A Simultaneous Localization and Mapping Algorithm in Complex Environments: SLASEM

Rongchuan Sun a,b, Shugen Ma a,c,*, Bin Li a, Minghui Wang a and Yuechao Wang a

a State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, P. R. China
b Graduate School of the Chinese Academy of Sciences, Beijing 100039, P. R. China
c Department of Robotics, Ritsumeikan University, Kusatsu-Shi 525-8577, Japan

Received 21 January 2010; accepted 12 May 2010

Abstract
In this paper we present an algorithm for the application of simultaneous localization and mapping in complex environments. Instead of building a grid map or building a feature map with a small number of the obstacles’ geometric parameters, the proposed algorithm builds a sampled environment map (SEM) to represent a complex environment with a set of environment samples. To overcome the lack of one-to-one correspondence between environment samples and raw observations, the signed orthogonal distance function is proposed to be used as the observation model. A method considering geometric constraints is presented to remove redundant environment samples from the SEM. We also present a method to improve the SEM’s topological consistency by using corner constraints. The proposed algorithm has been verified in a simulation and an indoor experiment. The results show that the algorithm can localize the robot and build a complex map effectively.


Keywords
Localization, mapping, SLAM, Kalman filter, mobile robots

1. Introduction
Simultaneous localization and mapping (SLAM) addresses the problem that a robot localizes itself and builds a map simultaneously while it is exploring in an unknown environment [1, 2]. It plays an important role in autonomous navigation of mobile robots, especially in situations where global position information is unavailable.

Over the past two decades, many approaches have been proposed to solve this problem [3–10]. One of the famous approaches is extended Kalman filter (EKF)-
based SLAM. Dissanayake et al. proposed three theorems to prove that the standard EKF-SLAM algorithm is convergent in the case of linear motion and observation models with white Gaussian noise [11]. Montemerlo et al. proposed FastSLAM 1.0 and FastSLAM 2.0, which used a Rao-Blackwellized particle filter instead of an EKF to avoid the problem introduced by nonlinear models and non-Gaussian noises [12, 13]. In the early studies, researchers focused on simple environments, in which a small number of geometric parameters are used to represent the obstacles’ positions and shapes. Examples of these obstacles include trees in a park [14], and corners and walls in office rooms [15]. However, mobile robots usually work in complex environments where obstacles cannot be represented by a number of parameters explicitly or we do not know which geometric parameters can be used to represent the obstacles in advance. Therefore, a SLAM algorithm that can work in unknown complex environments is necessary.

In the literature, many approaches have been proposed to build a map to describe a complex environment with a priori knowledge of a robot’s pose [16]. However, only a few SLAM algorithms build maps to describe complex environments in detail without using explicit geometric parameters. These algorithms fall into two categories. (i) Use a Rao-Blackwellized particle filter to estimate a mobile robot’s trajectory, then build a grid map for each particle based on the known trajectory [17, 18]. A similar algorithm is the DP-SLAM which uses a distributed particle (DP) map instead of a grid map [19]. The difference between a grid map and a DP map is that each single grid is a value denoting the probability of occupancy in the grid map, while it is an ancestry tree in the DP map. (ii) Implement a feature-based SLAM algorithm and attach a local rich map to each feature. An example is the Scan-SLAM approach, which follows the conventional EKF-SLAM to building a landmark map [20]. A landmark description template called Sum of Gaussians (SoG) is attached to each landmark to describe the complex environment in detail. Another example is the DenseSLAM approach. It first uses a part of the sensor data to run a feature-based SLAM that produces a feature map, then the rest of the sensor data are represented with local maps defined relatively to the features’ positions [21]. In all of the approaches mentioned above, the complex maps are built after knowing a robot’s pose or a feature map. They fail to estimate a robot’s pose and build a complex map in one filtering stage.

This paper introduces the simultaneous localization and sampled environment mapping (SLASEM) algorithm — a novel concept of applying SLAM in complex environments. The proposed algorithm uses the sampled environment map (SEM) as the map representation in SLAM to describe a complex environment. In this research, the proposed algorithm is applied in two-dimensional (2-D) complex environments. Moreover, it can be extended to 3-D complex environments. The challenge of adopting the SEM is that the innovation that is needed in an EKF cannot be calculated for the lack of one-to-one correspondence between environment samples in the SEM and raw observations. We solve this problem by using a new observation model — the signed orthogonal distance model. We also im-
prove the performance of the algorithm from the following two aspects: (i) apply the equality-constrained Kalman filter (ECKF) to transform the current state into a new state that satisfies geometric constraints before removing redundant environment samples and (ii) apply corner constraints to all of the environment samples. The first job ensures that removing redundant environment samples does not change contour shapes compared with the ones before removing them. The second job prohibits the existence of the topological inconsistency. Due to the compactness and adaptability of the SEM, the proposed algorithm can describe complex environments efficiently. In addition, the most important feature of this algorithm is that the complex map (i.e., the SEM) is built together with the estimation of the robot’s pose in one filtering stage. This makes the map and the estimation of the robot’s pose optimal from the point view of optimization.

