

# 可移动机器人的马尔可夫自定位算法研究<sup>1)</sup>

吴庆祥<sup>1</sup> Bell David<sup>2</sup>

<sup>1</sup>(福建师范大学网络中心 福州 350007)

<sup>2</sup>(School of Information and Software Engineering, University of Ulster at Jordanstown, BT37 0QB, UK)

(E-mail: q. wu@ulst. ac. uk; qxwu@fjtu. edu. cn)

**摘要** 马尔可夫定位算法是利用机器人运动环境中的概率密度分布进行定位的方法. 使用该方法机器人可在完全不知道自己位置的情况下通过传感器数据和运动模型来估计自己的位置. 但是, 在研究中发现它还存在着一些问题, 如概率减小到零后就无法恢复. 对只有距离传感器的机器人在对称的环境中仅仅采用该算法就无法确定位置. 为了解决这些问题, 文中给出了修正算法, 并建议在机器人上装上方向仪(如指南针或陀螺仪等), 然后利用定义的一个角度高斯分布函数来构造新的机器人感知模型. 在此基础上详细地阐述了一种新的自定位技术. 最后, 采用仿真程序验证了机器人在对称环境中运动时这一新算法的可行性.

**关键词** 机器人定位, 马尔可夫算法, 对称环境

**中图分类号** TP242

## A Study on Markov Localization for Mobile Robots

WU Qing-Xiang<sup>1</sup> Bell David<sup>2</sup>

<sup>1</sup>(Network Information Center, Fujian Teachers University, Fuzhou 350007)

<sup>2</sup>(School of Information and Software Engineering, University of Ulster at Jordanstown, BT37 0QB, UK)

(E-mail: q. wu@ulst. ac. uk; qxwu@fjtu. edu. cn)

**Abstract** The Markov localization algorithm is a means of estimating position of a mobile robot using a probability density over the environment of the robot's moving. By means of sensory data and motion model, it can be used to estimate robot's position under global uncertainty. However, some problems are found in our study. For example, the probability density cannot be recovered when it decreases to zero. A robot with only distance sensors cannot find its position in a symmetrical environment by means of Markov localization algorithm alone. In order to solve these problems a modified Markov localization algorithm is presented, and an approach in which a robot is equipped with a compass or gyroscope, has been proposed. An angle Gaussian distribution defined in this paper is used to construct a new perceptual model for the robot and the new localization technique based on these ideas is thoroughly presented. A simulation program is

1) 英国英中文化交流委员会基金(ALCS:CTN/992/244)及福建省高新科技项目(99-H38)资助

Supported by British Council for Academic Link China Scheme(ALCS) and Fujian Science and Technology Committee(99-H38)

收稿日期 2001-07-16 收修改稿日期 2002-01-21

Received July 16, 2001; in revised form January 21, 2002

used to demonstrate the effectiveness of the new technique for a robot moving in a symmetrical environment.

**Key words** Mobile robot, Markov localization, symmetrical environment

## 1 引言

机器人的智能定位问题是可移动机器人研究的重要课题<sup>[1,2]</sup>,常常涉及到位置的估算和位置控制两个方面,是当前非常活跃的研究领域.目的是通过给定的地图和传感器数据来估计机器人在它活动环境中的位置.在文献[3]中,可以看到许多不同方法的评述.自定位问题常常从两个方面入手.1)位置追踪,即机器人移动时的位置估算问题.漂移与滑动使得机器人难以通过全局地图来估计其位置,甚至很小的计程误差可以导致完全迷失方向.追踪或者说局部定位技术的目的在于补偿机器人导航过程中的计程错误.这种技术需要知道机器人的起始位置,并且在它失去追踪后无法恢复.2)全局定位技术可在没有任何先前位置信息的情况下,确定机器人的位置.这种技术可以解决所谓唤醒机器人问题.当机器人被偷偷地移到一个任意位置时,这种技术可以处理这种拐骗机器人问题.因此,比位置追踪更有效.它可以应付机器人遇到严重定位错误.马尔可夫定位就是一种全局定位技术.并且被成功地应用于德国波恩 Deutsches 博物馆的导航机器人 Rhino<sup>[4]</sup>和美国国家历史博物馆的导航机器人 Minerva.结合商滤波器和距离滤波器,马尔可夫定位算法还可被应用于动态环境中<sup>[5]</sup>.

