

# 基于分治法的同步定位与环境采样地图创建

孙荣川<sup>1,2</sup> 马书根<sup>1,3</sup> 李斌<sup>1</sup> 王明辉<sup>1</sup> 王越超<sup>1</sup>

**摘要** 在不使用几何参数描述大规模环境的前提下, 提出了基于分治法的同步定位与环境采样地图创建 (Simultaneous localization and sampled environment mapping, SLASEM) 算法来同时进行定位与地图创建. 该算法采用环境采样地图 (Sampled environment map, SEM) 描述环境, 使算法不局限于用几何参数描述的规则环境. 同时该算法实时地创建局部地图, 并基于分治法合并局部地图, 保证了算法的实时性. 在合并两个子地图时, 算法首先从环境采样地图中提取出角点, 利用角点约束初步更新子地图; 然后利用符号正交距离函数作为虚拟测量函数, 再次细微地更新子地图; 最后将两个子地图合并到一个大地图, 约简冗余的环境采样粒子, 以提高地图的紧凑性. 两个实验的结果验证了所提算法的有效性和实时性.

**关键词** 同步定位与地图创建, 移动机器人, 分块地图, 导航, 卡尔曼滤波器

**DOI** 10.3724/SP.J.1004.2010.01697

## Simultaneous Localization and Sampled Environment Mapping Based on a Divide-and-conquer Ideology

SUN Rong-Chuan<sup>1,2</sup> MA Shu-Gen<sup>1,3</sup> LI Bin<sup>1</sup> WANG Ming-Hui<sup>1</sup> WANG Yue-Chao<sup>1</sup>

**Abstract** This paper presents an algorithm of simultaneous localization and sampled environment mapping (SLASEM) with a divide-and-conquer ideology to localize a robot and map large scale environments without using the environments' geometric parameters. The usage of sampled environment map (SEM) prevents the algorithm from being limited to structured environments which can be described by geometric parameters. The algorithm builds local maps in real-time firstly, then combines them by the means of divide and conquer. This enables the proposed algorithm to be an on-line algorithm. To combine two local maps, firstly the algorithm extracts corner points from the maps and uses them to update the maps. Then, the algorithm takes the signed orthogonal distance function as the virtual measurement function to update the local maps in detail. Finally, the two local maps are combined into one and the redundant environment samples are removed to make the map compact. The results of two real experiments validate the efficiency and the real-time capability of the proposed algorithm.

**Key words** Simultaneous localization and mapping (SLAM), mobile robot, sub-map, navigation, Kalman filter

移动机器人同步定位与地图创建 (Simultaneous localization and mapping, SLAM) 问题是指机器人探索一个未知的环境, 利用其搭载的传感器进行定位, 同时创建关于周围环境的地图<sup>[1-2]</sup>. 它在自主式移动机器人导航中有着极其重要的地位, 尤其在 GPS 等绝对定位信息无法获取的情况下, SLAM 的作用显得更为明显.

在过去的 20 多年中, 学者们提出了 SLAM 问题的解决办法<sup>[3-8]</sup>. 其中, 基于扩展卡尔曼滤波器 (Extended Kalman filter, EKF) 的 SLAM 算

法是非常出名的一种, 在线性运动学模型和线性观测模型以及高斯白噪声的前提下, Dissanayake 等提出了关于 EKF-SLAM 收敛性的 3 条重要定理<sup>[9]</sup>. 针对 EKF-SLAM 由于非线性模型引起的不一致性 (Inconsistency) 现象, Montemerlo 等提出了 FastSLAM 1.0 和 FastSLAM 2.0. 这两种方法采用 Rao-Blackwellized 粒子滤波器 (Particle filter), 很好地解决了不一致性问题<sup>[10-11]</sup>. 在早期的 SLAM 研究中, 地图主要采用一系列的关于环境的几何参数来描述环境中障碍物轮廓的位置和形状, 例如公园里的树<sup>[12]</sup>, 办公室环境里的墙和墙角<sup>[13]</sup>. 但是, 移动机器人的作业环境通常比较复杂, 环境中的障碍物轮廓不能够用确定的几何参数描述, 或者, 事先无法确定该用什么参数来描述. 因此, 研究能够应用在复杂环境中并且不使用几何参数的 SLAM 显得非常重要.

目前只有少数 SLAM 算法能够应用在复杂环境中, 并且不需要确切的几何参数. 这些算法可以分为以下两类: 1) 利用 Rao-Blackwellized 粒子滤

收稿日期 2010-02-08 录用日期 2010-07-20  
Manuscript received February 8, 2010; accepted July 20, 2010  
国家高技术研究发展计划 (863 计划) (2007AA041502-5) 资助  
Supported by National High Technology Research and Development Program of China (863 Program) (2007AA041502-5)  
1. 中国科学院沈阳自动化研究所机器人学国家重点实验室 沈阳 110016 中国 2. 中国科学院研究生院 北京 100039 中国 3. 日本立命馆大学机器人系 草津 525-8577 日本  
1. State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, P. R. China 2. Graduate University, Chinese Academy of Sciences, Beijing 100039, P. R. China 3. Department of Robotics, Ritsumeikan University, Kusatsu-Shi 525-8577, Japan

波器估计一组机器人的轨迹, 对于每一个机器人轨迹再创建相应的栅格地图<sup>[14-15]</sup>. 另外一种类似的方法是 DP-SLAM. 区别是该方法采用的是分布式粒子 (Distributed particle, DP) 地图, 而不是栅格地图<sup>[16]</sup>. 2) 先实现一个基于特征地图的 SLAM 算法, 然后基于其中的一个特征或者几个特征创建能够描述复杂环境的局部地图. 例如, Scan-SLAM 首先创建一个特征地图, 然后基于地图中的每一个特征创建一个高斯和 (Sum of Gaussian, SoG) 地图<sup>[17]</sup>. 另外一个例子是 DenseSLAM. 该方法首先利用部分传感器数据运行 EKF-SLAM 算法, 从而得到一个特征地图. 之后, 再将剩下的传感器数据绑定到特征上去<sup>[18]</sup>. 以上提到的方法, 复杂地图都是在机器人位姿已知或者特征地图已知的基础上创建的. 它们没有能够在一个滤波器的更新步骤中同时估计机器人位姿和创建复杂地图.

在文献 [19] 中, 作者提出了同步定位与环境采样地图创建 (Simultaneous localization and sampled environment mapping, SLASEM) 算法. 该方法使用环境采样地图 (Sampled environment map, SEM) 来描述复杂环境, 并用 EKF 同步地估计机器人位姿和地图. 相对于已有的能够应用在复杂环境中的算法, SLASEM 最大的特点是机器人位姿和复杂地图是在一个滤波器循环中同时进行估计与更新的, 因此具有更高的精度. SLASEM 采用 EKF 来估计机器人位姿和地图, 其单步执行的复杂度为  $O(N^2)$ ,  $N$  是系统状态向量中环境采样粒子的个数. 随着时间的推移,  $N$  越来越大, 导致算法执行速度越来越慢, 以致不能实时运行. 本文采用分治法来降低 SLASEM 的复杂度, 该算法实时地创建一系列的子地图. 利用创建子地图的空余时间合并先前创建的子地图, 从而能使算法单步执行的复杂度降低为  $O(N)$ . 在不影响精度的情况下, 大大提高算法的运行速度.

