

# 基于中心差分粒子滤波的 SLAM 算法

祝继华<sup>1</sup> 郑南宁<sup>1</sup> 袁泽剑<sup>1</sup> 张强<sup>1</sup>

**摘要** 针对移动机器人同时定位与地图创建 (Simultaneous localization and mapping, SLAM) 中的 FastSLAM 算法, 存在非线性系统线性化处理和计算雅可比矩阵的缺点, 本文提出了基于 Sterling 多项式插值处理非线性系统的 SLAM 方法. 该方法基于 Rao-Blackwellized 粒子滤波框架, 利用中心差分滤波方法产生改进的建议分布函数, 提高了机器人位姿估计的精度; 利用中心差分滤波初始化特征和更新地图中的特征, 提高了地图创建的精度; 针对实际应用中存在虚假特征的情况提出了一种有效的地图管理方法. 在同等粒子数的情况下, 该方法改进了 SLAM 结果的精度. 基于仿真和实际数据的实验结果验证了该方法的有效性.

**关键词** 同时定位与地图创建, Rao-Blackwellized 粒子滤波, 中心差分滤波器, 建议分布函数

**DOI** 10.3724/SP.J.1004.2010.00249

## A SLAM Algorithm Based on Central Difference Particle Filter

ZHU Ji-Hua<sup>1</sup> ZHENG Nan-Ning<sup>1</sup> YUAN Ze-Jian<sup>1</sup> ZHANG Qiang<sup>1</sup>

**Abstract** There are two serious drawbacks in FastSLAM (Simultaneous localization and mapping), which are the derivation of the Jacobian matrices and the linear approximations of nonlinear functions. To overcome the serious drawbacks of the previous frameworks, this paper provides a robust SLAM algorithm based on the Sterling polynomial interpolation. It uses the central difference filter (CDF) to compute the proposal distribution in Rao-Blackwellized particle filter, then to initialize and update each feature state. For practical application, an effective mechanism for feature management is proposed. This approach improves the state estimation accuracy, and requires a smaller number of particles than previous approaches. Both simulation and experimental results are used to validate the effectiveness of the proposed algorithm.

**Key words** Simultaneous localization and mapping, Rao-Blackwellized particle filter, central difference filter (CDF), proposal distribution

移动机器人在未知环境中依靠自身携带的传感器获取环境的感知信息, 递归地创建周围环境的地图, 同时利用所创建的地图实现自身定位, 即同时定位和地图创建问题是移动机器人实现真正自主的关键. 该研究始于上世纪 80 年代中期<sup>[1]</sup>, 目前已有多种解决此问题的方法<sup>[2]</sup>, 主要包括非概率方法和基于概率估计方法, 并以基于概率估计方法为主流<sup>[3-4]</sup>.

最早的概率估计方法是基于扩展卡尔曼滤波的 SLAM 算法<sup>[5-6]</sup>, 但此类算法存在难以解决的数据关联问题和正比于特征个数二次方的计算量与存储量问题. 近年来, Doucet 等提出了利用 Rao-Blackwellized 粒子滤波解决 SLAM (Simultaneous localization and mapping) 问题<sup>[7]</sup> 的有效方法. 它

可以根据需要生成特征地图、栅格地图, 并且很好地解决了数据关联问题. 但由于此类方法中, 每个粒子中存储着各自的机器人定位结果和地图, 故面临着正比于粒子数的计算量和存储量问题. 因此, 在不影响 SLAM 结果精度的前提下, 如何降低 SLAM 过程中所需的最少粒子数是本文研究的重点. 基于 Rao-Blackwellized 粒子滤波框架, Motemerlo 等提出了适用于创建特征地图的有效方法: FastSLAM 算法<sup>[8-9]</sup>. 该方法利用粒子滤波估计机器人的状态信息, 并利用多个扩展卡尔曼滤波器分别估计每个特征的状态信息. 上述方法由于均采用泰勒展开近似非线性系统并需要计算雅可比矩阵, 因此估计结果的误差较大并且计算繁琐. 为此有学者提出了利用无味卡尔曼滤波<sup>[10]</sup> 替代 FastSLAM 算法中的扩展卡尔曼滤波的方法<sup>[11-12]</sup>, 该方法进一步降低了地图创建所需的粒子数, 但需要根据经验确定尺度无味变换 (Scaled unscented transformation, SUT) 中所需的 3 个未知参数, 参数的选择将影响定位和地图创建的精度.

本文提出了基于 Sterling 多项式插值<sup>[13-14]</sup> 处

收稿日期 2008-11-12 录用日期 2009-04-03

Manuscript received November 12, 2008; accepted April 3, 2009  
国家重点基础研究发展计划 (973 计划) (2007CB311005) 和国家自然科学基金 (90820017) 资助

Supported by National Basic Research Program of China (973 Program) (2007CB311005) and National Natural Science Foundation of China (90820017)

1. 西安交通大学人工智能与机器人研究所 西安 710049  
1. Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049

理非线性系统的 SLAM 方法. 该方法在利用中心差分滤波方法产生改进的建议分布函数基础之上, 利用中心差分滤波方法初始化特征和更新地图中的特征. 此外, 为了剔除实际应用中的虚假特征, 本文提出了一种有效的地图管理策略. 在同等粒子数下, 这些改进的措施提高了移动机器人同时定位和地图创建的精度.