## 2 Markov 自定位算法

可移动机器人的位置通常用三维变量  $l = \langle x, y, \theta \rangle$  来表示.这里的  $x-y$  是机器人位置的笛卡儿坐标,  $\theta$  表示机器人脸的朝向.  $l_t$  表示机器人在  $t$  时刻的真正位置,  $L_t$  表示相应的变量.通常机器人不知道它的准确位置,而是持有它可能在哪里的信度.我们用  $Bel(L_t)$  表示机器人可能位置的信度分布,其实就是机器人在整个位置空间的概率分布.例如,  $Bel(L_t = l)$  就是机器人在  $t$  时刻在  $l$  处的概率.这个概率分布在下列两个事件发生时进行更新:一是机器人的传感器测量数据到达后;二是读取计程器数据(例如轮子计数器的转数等)后.假定用  $s$  表示环境传感器的测量数据,用  $a$  表示计程器的数据,而  $S$  和  $A$  分别表示相应的变量.设

$$d = \{d_0, d_1, \dots, d_T\} \quad (1)$$

表示测量数据流,这里具有下标  $t(0 \leq t \leq T)$  的  $d_t$  可以是传感器测量数据  $s$  或者是计程器读数  $a$ .所有的数据按变量  $t$  进行索引,而  $T$  表示最近采集的数据项.涉及到这些数据项的集合,  $d$  包含了所有传感器的数据.

马尔可夫自定位算法,以满足所有先前数据为条件来估计在  $L_T$  上概率分布,即

$$P(L_T = l | d) = P(L_T = l | d_0, \dots, d_T) \quad (2)$$

根据马尔可夫假设和贝叶斯规则可以推导出<sup>[4]</sup>:

1) 最近采集的数据项是传感器测量数据  $d_T = s_T$ ,

$$Bel(L_T = l) = \alpha_T P(s_T | l) Bel(L_{T-1} = l) \quad (3)$$

由于  $P(s_T|L_T=l)$  与时间无关所以用替换  $P(s_T|l)$ , 这里  $\alpha_T = 1/P(s_T|d_0, \dots, d_{T-1})$  是个常数, 式(3)具有递增的属性, 这可使算法变得很简单;

2) 最近采集的数据是计程器读数  $d_T = a_T$ ,

$$P(L_T = l) = \int P(l|a_T, l') Bel(L_{T-1} = l') dl' \quad (4)$$

由于  $P(L_T=l|a_T, L_{T-1}=l')$  不随时间变化, 用  $P(l|a_T, l')$  取代它, 式(4)也具有递增的属性.

用方程(3)和(4)来更新机器人位置信度分布就是马尔可夫自定位算法的核心. 完整的算法为 M1, 其程序流程如图 1 所示.

### M1 常规马尔可夫定位算法

- 1) for 每个位置  $l$  do /\* 初始化信度 \*/
- 2)  $Bel(L_0 = l) \leftarrow P(L_0 = l)$
- 3) end for
- 4) forever do
- 5) if 接收到新的传感器数据  $s_T$  do
- 6)  $\alpha_T \leftarrow 0$
- 7) for 每个位置  $l$  do /\* 应用感知模型 \*/
- 8)  $\underline{Bel}(L_T = l) \leftarrow P(s_T|l) \cdot Bel(L_{T-1} = l)$
- 9)  $\alpha_T \leftarrow \alpha_T + \underline{Bel}(L_T = l)$
- 10) end for
- 11) for 每个位置  $l$  do /\* 归一化信度 \*/
- 12)  $Bel(L_T = l) \leftarrow \alpha_T^{-1} \cdot \underline{Bel}(L_T = l)$
- 13) end for
- 14) end if
- 15) if 读出计程器数据  $a_T$  do
- 16) for 每个位置  $l$  do /\* 应用运动模型 \*/
- 17)  $Bel(L_T = l) \leftarrow P(L_T = l) = \int P(l|a_T, l') Bel(L_{T-1} = l') dl'$
- 18) end for
- 19) end if
- 20) end forever