The rest of this paper is organized as follows. Section 2 introduces the SEM. We present a new observation model in Section 3 and propose the SLASEM algorithm in Section 4. Section 5 introduces a method to remove redundant environment samples from the SEM. Section 6 presents an approach to improve the topological consistency of the SEM by using corner constraints. We demonstrate the SLASEM results in both simulation and experiment in Section 7, and give conclusions in Section 8.

2. SEM

Leal et al. introduced the SEM to represent complex environments [22, 23]. This method approximates an environment as a number of discrete and infinitesimal points, referred as environment samples. All of the environment samples form a map called the SEM. Figure 1 shows an example when the environment (an arciform wall with a groove) is approximated by a SEM composed of 29 environment samples. Note that there is no one-to-one correspondence between environment samples and real points on contours. The only known thing is that environment

![Figure 1. (a) Environment containing an arciform wall with a groove. (b) SEM of the environment.](image-url)
Samples should lie on contours they belong to. This information is used as the geometric constraint of environment samples in Section 5.

Similar to the SEM, a map representation called the sample-based map was introduced by Biber and Duckett in long-term SLAM [24]. In that algorithm, a number of sample-based maps are built to describe a complex environment. Each of the sample-based maps describes a local area in the neighborhood of a global reference point. The difference between the sample-based map and the SEM is that a sample-based map is a collection of local 360° scans relative to a global reference point and can only describe an environment locally, while a SEM is a set of points in a global reference frame and can describe a whole environment.

As declared by Leal et al., the SEM is more compact and adaptable than grid maps [23]. The SEM can allocate easily more environment samples in an interesting area and less environment samples in an non-essential area. The area without obstacles is not assigned environment samples. On the contrary, a grid map has to allocate the storage for the entire domain in advance. For a grid map, increasing or decreasing the resolution of an area individually is very challenging. Apart from these advantages, the SEM meets the framework of SLAM well. The SEM is formed by environment samples’ positions. These positions can be stacked into a vector. Therefore, combining the robot’s pose and the SEM into a vector is feasible, and the combined vector can be estimated by using existing filters in the same way as feature-based SLAM algorithms.

In Ref. [23], the SEM was built based on a known robot’s pose and was updated through a sample-based Bayesian data-fusion technique. In the SLAM problem, however, the robot’s pose is not known in advance and needs to be estimated while building a map. Therefore, the method in Ref. [23] to build the SEM is not feasible when solving the SLAM problem. In this paper, an EKF is used to estimate the robot’s pose and the SEM. In the process of adopting the SEM, we overcome the challenge of the lack of one-to-one correspondence between environment samples and observations when calculating the innovation that is needed by an EKF. Since the SEM has been built, the following problem is how to improve its quality (e.g., how to reduce redundant environment samples and how to improve the topological consistency).

In the following sections, we first present an observation model. Then the proposed algorithm (i.e., SLASEM) is explained. Finally, we give a method to eliminate redundant environment samples and a method to improve the SEM’s topological consistency. Figure 2 shows the flow diagram of the algorithm.

3. New Observation Model

To update the system state, an EKF needs to calculate the innovation between the predicted values and the observed ones. This requires the measurements to be explicit functions of the state. When the SEM is adopted in SLAM, raw observations are not functions of the state explicitly because environment samples and raw obser-
vations are not one-to-one corresponding. Thus, the functions of raw observations cannot be used as the observation model in an EKF. In this paper a new observation model is proposed to solve this problem.

Suppose that raw observations, usually collected by a scanning laser range finder, are a set of range-bearing points:

\[
\mathbf{z} = \begin{bmatrix} r_1 & r_2 & \cdots & r_N \\ \theta_1 & \theta_2 & \cdots & \theta_N \end{bmatrix}.
\]

(1)

Transformed into Cartesian coordinate, \( \mathbf{z} \) becomes

\[
\mathbf{z}_{xy} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \end{bmatrix}.
\]

(2)

The observations \( \mathbf{z}_{xy} \) are partitioned into primitive objects (e.g., line segments and arcs) [25]. Then these objects are fitted with implicit polynomials (IPs) of degree \( n \), as:

\[
p_0 + p_{10}x + p_{01}y + p_{20}x^2 + p_{11}xy + p_{02}y^2 + \cdots + p_{n0}x^n + p_{n-1,1}x^{n-1}y + \cdots + p_{0n}y^n = 0.
\]

(3)

An additional constraint of the IP’s coefficients is added to make the fitted polynomial unique:

\[
p_{00}^2 + p_{10}^2 + p_{01}^2 + p_{20}^2 + \cdots + p_{0n}^2 = 1.
\]

(4)

The most advantage of using IPs instead of parametric polynomials is its universality in representing different shapes. For example, a second-order IP can represent ellipses, hyperbolas and parabolas. For indoor environments, we find that first- and second-order IPs are capable of fitting most types of primitive objects partitioned from raw observations. If an object cannot be fitted well with a first- or second-order IP, higher-order IPs are adopted. In this way each of the objects is fitted with an IP well.

