

平方根容积 Rao-Blackwillised 粒子滤波 SLAM 算法

宋宇^{1,2} 李庆玲³ 康轶非¹ 闫德立¹

摘要 面向大尺度环境中的移动机器人同时定位与地图构建 (Simultaneous localization and mapping, SLAM) 问题, 提出平方根容积 Rao-Blackwillised 粒子滤波 SLAM 算法. 算法主要特点在于: 1) 采用容积律计算 SLAM 中的非线性函数高斯权重积分, 达到减小 SLAM 非线性模型线性化误差、提高 SLAM 精度的目的; 2) 在 SLAM 中直接传播误差协方差矩阵的平方根因子, 避免了耗费时间的协方差矩阵分解与重构过程, 提高了 SLAM 计算效率. 通过仿真、实验将提出的 SLAM 算法与 FastSLAM2.0、UFastSLAM 两种算法进行对比, 结果表明本文算法在 SLAM 性能上优于另两者.

关键词 移动机器人, 同时定位与地图构建, 粒子滤波, 容积律, 高斯权重积分

引用格式 宋宇, 李庆玲, 康轶非, 闫德立. 平方根容积 Rao-Blackwillised 粒子滤波 SLAM 算法. 自动化学报, 2014, 40(2): 357–367

DOI 10.3724/SP.J.1004.2014.00357

SLAM with Square-root Cubature Rao-Blackwillised Particle Filter

SONG Yu^{1,2} LI Qing-Ling³ KANG Yi-Fei¹ YAN De-Li¹

Abstract In this paper, we derive a new large-scale environment simultaneous localization and mapping (SLAM) algorithm based on square-root cubature Rao-Blackwillised particle filter. The main contributions are: 1) to enhance the SLAM performance, the effective cubature rule is utilized to calculate the Gaussian weighted integral of the nonlinear function; 2) the covariance square-root factors are directly propagated in our SLAM process. Hence, the time-expensive decompositions on covariance matrixes are avoided. The performance of the proposed algorithm is compared with FastSLAM2.0 and UFastSLAM using a serial simulations and experiments. Results show that the proposed SLAM outperforms FastSLAM2.0 and UFastSLAM.

Key words Mobile robot, simultaneous localization and mapping (SLAM), particle filter, cubature rule, Gaussian weighted integral (GWI)

Citation Song Yu, Li Qing-Ling, Kang Yi-Fei, Yan De-Li. SLAM with square-root cubature Rao-Blackwillised particle filter. *Acta Automatica Sinica*, 2014, 40(2): 357–367

即时定位与地图构建 (Simultaneous localization and mapping, SLAM) 是“鸡生蛋、蛋生鸡”的悖论问题, 一方面移动机器人利用自载传感器感知未知环境, 形成对局部环境的理解, 并增量式创建

全局环境地图; 另一方面, 机器人利用对已建立环境地图的重复观测机制来处理不确定信息 (包括运动和观测的不确定性), 实现对其自身准确定位. 近 20 年来, SLAM 问题已经成为国际机器人与自动化领域的研究热点, 并被认为是实现移动机器人真正自主的“圣杯”^[1–2].

基于 Rao-Blackwellized particle filter 的 FastSLAM 算法为学者 Montemerlo 于 2003 年提出^[3]. 相比于传统的卡尔曼滤波 SLAM 方法, FastSLAM 具有计算代价小、适用于大尺度环境、对数据关联误差鲁棒等优点, 已经成为 SLAM 研究中流行的方法框架. FastSLAM 算法根据 SLAM 问题的条件独立特性, 将高维的移动机器人轨迹和环境地图联合后验概率密度估计解耦为低维状态估计, 从而解决了状态空间 SLAM 算法的维数灾难问题, 提高了 SLAM 的求解效率. FastSLAM 算法有两个版本: FastSLAM1.0 和 FastSLAM2.0. FastSLAM1.0 利用粒子滤波器跟踪移动机器人位姿参数, 并且每一粒子单独维护一幅环境特征地图、每一个环境路标

收稿日期 2012-09-05 录用日期 2013-02-04
Manuscript received September 5, 2012; accepted February 4, 2013

国家自然科学基金 (60905055, 61005070), 哈尔滨工业大学机器人与系统国家重点实验室开放基金 (SKLRS-2009-ZD-04), 中央高校基本科研业务费 (2014JBM014) 资助

Supported by National Natural Science Foundation of China (60905055, 61005070), Open Function of State Key Laboratory of Robotics and System of Harbin Institute of Technology (SKLRS-2009-ZD-04), and Fundamental Research Funds for the Central Universities of China (2014JBM014)

本文责任编辑 胡小平

Recommended by Associate Editor HU Xiao-Ping

1. 北京交通大学电子信息工程学院 北京 100044 2. 哈尔滨工业大学机器人与系统国家重点实验室 哈尔滨 150080 3. 中国矿业大学 (北京) 机电与信息工程学院 北京 100083

1. School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044 2. State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150080 3. School of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing 100083

利用卡尔曼滤波 (Extended Kalman filter, EKF) 来维护. Montemerlo 在其学位论文中指出, 当移动机器人的环境观测噪声较小时 (即环境观测传感器过于精确, 如激光传感器), FastSLAM1.0 容易发生粒子集退化, 使得 SLAM 发散^[3]. 为克服 FastSLAM1.0 存在的上述问题, FastSLAM2.0 算法利用 EKF 将当前机器人的环境观测信息融入粒子滤波器的提议分布设计之中, 使得粒子集中地分布于高观测似然区域. 结果表明, FastSLAM2.0 较好地解决了粒子退化问题, 使用较少的粒子就可得到满意的 SLAM 结果. 基于类似的学术思想, 学者 Grisetti 等提出栅格地图模型 FastSLAM 算法^[4], 该方法联合激光雷达扫描匹配技术和机器人里程计信息来优化粒子滤波器的提议分布, 实现了提高有效粒子数和 SLAM 精度的目的; Sim 等提出了立体视觉 FastSLAM 算法^[5], 该算法的主要特点在于利用立体视觉里程计来设计粒子滤波器的重要性函数; Kim 等提出了 UFastSLAM 算法^[6], 其核心在于利用无迹变换^[7] 来估计 SLAM 中的转移概率密度, 与其类似还有中心差分粒子滤波 FastSLAM 算法^[8]; 在 UFastSLAM 基础上, Kim 等在文献 [9] 中还对粒子滤波器提议分布设计问题进行了探讨, 更正了原始 UFastSLAM 中的过信估计问题, 给出了改进型的 UFastSLAM 算法. 在我们的前期工作中, 提出利用容积卡尔曼滤波计算 FastSLAM 的提议分布并维护环境特征地图, 初步探讨了 CFastSLAM 算法^[10]. 值得说明的是, 与传统 SLAM 算法相似, 在 CFastSLAM 算法中会以协方差形式来传播 SLAM 的不确定性. 因而, 为利用容积律计算转移概率密度, 需要在 SLAM 滤波的迭代过程中 (包括预测和更新) 不断执行数值敏感且繁琐的协方差矩阵平方根分解、协方差矩阵重构操作. 为此, 在继承 CFastSLAM 优点的基础上, 探讨如何在 SLAM 中实现直接传播协方差平方根因子, 避免协方差矩阵平方根分解、协方差重构过程, 进一步提高 SLAM 性能, 是本文工作的基本出发点.

面向特征地图 SLAM 问题, 提出平方根容积 FastSLAM 算法, 并通过仿真、实验验证了其性能. 算法主要特点在于: 1) 利用容积律计算非线性转移概率密度, 继承了 CFastSLAM 的高阶矩估计精度、无需计算雅可比矩阵等优点; 2) 粒子结构由 SLAM 状态及其不确定性协方差平方根因子构成, 基于容积律的数值特性, 深入探讨并实现了在 SLAM 迭代过程中的直接协方差平方根因子传播, 避免了 SLAM 迭代中数值敏感且繁琐的协方差平方根分解与重构过程, 并可严格保证 SLAM 协方差矩阵的正定性与对称性.