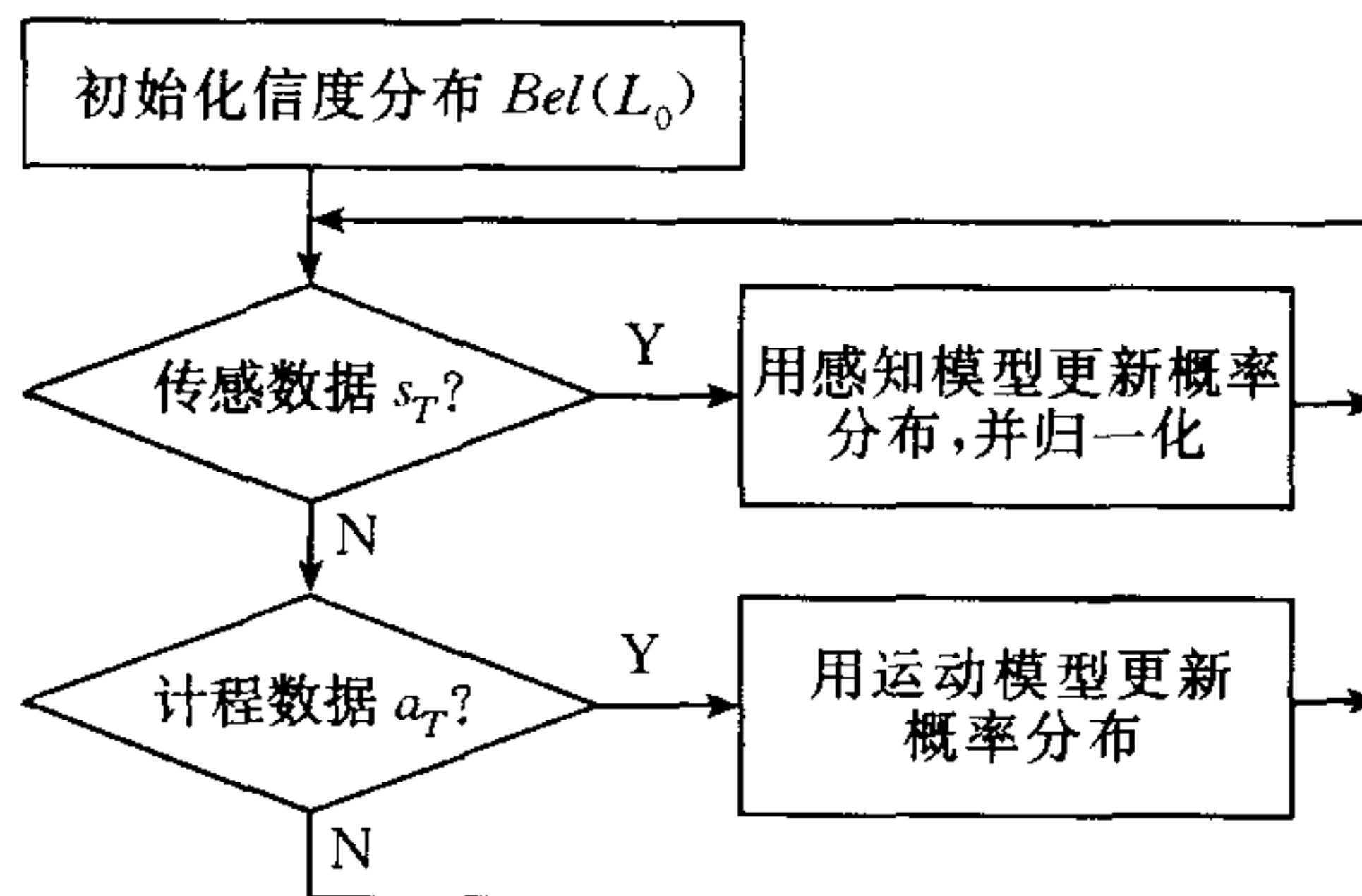


图 1 常规马尔可夫定位算法程序流程图

Fig. 1 Flow chart for regular Markov localization algorithm

这里把  $P(l|a, l')$  称为机器人的运动模型, 因为它描述了运动如何影响机器人位置. 而

条件概率  $P(s|l)$  称为感知模型, 因为它是机器人传感器所获得的结果. 设  $o_l$  表示在地图上从位置  $l$  沿着传感器射线方向到障碍物的距离,  $s$  为传感器在位置  $l$  处测量障碍物的实际距离. 感知模型可用高斯分布

$$P(s|l) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{(s-o_l)^2}{2\delta^2}} \quad (5)$$

来表示. 这里分布模型的标准差  $\delta$  取决于  $L$  离散程度和距离测量不确定程度. 在马尔可夫的算法中  $P(L_0=l)$ , 或信度  $Bel(L_0)$ , 反映了机器人在起动时关于其位置的先验知识. 这个分布可用任意值初始化, 但在实际上常常采用下列两种情况处理: 如果机器人完全不知道它在地图上的位置,  $P(L_0)$  取平均分布; 如果机器人的位置近似知道, 就可采用一个很窄的高斯分布.

早期的马尔可夫定位算法只是一种理论算法, 很少把它应用到实际中. 如果要实现上述算法, 就会发现存在很多问题. 首先让我们来看一个例子, 然后修正上述算法, 使其可以处理拐骗机器人问题.