Fitting a set of points with a first-order IP can be solved by minimizing the mean square error (MSE) produced by \( y = kx + b \) or \( x = ly + c \). Considering the numerical singularity, the points are fitted with both of the two forms at the same time in
this research. A better one of them is selected to be transformed into the standard IP model (3).

Fitting a set of points with a second- or higher-order IP can be solved by using the 3L-Fitting method efficiently [26].

Since the IPs of all the objects are achieved, they are used to generate the signed orthogonal distance function.

Orthogonal distance is the minimum Euclidean distance from the point \([x\ y]^T\) to the curve represented by \(p\) and is calculated as [27]:
\[
d_{\text{oth}0}(x, y) = |p(x, y)|/\|\nabla p_{xy}\|.
\] (5)

The Jacobian matrix of \(d_{\text{oth}0}(x, y)\) is not continuous in all of the defined intervals. Thus, the signed orthogonal distance is adopted in this study instead of the absolute one:
\[
d_{\text{oth}}(x, y) = p(x, y)/\|\nabla p_{xy}\|.
\] (6)

The signed orthogonal distance function can be used as the observation model in the proposed algorithm. Environment samples will converge to contours if their signed orthogonal distances approach zero. Apart from the signed orthogonal distance function, there may be other functions that can be employed as the observation model. For the generality of using other functions, a unified form \(g(x, y)\) is used to represent the observation model in the rest of the paper:
\[
d = g(x, y).
\] (7)

4. Our Algorithm — SLASEM

The basic structure of the SLASEM is composed of motion predict, measurement update and integration of new environment samples. This is similar to conventional EKF-SLAM algorithms.

4.1. State Definition

Denote the system state at time instant \(k\) as \(X(k)\), which is a combination of a robot’s pose and environment samples’ positions:
\[
X(k) = \begin{bmatrix} X_r^T(k) & X_{s1}^T & \cdots & X_{sn}^T \end{bmatrix}^T,
\] (8)
where \(X_r(k)\) describes the robot’s pose which is usually composed of position and orientation, and \(X_{si}(i = 1, \ldots, n)\) is the position of the \(i\)th environment sample, which is assumed to be lying on a certain obstacle’s contour. The mean and covariance of the state \(X(k)\) are denoted by \(\hat{X}(k|k)\) and \(P(k|k)\):
\[
\hat{X}(k|k) = \begin{bmatrix} \hat{X}_r^T(k|k) & \hat{X}_{s1}^T & \cdots & \hat{X}_{sn}^T \end{bmatrix}^T
\] (9)

\[
P(k|k) = \begin{bmatrix}
P_{rr}(k|k) & P_{r1}(k|k) & \cdots & P_{rn}(k|k) \\
P_{r1}(k|k) & P_{11}(k|k) & \cdots & P_{1n}(k|k) \\
\vdots & \vdots & \ddots & \vdots \\
P_{rn}(k|k) & P_{n1}(k|k) & \cdots & P_{nn}(k|k)
\end{bmatrix}.
\] (10)
4.2. The Motion and Observation Models

At time instant $k+1$, the robot moves to $X_r(k+1)$. The motion model is:

$$X(k+1) = f(X(k), u(k), w_k),$$  \hspace{1cm} (11)

where $u(k)$ is the control input, and $w_k$ is assumed to be uncorrelated Gaussian distributed noise with zero mean and covariance $Q_k$.

The robot gets raw observations $z(k+1)$ at position $X_r(k+1)$. According to the method described in Section 3, a new vector-valued observation function $h(X(k+1), V^d_{k+1})$ is formed by stacking the distance functions corresponding to the environment samples in current robot’s observation scope:

$$h(X(k+1), V^d_{k+1}) = \begin{bmatrix} g(x^p_{i_1}, y^p_{i_1}) + v^d_1 \\ g(x^p_{i_2}, y^p_{i_2}) + v^d_2 \\ \vdots \\ g(x^p_{i_m}, y^p_{i_m}) + v^d_m \end{bmatrix},$$ \hspace{1cm} (12)

where $[x^p_{ij}, y^p_{ij}]^T$ ($j = 1, \ldots, m$) is the position in Cartesian coordinate of the $ij$th environment sample in robot-centric Cartesian space, $m$ is the number of environment samples that are conjectured to be observed, $V^d_{k+1} = [v^d_1, v^d_2, \ldots, v^d_m]^T$ is the measurement noise, which is assumed to be zero mean and uncorrelated Gaussian distributed. Its covariance matrix $R^d_{k+1}$ is:

$$R^d_{k+1} = \begin{pmatrix} r_{11} & 0 & 0 & 0 \\ 0 & r_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & r_{mm} \end{pmatrix},$$ \hspace{1cm} (13)

$$r_{jj} = R_0/n_j \hspace{1cm} (j = 1, 2, \ldots, m).$$ \hspace{1cm} (14)

