

不完整地图中移动机器人蒙特卡罗定位研究*

张 恒, 刘艳丽, 孙 晋

(华东交通大学 信息工程学院, 南昌 330013)

摘要: 针对移动机器人在不完整地图中定位的问题, 提出了一种改进的粒子聚类蒙特卡罗定位 (Monte Carlo localization, MCL) 算法。在定位过程中, 将机器人的位姿分为六种状态, 每一种状态对应一个粒子簇。在机器人运动的过程中, 这六种状态之间可以相互转移, 在计算状态转移概率的基础上, 实现了不完整地图中移动机器人蒙特卡罗定位算法。实验验证了该算法在解决移动机器人在不完整地图中定位问题的有效性。

关键词: 移动机器人; 蒙特卡罗定位; 状态转移; 不完整地图

中图分类号: TP24 **文献标志码:** A **文章编号:** 1001-3695(2010)02-0509-03

doi:10.3969/j.issn.1001-3695.2010.02.029

Mobile robot Monte Carlo localization in partial map

ZHANG Heng, LIU Yan-li, SUN Jin

(School of Information Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: In order to overcome the difficulty of a mobile robot to perform localization in partial map, this paper proposed an improved particle-clustered Monte Carlo localization (MCL) algorithm. During the process of localization, the robot's states were classed to 6 types, and one type of these was corresponding to a particle cluster. Based on computing the transition probability, realized a MCL algorithm in partial environment, which broke the restriction that the traditional MCL algorithm could only be used in the situation of complete map. Experiment results illustrate the validity of the proposed approach in solving problems of localization in partial map.

Key words: mobile robot; Monte Carlo localization (MCL); state transition; partial map

机器人定位问题分为两类: 位姿跟踪问题和全局定位问题。前者指机器人的初始位姿已知, 根据传感器的观测信息来补偿码盘的误差; 如果机器人的初始位姿未知, 或者在跟踪过程中出现了“绑架”现象 (即在未通知机器人的情况下改变机器人的位姿), 那么就成为了全局定位问题。近年来, 越来越多的研究者把概率理论应用到移动机器人定位中^[1]。对于解决全局定位问题, 最具有代表性的是蒙特卡罗定位 (MCL) 算法^[2] 和马尔可夫定位 (Markov localization) 算法^[3]。其中 MCL 算法具有实现简单、定位精度高、所需计算资源少等优点而受到研究者的青睐, 并对其提出了许多改进的方法, 其中具有代表性的有 Mixture-MCL^[4]、自适应采样 MCL^[5]、遗传粒子滤波定位^[6]、粒子聚类蒙特卡罗定位^[7]等。

现有的定位方法大多是在地图完整的前提下条件进行的, 即机器人的位姿总是在地图模型所对应的环境区域 (本文中简称地图区域) 中。但在有些情况下地图往往是不完整的, 如地图是由机器人自主构建而非人工创建的, 这种地图很难保证是完整覆盖工作空间的。随着移动机器人的自主性越来越高, 这种不完整地图的情况会越来越多。另外, 在多机器人同步定位与地图构建问题中, 可以利用不完整地图机器人定位技术实现地图的合并。为了使机器人可以在不完整地图中定位, 本文利用先前提出的粒子聚类蒙特卡罗定位算法^[7] 的基本思想对 MCL 算法进行了进一步扩展。将定位过程中机器人的位姿分为六种状态, 每一种状态对应一个粒子簇。与文献 [7] 中算法不同的是, 在机器人运动的过程中, 这六种状态之间可以转移, 在计算转移概率的基础上实现了不完整地图中移动机器人蒙

特卡罗定位算法。

1 问题分析

完整地图和非完整地图中机器人定位的区别在于: 对于前者, 机器人总是在地图区域内运动, 因此在蒙特卡罗定位过程中, 粒子总是落入地图内部; 而对于后者, 任何时刻机器人可能在地图区域内也可能在地图区域外。根据前一刻和当前时刻机器人位姿情况的不同组合, 可分为六种情况, 或称为机器人的六种状态, 如表 1 所示。