**例 1.** 设  $\Omega = \{l_1, l_2, l_3, \dots, l_n\}$  表示机器人的位置空间. 假定开始时精确的知道机器人的位置  $L_0 = l_1$ , 根据 M1 算法, 则有

$$Bel(L_0 = l_1) = 1, Bel(L_0 \neq l_1) = 0 \quad (6)$$

现在我们拐骗机器人到  $L_1 = l_i$  ( $l_i \neq l_1$  和  $l_i \in \Omega$ ). 在这过程中, 运动模型对位置的概率分布没有影响. 根据式(3), 可得

$$Bel(L_1 = l_i) = \alpha_T P(s_T | l) Bel(L_{T-1} = l) = \alpha_T P(s_1 | l_i) Bel(L_0 = l_i).$$

从式(6)可知, 对于  $l_i \neq l_1$  和  $l_i \in \Omega$  的地方  $Bel(L_0 = l_i) = 0$ . 代入式(6)得

$$Bel(L_1 = l_i) = 0 \quad (7)$$

这就得出矛盾结论, 换句话说机器人迷路了. 根据式(3), 信度一旦变零后就无法再恢复. 为了解决这个问题, 在感知模型的递增算法计算式(3)中加上先验概率, 把计算式改造成

$$Bel(L_T = l) = \alpha_T P(s_T | l) (Bel(L_{T-1} = l) + P_0) \quad (8)$$

这里的  $P_0 = 1/|\Omega|$  是机器人对它的位置完全不知时, 对于整个位置空间的平均概率. 利用式(8)有替换算法 M1 中的第 8 行, 然后省去初始化程序第 1 到 3 行, 这就成为我们的修正马尔可夫定位算法. 根据修正算法重新计算例 1 情况, 则

$$Bel(L_1 = l_i) = \alpha_T P(s_T | l) (Bel(L_{T-1} = l) + P_0) = \alpha_T P(s_1 | l_i) (Bel(L_0 = l_i) + P_0).$$

把式(6)代入式(8), 得

$$Bel(L_1 = l_i) = \alpha_T P(s_1 | l_i) P_0 \quad (9)$$

读者应该注意到, 这不会象式(7)那样使  $Bel(L_1 = l_i)$  为零而造成矛盾, 这样即使机器人被拐骗也可在运动中利用传感器数据和运动模型来恢复它对位置的估计.

### 3 对称环境中的问题分析

假定图 2 表示一个无特征矩形大厅, 是对称环境. 把一个机器人放在图中  $l_1$  的位置, 并且不让机器人知道自己的初始位置, 即信度  $P(L_0) = 1/|\Omega|$  为均匀分布.

如果只有基于距离测量的传感器, 采用上述的马尔可夫定位法还是无法估计位置. 这个问题出在感知模型  $P(s_T | l)$  和运动模型  $P(l | a_T, l')$  上. 由于环境的对称性, 所以

感知模型有

$$P(s_T | l_1) = P(s_T | l_2) = P(s_T | l_3) = P(s_T | l_4) \quad (10)$$

运动模型有

$$P(l_1 | a_T, l'_1) = P(l_2 | a_T, l'_2) = P(l_3 | a_T, l'_3) = P(l_4 | a_T, l'_4) \quad (11)$$

在图 2 所示的对称环境中, 机器人无论走到那里, 用马尔可夫自定位算法都会得到 4 个具有同等可能的位置. 例如, 机器人在图 2 中  $l_1$  所示的位置上, 获取传感器数据后得到的概率分布在 4 个可能的位置  $l_1, l_2, l_3, l_4$ , 无论机器人如何移动或输入传感器数据总是获得 4 个等概率的位置.

综上所述, 用纯粹的算法无法解决机器人在对称环境中的自定位问题. 我们利用装在机器人上的指南针(或陀螺仪)传感器所读取的数据构造新的感知模型, 即让感知模型  $P(s_T | l)$  结合指南针传感器数据. 设  $b_T$  为指南针传感器上读到的机器人脸朝向角度, 这个角度的起始方向是北极方向.  $\theta_h$  表示机器人脸在地图上的朝向, 用从北极方向到这朝向的角度来衡量. 这样, 概率  $P(b_T | l)$  就可用下式来表示

$$P(b_T | l) = \left( \cos^2 \left( \frac{b_T - \theta_h}{2} \right) \right)^{\frac{1}{\epsilon}} \quad (12)$$