Here, $R_0$ is the basic covariance. Its value is determined by the sensor. $n_j$ is the number of raw observations in the neighborhood of $[x^p_{ij}, y^p_{ij}]^T$. All the $n_j$ raw observations are supposed to be observations of the $ij$th environment sample. As all of them are in a small neighborhood of the $ij$th environment sample, they are supposed to be identical and with a common covariance $R_0$ approximately. In the Kalman filter, the effect of $n_j$ identical observations with a common covariance $R_0$ is equal to the effect of an observation with covariance $R_0/n_j$. In this way, the measurement covariance of the $ij$th environment sample is set as $R_0/n_j$. Suppose $N$ is the number of raw observations, then we have:

$$\sum_{j=1}^{m} n_j = N.$$ \hspace{1cm} (15)

The vector-valued function $h(X(k+1), V^d_{k+1})$ is considered as the observation model at time instant $k+1$. The predicted measurements are calculated by:

$$z^d(k+1) = h(X(k+1), 0).$$ \hspace{1cm} (16)
In this paper, \( \mathbf{0} \) always denotes a dimension-compatible zero vector. The actual measurements are assumed to be zero in the procedure of measurement update.

4.3. Estimation Process

At time instant \( k+1 \), the system state \( X(k+1) \) can be updated by using an EKF when the new observation model \( h(X(k+1), V^d_{k+1}) \) has been acquired. A single step is described as below:

(i) **Motion predict.** This step projects the state at time instant \( k \) to the state at time instant \( k+1 \):

\[
\hat{X}(k+1|k) = f(\hat{X}(k|k), u(k), \mathbf{0})
\]

\[
P(k+1|k) = FP(k|k)F^T + GQ_kG^T,
\]

where:

\[
F = \frac{\partial f}{\partial X} \bigg|_{\hat{X}(k|k)} \quad \text{and} \quad G = \frac{\partial f}{\partial u} \bigg|_{u(k)},
\]

are the Jacobian matrices of the motion model with respect to the state \( X(k) \) and the control input \( u \), respectively. Note that \( Q_k \) is the covariance of the control input which is defined in Section 4.2.

(ii) **Measurement update.** This step updates the state \( X(k+1) \) by using the observation model (12):

\[
\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + W(k+1)\nu(k+1)
\]

\[
P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W^T(k+1),
\]

where:

\[
\nu(k+1) = -h(\hat{X}(k+1|k), \mathbf{0})
\]

\[
W(k+1) = P(k+1|k)H^T S^{-1}(k+1)
\]

\[
S(k+1) = HP(k+1|k)H^T + R^d_{k+1},
\]

where:

\[
H = \frac{\partial h}{\partial X} \bigg|_{\hat{X}(k+1|k)},
\]

is the Jacobian matrix of the observation model with respect to the state \( X(k+1) \) and \( R^d_{k+1} \) is the covariance of the measurement noise, which is given by (13).

4.4. Integration of a New Environment Sample

Observed points that are far away from the predicted map are selected to be inserted into the system state. These selected points should distribute on the contour they belong to as uniformly as possible. For each selected point, the minimum distance between it and its neighbors should also be longer than a predefined value. The detailed procedures of how to insert a point into the system state are given as follows.
Assume that a point whose range-bearing data is $\tilde{z}$ is waiting to be inserted into the system state, and:

$$pt = l(X, \tilde{z}), \quad (24)$$

is a function that transforms the range-bearing data into a coordinate $pt$ in the global reference frame. Here, $X$ is the system state. Then this point is inserted by:

$$\hat{X}(k+1|k+1) = \left[ \begin{array}{c} \hat{X}(k+1|k+1) \\ l(\hat{X}(k+1|k+1), \tilde{z}) \end{array} \right] \quad (25)$$

$$P(k+1|k+1) = \begin{bmatrix} P(k+1|k+1) & (k+1)P(k+1|k+1)L_{z}^{T} \\ L_{x}P(k+1|k+1) & L_{x}P(k+1|k+1)L_{x}^{T} + L_{z}\tilde{R}L_{z}^{T} \end{bmatrix} \quad (26)$$

$$\tilde{R} = R_{0}/\tilde{n}, \quad (27)$$

where:

$$L_{x} = \left. \frac{\partial l}{\partial X} \right|_{\hat{X}(k+1|k+1)} \quad \text{and} \quad L_{z} = \left. \frac{\partial l}{\partial z} \right|_{\tilde{z}},$$

are the Jacobian matrices of $l(X, z)$ respect to $X$ and $z$, respectively. $\tilde{n}$ is the number of observed points that are in the neighborhood of $pt$. The purpose of dividing the basic covariance $R_{0}$ by $\tilde{n}$ is to compensate for the effect caused by the observed points that would not be inserted into the state.

5. Removing Redundant Environment Samples

During the estimation process, environment samples in the SEM may be crowded somewhere. In this situation, only a part of them should be preserved to make the SEM compact. For the environment samples corresponding to the same contour, the environment samples to be preserved are selected using the following three steps:

(i) Initialize a preserving list using both the end-points of the contour.