## 1 FastSLAM 算法

假设  $\Theta$  表示地图状态变量 (静止),  $\theta_k$  表示第  $k$  个特征. 机器人在  $t$  时刻的状态变量为  $\mathbf{x}_t$ , 观测结果为  $\mathbf{z}_t$ ,  $(t-1)$  时刻到  $t$  时刻的控制输入为  $\mathbf{u}_t$ . 在概率论中, SLAM 可描述为: 给定 1 时刻到  $t$  时刻里程计读数  $\mathbf{u}^t = \{\mathbf{u}_1, \dots, \mathbf{u}_{t-1}\}$  和 1 时刻到  $t$  时刻传感器观测  $\mathbf{z}^t = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$ , 递归地利用贝叶斯滤波器求取移动机器人的路径  $\mathbf{x}^t = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$  和环境地图的联合后验概率密度函数  $p(\mathbf{x}^t, \Theta | \mathbf{z}^t, \mathbf{u}^t)$ . 在 SLAM 问题中, 给定机器人的姿态估计的前提下, 不同特征之间的估计是相互独立的<sup>[8-9]</sup>. 根据 SLAM 这一特点, 可以将联合后验概率密度  $p(\mathbf{x}^t, \Theta | \mathbf{z}^t, \mathbf{u}^t)$  分解成以下形式:

$$p(\mathbf{x}^t, \Theta | Y) = p(\mathbf{x}^t | Y) \prod_{k=1}^N p(\theta_k | Y) \quad (1)$$

其中  $Y$  代表  $(n^t, \mathbf{z}^t, \mathbf{u}^t)$ ,  $n^t = \{n_1, \dots, n_t\}$  为数据关联变量. 式 (1) 是利用粒子滤波器实现 SLAM 的理论基础和依据.

文献 [8] 提出了基于式 (1) 的 FastSLAM 算法. 该算法基于 Rao-Blackwellized 粒子滤波框架, 利用粒子滤波器估计机器人的状态, 并用多个扩展卡尔曼滤波器分别估计每个特征的状态. 此算法中的每个粒子分别包含了该粒子的权值, 机器人的路径信息和特征的状态:

$$\mathbf{X}_t^{[m]} = \left\langle w_t^{[m]}, \mathbf{x}_t^{[m]}, \boldsymbol{\mu}_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, \boldsymbol{\mu}_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \right\rangle \quad (2)$$

其中,  $\mathbf{x}_t^{[m]}$  代表第  $m$  个粒子关于机器人的路径估计结果,  $\boldsymbol{\mu}_{k,t}^{[m]}$  是第  $k$  个特征的状态估计结果,  $\Sigma_{k,t}^{[m]}$  为第  $k$  个特征的状态估计协方差矩阵. FastSLAM 算法包括 1.0 和 2.0 两个版本<sup>[8-9]</sup>. 前者的建议分布函数只融入了当前里程计的读数, 未考虑最新的观测信息. 由于传感器的精度一般远高于里程计的精度, 这将导致粒子滤波的抽样效率十分低<sup>[15]</sup>. 基于最新的观测信息, 后者获得了改进的建议分布函

数, 因此降低了地图创建所需的粒子数. 由此可知, FastSLAM 算法的关键在于如何计算粒子的权值  $w_t^{[m]}$ : 由于目标函数  $p(\mathbf{x}^t, \Theta | n^t, \mathbf{z}^t, \mathbf{u}^t)$  确定, 因此好的建议分布函数将改进该算法的估计结果.

在上述算法中, 每个粒子都存储着各自关于机器人运动轨迹的信息, 以及多个独立特征状态的估计. 通过引入二叉树的存储结构<sup>[9]</sup>, 可降低算法所需的存储空间. 此类算法很好地解决了 SLAM 问题中的数据关联问题. 然而该算法存在着线性化处理非线性系统的缺陷, 这种处理方式严重地影响了机器人状态以及特征状态估计结果的精确性, 并因此增加了完成 SLAM 过程所需的粒子数. 为了解决此类问题, 本文提出了基于中心差分粒子滤波的 SLAM 算法. 由于文中变量较多, 为方便起见, 大部分变量命名规则直接引自文献 [8, 12-13].

## 2 基于中心差分粒子滤波的 SLAM 算法

文献 [14] 提出了一种类似无味卡尔曼滤波处理非线性系统的方法, 该方法利用 Sterling 多项式插值原理, 结合一系列线性回归计算, 并基于卡尔曼滤波框架, 形成了性能略优于无味卡尔曼滤波器的中心差分滤波器. 随后文献 [13] 结合中心差分滤波和粒子滤波, 形成了性能优良的中心差分粒子滤波器. 基于 SLAM 问题的可分解性<sup>[8-9]</sup>, 结合 Rao-Blackwellized 粒子滤波框架, 可获得基于中心差分粒子滤波的 SLAM 算法 (Central difference Fast-SLAM, CDFastSLAM).

### 2.1 改进的建议分布函数

在 CDFastSLAM 算法中, 假设系统的过程噪声和观测噪声分别为  $Q_t$  和  $R_t$ , 第  $m$  个粒子  $(t-1)$  时刻的移动机器人状态估计为  $(\mathbf{x}_{t-1}^{[m]}, P_{t-1}^{[m]})$ . 则在  $t$  时刻系统的状态变量及对应协方差可扩展为:

$$\mathbf{x}_{t-1}^{a[m]} = \begin{bmatrix} \mathbf{x}_{t-1}^{[m]} \\ 0 \end{bmatrix}, \quad P_{t-1}^{a[m]} = \begin{bmatrix} P_{t-1}^{[m]} & 0 \\ 0 & Q_t \end{bmatrix}$$

其中状态变量  $\mathbf{x}_{t-1}^{a[m]}$  维数为  $L$ . 通过式 (3) 可获得用于后续线性回归计算的  $(2L+1)$  个点集:

$$\boldsymbol{\chi}_{t-1}^{a[m]} = [ \mathbf{x}_{t-1}^{a[m]} \quad \mathbf{x}_{t-1}^{a[m]} \pm h \sqrt{P_{t-1}^{a[m]}} ] \quad (3)$$

点集中的每个点在计算待估计变量的均值和方差时存在对应的权值:

$$\begin{aligned} w_0^{(c)} &= \frac{h^2 - L}{h}, & w_i^{(c)} &= \frac{1}{2h}, & i &= 1, \dots, 2L \\ w_i^{(c_1)} &= \frac{1}{4h^2}, & w_i^{(c_2)} &= \frac{h^2 - 1}{4h^2}, & i &= 1, \dots, L \end{aligned} \quad (4)$$