式(12)称为角度高斯分布. 应该注意角度高斯分布不能象式(5)那样的高斯函数来描述. 因为角度变量是一个周期变量, 它的周期是  $2\pi$ . 这里常数  $\epsilon$  取决于指南针的精度和角度的离散程度. 为了简单起见, 假设  $x$  方向为北极. 图 2 中 4 个等概率点的方向感知模型为

$$P(b_T | l_1) = \left( \cos^2 \left( \frac{b_T - \theta_1}{2} \right) \right)^{\frac{1}{\epsilon}},$$

$$P(b_T | l_2) = \left( \cos^2 \left( \frac{b_T - (\theta_1 + \pi/2)}{2} \right) \right)^{\frac{1}{\epsilon}},$$

$$P(b_T | l_3) = \left( \cos^2 \left( \frac{b_T - (\theta_1 + \pi)}{2} \right) \right)^{\frac{1}{\epsilon}},$$

$$P(b_T | l_4) = \left( \cos^2 \left( \frac{b_T - (\theta_1 + 3\pi/2)}{2} \right) \right)^{\frac{1}{\epsilon}} = \left( \cos^2 \left( \frac{b_T - (\theta_1 - \pi/2)}{2} \right) \right)^{\frac{1}{\epsilon}}.$$

在图 2 中  $l_1$  所示机器人, 朝向  $\theta_1 = -\pi/2$ . 如果机器人检测到的方向  $b_T = -\pi/2$ , 可得到概率分布如图 3 所示. 这可明显看出最大的可能位置. 读者应该注意到图 3(a)中,  $l_4$  处机器人朝向的概率密度可以用  $-\pi$  处的概率表示, 也可用  $+\pi$  处的概率表示, 而且在角度高斯分布中它具有同样的值. 这是角度高斯分布函数的重要特征.