(ii) For each remaining environment sample, add it into the preserving list if the minimum distance from it to the environment samples that are already in the preserving list is larger than a predefined distance.

(iii) Check the number of environment samples in the preserving list. If the number of environment samples in the preserving list is more than a predefined number, then the procedures stop and the environment samples to be preserved are the ones in the preserving list. Otherwise, the predefined distance is decreased and (ii) is re-executed.

Except for these selected environment samples, the others are considered as redundant ones. Those redundant environment samples need to be removed from the SEM to make it more compact.

A simple method to remove redundant environment samples from the SEM is eliminating their counterparts in the state mean $\hat{X}(k+1|k+1)$ and covariance $P(k+1|k+1)$ directly. This method does nothing about compensation when removing redundant environment samples, so there are great differences between the
Figure 3. (a) Pentagrams \{S_1, S_2, S_3, S_4, S_5\} are the environment samples used to represent a wall. The solid line is fitted with \{S_1, S_2, S_3, S_4, S_5\}. The dashed line is fitted with \{S_1, S_3, S_5\}. (b) Dots \{T_1, T_2, T_3, T_4, T_5\} are the environment samples transformed from \{S_1, S_2, S_3, S_4, S_5\} after applying geometric constraints, respectively. The solid line is fitted with \{T_1, T_3, T_5\}. It is the same as both the one fitted with \{T_1, T_2, T_3, T_4, T_5\} and the one fitted with \{S_1, S_2, S_3, S_4, S_5\}.

shapes represented by the remaining ones and the shapes before the elimination. This means that the accuracy of the SEM is decreased when using this method.

Here, we show an example to illustrate this problem. Assume that \{S_1, S_2, S_3, S_4, S_5\} are the environment samples used to represent a wall, as shown in Fig. 3a. The solid line is fitted with \{S_1, S_2, S_3, S_4, S_5\}. Following the principle above, \{S_1, S_3, S_5\} are selected as the well distributed environment samples to represent the wall and \{S_2, S_4\} are considered as the redundant ones. The dashed line is fitted with \{S_1, S_3, S_5\}. Obviously, there are great differences between the solid line and the dashed line. This means that \{S_1, S_3, S_5\} do not describe the wall well. From this example, we can see that the simple method cannot produce a satisfactory result.

In this section, a method considering geometric constraints is proposed to remove redundant environment samples. Geometric constraints are defined to restrict environment samples to lie on the contours they belong to exactly. Before removing redundant environment samples, geometric constraints are applied on the environment samples in crowded areas to make sure that they lie on the contours they belong to. In this way, no matter which environment samples are removed, the shapes represented by the remaining ones will not change compared with the previous shapes.

Geometric constraints used in this research are the IPs fitted with the environment samples in the crowded area. An IP is fitted for each contour in each crowded areas. For convenience of discussion, assume that there is only one crowded area. Furthermore, there is only one contour in this crowded area. Thus, an IP is fitted with the environment samples in this area.

Note that fitting an IP here only uses the environment samples in the crowded area, rather than all the environment samples belonging to the whole contour. The reason is that the shape of the whole contour may be too complex to be fitted well with a low-order IP. However, a small segment of the contour (the part in the crowded area) can be fitted well in most cases. In the case that the environment
samples in the crowded area cannot be fitted well, they will be partitioned into two or more sets, then two or more IPs are fitted accordingly.

The fitted IP is considered as a common constraint for all of the environment samples in the crowded area. Assume that the IP is denoted by \( p(x, y) = 0 \) and the environment samples in the crowded area are \( \{[x_{ij}, y_{ij}]^T, j = 1, \ldots, m\} \). Thus, geometric constraints for these environment samples can be written as:

\[
\begin{align*}
    p(x_{i_1}, y_{i_1}) &= 0 \\
p(x_{i_2}, y_{i_2}) &= 0 \\
    &\vdots \\
p(x_{i_m}, y_{i_m}) &= 0. \\
\end{align*}
\] (28)

By linearizing (28), a general form of the linear geometric constraints is achieved as:

\[
DX(k + 1) = d. 
\] (29)

Here, \( D \) is a \( m \times s \) matrix, \( d \) is a \( m \times 1 \) vector and \( s \) is the length of \( X(k + 1) \).

By using the ECKF [28], the state can be transformed into the one that satisfies the geometric constraints (29):

\[
\begin{align*}
    \hat{X}^P(k + 1|k + 1) &= \hat{X}(k + 1|k + 1) + K_{k+1}^P(d - \hat{d}) \\
P^P(k + 1|k + 1) &= P(k + 1|k + 1) - K_{k+1}^P P^{dd}(k + 1|k + 1) K_{k+1}^P, \\
\end{align*}
\] (30) (31)

where:

\[
\begin{align*}
    \hat{d}_{k+1} &= DX(k + 1|k + 1) \\
P^{dd}(k + 1|k + 1) &= DP(k + 1|k + 1)D^T \\
P^{xd}(k + 1|k + 1) &= P(k + 1|k + 1)D^T \\
K_{k+1}^P &= P^{xd}(k + 1|k + 1) (P^{dd}(k + 1|k + 1))^{-1}. \\
\end{align*}
\] (32) (33) (34) (35)

Here, \( \hat{X}^P(k + 1|k + 1) \) and \( P^P(k + 1|k + 1) \) are the mean and covariance of the new state, respectively.

