Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

A Computational Efficient SLAM Algorithm Based on Logarithmic-Map Partitioning

H. Jacky Chang, C. S. George Lee, Yung-Hsiang Lu, and Y. Charlie Hu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-2035
{chang26, csglee, yunglu, ychu}@purdue.edu
https://engineering.purdue.edu/ResearchGroups/DEAR

Abstract—**Simultaneous localization and map building (SLAM) is a fundamental and complex problem in mobile robot research. In SLAM, Kalman-Filter-like implementations are widely adopted to localize a mobile robot and build a map simultaneously and incrementally. However, this approach requires extensive computations of order $O(N^2)$, where N is the total number of landmarks. To make the computations more manageable, we propose a logarithmic-map partitioning algorithm that partitions the global map into one local region and several sub-maps. The size of each sub-map is based on its distance from the mobile robot, and in each sub-map, a centroid landmark is selected to represent all the landmarks in the sub-map for SLAM computations. With this logarithmic-map partitioning, it maintains correlation updates with each sub-map and provides an efficient suboptimal solution to the SLAM problem. The number of landmarks reduces from N to a logarithm-based function of N, and the computational requirement reduces from $O(N^3)$ to $O(N^2)$, where N_L is the number of local landmarks. Furthermore, utilizing the Compressed Extended Kalman Filter, the real-time computational complexity reduces to $O({N_L}^2)$. Computer simulation results showed that the proposed algorithm is consistent and efficient for a large number of landmarks.**

I. Introduction

Simultaneous localization and map building (SLAM) [1], [2], also refereed by many as the concurrent mapping and localization (CML) problem [3], [4], is a fundamental and complex problem in mobile robotics research. The goal of SLAM is to put a mobile robot in an unmapped terrain, and the mobile robot explores the terrain and constructs a map without any priori map information. An efficient SLAM algorithm can make a mobile robot truly "autonomous," which is invaluable in many real-world applications such as search-and-rescue operations, planetary exploration, undersea operation, and air-borne robotics surveillance [5]–[7].

The seminal SLAM paper was presented by Smith, *et al.* [1]. In their paper, the authors proposed a probabilistic procedure to build a stochastic map and localize the mobile robot in the map. Since then, the probabilistic techniques have been widely adopted to solve the SLAM problem. Generally speaking, these techniques are all based on Bayesian filters. They use previous sensory information and motion command to estimate the best mobile robot pose (i.e., position and orientation) and build a map simultaneously. If the noise model is Gaussian, Kalman Filter (KF) or Extended Kalman Filter (EKF) provides a recursive optimal solution for the SLAM problem [8]–[10]. However, as the number of landmarks increases, KF and EKF require $O(N^2)$ computation time in each iteration. When N is large, it is almost impossible to update the map in real time. To overcome this problem, several approaches were developed [11]–[16].

In [11], the authors divided the map into same-size local maps and applied the Compressed Extended Kalman Filter (CEKF) to achieve a computation time of $O({N_L}^2)$ in the local map. The major problem of this approach is that it requires a full SLAM update when the mobile robot moves from one local map to another, and it requires an update computation time of $O(N^2)$. By observing the correlation between landmarks, Sparse Extended Information Filter (SEIF) [15] has $O(1)$ performance through representing maps in local, web-like maps and neglects distance correlation. In [13], an overlapping map method with fixed size sub-map is adopted to achieve $O(1)$ computation. However, it loses the correlation between two non-adjacent sub-maps and ignores some measurements, which can assist the convergence of sub-maps more quickly. In [12], the local sub-map is independent to each other and it uses a covariance matrix to represent the relationship between the local frames and the global frame. In their approach, the number of sub-maps is still proportional to the number of the landmarks, and the computational requirement is still $O(N^2)$ from a global point of view. In [17], decoupled stochastic mapping (DSM), an $O(1)$ algorithm, divides the environment into multiple overlapping sub-maps, each with its own stochastic map but only one activated sub-map is updated with the mobile robot.

FastSLAM [18] with $O(\log N)$ computational complexity successfully deals with the nonlinear SLAM problem. It divides SLAM into two subproblems, localization and mapping, and solves them by using Rao Blackwellized particle filter and EKF, respectively. However, the computational complexity depends linearly on the number of particles, which scales with the environment size.

From previous research, it has been shown that map

†This work was supported in part by the National Science Foundation under Grant IIS-0329061.