表 1 机器人六种状态分类

状态编号	前一刻机器人位置	当前时刻机器人位置	历史状况
1	地图区域外	地图区域外	从来就没有进入过地图区域
2	地图区域外	地图区域内	首次进入地图区域
3	地图区域内	地图区域内	—
4	地图区域内	地图区域外	—
5	地图区域外	地图区域外	曾经进入过地图区域
6	地图区域外	地图区域内	再次进入地图区域

记 t 时刻这六种状态的概率分别为 $B_{1,t}, B_{2,t}, \dots, B_{6,t}$, $\sum_{i=1}^6 B_{i,t} = 1$ 。本文的基本思路就是将这六种状态对应的粒子归为对应的六个粒子簇, 利用文献 [7] 介绍的粒子聚类蒙特卡罗算法来解决不完整地图机器人定位问题。下面介绍这六种情况粒子的生成和对应权值的计算方法。

收稿日期: 2009-05-04; 修回日期: 2009-06-16 基金项目: 江西省教育厅青年科学基金资助项目 (GJJ09507, GJJ09501)

作者简介: 张恒 (1979-), 男, 湖北汉川人, 讲师, 博士, 主要研究方向为移动机器人导航技术、机器视觉等 (hbzhangheng@sohu.com); 刘艳丽 (1979-), 女, 讲师, 硕士, 主要研究方向为人工智能、计算机视觉; 孙晋 (1988-), 男, 安徽安庆人, 本科。

2 新一代粒子生成

由于实际环境区域未知,无法对第一种情况生成有效的粒子,但是如果能计算 $B_{1,t}$,则并不影响在地图区域内的定位估计, $B_{1,t}$ 的具体计算方法见下文。为生成第二种情况的粒子,首先给出如下的定义和合理的假设:

定义 1 地图的探测边界(exploration frontiers)。是指已探测区域与为探测区域有通道的交界处,具体的就是地图已知区域与未知区域(地图外区域)的临界处,且已知区域这边为空闲。

假设 1 机器人只能从地图的探测边界进入地图区域。

于是,第二种情况的粒子在地图探测边界附近的地图空闲区域内按均匀分布生成,该区域的大小根据机器人的当前运动步长(当前时刻与前一个时刻机器人位姿差)决定,样本的个数 N_2 在本文中取一固定值。

在机器人运动过程中其状态之间可以转移,设 $p_{ij,t}$ 表示在 $t-1$ 时刻机器人处于第 i 种状态的情况下,由于机器人移动使得 t 时刻转为第 j 种状态的概率。分析具体的情况可以得出如图 1 所示的状态概率转移图。

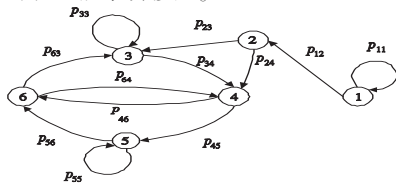


图 1 不完整地图中定位机器人状态转移图

根据图 1,新一代第 3 类粒子有三个来源:前一代第 2 类、第 3 类、第 6 类中部分粒子;新一代第 4 类粒子也有三个来源:前一代第 2 类、第 3 类、第 6 类中部分粒子;新一代第 5 类粒子有两个来源:前一代第 4 类、第 5 类中部分粒子;新一代第 6 类粒子有两个来源:前一代第 4 类、第 5 类中部分粒子。总之,采用前一代粒子和当前控制输入,根据运动模型生成新一代粒子,然后对它们重新分类,具体实现见算法 1。

3 权值计算

与文献[7]的粒子聚类蒙特卡罗定位算法一样,权值计算包括两部分:a)机器人处于各种状态的概率;b)各粒子簇中粒子的权值。由于存在状态转移,整个计算分为两个阶段,即预测阶段和环境观测阶段。

3.1 预测阶段

根据图 1,显然有如下一些等式成立:

$$\begin{aligned} p_{11,t} + p_{12,t} &= 1, \quad p_{23,t} + p_{24,t} = 1, \quad p_{33,t} + p_{34,t} = 1 \\ p_{45,t} + p_{46,t} &= 1, \quad p_{55,t} + p_{56,t} = 1, \quad p_{63,t} + p_{64,t} = 1 \end{aligned} \quad (1)$$