其中  $h$  为中心插值的步长. 虽然对于同等维数的状态变量, 文献 [12] 所提出的算法与本文所提出的算法生成点集所含点数相同, 但由于赖以产生点集的原理不同, 针对相同状态变量和方差, 两者所产生点的数值和对应权重均不相同, 计算估计方差和协方差的方式也截然不同. 对于任意维数的高斯随机变量,  $h$  的最优值为  $\sqrt{3}$ <sup>[13]</sup>. 由式 (3) 获得的随机变量包含两种分量: 机器人状态分量和输入控制分量, 即:

$$\mathbf{x}_{t-1}^{a[i][m]} = \left[ \mathbf{x}_{t-1}^{[i][m]}, \quad \mathbf{x}_t^{u[i][m]} \right]^T \quad (5)$$

在利用传感器观测更新机器人状态之前, 需要根据里程计读数对机器人状态进行一步预测, 即利用机器人状态转移方程  $f(\cdot)$  对所有的点按式 (6) 进行变换:

$$\bar{\mathbf{x}}_t^{[i][m]} = f(\mathbf{x}_{t-1}^{[i][m]}, \quad \mathbf{u}_t^{[m]} + \mathbf{x}_t^{u[i][m]}) \quad (6)$$

变换后的点集经过以下线性回归计算后, 可获得机器人状态的一步预测结果和方差:

$$\mathbf{x}_{t|t-1}^{[m]} = \sum_{i=0}^{2L} w_i^{(c)} \bar{\mathbf{x}}_t^{[i][m]} \quad (7)$$

$$\begin{aligned} P_{t|t-1}^{[m]} &= \sum_{i=1}^L [w_i^{(c_1)} (\bar{\mathbf{x}}_t^{[i][m]} - \bar{\mathbf{x}}_t^{[i+L][m]})^2 + \\ &w_i^{(c_2)} (\bar{\mathbf{x}}_t^{[i][m]} + \bar{\mathbf{x}}_t^{[i+L][m]} - 2\bar{\mathbf{x}}_t^{[0][m]})^2] \end{aligned} \quad (8)$$

在特征稀疏的场景中可能没有观测, 这时只需利用式 (7) 和 (8) 对机器人位姿进行预测. 当传感器获得地图中已存在特征的观测信息时, 可由相应的数据关联算法<sup>[6]</sup> 获得关联结果. 利用此观测信息对机器人的状态进行更新, 先对状态变量的预测进行扩展并计算点集:

$$\begin{aligned} \mathbf{x}_{t|t-1}^{n[m]} &= \begin{bmatrix} \mathbf{x}_{t|t-1}^{[m]} \\ 0 \end{bmatrix}, & P_{t|t-1}^{n[m]} &= \begin{bmatrix} P_{t|t-1}^{[m]} & 0 \\ 0 & R_t \end{bmatrix} \\ \mathbf{x}_{t|t-1}^{n[m]} &= \left[ \mathbf{x}_{t|t-1}^{n[m]}, \quad \mathbf{x}_{t|t-1}^{n[m]} \pm h\sqrt{P_{t|t-1}^{n[m]}} \right] \end{aligned} \quad (9)$$

由式 (9) 获得的随机变量包含两种分量: 机器人状态分量和观测分量.

$$\mathbf{x}_{t|t-1}^{n[i][m]} = \left[ \mathbf{x}_{t|t-1}^{[i][m]}, \quad \mathbf{x}_t^{z[i][m]} \right]^T \quad (10)$$

与 FastSLAM 2.0 不同, 这里需要借助线性回归计算获得特征观测的预测  $\hat{\mathbf{n}}_t^{[m]}$ .

$$\bar{\mathbf{N}}_t^{[i][m]} = h(\mathbf{x}_{t|t-1}^{[i][m]}, \quad \boldsymbol{\mu}_{\hat{k}, t-1}^{[m]}) + \mathbf{x}_t^{z[i][m]} \quad (11)$$

$$\hat{\mathbf{n}}_t^{[m]} = \sum_{i=0}^{2L} w_i^{(c)} \bar{\mathbf{N}}_t^{[i][m]} \quad (12)$$

其中  $h(\cdot)$  为非线性的观测方程,  $\boldsymbol{\mu}_{\hat{k}, t-1}^{[m]}$  为第  $\hat{k}$  个特征前一时刻的状态估计. 并利用线性回归计算获得用于计算卡尔曼增益的新息协方差  $S_t^{[m]}$  和互协方差  $\Sigma_t^{[m]}$ .

$$\begin{aligned} S_t^{[m]} &= \sum_{i=1}^L [w_i^{(c_1)} (\bar{\mathbf{N}}_t^{[i][m]} + \bar{\mathbf{N}}_t^{[i+L][m]})^2 + \\ &w_i^{(c_2)} (\bar{\mathbf{N}}_t^{[i][m]} + \bar{\mathbf{N}}_t^{[i+L][m]} - 2\bar{\mathbf{N}}_t^{[0][m]})^2] \end{aligned} \quad (13)$$

$$\Sigma_t^{[m]} = P_{(1:L', :)} [\bar{\mathbf{N}}_t^{[1:L][m]} - \bar{\mathbf{N}}_t^{[1+L:2L][m]}]^T \quad (14)$$

其中,  $P = \sqrt{w_1^{(c_1)} P_{t|t-1}^{n[m]}}$ ,  $P_{(1:L', :)}$  由矩阵  $P$  的第 1 行到第  $L'$  (机器人状态维数) 行元素组成. 卡尔曼增益  $K_t^{[m]}$  可由下式计算获得:

$$K_t^{[m]} = \Sigma_t^{[m]} (S_t^{[m]})^{-1} \quad (15)$$

最后可通过标准的卡尔曼滤波更新公式对机器人的状态及协方差进行更新:

$$\bar{\mathbf{x}}_t^{[m]} = \mathbf{x}_{t-1}^{[m]} + K_t^{[m]} (\mathbf{z}_t - \hat{\mathbf{n}}_t^{[m]}) \quad (16)$$

$$P_t^{[m]} = P_{t-1}^{[m]} - K_t^{[m]} S_t^{[m]} (K_t^{[m]})^T \quad (17)$$

上述方法只适合于传感器一次有且只有一个观测的情景. 当某时刻获得多个特征的观测时, 可重复利用式 (9)~(17) 对机器人的状态进行更新. 但在每次重复上述更新前, 需利用式 (16) 和 (17) 中的机器人最新状态估计结果替代式 (7) 和 (8) 中机器人的一步预测结果. 通过上述计算可获得改进的建议分布函数, 最后可从建议分布函数中抽取新的粒子:

$$\mathbf{x}_t^{[m]} \sim N(\bar{\mathbf{x}}_t^{[m]}, P_t^{[m]}) \quad (18)$$

在计算建议分布函数时, 本文利用 Sterling 多项式插值处理非线性系统, 并借助于一系列线性回归计算, 解决了 FastSLAM 2.0 中利用泰勒展开近似非线性系统并需要计算雅可比矩阵的不足, 获得了改进的建议分布函数.