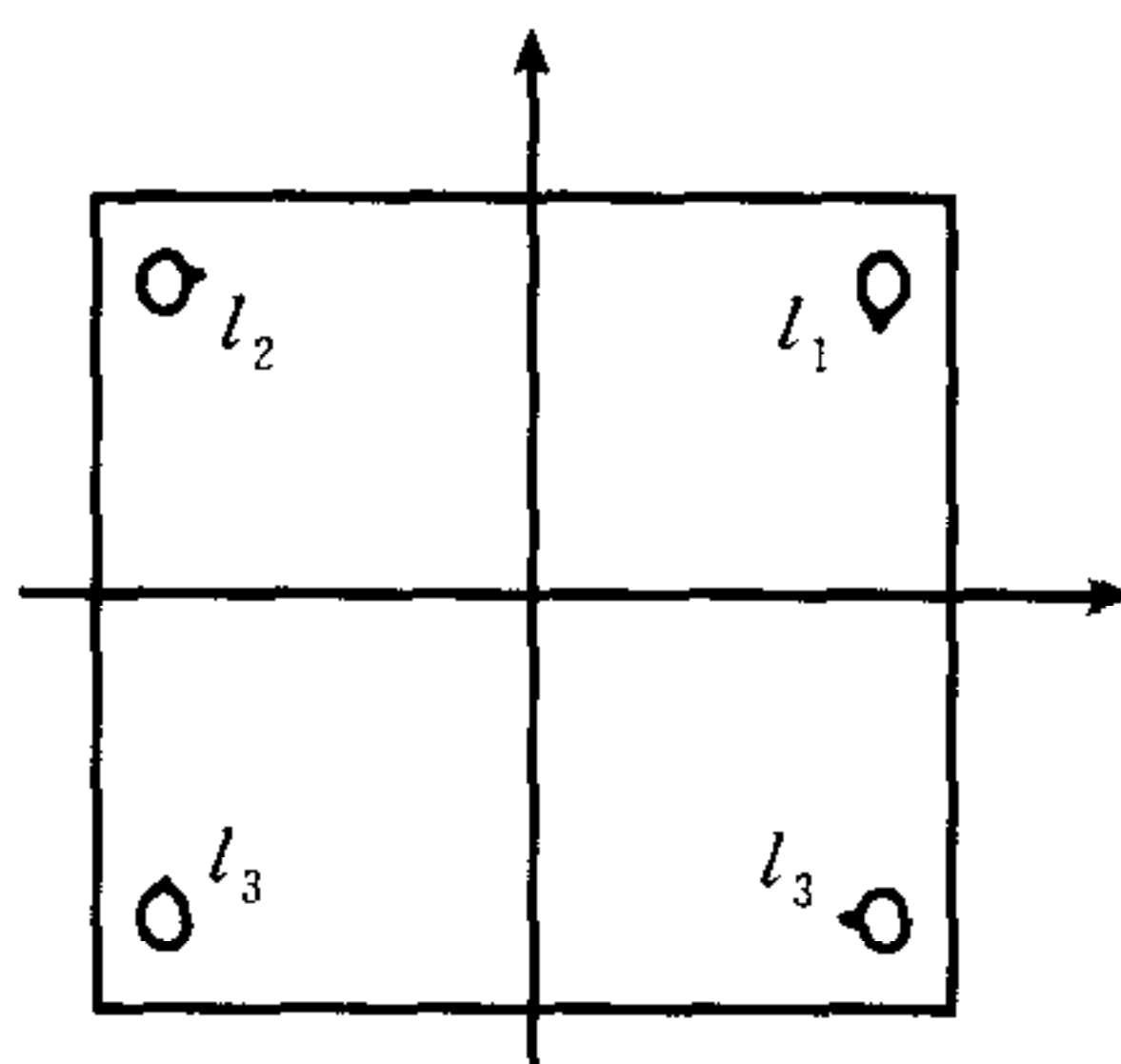


图 2 机器人的位置  
Fig. 2 Robot's position

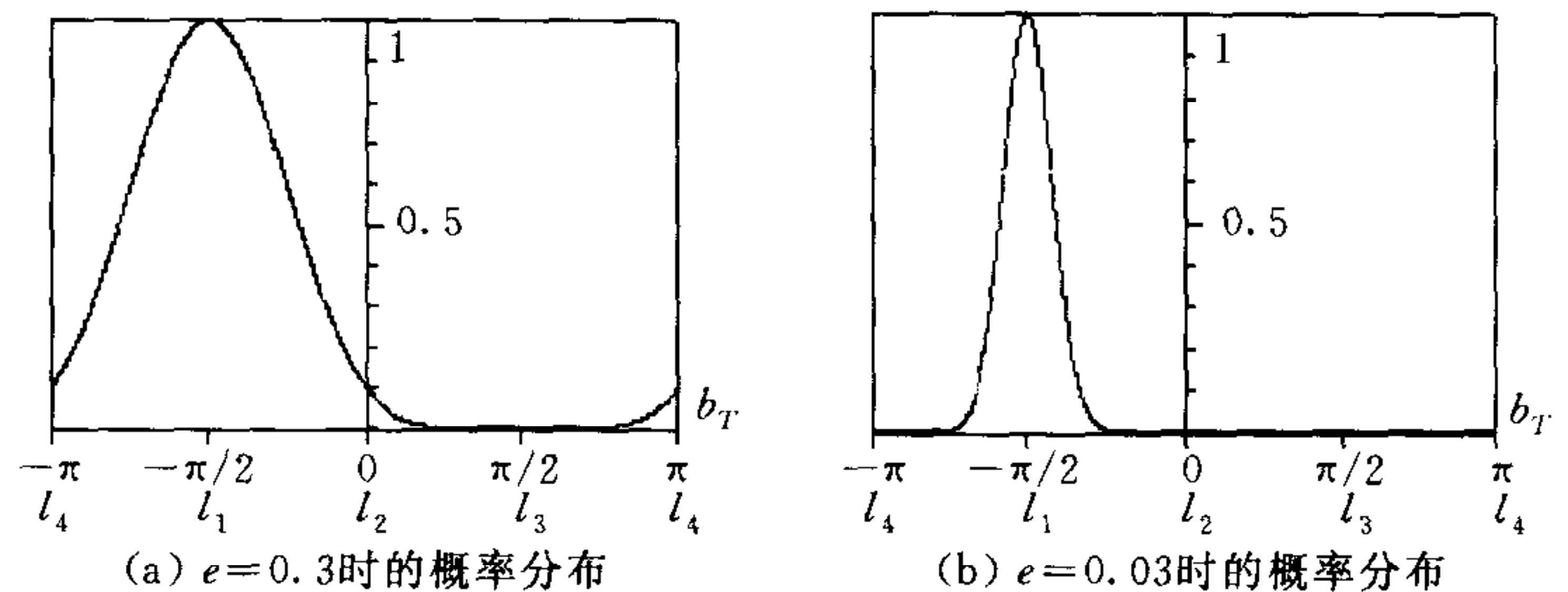


图 3  $b_T = -\pi/2$  处角度高斯分布  
Fig. 3 Gaussian distribution in direction  $b_T = -\pi/2$

很明显  $P(b_T | l)$  即使在对称环境中也是不对称的. 所以, 我们可以定义一个新的感知模型

$$P(w_T | l) = P(s_T | l) \cdot P(b_T | l) \quad (13)$$

$P(w_T | l)$  对于对称环境也是不对称的. 采用这个新的感知模型所获得完整的新算法 M2.