In this way, the environment samples in the crowded area lie on the contour they belong to exactly. Later, a set of the environment samples that distribute on the contour uniformly are selected. Except for these selected environment samples, the others in the crowded area are considered as the redundant ones. These redundant environment samples are eliminated by removing their counterparts in \( \hat{X}^P(k + 1|k + 1) \) and \( P^P(k + 1|k + 1) \) directly. The application of geometric constraints ensures that the shape represented by the remaining environment samples does not change compared with the previous one. In the cases where there are more crowded areas or there are more contours in a crowded area, removing redundant environment samples can be solved in an analogical way.

In Fig. 3b, the geometric constraint is the IP fitted with \( \{S_1, S_2, S_3, S_4, S_5\} \). After applying this constraint, \( \{S_1, S_2, S_3, S_4, S_5\} \) are transformed into \( \{T_1, T_2, T_3, T_4, T_5\} \).
Then \( \{T_1, T_3, T_5\} \) are selected as the well-distributed environment samples and \( \{T_2, T_4\} \) are considered as the redundant ones to be eliminated. The line fitted with the left environment samples \( \{T_1, T_3, T_5\} \) is the same as the one fitted with \( \{S_1, S_2, S_3, S_4, S_5\} \). Thus, there is no difference between the shape represented by the left environment samples and the one represented by all the environment samples.

6. Improving the Topological Consistency of the SEM

The SEM is said to be topologically inconsistent when the topological structure of the SEM violates the real one. For example, two connected walls are represented as two crossed walls, as shown in Fig. 4a and b. The aim of this section is to improve the SEM’s topological consistency, such as prohibiting the existence of diamonds and pentagrams in Fig. 4b.

To improve the topological consistency of the environment samples belonging to the same contour, an inequality constraint \( c(x, y) \leq 0 \) is used to restrict them to lie on the same side of the curve that is represented by \( c(x, y) = 0 \). A natural choice of the constraint is the IP fitted with the environment samples belonging to its adjacent contour. If the adjacent contour is too complex to be fitted with a first- or second-order IP, part of its environment samples near the corner are selected to be fitted. As this kind of inequality constraint is used to improve the inconsistencies in
corners, we call it a corner constraint. For all the environment samples, their corner constraints are described as:

$$\tilde{D}(X) \leq \tilde{d}.$$  \hspace{1cm} (36)

Here $\tilde{D}(\cdot)$ is a vector-valued function and $\tilde{d}$ is a dimension compatible vector.

We give an example here. In Fig. 4c, the inequality constraint $p_1(x, y) < 0$ is used to restrict the environment samples belonging to the thin wall to lie on the right side of the thick wall. In Fig. 4d, the inequality constraint $p_2(x, y) < 0$ is used to restrict the environment samples belonging to the thick wall to lie below the thin wall.

A solution to the Kalman filter with inequality constraints is the active set method [29]. This method ignores the inequality constraints that are satisfied. The inequality constraints that are unsatisfied are translated into equality constraints. Thus, the inequality constraint problem is changed into an equality constraint problem, which can be solved in the same way as that in Section 5.

Assume that at time instant $k + 1$, the inequality constraints (36) are checked out and the ones that are unsatisfied are selected to be linearized as:

$$\tilde{D}X(k + 1) \leq \tilde{d}.$$  \hspace{1cm} (37)

Here, $\tilde{D}$ is a $\tilde{m} \times s$ matrix, $\tilde{d}$ is a $\tilde{m} \times 1$ vector, $\tilde{m}$ is the number of constraints and $s$ is the length of $X(k + 1)$.

Following the active set method, the inequality constraints are translated into the equality ones:

$$\tilde{D}X(k + 1) = \tilde{d}.$$ \hspace{1cm} (38)

Finally, the ECKF with equality constraints (38) is used to update the state following procedures (30)–(35). In this way, all the environment samples are restricted to satisfy the corner constraints. Consequently, the topological consistency of the SEM is improved.

7. Simulation and Experimental Results

In this section, we show simulation and experimental results to illustrate that SLASEM is effective in localizing the robot while building a map of the complex environment.

7.1. Simulation

This simulation is used to verify the convergence of SLASEM. All the noise settings are chosen to simulate a real environment.