## 2.2 特征状态估计

特征的状态估计包括: 特征初始化和特征状态更新两部分. 当观测到一个新特征时, 需要将此特征在地图中进行初始化; 当观测到一个已经存在于地图中的特征时, 需要利用观测信息对已有特征的状态进行更新. FastSLAM 算法利用扩展卡尔曼滤波估计特征状态, 由于扩展卡尔曼滤波的一些缺陷, 导致所获得的特征状态的估计不够精确.

当机器人携带的传感器获得观测时, 数据关联算法给出对应的匹配结果. 如果某特征还未存在于地图中, 则可用下述方法对特征进行初始化. 通过下式获得用于后续线性回归计算的  $(2l+1)$  个点 ( $l$  为特征状态的维数):

$$\boldsymbol{\varphi}^{[m]} = [z_t \quad z_t \pm h\sqrt{R_t}] \quad (19)$$

利用观测函数的反函数  $h^{-1}(\cdot)$  对上述点集中所有的点进行非线性变换:

$$\bar{\mathbf{M}}_t^{[i][m]} = h^{-1}(\mathbf{x}_t^{[m]}, \boldsymbol{\varphi}^{[i][m]}) \quad (20)$$

利用线性回归计算获得特征初始化估计和方差:

$$\boldsymbol{\mu}_{n_t, t}^m = \sum_{i=0}^{2l} w_i^{(c)} \bar{\mathbf{M}}_t^{[i][m]} \quad (21)$$

$$\begin{aligned} \Sigma_{n_t, t}^{[m]} = & \sum_{i=1}^l [w_i^{(c_1)} (\boldsymbol{\varphi}^{[i][m]} - \boldsymbol{\varphi}^{[i+l][m]})^2 + \\ & w_i^{(c_2)} (\boldsymbol{\varphi}^{[i][m]} + \boldsymbol{\varphi}^{[i+l][m]} - 2\boldsymbol{\varphi}^{[0][m]})^2] \quad (22) \end{aligned}$$

在分析特征状态过程中多次用到带权重的线性运算, 所需的权值可通过用  $l$  替代  $L$  后, 按式 (4) 所示方法计算获得.

当传感器获得地图中已有特征的观测时, 可利用已知特征的状态估计结果获取用于后续计算的点集:

$$\boldsymbol{\chi}^{[m]} = [ \boldsymbol{\mu}_{n_t, t-1}^{[m]} \quad \boldsymbol{\mu}_{n_t, t-1}^{[m]} \pm h\sqrt{\Sigma_{n_t, t-1}^{[m]}} ] \quad (23)$$

该点集包含  $(2l+1)$  个点, 利用观测方程对这些点进行变换:

$$\bar{\mathbf{Z}}_t^{[i][m]} = h(\mathbf{x}_t^{[m]}, \boldsymbol{\chi}^{[i][m]}) \quad (24)$$

获得的点集可用于计算特征观测的预测  $\hat{\mathbf{z}}_t^{[m]}$ :

$$\hat{\mathbf{z}}_t^{[m]} = \sum_{i=0}^{2l} w_i^{(c)} \bar{\mathbf{Z}}_t^{[i][m]} \quad (25)$$

为了对特征状态进行更新还需获得卡尔曼增益  $K_t^{[m]}$ , 以及用于计算卡尔曼增益的新息协方差矩阵  $\bar{S}_t^{[m]}$  和互协方差矩阵  $\bar{\Sigma}_t^{[m]}$ . 借助下面的线性回归计算可获得新息协方差矩阵和互协方差矩阵:

$$\begin{aligned} \bar{S}_t^{[m]} = & \sum_{i=1}^l \left[ w_i^{(c_1)} (\bar{\mathbf{Z}}_t^{[i][m]} - \bar{\mathbf{Z}}_t^{[i+l][m]})^2 + \right. \\ & \left. w_i^{(c_2)} (\bar{\mathbf{Z}}_t^{[i][m]} + \bar{\mathbf{Z}}_t^{[i+l][m]} - 2\bar{\mathbf{Z}}_t^{[0][m]})^2 \right] + R_t \quad (26) \end{aligned}$$

$$\bar{\Sigma}_t^{[m]} = \sqrt{w_1^{(c_1)} \Sigma_{n_t, t-1}^{[m]}} \left[ \bar{\mathbf{Z}}_t^{[1:l][m]} - \bar{\mathbf{Z}}_t^{[l+1:2l][m]} \right]^T \quad (27)$$

同时可按下式计算卡尔曼增益  $\bar{K}_t^{[m]}$ :

$$\bar{K}_t^{[m]} = \bar{\Sigma}_t^{[m]} (\bar{S}_t^{[m]})^{-1} \quad (28)$$

最后可以利用标准的卡尔曼滤波的更新公式对特征的状态和方差进行更新:

$$\boldsymbol{\mu}_{n_t, t}^{[m]} = \boldsymbol{\mu}_{n_t, t-1}^{[m]} + \bar{K}_t^{[m]} (z_t - \hat{\mathbf{z}}_t^{[m]}) \quad (29)$$

$$\Sigma_{n_t, t}^{[m]} = \Sigma_{n_t, t-1}^{[m]} - \bar{K}_t^{[m]} \bar{S}_t^{[m]} (\bar{K}_t^{[m]})^T \quad (30)$$

在上述分析特征状态的过程中, 本文借助于 Sterling 多项式插值处理非线性系统, 克服了扩展卡尔曼滤波中利用泰勒展开近似非线性系统并需要计算雅可比矩阵的不足. 在计算建议分布函数和分析特征的状态过程中, 均需要对协方差矩阵作开方运算. 为了保证数值的稳定性, 程序实现时可直接采用文献 [13] 所提出的平方根滤波器方法.