0-7803-8463-6/04/$20.00 ©2004 IEEE

partitioning approaches were widely used and successfully applied to the SLAM problem. However, in many $O(1)$ algorithms, the relations between two non-adjacent sub-maps are ignored or treated equally important in the same-size sub-maps. In this paper, we assume that data association is known and it can be solved by using techniques such as Joint Compatibility Testing [19] or scan matching [20]. With this assumption, we investigate a new efficient SLAM algorithm based on logarithmic-map partitioning, which divides a global map into a local region and several sub-maps with different sizes. The size of each sub-map depends on its distance from the mobile robot. In addition, the proposed SLAM algorithm maintains acceptable covariance approximation between each sub-map. We then use the Compressed Extended Kalman Filter (CEKF) to compute the local map in $O({N_L}^2)$ and update the global map in $O((N_L + \log N)^2)$, where $N_L \ll N$. Once the mobile robot moves from one local map to another, the proposed SLAM algorithm updates all the landmarks in each sub-map without running the full SLAM algorithm, and then partitions the global map again. Using the proposed logarithmic-map partitioning method, the computational requirement of constructing a whole global map reduces from $O(N^3)$ to $O(N^2)$.

II. SLAM WITH EXTENDED KALMAN FILTER

This section briefly describes the notations in EKF and the mobile robot model. The system state is defined as $X(k) = [X_m(k), X_1, X_2, ..., X_i, ..., X_N]^T$, where k is the time index, $X_m(k) = [x_m(k), y_m(k), \theta_m(k)]^T$ is the mobile robot position and orientation angle (i.e., pose) at time k, and $X_i = (x_i, y_i)^T$, $i = 1, \ldots, N$, are the stationary landmark locations, and N is the number of landmarks. The system can be described as

$$\begin{aligned} X(k+1) &= F(X(k), U(k)) + v(k) \\ Z(k) &= H(X(k)) + w(k) \end{aligned} \quad (1)$$

where $F(\cdot)$ is a nonlinear function that generates the next state of the system from the current state and robot command $U(k)$, $v(k)$ is the system noise, $H(\cdot)$ is a nonlinear observation function, $w(k)$ is the measurement noise, and $Z(k)$ is the measurement of landmarks from the mobile robot. In the EKF approach, both $v(k)$ and $w(k)$ are independent Gaussian noises with zero mean and covariance matrices Q_F and Q_H, respectively.

As shown in Fig. 1, the command of the mobile robot consists of velocity $V(k)$ and steering angle $\varphi(k)$,

$$U(k) = [V(k), \varphi(k)]^T. \quad (2)$$

From the kinematics shown in Fig. 1, $X_m(k)$ is expressed as:

$$X_m(k) = \begin{bmatrix} V(k) \times \Delta t \times \cos(\theta_m(k-1)) + x_m(k-1) \\ V(k) \times \Delta t \times \sin(\theta_m(k-1)) + y_m(k-1) \\ \frac{V(k) \times \Delta t}{L} \times \tan(\varphi(k)) + \theta_m(k-1) \end{bmatrix}$$

Observation function $H(\cdot)$ generates the observation of the landmarks from the current mobile robot pose and decomposes it into $H(X(k)) = [H_1, \cdots, H_i, \cdots, H_N]^T$.

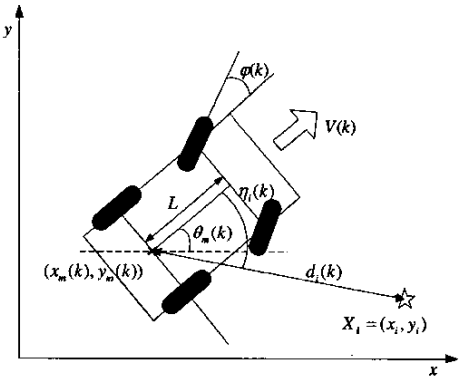


Fig. 1. The kinematics of the mobile robot.