In the simulation, a mobile robot moves along a dark curve (nearly overlapped with the light curve), while observing the obstacles, as shown in Fig. 5a. The control noises are assumed to be Gaussian distributed ones with zero mean and covariance $\text{diag}((v \times 30\%)^2, (5^\circ)^2)$. Here, $v$ denotes the current speed of the robot. The noise
Figure 5. Simulation results. (a) SEM built by SLASEM without corner constraints. The SEM is composed of 95 environment samples. Grey areas with line contours denote the actual obstacles. Dots are the SEM built by our algorithm. The dark curve is the robot’s real trajectory. The light curve denotes the estimated trajectory. Both curves nearly overlap with each other. The dashed line shows the odometry data. For the eight objects, (b) shows the mean distances from their environment samples to them.

of each laser beam is assumed to be additive Gaussian distributed with zero mean and covariance diag((0.05m)², (0.5°)²).

Figure 5a shows the SEM built by SLASEM without corner constraints. It can be seen from Fig. 5b that the SEM describes the environment well. All the environment samples lie around the actual obstacles’ contours. The distances from the environment samples to the line shape contours distribute uniformly. The reason is that the observation models of line shape contours are linear functions of the state, thus the environment samples can be updated effectively. The environment samples belonging to the arc shape contours distribute well in the middle part and have biases at the boundary zones of each arcs. The main reason for this phenomenon is that there are insufficient observations at the boundary zones, causing the measurement covariance computed in (14) to be larger than the one in the middle part. Thus, in the process of measurement update (19), the environment samples at the boundary zones receive little modifications. The nonlinear observation models of the environment samples belonging to the arc shape contours may also contribute to the negative influence. For the objects in the simulated environment, Fig. 5b shows the mean distance from the environment samples to them as a function of time. The distance curves do not converge to zero. However, all of them keep at a low level. This is the inconsistency phenomenon caused by the linearization error [30].

The SEM shown in Fig. 5a is nearly topologically consistent. So there is no need to add corner constraints when building the map.
7.2. Experimental Results

SLASEM has also been tested in two experiments carried out in the Shenyang Institute of Automation. In both experiments, a scanning laser range finder URG-04LX was used to collect measurements. Sensor measurements were made ranging from $0^\circ$ to $240^\circ$ with approximately $0.36^\circ$ steps and with an effective distance up to 4 m. The computational time of the algorithm was determined primarily by the time required for partitioning raw observations into primitive objects due to the simple method we used. During the experiments, data were saved in log files and processed offline on a fast PC. The developed maps were different with the schematic diagrams due to the existence of objects replenished later, such as doors and tables. For comparison, a grid map was built by GMapping in each experiment [31]. The GMapping was run with 100 particles and with a resolution of 0.05 m.

7.2.1. Experimental Result I

The first experiment was carried out in the 10th floor of the students’ apartment. URG-04LX was equipped on the front of a handcart, which was slowly pushed by a man. There was no odometry sensor equipped on the handcart. Instead we used the iterative closest point (ICP) method to obtain the odometry data from laser data [32]. The environment was a semicircular corridor. Most of the objects are described in the SEM except for some objects that reflect too few laser beams, such as the narrow nooks formed by the downcomers and their adjacent walls. The schematic diagram of the building is shown in Fig. 6a, in which the reference frame and a loop of the trajectory are depicted approximately. Figure 6b shows a snapshot of the environment.

![Figure 6. (a) Schematic diagram of the environment. The light curve shows a loop of the trajectory approximately. The axes shows the reference frame that the developed map is relative to. (b) Snapshot of the environment.](image-url)
Figure 7. (a and b) SEMs built by SLASEM without and with corner constraints, respectively. The curve is the estimated trajectory of the handcart. Points are the SEMs. The dot and square nearby the origin denote the start and end point, respectively. (c) Grid map built by GMapping. In all of the three panels, the upper left zoomed axes describes a common area.

Figure 7a shows the SEM built by SLASEM without corner constraints. The map, which is composed of 558 environment samples, represents the structure of the environment well. The estimated trajectory is consistent with the map in the sense that the trajectory never produces collision with walls. Figure 7b shows the SEM built by SLASEM with corner constraints. It has 556 environment samples. From Fig. 7b, we can see that the SEM not only describes the environment well, as in Fig. 7a, but also depicts the environment with consistent topology. A typical interval corresponding to the same zone is zoomed in to show the effects of corner constraints. It can be seen that the algorithm with corner constraints produces a
Figure 7c gives a grid map built by GMapping. Taking the schematic diagram as a reference, we can see that both SLASEM and GMapping give a map that describes the structure of the environment correctly. Moreover, the map built by SLASEM describes the environment with more details compared with the grid map. For example, in the zoomed axes, the corners formed by adjacent walls can be seen clearly in Fig. 7b, while they are unclear in Fig. 7c.

7.2.2. Experimental Result II

The second experiment was carried out in the third floor of the Robotic Building. URG-04LX was equipped on the top of a shape-shifting robot, AMOEBA-I [33]. In this experiment, the robot moved around our laboratory in the first loop. Then it moved around the corridor in the second loop. It finally stopped nearby where it started. The schematic diagram of the environment is shown in Fig. 8a, in which

Figure 8. (a) Schematic diagram of the environment. The curve shows the trajectory approximately. The axes show the reference frame that the developed map is relative to. (b) Snapshot of the environment in area A. (c) Snapshot of the environment in area B.
the reference frame and the trajectory are depicted approximately. Figure 8b and c shows two snapshots of the environment.