1 背景知识

1.1 FastSLAM 算法框架

FastSLAM 算法的核心思想在于利用机器人对环境观测的历史信息 $z^k = z_1, \dots, z_k$ 和机器人的控制的历史信息 $u^{k-1} = u_1, \dots, u_{k-1}$ 来估计环境地图 $\Theta = \theta_1, \dots, \theta_M$ (环境路标的数量用 M 来表示) 与机器人运动轨迹 $s^k = s_1, \dots, s_k$ 的联合后验概率分布 $p(s^k, \Theta | z^k, u^{k-1})$. 基于 SLAM 问题的条件独立性结论: 若移动机器人轨迹 s^k 已知, 环境特征路标 $\Theta = \theta_1, \dots, \theta_M$ 相互独立. 因而, 联合后验分布 $p(s^k, \Theta | z^k, u^{k-1})$ 可以表示为

$$p(s^k, \Theta | z^k, u^{k-1}) = \prod_{m=1}^M p(\theta_m | s^k, z^k, u^{k-1}) \quad (1)$$

基于上式表达的条件独立特性, FastSLAM 算法采用不同的估计器来估计机器人轨迹后验概率与环境路标的后验概率: 对于轨迹后验 $p(s^k | z^k, u^{k-1})$, 采用 N 个粒子的粒子滤波器估计, 其中, 每一个粒子单独维护一幅环境地图 Θ 和机器人路径 s^k 的假设; 在机器人轨迹 s^k 已知前提下, 由于路标之间相互独立, 则对每个环境路标后验 $p(\theta_m | s^k, z^k, u^{k-1})$ 利用单独的卡尔曼滤波器估计. 这样, FastSLAM 共由 $N \times M + 1$ 个滤波器组成.

FastSLAM 算法中每一个粒子可以表述为

$$S_k^{[i]} = \langle (s_k^{[i]}, P_k^{[i]}), \dots, (\mu_k^{[i][m]}, \Sigma_k^{[i][m]}), \dots \rangle \quad (2)$$

式中, $S_k^{[i]}$ 表示 FastSLAM 算法中的索引为 i 的粒子; $s_k^{[i]}$ 为该粒子对机器人当前位姿的假设, $P_k^{[i]}$ 为 $s_k^{[i]}$ 的协方差; $(\mu_k^{[i][m]}, \Sigma_k^{[i][m]})$ 为该粒子所维护的特征地图中第 m 个路标的全局坐标及其不确定性协方差.

1.2 高斯权重积分的容积律方法

为实现高斯域中的贝叶斯滤波, 最为重要的步骤是计算高斯分布的非线性转移密度, 即计算高斯权重积分 (Gaussian weighted integral, GWI), 表示为

$$I = \int g(\mathbf{x}) \mathcal{N}(\mathbf{x}, P_x) d\mathbf{x} \quad (3)$$

其中, $\mathbf{x} \in \mathbf{R}^{n_x}$, n_x 代表变量 \mathbf{x} 的维数, $g(\mathbf{x})$ 表示非线性函数; $\mathcal{N}(\mathbf{x}, P_x)$ 代表 n_x 维高斯分布.

在文献 [11] 中, 学者 Arasaratnam 等面向 GWI 的求解问题提出了容积律方法, 该方法利用 $2n_x$ 个等权重容积点计算积分 I :

$$I \approx \frac{1}{2n_x} \sum_{j=1}^{2n_x} g(\sqrt{P_x} \zeta_j + \mathbf{x}) \quad (4)$$

式中, 容积点集为 $\sqrt{P_x}\zeta_j + \mathbf{x}$, 点集 ζ_j 表示为

$$\{\zeta_j\} = \sqrt{n_x}\{[\mathbf{1}]_j\}, \quad j = 1, 2, \dots, 2n_x \quad (5)$$

式中, $\{[\mathbf{1}]_j\}$ 代表 n_x 维笛卡尔坐标系的坐标轴与 n_x 维单位超球的交点坐标。

2 提出的 SLAM 算法

为研究 SLAM 问题, 不失一般性地定义移动机器人系统的含噪声运动学方程和环境观测方程为

$$\begin{cases} \mathbf{s}_k = f(\mathbf{s}_{k-1}, \mathbf{u}_{k-1} + \delta\mathbf{u}_{k-1}) \\ \mathbf{z}_k = h(\mathbf{s}_k, \boldsymbol{\mu}_k^{[m]}) + \delta\mathbf{z}_k \end{cases} \quad (6)$$

其中, $\mathbf{s}_k \in \mathbf{R}^{n_s}$ 为机器人位姿参数; $\mathbf{z}_k \in \mathbf{R}^{n_z}$ 为环境观测; f 和 h 分别为非线性机器人运动学模型和环境观测模型; $\mathbf{u}_{k-1} \in \mathbf{R}^{n_u}$ 为在离散时间段 $[k, k-1]$ 内对机器人的控制作用, $\delta\mathbf{u}_{k-1} \in \mathbf{R}^{n_u}$ 为控制噪声, 其协方差 $Q \in \mathbf{R}^{n_u \times n_u}$; $\delta\mathbf{z}_k \in \mathbf{R}^{n_z}$ 为观测噪声, 其协方差为 $R \in \mathbf{R}^{n_z \times n_z}$; $\boldsymbol{\mu}_k^{[m]} \in \mathbf{R}^{n_\mu}$ 代表第 m 个环境路标的全局坐标系位置参数。

不同于由式 (2) 所定义的传统 FastSLAM 算法粒子结构, 提出算法的粒子结构中利用协方差矩阵平方根因子代替传统算法粒子结构中的协方差矩阵, 并且在 SLAM 迭代过程中利用平方根因子传播代替传统算法中的协方差传播。提出算法中的粒子结构为

$$S_k^{[i]} = \langle (\mathbf{s}_k^{[i]}, v_k^{[i]}, \dots, (\boldsymbol{\mu}_k^{[i][m]}, l_k^{[i][m]}), \dots) \rangle \quad (7)$$

其中, $\mathbf{s}_k^{[i]}$ 为该粒子对机器人当前位姿的假设, $v_k^{[i]}$ 为协方差矩阵 $P_k^{[i]}$ 的平方根因子, 满足 $v_k^{[i]}(v_k^{[i]})^T = P_k^{[i]}$; $\boldsymbol{\mu}_k^{[i][m]}$ 为该粒子中索引为 m 的环境路标位置, $l_k^{[i][m]}$ 为其协方差矩阵 $\Sigma_k^{[i][m]}$ 的平方根因子, 满足 $l_k^{[i][m]}(l_k^{[i][m]})^T = \Sigma_k^{[i][m]}$ 。

2.1 机器人状态估计

2.1.1 状态预测

为预测 k 时刻机器人位姿状态, 需要联合控制输入 \mathbf{u}_{k-1} 及其不确定性协方差的平方根因子来增广 $k-1$ 时刻机器人状态及其协方差平方根因子

$$\mathbf{a} = \begin{bmatrix} \mathbf{s}_{k-1}^{[i]} \\ \mathbf{u}_{k-1} \end{bmatrix}, \quad V_a = \begin{bmatrix} v_{k-1}^{[i]} & 0 \\ 0 & S_Q \end{bmatrix} \quad (8)$$

式中, $\mathbf{a} \in \mathbf{R}^{n_a}$ 、 $V_a \in \mathbf{R}^{n_a \times n_a}$ ($n_a = n_s + n_u$) 分别为增广机器人状态及增广协方差平方根因子; $\mathbf{s}_{k-1}^{[i]}$ 、 $v_{k-1}^{[i]}$ 为 $k-1$ 时刻的机器人状态及协方差因子; S_Q 为控制噪声的协方差平方根因子, 满足 $S_Q S_Q^T = Q$ 。增广状态 \mathbf{a} 服从多维高斯分布 $\mathcal{N}(\mathbf{a}, V_a V_a^T)$ 。

为计算机器人在 $k-1$ 的状态 $\mathbf{s}_{k-1}^{[i]}$ 经运动学模型 f 传播后的转移概率密度, 需按下式计算多维高斯分布 $\mathcal{N}(\mathbf{a}, V_a V_a^T)$ 的容积点集 $\{\mathbf{C}_j^{[i]}\}$:

$$\{\mathbf{C}_j^{[i]}\} = \{V_a \zeta_j + \mathbf{a}\}, \quad j = 1, 2, \dots, 2n_a \quad (9)$$