If there is an observation of landmark X_i at system state $X(k+1)$, then H_i is expressed as

$$H_i(X(k+1)) = \begin{bmatrix} d_i(k+1) \\ \eta_i(k+1) \end{bmatrix} \quad (3)$$

Next, we apply the EKF algorithm to this system. Two stages are included in the EKF algorithm: prediction stage and update stage. In the prediction stage, EKF predicts system state $X(k+1|k)$ and covariance matrix $P(k+1|k)$ from the previous system state $X(k|k)$ and covariance matrix $P(k|k)$:

$$\begin{aligned} X(k+1|k) &= F(X(k|k), U(k)) \\ P(k+1|k) &= J_F(k) P(k|k) [J_F(k)]^T + Q_F(k) \end{aligned} \quad (4)$$

where $J_F(k)$ is the Jacobian matrix of F with respect to $X(k)$. In the update stage, EKF updates the previous prediction (in Eq. (4)) using the current observation, the Kalman innovation matrix S and the Kalman gain matrix W:

$$\begin{aligned} S(k+1) &= J_H(k+1) P(k+1|k) [J_H(k+1)]^T \\ &\quad + Q_H(k+1) \\ W(k+1) &= P(k+1|k) [J_H(k+1)]^T [S(k+1)]^{-1} \\ X(k+1|k+1) &= X(k+1|k) + W(k+1)(Z(k+1) \\ &\quad - H(X(k+1|k)) \\ P(k+1|k+1) &= P(k+1|k) - W(k+1) S(k+1) \times \\ &\quad [W(k+1)]^T \end{aligned} \quad (5)$$

where $J_H(k)$ is the Jacobian matrix of H with respect to $X(k)$. Since the landmarks are stationary, the Jacobian matrix $J_F(k)$ can be decomposed into

$$J_F(k) = \begin{bmatrix} J_{F_m}(k) & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \quad (6)$$

where J_{F_m} is the partitioned Jacobian matrix relating to the mobile robot pose, $\mathbf{0}$ and I are appropriate zero matrix and identity matrix, respectively. If the mobile robot observes landmark X_i, the Jacobian matrix of this observation is J_{H_i} and can be computed as following:

$$J_{H_i}(k) = \begin{bmatrix} K_{H_i}(k) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{\Delta x_i}{\Delta_i} & \frac{\Delta y_i}{\Delta_i} & 0 & 0 & \cdots & -\frac{\Delta x_i}{\Delta_i} & -\frac{\Delta y_i}{\Delta_i} & \cdots & 0 \\ -\frac{\Delta y_i}{\Delta_i^2} & \frac{\Delta x_i}{\Delta_i^2} & -1 & 0 & \cdots & \frac{\Delta y_i}{\Delta_i^2} & -\frac{\Delta x_i}{\Delta_i^2} & \cdots & 0 \end{bmatrix} \quad (7)$$

where Δx_i, Δy_i, and Δ_i are, respectively, defined as:

$$\Delta x_i = x_m - x_i, \Delta y_i = y_m - y_i; \Delta_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}. \quad (8)$$

III. SLAM WITH LOGARITHMIC-MAP PARTITIONING

The objective of map partitioning is to partition a large map into several more manageable sub-maps. Once the global map is properly partitioned, it can reduce the number of landmarks that needs to be computed in the EKF. Hence, the computation of the EKF will be reduced to an acceptable order. The concept of logarithmic-map partitioning is inspired from the observations in SEIF and DSM. In SEIF and DSM, they have shown that the updates for landmarks surrounding the mobile robot are necessary while the updates for distant landmarks can be ignored. However, in a map with uniformly distributed landmarks, it is difficult to determine a reasonable boundary between the surrounding and distant landmarks. From this point of view, the proposed logarithmic-map partitioning provides an efficient scheme in determining the different sizes of sub-maps.

The performance of the proposed logarithmic-map partitioning depends on three major issues: (1) Partitioning the terrain into global and local regions, (2) centroid-landmark representation in each sub-map, and (3) determining the size of each sub-map.

A. Partitioning Map into Global and Local Regions

In the logarithmic-map partitioning, the first step is to divide the terrain into a local region and a global region. The landmarks in the local region have strong correlations to each other and they cannot be ignored in EKF computations. Thus, the estimated locations of the landmarks in the local region will be predicted and updated by the EKF in realtime. The landmarks in the global region are less correlated to the landmarks in the local region, and they can be updated less frequent.

As shown in Fig. 2, a local region is defined as the rectangular area bounded by the boldline. The area outside the local region is the global region. Inside the local region, the gray area is the effective sensor range of the mobile robot when it is at the center of the area. This local-region partition will be updated when the mobile robot moves outside of the local region.

