标位置的独立环境（保证环境特征的惟一性）中容易实现，但难以推广到较大环境中；
- 基本行为程序，如“跟踪左墙壁”，必须仔细设计以产生期望的行为——比较耗时，对机器人硬件和环境依赖严重（如左墙不平整）；
- 在任一时刻，基于行为的系统需要有多个主动行为，即便单个行为被调整到最优性能，但多个行为之间融和和切换后会失去性能的最优性（甚至会变得很糟，不稳定）。添加新行为必须非常仔细，以保证新引入行为和所有已有行为的结合是稳定的。

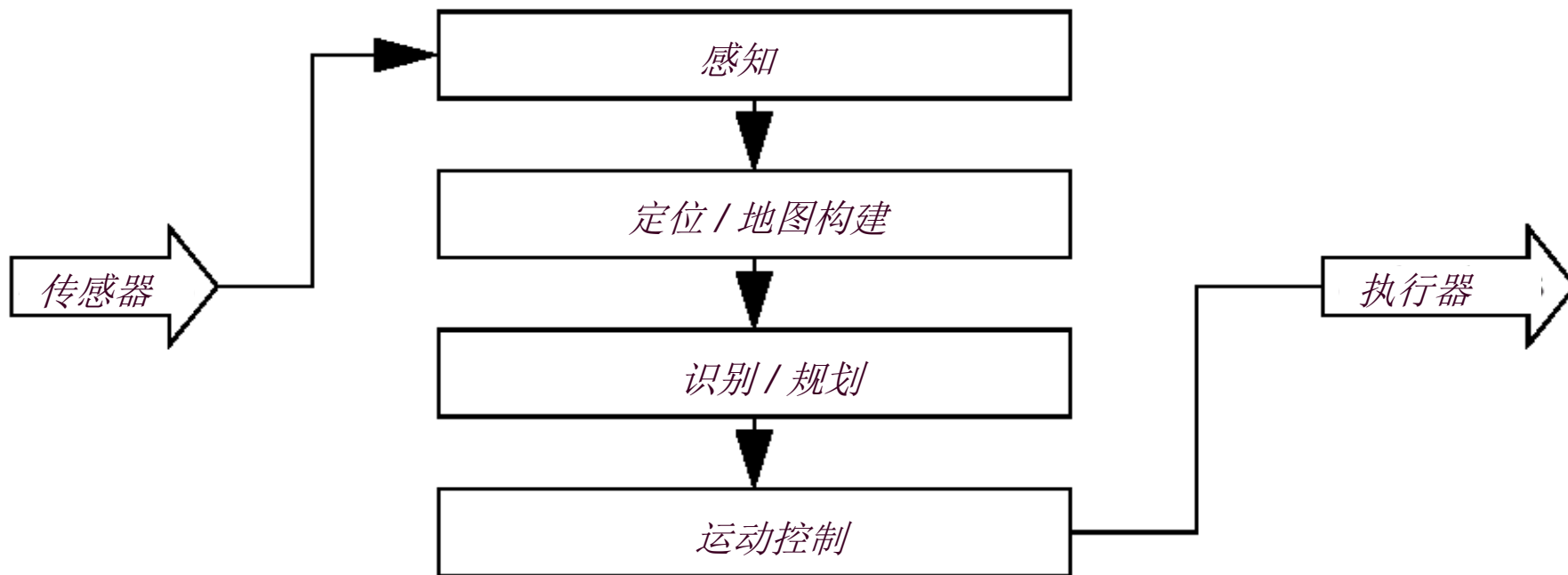


基于模型的导航

- 基于模型的导航包括定位和认知两个模块
- 机器人明确地力图通过收集传感器的数据，随后根据环境地图对它的所在位置和信任度进行更新，从而实现定位。
- 主要特点：
 - 得到基于地图的、显式的位置概念，对人类操作员直接可用；
 - 地图本身代表了人与机器人之间交流的一个媒介，人可以将新环境的地图直接输入给机器人；
 - 如果机器人具有自行建立地图的功能，所建立的地图人可以利用。
- 对地图依赖的风险——如果模型偏离实际？



基于模型的导航





基于模型的导航

- 基于地图的定位系统，其最基本的问题是“表示”——环境的表示方式
 - 地图中包含环境哪些方面？
 - 地图以何种精确度表示环境？
- 机器人还必须有一个关于在地图上表示它的位置信任度的方法
 - 机器人是辨识 一个特定位置 作为它当前位置，还是用 一组可能的位置 描述它的当前位置？
 - 如果用一个信任度完全表达多个可能位置，如何对这些位置排序？
- 信任度表示：
 - 单假设信任度：机器人假定其惟一位置
 - 多假设信任度：机器人描述它位置的不确定度



位置假设信任度的表示

◆ 连续的

- ◆ 精度受传感器数据限制
- ◆ 典型的，单假设位姿估计
- ◆ 如果发散则失败 (对于单假设)
- ◆ 表示形式紧凑；典型情况下，处理能力适中。

◆ 离散的

- ◆ 精度受离散化时的分辨率限制
- ◆ 典型的为：多假设姿态估计
- ◆ 从不失败 (当发散时，收敛到另一个单元)
- ◆ 对存储器和处理能力要求高 (拓扑地图除外)



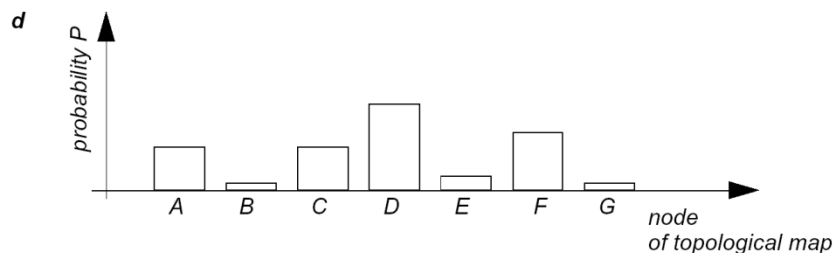
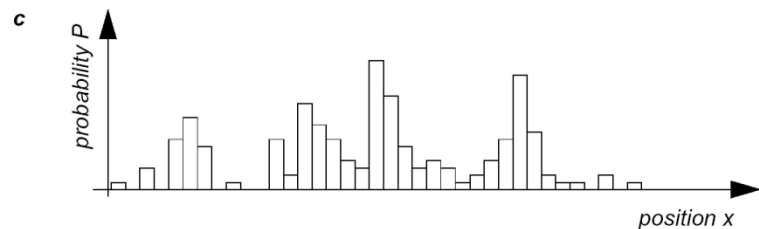
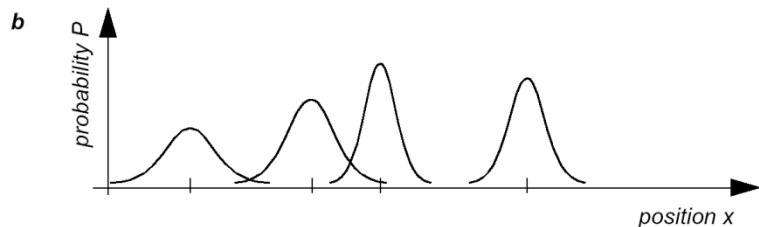
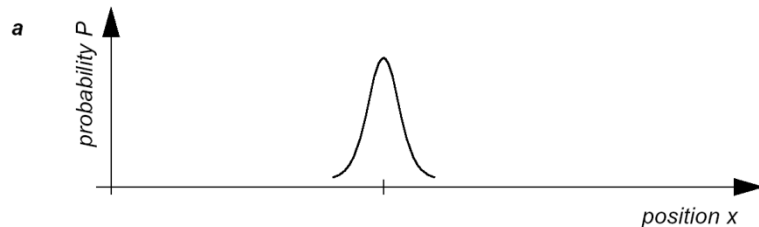
信任度的表示

◇ a) 连续型地图
具有单假设

◇ b) 连续型地图
具有多假设

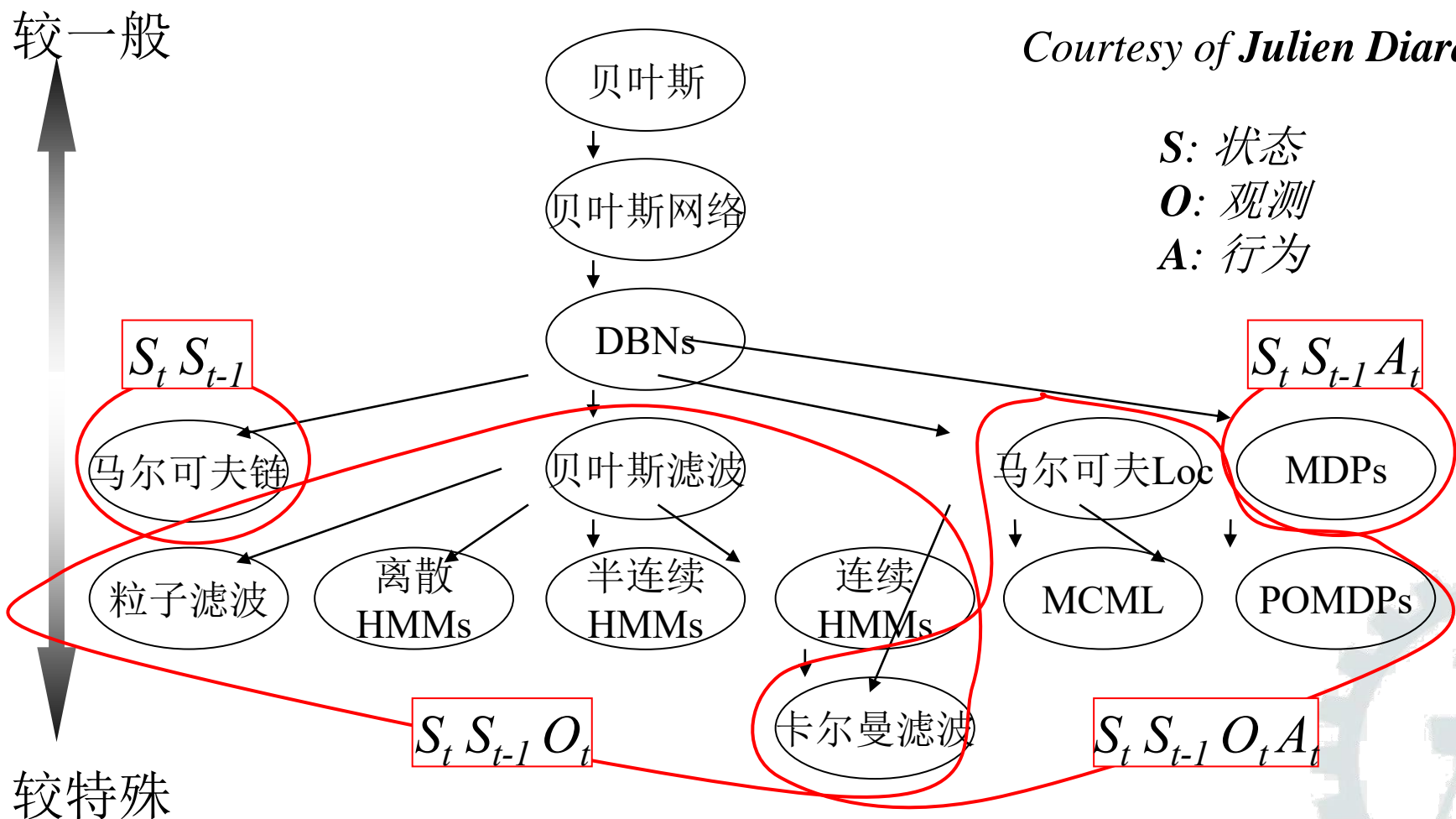
◇ c) 离散（栅格）地图
具有概率离散

◇ d) 离散（栅格）拓扑
具有概率离散



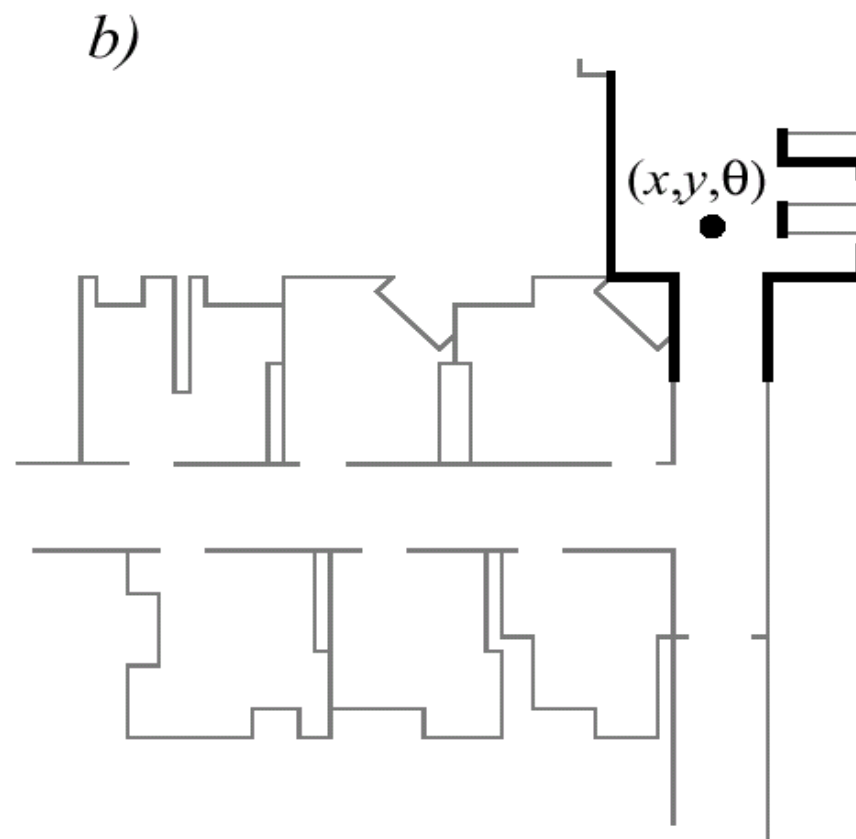
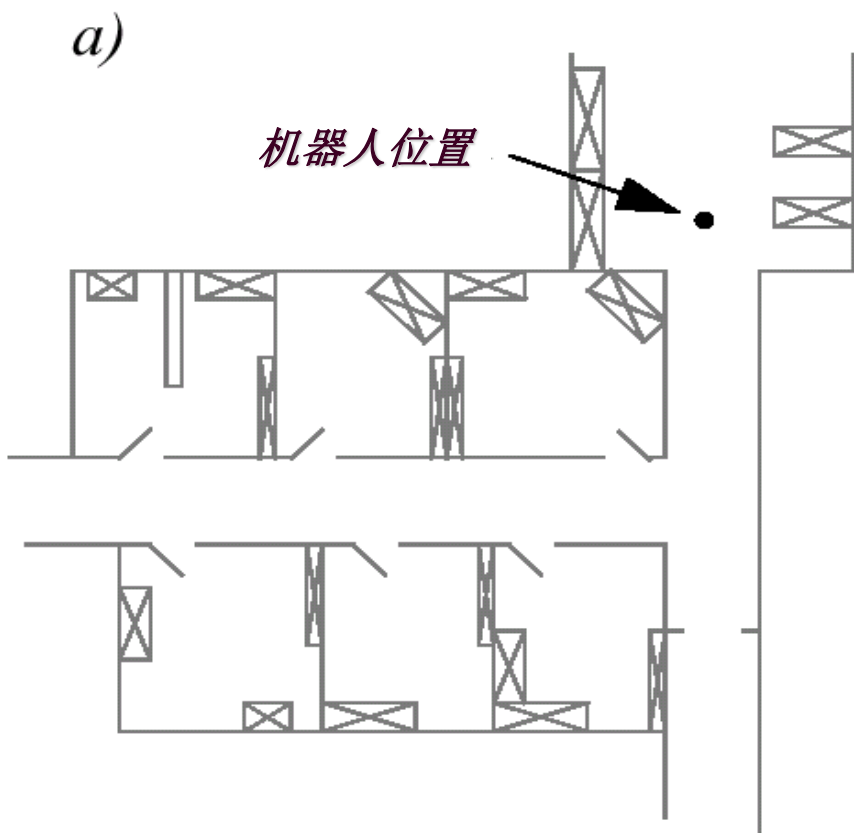


贝叶斯方法：：一种随机模型分类学





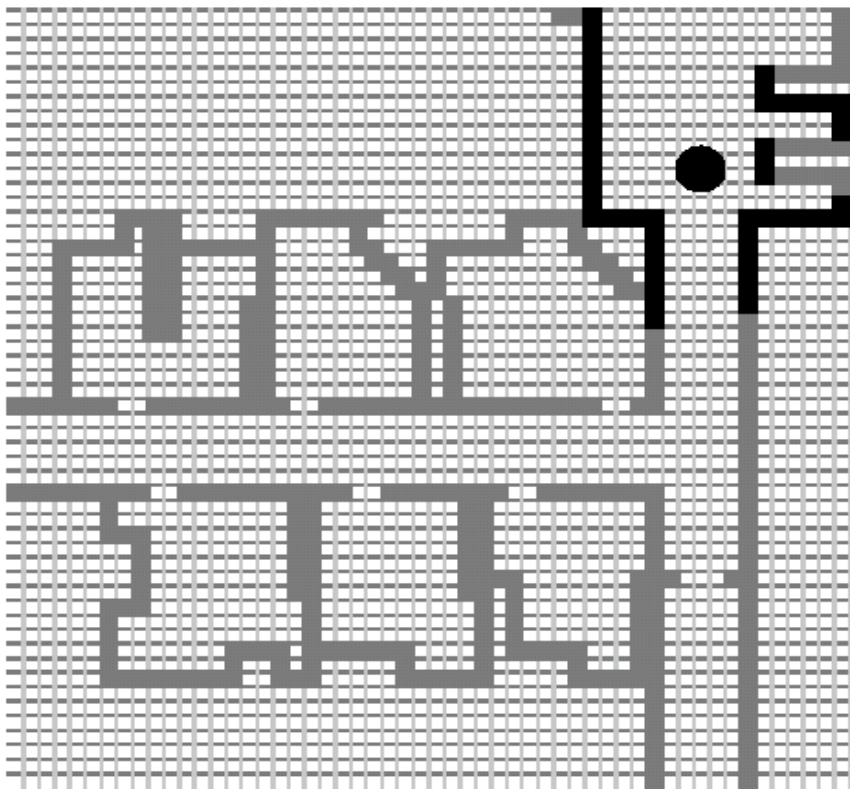
单假设信任度 – 连续线地图



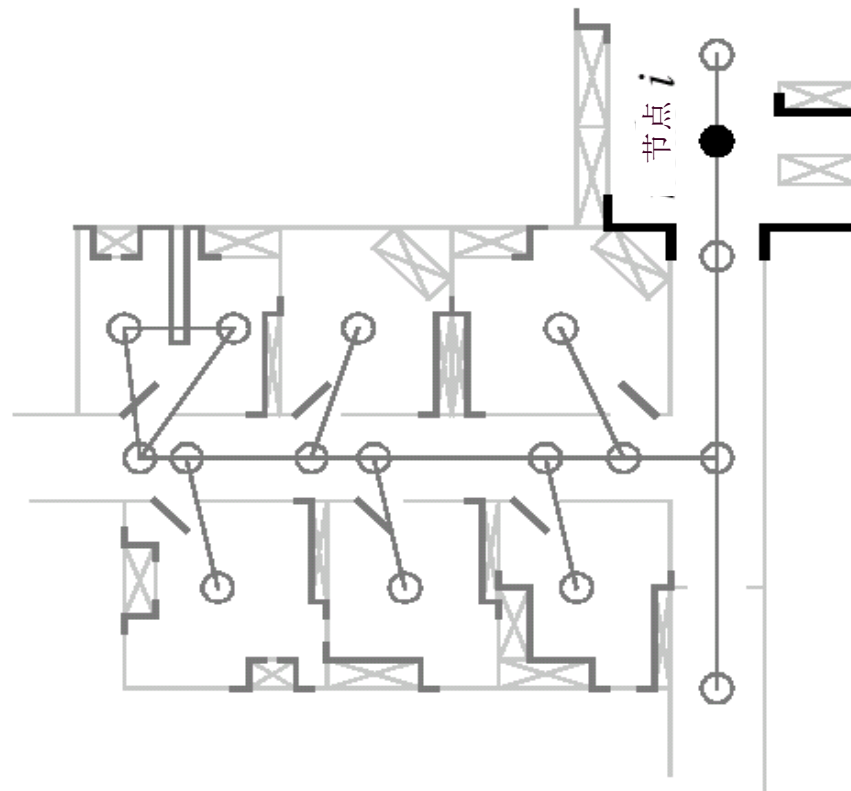


单假设信任度 – 栅格和拓扑地图

c)



d)

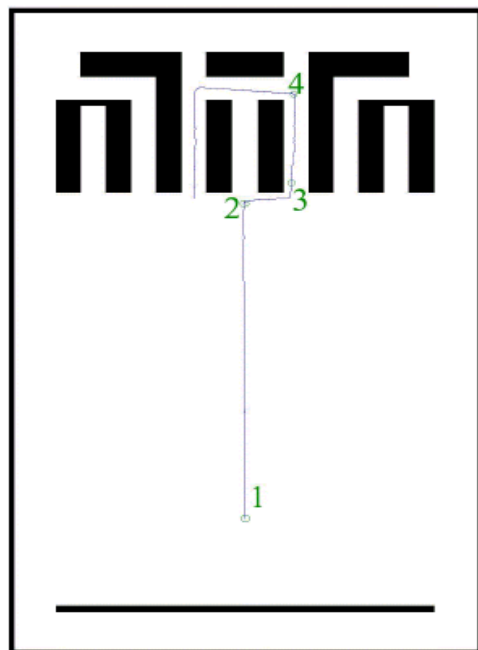




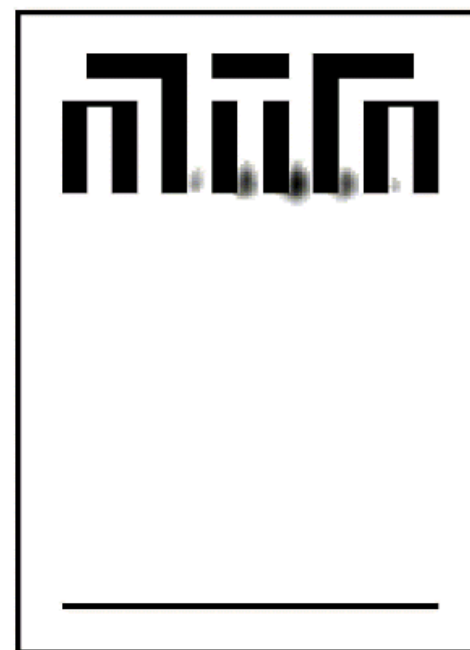
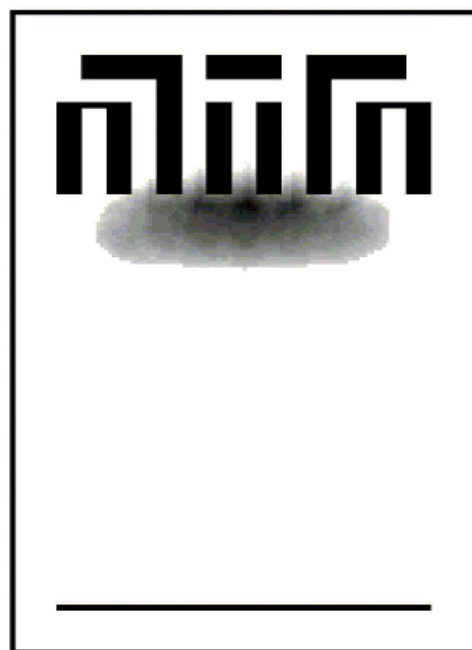
基于栅格的表示法 – 多假设

◇ 栅格大小约为 20 cm^2 .