式中, j 为容积点索引。这里值得注意的是, 由于算法以协方差平方根因子传播代替协方差传播不确定性传播过程, 上式计算容积点时, 无需执行数值敏感且计算复杂度较高的协方差矩阵的平方根分解操作。

将容积点 $\mathbf{C}_j^{[i]}$ 记为机器人状态 \mathbf{s} 和控制作用 \mathbf{u} 的组合形式, 即 $\mathbf{C}_j^{[i]} = [\mathbf{C}_j^{s[i]}, \mathbf{C}_j^{u[i]}]^T$, 并通过机器人运动学模型 f 传递容积点集, 得到变换容积点集 $\{\mathbf{C}_j^{*[i]}\} = \{f(\mathbf{C}_j^{s[i]}, \mathbf{C}_j^{u[i]})\}$ 。则机器人在 k 时刻的状态预测为

$$\mathbf{s}_{k|k-1}^{[i]} = \int f(\mathbf{a}) \mathcal{N}(\mathbf{a}, V_a V_a^T) d\mathbf{a} \approx \frac{1}{2n_a} \sum_{j=1}^{2n_a} \mathbf{C}_j^{*[i]} \quad (10)$$

为预测机器人状态协方差的平方根因子 $v_{k|k-1}^{[i]}$, 定义误差矩阵 $Err \in \mathbf{R}^{n_s \times 2n_a}$

$$Err = \frac{1}{\sqrt{2n_a}} [\dots, \mathbf{C}_j^{*[i]} - \mathbf{s}_{k|k-1}^{[i]}, \dots] \quad (11)$$

通过对 Err^T 进行 QR 分解, 机器人状态协方差平方根因子 $v_{k|k-1}^{[i]}$ 为上三角方阵的转置, 表示为

$$[q, r] = qr(Err^T), v_{k|k-1}^{[i]} = r^T \quad (12)$$

式中, $qr(\cdot)$ 为 QR 分解操作符; r 为 $n_s \times n_s$ 上三角阵。

2.1.2 状态更新

当索引为 m 的环境路标被移动机器人重新观测到后, 利用机器人对该路标的重复观测信息来更新预测状态 $\mathbf{s}_{k|k-1}^{[i]}$ 及其协方差平方根因子 $v_{k|k-1}^{[i]}$ 。首先, 联合环境路标状态及其协方差平方根因子对 $\mathbf{s}_{k|k-1}^{[i]}$ 和 $v_{k|k-1}^{[i]}$ 进行增广。

$$\mathbf{b} = \begin{bmatrix} \mathbf{s}_{k|k-1}^{[i]} \\ \boldsymbol{\mu}_{k-1}^{[i][m]} \end{bmatrix}, \quad V_b = \begin{bmatrix} v_{k|k-1}^{[i]} & 0 \\ 0 & l_{k-1}^{[i][m]} \end{bmatrix} \quad (13)$$

式中, $\mathbf{b} \in \mathbf{R}^{n_b}$ 、 $V_b \in \mathbf{R}^{n_b \times n_b}$ ($n_b = n_s + n_\mu$) 为增广机器人状态和增广协方差平方根因子。增广状态 \mathbf{b} 服从多维高斯分布 $\mathcal{N}(\mathbf{b}, V_b V_b^T)$ 。与式 (9) 类似, 分布 $\mathcal{N}(\mathbf{b}, V_b V_b^T)$ 的容积点集为

$$\{\mathbf{X}_j^{[i][m]}\} = \{V_b \zeta_j + \mathbf{b}\}, \quad j = 1, 2, \dots, 2n_b \quad (14)$$

将每一个容积点表述为机器人状态和环境路标状态的组合形式 $\mathbf{X}_j^{[i][m]} = [\mathbf{X}_j^{s[i][m]}, \mathbf{X}_j^{\mu[i][m]}]^T$, 并利用机器人环境观测模型 h 传播容积点集, 得到变换

容积点集 $\{\mathbf{X}_j^{*[i][m]}\} = \{h(\mathbf{X}_j^{s[i][m]}, \mathbf{X}_j^{\mu[i][m]})\}$, 进而机器人对该路标的观测预测为下式的高斯权重积分解:

$$\mathbf{z}_{k|k-1}^{[i][m]} = \int h(\mathbf{b})\mathcal{N}(\mathbf{b}, V_b V_b^T) d\mathbf{b} \approx \frac{1}{2n_b} \sum_{j=1}^{2n_b} \mathbf{X}_j^{*[i][m]} \quad (15)$$

为利用卡尔曼滤波技术更新机器人状态, 需计算移动机器人状态与环境路标观测间的交互协方差与观测新息协方差. 为此, 定义误差矩阵如下:

$$\begin{aligned} Err_z &= \frac{1}{\sqrt{2n_b}} [\dots, \mathbf{X}_j^{*[i][m]} - \mathbf{z}_{k|k-1}^{[i][m]}, \dots] \\ Err_s &= \frac{1}{\sqrt{2n_b}} [\dots, \mathbf{X}_j^{s[i][m]} - \mathbf{s}_{k|k-1}^{[i]}, \dots] \end{aligned} \quad (16)$$

进而, 机器人状态与观测间的交互协方差为

$$P_{sz}^{[i][m]} = Err_s Err_z^T \quad (17)$$

观测新息协方差平方根因子 $d_{zz}^{[i][m]}$ 由误差矩阵 $[Err_z, S_R]^T$ 的 QR 分解计算:

$$[q, r] = qr([Err_z, S_R]^T), d_{zz}^{[i][m]} = r^T \quad (18)$$

式中, r 为 $n_z \times n_z$ 上三角方阵; S_R 为观测噪声协方差 R 的平方根因子, 即 $S_R S_R^T = R$.

基于卡尔曼滤波算法, 机器人状态更新为

$$\begin{aligned} K_s &= P_{sz}^{[i][m]} (d_{zz}^{[i][m]} d_{zz}^{[i][m]T})^{-1} \\ \mathbf{s}_k^{[i]} &= \mathbf{s}_{k|k-1}^{[i]} + K_s (\mathbf{z}_k^{[i][m]} - \mathbf{z}_{k|k-1}^{[i][m]}) \end{aligned} \quad (19)$$

式中, $\mathbf{z}_k^{[i][m]}$ 为机器人在当前时刻 k 对索引为 m 的环境路标的真实传感器测量.

机器人状态的协方差平方根因子 $v_k^{[i]}$ 由误差矩阵 $[Err_s - K_s Err_z, K_s S_R]^T$ 的 QR 分解计算, 即

$$[q, r] = qr([Err_s - K_s Err_z, K_s S_R]^T), v_k^{[i]} = r^T \quad (20)$$

式中, r 为 $n_s \times n_s$ 上三角方阵; S_R 为观测噪声协方差 R 的平方根因子; K_s 为式 (19) 中计算的滤波增益.

2.1.3 重要性采样

如果当前时刻移动机器人同时观测到多个已存储在地图中的环境路标, 则依次根据对每一个路标的观测对机器人状态 $\mathbf{s}_k^{[i]}$ 及其协方差平方根因子 $v_k^{[i]}$ 进行更新 (依次执行式 (13) ~ (20)), 每次更新均以前次更新结果作为初始值. 更新完毕后, 从机器人状态分布中采集新一代粒子, 即

$$\mathbf{s}_k^{[i]} \sim \mathcal{N}(\mathbf{s}_k^{[i]}, v_k^{[i]} v_k^{[i]T}) \quad (21)$$

进而, 根据观测新息及新息协方差来计算新粒子的重要性权重 $\omega_k^{[i]}$:

$$\omega_k^{[i]} \propto \prod_{m=1} \exp \left\{ -\frac{(\mathbf{z}_k^{[m]} - \mathbf{z}_{k|k-1}^{[i][m]})^T (\mathbf{z}_k^{[m]} - \mathbf{z}_{k|k-1}^{[i][m]})}{d_{zz}^{[i][m]} d_{zz}^{[i][m]T}} \right\} \quad (22)$$

2.2 环境地图估计

对于提出算法中的每一个粒子 $S_k^{[i]}$, 在更新其中代表移动机器人位姿的部分后, 进入更新环境路标地图阶段. 在该阶段, 需对当前时刻的重复观测路标和首次观测路标采用不同的策略分别处理.