## 4 仿真结果

假设环境是一个无特征的  $15 \times 15\text{m}$  矩形区域.  $x-y$  平面离散成  $10 \times 10\text{cm}$  方格, 取角度的分辨率为  $5^\circ$ . 我们编写了一个仿真程序来仿真一个带有指南针和 4 个距离传感器的机器人在这个环境中的运动. 4 个传感器分别位于机器人的前方、左边、右边和后面. 因为读传感器数据不影响机器人的状态, 可把 4 个方向的传感器数据  $P(s_T^i | l)$  综合成下式

$$P(s_T | l) = \prod_i P(s_T^i | l) \quad (14)$$

这里的  $i=1, \dots, 4$ . 再把指南针所测量的结合进去, 新感知模型的计算式为

$$P(w_T | l) = P(s_T | l), P(b_T | l) = \prod_i P(s_T^i | l) \left( \cos^2 \left( \frac{\{b_T - \theta_i\}}{2} \right) \right)^{\frac{1}{i}} \quad (15)$$

在仿真程序中, 采用了 2 个运动模型, 左转或右转  $5^\circ$  和向前运动  $10\text{cm}$ , 在运动  $1\text{m}$  后, 显示一个最大概率点. 结果显示算法运行很好. 我们对多种对称性的环境进行了测试, 图 4 和图 5 显示的只是其中的两个情况.