Courtesy of W. Burgard



机器人路径



在位置2, 3, 4的信任度状态



地图表示方法

1. 地图精度 vs. 应用
2. 特征的精度 vs. 地图精度
3. 精度 vs. 计算复杂性

◇ 连续型表示方法

◇ 分解 (离散化)



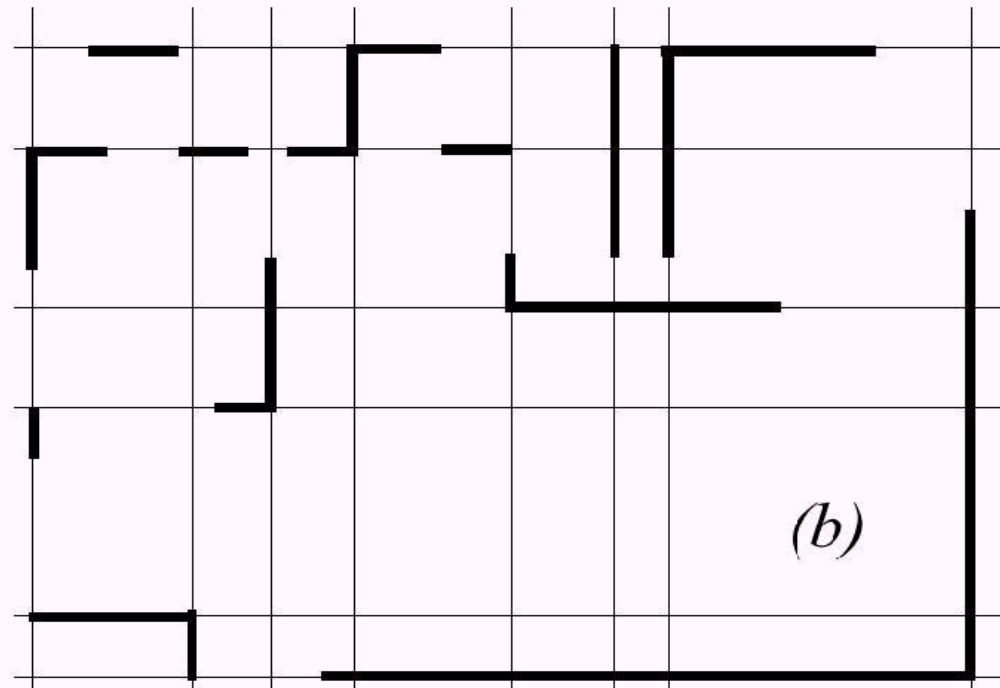
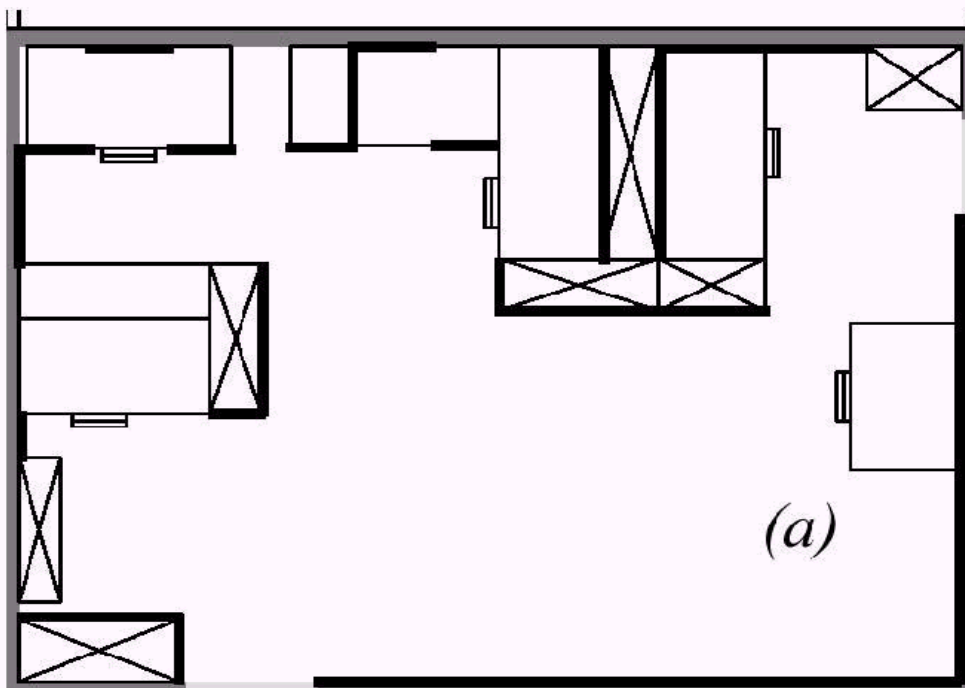
环境的表示方法

- 环境表示
 - 连续度量 -> x, y, θ
 - 离散度量 -> 度量栅格
 - 离散拓扑 -> 拓扑栅格
- 环境建模
 - 原始感知数据, 如: 激光扫描距离数据, 图像灰度
 - 数据量大, 独特性小
 - 利用所有获得的信息
 - 低级特征, 如: 直线或其它几何特征
 - 数据量中等, 独特性一般
 - 过滤得到有用信息, 但仍然含糊
 - 高级特征, 如: 门, 一辆车, 埃菲尔铁塔
 - 数据量小, 高独特性
 - 过滤得到有用信息, 很少或没有含糊性, 信息不充分



地图表示方法：基于连续线

- a) 结构图
- b) 用一组无穷多线的表示

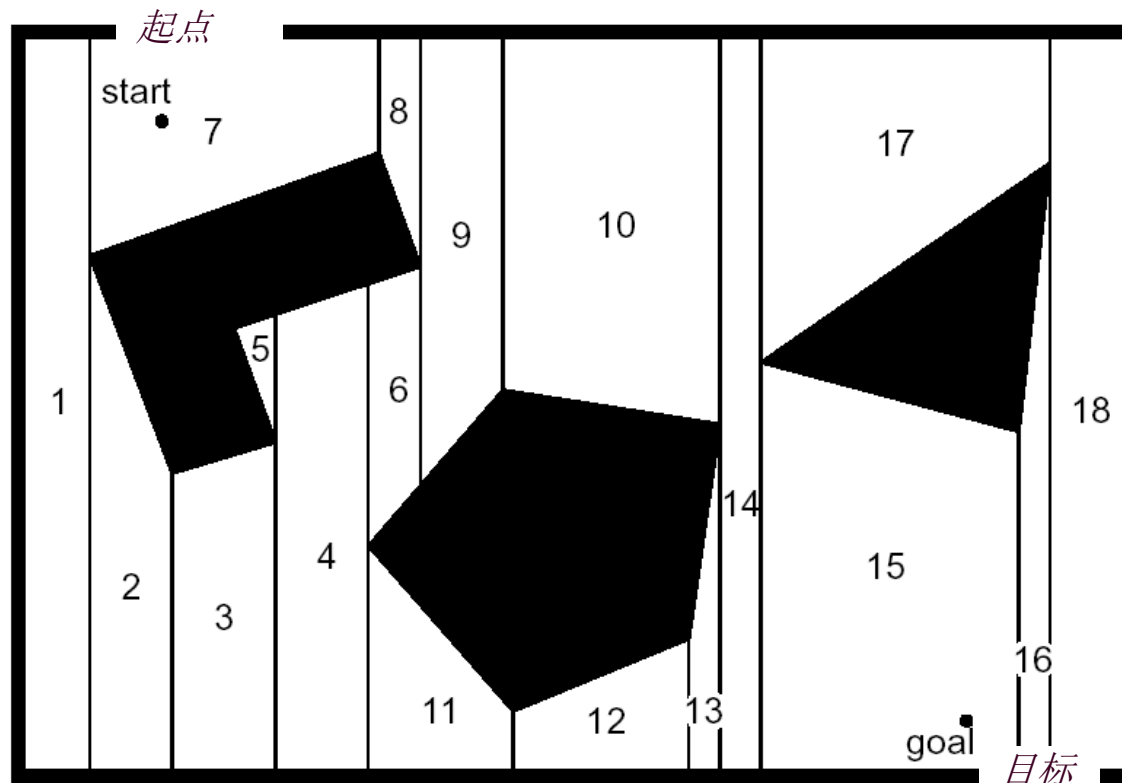




地图表示方法：分解 (1)

◇ 精确单元分解（多边形障碍物）

◇ 基本假定：在各自由控制区内，具体位置是无关紧要的！

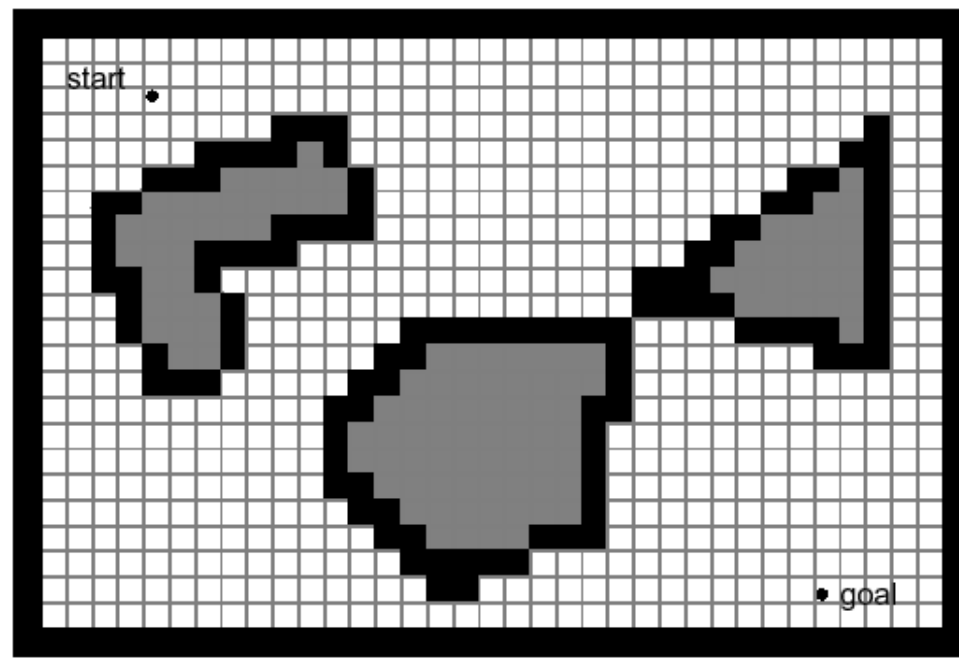
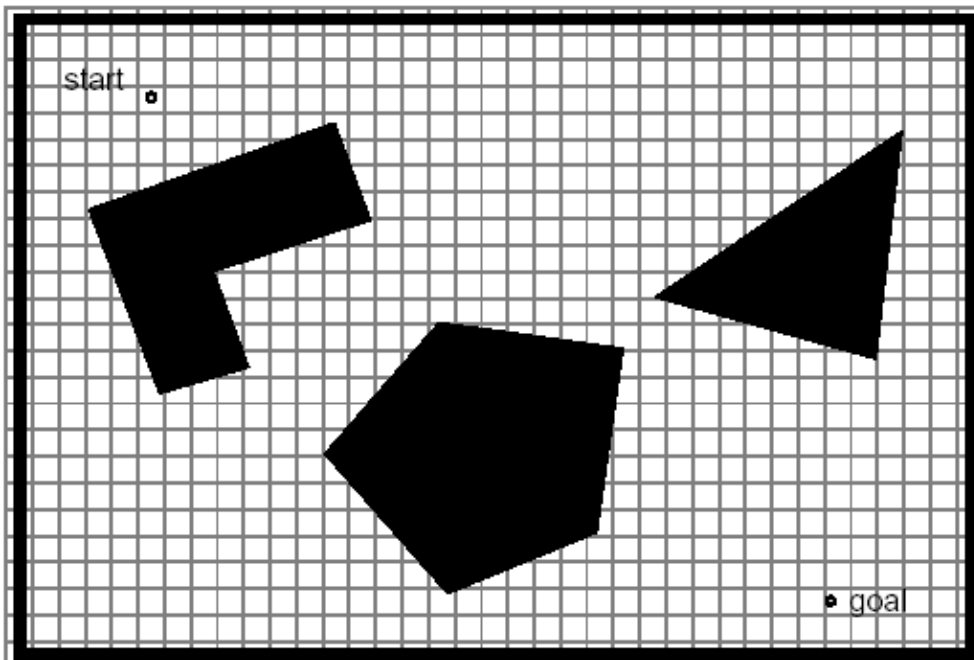




地图表示方法：分解 (2)

◇ 固定单元分解

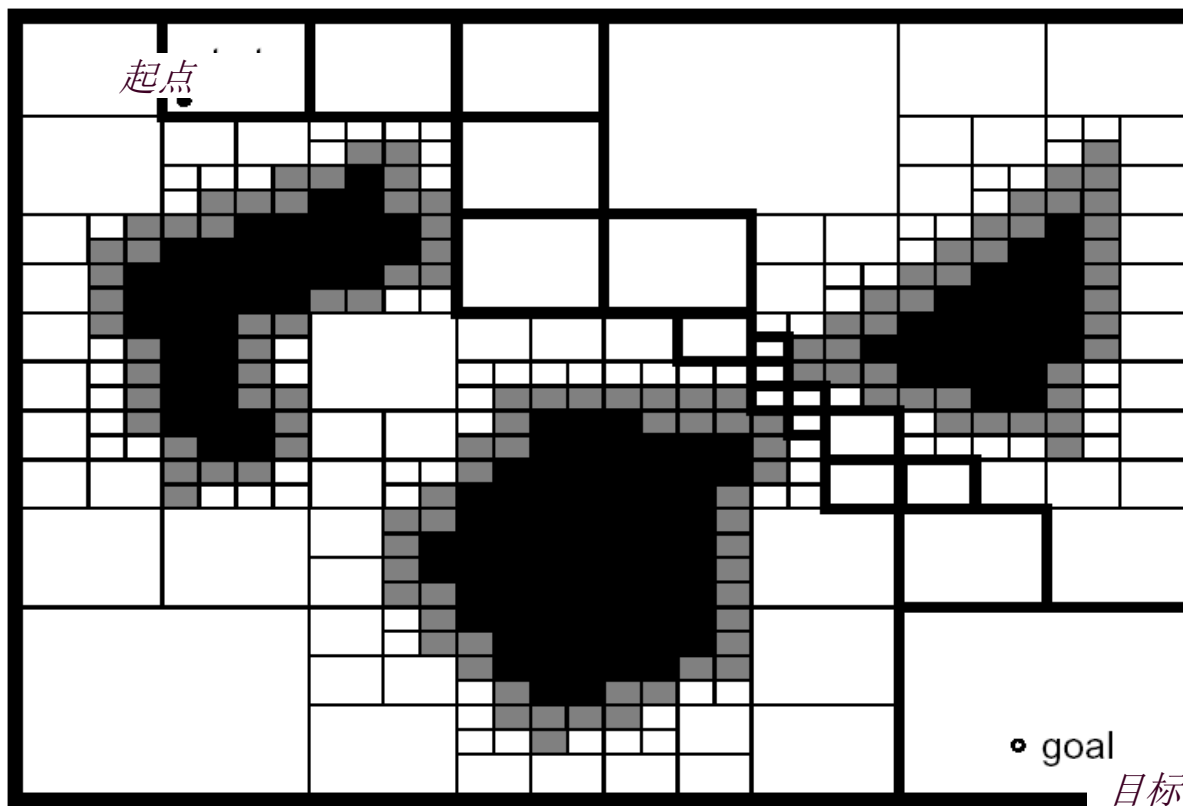
◇ 狭窄的通道消失了





地图表示方法：分解 (3)

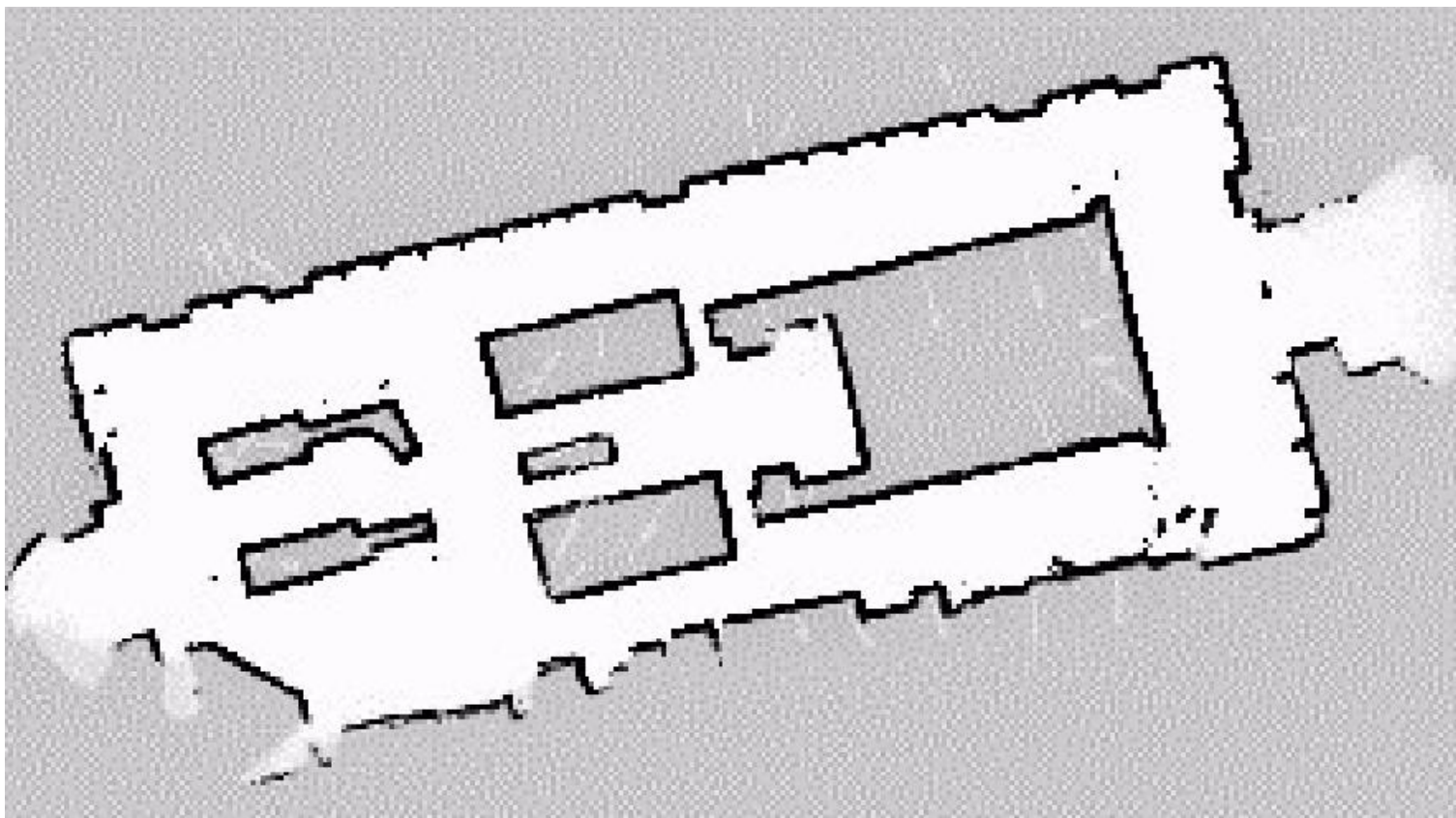
◇ 自适应单元分解





地图表示方法：分解 (4)

- ◇ 固定单元分解 – 非常小单元的例

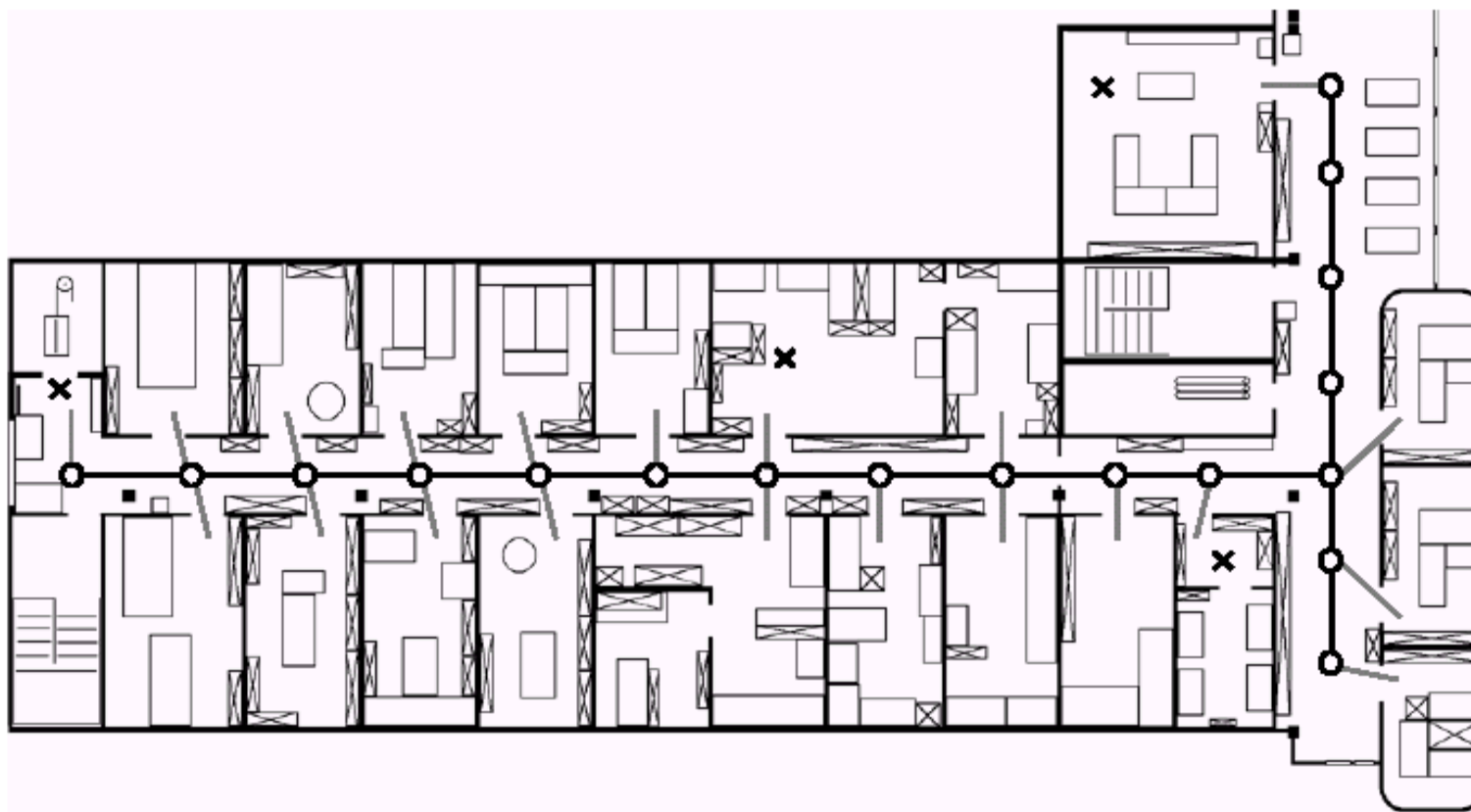


Courtesy of S. Thrun



地图表示方法：分解 (5)

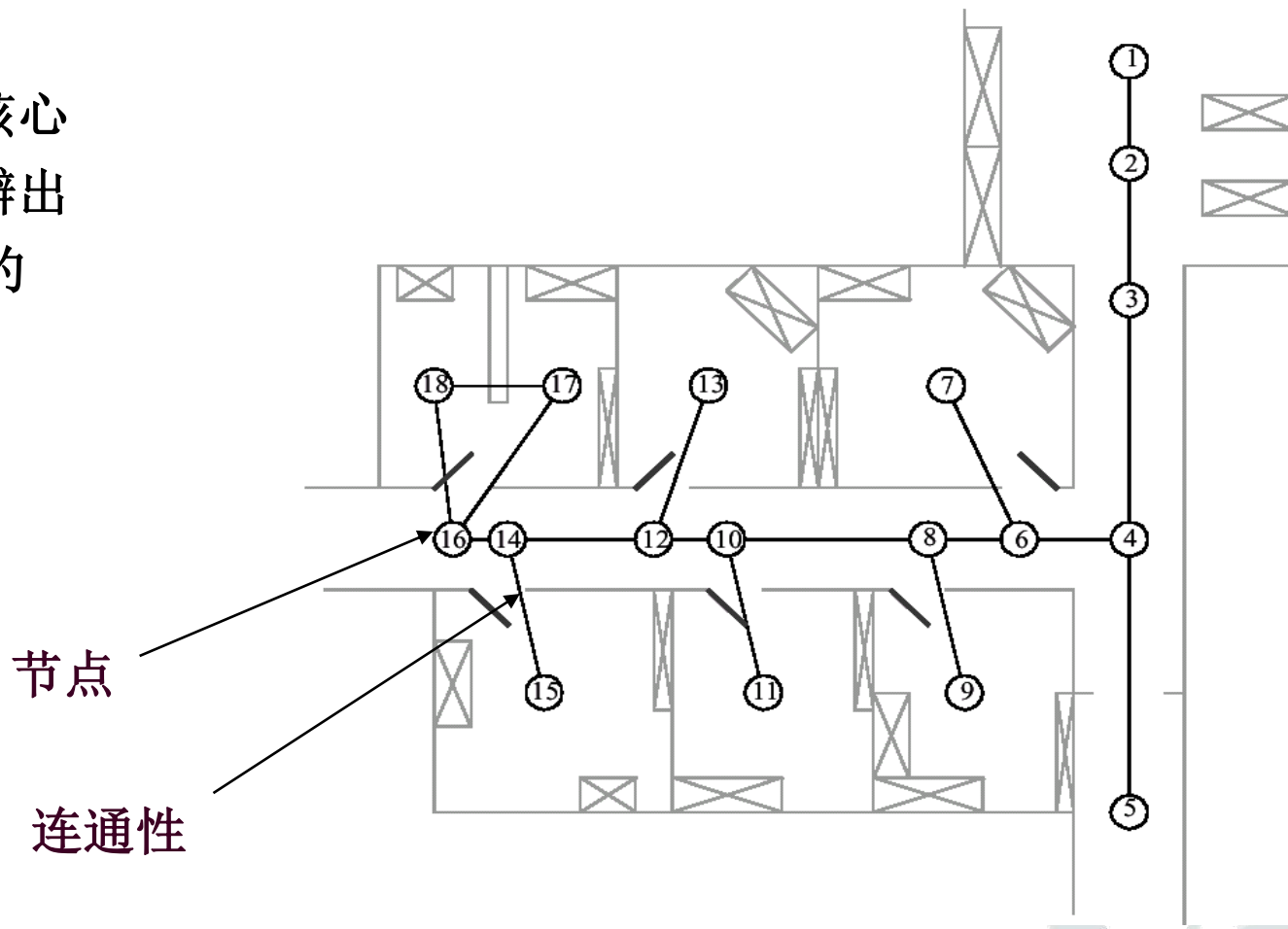
- 拓扑分解





地图表示方法：分解 (6)

- ◇ 拓扑分解
- ◇ 邻接是拓朴方法的核心
- ◇ 假定机器人可以分辨出每个节点（走廊与门的交点、走廊间的交点）





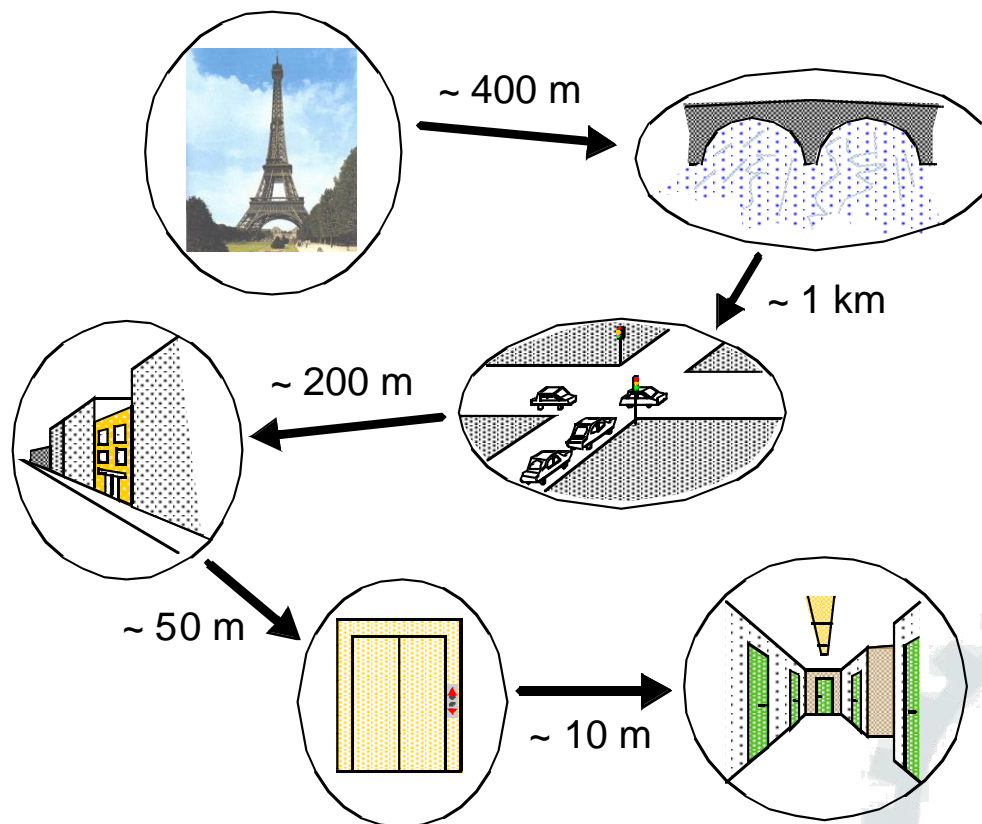
地图表示方法：分解 (7)

◆ 拓扑分解

◆ 机器人必须具备：

- ◆ 检测当前所处节点的能力
- ◆ 在节点间行走的能力

◆ 安防辅助路标





发展现状：地图表示方法的最新挑战

- ◇ 真实世界是动态的（修正地图时不易区分永久性和临时性障碍）
- ◇ 感知仍然是主要挑战
 - ◇ “误差趋势”
 - ◇ 提取有用信息的困难
- ◇ 开放空间的穿越（缺乏可“见”物体）
- ◇ 如何构建拓扑结构（节点的边界，不易感知）
- ◇ 传感器融合
- ◇ ...