本文将首先简要介绍 SLASEM 以及基于分治法的 SLAM 算法, 然后提出基于分治法的 SLASEM 算法, 给出子地图融合的算法, 并用实验验证算法的有效性, 最后给出结论.

## 1 同步定位与环境采样地图创建

### 1.1 环境采样地图 (Sampled environment map, SEM)

Leal 首先提出用环境采样地图来描述一个复杂环境<sup>[20]</sup>. 该方法用一系列离散的点集合来刻画环境的轮廓. 这些离散的点集合就称为环境采样粒子, 而由这些粒子所形成的地图就称为环境采样地图. 图 1 给出了一个环境采样地图的例子.

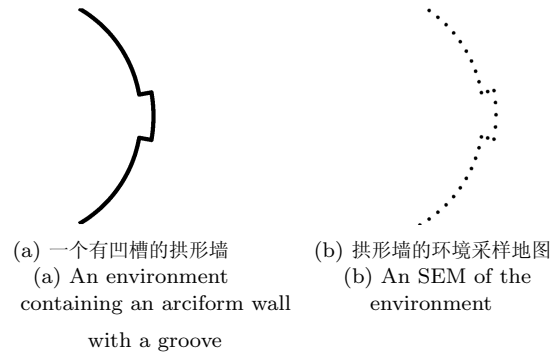


图 1 环境采样地图示例

Fig. 1 An example of SEM

SEM 在描述环境时不需要使用环境模型的几何参数, 且能描述任意形状的环境. 另外, 相对于另外一个传统的地图描述方式 — 栅格地图, 它能够更方便地增加或降低某个特定区域的精度. 在 SEM 中, 增加或降低某一个特定区域的精度只需要相应地增加或者减少对应当于该区域的环境采样粒子的个数. 而对于一个栅格地图来说, 增加或降低某一个区域的精度意味着需要放弃使用简单的数组数据结构, 采用复杂的数据结构, 这使 SLAM 问题中的地图更新变得困难, 所以目前基于栅格地图的 SLAM 算法都是事先设定好一个统一的精度, 并且在整个算法执行过程中不再修改.

### 1.2 同步定位与环境采样地图创建

在文献 [20] 中, 创建 SEM 需要事先知道机器人位姿. 在 SLAM 问题中, 机器人位姿并不能事先给定, 相反, 需要和地图同时估计. 因此, 文献 [20] 中创建并更新 SEM 的方法不适用于 SLAM 领域. 另外, 由于环境采样粒子只是分布在环境轮廓上的一些随机的点, 我们只知道这些粒子分布在某个轮廓上面, 而不知道这些粒子究竟对应于该轮廓的哪个位置. 因此, 传统上创建并更新特征地图的方法不适用于 SEM. 在文献 [19] 中, 作者提出了 SLASEM 算法, 该算法能够用于创建和更新 SEM. 下面简要介绍该算法.

#### 1.2.1 运动学模型和测量模型

在离散时间  $k$  时刻, 假定系统的状态变量为  $\mathbf{X}(k)$ :

$$\mathbf{X}(k) = [\mathbf{X}_r^T(k) \quad \mathbf{X}_{s1}^T \quad \cdots \quad \mathbf{X}_{sN}^T]^T \quad (1)$$

其中,  $\mathbf{X}_r(k)$  为时刻  $k$  机器人的位姿,  $\mathbf{X}_{si}$  ( $i = 1, \dots, N$ ) 是第  $i$  个环境采样粒子的位置,  $N$  为 SEM 中环境采样粒子的个数.

从离散时刻  $k$  到  $k+1$ , 系统的离散运动学模型为

$$\mathbf{X}(k+1) = f(\mathbf{X}(k), \mathbf{u}(k), \mathbf{w}_k) \quad (2)$$

式中,  $\mathbf{u}(k)$  为  $(k, k+1]$  时间段的控制输入量,  $\mathbf{w}_k$  是均值为零方差为  $Q_k$  的高斯白噪声, 即  $\mathbf{w}_k \sim N(\mathbf{0}, Q_k)$ .

由于 SEM 中的粒子与环境不存在一一对应关系, 所以原始的测量数据不能被当作环境采样粒子的直接观测. 文献 [19] 中, 作者提出采用符号正交距离函数作为新的观测函数. 对于一个隐函数多项式 (Implicit polynomial, IP) 曲线  $p(x, y) = 0$ , 环境采样粒子  $[x \ y]^T$  到该曲线的符号正交距离定义如下:

$$g(x, y) = \frac{p(x, y)}{\|\nabla p_{xy}\|} \quad (3)$$

式中,  $\nabla p_{xy}$  为  $p(x, y)$  关于  $x$  和  $y$  的梯度.

假设在离散时刻  $k+1$ , 机器人获得传感器数据  $\mathbf{z}(k+1)$ . 将这些数据点按照轮廓的平滑度分割成形状比较简单的集合, 然后用低阶 IP 拟合这些数据点. 对于被观测到的环境采样粒子, 用该粒子到与其对应 IP 的符号正交距离作为测量方程. 对于所有被观测到的环境采样粒子, 记其总的测量方程为

$$h(\mathbf{X}(k+1), \mathbf{V}_{k+1}^d) = \begin{bmatrix} g(x_{i_1}^p, y_{i_1}^p) + v_1^d \\ g(x_{i_2}^p, y_{i_2}^p) + v_2^d \\ \vdots \\ g(x_{i_m}^p, y_{i_m}^p) + v_m^d \end{bmatrix} \quad (4)$$

式中,  $\mathbf{V}_{k+1}^d = [v_1^d \ \cdots \ v_m^d]^T$  为测量噪声,  $(x_{i_k}^p, y_{i_k}^p)$ ,  $k = 1, \dots, m$  为第  $i_k$  个环境采样粒子在以机器人位姿为坐标原点的坐标系中的坐标.