## 2.3 权值计算及重抽样

权值的计算公式如下:

$$w_t^{[m]} = w_{t-1}^{[m]} \frac{p(\mathbf{x}_t^{[m]} | \mathbf{x}_{t-1}^{[m]}, \mathbf{u}_t) p(z_t | \Theta^{[m]}, \mathbf{x}_t^{[m]})}{\pi(\mathbf{x}_t | \mathbf{x}^{[t]}, \mathbf{z}^t, \mathbf{u}^t, n^t)} \quad (31)$$

上式中分母部分代表 FastSLAM 2.0 中所提出的改进建议分布函数, 文献 [16] 中给出了上式的具体实现方式. 权值的计算有赖于数据关联结果的精确性. 目前最简单有效的数据关联算法是最大似然分析 (Maximum likelihood, ML) 的方法.

重抽样是提高粒子滤波性能的有效方法. 以有限的粒子数近似后验概率密度时, 重抽样过程是必要的, 但抽样的过程中会丢失一些好的粒子. 文献 [15] 首次将自适应重抽样方法引入 SLAM 算法中, 该方法对粒子进行归一化处理, 通过计算有效粒子数  $N_{\text{eff}}^{[17]}$  确定进行重抽样的时间.

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^M (\tilde{w}^{[m]})^2} \quad (32)$$

当  $N_{\text{eff}}$  小于某个域值 (如  $M/2$ ,  $M$  为粒子总数) 时, 可进行重抽样, 以减弱粒子退化现象. 目前各类文献中介绍了多种重抽样的方法, 不同方法的抽样效果较为接近<sup>[17]</sup>, 考虑到程序运行时间, 建议采用残差重抽样方法.

## 2.4 地图管理策略

在实际的应用中, 环境并非绝对静止或特征提取算法不够稳定, 会出现虚假特征的情况. 因此, 文献 [9] 提出了利用“负信息”的方法对地图中的特征进行管理. 但该方法将虚假特征的信息融入了建议分布函数中, 恶化了建议分布函数, 此外未能充分考虑物体遮挡造成观测不当以及特征提取算法不稳定等因素. 根据基于扩展卡尔曼滤波 SLAM 算法的地图管理策略<sup>[5]</sup>, 本文提出了一种有效的地图管理策略: 在每个粒子中, 设置一个候选特征列表, 当在  $t$  时刻第一次观测到某特征时, 可将该特征存储在候选列表中, 并记录时间信息. 在作数据关联时, 先将观测与地图中的特征进行关联, 后将未关联上的观测再与候选列表中的特征进行关联. 在后续的一段时刻  $T$  内, 如该特征被观测到的次数超过  $K$  次, 则可认为此特征为真实特征, 将此特征移入地图中; 否则可认为该特征为虚假特征并删除之. 此方法使得虚假特征在初始阶段得到了确认, 且未将虚假特征用于计算建议分布函数. 因此减少了计算量, 提高移动机器人的位姿估计精度.

## 2.5 算法流程及复杂度分析

CDFastSLAM 算法的基本流程如下:

1) 抽样: 基于前一时刻估计结果  $\{\mathbf{X}_{t-1}^{[m]}\}$  和当前的观测结果, 利用第 2.1 节中的方法, 获得改进的建议分布函数 (18), 并从中抽样得到新的粒子集

$\{\mathbf{X}_t^{[m]}\}$ ;

2) 计算权值: 根据式 (31) 计算每个粒子的权值;

3) 计算  $N_{\text{eff}}$ : 在获取所有粒子的权值后, 根据式 (32) 计算有效粒子数;

4) 是否进行重抽样: 比较  $N_{\text{eff}}$  与给定域值的大小: 如前者小于后者, 则将具有归一化权值的样本集  $\{(\mathbf{X}_t^{[m]}, w_t^{[m]})\}$  映射为具有等权值的样本集  $\{(\mathbf{X}_t^{*[j]}, 1/N)\}$ ; 反之则不进行重抽样;

5) 地图管理和扩展: 针对每个粒子, 将候选特征列表中获得确认的真实特征移入地图内, 并删除获得确认的虚假特征. 将首次观测到的特征加入候选特征列表中.

6) 特征更新: 根据当前的定位结果与前一时刻已创建的地图和候选特征列表, 利用当前观测数据, 借助第 2.2 节中的方法更新特征状态;

7) 重复执行上述过程, 直到无新的观测为止.

假设以  $M$  个粒子近似目标函数的后验分布, 特征的总个数为  $N$ , 传感器一次观测到  $n$  个特征, 其中  $p$  个已存在于地图中. 在利用 CDFastSLAM 算法进行机器人同时定位与地图创建时: 由于用于线性回归计算的点集所包含的点数较少, 当  $M$  较大时, 可认为线性回归计算以及计算点集的复杂度为常数. 因此计算建议分布函数的复杂度为  $O(Mp)$ . 而计算粒子权值的复杂度正比于粒子个数  $O(M)$ . 对某个特征进行数据关联和更新或初始化的复杂度为  $O(M)$ , 即更新地图的总复杂度为  $O(Mn)$ . 为了确定是否需要进行重抽样, 需要计算有效粒子数  $N_{\text{eff}}$ , 复杂度为  $O(M)$ . 由于每个粒子均保存着各自的特征地图, 故重抽样的复杂度为  $O(MN)$ , 利用文献 [9] 中的二叉树结构存储特征, 复杂度可下降至  $O(M \log(N))$ . 本文引入了自适应抽样策略, 大大降低滤波过程中所需抽样的次数, 因此降低了 CDFastSLAM 算法总的复杂度. 通过分析可知, CDFastSLAM 算法的复杂度等价于 FastSLAM 2.0 算法的复杂度.