概率性的，基于地图的定位 (1)

◇ 基于地图定位的基本描述：

- ◇ 考虑一个在已知环境中运行的移动机器人。
- ◇ 假如它从一个精确已知的位置开始移动，它可能会使用“里程表法”保持它的位置在要求的路径上。
- ◇ 然而，移动一定的距离后，机器人对它的位置非常不确定了。
→ 根据对其环境的观测做位置更新。
- ◇ 观测也可导致对机器人位置的一个估计，这个估计可以与里程表法估计结果融合，得到对最可能的机器人实际位置的更新。



概率性的，基于地图的定位 (2)

◇ 动作更新

◇ 动作模型 ACT (开环推算)

$$s'_t = Act(o_t, s_{t-1})$$

其中， o_t : 编码器测量， s_{t-1} : 先验信任度状态

◇ 增加了不确定性

◇ 感知更新

◇ 感知模型 SEE

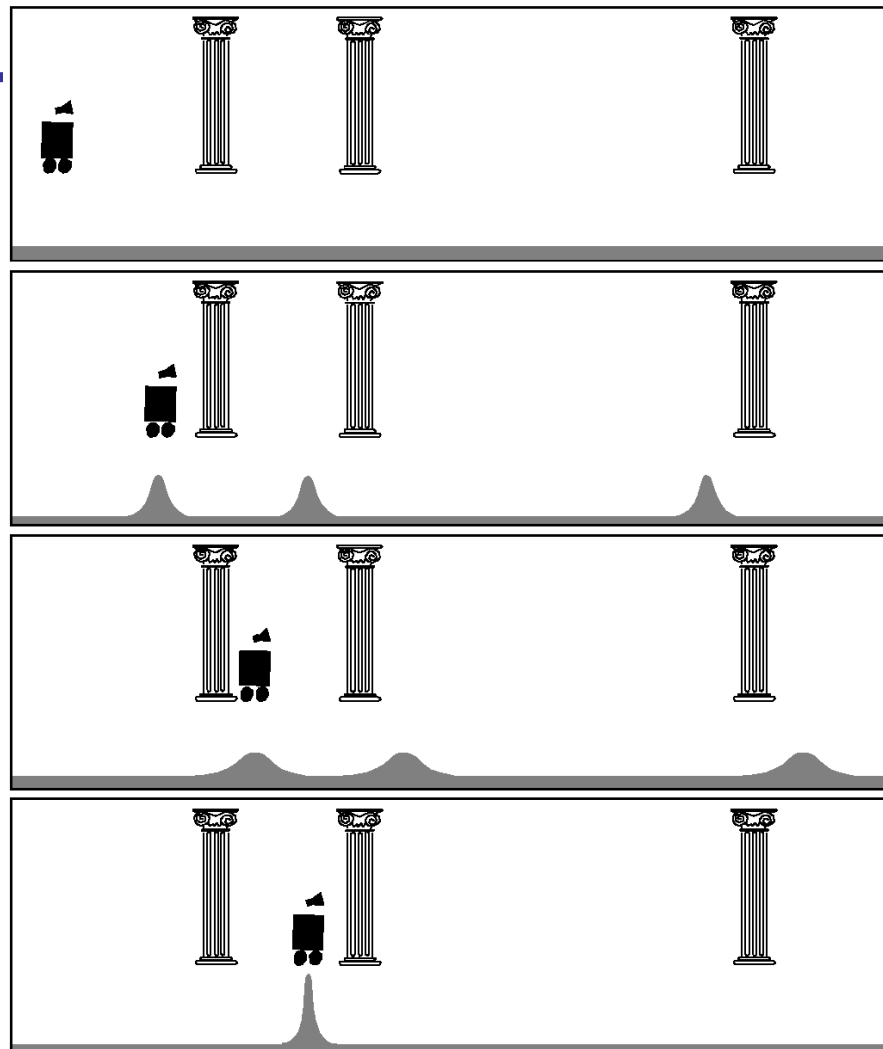
$$s_t = See(i_t, s'_t)$$

其中， i_t : 外感受传感器输入， s'_t : 更新的信任度状态

◇ 可降低不确定性



◇ 借助运动改善信任度





概率性的，基于地图的定位 (3)

◇ 给定

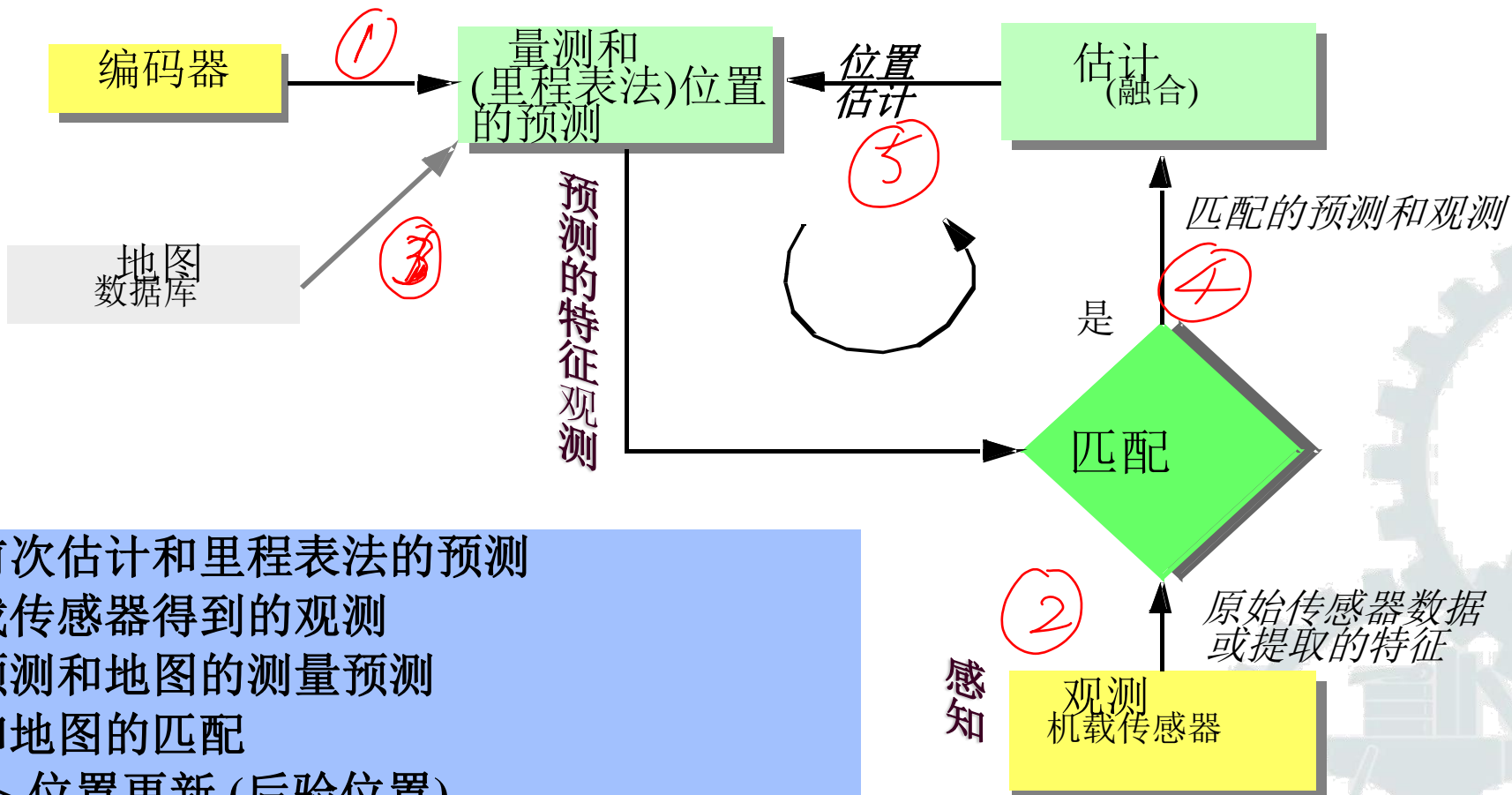
- ◇ $k-1$ 时刻的位置估计 $p(k-1)$
- ◇ 该估计的协方差矩阵 $\Sigma_p(k-1)$
- ◇ k 时刻的控制输入 $u(k)$
- ◇ k 时刻的观测 $Z(k)$
- ◇ k 时刻的地图 $M(k)$

◇ 计算

- ◇ k 时刻位置（后验位置）估计 $p(k|k-1)$
- ◇ 该估计的协方差矩阵 $\Sigma_p(k|k-1)$



基于地图定位的五个步骤



1. 基于前次估计和里程表法的预测
2. 由机载传感器得到的观测
3. 基于预测和地图的测量预测
4. 观测和地图的匹配
5. 估计 -> 位置更新 (后验位置)



马尔可夫 \Leftrightarrow 卡尔曼滤波 定位

◇ 马尔可夫定位

- ◇ 从任意未知点出发的定位（与里程表法差别）
- ◇ 恢复不明确的状态
- ◇ 然而，要在任何时刻、在整个状态空间内更新所有位置的概率，就需要空间的离散表示（栅格）。于是所需要的存储器和计算能力可能成为限制使用精确栅格的重要因素。

◇ 卡尔曼滤波定位

- ◇ 从已知起始位置开始跟踪机器人，具有固有的高度精确性和有效性。
- ◇ 然而，如果机器人的不确定性变得很大（例如，由于机器人与物体碰撞），卡尔曼滤波将失败，位置必然丢失。



马尔可夫定位 (1)

- ◇ 马尔可夫定位，采用显式的离散形式，表示状态空间中所有位置的概率。
- ◇ 通常，用可能状态（位置）为有限多个的栅格或拓扑图表示环境。
- ◇ 每次更新时，整个空间中每个状态（单元）的概率都得到更新，是对概率分布的更新。



马尔可夫定位 (2): 概率论应用于机器人定位

◇ **P(A): A 是真的概率**

◇ 例如, $p(r_t = l)$: t 时刻机器人 r 在位置 l 的概率

◇ 我们希望当给出动作和传感器测量时, 计算机器人各不同位置的概
率。

◇ **P(A/B): 已知 B 时, A 的条件概率。**

◇ 例如, $p(r_t = l | i_t)$: 给出传感器输入 i_t 时, t 时刻机器人 r 在位置 l 的条件概
率。

◇ 乘法规则:

$$p(A \wedge B) = p(A|B)p(B)$$

$$p(A \wedge B) = p(B|A)p(A)$$



马尔可夫定位 (2): 概率论应用于机器人定位

$$\begin{bmatrix} p_{t-1}(1) \\ p_{t-1}(2) \\ p_{t-1}(3) \\ \vdots \\ p_{t-1}(L) \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} p(r_{t-1}=l_1) \\ p(r_{t-1}=l_2) \\ p(r_{t-1}=l_3) \\ \vdots \\ p(r_{t-1}=l_L) \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.35 \\ \vdots \\ 0.02 \end{bmatrix} \begin{array}{c} \text{█} \\ \text{█} \\ \text{█} \\ \vdots \\ \text{█} \end{array}$$

$$\begin{bmatrix} p_t(1|o_t) \\ p_t(2|o_t) \\ p_t(3|o_t) \\ \vdots \\ p_t(L|o_t) \end{bmatrix} = \begin{bmatrix} \sum_l p(1|l, o_t) p_{t-1}(l) \\ \sum_l p(2|l, o_t) p_{t-1}(l) \\ \sum_l p(3|l, o_t) p_{t-1}(l) \\ \vdots \\ \sum_l p(L|l, o_t) p_{t-1}(l) \end{bmatrix}$$

- ◇ $p(l'|l, o_t)$ 表示当 $t-1$ 时刻状态为 l ，当前的控制是 o_t ， t 时刻状态 l' 的概率，相当于动作模型 ACT: $s'_t = Act(o_t, s_{t-1})$
- ◇ 依赖上次状态的概率分布和控制，得到当前状态的验前概率分布



马尔可夫定位 (3) : 概率论应用于机器人定位

◇ 贝叶斯规则:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

◇ 从一个信任状态和传感器输入 i , 到精确的信任状态(SEE)的映射:

$$p(l|i) = \frac{p(i|l)p(l)}{p(i)} \quad p = p_t$$

◇ $p(l)$: 在“感知修正过程”之前的信任状态的概率, 即 $p_t(l|o_t)$

◇ $p(i|l)$: 在位置 l 时, 得到测量 i 的概率 (所谓“传感器输入”)

◇ 查阅地图, 对地图中各种可能位置, 鉴别某个传感器读数的概率。

◇ $p(i)$: 归一化因子, 对所有 i 其和=1。 (实为得到测量为 i 的概率)



马尔可夫定位 (3) : 概率论应用于机器人定位

- ◇ 从上时刻信任位置和一个动作（即编码器测距）得到信任位置 (ACT):

$$p_t(l|o_t) = \int p(l_t|l'_{t-1}, o_t) p(l'_{t-1}) dl'_{t-1}$$

$$p_t(l|o_t) = \sum_{l'} p(l|l', o_t) p_{t-1}(l'), \quad l = 1, \dots, L$$

- ◇ 再根据感知 i ，得到感知模型 (SEE):

$$p_t(l) \stackrel{\Delta}{=} p_t(l|o_t, i) = \frac{p(i|l) p_t(l|o_t)}{p(i)}, \quad l = 1, \dots, L$$

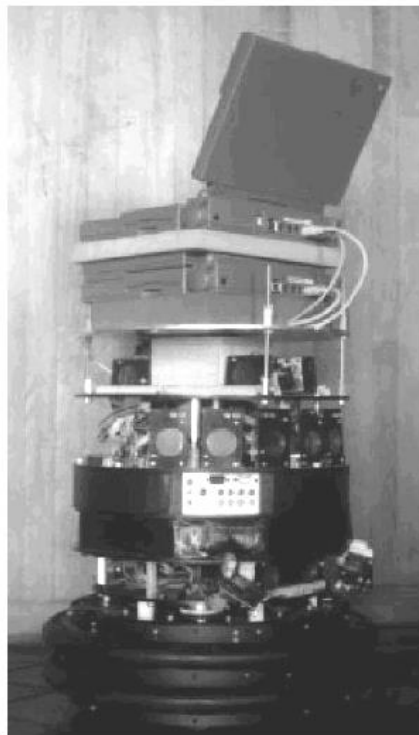
- ◇ 马尔可夫假设：更新仅仅依赖前一时刻的状态以及最近时刻的动作和感知。
- ◇ 信任状态：

$$p_t(l) = p(i|l) \sum_{l'} p(l|l', o_t) p_{t-1}(l')$$



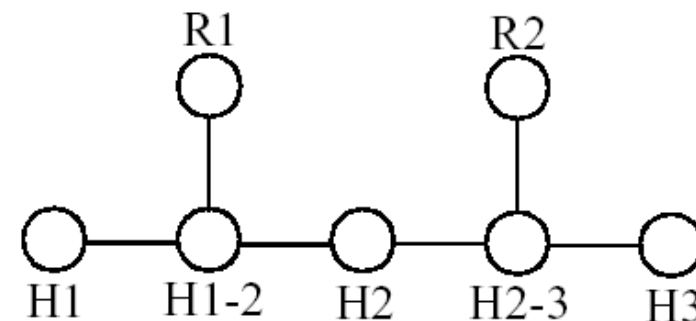
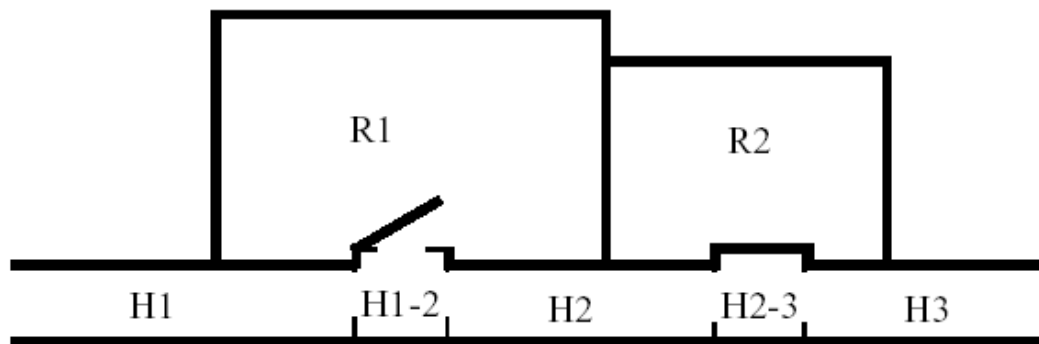
马尔可夫定位：实例研究 1 – 拓扑地图

- ◇ Dervish 机器人
- ◇ 使用声纳的拓扑定位





马尔可夫定位：实例研究 1 – 拓扑地图



	墙	门关	门开	开放型走廊	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50



马尔可夫定位：实例研究 1 – 拓扑地图

◇ 给定感知对 i ，针对位置 n 更新信任状态：

$$p(n|i) = p(i|n)p(n)$$

- ◇ $p(n|i)$: 在位置 n 的新信任状态
- ◇ $p(n)$: 当前信任状态
- ◇ $p(i|n)$: 在位置 n 看到 i 的概率 (参见表)

	Wall	Closed door	Open door	Open hallway	Foyer
Nothing detected	0.70	0.40	0.05	0.001	0.30
Closed door detected	0.30	0.60	0	0	0.05
Open door detected	0	0	0.90	0.10	0.15
Open hallway detected	0	0	0.001	0.90	0.50

◇ 动作更新+感知更新

- ◇ 结合动作更新和感知更新

$$p(n_t|i_t) = \int p(n_t|n'_{t-i}, i_t)p(n'_{t-i})dn'_{t-i}$$

- ◇ 使用下标 $t-i$ 替代 $t-1$ ，是因为从 n' 到 n 的拓扑距离取决于特定的拓扑地图。



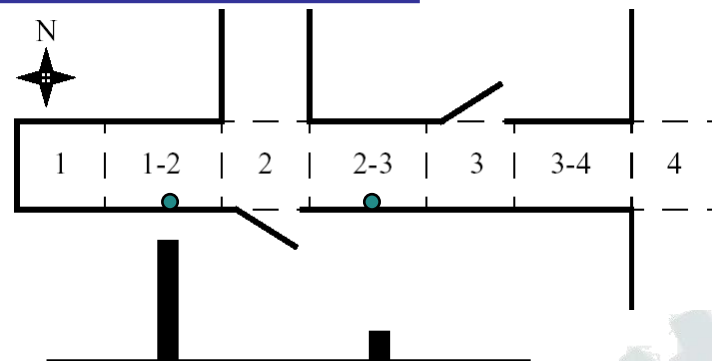
马尔可夫定位：实例研究 1 – 拓扑地图

◇ 实例计算

- ◇ 假设初始时刻机器人有 2 个信任状态：

- ◇ $p(1-2) = 1.0$; $p(2-3) = 0.2$ *

- 且面向东（确定）



两个状态的似然值

- ◇ 潜在地，状态 2-3 有进到 3、3-4、4 的可能；状态 1-2 有进到 2 的可能。
- ◇ 感知事件：左边是开放的走廊，右边是门开

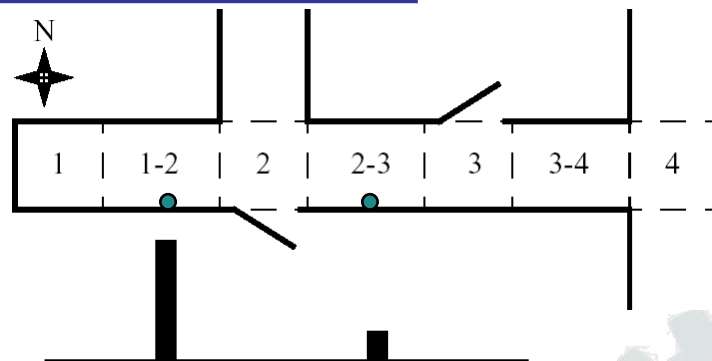
* 注意，其和不等于 0。为简单起见，此例中不做归一化处理。



马尔可夫定位 (2): 概率论应用于机器人定位

◇ 初始态（只有可能在两个位置）：

$$\begin{bmatrix} p_{t-1}(1-2) \\ p_{t-1}(2-3) \end{bmatrix} = \begin{bmatrix} 1 \\ 0.2 \end{bmatrix} \begin{matrix} \diagdown \\ \diagup \end{matrix} \mathbf{5:1}$$



◇ 下个时刻，从两个初始态分别到达不同位置的似然值：

$$\begin{cases} p(2|1-2, o_t) = 1 \\ p(2-3|1-2, o_t) = 0 \\ p(3|1-2, o_t) = 0, \\ p(3-4|1-2, o_t) = 0 \\ p(4|1-2, o_t) = 0 \end{cases}, \quad \begin{cases} p(2|2-3, o_t) = 0 \\ p(2-3|2-3, o_t) = 0 \\ p(3|2-3, o_t) = 1 \\ p(3-4|2-3, o_t) = 1 \\ p(4|2-3, o_t) = 1 \end{cases}$$



马尔可夫定位 (2): 概率论应用于机器人定位

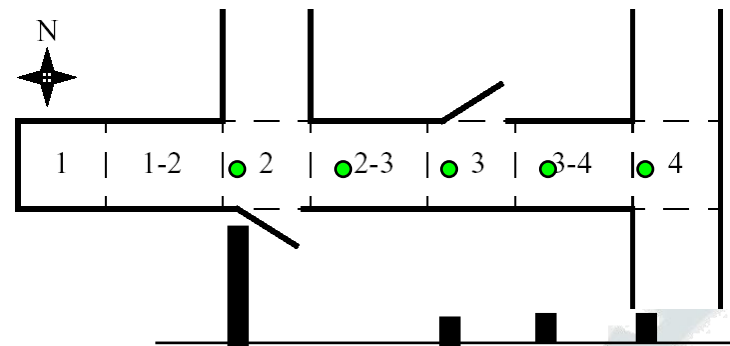
◇ 到达不同位置的验前（观察前）推算的可能度：

根据

$$p_t(l|o_t) = \sum_{l'} p(l|l', o_t) p_{t-1}(l')$$

可得

$$\begin{cases} p_t(2|o_t) = p(2|1-2, o_t) p_{t-1}(1-2) + p(2|2-3, o_t) p_{t-1}(2-3) = 1*1 + 0*0.2 = 1 \\ p_t(2-3|o_t) = p(2-3|1-2, o_t) p_{t-1}(1-2) + p(2-3|2-3, o_t) p_{t-1}(2-3) = 0*1 + 0*0.2 = 0 \\ p_t(3|o_t) = p(3|1-2, o_t) p_{t-1}(1-2) + p(3|2-3, o_t) p_{t-1}(2-3) = 0*1 + 1*0.2 = 0.2 \\ p_t(3-4|o_t) = p(3-4|1-2, o_t) p_{t-1}(1-2) + p(3-4|2-3, o_t) p_{t-1}(2-3) = 0*1 + 1*0.2 = 0.2 \\ p_t(4|o_t) = p(4|1-2, o_t) p_{t-1}(1-2) + p(4|2-3, o_t) p_{t-1}(2-3) = 0*1 + 1*0.2 = 0.2 \end{cases}$$





马尔可夫定位：实例研究 1 – 拓扑地图 (5)

◇ 观察

◇ i : “左边是开放的走廊，右边是门开”

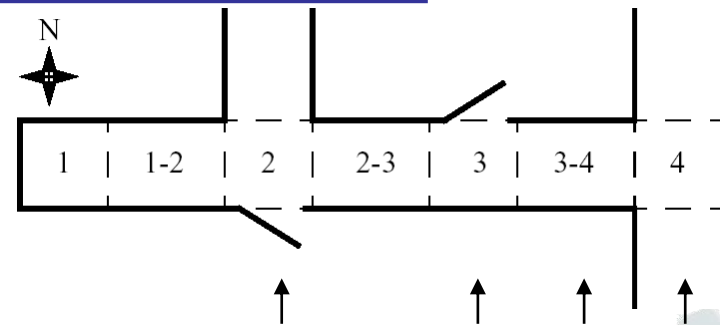
◇ 为得到:

$$p_t(l) = p(i|l) p_t(l|o_t)$$

先要计算: *

$$p(i|2), p(i|3), p(i|3-4), p(i|4)$$

*因 $p_t(2-3|o_t) = 0$, $p(i|2-3)$ 不需计算。





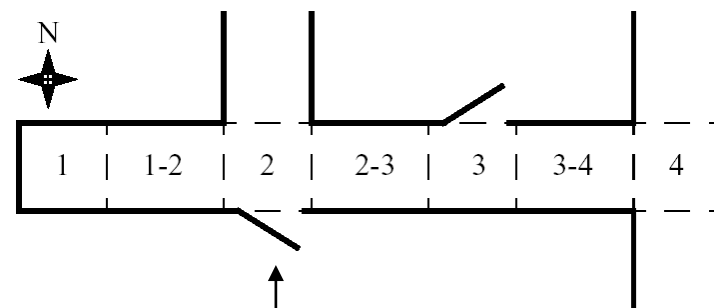
马尔可夫定位：实例研究 1 – 拓扑地图 (5)

 $p(i|2)$

◇ 感知：“**左边是开放的走廊，右边是门开**”

◇ 在 2 处，

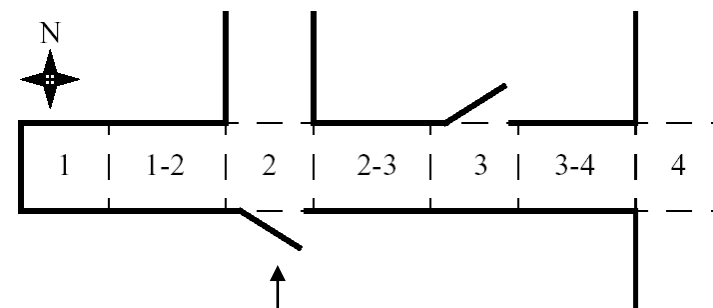
◇ 左边实为“开放的走廊”，正确地得到检测，表中查对应的似然值为：**0.9**；



	墙	门关 ^{0.6}	门开 ^{0.4}	开放型走廊	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50



马尔可夫定位：实例研究 1 – 拓扑地图 (5)

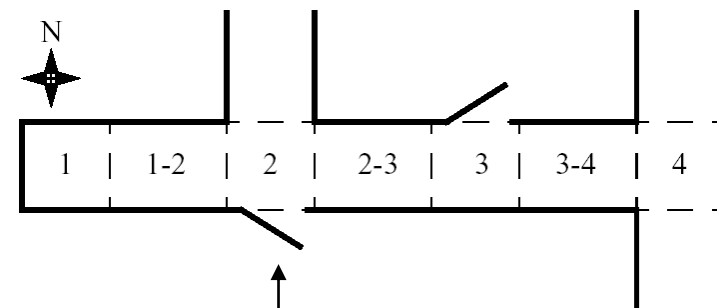


- ◇ 右边实为“门”（可能开着也可能关着，设可能性分别是 **0.4** 和 **0.6**），现检测为“门开”，因此查表后得似然值计算为： **$0.6*0+0.4*0.9$** 。

	墙	门关 ^{0.6}	门开 ^{0.4}	开放型走廊	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50



马尔可夫定位：实例研究 1 – 拓扑地图 (5)



◇ 于是，在 2 处得到感知：“**左边是开放的走廊，右边是门开**”的似然值是

$$p(i|2) = 0.9 * (0.6 * 0 + 0.4 * 0.9) = 0.324$$



马尔可夫定位：实例研究 1 – 拓扑地图 (5)

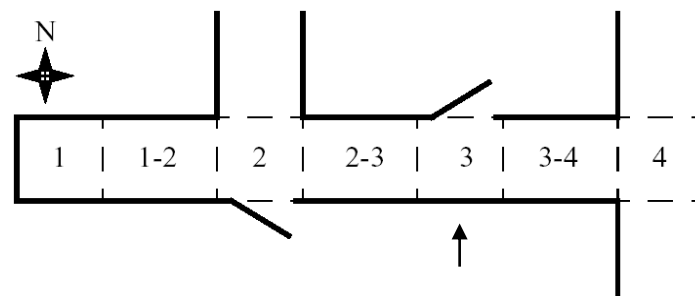
$$p(i|3)$$

◇ 感知：“左边是开放的走廊，右边是门开”

◇ 在 3 处，

◇ 右边实为“墙”，但检测到的是“门开”，表中查对应的似然值为 0，因此有：

$$p(i|3) = 0$$



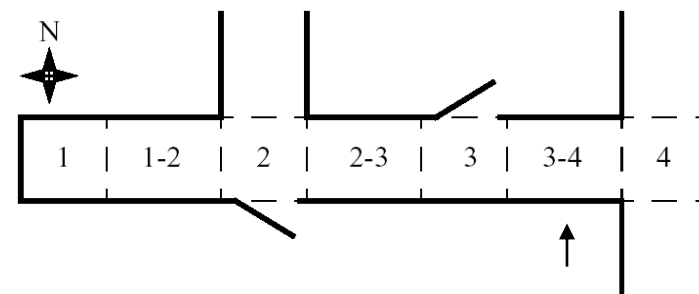
	墙	门关 ^{0.6}	门开 ^{0.4}	开放型走廊	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50