### 1.2.2 基于 EKF 的预测和更新

基于 EKF 的 SLASEM 算法估计机器人位姿和环境采样地图一般分为以下两个循环递推的过程:

1) 预测. 根据  $k$  时刻系统状态变量  $\mathbf{X}(k)$  的估计值  $\hat{\mathbf{X}}(k|k)$  和控制输入  $\mathbf{u}(k)$ , 利用系统的运动学模型, 预测  $k+1$  时刻的状态值<sup>1</sup>:

$$\hat{\mathbf{X}}(k+1|k) = f(\hat{\mathbf{X}}(k|k), \mathbf{u}(k), \mathbf{0}) \quad (5)$$

$$P(k+1|k) = FP(k|k)F^T + GQ_kG^T \quad (6)$$

其中,  $F = \frac{\partial f}{\partial \mathbf{X}} \Big|_{\hat{\mathbf{X}}(k|k)}$  和  $G = \frac{\partial f}{\partial \mathbf{u}} \Big|_{\mathbf{u}(k)}$  分别是运动学模型 (2) 关于状态变量  $\mathbf{X}(k)$  和控制输入  $\mathbf{u}(k)$  的雅可比矩阵.

2) 更新. 利用  $k+1$  时刻机器人的测量信息和测量方程, 修正系统状态变量:

$$\begin{aligned} \hat{\mathbf{X}}(k+1|k+1) &= \hat{\mathbf{X}}(k+1|k) + \\ &W(k+1)\boldsymbol{\nu}(k+1) \end{aligned} \quad (7)$$

$$\begin{aligned} P(k+1|k+1) &= P(k+1|k) - \\ &W(k+1)S(k+1)W^T(k+1) \end{aligned} \quad (8)$$

其中,

$$\boldsymbol{\nu}(k+1) = -h(\hat{\mathbf{X}}(k+1|k), \mathbf{0}) \quad (9)$$

$$W(k+1) = P(k+1|k)H^T S^{-1}(k+1) \quad (10)$$

$$S(k+1) = HP(k+1|k)H^T + R_{k+1}^d \quad (11)$$

$H = \frac{\partial h}{\partial \mathbf{X}} \Big|_{\hat{\mathbf{X}}(k+1|k)}$  是系统测量方程 (4) 关于系统状态变量的雅可比矩阵.

## 2 基于分治法的 SLAM

传统的基于特征地图的 SLAM 算法采用 EKF 融合传感器数据, 其复杂度为  $O(N^2)$ . 在算法执行过程中, 状态的维数随着特征的增多而增大, 从而导致算法的执行速度逐渐降低, 甚至使算法不能实时运行. 为了提高算法的执行速度, 使算法能够实时运行, Paz 等提出了基于分治法的 SLAM 算法<sup>[21]</sup>. 所谓分治法就是将一个大问题分成多个小问题, 然后逐步地合并小问题的结果, 并最终得到原来大问题的结果. 在 Paz 等提出的方法中, 为了要创建一个全局的大地图, 首先创建一系列的小地图, 然后将这些小地图逐步地由小到大进行合并, 最终得到全局的大地图. 这样做的好处是能将 SLAM 算法的单步执行时间的复杂度降低为  $O(N)$ .

基于分治法的 SLAM 最关键的步骤是如何融合两个子地图. Paz 等的方法是采用一种虚拟观测函数将两个子地图中共同的路标更新到相同的位置, 然后再删除其中一项. 假设两个要融合的子地图  $M_1$  和  $M_2$  中的状态变量分别是  $\mathbf{X}^{M_1}$  和  $\mathbf{X}^{M_2}$ , 对它们共同的特征建立虚拟观测方程:

$$h(\mathbf{X}^{M_1}, \mathbf{X}^{M_2}) = \begin{pmatrix} h_{f_1, g_1} \\ \vdots \\ h_{f_r, g_r} \end{pmatrix} = 0 \quad (12)$$

其中,  $\{f_i, g_i\}$ ,  $i = 1, 2, \dots, r$  分别是子地图  $M_1$  和  $M_2$  中共同的特征.  $h_{f_i, g_i}$  是虚拟观测函数:

$$h_{f_i, g_i} = \mathbf{X}_{f_i} - \mathbf{X}_{M_1, M_2} \oplus \mathbf{X}_{g_i} \quad (13)$$

其中,  $\mathbf{X}_{f_i}$  和  $\mathbf{X}_{g_i}$  分别是特征  $f_i$  和  $g_i$  在  $M_1$  和  $M_2$  中的坐标.  $\mathbf{X}_{M_1, M_2}$  是  $M_2$  的坐标系到  $M_1$  的坐标系的变换.  $\mathbf{X}_{M_1, M_2} \oplus \mathbf{X}_{g_i}$  表示将  $M_2$  中的特征  $g_i$  转换到  $M_1$  的坐标系中.

融合两个子地图时, 将虚拟的测量值和方差均置为零, 然后利用 EKF 进行状态更新. 将虚拟测量值的方差设置为零可以保证在 EKF 更新之后  $f_i, i$

<sup>1</sup>本文中,  $\mathbf{0}$ ,  $\mathbf{1}$  和  $I$  分别表示维数相容的  $\mathbf{0}$  向量,  $\mathbf{1}$  向量和单位对角矩阵.

$= 1, 2, \dots, r$  能够与  $g_i, i = 1, 2, \dots, r$  完全相同, 从而可以去除其中的任意一个.

### 3 基于分治法的 SLASEM

在不使用几何参数的情况下, 为了同步地进行定位, 并创建大规模环境的地图, 本文提出了基于分治法的同步定位与环境采样地图创建算法. 该算法采用分治法思想, 将创建一个全局地图的问题分解为创建一系列子地图以及合并这些子地图的问题. 由于子地图的维数比较小, 本算法能够实时地创建并更新子地图. 同时, 通过平摊合并子地图的时间, 本算法的单步执行的复杂度降低为  $O(N)$ . 由于算法复杂度的分析与文献 [21] 一样, 本文将针对 SEM 的特殊性重点介绍在已知分块地图的基础上, 如何合并两个分块地图, 并用实验来验证算法的有效性和实时性.

文献 [21] 中的方法应用于基于特征地图的 SLAM 算法, 地图中的特征与环境具有一一对应性; 而在 SLASEM 中, 地图中的环境采样粒子与环境不具有有一一对应性, 从而不能建立两个子地图中状态变量之间的虚拟观测方程, 无法直接使用文献 [21] 中的地图融合方式. 针对这种情况, 我们首先从两个子地图中提取出角点, 利用角点之间的一一对应性初步地更新子地图; 然后, 我们采用符号正交距离函数作为两个子地图中环境采样粒子的一个间接的虚拟观测函数来进一步更新子地图, 从而最终将两个子地图合并; 最后, 我们给出粒子约简方法解决融合之后的地图出现的局部区域环境采样粒子过于拥挤的问题.