With this local-global-region partitioning, landmarks computed in EKF can be considered as consisting of two different sets, one in the local region, X_L, and the other in the global region, X_G.

$$X(k) = \begin{bmatrix} X_L(k) \\ X_G(k) \end{bmatrix}$$

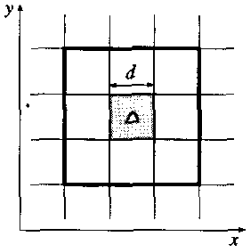


Fig. 2. Partitioning a local region and the global region from the map. The gray area is the effective sensor range of the mobile robot. The local region is defined as the rectangular area bounded by the bold line. The global region is outside the local region.

Their corresponding covariance matrix $P(k|k)$ is defined as

$$P(k|k) = \begin{bmatrix} P_{LL}(k|k) & P_{LG}(k|k) \\ P_{GL}(k|k) & P_{GG}(k|k) \end{bmatrix} \quad (9)$$

where the covariance matrix, P_{LL}, indicates the correlations between the mobile robot and local region landmarks.

From Section II, the prediction stage of the partitioned covariance matrix can be expressed as:

$$\begin{aligned} P_{LL}(k+1|k) &= J_{F_m}(k+1)P_{LL}(k|k)[J_{F_m}(k+1)]^T \\ &\quad +Q_{F_m}(k+1) \\ P_{LG}(k+1|k) &= [P_{LG}(k+1|k)]^T \\ &= J_{F_m}(k+1)P_{LG}(k|k) \\ P_{GG}(k+1|k) &= P_{GG}(k|k) \end{aligned}$$

The innovation matrix $S_i(k+1)$ and the Kalman gain matrix $W(k+1)$ of this observation are, respectively,

$$\begin{aligned} S_i(k+1) &= K_{H_i}(k+1)P_{LL}(k+1|k)[K_{H_i}(k+1)]^T \\ &\quad +Q_{H_i}(k+1) \end{aligned}$$

$$W(k+1) = \begin{bmatrix} P_{LL}(k+1|k)[K_{H_i}(k+1)]^T[S_i(k+1)]^{-1} \\ P_{GL}(k+1|k)[K_{H_i}(k+1)]^T[S_i(k+1)]^{-1} \end{bmatrix}$$

where $K_{H_i}(k)$ is defined in Eq. (7). The covariance $P(k+1|k+1)$ can be derived from the Kalman gain matrix $W(k+1)$,

$$P(k+1|k+1) = P(k+1|k) - \Delta P(k+1) \quad (10)$$

where $\Delta P(k+1)$ is

$$\begin{aligned} \Delta P(k+1) &= W(k+1)S(k+1)[W(k+1)]^T \\ &= \begin{bmatrix} \Delta P_{LL}(k+1) & \Delta P_{LG}(k+1) \\ \Delta P_{GL}(k+1) & \Delta P_{GG}(k+1) \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \Delta P_{LL}(k+1) &= P_{LL}(k+1|k)\psi(k+1)P_{LL}(k+1|k) \\ \Delta P_{GG}(k+1) &= P_{GL}(k+1|k)\psi(k+1)P_{LG}(k+1|k) \\ \Delta P_{LG}(k+1) &= [\Delta P_{GL}(k+1)]^T \\ &= P_{LL}(k+1|k)\psi(k+1)P_{LG}(k+1|k) \\ \psi(k+1) &= [K_{H_i}(k)]^T[S_i(k+1)]^{-1}K_{H_i}(k) \end{aligned}$$

From the above equations, the correlations between the global landmarks and the local landmarks are determined

by $P_{LG}(k|k)$. By applying CEKF, the computational requirement in the local region is bounded by $O({N_L}^2)$. When the mobile robot leaves the local region, $P_{GG}(k|k)$ and $P_{LG}(k|k)$ are updated by running the full SLAM algorithm. To overcome the full SLAM computational burden, only one landmark (i.e., a centroid landmark) is selected to represent all the landmarks in a sub-map in the global region. In the next subsection, the effect of the centroid-landmark representation is analyzed.

B. Analysis of Centroid-landmark Representation

Since the number of landmarks dominates the computational requirement in the EKF algorithm, an intuitive way is to reduce the number of landmarks in the global region since they are less correlated to the landmarks in the local region. An effective way is to select one representative landmark (i.e., a centroid landmark) to represent all the landmarks in each partitioned sub-map in the global region.