马尔可夫定位：实例研究 1 – 拓扑地图 (5)

$$p(i|3-4)$$

- ◇ 感知：“**左边是开放的走廊，右边是门开**”
- ◇ 在 3-4 处，
 - ◇ 右边实为“墙”，但检测到的也是“门开”，表中查对应的似然值为 0，因此有：



$$p(i|3-4) = 0$$

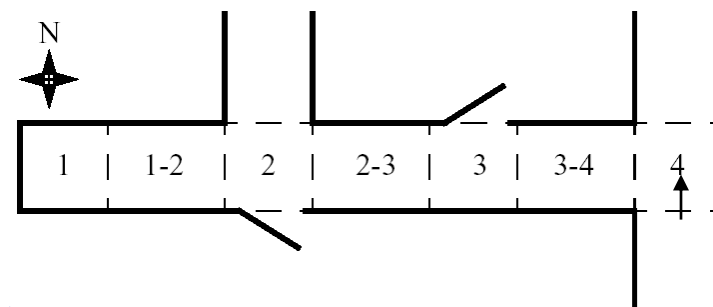
	墙	门关 ^{0.6}	门开 ^{0.4}	开放型走廊	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50



马尔可夫定位：实例研究 1 – 拓扑地图 (5)

$$p(i|4)$$

- ◇ 感知：“**左边是开放的走廊，右边是门开**”
- ◇ 在 4 处，
 - ◇ 左边实为“开放的走廊”，得到正确检测，表中查对应的似然值为 **0.9**；



	墙	门关 ^{0.6}	门开 ^{0.4}	开放型走廊	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50

* 为易于理解，省略书中对行进途中似然值的计算

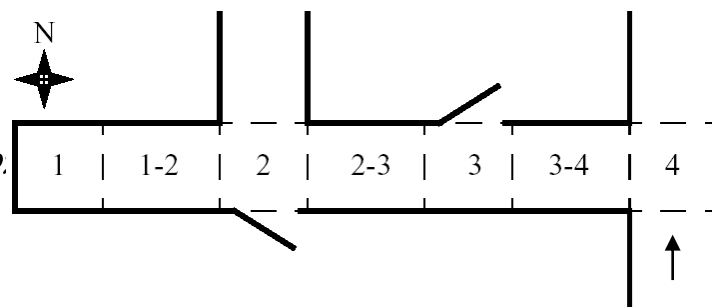


马尔可夫定位：实例研究 1 – 拓扑地图 (5)

◆ 感知：“**左边是开放的走廊，右边是门开**”

◆ 在 4 处，

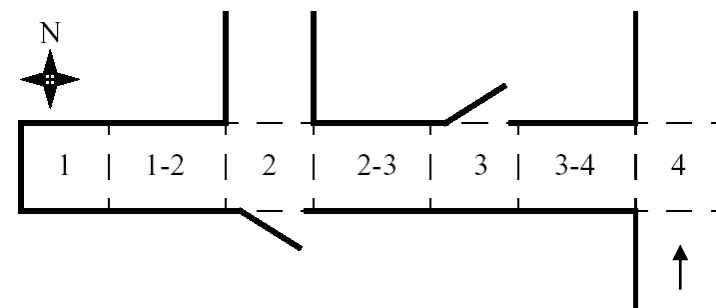
◆ 右边实为“开放的走廊”，检测到的是“门开”，表中查对应的似然值为 **0.1**。



	墙	门关 ^{0.6}	门开 ^{0.4}	开放型走廊	大厅
检测不到东西	0.70	0.40	0.05	0.001	0.30
检测到门关	0.30	0.60	0	0	0.05
检测到门开	0	0	0.90	0.10	0.15
检测到走廊开放	0	0	0.001	0.90	0.50



马尔可夫定位：实例研究 1 – 拓扑地图 (5)



- ◇ 于是，在 4 处得到感知：“**左边是开放的走廊，右边是门开**”的似然值是

$$p(i|4) = 0.9 * 0.1 = 0.09$$

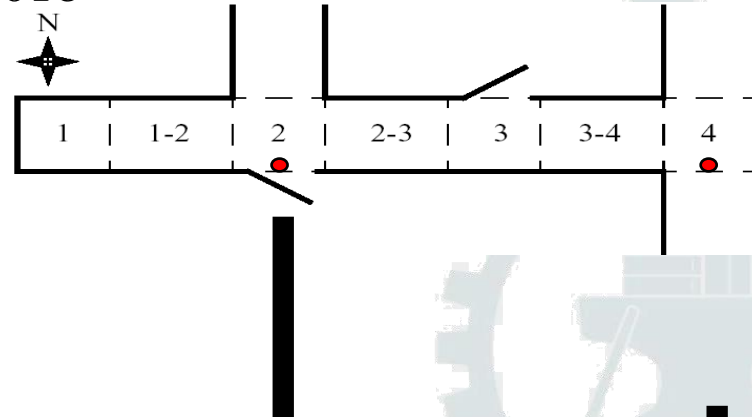


马尔可夫定位 (2): 概率论应用于机器人定位

◇ 最后得到验后似然值:

$$\begin{cases} p_t(1|o_t, i) = p(i) & 1) p_{t-1}(1-2) = 0.324 * 1 = 0.324 \\ p_t(2-3|o_t, i) = 0 \\ p_t(3|o_t, i) = p(i) & 3) p_{t-1}(2-3) = 0 * 0.2 = 0 \\ p_t(3-4|o_t, i) = p(i) & 4) p_{t-1}(2-3) = 0 * 0.2 = 0 \\ p_t(4|o_t, i) = p(i) & 4) p_{t-1}(2-3) = 0.09 * 0.2 = 0.018 \end{cases}$$

18:1





马尔可夫定位：实例研究 2 – 栅格地图 (1)

◇ 精确的固定分解栅格 (x, y, θ) , $15 \text{ cm} \times 15 \text{ cm} \times 1^\circ$

◇ 基于动作和感知的更新

◇ 基于动作的更新:

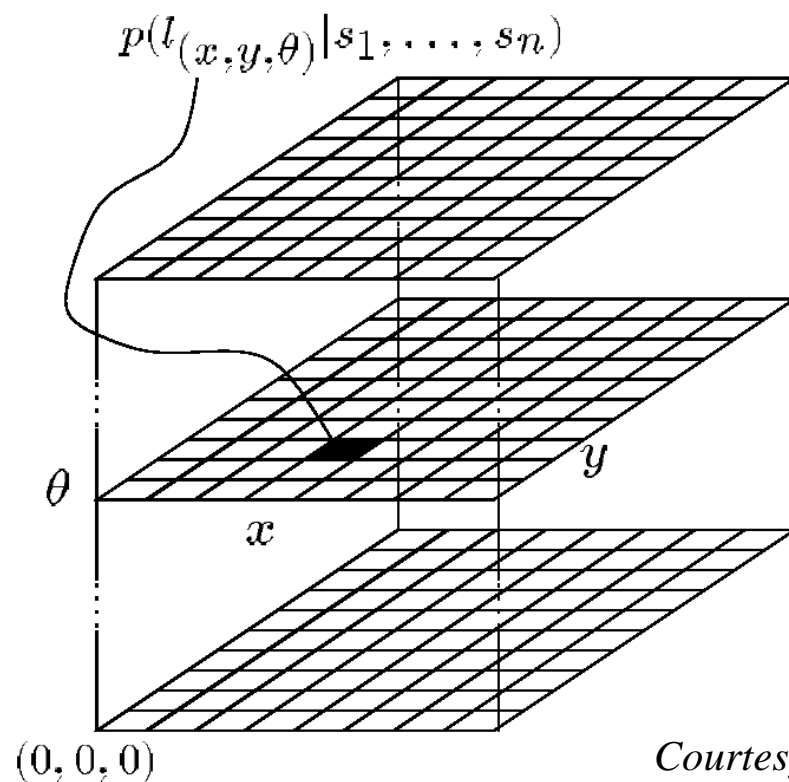
◇ 遍历先前所有可能的位置及运动模式求和

$$P(l_t | o_t) = \sum_l P(l_t | l_{t-1}, o_t) \cdot P(l_{t-1})$$

◇ 基于感知的更新:

◇ 给定距离感知 i , 机器人在位置 l 的概率按下式更新

$$p(l|i) = \frac{p(i|l)p(l)}{p(i)}$$



Courtesy of
W. Burgard

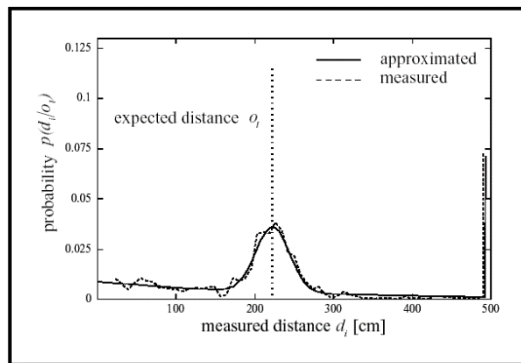


马尔可夫定位：实例研究 2 – 栅格地图(2)

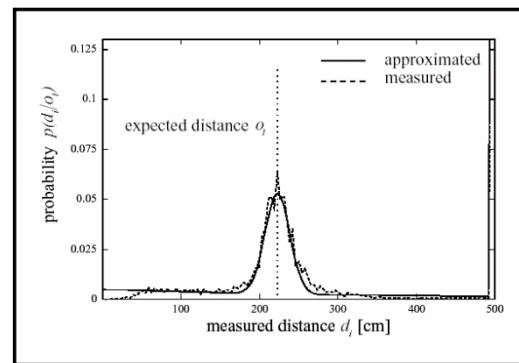
◆ 主要的难题是计算 $p(i|l)$

- ◆ 可能的传感器读数和几何上下文关系的数目是极多的
- ◆ 使用机器人传感器行为模型、它的位置 l 以及环绕 l 的局部环境度量地图，计算 $p(i|l)$ 。
- ◆ 假设
 - ◆ 测量误差可以用具有均值的分布予以描述
 - ◆ 任何测量都有非零机会

$$p(l|i) = \frac{p(i|l)p(l)}{p(i)}$$



Ultrasound.



Laser range-finder.

Courtesy of
W. Burgard



马尔可夫定位：实例研究 2- 栅格地图 (3)

◇ 1D 情况

1. 开始

- 开始时没有知识，所以有均匀概率分布。

2. 机器人感知到第一根柱子

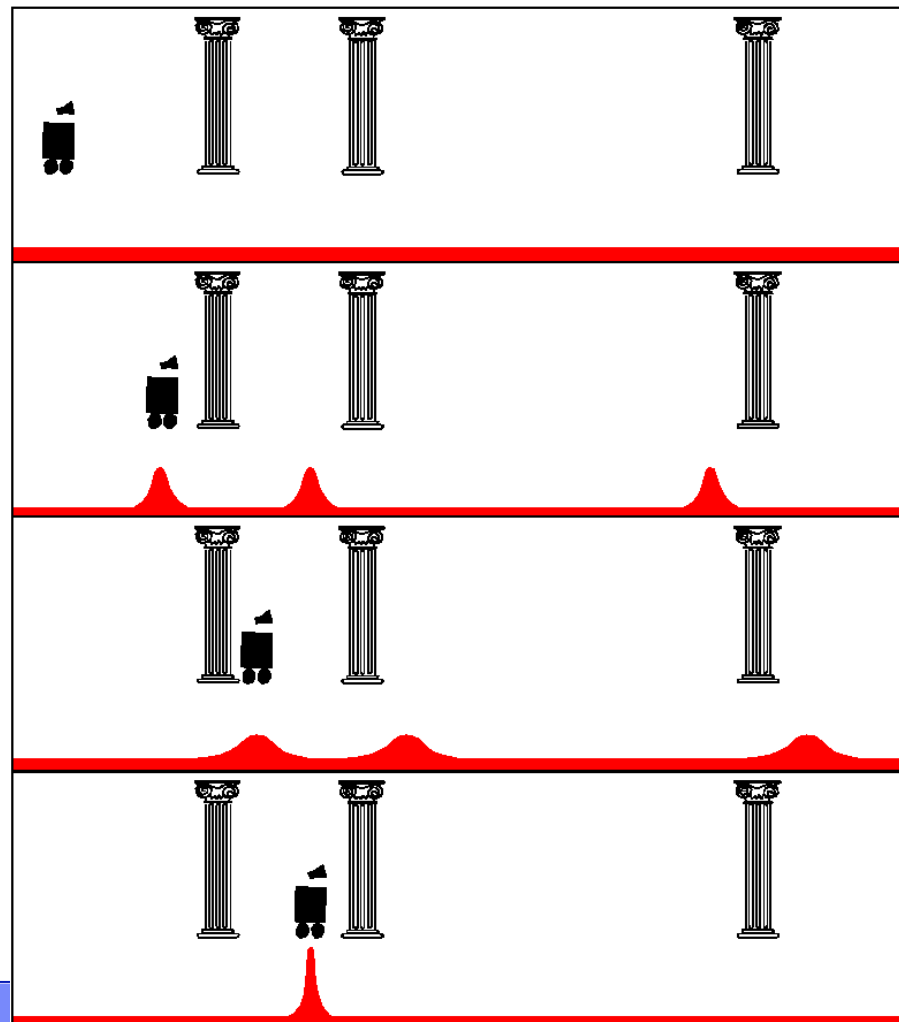
- 仅看到一根柱子，在柱子 1, 2 或 3 旁的概率相等。

3. 机器人移动

- 动作模式，使得根据以前分布和运动，估计新的概率分布成为可能

4. 机器人感知到第二根柱子

- 根据所有的先验知识，在柱子 2 旁的概率成为占优势的了

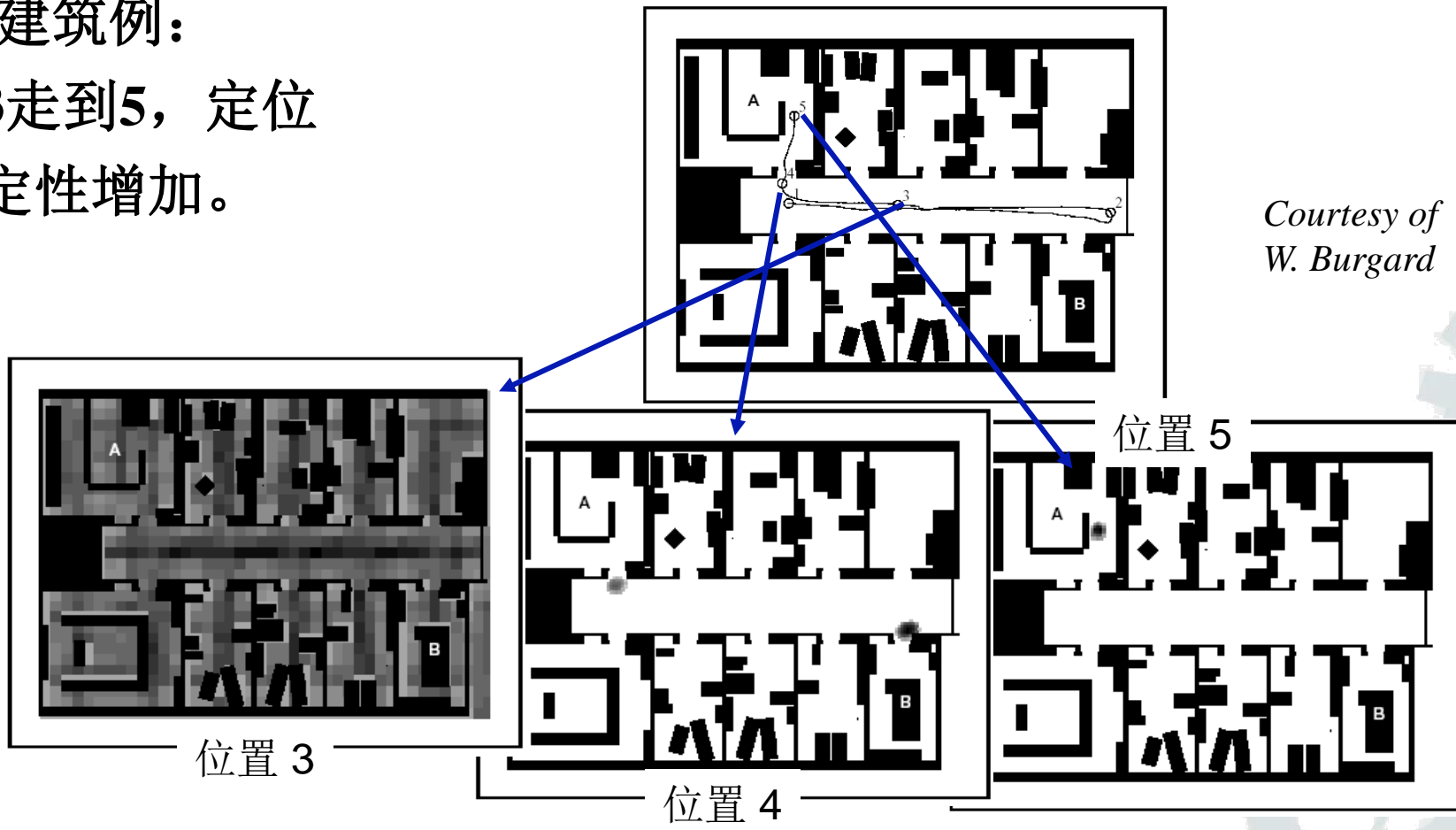




马尔可夫定位：实例研究 2- 栅格地图 (4)

◇ 办公建筑例：

从位置3走到5，定位的不确定性增加。



*Courtesy of
W. Burgard*

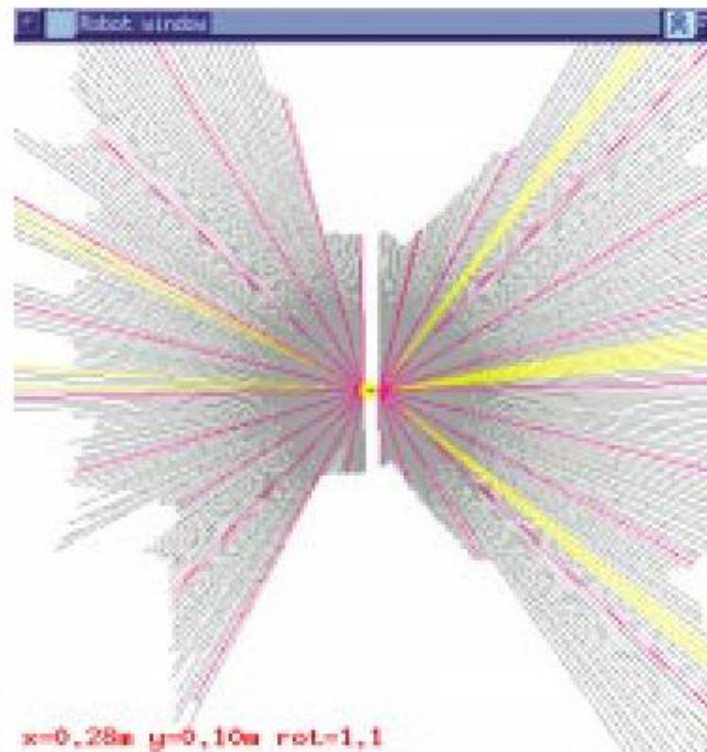
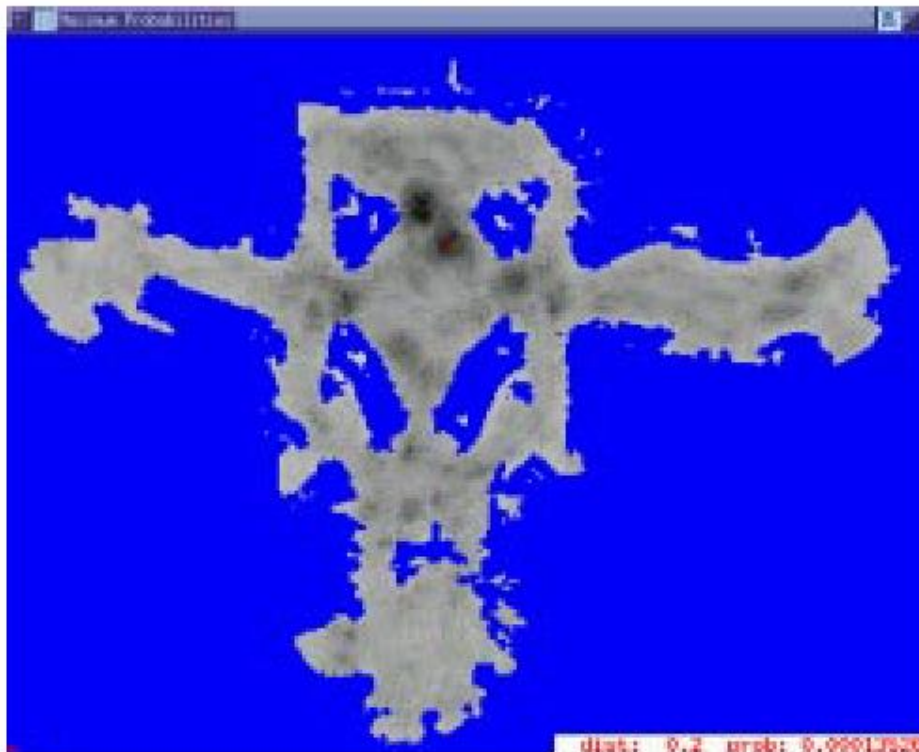


马尔可夫定位：实例研究 2- 栅格地图(5)

◇ 博物馆例：

◇ 激光扫描 1

*Courtesy of
W. Burgard*



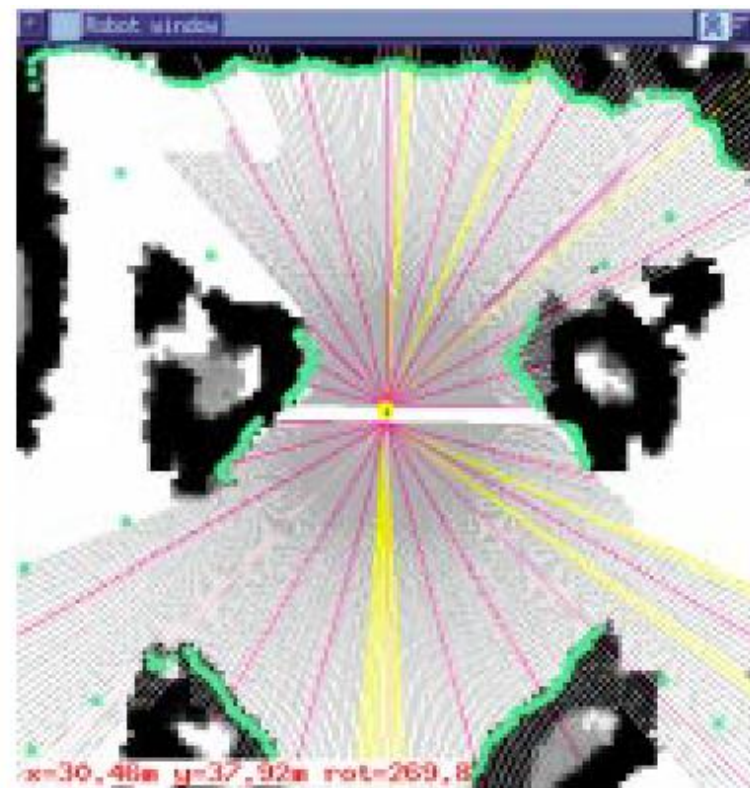
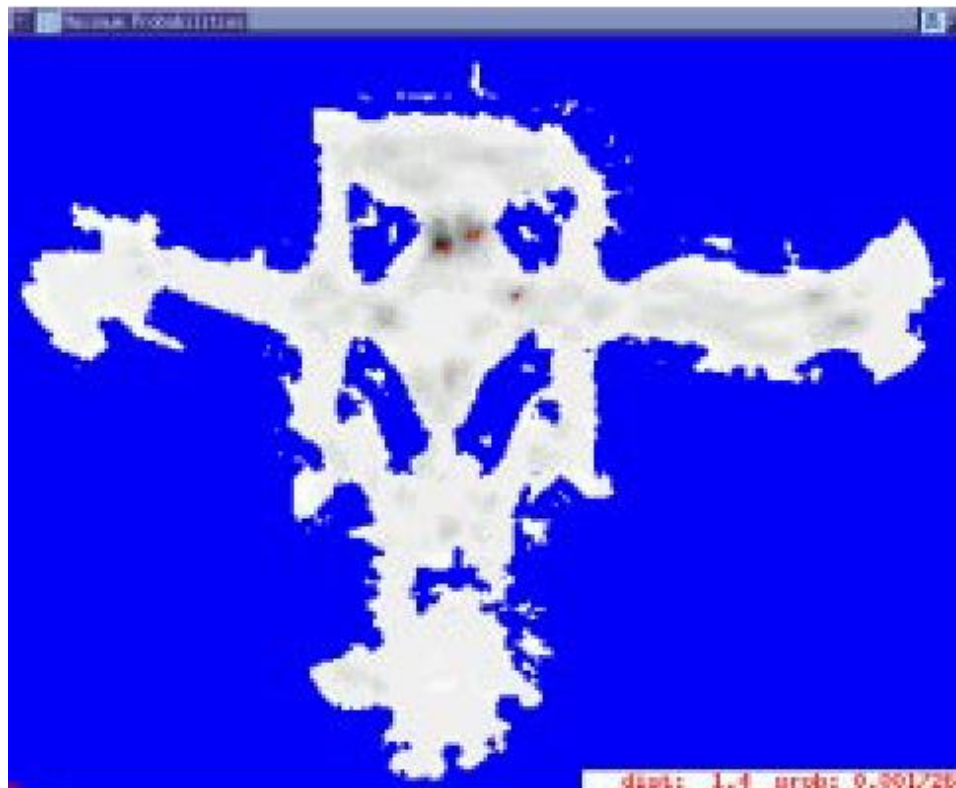


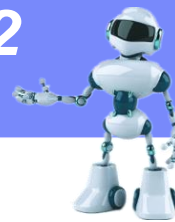
马尔可夫定位：实例研究 2- 栅格地图 (7)