在 t 时刻机器人对环境观测之前,各种状态的概率可根据 $t-1$ 时刻的值递推得到:

$$B'_{1,t} = B_{1,t-1} \cdot p_{11,t} \quad (2)$$

$$B'_{2,t} = B_{1,t-1} \cdot p_{12,t} \quad (3)$$

$$B'_{3,t} = B_{2,t-1} \cdot p_{23,t} + B_{3,t-1} \cdot p_{33,t} + B_{6,t-1} \cdot p_{63,t} \quad (4)$$

$$B'_{4,t} = B_{2,t-1} \cdot p_{24,t} + B_{3,t-1} \cdot p_{34,t} + B_{6,t-1} \cdot p_{64,t} \quad (5)$$

$$B'_{5,t} = B_{4,t-1} \cdot p_{45,t} + B_{5,t-1} \cdot p_{55,t} \quad (6)$$

$$B'_{6,t} = B_{4,t-1} \cdot p_{46,t} + B_{5,t-1} \cdot p_{56,t} \quad (7)$$

要计算上述各式,需确定转移概率 $p_{12,t}, p_{23,t}, p_{34,t}, p_{45,t},$

$p_{56,t}$ 和 $p_{63,t}$ (或 $p_{11,t}, p_{24,t}, p_{33,t}, p_{46,t}, p_{55,t}$ 和 $p_{64,t}$) 的值。由于 $p_{12,t}$ 的值无法计算,可以给定一常数(一般取一较小的值,本文中取 0.1)。设 $t-1$ 时刻,第 2~6 类粒子簇为 $\{(C_{k,t-1}, B_{k,t-1}) \mid k=2, \dots, 6\}$ 。其中, $C_{k,t-1} = \{\langle \xi_{k,t-1}^{(j)}, w_{k,t-1}^{(j)} \rangle \mid j=1, 2, \dots, N_{k,t-1}\}, \sum_{j=1}^{N_{k,t-1}} w_{k,t-1}^{(j)} = 1$; 记 $\xi'_{k,t}$ 为根据运动模型得到的 $t-1$ 时刻第 k 类粒子簇中第 j 号粒子的预测值,即 $\xi'_{k,t} \sim p(\xi_j \mid \xi_{k,t-1}, u_t)$ 。根据 $\xi'_{k,t}$ 是否在地图区域中和其前驱样本 $\xi_{k,t-1}^{(j)}$ 的类别号 k ,可以确定新样本 $\xi'_{k,t}$ 的类别,最终得到新的粒子簇 $\{(C_{k,t}, B_{k,t}) \mid k=2, \dots, 6\}$ 。其他转移概率计算方法如下(这里 $\xi \in M$ 表示 ξ 处于地图 M 的表示区域中,否则为 $\xi \notin M$):

$$p_{23,t} = \frac{\sum_{i=1}^{N_{2,t-1}} \xi'_{2,t} \in M w_{2,t-1}^{(i)}}{\sum_{i=1}^{N_{2,t-1}} w_{2,t-1}^{(i)}} \quad (8) \quad p_{34,t} = \frac{\sum_{i=1}^{N_{3,t-1}} \xi'_{3,t} \notin M w_{3,t-1}^{(i)}}{\sum_{i=1}^{N_{3,t-1}} w_{3,t-1}^{(i)}} \quad (9)$$

$$p_{45,t} = \frac{\sum_{i=1}^{N_{4,t-1}} \xi'_{4,t} \in M w_{4,t-1}^{(i)}}{\sum_{i=1}^{N_{4,t-1}} w_{4,t-1}^{(i)}} \quad (10) \quad p_{56,t} = \frac{\sum_{i=1}^{N_{5,t-1}} \xi'_{5,t} \in M w_{5,t-1}^{(i)}}{\sum_{i=1}^{N_{5,t-1}} w_{5,t-1}^{(i)}} \quad (11)$$

$$p_{63,t} = \frac{\sum_{i=1}^{N_{6,t-1}} \xi'_{6,t} \in M w_{6,t-1}^{(i)}}{\sum_{i=1}^{N_{6,t-1}} w_{6,t-1}^{(i)}} \quad (12)$$