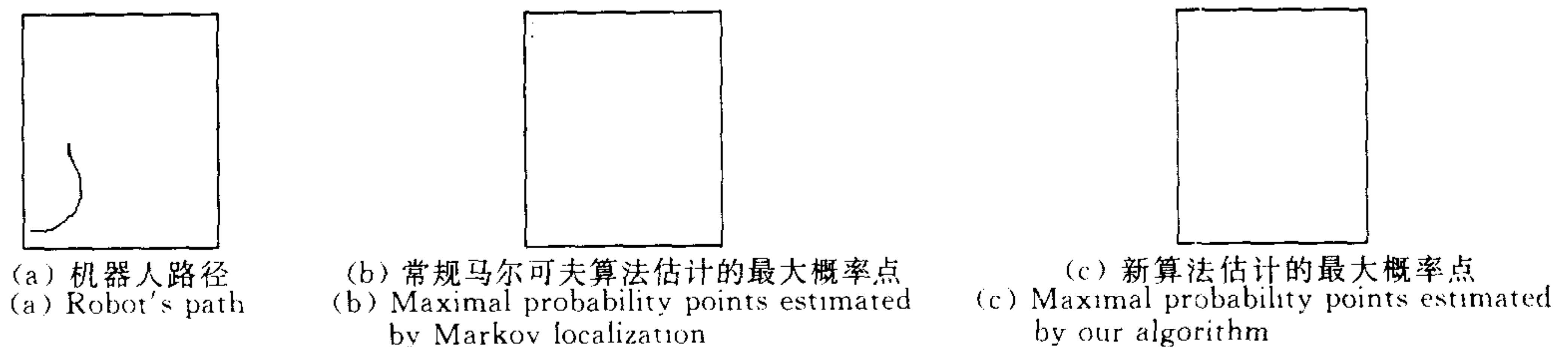


图 4 无特征矩形环境的仿真结果

Fig. 4 Result of the simulation for featureless square

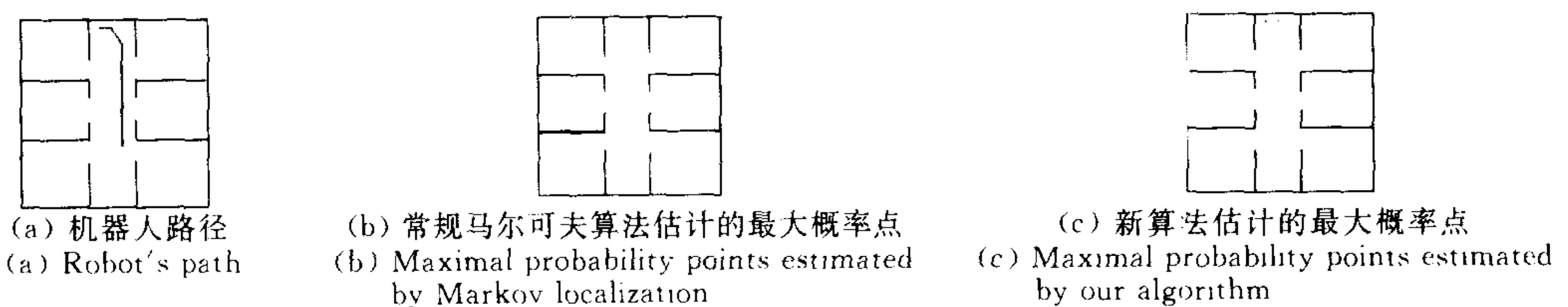


图 5 楼道环境的仿真结果

Fig. 5 Result of the simulation for a corridor

### M2 采用新感知模型的马尔可夫算法

forever do

if 接收到新的传感器数据  $w_T$  do

$\alpha_T \leftarrow 0$

for 每个位置  $l$  do /\* 应用感知模型 \*/

$\underline{Bel}(L_T=l) \leftarrow P(w_T | l) \cdot (\underline{Bel}(L_{T-1}=l) + P_0)$  /\* 这里应用了先验概率  $P_0$  和新的感知模型 \*/

$\alpha_T \leftarrow \alpha_T + \underline{Bel}(L_T=l)$

end for

for 每个位置  $l$  do /\* 归一化信度 \*/

```

 $Bel(L_T=l) \leftarrow \alpha_T^{-1} \cdot \underline{Bel}(L_T=l)$ 
end for
end if
if 读出计程器数据  $a_T$  do
for 每个位置  $l$  do /* 应用运动模型 */
 $Bel(L_T=l) \leftarrow P(L_T=l) = \int P(l|a_T, l') Bel(L_{T-1}=l') dl'$ 
end for
end if
end forever

```

## 5 结论

综上所述,采用先验概率修改了常规马尔可夫算法,解决了拐骗机器人的问题;定义了角度高斯分布函数,并用它构造新的感知模型,使新算法可用于对称环境中估计机器人的位置.通过仿真程序证实了它的可行性.如果利用同样的方法将机器人识别图像的算法<sup>[6]</sup>结合到感知模型中,可以使机器人在自定位时更加智能,这是我们下一步的工作.本文的方法显示,技术在机器人的研究领域是非常重要的,只要在机器人上装上一个方向仪(如指南针或陀螺仪)就可以解决用纯粹数学算法无法解决的问题.

## References

- 1 Borenstein J, Everett B, Feng L. Navigating Mobile Robots: Systems and Techniques. Natick: A K Peters Press, 1996. 67~96
- 2 Cox I J, Wilfong GT. Autonomous Robot Vehicles. New York: Springer-Verlag, 1990. 25~31
- 3 Feng L, Borenstein J, Everett H R. "Where am I?" sensors and methods for autonomous mobile robot positioning. In: Technical Report UM-MEAM-94-12, USA: University of Michigan, 1994. 1~55
- 4 Thrun S, Buecken A, Burgard W *et al.* Map learning and high-speed navigation in RHINO. In: AI-based Mobile Robots: Case Studies of Successful Robot Systems, Kortenkamp D, Bonasso R P, Murphy R (eds.), USA: MIT Press, 1998. 21~49
- 5 Dieter Fox, Wolfram Burgard, Sebastian Thrun. Markov Localization For Mobile Robots In Dynamic Environments. *Journal of Artificial Intelligence Research*, 1999, 11:391~427
- 6 Wu Qing-Xiang, Bell David. Related value set algorithm for robot to distinguish image. In: Proceedings of the 3rd World Congress on Intelligent Control and Automation, Heifei: IEEE Press, 2000, 2:1546~1550

**吴庆祥** 福建师范大学网络中心副教授. 主要研究领域是智能机器人、知识发现与网络信息系统、疾病诊断、计算机辅助教学和汉字信息处理等.

(**WU Qing-Xiang** Associate professor in Network Information Center, Fujian Teachers University. His research interests include intelligent system and autonomous robot, knowledge discovery and information system based on Web, automatic diagnosis for diseases, computer aid instruction, and Chinese information processing. )

**Bell David** 英国 Ulster 大学教授,人工智能研究室主任. 主要研究领域是人工智能理论、不确定性智能推理、智能机器人、知识发现与数据库理论等.

(**Bell David** Professor and head of the Artificial Intelligence Division in School of Information and Software Engineering, University of Ulster, UK. His research interests include artificial intelligence, uncertainty reasoning, knowledge discovery and data base theory. )