2.2.1 重复观测环境路标的更新

在当前时刻 k , 对于那些已经存储在地图中且被移动机器人重复观测到的环境路标 m , 其全局坐标的高斯描述为 $\mathcal{N}(\boldsymbol{\mu}_{k-1}^{[i][m]}, l_{k-1}^{[i][m]} l_{k-1}^{[i][m]T})$. 则基于容积律, 路标状态分布的容积点集为 $\{\mathbf{V}_j^{[i][m]}\} = \{l_{k-1}^{[i][m]} \boldsymbol{\zeta}_j + \boldsymbol{\mu}_{k-1}^{[i][m]}\}, j = 1, 2, \dots, 2n_\mu$. 容积点集 $\{\mathbf{V}_j^{[i][m]}\}$ 经非线性环境观测模型 h 作用后, 得到变换容积点集 $\{\mathbf{Z}_j^{*[i][m]}\} = \{h(\mathbf{s}_k^{[i]}, \mathbf{V}_j^{[i][m]})\}$. 因而, 在机器人位姿参数 $\mathbf{s}_k^{[i]}$ 已知的前提下, 机器人对索引为 m 的路标的观测预测计算为

$$\mathbf{z}_{k|k-1}^{*[i][m]} = \int h(\mathbf{s}_k^{[i]}, \boldsymbol{\mu}) \mathcal{N}(\boldsymbol{\mu}, l^T) d\boldsymbol{\mu} \approx \frac{1}{2n_\mu} \sum_{j=1}^{2n_\mu} \mathbf{Z}_j^{*[i][m]} \quad (23)$$

式中, $\mathbf{s}_k^{[i]}$ 为由式 (21) 计算.

与机器人状态更新类似, 定义误差矩阵:

$$\begin{aligned} Err_z^* &= \frac{1}{\sqrt{2n_\mu}} [\dots, \mathbf{Z}_j^{*[i][m]} - \mathbf{z}_{k|k-1}^{*[i][m]}, \dots] \\ Err_\mu^* &= \frac{1}{\sqrt{2n_\mu}} [\dots, \mathbf{V}_j^{[i][m]} - \boldsymbol{\mu}_{k-1}^{[i][m]}, \dots] \end{aligned} \quad (24)$$

则路标状态与环境测量间的交互协方差矩阵为

$$P_{\mu z}^{[i][m]} = Err_\mu^* Err_z^{*T} \quad (25)$$

观测新息协方差平方根因子 $d_{zz}^{*[i][m]}$ 由矩阵 $[Err_z^*, S_R]^T$ 的 QR 得到, 即

$$[q, r] = qr([Err_z^*, S_R]^T), d_{zz}^{*[i][m]} = r^T \quad (26)$$

基于卡尔曼滤波, 重复观测环境路标状态更新为

$$\begin{aligned} K_\mu &= P_{\mu z}^{[i][m]} (d_{zz}^{*[i][m]} d_{zz}^{*[i][m]T})^{-1} \\ \boldsymbol{\mu}_k^{[i][m]} &= \boldsymbol{\mu}_{k-1}^{[i][m]} + K_\mu (\mathbf{z}_k^{[i][m]} - \mathbf{z}_{k|k-1}^{*[i][m]}) \end{aligned} \quad (27)$$

式中, $\mathbf{z}_k^{[i][m]}$ 为机器人在当前时刻 k 对索引为 m 的环境路标的真实传感器测量.

环境路标的协方差平方根因子 $l_k^{[i][m]}$ 由对误差矩阵 $[Err_{\mu}^* - K_{\mu}Err_z^*, K_{\mu}S_R]^T$ 的 QR 分解得到:

$$[q, r] = qr([Err_{\mu}^* - K_{\mu}Err_z^*, K_{\mu}S_R]^T), l_k^{[i][m]} = r^T \quad (28)$$

其中, r 为 $n_{\mu} \times n_{\mu}$ 上三角方阵; S_R 为观测噪声协方差 R 的平方根因子; K_{μ} 为式 (27) 计算的滤波增益.

2.2.2 首次观测环境路标的初始化

对于首次观测到的路标 (这里记其索引为 n), 利用观测 $z_k^{[n]}$ 计算其全局坐标并加入粒子 $S_k^{[i]}$ 维护的全局路标地图. 由于环境测量 $z_k^{[n]}$ 不确定性分布的容积点集为 $\{F_j^{[i][n]}\} = \{S_R \zeta_j + z_k^{[n]}\}, j = 1, 2, \dots, 2n_z$, 则经逆观测模型 $\mu = h^{-1}(s, z)$ 传播后的变换容积点集为 $\{F_j^{*[i][n]}\} = \{h^{-1}(s_k^{[i]}, F_j^{[i][n]})\}$. 进而, 索引为 n 的环境路标状态 $\mu_k^{[i][n]}$ 为

$$\mu_k^{[i][n]} = \int h^{-1}(s_k^{[i]}, z) \mathcal{N}(z, R) dz \approx \frac{1}{2n_z} \sum_{j=1}^{2n_z} F_j^{*[i][n]} \quad (29)$$

为初始化该路标的协方差平方根因子, 定义 $n_z \times 2n_z$ 误差矩阵 Err_{μ} :

$$Err_{\mu} = \frac{1}{\sqrt{2n_z}} [\dots, F_j^{*[i][n]} - \mu_k^{[i][n]}, \dots] \quad (30)$$

通过对误差矩阵 Err_{μ}^T 执行 QR 分解, 路标状态协方差平方根因子 $l_k^{[i][n]}$ 计算为

$$[q, r] = qr(Err_{\mu}^T), l_k^{[i][n]} = r^T \quad (31)$$

式中, r 为 $n_{\mu} \times n_{\mu}$ 上三角方阵.

2.3 算法流程

综上, 提出的 SLAM 算法具体执行过程见算法 1.

算法 1. 本文提出的 SLAM 算法

给定粒子数 N , 控制噪声 Q , 观测噪声 R , 初始化粒子集 $S_0 = \emptyset$;

For $k = 1, 2, 3, \dots$ //对于每一离散时刻

For $i = 1, 2, 3, \dots$ //对于每一粒子

1) 索引粒子 $S_{k-1}^{[i]}$;

2) 根据式 (8)~(12) 来预测移动机器人状态及其协方差平方根因子;

3) 对当前机器人观测执行数据关联, 区分重复观测路标和首次观测路标;

4) 根据机器人对已存储在地图中路标的重复观测, 执行机器人状态更新 (式 (13)~(20));

5) 采样新粒子 (式 (21)) 并计算其重要性权重 (式 (22));

6) 对已存储在粒子中的重复观测路标, 根据式 (23)~(28) 更新其状态及其协方差平方根因子;

7) 对于首次观测路标, 根据式 (29)~(31) 初始化路标状态及其协方差平方根因子, 加入环境地图;

End for

基于粒子权重对粒子集执行重采样处理;

返回粒子集 $S_k^{[i]}$;

End for

2.4 提出算法的优势分析

非线性模型线性化引入的截断误差将在 SLAM 迭代过程中逐步累积. 因而, 为提高 SLAM 性能, 有效减小误差累积是值得关注的核心问题. 本质上, SLAM 中非线性模型线性化的目的在于计算高斯分布经非线性函数作用后的转移密度, 如式 (10)、(15)、(23)、(29), 其被积函数均具有非线性函数 \times 高斯密度的基本形式. 为说明该问题, 不失一般性地定义 SLAM 问题中涉及的非线性函数为 $y = g(x)$, 其中, $x \in \mathbf{R}^{n_x}$ 服从高斯分布 $\mathcal{N}(\hat{x}, P_x)$, 为计算转移密度 $\mathcal{N}(\hat{y}, P_y)$, 需要计算如下两个 GWI:

$$\begin{aligned} \hat{y} &= \int g(x) \mathcal{N}(\hat{x}, P_x) dx \\ P_y &= \int [g(x) - \hat{y}][g(x) - \hat{y}]^T \mathcal{N}(\hat{x}, P_x) dx \end{aligned} \quad (32)$$

在 FastSLAM2.0 中, 以上两个 GWI 是通过对非线性函数 $g(x)$ 在 \hat{x} 处进行线性化所得, 即 $g(x) = g(\hat{x}) + \nabla g(\hat{x})(x - \hat{x})$. 其中, $\nabla g(\hat{x})$ 为雅可比矩阵. 因而, 根据高斯分布的性质有 $\hat{y} = g(\hat{x})$, $P_y = \nabla g(\hat{x}) P_x \nabla g(\hat{x})^T$. 显而易见, 当 g 的非线性程度较高时, 转移密度 $\mathcal{N}(\hat{y}, P_y)$ 的计算将不准确. 对于 UFastSLAM 算法, 上述积分由 Sigma 支撑点集的线性回归计算, 由于 Sigma 点方法在理论上可以达到 3 阶非线性, UFastSLAM 的精度要高于 FastSLAM2.0. 然而, Sigma 点方法存在于数值稳定性差、难以保证协方差矩阵的正定性、需设定的自由参数多等问题, 限制了其在 SLAM 中的应用.