记 $w'_{k,t}$ 为样本 $\xi'_{k,t}$ 的全局权值(用系统中所有样本作为普通粒子滤波器的样本集时各样本的权值称为该样本的全局权值)。在这一阶段,由于没有考虑机器人对环境的观测信息,新一代粒子的全局权值不变:

$$w'_{k,t} = w_{k,t-1} \cdot B_{k,t-1} \quad (13)$$

3.2 环境观测阶段

机器人对环境观测后,机器人所处各状态的概率和粒子的权值需进一步更新。对于新一代粒子中 2、3、6 这三类粒子,机器人观测信息可与地图进行匹配,权值的更新方法与完整地图中蒙特卡罗定位算法类似;对于第 4、5 类粒子,观测信息无法利用,粒子权值不用再更新,即

$$w_{k,t} = \begin{cases} w'_{m,t} \cdot p(z_t \mid \xi_{k,t}^{(i)}) & \text{if } k=2,3,6 \\ w'_{m,t} & \text{if } k=4,5 \end{cases} \quad (14)$$

在计算 $B_{k,t}$ 之前,先对 $C_{k,t} (k=2, \dots, 6)$ 中粒子的权值 $w_{k,t}$ 作归一化处理。根据文献[7]中的结论, $B_{k,t}$ 的计算方法如下:

$$B_{1,t} = B'_{1,t} \quad (15)$$

$$B_{2,t} = \frac{\alpha_2 \cdot (B'_{2,t} + B'_{3,t} + B'_{6,t})}{\alpha_2 + \alpha_3 + \alpha_6} \quad (16)$$

$$B_{3,t} = \frac{\alpha_3 \cdot (B'_{2,t} + B'_{3,t} + B'_{6,t})}{\alpha_2 + \alpha_3 + \alpha_6} \quad (17)$$

$$B_{4,t} = B'_{4,t} \quad (18)$$

$$B_{5,t} = B'_{5,t} \quad (19)$$

$$B_{6,t} = \frac{\alpha_6 \cdot (B'_{2,t} + B'_{3,t} + B'_{6,t})}{\alpha_2 + \alpha_3 + \alpha_6} \quad (20)$$

其中:

$$\alpha_2 = B'_{2,t} \cdot \sum_{(\xi_{2,t}^{(i)}, w_{2,t}^{(i)}) \in C_{2,t}} (w_{2,t}^{(i)} \cdot p(z_t \mid \xi_{2,t}^{(i)})) \quad (21)$$

$$\alpha_3 = B'_{3,t} \cdot \sum_{(\xi_{3,t}^{(i)}, w_{3,t}^{(i)}) \in C_{3,t}} (w_{3,t}^{(i)} \cdot p(z_t \mid \xi_{3,t}^{(i)})) \quad (22)$$

$$\alpha_6 = B'_{6,t} \cdot \sum_{(\xi_{6,t}^{(i)}, w_{6,t}^{(i)}) \in C_{6,t}} (w_{6,t}^{(i)} \cdot p(z_t \mid \xi_{6,t}^{(i)})) \quad (23)$$

最后,对 $\{B_{k,t} \mid k=1, \dots, 6\}$ 作归一化处理。具体的不完整地图蒙特卡罗定位算法如算法 1。

算法 1 不完整地图蒙特卡罗定位算法

输入: $t-1$ 时刻的粒子簇 $\{(C_{k,t-1}, B_{k,t-1}) \mid k=2, \dots, 6\}$, 控制输入 u_t , 机器人对环境的观测 z_t , 机器人在 $t-1$ 时刻处于状态 1 的情况下在 t 时刻变为状态 2 的概率 p_{12} 。