◇ 博物馆例：

◇ 激光扫描 3

*Courtesy of
W. Burgard*



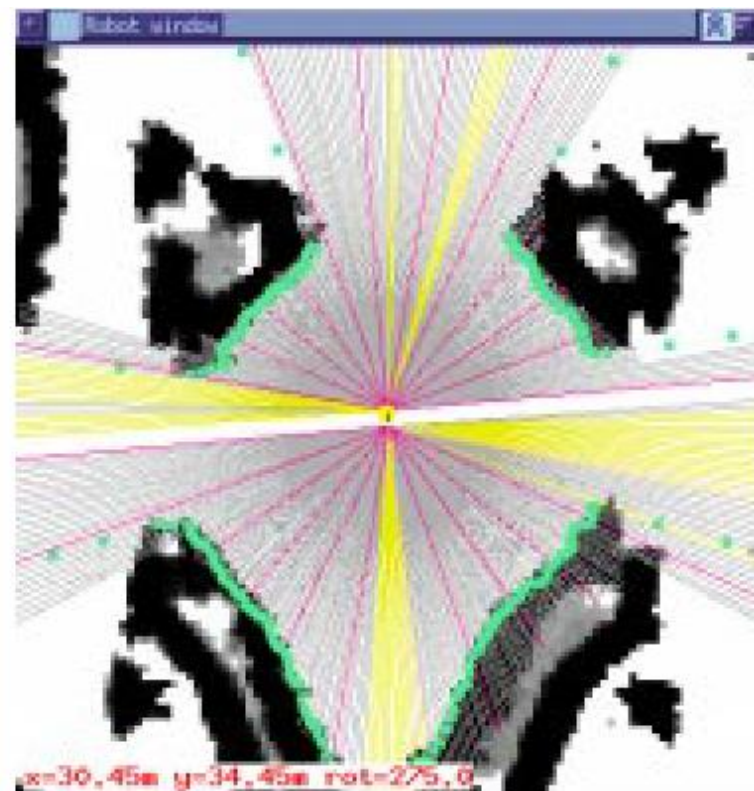
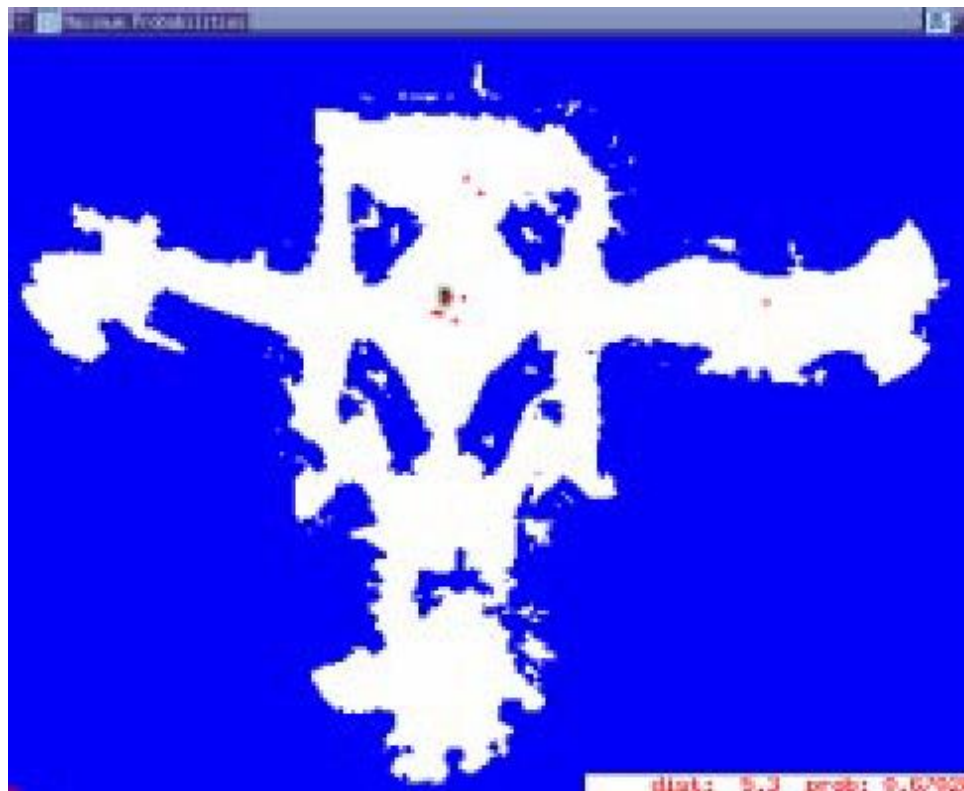


马尔可夫定位：实例研究 2- 栅格地图 (8)

◇ 博物馆例：

◇ 激光扫描 13

*Courtesy of
W. Burgard*



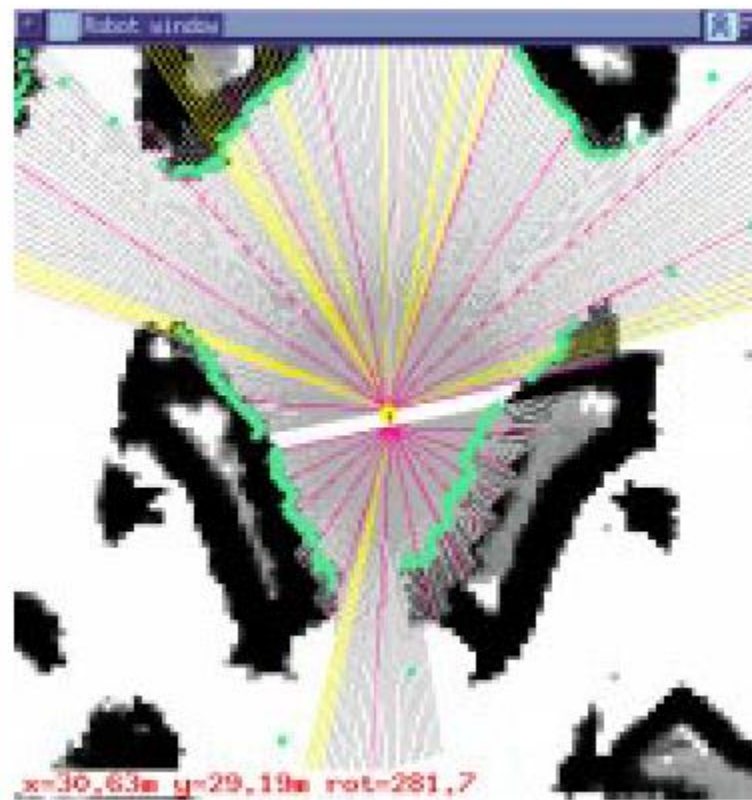
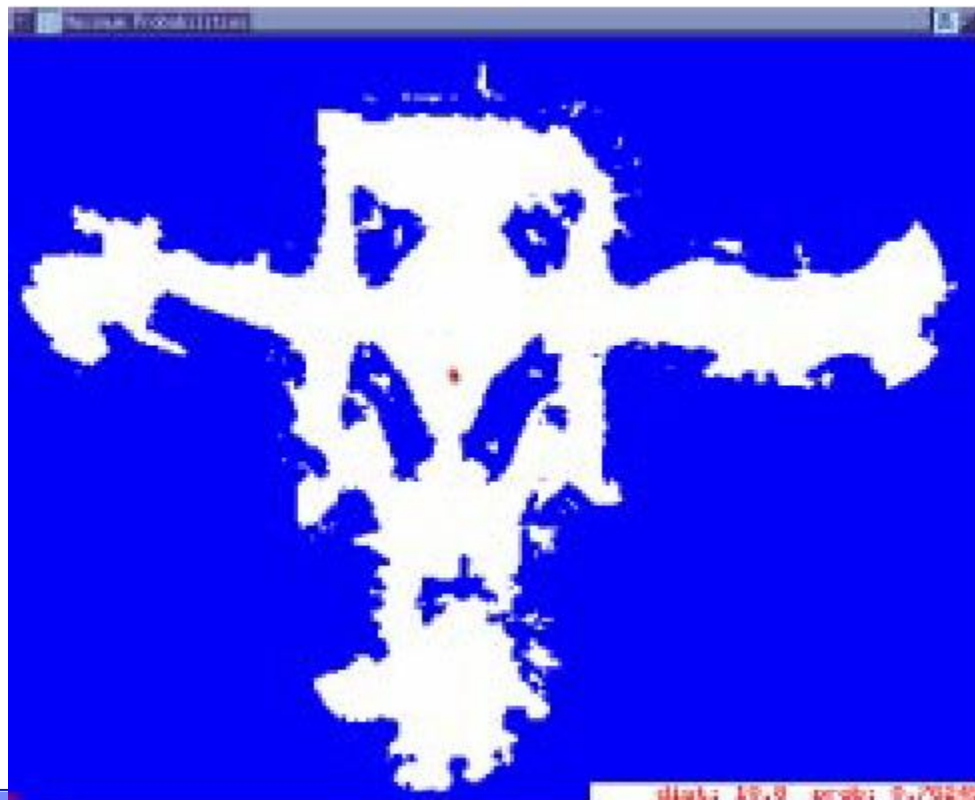


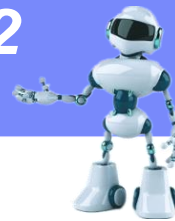
马尔可夫定位：实例研究 2- 栅格地图 (9)

◇ 博物馆例：

◇ 激光扫描 21

*Courtesy of
W. Burgard*



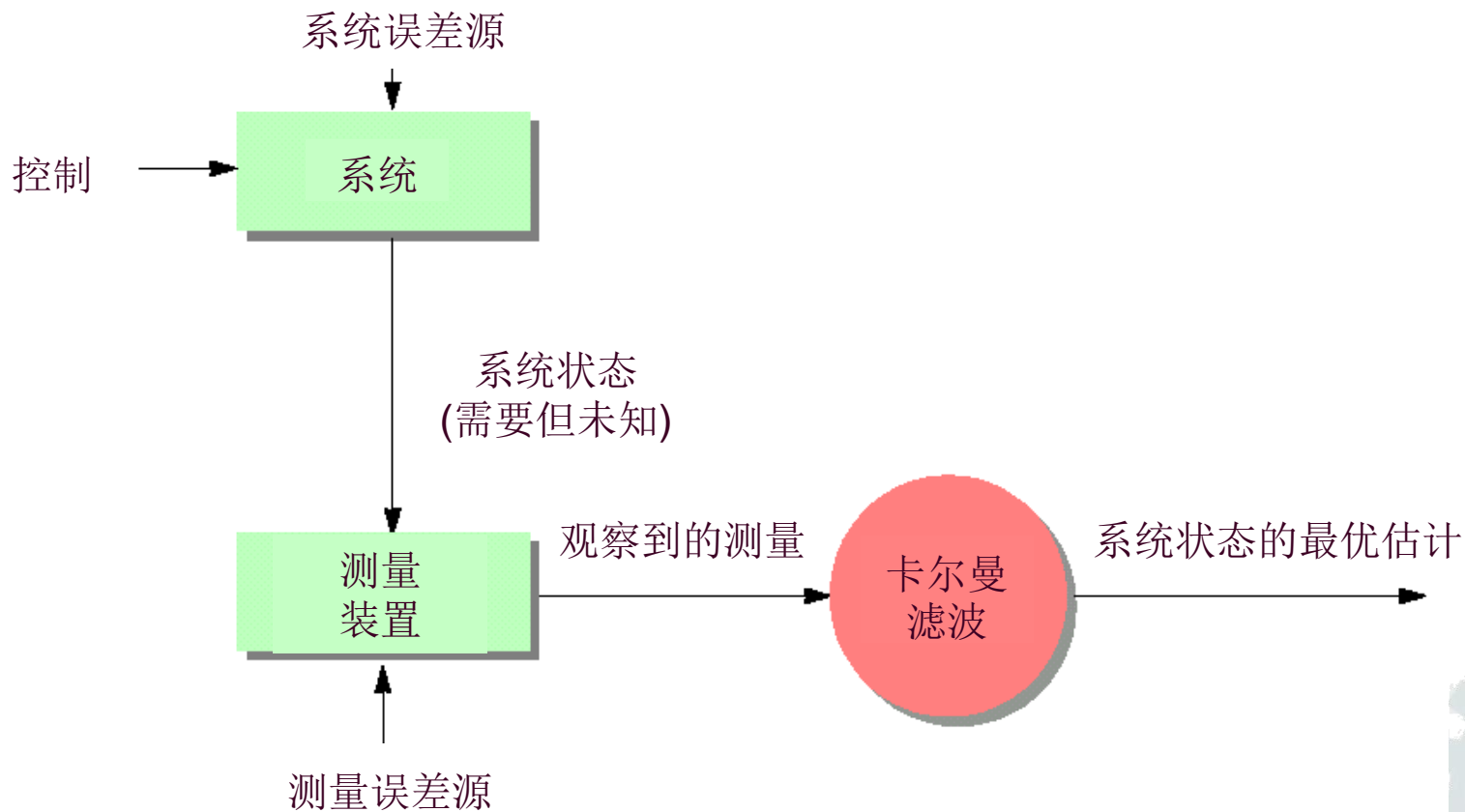


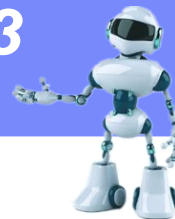
马尔可夫定位：实例研究 2- 栅格地图 (10)

- ◇ 精细固定分解的栅格导致巨大的状态空间
 - ◇ 需要非常大的处理能力
 - ◇ 需要大存储空间
- ◇ 降低复杂度
 - ◇ 已经提出了降低复杂度的各种方法
 - ◇ 主要目标是减少在每一步都进行更新的状态数目。
- ◇ 随机采样 / 粒子滤波
 - ◇ 只表示所有状态（可能位置）中有代表性的子集，以构造一个近似的信任度状态
 - ◇ 例如，仅更新所有可能位置中的 10%
 - ◇ 采样过程被特色加权，例如，在概率密度函数局部峰值周围设置更多的采样。
 - ◇ 然而，必须确保能够跟踪较小可能性的位置，否则，机器人可能会迷路。



卡尔曼滤波定位





卡尔曼滤波介绍 (1)

◇ 对位置的两次估计

第一次估计得 q_1 , 估计方差 σ_1^2

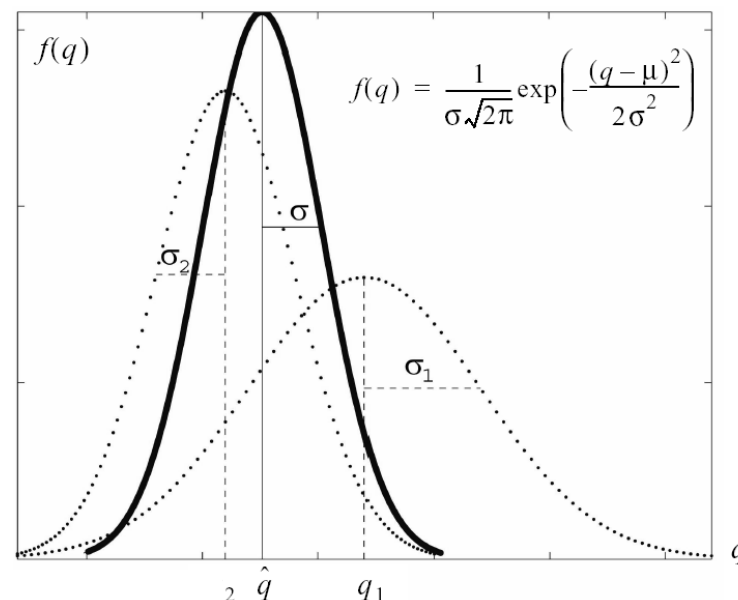
第二次估计得 q_2 , 估计方差 σ_2^2

◇ 求 n 次估计的加权最小二乘解 \hat{q}

$$S = \sum_{i=1}^n w_i (\hat{q} - q_i)^2$$

◇ 寻找最小偏差 $\frac{\partial S}{\partial \hat{q}} = \frac{\partial}{\partial \hat{q}} \sum_{i=1}^n w_i (\hat{q} - q_i)^2 = 2 \sum_{i=1}^n w_i (\hat{q} - q_i) = 0$

$$\sum_{i=1}^n w_i \hat{q} - \sum_{i=1}^n w_i q_i = 0 \Rightarrow \hat{q} = \frac{\sum_{i=1}^n w_i q_i}{\sum_{i=1}^n w_i}$$





卡尔曼滤波介绍 (2)

◇ $n=2$ 时, \hat{q} 可确定如下 (权 $w_i = 1/\sigma_i^2$) :

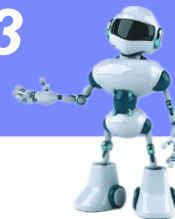
$$\hat{q} = \frac{\frac{1}{\sigma_1^2} q_1 + \frac{1}{\sigma_2^2} q_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} q_2$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 \sigma_2^2}; \quad \sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

◇ 合成后的方差比单独方差小。

◇ 还可写为:

$$\hat{q} = q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} (q_2 - q_1)$$



卡尔曼滤波介绍 (3)

◇ 卡尔曼滤波表示

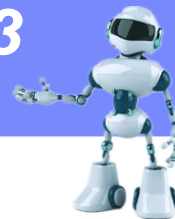
$$\hat{x}_{k+1} = \hat{x}_k + K_{k+1} \underbrace{(z_{k+1} - \hat{x}_k)}_{\text{新测量}}$$

◇ 式中

$$K_{k+1} = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2} ; \sigma_k^2 = \sigma_1^2 ; \sigma_z^2 = \sigma_2^2$$

◇ 方差更新:

$$\sigma_{k+1}^2 = \sigma_k^2 - K_{k+1} \sigma_k^2 \left(= \frac{\sigma_k^2 (\sigma_k^2 + \sigma_z^2) - \sigma_k^4}{\sigma_k^2 + \sigma_z^2} = \frac{\sigma_k^2 \sigma_z^2}{\sigma_k^2 + \sigma_z^2} > 0 \right)$$



卡尔曼滤波介绍 (4)

◇ 动态预测 (机器人运动)

$$\frac{dx}{dt} = u + w \quad \begin{array}{l} u = \text{velocity} \\ w = \text{noise} \end{array}$$

◇ 运动

$$\hat{x}_{k'} = \hat{x}_k + u(t_{k+1} - t_k)$$

$$\sigma_{k'}^2 = \sigma_k^2 + \sigma_w^2[t_{k+1} - t_k]$$

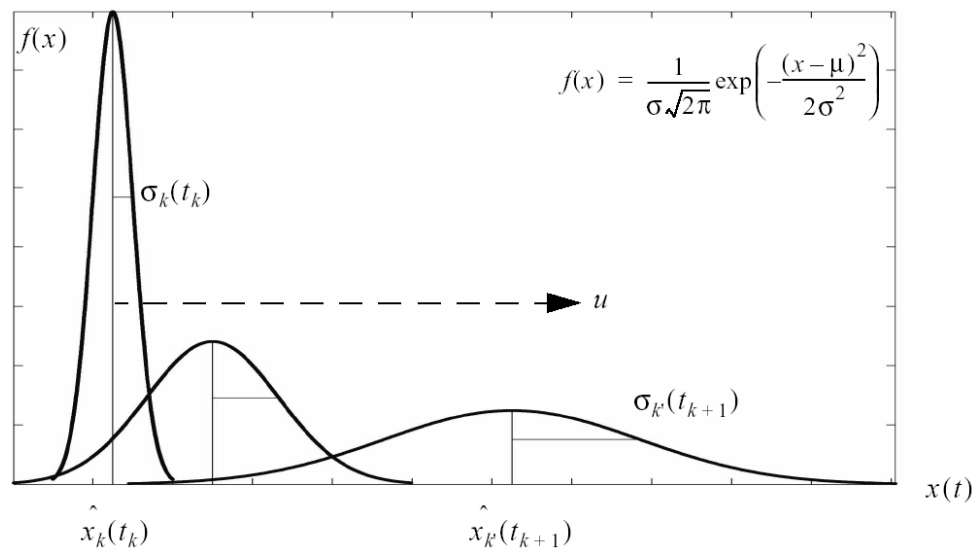
◇ 结合性的融合和动态预测

$$\hat{x}_{k+1} = \hat{x}_{k'} + K_{k+1}(z_{k+1} - \hat{x}_{k'})$$

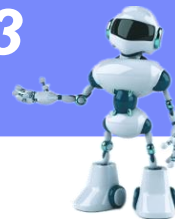
$$= [\hat{x}_k + u(t_{k+1} - t_k)] + K_{k+1}[z_{k+1} - \hat{x}_k - u(t_{k+1} - t_k)]$$

$$K_{k+1} = \frac{\sigma_{k'}^2}{\sigma_{k'}^2 + \sigma_z^2} = \frac{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k]}{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k] + \sigma_z^2}$$

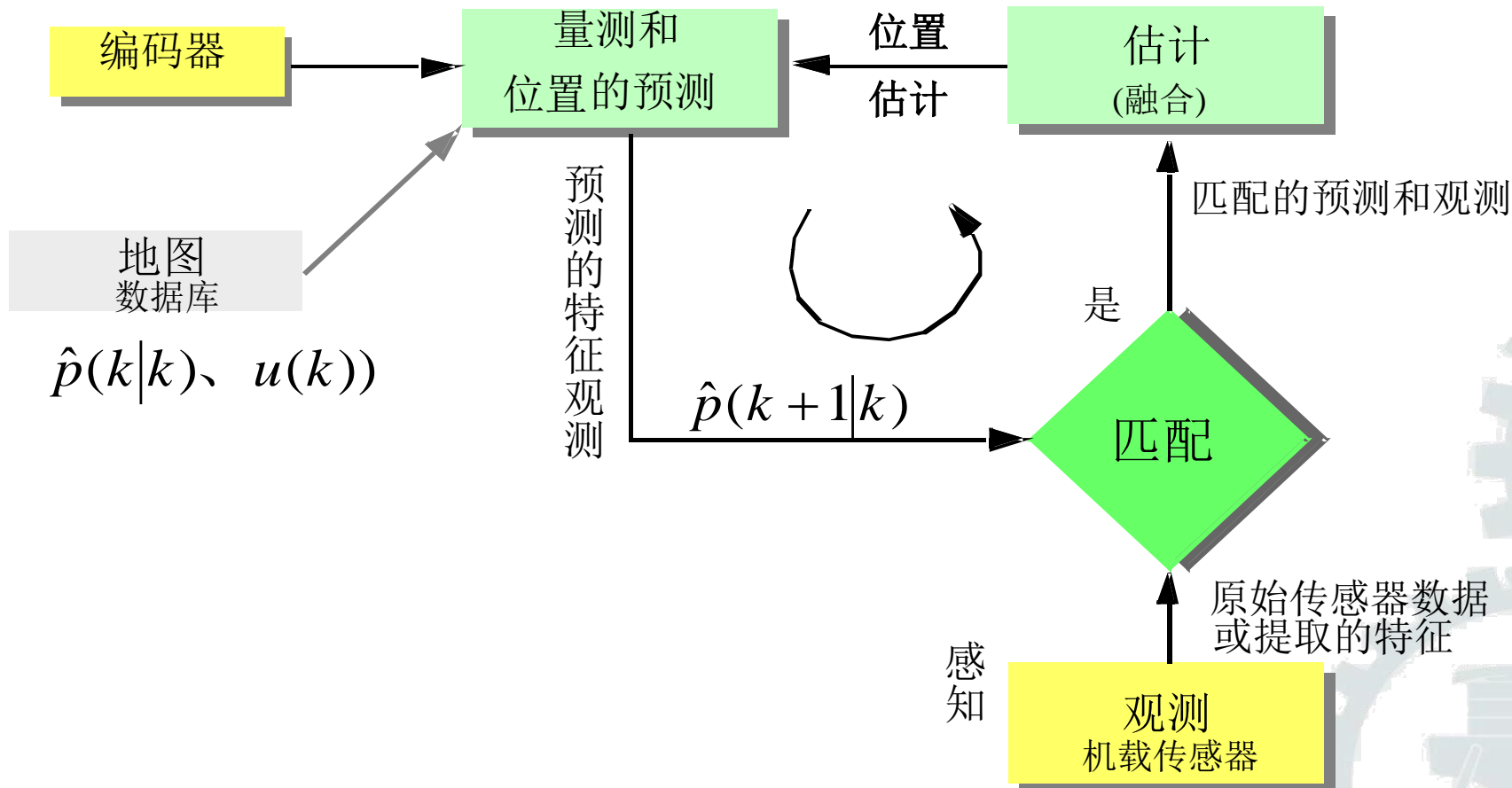
$$\sigma_{k+1}^2 = \sigma_{k'}^2 - K_{k+1}\sigma_{k'}^2$$

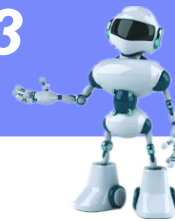


运动机器人概率密度的传播



用于机器人定位的卡尔曼滤波





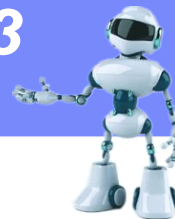
用于机器人定位的卡尔曼滤波 机器人位置预测

- ◇ 在第一步，根据机器人在时刻 k 的位置以及控制输入 $u(k)$ 产生的运动，预测 $k+1$ 时刻的机器人位置：

$$\hat{p}(k+1|k) = f(\hat{p}(k|k), u(k)) \quad f: \text{计程函数}$$

- ◇ 根据 $\hat{p}(k|k)$ 、 $u(k)$ 的方差计算 $\hat{p}(k+1|k)$ 的方差：

$$\Sigma_p(k+1|k) = \nabla_p f \cdot \Sigma_p(k|k) \cdot \nabla_p f^T + \nabla_u f \cdot \Sigma_u(k) \cdot \nabla_u f^T$$



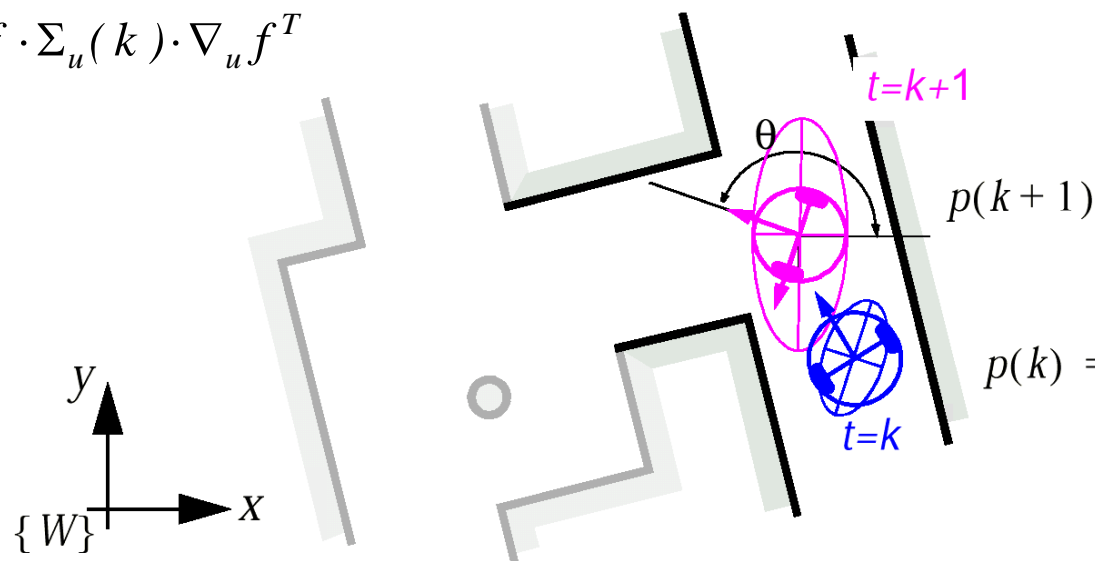
用于机器人定位的卡尔曼滤波 机器人位置预测：例

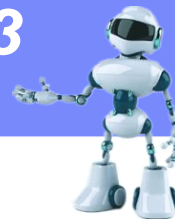
$$\hat{p}(k+1|k) = \hat{p}(k|k) + u(k) = \begin{bmatrix} \hat{x}(k) \\ \hat{y}(k) \\ \hat{\theta}(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix}$$

计程法

$$\Sigma_p(k+1|k) = \nabla_p f \cdot \Sigma_p(k|k) \cdot \nabla_p f^T + \nabla_u f \cdot \Sigma_u(k) \cdot \nabla_u f^T$$

$$\Sigma_u(k) = \begin{bmatrix} k_r |\Delta s_r| & 0 \\ 0 & k_l |\Delta s_l| \end{bmatrix}$$