Figure 9a and b shows the SEMs produced by SLASEM without and with corner constraints, respectively. There are 1338 environment samples in Fig. 9a and 1330 in Fig. 9b. Walls and flowerpots are represented well in both of the SEMs. As there are some objects reflecting too few laser beams and some objects whose reflected laser beams cannot be received by the sensor effectively, such as pillars and glass doors, the resultant SEMs differ from the real world a little. The SEM in Fig. 9a can represent the environment well in most parts of the environment. As it is built without corner constraints, there are topologically inconsistent phenomena that can be seen from the zoomed axes. Although the robot closes the loop finally, the shape nearby the origin is distorted. The SEM in Fig. 9b is built with corner constraints. Its topological consistency is improved, which can be seen from the zoomed axes. The improvement of topological consistency also increases the accuracy of the SEM. Thus, the robot can close the loop well. Figure 9c shows a grid map produced by GMapping. Contours and corners in the grid map are unclear. Since the grid map and the robot’s trajectory are not updated in one filtering stage, the result produced by GMapping is not accurate enough to enable the robot to close the loop. This leads to a mismatch in the grid map near the origin. By comparing the global structure and details of the maps in Fig. 9b and c, we can see that the map built by SLASEM with corner constraints is more accurate than GMapping.

8. Conclusions

This paper has presented a novel algorithm for the problem of SLAM in complex environments. The SEM adopted in this algorithm can represent complex environments efficiently. An observation model has been proposed to deal with the problem of the lack of one-to-one correspondence between the robot’s observations and environment samples. Furthermore, we have introduced an effective approach to remove redundant environment samples from the SEM and an approach to improve the topological consistency of the SEM by using corner constraints. A simulation and two indoor experiments have been performed to verify the proposed algorithm.

The proposed algorithm builds a SEM and estimates the robot’s pose in one filtering stage. This endows the algorithm with a more accurate result compared with the ones produced by using the intermediate results of the measurements. Moreover, the property of using one filter to estimate the whole state enables the algorithm to achieve better performances by filter design. For example, (i) we can use the method in Ref. [34] to reduce the computational complexity, and (ii) we can improve the accuracy by using an unscented Kalman filter (UKF) instead of an EKF to estimate the state, as the featured-based UKF-SLAM algorithms do [35, 36].
Figure 9. (a and b) SEMs built by SLASEM without and with corner constraints, respectively. The curve is the estimated trajectory. Points are the SEMs. The dot and square nearby the origin denote the start point and end point, respectively. (c) Grid map built by GMapping. In all of the three panels, three common areas are zoomed in to show details of the maps.
Acknowledgements

This work is supported by the National High Technology Research and Development Program of China (863 Program) (2006AA042230 and 2007AA041502-5).

References

About the Authors

**Rongchuan Sun** received the BE degree in Automation from the University of Science and Technology of China, Hefei, P. R. China, in 2004. He is a PhD Candidate in the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, P. R. China. His research interests include mobile robot navigation, estimation and system modeling.

**Shugen Ma** received the BE degree in Mechanical Engineering from the Hebei Institute of Technology, Tianjin, P. R. China, in 1984, and the ME and DE degrees in Mechanical Engineering Science from the Tokyo Institute of Technology, Tokyo, Japan, in 1988 and 1991, respectively. From 1991 to 1992, he worked for Komatsu Ltd as a Research Engineer, and from 1992 to 1993, he was at the University of California, Riverside, USA, as a Visiting Scholar. Since July 1993, he was with Ibaraki University, Japan, as an Associate Professor of the Department of Systems Engineering. In October 2005, he joined Ritsumeikan University and now is a Professor of the Department of Robotics. He also holds a Professor position with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, P. R. China. His research interests are in the design and control theory of new types of robots and biorobotics.

**Bin Li** received the MS degree in Medical Science from China Medical University, Shenyang, P. R. China, in 1988. Since 1989, he has been with Shenyang Institute of Technology, Shenyang, where he is currently a Professor of the State Key Laboratory on Robotics. His research interests are in rescue robotics and biorobotics.

**Minghui Wang** received the BE degree in Computer Application from Liaoning University, Shenyang, P. R. China, in 2002, and the DE in Mechatronics from the Shenyang Institute of Automation, Chinese Academy of Science, Shenyang, in 2007. Since 2007, he has been with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, where he is currently an Assistant Professor. His research interests are in reconfigurable robots and modular robots.

**Yuechao Wang** received the DE degree in Mechanical Engineering from Harbin Institute of Technology, Harbin, P. R. China, in 1999. Since 1987, he has been with the Shenyang Institute of Automation, where he is currently a Professor and the Director of the Shenyang Institute of Automation, Chinese Academy of Sciences. He also holds the Director position of the State Key Laboratory on Robotics. His research interests are in robotics and automation.