输出: t 时刻新的粒子簇 $\{(C_{k,t}, B_{k,t}) \mid k=2, \dots, 6\}$ 。

/* 在预测阶段第 k 类粒子簇预测生成 N_k 个新一代粒子 */

```

1 for k=2 to 6 do Ck,t置空;
2 for i=1 to N2 do
3   在地图探测边界区域随机生成一位姿样本 ξ2,t(i)
4   w2,t(i) = p(zt | ξ2,t(i)); C2,t = C2,t ∪ { < ξ2,t(i), w2,t(i) > };
5 end
6 p23,t = 0; p34,t = 0; p45,t = 0; p56,t = 0; p63,t = 0;
7 for k=2 to 6 do
8   for i=1 to Nk do
9     从 Ck,t-1 根据权值 { wk,t-1(i) } 随机抽取一样本 ξk,t-1(n)
10    ξk,tr(i) ~ p(ξk,t | ξk,t-1(n), ut); wk,tr(i) = 1/Nk;
11  switch k do
12    case 2
13      if ξk,tr(i) ∈ M then
14        C3,t = C3,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
15        p23,t = p23,t + wk,tr(i);
16      else
17        C4,t = C4,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
18      case 3
19        if ξk,tr(i) ∈ M then
20          C3,t = C3,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
21          else
22            C4,t = C4,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
23            p34,t = p34,t + wk,tr(i);
24          case 4
25            if ξk,tr(i) ∈ M then
26              C6,t = C6,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
27            else
28              C5,t = C5,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
29              p45,t = p45,t + wk,tr(i);
30            case 5
31              if ξk,tr(i) ∈ M then
32                C6,t = C6,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
33                p56,t = p56,t + wk,tr(i);
34              else
35                C5,t = C5,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
36            case 6
37              if ξk,tr(i) ∈ M then
38                C3,t = C3,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
39                p63,t = p63,t + wk,tr(i);
40              else
41                C4,t = C4,t ∪ { < ξk,tr(i), wk,tr(i) • Bk,t-1 > };
42            end
43          end
44        end
45        B'1,t = B1,t-1 • (1 - p12); B'2,t = B1,t-1 • p12;
46        B'3,t = B2,t-1 • p23,t + B3,t-1 • (1 - p34,t) + B6,t-1 • p63,t;
47        B'4,t = B2,t-1 • (1 - p23,t) + B3,t-1 • p34,t + B6,t-1 • (1 -
48        p63,t);
49        B'5,t = B4,t-1 • p45,t + B5,t-1 • (1 - p56,t); B'6,t = B4,t-1 •
50        (1 - p45,t) + B5,t-1 • p56,t;
51        for i=1 to N2, do w2,t(i) = w2,t(i) • p(zt | ξ2,t(i));
52        for i=1 to N3, do w3,t(i) = w3,t(i) • p(zt | ξ3,t(i));
53        for i=1 to N6, do w6,t(i) = w6,t(i) • p(zt | ξ6,t(i));
54        for k=2 to 6 do Ck,t 中粒子权值归一化;
55        根据式(15)~(23)计算 Bk,t;
56        返回 { Ck,t, Bk,t } | k=2, ..., 6 ;

```

4 实验及结果分析

针对 PlayerStage (<http://playerstage.sourceforge.net>) 移动机器人软件平台,笔者编写了一个新的 player 插件程序(pmcl)实现了本文提出的不完整地图蒙特卡罗定位算法。

实验之前,需得到环境的不完整地图。用笔者设计开发的改进粒子滤波同步定位与地图构建(simultaneous localization and mapping, SLAM)算法^[8]得到一个模拟环境(实际上为一完整栅格地图,如图2(a)所示)的不完整地图,如图2(b)所示。与文献[8]不同的是,该 SLAM 算法也实现为一个 player 插件程序(rbpfslam)。利用 playerv 工具控制机器人在模拟环境中的运动并不断对环境进行观测,rbpfslam 插件程序将模拟