假设要融合的两个子地图分别是  $M_1$  和  $M_2$ , 其状态变量的估计值和方差矩阵分别是  $\langle \hat{\mathbf{X}}_{M_1}, C_{M_1} \rangle$

和  $\langle \hat{\mathbf{X}}_{M_2}, C_{M_2} \rangle$ :

$$\hat{\mathbf{X}}_{M_1} = \begin{bmatrix} \hat{\mathbf{X}}_r^{M_1} \\ \hat{\mathbf{X}}_1^{M_1} \\ \vdots \\ \hat{\mathbf{X}}_{n_1}^{M_1} \end{bmatrix}, \quad \hat{\mathbf{X}}_{M_2} = \begin{bmatrix} \hat{\mathbf{X}}_r^{M_2} \\ \hat{\mathbf{X}}_1^{M_2} \\ \vdots \\ \hat{\mathbf{X}}_{n_2}^{M_2} \end{bmatrix} \quad (14)$$

将  $\hat{\mathbf{X}}_{M_1}$  和  $\hat{\mathbf{X}}_{M_2}$  组合成一个状态向量  $\hat{\mathbf{X}}_{\text{aug}}$ :

$$\hat{\mathbf{X}}_{\text{aug}} = \begin{bmatrix} \hat{\mathbf{X}}_{M_1}^T & \hat{\mathbf{X}}_{M_2}^T \end{bmatrix}^T \quad (15)$$

$$C_{\text{aug}} = \begin{bmatrix} C_{M_1} & \emptyset \\ \emptyset & C_{M_2} \end{bmatrix} \quad (16)$$

#### 3.1 角点约束更新

角点是指相邻的两个墙壁的交点, 如图 2(a) 和 2(b) 中的五角星和三角. 虽然环境采样粒子之间不存在一一对应性, 但是角点之间具有一一对应性. 在提取出角点之后, 我们可以使用第 2 节描述的虚拟观测函数更新角点的位置. 由于角点与其所在的子地图中的环境采样粒子之间存在协方差, 更新角点的同时将会间接地更新环境采样粒子, 从而可以将两个存在较大位置误差的地图调整成位置误差较小的地图.

假设子地图  $M_1$  和  $M_2$  中存在  $l$  个共同的角点.  $M_1$  中的角点和雅可比矩阵记为  $\hat{\mathbf{X}}_c^{M_1}$  和  $J_{M_1}$ ,  $M_2$  中的角点和雅可比矩阵记为  $\hat{\mathbf{X}}_c^{M_2}$  和  $J_c^{M_2}$  (两个点集合的角点和相应的雅可比矩阵的计算见附录). 将这些角点添加到状态  $\hat{\mathbf{X}}_{\text{aug}}$  中:

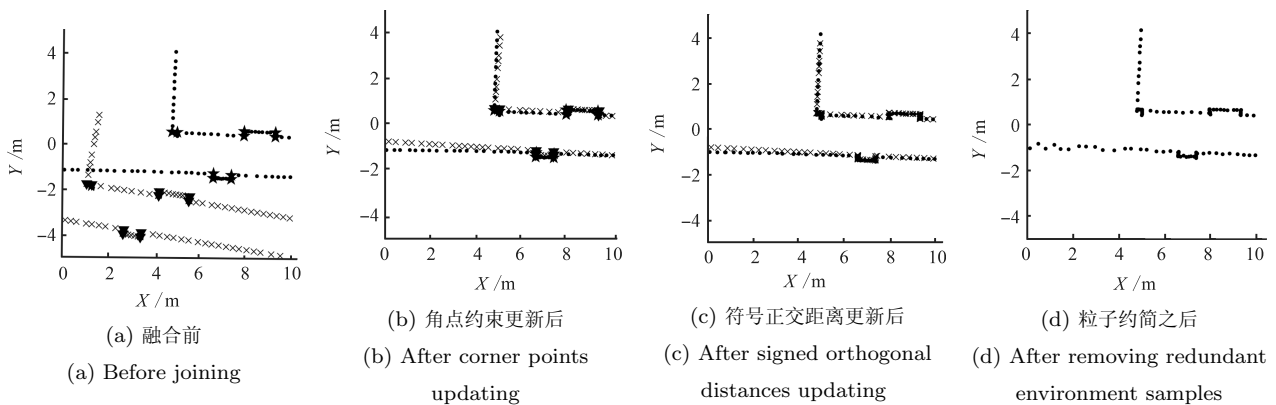


图 2 两个子地图融合过程 (圆点和十字号分别表示两个子地图, 五角星和三角分别是这两个子地图中的角点)

Fig. 2 The process of joining two maps (Dots and crosses represent two local maps and pentacles, respectively. Pentacles and triangles are the two maps' corner points, respectively.)

$$\hat{\mathbf{X}}_{\text{aug}^*} = \begin{bmatrix} \hat{\mathbf{X}}_{\text{aug}}^T & (\hat{\mathbf{X}}_c^{M_1})^T & (\hat{\mathbf{X}}_c^{M_2})^T \end{bmatrix}^T \quad (17)$$

$$C_{\text{aug}^*} = \begin{bmatrix} C_{\text{aug}} & C_{M_1}^T J_{M_1}^T & C_{M_2}^T J_{M_2}^T \\ J_{M_1} C_{M_1} & J_{M_1} C_{M_1} J_{M_1}^T & 0 \\ J_{M_2} C_{M_2} & 0 & J_{M_2} C_{M_2} J_{M_2}^T \end{bmatrix} \quad (18)$$

利用第 2 节中的方法, 对  $\hat{\mathbf{X}}_{\text{aug}^*}$  中的  $\hat{\mathbf{X}}_c^{M_1}$  和  $\hat{\mathbf{X}}_c^{M_2}$  用相应的虚拟观测函数 (12) 进行更新, 从而可以得到新的状态估计  $\hat{\mathbf{X}}_{a^*}$  及方差矩阵  $C_{a^*}$ . 去除里面角点对应的项, 得到:

$$\hat{\mathbf{X}}_{\text{aug}1} = [\tilde{\mathbf{X}}_{M_1}^T \quad \tilde{\mathbf{X}}_{M_2}^T]^T \quad (19)$$

$$C_{\text{aug}1} = \begin{bmatrix} \check{C}_{M_1, M_1} & \check{C}_{M_1, M_2} \\ \check{C}_{M_2, M_1} & \check{C}_{M_2, M_2} \end{bmatrix} \quad (20)$$

对于图 2 (a) 中位置误差比较大的两个子地图, 利用角点约束更新可以减少两者之间的位置误差, 如图 2 (b) 所示.

### 3.2 基于符号正交距离的状态更新

基于符号正交距离的状态更新主要用来在局部区域更新/融合两个子地图, 使两块地图在局部细节上能够相互一致.