2.4.1 提出算法中利用容积律的优势

在矩特性方面, 容积律 (式 (4) 和 (5)) 是从数值积分观点提出的一种 GWI 求解方法^[11]. 已经证明了多变量矩积分 $\int x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \mathcal{N}(\hat{x}, P_x) dx$ (其中 x_j 指明向量 x 的 j -th 分量) 在阶数满足条件 $\sum_{j=1}^{i_n} i_j \leq 3$ 时可由容积律来精确计算. 对于非线性函数 $y = g(x)$, 通过对 $g(x)$ 在 $x = \hat{x}$ 处进行泰勒展开, 则积分 $\int g(x) \mathcal{N}(\hat{x}, P_x) dx$ 可以表示为多个矩积分的线性组合. 由于利用容积律可以精确计算展开式中所有前三阶矩积分, 故而从泰勒展开的意义上, 容积律具有 3 阶非线性估计精度.

从估计的矩特性角度, 提出 SLAM 算法中采用的容积律与 UFastSLAM 和 FastSLAM2.0 中分别采用的无迹变换、线性化方法的对比见表 1. 对于 SLAM 问题, 均为多维高斯情况, 因而在 SLAM 算

法中采用容积律可以达到与无迹变换相同的矩估计特性.

表 1 不同高斯权重积分求解方法的矩特性对比
Table 1 Moment characteristic for GWI solutions

GWI 解	1 维高斯 ($n_x = 1$)	高维高斯 ($n_x > 1$)
线性化方法	1 阶精度	1 阶精度
无迹变换	5 阶精度	3 阶精度
容积律	3 阶精度	3 阶精度

从数值积分的角度, GWI 数值 $\sum \omega_i g(\mathbf{x}_i)$ 解应满足的基本条件是: 所有权重 ω_i 非负. 如式 (4), 显而易见, 容积律方法中的所有容积点权重均为 $1/(2n_x)$, 满足该条件. 进而, 基于容积律计算的协方差矩阵 P_y 可保证其非负定性. 与之相比较, UFastSLAM 中采用的无迹变换并不能满足该条件. 当高斯维数大于 3 时 (对于 SLAM, 增广状态一般高于 3 维, 即 $n_x > 3$), 其中心 Sigma 点的权重 $\omega_0 = 1 - n_x/3 < 0$, 这将可能导致求解的协方差矩阵 P_y 负定, 进而无法求解其平方根, 导致 SLAM 无法执行. 另外, 如文献 [11] 所述, 对于数值积分解, 当稳定性因子 $\sum \omega_i / \sum |\omega_i|$ 大于 1 时, 在有限字长的计算机中实施数值计算会引入圆整误差. 对于容积律, 由于全部容积点的权重均为 $1/(2n_x)$, 因而其稳定性因子将始终为 1. 与之相比较, 无迹变换的稳定性因子为 $n_x/3 + |1 - n_x/3|$, 即当维数 $n_x \leq 3$ 时数值稳定性为 1, 而当 $n_x > 3$ 时, 其稳定性因子将伴随着维数的增长而线性增长, 数值计算稳定性变差.

提出 SLAM 中采用的容积律和 UFastSLAM 中采用的无迹变换的数值特性对比见表 2. 对于 SLAM 实际应用, 一般增广状态向量的维数高于 3 (在式 (8)、式 (13) 中, 机器人状态包括其坐标和航向角, 则增广后状态维数大于 3), 因而在 SLAM 中应用容积律较无迹变换具有数值计算上的优势: 数值精度高、可严格保证协方差的正定性.

表 2 容积律与无迹变换的数值特性对比
Table 2 Numerical characteristic for GWI solutions

数值特性	无迹变换	容积律
数值稳定性因子 ($n_x \leq 3$)	1	1
数值稳定性因子 ($n_x > 3$)	$\frac{2n_x}{3} - 1 > 1$	1
协方差 P_y 正定性 ($n_x \leq 3$)	可保证	可保证
协方差 P_y 正定性 ($n_x > 3$)	不确定	可保证

计算代价方面, 提出 SLAM 算法较基于无迹变换的 UFastSLAM 在计算代价方面也具有优势, 其原因在于: 1) 提出算法采用容积律计算 GWI, 对于相同的高斯维数 n_x , 容积点数为 $2n_x$, 数量低于

UFastSLAM 中无迹变换需要的 $2n_x + 1$ 个 Sigma 点; 2) 提出算法中的容积点为等权重点, 因而利用容积点计算 GWI 仅执行简单的平均操作 (1 次乘法操作), 而无迹变换中的 Sigma 点为非等权重点, 需要使用线性权重回归方式来计算 GWI.

2.4.2 提出算法中采用平方根因子传播的优势

利用容积律来计算高斯权重积分, 需要计算“容积点集”, 而计算“容积点集”需要利用到协方差矩阵的平方根因子, 如式 (4) 所说明. 因而, 为计算式 (32) 的非线性转移概率密度 $\mathcal{N}(\hat{\mathbf{y}}, P_y)$, 需要首先计算协方差矩阵 P_x 的平方根 $\sqrt{P_x}$, 并根据 $\sqrt{P_x}$ 计算容积点集. 容积点集经非线性函数 g 传播后, 最终用来重构均值 $\hat{\mathbf{y}}$ 及协方差 P_y , 完成转移概率密度的估计. 由文献 [12], 利用 Cholesky 分解来计算协方差矩阵 P_x 平方根的复杂度为 $O(\frac{n_x^3}{6})$, 且由于容积点的数量为 $2n_x$, 则重构协方差 P_y 的复杂度为 $O(2n_x^3)$, 即传播协方差矩阵的复杂度为 $O(2n_x^3 + n_x^3/6)$. 与其相比较, 本文算法在 SLAM 中直接传播协方差平方根因子 $\sqrt{P_x}$, 并利用 QR 分解实现协方差平方根因子的更新, 其计算复杂度为 $O(2n_x^3)$.

另外, 计算协方差矩阵的平方根 $\sqrt{P_x}$ 是一种应尽力避免的数值敏感操作. 为利用 Cholesky 分解计算 $\sqrt{P_x}$, 其基本前提是要严格保证协方差矩阵 P_x 的正定性. 如果在 SLAM 中传播协方差矩阵 P_x , 由于在计算过程难以避免引入圆整误差, 在卡尔曼滤波的协方差更新方程中并不能保证更新后协方差的正定性 (卡尔曼滤波算法的协方差更新方程具有矩阵差的形式, 形如 $P^+ = P - KQK^T$), 这将可能导致滤波过程的数值不稳定^[11-12], 甚至滤波器无法执行. 因而, 在卡尔曼滤波器协方差更新的具体实施中还需要协方差矩阵正定性的检验过程^[13]. 而提出 SLAM 算法采用直接传播协方差平方根因子 $\sqrt{P_x}$, 由其重构的协方差矩阵为 $\sqrt{P_x}\sqrt{P_x}^T$, 可以始终严格保证正定性与对称性. 因而, 在 SLAM 中直接传播协方差平方根因子比传播协方差矩阵具有数值计算稳定性上的明显优势. 再者, 本文算法中协方差平方根因子的预测和更新均由 QR 分解实现, 得到的协方差平方根因子为三角方阵, 这不仅严格保证了重构的协方差的对称性与正定性, 还可以利用高效的回代算法 (Back substitution) 求解 SLAM 算法中涉及矩阵逆, 如求解式 (19)、(22) 和式 (27) 中的矩阵逆, 实现进一步提高 SLAM 的求解效率.