## 3 仿真及实验结果

为了进一步了解并比较不同算法之间的性能, 我们选取了文献 [9] 中的 FastSLAM 2.0, 文献 [12] 中的 UFastSLAM 算法, 并基于仿真实验和实际数据的实验与本文提出的 CDFastSLAM 算法进行了比较. 在后续的叙述中, 我们不对四轮小车和机器人进行区分. 仿真程序的运行环境为 Matlab.

### 3.1 仿真结果

在文献 [18] 中, T. Bailey 等公布了用于测试 SLAM 算法仿真软件. 借助此软件, 可以比较不同 SLAM 算法的性能. 其中 FastSLAM 2.0 算法采用软件包自带的程序, 在此基础之上我们编写了 UFastSLAM 算法与 CDFastSLAM 算法的程序. 图 1 所示为仿真环境图, 其中 “\*” 为点特征, 曲线为小车行使的参考路径. 小车从原点开始按参考路径运行一圈后停止. 为了消除数据关联不当引入的误差, 在仿真程序中均认为数据关联结果已知. 小车的前后轴距离为 4 m. 传感器的有效距离为 40 m, 并具有  $180^\circ$  的前视角. 在滤波的过程中, 通过判断有效粒子数决定是否进行重抽样: 即当  $N_{\text{eff}} < M/2$  时, 进行重抽样.

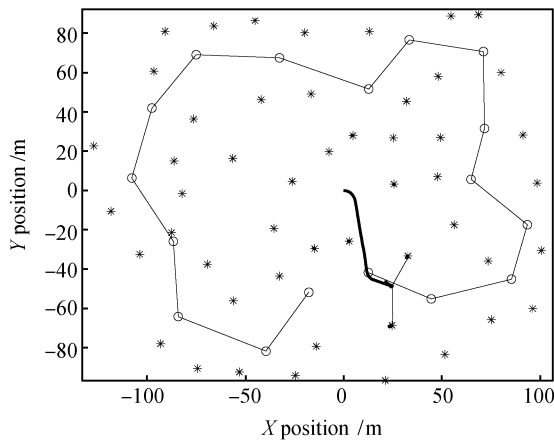


图 1 包含 51 个点特征的仿真环境

Fig. 1 The simulated environment with 51 point features

在第一组仿真实验中, 里程计的噪声均设为 ( $\sigma_V = 0.3 \text{ m/s}$ ;  $\sigma_G = 3^\circ$ ), 观测噪声设为 ( $\sigma_r = 0.2 \text{ m}$ ;  $\sigma_\theta = 3^\circ$ ). 针对三种不同的算法, 分别用三组粒子数: 1)  $M = 10$ ; 2)  $M = 50$ ; 3)  $M = 100$  进行实验. 为了消除随机性, 每种算法在不同的粒子数下分别进行 30 次有效的蒙特卡罗实验. 图 2 为不同算法, 在不同粒子数下关于小车位置估计的均方根 (Root mean square, RMS) 误差结果图. 其中线棒代表方差, 曲线代表平均值.

在第二组仿真实验中, 假设粒子数均为 20 个. 里程计的噪声均设为 ( $\sigma_V = 0.3 \text{ m/s}$ ;  $\sigma_G = 3^\circ$ ), 观测噪声分别为: 1) ( $\sigma_r = 0.1 \text{ m}$ ;  $\sigma_\theta = 1^\circ$ ); 2) ( $\sigma_r = 0.2 \text{ m}$ ;  $\sigma_\theta = 3^\circ$ ); 3) ( $\sigma_r = 0.2 \text{ m}$ ;  $\sigma_\theta = 6^\circ$ ); 4) ( $\sigma_r = 0.3 \text{ m}$ ;  $\sigma_\theta = 8^\circ$ ); 为了消除随机性, 每种算法在不同的噪声下分别进行 50 次有效的蒙特卡罗实验. 图 3 为不同算法在不同噪声下小车位置估计

的 RMS 误差结果图, 其中线棒代表方差, 柱状图代表平均值.

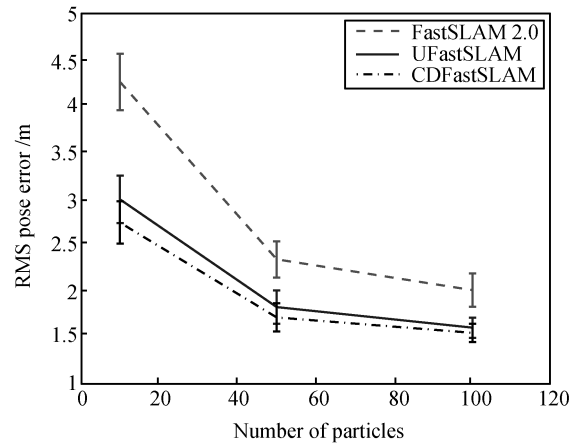


图 2 不同粒子数下的 SLAM 算法比较结果

Fig. 2 The comparison results of SLAM algorithms based on different numbers of particles

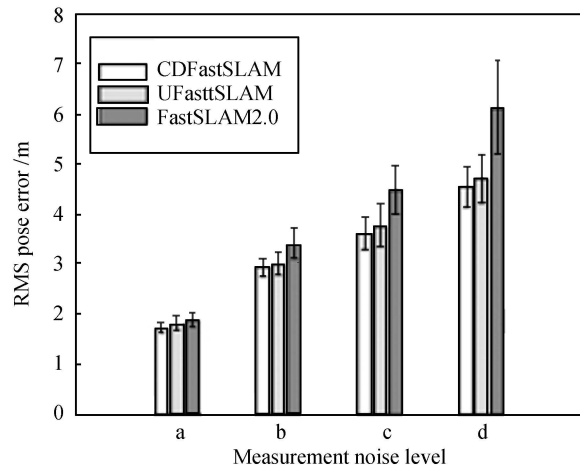


图 3 不同噪声水平下的 SLAM 算法比较结果

Fig. 3 The comparison results of SLAM algorithms based on different noise levels

由上述两组仿真实验的结果可知, 本文所提出的算法在精度和稳定性方面均优于同类型的另外两种算法. 由于 FastSLAM 算法利用一阶泰勒级数近似非线性系统, 故系统状态的估计误差较大. CDFastSLAM 算法利用带权重的点集, 并借助一系列的线性回归计算, 改进了粒子滤波中的建议分布函数, 提高了特征状态估计精度, 从而改进了算法的性能. 由于后两种算法均是以二阶精度逼近非线性方程, 故估计结果的精度相差并不大. CDFastSLAM 算法需确定一个参数  $h$ , 对于高斯随机变量, 该参数有确切的最优值<sup>[13]</sup>. 而 UFastSLAM 算法需要根据待估计变量的性质 (是否为高斯随机变量), 以及系