假设  $F^{M_1} = \{\mathbf{F}_{i_1}^{M_1}, \dots, \mathbf{F}_{i_l}^{M_1}\}$  和  $G^{M_2} = \{\mathbf{G}_{j_1}^{M_2}, \dots, \mathbf{G}_{j_l}^{M_2}\}$  分别是  $M_1$  和  $M_2$  中对应于相同轮廓的环境采样粒子的集合. 与第 1.2.1 节类似, 对于每一对匹配  $\langle \mathbf{F}_{i_k}^{M_1}, \mathbf{G}_{j_k}^{M_2} \rangle$  ( $k = 1, 2, \dots, l$ ), 首先将  $\mathbf{G}_{j_k}^{M_2}$  转换到  $M_1$  的坐标系下, 即  $\mathbf{F}_{j_k}^{M_2} = \mathbf{X}_{M_1, M_2} \oplus \mathbf{G}_{j_k}^{M_2}$ . 然后利用  $\mathbf{F}_{i_k}^{M_1}$  和  $\mathbf{F}_{j_k}^{M_2}$  拟合一个低阶隐函数多项式  $p_k$ .  $p_k$  是在  $M_1$  的坐标系中的一个多项式, 将它变换到  $M_2$  的坐标系中, 得到另外一个多项式  $q_k$ . 对于点集合  $\mathbf{F}_{i_k}^{M_1}$  和点集合  $\mathbf{G}_{j_k}^{M_2}$ , 将  $p_k$  和  $q_k$  分别看作是它们的间接的虚拟观测方程. 对于  $F^{M_1}$  和  $G^{M_2}$  中其他的粒子集合进行类似操作, 最终得到统一的与式 (4) 类似的观测方程. 区别是在这里观测方程是虚拟的, 其噪声和方差均为零. 利用式 (7) 和 (8) 对  $\langle \hat{\mathbf{X}}_{\text{aug}1}, C_{\text{aug}1} \rangle$  进行更新, 得到  $\langle \hat{\mathbf{X}}_{\text{aug}2}, C_{\text{aug}2} \rangle$ :

$$\hat{\mathbf{X}}_{\text{aug}2} = [\tilde{\mathbf{X}}_{M_1}^T \quad \tilde{\mathbf{X}}_{M_2}^T]^T \quad (21)$$

$$C_{\text{aug}2} = \begin{bmatrix} \check{C}_{M_1, M_1} & \check{C}_{M_1, M_2} \\ \check{C}_{M_2, M_1} & \check{C}_{M_2, M_2} \end{bmatrix} \quad (22)$$

在经过角点约束更新得到两个位置误差较小的子地图之后, 符号正交距离更新将会从局部上细微地调整这两张地图, 如图 2 (c) 所示.

### 3.3 粒子约简

经过角点约束更新和基于符号正交距离的状态更新之后, 子地图  $M_1$  和  $M_2$  能够较好地重合, 从而  $M_1$  和  $M_2$  可以按如下步骤合并成一个新地图: 1) 将  $M_2$  中的机器人位姿和环境采样粒子转换到  $M_1$  的坐标系下; 2) 将  $M_2$  中的机器人位姿, 以及  $M_1$  和  $M_2$  中的环境采样粒子组合, 从而得到新地图, 其状态向量的估计和方差分别是  $\hat{\mathbf{X}}_{M_1, M_2}$  和  $C_{M_1, M_2}$ , 其中:

$$\hat{\mathbf{X}}_{M_1, M_2} = \begin{bmatrix} \tilde{\mathbf{X}}_r^{M_1} \oplus \tilde{\mathbf{X}}_r^{M_2} \\ \tilde{\mathbf{X}}_1^{M_1} \\ \vdots \\ \tilde{\mathbf{X}}_{n_1}^{M_1} \\ \tilde{\mathbf{X}}_r^{M_1} \oplus \tilde{\mathbf{X}}_1^{M_2} \\ \vdots \\ \tilde{\mathbf{X}}_r^{M_1} \oplus \tilde{\mathbf{X}}_{n_2}^{M_2} \end{bmatrix} \quad (23)$$

由于新地图合并了  $M_1$  和  $M_2$  中所有的环境采样粒子, 所以在被  $M_1$  和  $M_2$  所共同描述的区域里, 粒子将会变得特别拥挤. 这时, 需要对这些拥挤的区域进行约简, 以便使地图更加紧凑.

假设环境采样粒子集合  $H$  在其所描述的区域内发生了拥挤现象, 需要进行约简. 那么, 首先利用  $H$  中的环境采样粒子拟合一个低阶隐函数曲线  $p(x, y) = 0$ . 如果地图中存在多个拥挤区域, 那么对这些区域各自拟合相应的低阶隐函数多项式. 将这些低阶隐函数多项式看作是这些粒子集合各自的一个约束, 利用基于约束的卡尔曼滤波器对状态  $\hat{\mathbf{X}}_{M_1, M_2}$  进行约束更新<sup>[22-23]</sup>. 这样可以将环境采样粒子明确地约束到其所描述的轮廓上面, 从而删除其中任意一个或多个环境采样粒子时, 剩下的环境采样粒子所描述的形状将不会发生太大变化.

之后, 对每一个拥挤的区域采用下面的步骤选择待保留的环境采样粒子:

**步骤 1.** 初始化一个环境采样粒子的列表  $L$ , 该列表包含有该区域的两个端点.

**步骤 2.** 对于剩下的每一个环境采样粒子, 如果它距离列表  $L$  中所有环境采样粒子的最小距离大于一个事先设定的门阈值  $D_0$ , 那么就将其添加到  $L$  中.

**步骤 3.** 检查列表  $L$  中环境采样粒子的个数, 如果大于事先定义的一个数值, 那么结束循环; 否则, 减小  $D_0$ , 再次执行步骤 2.

除去  $L$  中的环境采样粒子, 该区域中剩下的环境采样粒子都需要被删除. 由于  $\hat{\mathbf{X}}_{M_1, M_2}$  和  $C_{M_1, M_2}$  是一个正态分布的均值和方差矩阵, 所以从中删

除冗余的环境采样粒子只需要去除  $\hat{X}_{M_1, M_2}$  和  $C_{M_1, M_2}$  中这些冗余粒子对应的项即可。

#### 4 实验与分析

本节用两个实验来验证基于分治法的 SLASEM 算法的有效性和实时性。实验分别在作者所在机构的 R 楼的 3 楼和 4 楼 (两个比较大的室内环境) 进行。实验采用可变形机器人 Amoeba-I 作为移动平台, 以及激光扫描传感器 URG-04LX 探测环境, 如图 3 所示。

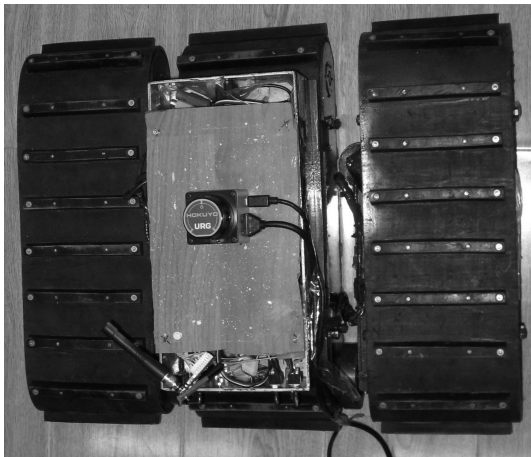


图 3 可变形机器人 Amoeba-I 和激光扫描传感器 URG-04LX

Fig. 3 The shape-shift robot Amoeba-I and the scanning laser range finder URG-04LX