3 仿真研究

在 Matlab2010 平台下 (计算机主频 2.6 GHz), 基于悉尼大学野外机器人中心发布的 SLAM 算法仿真器开展仿真研究^[13]. 并通过将提出的算法与

FastSLAM2.0、UFastSLAM 两种方法的综合对比来验证算法性能. 在该部分研究中, 机器人运动学模型为前轮驱动并转向的 Car-like 运动学模型, 环境观测模型为 Bearing-Rang 激光雷达扫描模型. 相关仿真参数设定为: 移动机器人航向运动速度 $v = 4 \text{ m/s}$; 机器人车体前后轮间距 $WB = 4 \text{ m}$; 环境观测传感器最大观测距离范围 40 m , 最大前向视角 180° ; 控制周期 25 ms , 环境路标观测周期 200 ms .

仿真中设计的地图环境为图 1 所示的大尺度、密集路标环境. 其地图尺度为 230×190 , 包含路标 302 个 (图中 “*” 表示实际路标位置, “.” 表示估计路标位置). 仿真中设定移动机器人从初始状态 $\mathbf{s}_0 = [0, 0, 0]^T$ 开始运动, 并按照全局规划的一系列目标点 (图中 “o” 表示) 行走, 直至运动一周后返回起始点 (环形闭合).

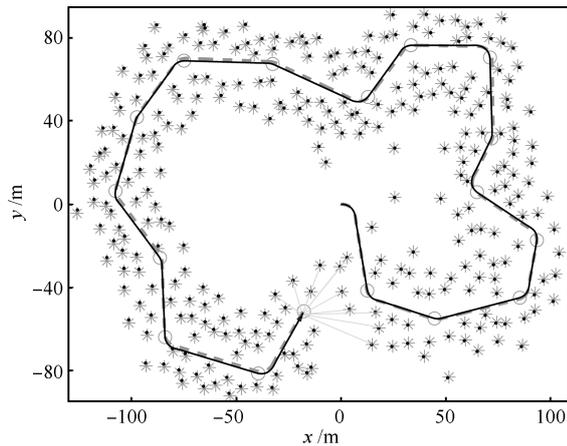


图 1 大尺度、密集路标下仿真环境

Fig. 1 Large map with dense landmarks

3.1 增长测量噪声条件下的 SLAM 算法性能

由于在粒子滤波器的重要性函数设计中缺少了当前的环境观测信息, 经典的 FastSLAM 算法的主要缺点在于粒子集退化问题 (如 FastSLAM1.0). 特别当环境观测传感器精确时, 观测似然函数的外形尖锐, 容易造成绝大多数粒子权重很小甚至为零, 导致粒子集退化、SLAM 滤波器失效. 为验证提出的 SLAM 算法有效地避免了粒子退化问题, 在图 1 所示的特征地图下开展一系列的仿真实验: 仿真中设定粒子数为 10 个, 车辆速度控制噪声设定为 0.3 m/s ; 驾驶角控制噪声设定为 3° ; 环境特征测量的角度噪声为 0.5° ; 环境特征测量的距离噪声为 11 组增长的值, 分别为 0.01 m , 0.03 m , 0.05 m , 0.07 m , 0.09 m , 0.1 m , 0.15 m , 0.2 m , 0.5 m , 0.7 m , 1.0 m . 算法性能评估标准为均方根误差 (Root mean square error, RMSE) 和有效粒子百分比 (Number of effective particles, NEFF), 定义

为

$$\text{RMSE} = \left(\frac{\sum_{k=0}^{N_s} |\mathbf{s}_k - \hat{\mathbf{s}}_k|^2}{N_s - 1} \right)^{\frac{1}{2}}$$

$$\text{NEFF} = \frac{100\%}{N \sum_{i=1}^N (\omega_k^{[i]})^2} \quad (33)$$

式中, \mathbf{s}_k 为机器人在 k 时刻的真实状态; $\hat{\mathbf{s}}_k$ 为 SLAM 算法的估计值; N_s 为一次 SLAM 仿真中的离散采样点数量; $\omega_k^{[i]}$ 为第 i 个粒子的重要性权重; N 为粒子数.

对于每组选定的距离测量噪声水平, 均对 FastSLAM2.0、UFastSLAM 和提出 SLAM 算法执行 50 次仿真实验. 并以 50 次仿真的机器人路径 RMSE 平均值 (评估 SLAM 估计精度) 及标准差 (评估 SLAM 执行稳定性) 和 NEFF 平均值 (及标准差) 作为评价标准, 实验结果如图 2 和图 3 所示. 图 2 为三种算法的 RMSE 均值及标准差对比. 可以看出, 随着测量噪声增长, 三种 SLAM 算法对车辆路径的估计误差及标准差都逐步增加. 然而, 提出 SLAM 算法的 RMSE 均值要低于另外两种算法, 意味着提出方法的估计精度要优于另两者, 同时提出 SLAM 算法在不同观测噪声下的 RMSE 标准差增长速度也低于另外两种算法, 说明其具有较好的运行稳定性. 进一步地, 这三种算法都避免了粒子集退化问题, 当测量噪声较低时 (如 0.01 m , 0.03 m , 0.05 m , 0.07 m , 0.09 m), SLAM 算法并没有出现较高的 RMSE, 其原因在于在这三种算法的重要性函数设计过程中均融入了当前环境观测信息. 即便观测噪声较低、观测似然 $p(\mathbf{z}_k | \mathbf{x}_k)$ 外形尖锐, 经运动

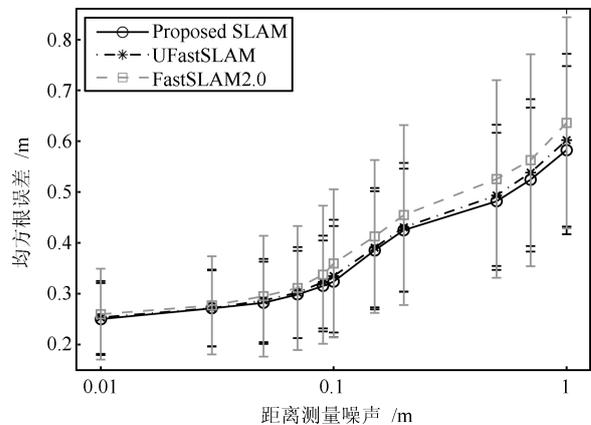


图 2 增长测量噪声条件下的 3 种算法 RMSE

Fig. 2 Performance of robot path estimation with increasing measurement noise levels in range

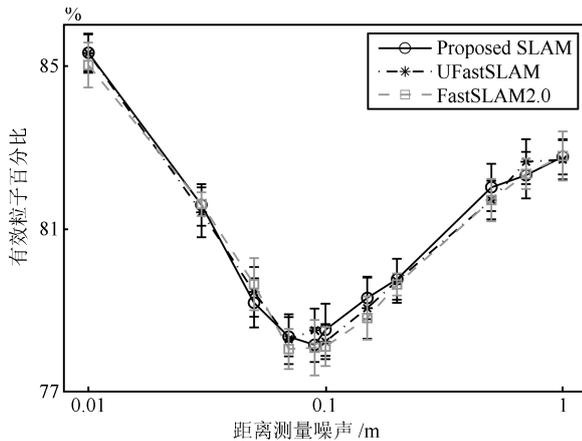


图3 增长测量噪声条件下的3种算法有效粒子百分比
Fig. 3 Percentage of effective particle number with increasing measurement noise levels in range

学预测的粒子集在卡尔曼滤波更新阶段会被吸引到高观测似然区域(式(13)~(20)).这一事实还可以通过图3的有效粒子百分比 NEFF 来验证:对于11组测量噪声,提出SLAM算法的效粒子百分比均高于77%,这意味着仅有低于23%的粒子对于SLAM滤波器是无用的.