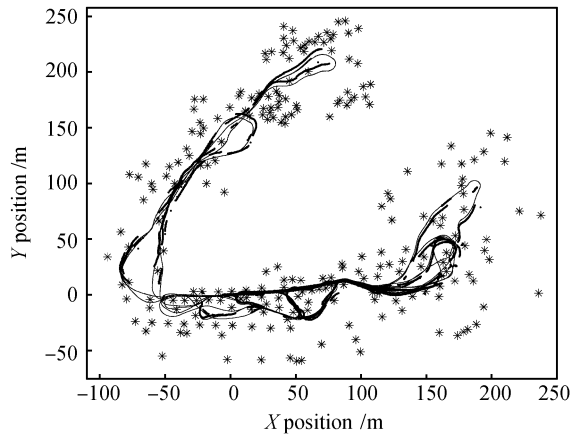
统状态方程的非线性度同时确定三个参数, 这些参数的选取将直接影响到该算法的精度<sup>[13]</sup>. 由于不同系统的非线性程度存在差异, 且目前并没有一种能精确量化某一系统的非线性程度的方法, 故参数的确定存在主观性, 也因此会对 UFastSLAM 算法的精度造成影响. 针对不同的系统, 需要调整 UFastSLAM 算法的三个参数, 才能使得 UFastSLAM 算法性能接近于 CDFastSLAM 算法的性能.

### 3.2 实验结果

在实验部分, 我们采用维多利亚公园数据集<sup>[19]</sup>对上述三种算法进行比较. 该数据集由安装在小车前方的 SICK 公司的激光测距仪采集得到. 采集的地点为具有许多树特征的悉尼维多利亚公园. 该激

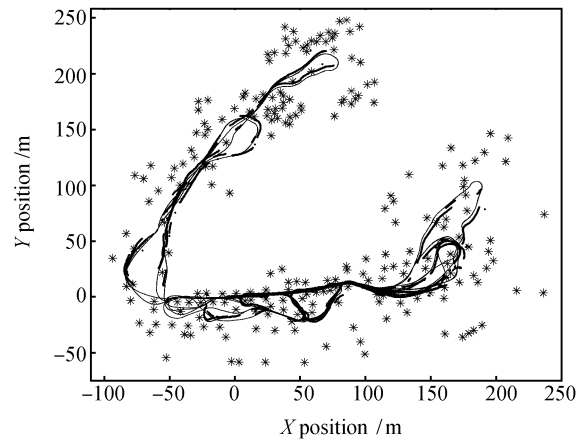
光测距仪具有  $180^\circ$  的前视角, 有效距离为 81 m. 利用文献 [19] 提供的树特征提取程序可从激光数据中提取点特征. 汽车轮子的线速度和转向角速度可由安装在汽车上的里程计测量获得. 为了验证算法的精度, 激光测距仪的正上方安装了 GPS 天线用于接受 GPS 信号. 由于树和建筑等物体的遮挡, 天线在部分位置不能收到信号. 但所收到的信号已足以用来验证 SLAM 算法的精度. 在程序中我们并不使用 GPS 信号估计小车的位置. 采集该数据集时, 小车运行了将近 30 min, 行使的距离约为 4 km. 受地面摩擦以及其他原因影响, 里程计读数噪声十分大, 由此估计出的小车路径与 GPS 信号估计出的小车路径差别巨大 (文献 [9] 给出了结果图).

在算法的程序实现中, 我们采用了最大似然分



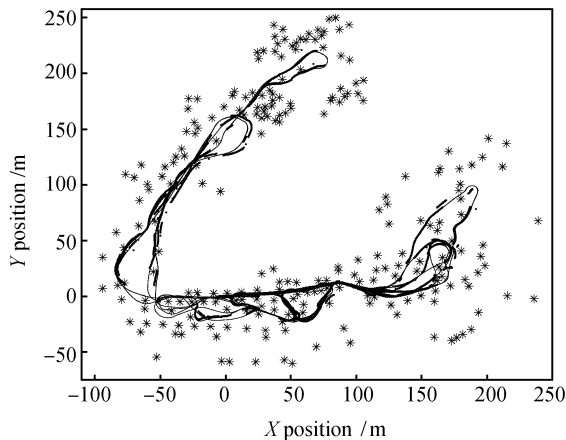
(a) 基于 FastSLAM 2.0 的实验结果

(a) Experimental results of FastSLAM 2.0



(b) 基于 UFastSLAM 的实验结果

(b) Experimental results of UFastSLAM



(c) 基于 CDFastSLAM 的实验结果

(c) Experimental results of CDFastSLAM



(d) 配准了 Google 卫星图的结果

(d) Experimental results with Google earth

图 4 不同 SLAM 算法的实验比较结果

Fig. 4 The results of experimental comparison for different SLAM algorithms

析的数据关联方法<sup>[6]</sup>,并在 CDFastSLAM 中利用本文所提出的地图管理策略剔除虚假特征.为了降低程序运行时间,可利用二叉数结构存储特征<sup>[9]</sup>.实验中不同的算法均采用 10 个粒子近似小车和地图状态的联合后验概率密度函数.过程噪声和观测噪声均分别设为 ( $\sigma_V = 1 \text{ m/s}$ ;  $\sigma_G = 3^\circ$ ), ( $\sigma_r = 0.4 \text{ m}$ ;  $\sigma_\theta = 2^\circ$ ).三种算法的实验对比结果如图 4 所示,其中图 4(d) 为配准了 Google 卫星图后的 CDFastSLAM 实验结果.各图中“\*”为特征的位置估计,深且不连续的线为 GPS 测量获得的小车位置,淡且连续的线为 SLAM 算法估计获得的小车位置.上述三种算法的程序运行时间均小于 15 min,远小于采集该数据集所需要时间.从图中可知,FastSLAM 2.0 算法的结果最差,本文提出的算法实验结果略好于文献 [12] 的算法结果.从而进一步证实了本文所提出算法的有效性.