实验过程中数据被保存下来, 然后在一台 CPU 3.60 GHz, 内存 3.25 G 的计算机上作离线分析。算法假设机器人每行走超过 10 cm 或者旋转超过  $10^\circ$  进行一次状态更新。在运行基于分治法的 SLASEM 算法时, 如果当前系统状态变量中的环境采样粒子超过 300 个, 那么就将当前状态变量保存, 并以当前位置为原点, 重新开始创建新的局部地图。

实验 1 在 R 楼 3 楼进行, 机器人进入其中一个房间, 并行走一圈, 出去之后再绕走廊一圈, 整个实验过程中机器人的轨迹一共形成了两个闭环, 行程大约 115 m, 如图 4(a) 所示。图 4(b) 给出了基本的 SLASEM 算法创建的地图, 图 4(c) 和 4(d) 是基于分治法的 SLASEM 创建的子地图以及由这些子地图合并生成的大地图。合并之后的地图与图 4(b) 基本一致, 也与图 4(a) 所示的楼层示意图一致, 这验证了本文所提出的基于分治法的 SLASEM 所创建地图的准确性。同时, 合并之后的机器人轨迹能够形成一个闭环, 与实际实验相符合, 从侧面说明了所提算法在定位上的准确性。图 5(a) 显示了算法单步执行的时间。由于 SLASEM 的单步执行时间

平方于地图中环境采样粒子的个数, 随着时间的推移, SLASEM 单步执行所需的时间呈现抛物线式增长, 最终远远超过数据采集的间隔, 以致无法实时运行; 而采用了分治法的 SLASEM 的单步执行时间只与子地图中环境采样粒子的个数有关, 与全局地图中环境采样粒子的个数不相关, 所以它的单步执行时间基本上都小于数据采集间隔, 并且没有随着时间的推移而出现剧烈的增长, 从而保证了算法能够实时运行。本实验共耗时约 50 min, SLASEM 算法总共需要 183 min, 而基于分治法的 SLASEM 则只需要 13 min, 如图 5(b) 所示。对于基于分治法的 SLASEM 来说, 图 5(a) 中有 4 个时刻的执行时间相对比较长, 这是因为在这 4 个时刻进行了子地图的融合。在移动机器人导航中, 某些情况下只需要机器人当前位置附近的局部地图, 在这类情况下, 我们可以用一个高优先级的线程实时地创建局部地图并定位, 同时用一个低优先级的线程融合子地图, 这样可以把子地图融合的时间平摊到当前时刻与下一次需要使用该融合地图之间的时间段里。从而可以提高算法的执行速度。图 5(c) 显示了经过平摊之后的时间曲线图, 从图上可以看出, 经过平摊之后, 基于分治法的 SLASEM 在原来合并子地图时刻的单步执行时间都低于数据采集间隔, 从而保证了实时性需求。

实验 2 在 R 楼 4 楼进行, 机器人绕走廊行走一圈, 形成了一个闭环, 行程大约 170 m, 耗时 58 min, 如图 6(a) 所示。图 6(b) 是 SLASEM 得到的地图, 图 6(c) 和 6(d) 是基于分治法的 SLASEM 得到的子地图和全局地图。对比图 6(d) 和图 6(a), 我们可以看出基于分治法的 SLASEM 能够得到一个结构正确的全局地图, 并且机器人轨迹也能够成功闭环, 这就验证了所提算法在建图和定位方面的准确性。图 7 显示了算法运行时间的对比, 和实验 1 类似, SLASEM 的单步执行时间平方于地图中环境采样粒子的个数, 随着时间的推移, 执行速度越来越慢, 从而不能实时运行, 最终该算法耗时 420 min; 而基于分治法的 SLASEM 的运行时间只和子地图中环境采样粒子的个数有关, 与全局地图中环境采样粒子的个数无关, 从而能够大大降低算法执行时间, 满足实时性需求, 该算法最终只耗时约 14 min。

#### 5 结论

本文提出了基于分治法的同步定位与环境采样地图创建算法, 并重点介绍了合并两个环境采样地图的算法。该算法首先采用角点约束更新将两个位置误差较大的子地图更新为位置误差较小的地图; 在此基础上采用基于符号正交距离函数的更新方法在局部细节上调整这两个子地图; 最后, 对环境采样

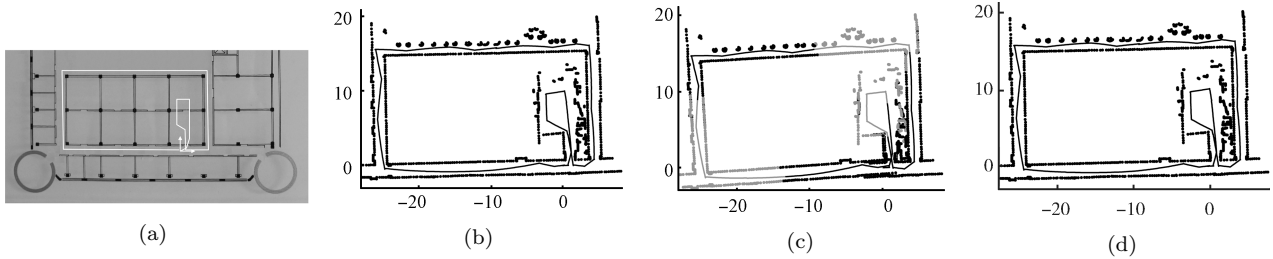


图 4 实验 1 ((a) 楼层示意图, 白色坐标轴显示了地图创建的参考坐标系, 白色曲线为机器人大致的轨迹; (b) 没有采用分治法的 SLASEM 的结果图; (c) 和 (d) 分别为 SLASEM 采用分治法创建的 7 个子地图和最终合并生成的地图. (b) ~ (d) 的横轴和纵轴的计量单位都是 m.)

Fig. 4 The first experiment ((a) The schematic diagram of the environment, in which the white axis is the reference coordinate frame and the white curve describes the robot's trajectory approximately; (b) The map built by SLASEM; (c) and (d) show the local maps and the joined map produced by SLASEM with divide and conquer, respectively. The units of the  $X$  and  $Y$  axes in (b) ~ (d) both are m.)