用于机器人定位的卡尔曼滤波 观测

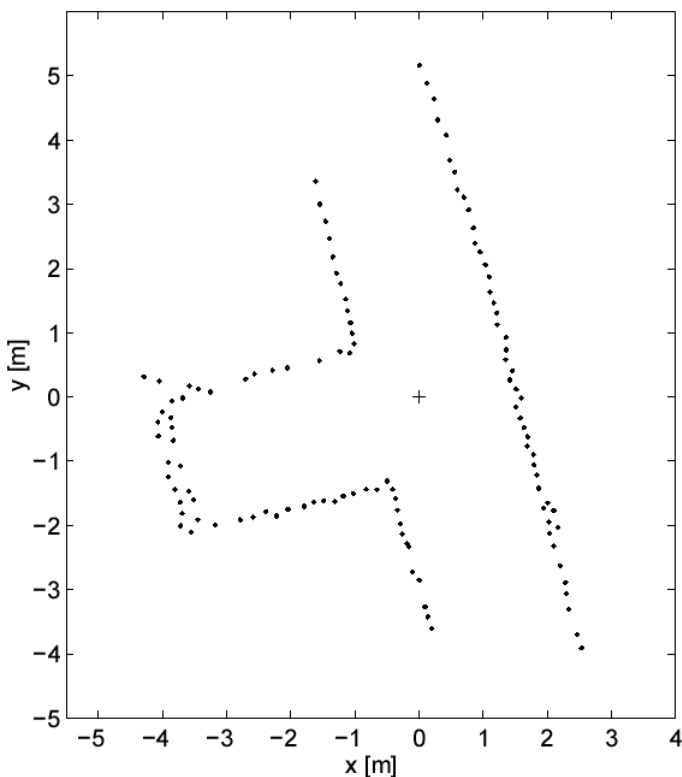
- ◇ 第二步， $k+1$ 时刻从机器人传感器得到在新位置的观测（测量）值 $Z(k+1)$
- ◇ 通常，此观测包含了从不同传感器信号提取的单独观测 $z_j(k+1)$ 的集合 n_0 。它可以表示原始数据，也可以表示特征，如直线、门或任何种类的地标。
- ◇ 对目标参数的观测，通常是以传感器坐标系{S}为参照的。
 - ◇ 因此，必须将观测转换到世界坐标系{W}下，或
 - ◇ 测量的预测，必须转换到传感器坐标系{S}下
 - ◇ 这个转换，在函数 h_i 中被指定(参照后面)

$$z_j(k+1) = \begin{bmatrix} \alpha_j \\ r_j \end{bmatrix} \quad \Sigma_{R,j}(k+1) = \begin{bmatrix} \sigma_{\alpha\alpha} & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_{rr} \end{bmatrix}_j$$

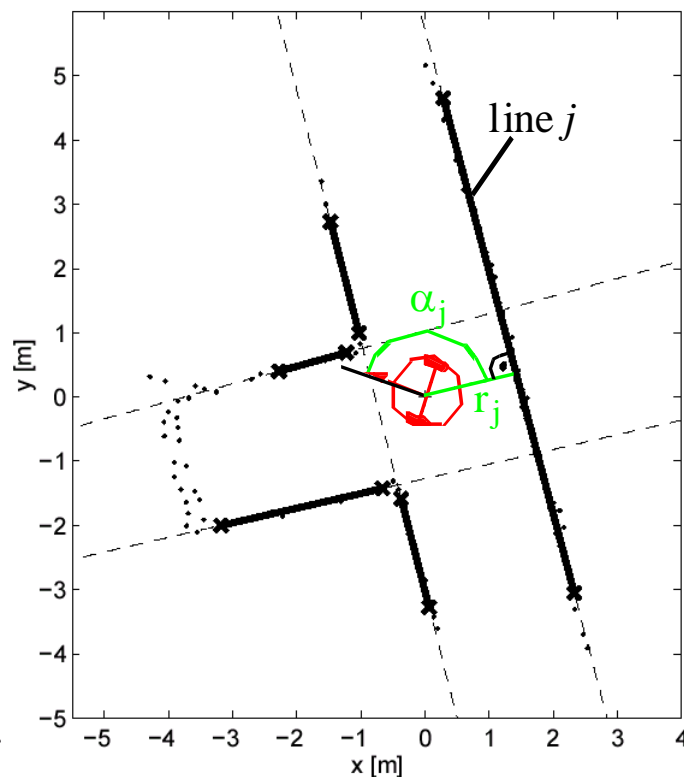


用于机器人定位的卡尔曼滤波 观测：例

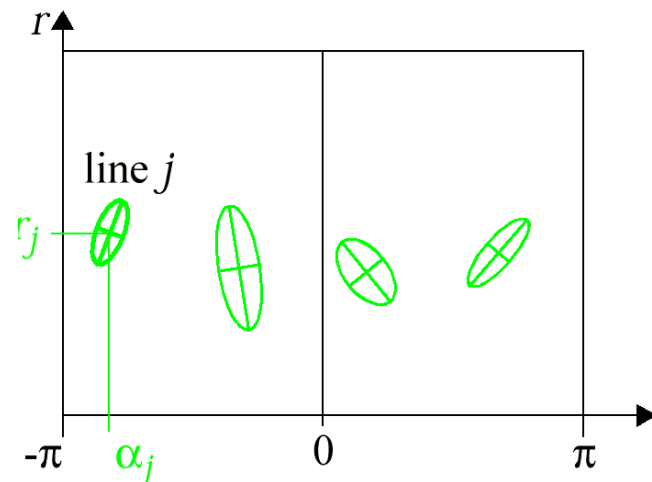
激光扫描器 得到的原始数据



提取的直线



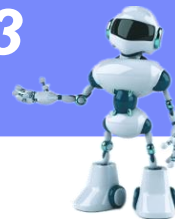
在模式空间中提取的直线 表示 (r, α)



$$z_j(k+1) = {}^R \begin{bmatrix} \alpha_j \\ r_j \end{bmatrix}$$

传感器
(机器人)
坐标系

$$\Sigma_{R,j}(k+1) = \begin{bmatrix} \sigma_{\alpha\alpha} & \sigma_{\alpha r} \\ \sigma_{r\alpha} & \sigma_{rr} \end{bmatrix}_j$$



用于机器人定位的卡尔曼滤波 测量预测

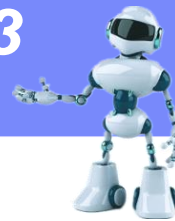
- ◇ 下一步，使用预测的机器人位置 $\hat{p} = (k+1|k)$ 和地图 $M(k)$ ，产生多个观测的预测值 z_t 。
- ◇ 必须将它们转换到传感器坐标系下：

$$\hat{z}_i(k+1) = h_i(z_t, \hat{p}(k+1|k))$$

- ◇ 我们可以把测量预测定义为包含所有 n_i 个观测的预测值的集合：

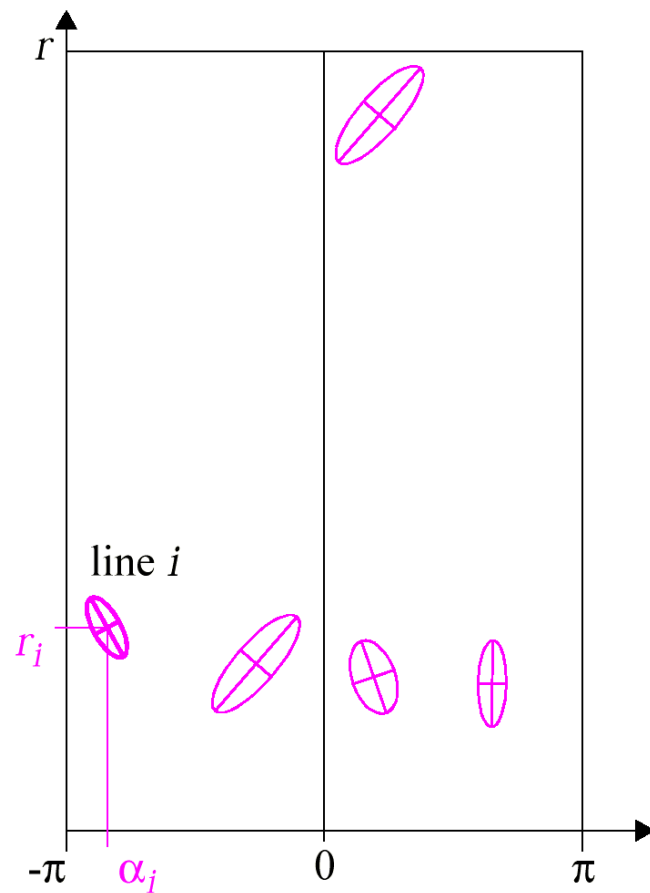
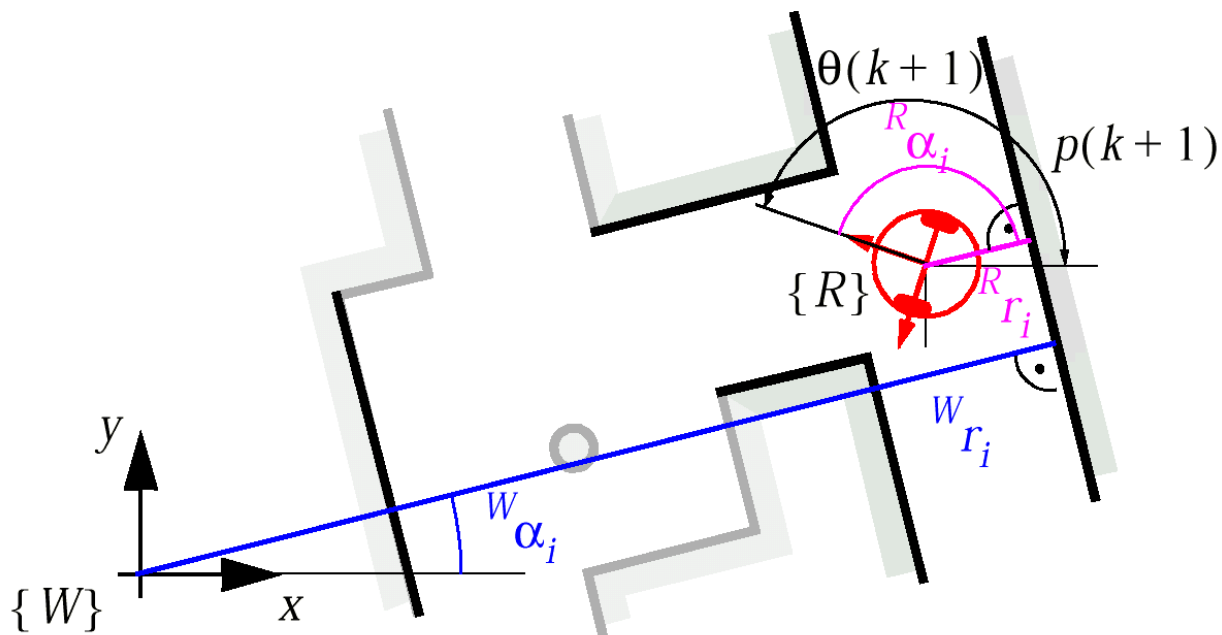
$$\hat{Z}(k+1) = \{\hat{z}_i(k+1) | (1 \leq i \leq n_i)\}$$

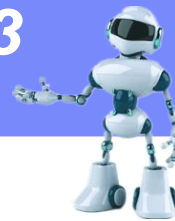
- ◇ 函数 h_i 主要实现从世界坐标系到传感器坐标系的坐标转换。



用于机器人定位的卡尔曼滤波 测量预测：例

- ◇ 对于预测，仅选择在机器人视场之内的墙。
- ◇ 通过将独立的直线与路径的节点相连接来做。





用于机器人定位的卡尔曼滤波 测量预测：例

- ◇ 产生的测量预测，必须转换到机器人坐标系 $\{R\}$

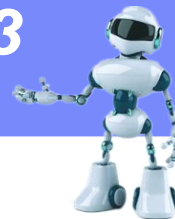
$${}^W z_{t,i} = \begin{bmatrix} \alpha_{t,i} \\ r_{t,i} \end{bmatrix} \rightarrow {}^R z_{t,i} = \begin{bmatrix} \alpha_{t,i} \\ r_{t,i} \end{bmatrix}$$

- ◇ 依照上一页图，该转换由下式给出：

$$\hat{z}_i(k+1) = \begin{bmatrix} \alpha_{t,i} \\ r_{t,i} \end{bmatrix} = h_i(z_{t,i}, \hat{p}(k+1|k)) = \begin{bmatrix} {}^W \alpha_{t,i} - {}^W \hat{\theta}(k+1|k) \\ {}^W r_{t,i} - ({}^W \hat{x}(k+1|k) \cos({}^W \alpha_{t,i}) + {}^W \hat{y}(k+1|k) \sin({}^W \alpha_{t,i})) \end{bmatrix}$$

式中 $\Sigma_{z,i}(k+1) = \nabla h_i \cdot \Sigma_p(k+1|k) \cdot \nabla h_i^T$

$$\nabla h_i = \begin{bmatrix} \frac{\partial \alpha_{t,i}}{\partial \hat{x}} & \frac{\partial \alpha_{t,i}}{\partial \hat{y}} & \frac{\partial \alpha_{t,i}}{\partial \hat{\theta}} \\ \frac{\partial r_{t,i}}{\partial \hat{x}} & \frac{\partial r_{t,i}}{\partial \hat{y}} & \frac{\partial r_{t,i}}{\partial \hat{\theta}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ -\cos {}^W \alpha_{t,i} & -\sin {}^W \alpha_{t,i} & 0 \end{bmatrix}$$



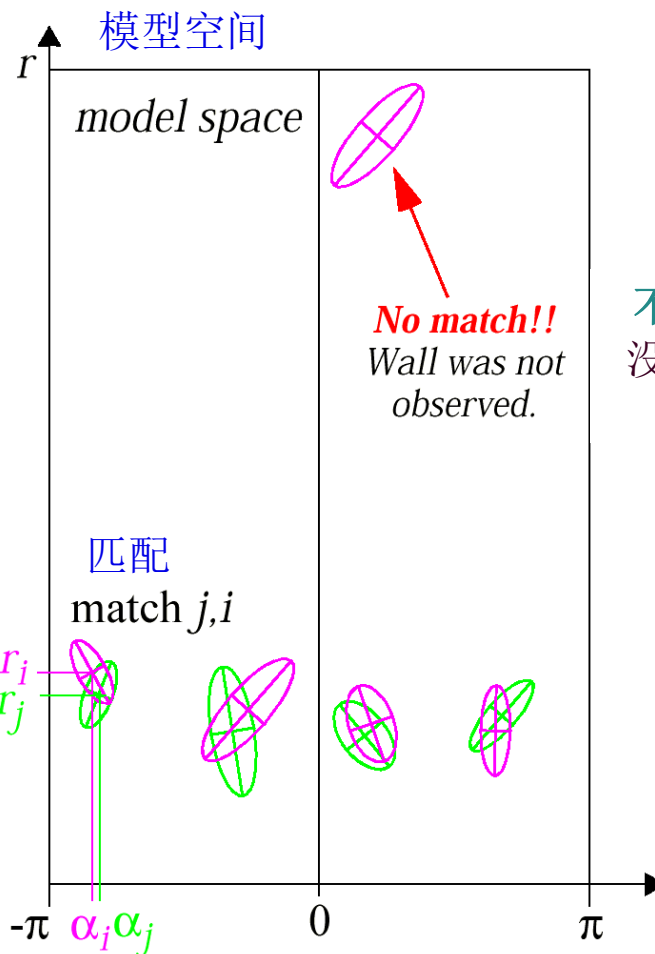
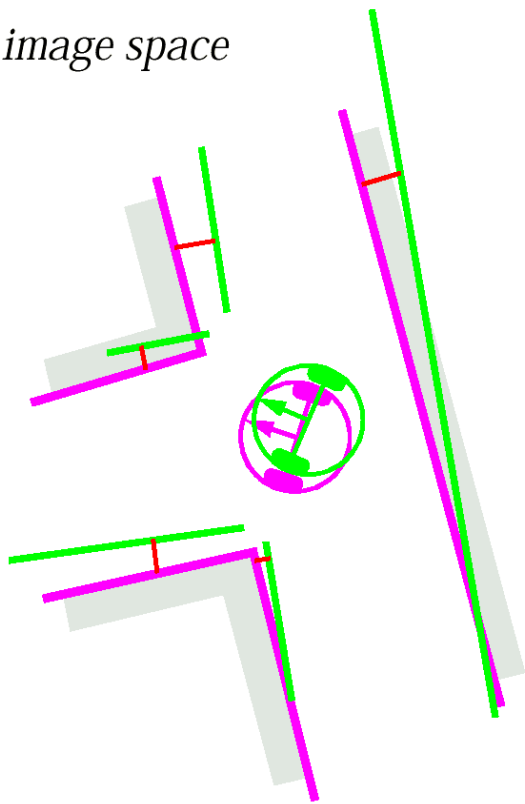
用于机器人定位的卡尔曼滤波 匹配

- ◇ 将观测 $z_j(k+1)$ (传感器获得) 对应到目标 z_t (地图存储)。(配对)

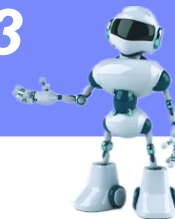


用于机器人定位的卡尔曼滤波 匹配

图像空间
image space



不匹配!!
没有观测到墙



用于机器人定位的卡尔曼滤波 匹配

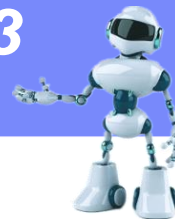
◇ 为寻找预测特征和观测特征的对应（对），我们采用 Mahalanobis 距离：

$$v_{ij}(k+1) \cdot \Sigma_{IN,ij}^{-1}(k+1) \cdot v_{ij}^T(k+1) \leq g^2$$

其中，

$$\begin{aligned} v_{ij}(k+1) &= [z_j(k+1) - h_i(z_t, \hat{p}(k+1|k))] \\ &= \begin{bmatrix} \alpha_j \\ r_j \end{bmatrix} - \begin{bmatrix} {}^W \alpha_{t,i} - {}^W \hat{\theta}(k+1|k) \\ {}^W r_{t,i} - ({}^W \hat{x}(k+1|k) \cos({}^W \alpha_{t,i}) + {}^W \hat{y}(k+1|k) \sin({}^W \alpha_{t,i})) \end{bmatrix} \end{aligned}$$

$$\Sigma_{IN,ij}(k+1) = \nabla h_i \cdot \Sigma_p(k+1|k) \cdot \nabla h_i^T + \Sigma_{R,j}(k+1)$$



用于机器人定位的卡尔曼滤波 匹配

◇ 设找到了 M 个匹配对，写出总的新息向量：

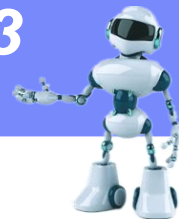
$$v(k+1) = \begin{bmatrix} z_{j_1}(k+1) - h_{i_1}(z_t, \hat{p}(k+1)|k) \\ z_{j_2}(k+1) - h_{i_2}(z_t, \hat{p}(k+1)|k) \\ \vdots \\ z_{j_M}(k+1) - h_{i_M}(z_t, \hat{p}(k+1)|k) \end{bmatrix}$$

◇ 协方差阵：

$$\Sigma_{IN}(k+1) = \text{diag}(\nabla h_{i_m} \cdot \Sigma_p(k+1|k) \cdot \nabla h_{i_m}^T + \Sigma_{R,j_m}(k+1))$$

◇ 以及（后面用）：

$$\nabla h = \begin{bmatrix} \nabla h_{i_1} \\ \nabla h_{i_2} \\ \vdots \\ \nabla h_{i_M} \end{bmatrix}$$



用于机器人定位的卡尔曼滤波 估计：采用卡尔曼滤波

◇ 卡尔曼滤波增益：

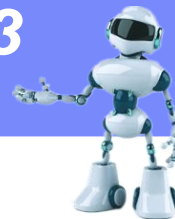
$$K(k+1) = \Sigma_p(k+1|k) \cdot \nabla h^T \cdot \Sigma_{IN}^{-1}(k+1)$$

◇ 机器人位置估计值的更新：

$$\hat{p}(k+1|k+1) = \hat{p}(k+1|k) + K(k+1) \cdot v(k+1)$$

◇ 协方差

$$\Sigma_p(k+1|k+1) = \Sigma_p(k+1|k) - K(k+1) \cdot \Sigma_{IN}(k+1) \cdot K^T(k+1)$$



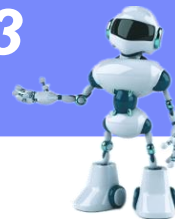
用于机器人定位的卡尔曼滤波

估计：1D 情况

◇ 对于1D情况且有 $h_i(z_t, \hat{p}(k+1|k)) = z_t$ ，可以证明，估计与前面介绍的一维卡尔曼滤波是一致的。

$$K(k+1) = \frac{\sigma_p^2(k+1|k)}{\sigma_{IN}^2(k+1)} = \frac{\sigma_p^2(k+1|k)}{\sigma_p^2(k+1|k) + \sigma_R^2(k+1)}$$

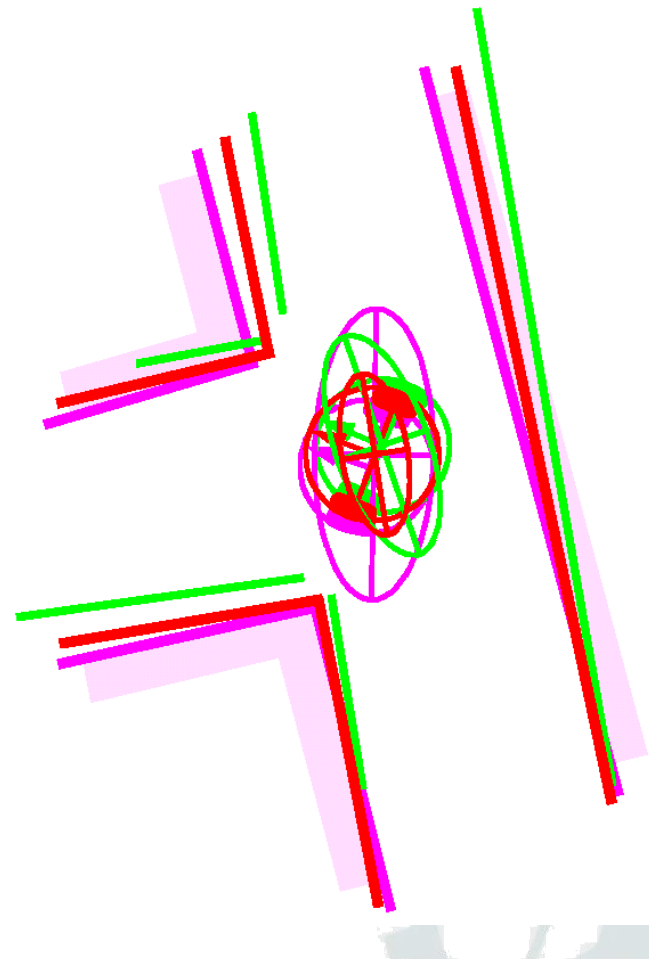
$$\begin{aligned}\hat{p}(k+1|k+1) &= \hat{p}(k+1|k) + K(k+1) \cdot v(k+1) \\ &= \hat{p}(k+1|k) + K(k+1) \cdot [z_j(k+1) - h_i(z_t, \hat{p}(k+1|k))] \\ &= \hat{p}(k+1|k) + K(k+1) \cdot [z_j(k+1) - z_t]\end{aligned}$$



用于机器人定位的卡尔曼滤波

估计：例

- ◇ 机器人新位置 $\hat{p}(k|k)$ 的卡尔曼滤波估计：
 - ◇ 融和机器人位置的预测量 (洋红色) 和根据测量得到的改进 (绿色)，可以得到机器人位置 (红色) 的更新估计。





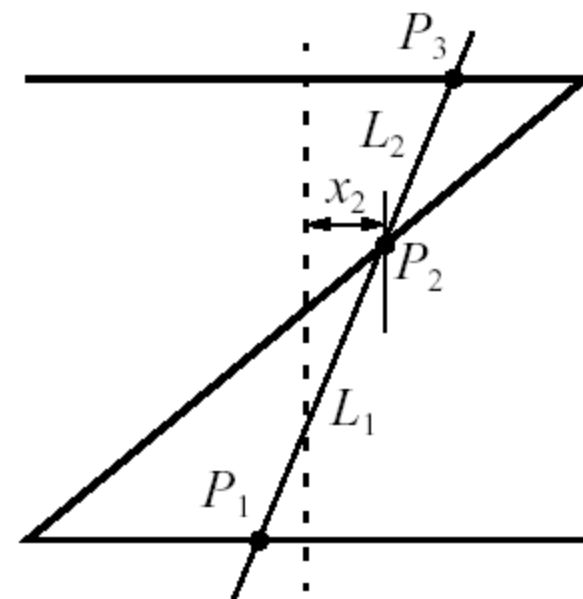
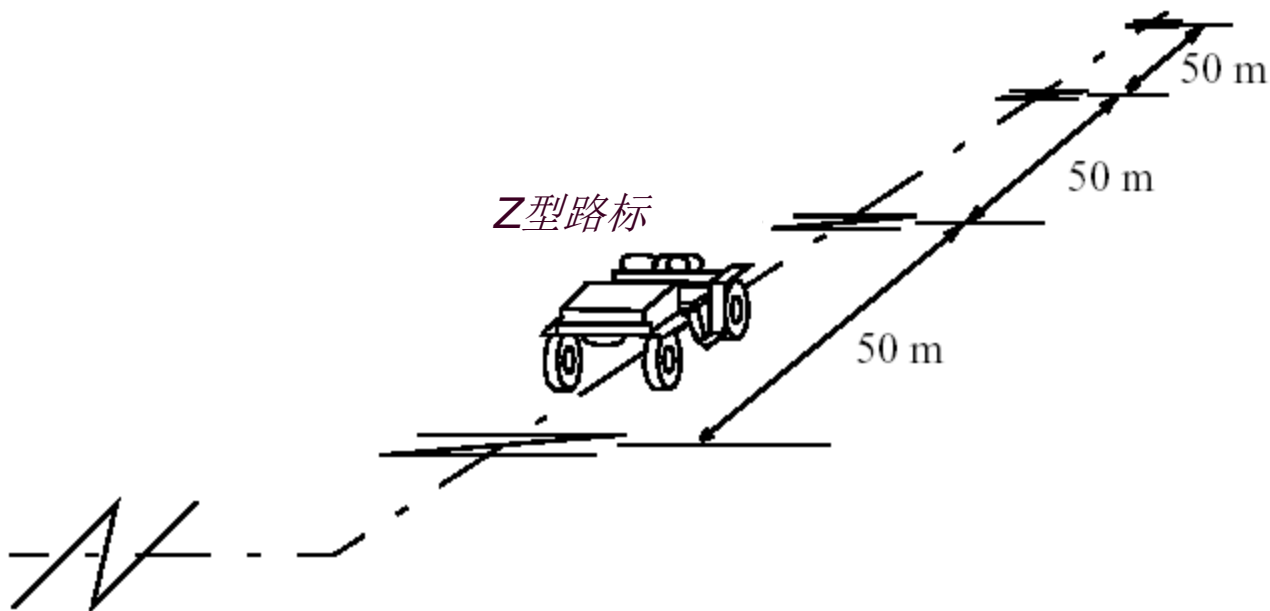
西安交通大学
Xi'an Jiaotong University

*Systems Engineering Institute,
Xi'an Jiaotong University,
Xi'an ShaanXi,
710049, P.R.China
Phone: 86-29-82667771*