## 4 结论

如何获取优良的建议分布函数是基于粒子滤波的 SLAM 算法的关键.本文通过利用性能优良的中心差分滤波替代 FastSLAM 算法中性能较差的扩展卡尔曼滤波,结合有效的地图管理策略,提高了机器人状态估计的精度,获得了改进的建议分布函数,同时也提高了特征状态估计的精度.这些改进都将提高 SLAM 算法中的数据关联结果可靠性,从而确保了算法的稳定性.最后通过一系列的仿真实验和用于测试 SLAM 算法基准数据集的实验,验证了该算法在精度和可靠性方面优于其他同类算法的结论.

## 致谢

感谢悉尼大学研究者提供的 SLAM 仿真软件包,以及维多利亚公园数据和数特征提取算法的程序.

## References

- 1 Smith R, Self M, Cheeseman P. A stochastic map for uncertain spatial relationships. In: Proceedings of the 4th International Symposium of Robotics Research. California, USA: MIT Press, 1987. 467–474
- 2 Thrun S, Burgard W, Fox D. *Probabilistic Robotics*. Cambridge: MIT Press, 2005
- 3 Durrant-Whyte H F, Bailey T. Simultaneous localization and mapping: part I. *IEEE Transactions on Robotics and Automation*, 2006, **13**(2): 99–108
- 4 Bailey T, Durrant-Whyte H F. Simultaneous localization and mapping: part II. *IEEE Transactions on Robotics and Automation*, 2006, **13**(2): 108–117
- 5 Dissanayake M W M G, Newman P, Clark S, Durrant-Whyte H F, Csorba M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 2001, **17**(3): 229–241
- 6 Bailey T. Mobile Robot Localisation and Mapping in Extensive Outdoor Environments [Ph.D. dissertation], University of Sydney, Australia, 2002
- 7 Doucet A, de Freitas N, Murphy K P, Russell S J. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. Stanford, USA: Morgan Kaufmann Publishers, 2000. 176–183
- 8 Montemerlo M, Thrun S, Koller D, Wegbreit B. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Proceedings of the AAAI National Conference on Artificial Intelligence. Edmonton, Canada: Springer, 2002. 1–6
- 9 Montemerlo M, Thrun S, Koller D, Wegbreit B. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence. Acapulco, Mexico: Springer, 2003. 1151–1156
- 10 Julier S, Uhlmann J, Durrant-Whyte H F. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 2000, **45**(3): 477–482
- 11 Xiang W, Hong Z. A UPF-UKF framework for SLAM. In: Proceedings of the Conference on Robotics and Automation. Roma, Italy: IEEE, 2007. 1664–1669
- 12 Kim C, Sakthivel R, Chung W K. Unscented FastSLAM: a robust and efficient solution to the SLAM problem. *IEEE Transactions on Robotics*, 2008, **24**(4): 808–820
- 13 van der Merwe R. Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models [Ph.D. dissertation], Oregon Health and Science University, USA, 2004
- 14 Ito K, Xiong K. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 2000, **45**(5): 910–927
- 15 Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 2007, **23**(1): 34–46
- 16 Bailey T. SLAM simulations [Online], available: <http://www-personal.acfr.usyd.edu.au/tbailey/>, November 12, 2008
- 17 Doucet A, de Freitas N, Gordon N. *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001



- 18 Beevers K R, Huang W H. Fixed-lag sampling strategies for particle filtering SLAM. In: Proceedings of the Conference on Robotics and Automation. Roma, Italy: IEEE, 2007. 2433–2438
- 19 Guivant J, Nebot E. Bench mark SLAM data set: Victoria park data set [Online], available: [http://www-personal.acfr.usyd.edu.au/nebot/victoria\\_park.htm](http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm), November 12, 2008



祝继华 西安交通大学人工智能与机器人研究所博士研究生. 主要研究方向为概率机器人. 本文通信作者.

E-mail: zhujh@aiar.xjtu.edu.cn

(**ZHU Ji-Hua** Ph.D. candidate at the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University.

His main research interest is probabilistic robotics. Corresponding author of this paper.)

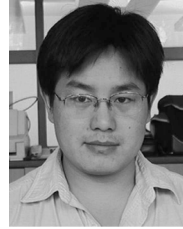


郑南宁 西安交通大学人工智能与机器人研究所教授. 主要研究方向为模式识别、机器视觉与图象处理和智能信息处理系统.

E-mail: nnzheng@mail.xjtu.edu.cn

(**ZHENG Nan-Ning** Professor at the Institute of Artificial Intelligence

and Robotics, Xi'an Jiaotong University. His research interest covers pattern recognition, machine vision and image processing, and hardware implementation of intelligent systems.)

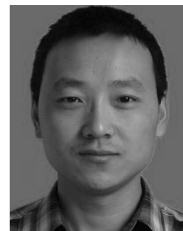


袁泽剑 西安交通大学人工智能与机器人研究所副教授. 主要研究方向为计算机视觉与机器学习.

E-mail: zjyuan@mail.xjtu.edu.cn

(**YUAN Ze-Jian** Associate professor at the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong

University. His research interest covers computer vision and machine learning.)



张强 西安交通大学人机所硕士研究生. 主要研究方向为移动机器人的环境感知.

E-mail: q.zhang.aiar@stu.xjtu.edu.cn

(**ZHANG Qiang** Master student at the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University.

His main research interest is perception of mobile robot.)