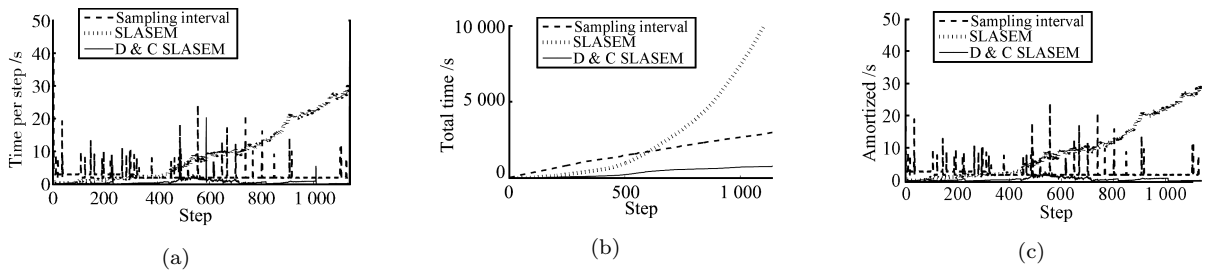


图 5 实验 1 的时间曲线图 ((a) 单步执行的时间曲线; (b) 总共消耗的时间曲线; (c) 平摊之后的单步执行时间曲线)  
Fig. 5 The time curves of the first experiment ((a) The time cost per step; (b) The total cost time; (c) The time cost per step after amortization)

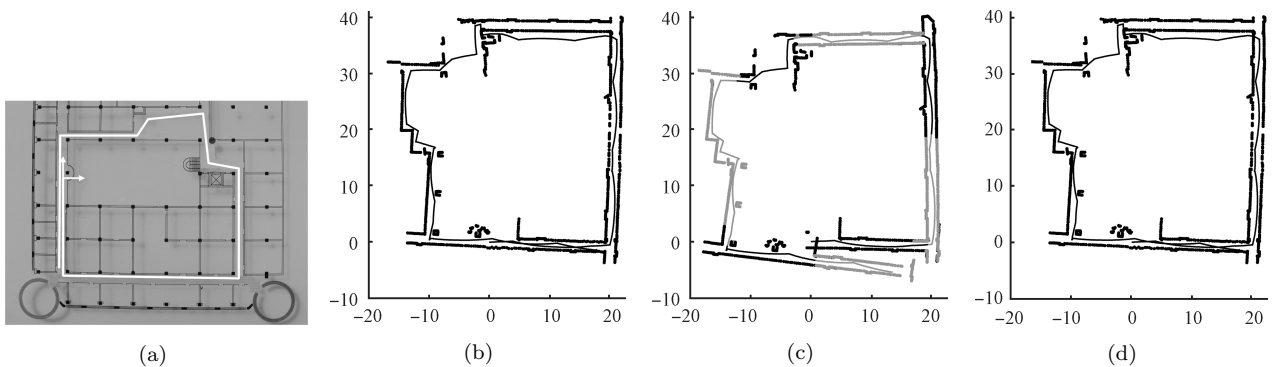


图 6 实验 2 ((a) 楼层示意图, 白色坐标轴显示了地图创建的参考坐标系, 白色曲线为机器人大致的轨迹; (b) 没有采用分治法的 SLASEM 的结果图; (c) 和 (d) 分别为 SLASEM 采用分治法创建的 8 个子地图和最终合并生成的地图. (b) ~ (d) 的横轴和纵轴的计量单位都是 m.)

Fig. 6 The second experiment ((a) The schematic diagram of the environment, in which the white axis is the reference coordinate frame and the white curve describes the robot's trajectory approximately; (b) The map built by SLASEM; (c) and (d) show the local maps and the joined map produced by SLASEM with divide and conquer, respectively. The units of the  $X$  and  $Y$  axes in (b) ~ (d) both are m.)

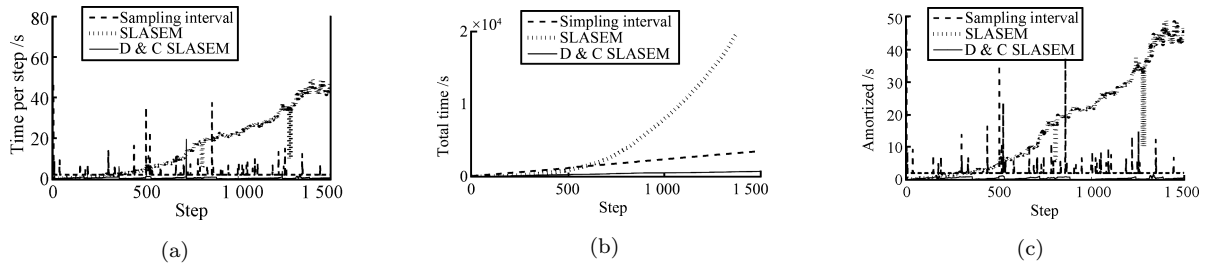


图7 实验2的时间曲线图((a) 单步执行的时间曲线; (b) 总共消耗的时间曲线; (c) 平摊之后的单步执行时间曲线)

Fig. 7 The time curves of the second experiment

((a) The time cost per step; (b) The total cost time; (c) The time cost per step after amortization)

粒子拥挤的区域进行粒子约简, 维持了地图的紧凑性. 两个较大范围的实验验证了所提出算法的有效性和实时性.

## 致谢

作者衷心地感谢郭帅和常健在实验过程中提供的热心帮助.

## 附录 计算两个点集合的角点及雅可比矩阵

假设有两个点集合  $S_1$  和  $S_2$ ,  $\mathbf{x}_1$  和  $\mathbf{y}_1$  分别是  $S_1$  中点的横纵坐标矢量,  $\mathbf{x}_2$  和  $\mathbf{y}_2$  分别是  $S_2$  中点的横纵坐标矢量.

利用最小二乘法求解方程组  $\mathbf{y}_1 = k_1\mathbf{x}_1 + b_1$ , 从而可以得到拟合  $S_1$  的直线  $p_1: y = k_1x + b_1$ , 其中:

$$\begin{bmatrix} k_1 \\ b_1 \end{bmatrix} = ([\mathbf{x}_1 \mathbf{1}]^T [\mathbf{x}_1 \mathbf{1}])^{-1} [\mathbf{x}_1 \mathbf{1}]^T \mathbf{y}_1 \quad (\text{A1})$$

同理, 可以得到拟合  $S_2$  的直线  $p_2: y = k_2x + b_2$ , 其中:

$$\begin{bmatrix} k_2 \\ b_2 \end{bmatrix} = ([\mathbf{x}_2 \mathbf{1}]^T [\mathbf{x}_2 \mathbf{1}])^{-1} [\mathbf{x}_2 \mathbf{1}]^T \mathbf{y}_2 \quad (\text{A2})$$

$S_1$  和  $S_2$  的角点  $\mathbf{X}_c$  可以通过计算直线  $p_1$  和  $p_2$  的交点获得:

$$\mathbf{X}_c = \begin{bmatrix} -k_1 & 1 \\ -k_2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (\text{A3})$$

得到  $S_1$  与  $S_2$  的角点之后, 我们用数值的方法计算  $\mathbf{X}_c$  关于  $\mathbf{x}_1, \mathbf{y}_1$  和  $\mathbf{x}_2, \mathbf{y}_2$  的雅可比矩阵. 例如计算  $\mathbf{X}_c$  关于  $S_1$  中第  $i$  个点横坐标的雅可比矩阵, 首先计算点集合  $\{\mathbf{x}_1 + \delta_x, \mathbf{y}_1\}$  和点集合  $S_2$  的角点  $\mathbf{X}_c^i$ , 其中:

$$\delta = \begin{bmatrix} 0 & 0 & \cdots & d_0 & \cdots & 0 \end{bmatrix}^T \quad \begin{matrix} \uparrow & \uparrow & & \uparrow & & \uparrow \\ 1 & 2 & & i & & n_1 \end{matrix} \quad (\text{A4})$$

$d_0$  是一个预先设定的步长,  $n_1$  是  $S_1$  中点的个数.