3.2 增长粒子数条件下的SLAM算法性能

研究提出算法在增长粒子数条件下的性能.车速控制噪声设定为0.3 m/s;驾驶角控制噪声设定为3°;环境特征测量的角度噪声为0.5°;环境特征测量的距离噪声为0.1 m;设定的6组增长的粒子数分别为1, 5, 10, 20, 50, 100.对于每一组设定的粒子数,均对3种SLAM算法进行50次仿真,并以机器人轨迹RMSE均值及标准差来评价仿真结果,实验结果如图4所示.可以看出,对于1个粒子、5个粒子和10个粒子的情况,提出SLAM算法的估计精度要高于UFastSLAM和FastSLAM2.0,随着粒子数增长,三种算法间的差距逐渐减小,当粒子数高于50时,三种算法的性能已经接近(包括精度和稳定性),且SLAM精度不再随粒子数的增长而明显提高.

在仿真中,通过记录SLAM算法不确定性处理的主要部分(包括粒子状态预测、粒子状态更新、重复观测的环境路标更新以及新路标初始化)的计算机CPU执行时间来对比不同粒子数条件下的SLAM算法计算复杂度.对于每组设定的粒子数,其50次仿真的平均计算复杂度如表3所示.

从表3的结果可以看出,在采用相同粒子数的条件下,提出算法的复杂度介于FastSLAM2.0与UFastSLAM之间.另外,由于提出SLAM算法在粒子数相同的条件下,估计性能上要强于FastSLAM2.0,因此,为实现相同的SLAM估计精度,提

出算法的计算代价仍然要低于FastSLAM2.0.见图4和表3,10个粒子条件下的提出算法SLAM精度要高于20个粒子条件下的FastSLAM2.0算法,而10个粒子条件下提出算法的计算代价为121.42 s,低于20个粒子条件下FastSLAM2.0算法需要的139.83 s.

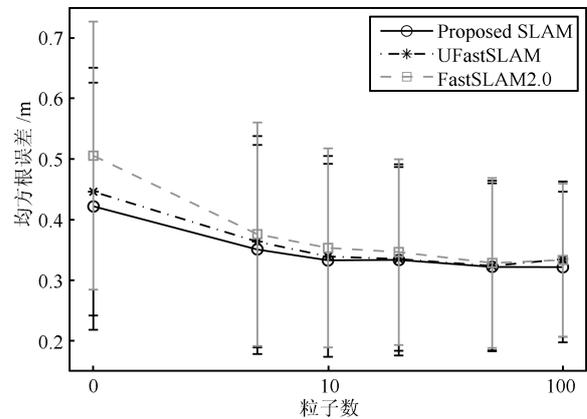


图4 增长粒子数条件下的3种算法性能

Fig. 4 Performance of robot path estimation with increasing particle numbers

表3 不同粒子数条件下SLAM算法的计算复杂度对比
Table 3 Computational cost for SLAM algorithms

SLAM 算法	1 粒子	5 粒子	10 粒子	20 粒子	50 粒子	100 粒子
本文算法	12.70 s	62.12 s	121.42 s	245.84 s	654.36 s	1 187.85 s
UFastSLAM	14.24 s	68.06 s	136.98 s	288.51 s	681.49 s	1 429.92 s
FastSLAM2.0	7.14 s	34.39 s	68.19 s	139.83 s	345.25 s	683.07 s

4 实验研究

基于澳大利亚悉尼维多利亚公园数据集^[14-15]开展SLAM实验研究.该数据集为人工驾驶车辆在悉尼维多利亚公园中所采集,车辆行驶时间约30分钟,运行路径超过4 km.实验车装备有SICK激光传感器、GPS、里程计传感器,并在实验中利用车辆周围的树干作为自然路标.其中,激光传感器用于感知自然路标相对于车辆的距离-角度信息;GPS传感器用于捕获车辆实际运行轨迹;车辆里程计传感器测量车辆速度和驾驶角度,用于执行车辆航迹推算.值得说明的是,该数据集中的GPS定位信息仅用于以评估SLAM算法的效果,并不参与SLAM算法的运算.图5显示了数据集中车辆的GPS轨迹和基于里程计信息的车辆航迹推算轨迹,由于公园建筑物、植被等对车辆的遮挡,GPS传感器仅提供间断的车辆定位信息.同时,由于里程计的测量噪声、车辆运动学模型不精确以及车辆的运行地形不平坦等因素,航迹推算轨迹发散,严重偏离GPS轨迹.

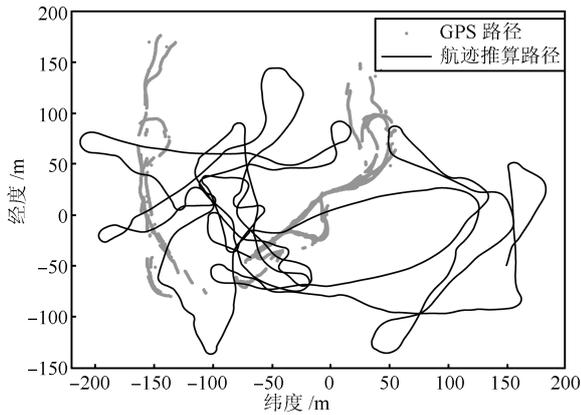


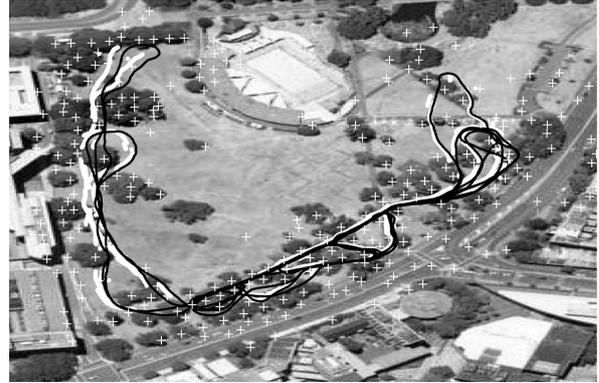
图5 维多利亚数据集的车辆 GPS 路径和里程计路径
Fig.5 GPS and odometry paths of the dataset

基于该数据集, 对提出 SLAM、FastSLAM2.0 以及 UFastSLAM 展开对比研究. 需说明的是, 悉尼维多利亚公园数据集并未提供相关噪声参数的信息^[16]. 因而, 在实验中对三种 SLAM 算法采用了与文献 [6] 相同的噪声参数: 车辆速度控制噪声为 0.8 m/s , 驾驶角控制噪声为 1.8° ; 环境特征测量的角度噪声为 2.8° , 测距噪声为 1.5 m ; 三种 SLAM 滤波器均采用 10 个粒子来描述车辆轨迹和环境地图的联合后验密度. 图 6 为与 Google Earth 配准后不同 SLAM 算法的实验结果 (图中 “+” 表示估计的路标, 黑色连续线为估计的路径, 白色不连续线为 GPS 路径). 图 6 (a) 为 FastSLAM2.0 的结果, 由于累积非线性模型的线性化误差, 在图 6 (a) 中的左半侧区域内, 该算法得到的车辆运行轨迹与 GPS 轨迹发生了明显的不匹配; 图 6 (b) 和图 6 (c) 分别为 UFastSLAM 结果和提出算法的 SLAM 实验结果, 可以看出, 由于累积误差小, 这两种算法得到的车辆轨迹与 GPS 轨迹间的误差均小于 FastSLAM2.0 算法, 且提出 SLAM 算法的轨迹与 GPS 轨迹吻合程度更高, 说明其性能要优于 UFastSLAM 和 FastSLAM2.0.

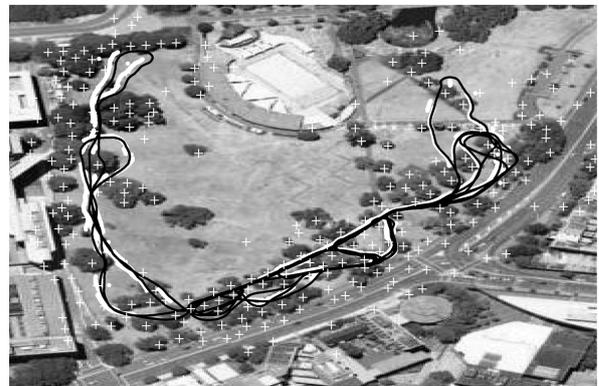
为了定性评估 SLAM 算法的性能, 基于该数据集对 3 种 SLAM 算法分别进行 10 次实验, 并以 10 次实验的 SLAM 轨迹的平均 RMSE 和平均计算时间作为评价标准, 结果见表 4. 其中, 提出算法具有最小 RMSE, 即该算法有最好的 SLAM 估计精度. 在算法执行时间方面, 提出算法的执行时间要小于 UFastSLAM 算法.