The End



其他定位方法 (非概率方法) 基于人工路标的定位





其他定位方法 (非概率方法) 基于人工路标的定位

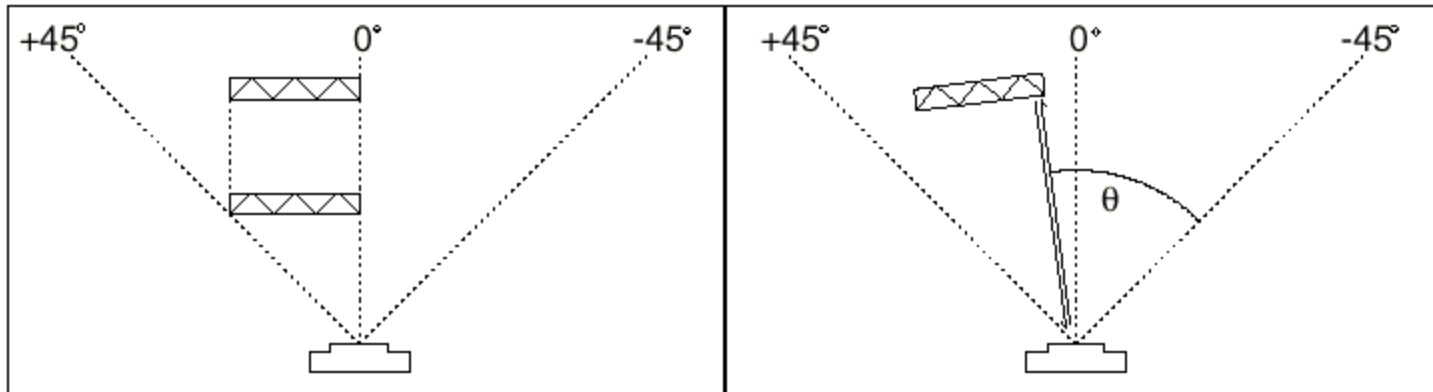


图6.11:a.根据已知大小反射目标的可感知宽度计算距离； b.从开始扫描到检测到目标边缘的时间得到目标方位

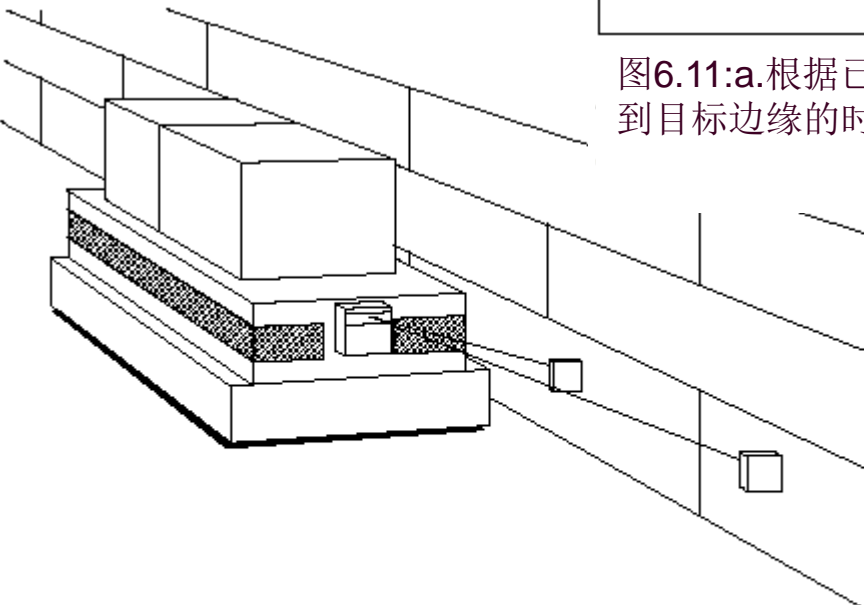


图6.10: LASERNET系统能够借助突出安装在墙上的靶子，以预先定义的偏移距离对AGV导航。



其他定位方法 (非概率方法)

基于人工路

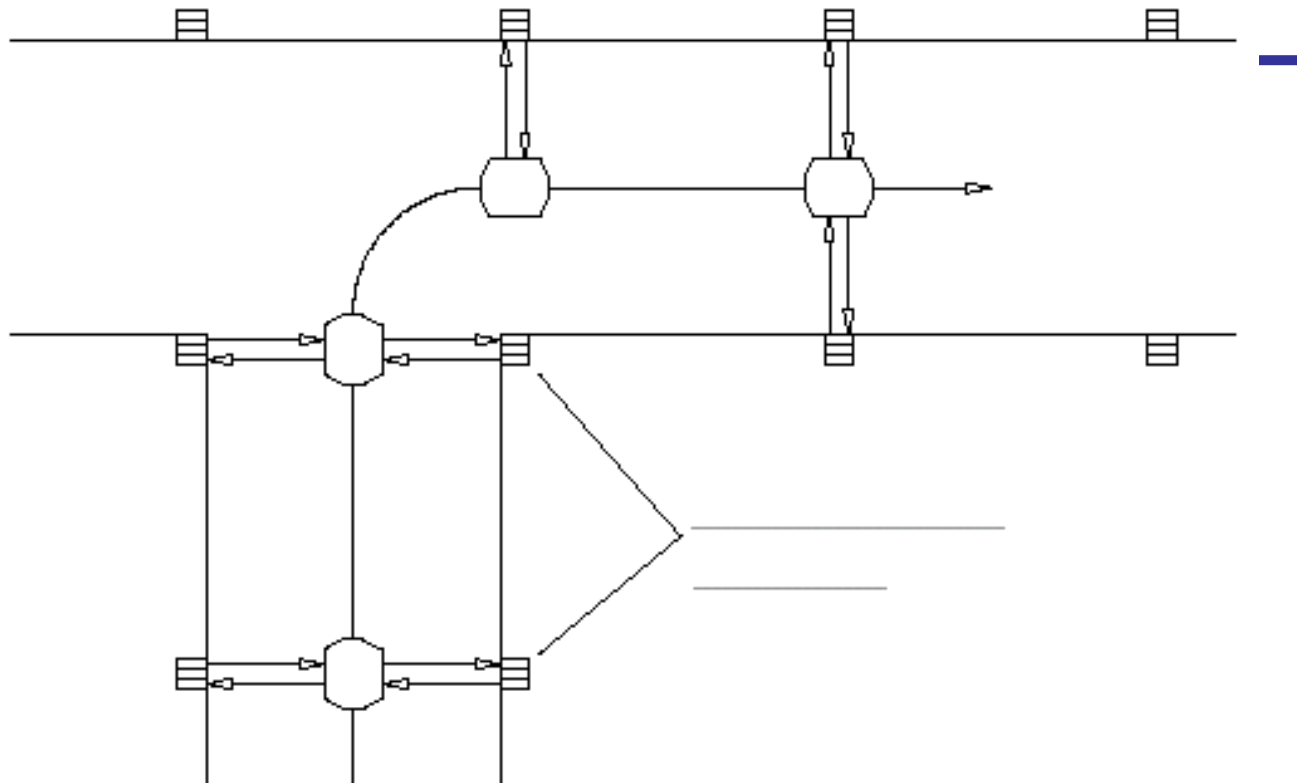
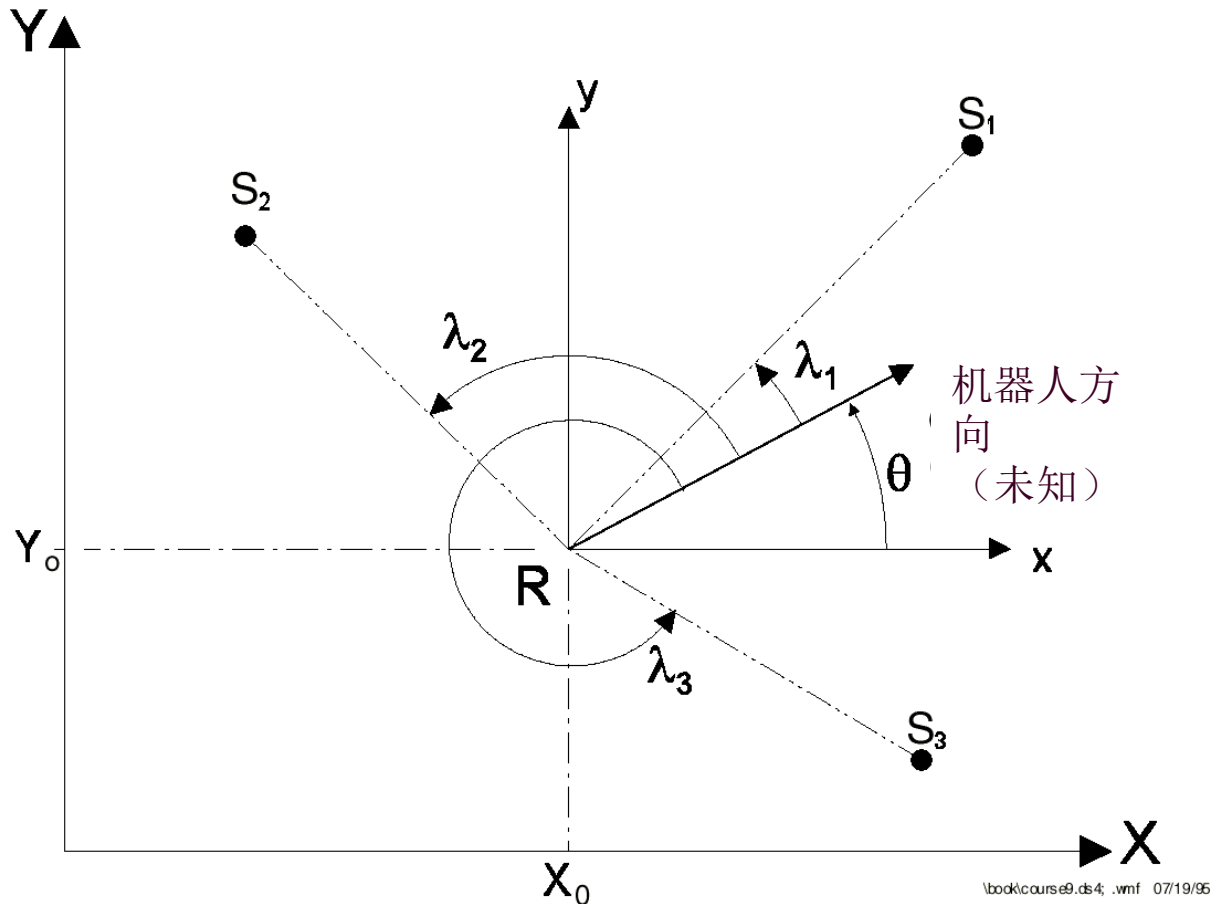


图7.5 在Camp Elliott仓库里，使用偏振型反射接近传感器对贴在棚架立柱上的反射带竖条定位



其他定位方法 (非概率方法)

定位灯塔系



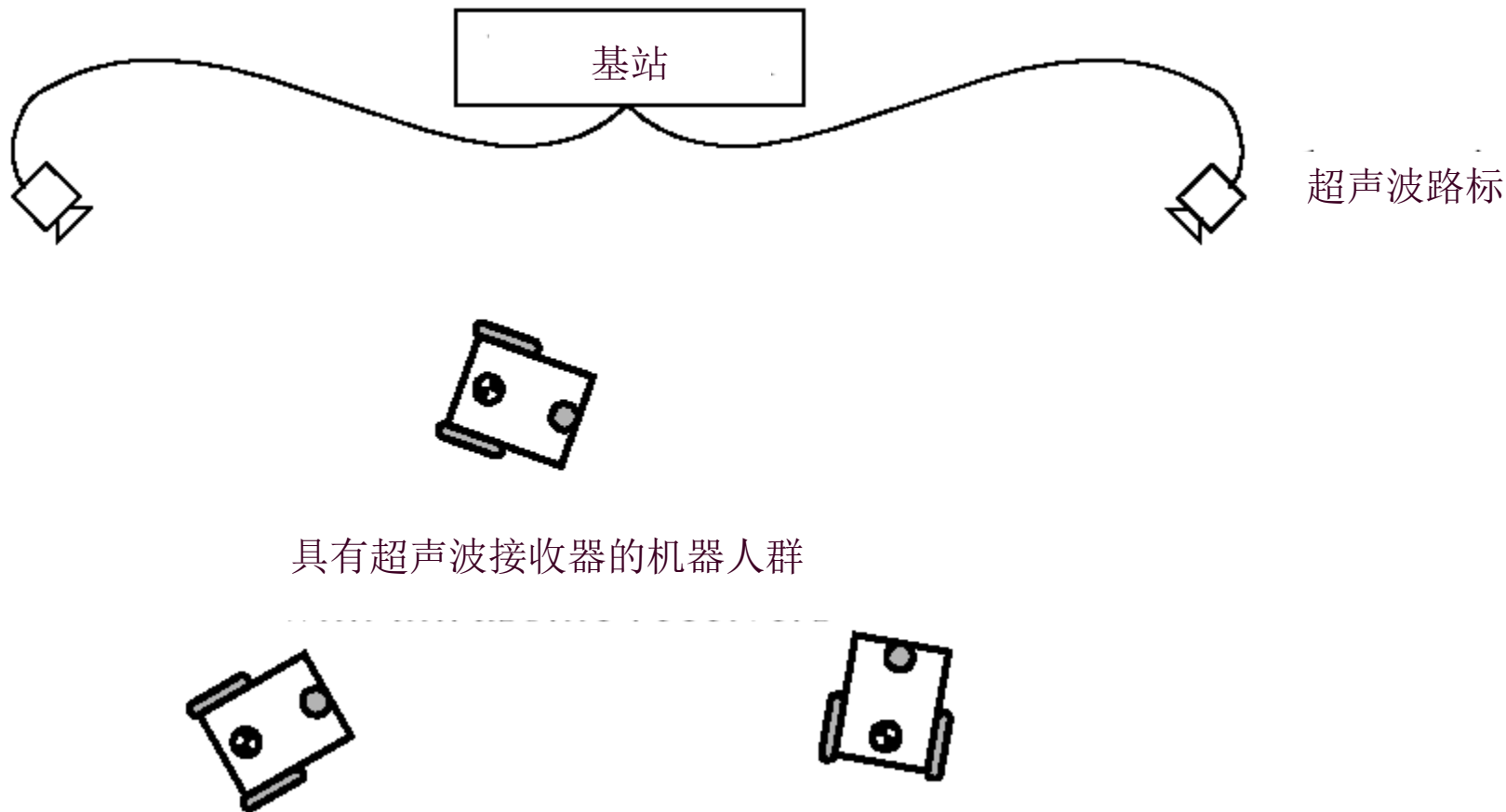
book\course9.ds4;.wmf 07/19/95

图6.1: 基本的三角测量问题: 一个旋转传感器头, 测量从车辆纵轴到三个源 S_1 , S_2 , S_3 之间的三个角度 λ_1 , λ_2 , λ_3



其他定位方法 (非概率方法)

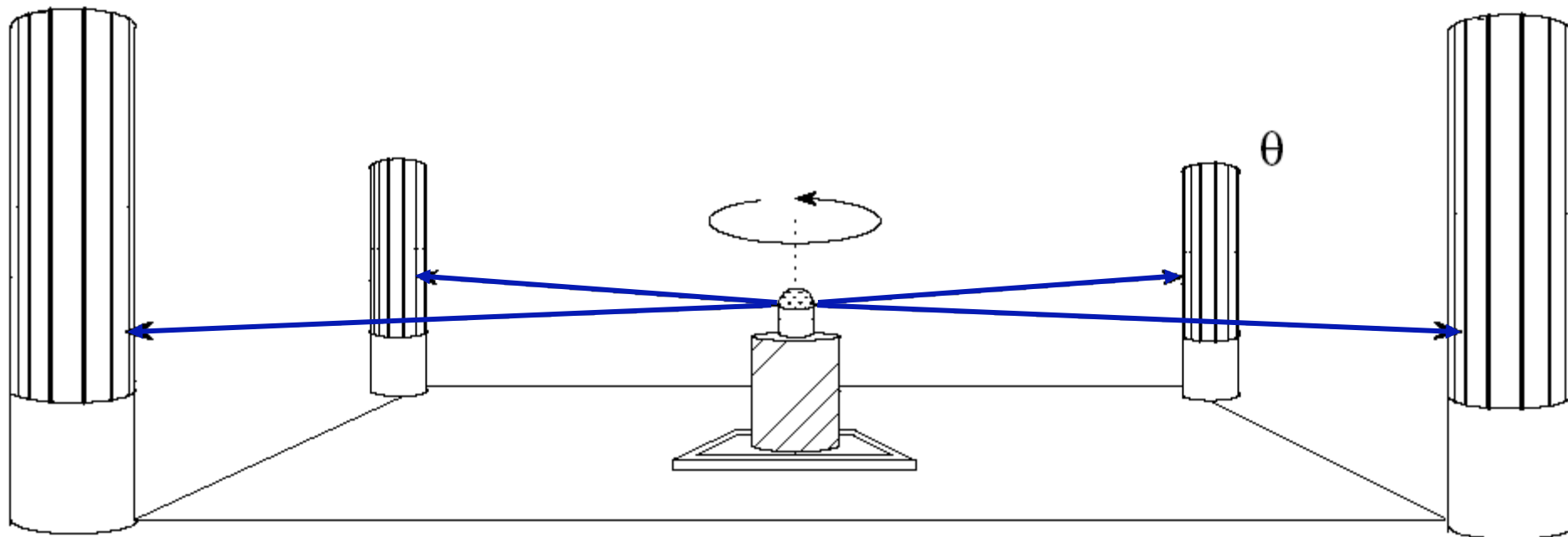
定位灯塔系统：三角测量





其他定位方法 (非概率方法)

定位灯塔系统：三角测量





其他定位方法 (非概率方法)

定位灯塔系

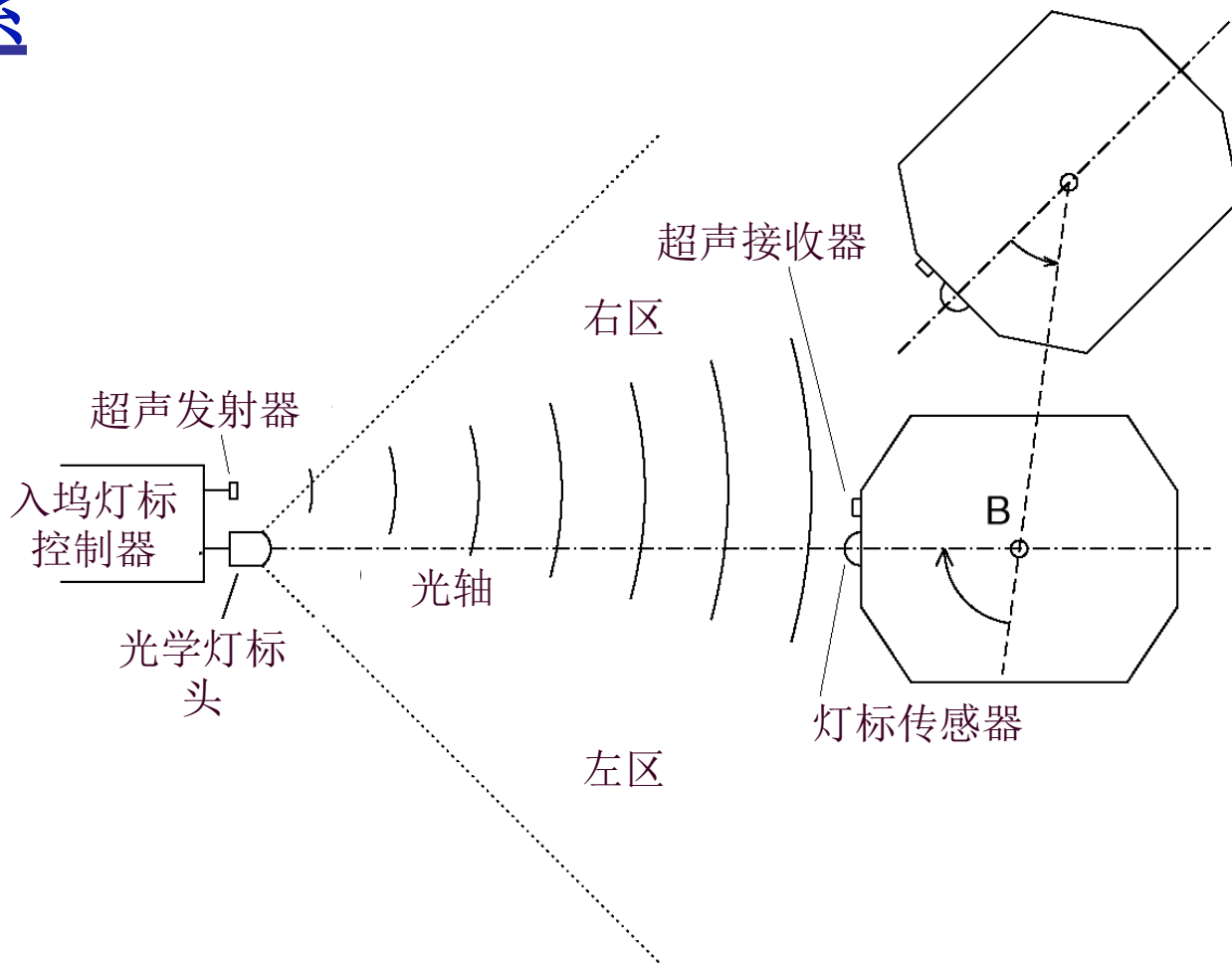


图6.6 用于充电站结构光近红外灯标灯标，对于K2A Navmaster机器人可以引导最优的接近路径。



其他定位方法 (非概率方法)

定位灯塔系统：条形码

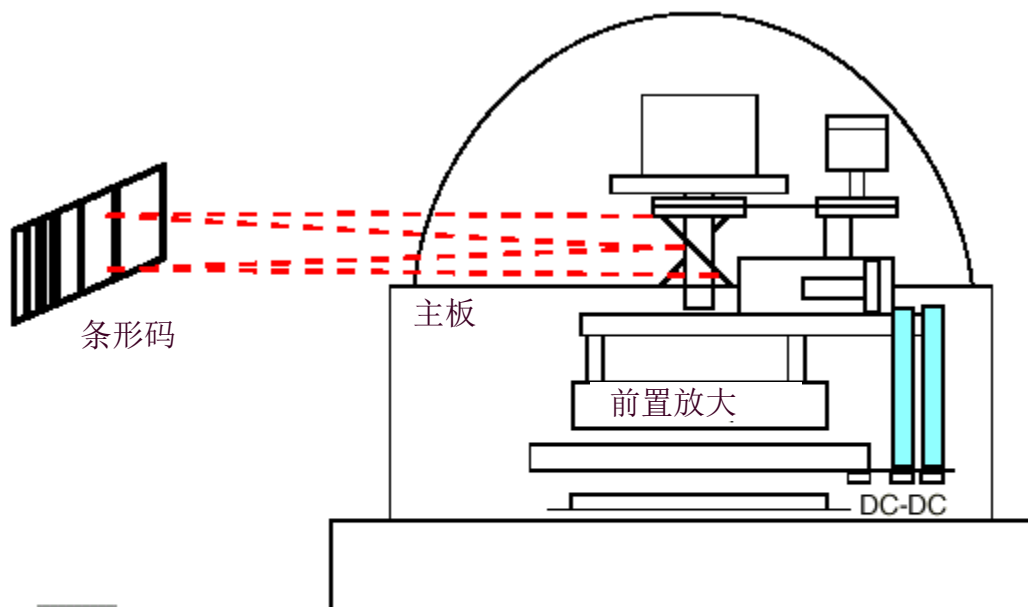


图6.14 国际Denning Branch机器人LaserNav型
基于激光扫描的灯塔系统草图

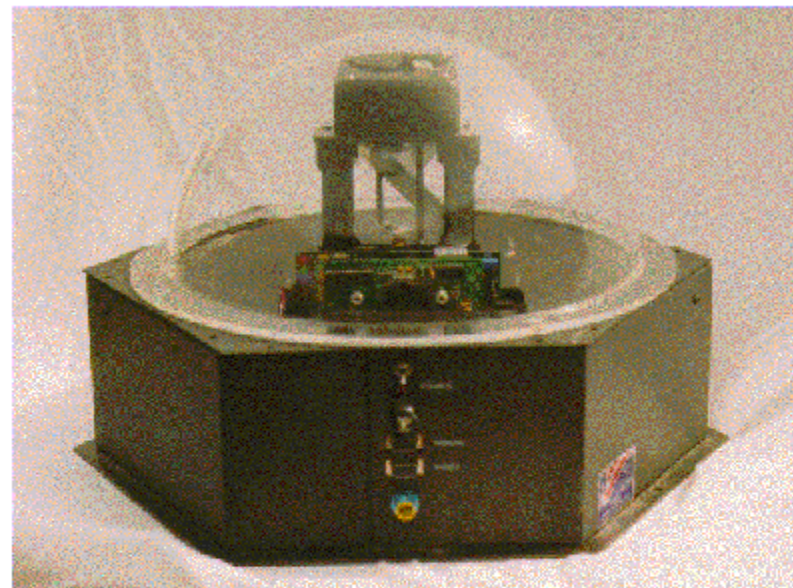


图6.15 国际Denning Branch机器人可以
看到183米以外主动目标。它最多可分辨
32个主动或被动目标



其他定位方法 (非概率方法) 定位灯塔系统.