的控制输入信息和观测信息作为数据源,在线地实现机器人的定位和地图的构建,也可将数据序列保存为 log 文件离线地执行 SLAM。这里,将 SLAM 后的栅格地图作为本文实验的不完整地图。实验过程与刚才的 SLAM 实验差不多,不同的是将 rbpfslam 插件(实际上是 cfg 文件中设定的机器人驱动器)换成根据本文算法实现的 pmcl 插件,模拟环境不变。同样利用 playerv 工具控制机器人在模拟环境中运动并不断地对环境进行观测,pmcl 插件程序将模拟的控制输入信息和观测信息作为数据源,在线地实现机器人在不完整地图中的全局定位,定位过程如图2所示。图中只显示了第2类粒子(边界处)和第3类粒子(地图内部)。

对比图2(c)和(d)可以看出,开始定位时,第2类粒子均匀地分布在地图内部区域;在定位过程中,第2类粒子逐渐聚集于若干簇,也就是若干个位姿假设。与完整地图定位不同的是,实际的机器人位姿不一定对应于这些位姿假设中的一个,如图2(e)和(f)所示,当机器人处在地图范围之外就不对应了。但这并不影响机器人在后面过程的定位,因为在整个定位过程中不断地生成第3类粒子,一旦机器人进入地图区域,刚生成的第3类粒子将部分转换为第2类粒子并获得较大的权值,如图2(g)和(h)所示。

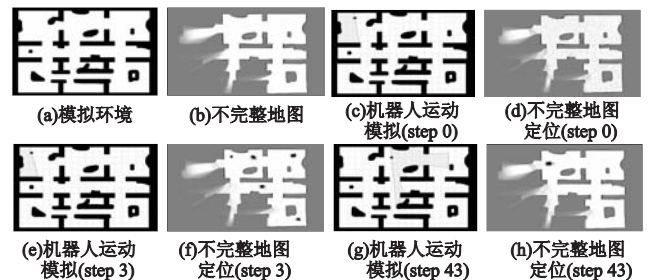


图2 不完整地图蒙特卡罗定位实验结果

5 结束语

本文利用粒子聚类蒙特卡罗定位算法的基本思想对 MCL 算法进行了进一步扩展。将定位过程中机器人的位姿分为六种状态,每一种状态对应一个粒子簇,从而可以运用粒子聚类蒙特卡罗定位算法。与文献[7]不同的是,这六种状态之间可以转移,也就是说各粒子簇不是完全独立地对应于一个基本的 MCL 过程。在计算转移概率的基础上,实现了新一代粒子生成方法和粒子权值计算方法。实验结果初步证明了该算法的有效性。

参考文献:

- [1] 厉茂海, 洪炳熔. 移动机器人的概率定位方法研究进展[J]. 机器人, 2005, 27(4): 380-384.
- [2] FOX D, BURGARD W, DELLAERT F, et al. Monte Carlo localisation: efficient position estimation for mobile robots[C]//Proc of the National Conference on Artificial Intelligence. Menlo Park: AAAI Press, 1999:343-349.
- [3] FOX D. Markov localization: a probabilistic framework for mobile robot localization and navigation[D]. Bonn, Germany: University of Bonn, 1998.
- [4] THRUN S, FOX D, BURGARD W, et al. Robust Monte Carlo localization for mobile robots[J]. Artificial Intelligence, 2001, 128(1, 2): 99-141.
- [5] JIANG Zheng-wei, GU Yuan-tao. Novel adaptive particle filters in robot localization[J]. Acta Automatica Sinica, 2005, 31(6): 833-838.
- [6] LUO Rong-hua, HONG Bing-rong, PIAO Song-hao, et al. Coevolution-based adaptive particle filters for global localization[J]. Chinese Journal of Electronics, 2005, 14(3): 458-462.
- [7] 张恒, 樊晓平, 翟志华. 基于多假设跟踪的移动机器人自适应蒙特卡罗定位研究[J]. 自动化学报, 2007, 33(9): 941-946.
- [8] 张恒, 刘艳丽. 基于固定滞后 Gibbs 采样粒子滤波的移动机器人 SLAM[J]. 计算机应用研究, 2008, 25(11): 3292-3295.