$\mathbf{X}_c$  关于  $S_1$  中第  $i$  个点的横坐标的雅可比矩阵  $\mathbf{J}_{x_i}^{S_1}$  可以通过下面的公式计算:

$$\mathbf{J}_{x_i}^{S_1} = \frac{(\mathbf{X}_c^i - \mathbf{X}_c)}{d_0} \quad (\text{A5})$$

$\mathbf{X}_c$  关于  $S_1$  和  $S_2$  中其他点的横纵坐标的雅可比矩阵可以用同样的方法计算得出.

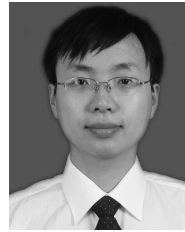
## References

- Durrant-Whyte H F, Bailey T. Simultaneous localization and mapping: part I. *IEEE Robotics and Automation Magazine*, 2006, **13**(2): 99–110
- Bailey T, Durrant-Whyte H F. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics and Automation Magazine*, 2006, **13**(3): 108–117
- Thrun S, Liu Y F, Koller D, Ng A Y, Ghahramani Z, Durrant-Whyte H F. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 2004, **23**(7–8): 693–716
- Frese U. Treemap: an  $O(\log n)$  algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 2006, **21**(2): 103–122
- Kaess M, Dellaert F. Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems*, 2009, **57**(12): 1198–1210
- Wu E Y, Li G Y, Xiang Z Y, Liu J L. Stereo vision based SLAM using Rao-Blackwellised particle filter. *Journal of Zhejiang University – Science A*, 2008, **9**(4): 500–509
- Zhou W, Zhao C X, Guo J H. The study of improving Kalman filters family for nonlinear SLAM. *Journal of Intelligent and Robotic Systems*, 2009, **56**(5): 543–564
- Chen H Y, Sun D, Yang J. Global localization of multirobot formations using ceiling vision SLAM strategy. *Mechatronics*, 2009, **19**(5): 617–628
- 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
- Montemerlo M, Thrun S, Koller D, Wegbreit B. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: *Proceedings of the 18th National Conference on Artificial Intelligence*. Edmonton, Canada: ACM, 2002. 593–598



- 11 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 18th International Joint Conference on Artificial Intelligence. Acapulco, Mexico: Morgan Kaufmann, 2003. 1151–1156
- 12 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
- 13 Martinelli A, Nguyen V, Tomastis N, Siegwart R. A relative map approach to SLAM based on shift and rotation invariants. *Robotics and Autonomous Systems*, 2007, **55**(1): 50–61
- 14 Hahnel D, Burgard W, Fox D, Thrun S. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. New York, USA: IEEE, 2003. 206–211
- 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 Eliazar A, Parr R. DP-SLAM: fast, robust simultaneous localization and mapping without predetermined landmarks. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco, Mexico: Morgan Kaufmann, 2003. 1135–1142
- 17 Nieto J, Bailey T, Nebot E. Scan-SLAM: combining EKF-SLAM and scan correlation. In: Proceedings of the Field and Service Robotics, Results of the 5th International Conference. Port Douglas, Australia: Springer, 2006. 167–178
- 18 Nieto J, Guivant J, Nebot E. DenseSLAM: simultaneous localization and dense mapping. *The International Journal of Robotics Research*, 2006, **25**(8): 711–744
- 19 Sun R C, Ma S G, Li B, Wang Y C. Simultaneous localization and sampled environment mapping. In: Proceedings of the 48th IEEE Conference on Decision and Control and the 28th Chinese Control Conference. Shanghai, China: IEEE, 2009. 6484–6489
- 20 Leal J. Stochastic Environment Representation [Ph.D. dissertation], The University of Sydney, Australia, 2003
- 21 Paz L M, Tardos J D, Neira J. Divide and conquer: EKF SLAM in  $O(n)$ . *IEEE Transactions on Robotics*, 2008, **24**(5): 1107–1120
- 22 Teixeira B O S, Chandrasekar J, Torres L A B, Aguirre L A, Bernstein D S. State estimation for linear and non-linear equality-constrained systems. *International Journal of Control*, 2009, **82**(5): 918–936

- 23 Simon D, Simon D L. Constrained Kalman filtering via density function truncation for turbofan engine health estimation. *International Journal of Systems Science*, 2010, **41**(2): 159–171



**孙荣川** 中国科学院沈阳自动化研究所博士研究生. 主要研究方向为同步定位与地图创建, 离散卡尔曼滤波.  
E-mail: rcsun@sia.cn  
(**SUN Rong-Chuan** Ph.D. candidate at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interest covers SLAM and discrete Kalman filter.)



**马书根** 日本立命馆大学机器人系教授, 中国科学院沈阳自动化研究所研究员. 主要研究方向为仿生机器人、移动机器人和多机器人的协作. 本文通信作者.  
E-mail: shugen@se.ritsumei.ac.jp  
(**MA Shu-Gen** Professor in the Department of Robotics, Ritsumeikan University, Japan. He is also a professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interest covers biomimetic robots, mobile robots, and multi-robot cooperation. Corresponding author of this paper.)



**李斌** 中国科学院沈阳自动化研究所研究员. 主要研究方向为仿生机器人、移动机器人和机器人控制.  
E-mail: libin@sia.cn  
(**LI Bin** Professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interest covers biomimetic robots, mobile robots, and robot control.)



**王明辉** 中国科学院沈阳自动化研究所副研究员. 主要研究方向为移动机器人、机器人控制和多机器人的协作.  
E-mail: mhwang@sia.cn  
(**WANG Ming-Hui** Associate professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interest covers mobile robots, robot control, and multi-robot cooperation.)



**王越超** 中国科学院沈阳自动化研究所研究员. 主要研究方向为机器人学.  
E-mail: yewang@sia.cn  
(**WANG Yue-Chao** Professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His main research interest is robotics.)