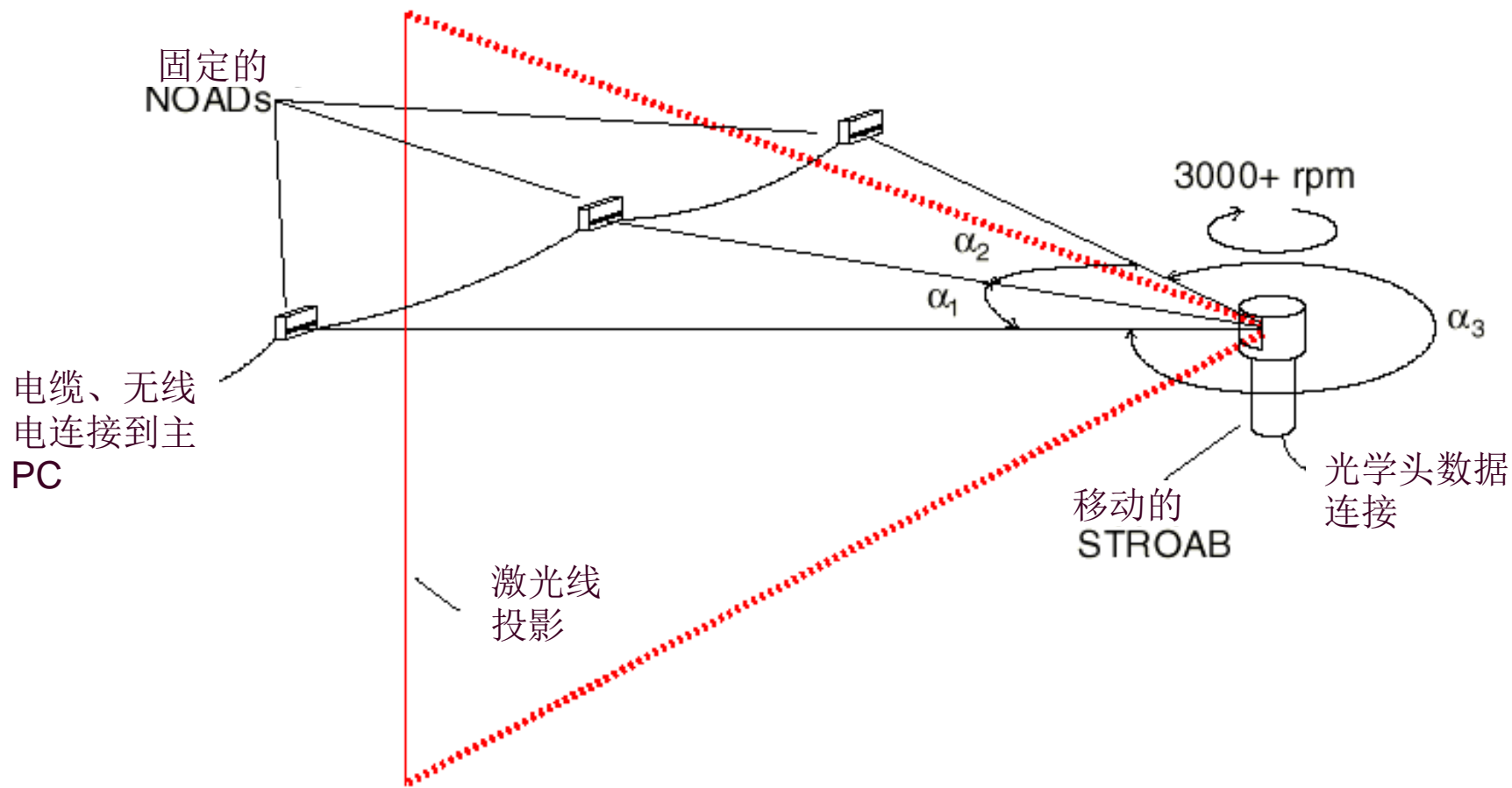


图6.21 电脑化的光电子导航和控制(CONAC™)系统，提供机载的快速旋转和竖直摆动激光束，连续地接触联网的探测器。



自主地图构建

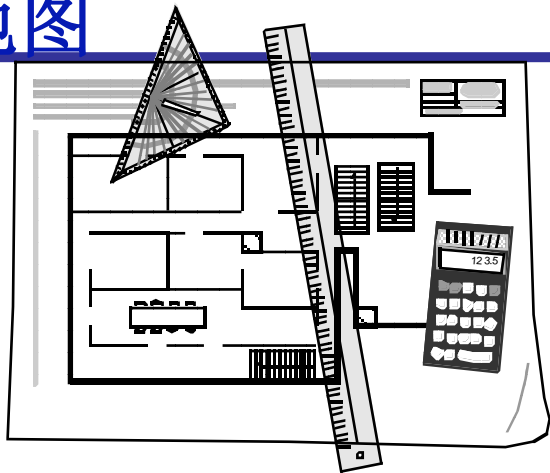
从一个任意的初始位置开始，
一台移动机器人应该能够用它的机载传感器自主地探索环境，
获得关于环境的知识，
解释场景，
构建合适的地图
并相对于地图将自己定位。

SLAM

同时定位和作图问题



构建地图： 如何建立地图



3. 对地图的基本需求:

- 具有将**新近感知**的信息合并到已有环境模型的手段
- 具有能够**估计机器人位置**的信息和程序
- 具有做**路径规划**和其他**导航任务**的信息 (例如, 避障)

- 地图质量的衡量

- 拓朴正确
- 度量正确



- 但是, 大多数环境都是可预测特征与不可预测特征的混合体→ 混合方法
基于模型 vs. 基于行为

2. 自动: 地图构建

机器人学习它所处的环境

动机:

- 人工: 困难, 代价高
- 环境动态变化
- 由不同的传感器, 得到不同的看法

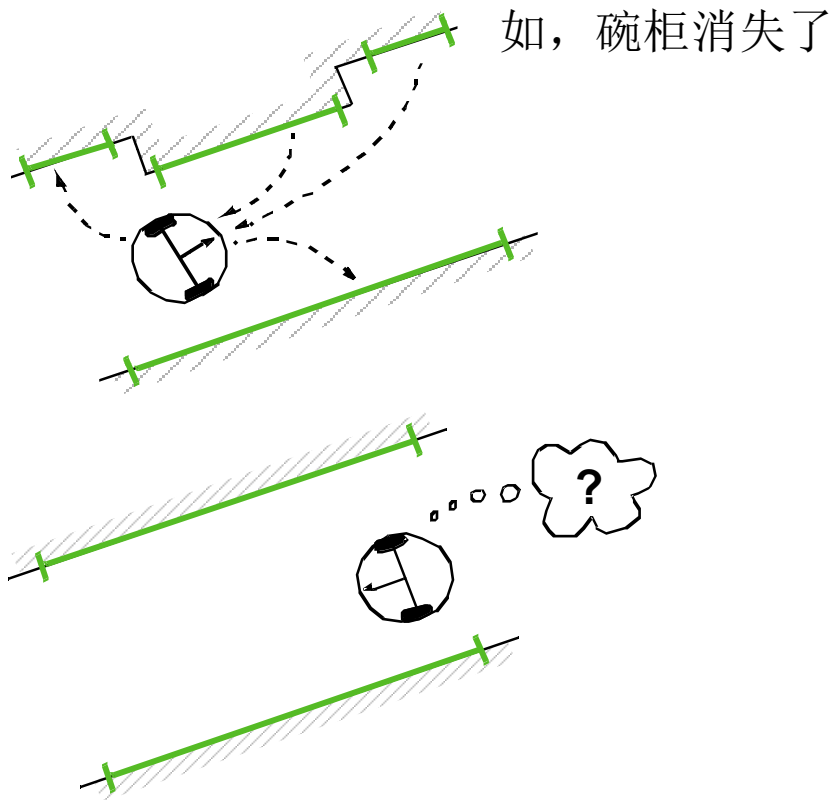


构建地图:

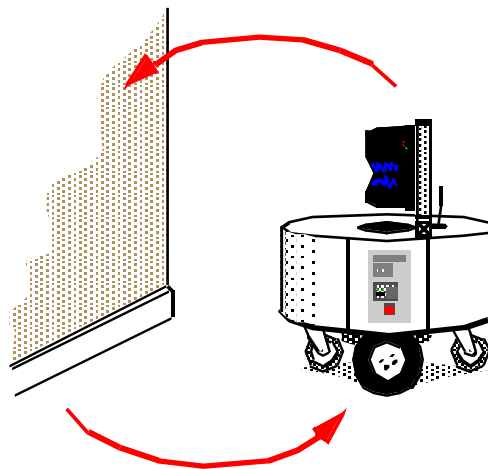
问题

地图维持: 对环境的变化保持跟踪

2. 不确定性的表示和减小



机器人位置 -> 墙的位置



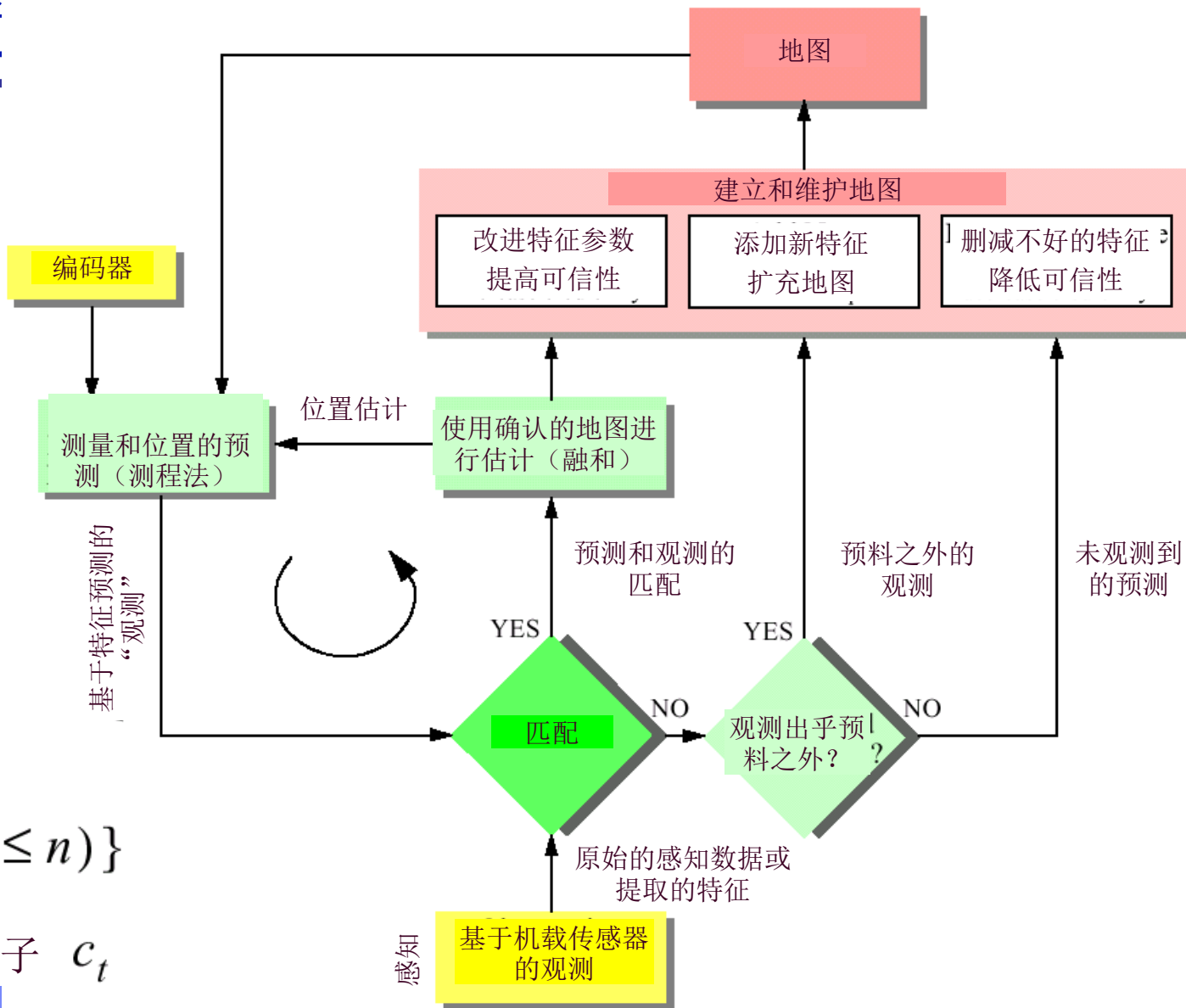
墙的位置 -> 机器人位置

- 如, 环境各特征的信任度测量

- 特征位置的概率密度
- 附加的探测策略



一般的建立地图过



$$M = \{ \hat{z}_t, \Sigma_t, c_t \mid (1 \leq t \leq n) \}$$

可信度因子 c_t



地图的表示方法

◇ M 是一个随机特征位置的集合 n

◇ 各特征可表示 $M = \{\hat{z}_t, \Sigma_t, c_t | (1 \leq t \leq n)\}$: 关联的可信度因子 c_t

◇ c_t 取值从 0 到 1 $c_t(k) = 1 - e^{-\left(\frac{n_s}{a} - \frac{n_u}{b}\right)}$ 环境中存在的信任度。

◇ a 和 b 分别定义了学习和遗忘速率, n_s 和 n_u 分别是到时



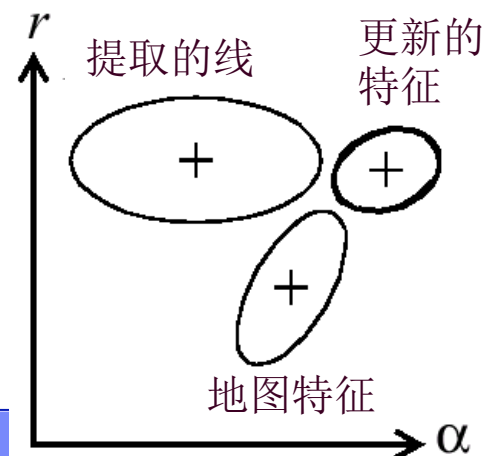
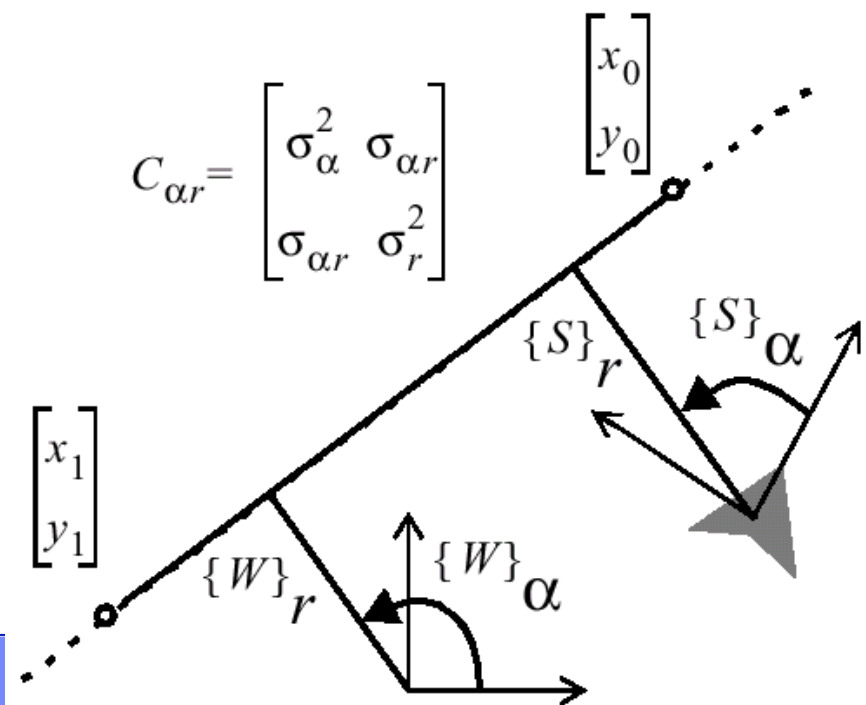
自动构建地图 随机构图技术

◆ 堆叠系统的状态向量:

$$X = \begin{bmatrix} x_r(k) & x_1(k) & x_2(k) & \dots & x_n(k) \end{bmatrix}^T$$

◆ 状态协方差矩阵:

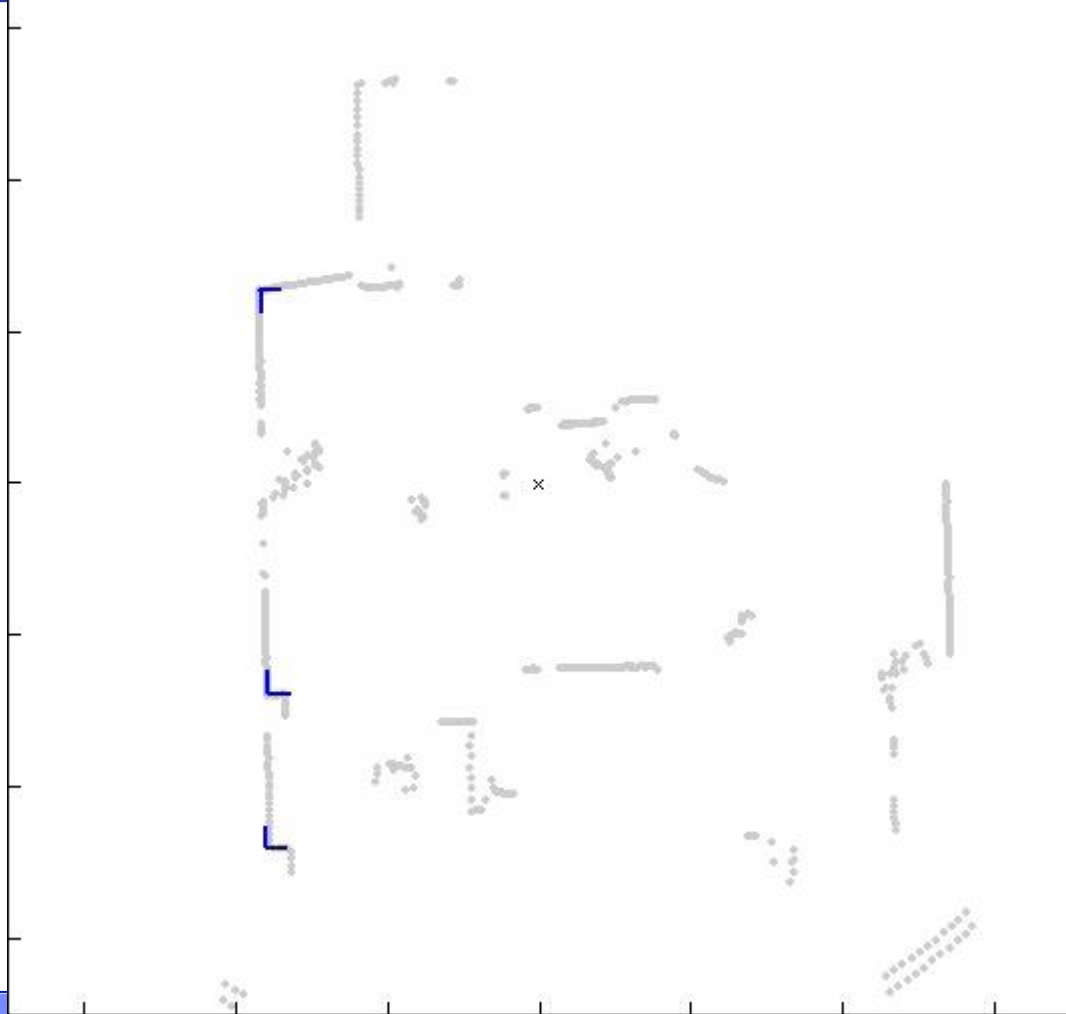
$$\Sigma = \begin{bmatrix} C_{rr} & C_{r1} & C_{r2} & \dots & C_{rn} \\ C_{1r} & C_{11} & \dots & \dots & C_{1n} \\ C_{2r} & \dots & \dots & \dots & C_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ C_{nr} & C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}$$





自动构建地图

例：基于特征的特征地图构建 (FFSL)

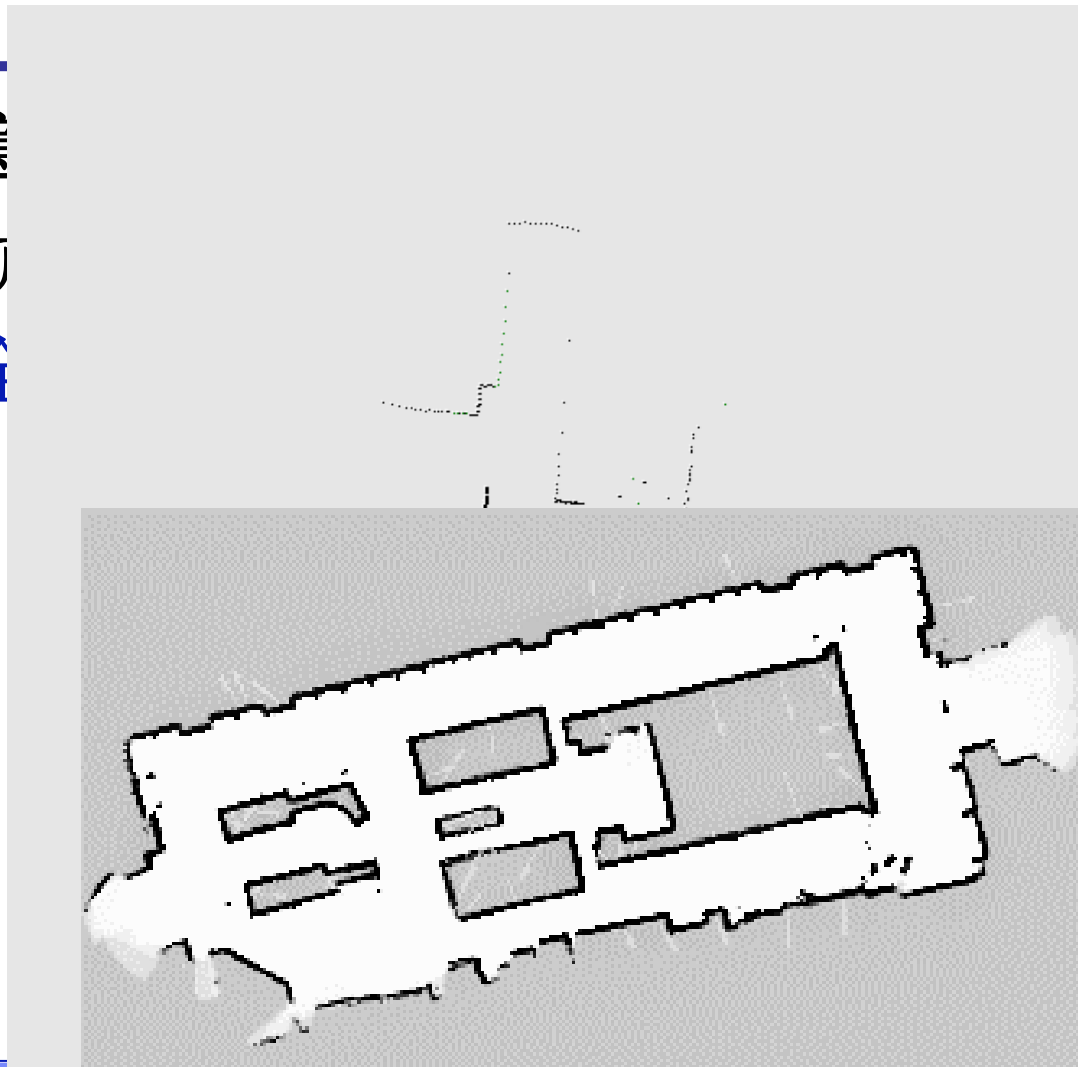
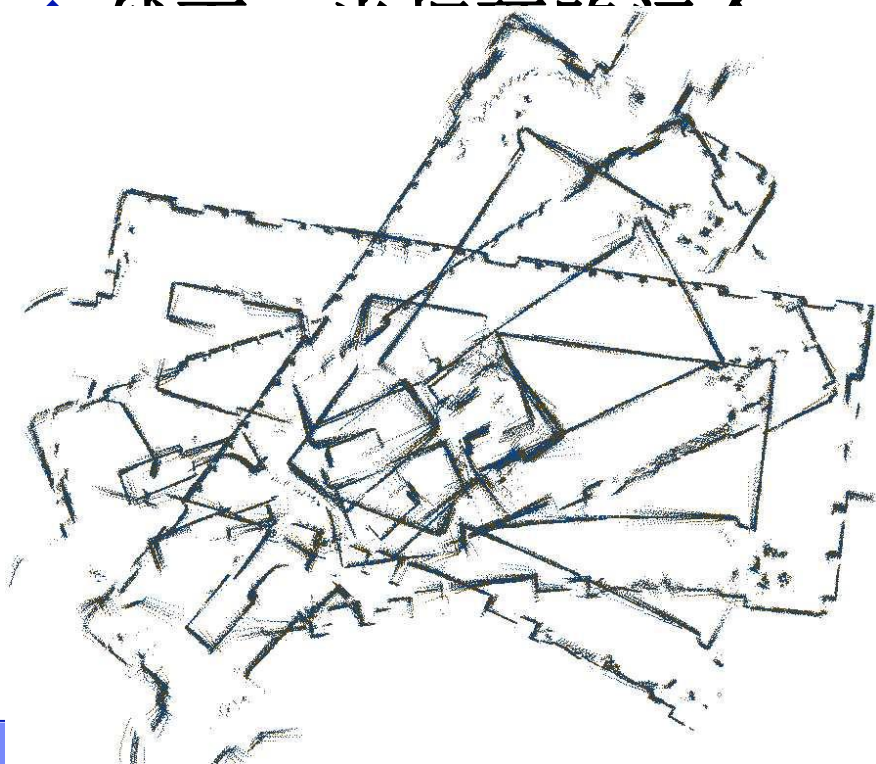




Courtesy of Sebastian Thrun

环状的环境

- ◇ 局部小误差，累积成任意
- ◇ 这种全局误差常常与导航





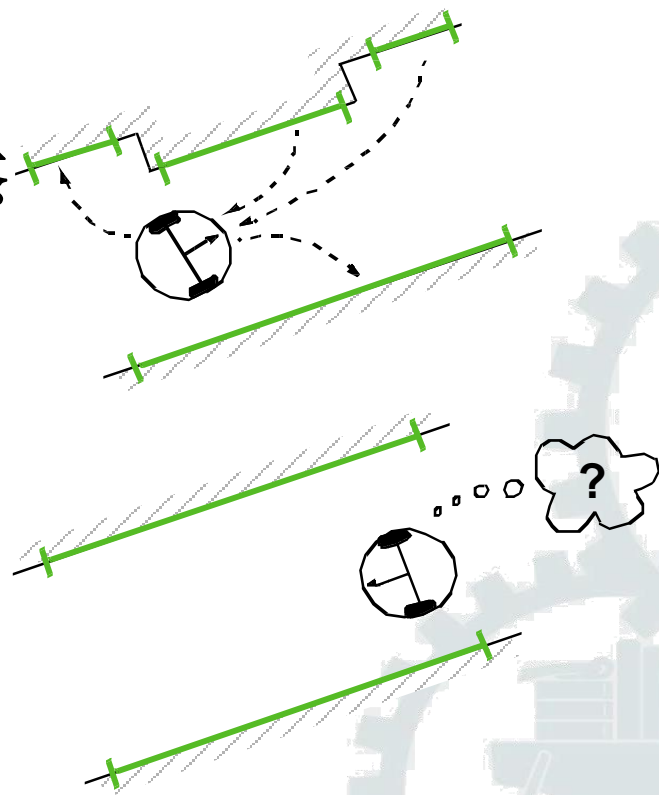
动态环境

◇ 环境动态变化，需要持续地构建地图

◇ 如果有可能提取高级别特征，在动态环境下构建地图就变得非常直截了当了。

◇ 例如，人和墙的差别

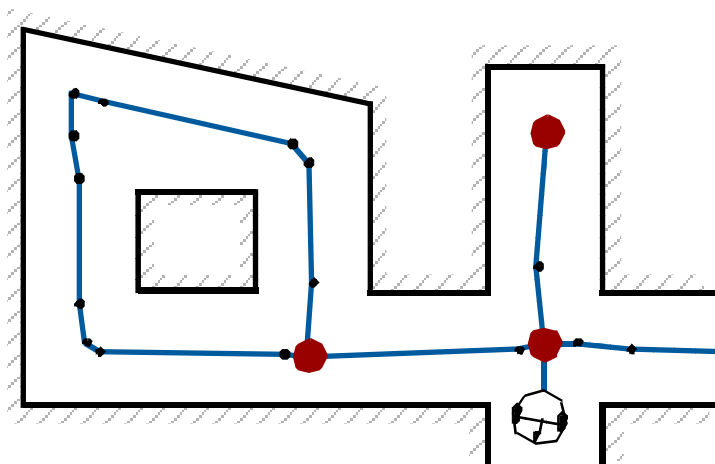
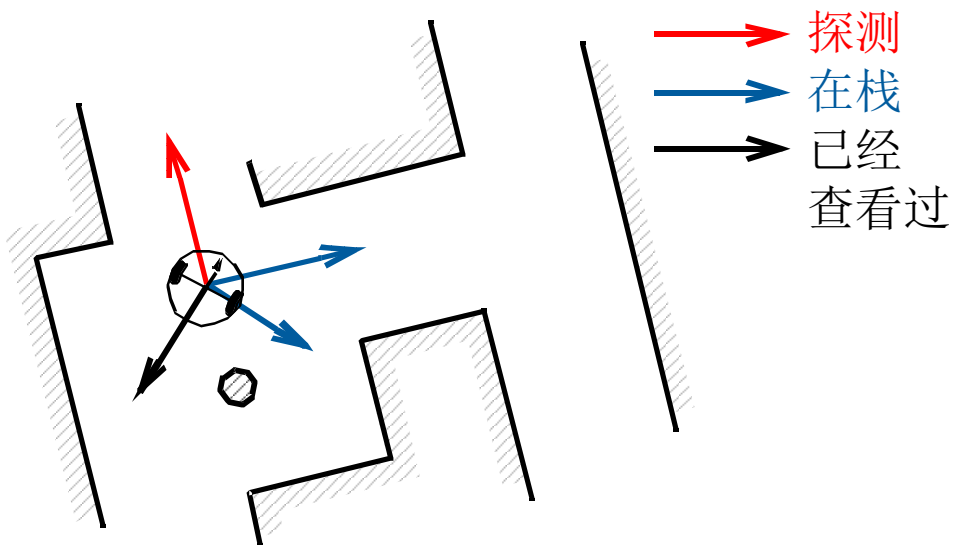
◇ 对于鲁棒性，环境建模是关键因素





构建地图： 勘察与图结构

2. 图结构



节点放在哪里？

- 基于拓扑：在有区别的位置上
- 基于度量：特征消失或变得可见的地方

- 提供正确的拓扑关系
- 必须能分辨已经访问过的位置
- 对未探查过的通路，原路返回