除了精度和计算效率方面的优势以外, 提出算法还具有无需计算雅可比矩阵、自由参数少等优点而便于实际 SLAM 应用. 在 UFastSLAM 前期实验中, 按文献 [7] 推荐, 将无迹变换的尺度参数设定为 $\kappa = 3 - n$. 由于实验中的增广车辆状态维数为 $n = 5$, 其中心 Sigma 点具有负权重 ($\omega_0 = -2/3$)

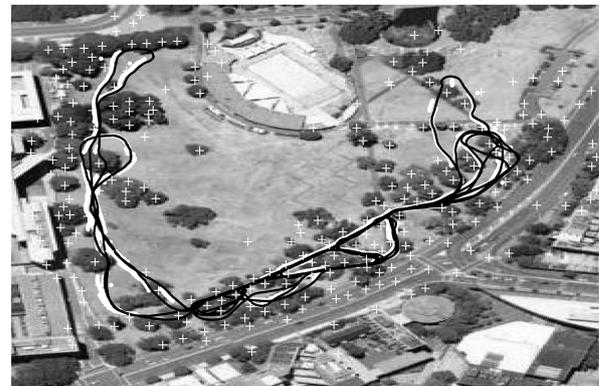
将可能导致求解的协方差矩阵负定, 由于无法计算负定矩阵平方根而导致 UFastSLAM 算法无法执行. 因而, 在实验中将尺度无迹变换引入 UFastSLAM, 经过逐步调整尺度无迹变换中涉及的自由参数, 如 λ, α, β (参见文献 [6–7]), 获得了有意义的 SLAM 实验结果. 与之相比较, 提出算法无需设置这些自由参数, 更方便于实际情况下的特征地图 SLAM 应用.



(a) FastSLAM2.0 算法的 SLAM 结果
(a) Result of FastSLAM2.0 algorithm



(b) UFastSLAM 算法的 SLAM 结果
(b) Result of UFastSLAM algorithm



(c) 提出算法的 SLAM 结果
(c) Result of the proposed SLAM algorithm

图 6 维多利亚数据集的 SLAM 实验结果

Fig.6 Results of the FastSLAM2.0, the UFastSLAM, and the proposed SLAM with Victoria Park dataset

表 4 基于维多利亚数据集的 SLAM 算法性能评估
Table 4 Performance of different SLAM algorithms with Victoria Park dataset

SLAM 算法	车辆路径 RMSE (m)	执行时间 (分钟)
本文算法	3.576	33.51
UFastSLAM	3.636	35.68
FastSLAM2.0	4.077	32.96

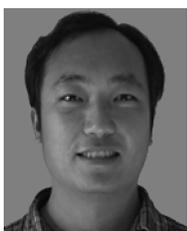
5 结论

面向未知环境中的机器人 SLAM 问题, 基于特征地图描述, 提出平方根容积 FastSLAM 算法. 主要成果总结如下: 1) 利用容积律计算 SLAM 中的高斯权重积分, 达到无需计算雅可比矩阵、减小 SLAM 的线性化累积误差、提高 SLAM 精度的目的; 2) 提出算法中采用协方差因子传播代替传统算法中的协方差传播, 因而避免了计算代价高、数值敏感的协方差矩阵平方根操作; 3) 通过仿真、实验验证了提出算法的性能, 结果表明提出算法是一种精度高、计算代价较小的 SLAM 算法.

References

- Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I. *IEEE Robotics and Automation Magazine*, 2006, **13**(2): 99–110
- Bailey T, Durrant-Whyte H. Simultaneous localization and mapping: part II. *IEEE Robotics and Automation Magazine*, 2006, **13**(3): 108–117
- Montemerlo M. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association [Ph. D. dissertation], Carnegie Mellon University, Pennsylvania, 2003
- 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
- Sim R, Elinas P, Little J J. A study of the Rao-Blackwellised particle filter for efficient and accurate vision-based SLAM. *International Journal of Computer Vision*, 2007, **74**(3): 303–318
- Kim C, Sakhivel R, Chung W K. Unscented FastSLAM: a robust and efficient solution to the SLAM problem. *IEEE Transactions on Robotics*, 2008, **24**(4): 808–820
- Julier S J, Uhlmann J K. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 2004, **92**(3): 401–422
- Zhu Ji-Hua, Zheng Nan-Ning, Yuan Ze-Jian, Zhang Qiang. A SLAM algorithm based on central difference particle filter. *Acta Automatica Sinica*, 2010, **36**(2): 249–257 (祝继华, 郑南宁, 袁泽剑, 张强. 基于中心差分粒子滤波的 SLAM 算法. 自动化学报, 2010, **36**(2): 249–257)
- Kim C, Kim H K, Chung W K. Exactly Rao-Blackwellized unscented particle filters for SLAM. In: *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE, 2011. 3589–3594
- Song Y, Li Q L, Kang Y F, Song Y D. CFastSLAM: a new jacobian free solution to SLAM problem. In: *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*. Saint Paul, USA: IEEE, 2012. 3063–3068
- Arasaratnam I, Haykin S. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 2009, **54**(6): 1254–1269
- van der Merwe R, Wan E A. The square-root unscented kalman filter for state and parameter-estimation. In: *Proceedings of the 2001 IEEE International Conference on Acoustic, Speech and Signal Processing*. Salt Lake City, USA: IEEE, 2001. 3461–3464
- Bailey T. SLAM simulations [Online], available: <http://www.personal.acfr.usyd.edu.au/tbailey/software/index.html>, September 2, 2012
- Nebot E. Victoria park dataset [Online], available: <http://www.personal.acfr.usyd.edu.au/nebot/dataset.htm>, September 2, 2012
- Guivant J E, Nebot E M. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 2001, **17**(3): 242–257

- 16 Huang S D, Wang Z, Dissanayake G. Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 2008, **24**(5): 1121–1130



宋宇 北京交通大学电子信息工程学院讲师. 2007 年在哈尔滨工业大学机器人技术与系统国家重点实验室获得博士学位. 主要研究方向为智能机器人技术, 计算机视觉. 本文通信作者.

E-mail: songyu@bjtu.edu.cn

(**SONG Yu** Lecturer at the School of Electronic and Information Engineering, Beijing Jiaotong University. He received his Ph.D. degree from the State Key Laboratory of Robotics and System, Harbin Institute of Technology in 2007. His research interest covers intelligent robotics and computer vision. Corresponding author of this paper.)



李庆玲 中国矿业大学(北京)机电与信息工程学院讲师. 2009 年在哈尔滨工业大学机器人技术与系统国家重点实验室获博士学位. 主要研究方向为康复机器人系统. E-mail: qingling.li@ia.ac.cn

(**LI Qing-Ling** Lecturer at the School of Mechanical Electronic and Information Engineering, China University of Mining and Technology. She received her Ph.D. degree from the State Key Laboratory of Robotics and Sys-

tem, Harbin Institute of Technology in 2009. Her research interest is rehabilitation robot system.)



康轶非 北京交通大学电子信息工程学院博士研究生. 2010 年获得北京交通大学电子信息工程学院自动化系学士学位. 主要研究方向为机器人定位.

E-mail: ccyclonel@gmail.com

(**KANG Yi-Fei** Ph.D. candidate at the School of Electronic and Information Engineering, Beijing Jiaotong University. He received his bachelor degree from Beijing Jiaotong University in 2010. His main research interest is robot localization.)



闫德立 北京交通大学电子信息工程学院博士研究生. 2010 年获得石家庄铁道大学自动化系硕士学位. 主要研究方向为移动机器人 SLAM.

E-mail: 11111044@bjtu.edu.cn

(**YAN De-Li** Ph.D. candidate at the School of Electronic and Information Engineering, Beijing Jiaotong University. He received his master degree from Shijiazhuang Tiedao University in 2010. His main research interest is mobile robot SLAM.)