Assume that there are two neighboring landmarks X_{G1} and X_{G2}, which are in the same sub-map. The coordinate difference between X_{G2} and X_{G1} is δX; that is,

$$X_{G2} = X_{G1} + \delta X, \qquad \text{where} \quad \delta X = [\delta x, \delta y]^T .$$

The covariance matrices $P_{LG}(k|k)$ and $P_{GG}(k|k)$ are redefined as:

$$P_{LG}(k|k) = \left[\; P_{LG1}(k|k), \quad P_{LG2}(k|k) \; \right]$$

$$P_{GL}(k|k) = [P_{LG}(k|k)]^T = \begin{bmatrix} P_{G1L}(k|k) \\ P_{G2L}(k|k) \end{bmatrix}$$

$$P_{GG}(k|k) = \begin{bmatrix} P_{G1G1}(k|k) & P_{G1G2}(k|k) \\ P_{G2G1}(k|k) & P_{G2G2}(k|k) \end{bmatrix}$$

$P_{LG1}(k|k)$ and $P_{LG2}(k|k)$ are obtained from Eq. (10), and their difference is $\Delta P_{LG1,LG2}(k+1|k+1)$

$$\begin{aligned} & \Delta P_{LG1,LG2}(k+1|k+1) \\ = \; & P_{LG1}(k+1|k+1) - P_{LG2}(k+1|k+1) \\ = \; & [I - P_{LL}(k+1|k)\psi(k+1)] \times \\ & [P_{LG1}(k+1|k) - P_{LG2}(k+1|k)] \end{aligned} \tag{11}$$

Hence the update of $P_{LG1}(k|k)$ and $P_{LG1}(k|k)$ relies on their initial value. However, the effect of the initial value will be diminished after several EKF iterations. Thus, $\Delta P_{LG1,LG2}$ is controlled by $\psi(k)$. If we use landmark X_{G2} to represent landmark X_{G1}, the covariance matrix P_{LG2} will be used to represent the covariance matrix P_{LG1}. The difference between P_{LG1} and P_{LG2} will dominate the accumulation error. To see the difference, assume we have an observation on landmark X_{Gi} to construct P_{LGi}

$$\begin{aligned} & P_{LGi}(k+1|k+1) \\ = \; & (I - P_{LL}(k+1|k)\psi_{Gi}(k+1)) \times P_{LGi}(k+1|k) \end{aligned}$$

$$\psi_{Gi}(k+1) = [J_{H_{Gi}}(k+1)]^T [S_{Gi}(k+1)]^{-1} J_{H_{Gi}}(k+1) \tag{12}$$

The difference between $P_{LG1}(k+1)$ and $P_{LG2}(k+1)$ is caused by the initial value and the Jacobian matrices $J_{H_{G1}}(k)$ and $J_{H_{G2}}(k)$. If we set them to have the same initial value, then the only difference is $J_{H_{G1}}(k) - J_{H_{G2}}(k) = \Delta J_{H_{G1,G2}}(k)$,

$$\Delta J_{H_{G1,G2}}(k) = \begin{bmatrix} \frac{\delta x}{\Delta_{G1}} & \frac{\delta y}{\Delta_{G1}} & 0 & 0 & \cdots & -\frac{\delta x}{\Delta_{G1}} & -\frac{\delta y}{\Delta_{G1}} \\ -\frac{\delta y}{\Delta^2_{G1}} & \frac{\delta x}{\Delta^2_{G1}} & -1 & 0 & \cdots & \frac{\delta y}{\Delta^2_{G1}} & -\frac{\delta x}{\Delta^2_{G1}} \end{bmatrix} \tag{13}$$

When X_{G1} is close to X_{G2}, they both have about the same distance, Δ_{G1}, to the mobile robot, and $\delta x \ll \Delta_{G1}$, $\delta y \ll \Delta_{G1}$. Hence, $J_{H_{G2}}(k) \approx J_{H_{G1}}(k)$ and it implies $S_{G2}(k+1) \approx S_{G1}(k+1)$ and $P_{LG1} \approx P_{LG2}$. The update of X_{G1} and X_{G2} can be replaced by Eq. (14) if there is an observation on local landmark X_i and the updates are shown in Fig. 3.

$$\begin{aligned} \Delta V_{G1} &= W_{G1}(k+1)\Delta Z_i \\ X_{G1}(k+1|k+1) &= X_{G1}(k+1|k) + \Delta V_{G1} \\ \hat{X}_{G2}(k+1|k+1) &= \hat{X}_{G2}(k+1|k) + \Delta V_{G1} \end{aligned} \tag{14}$$

$$\begin{aligned} \Delta Z_i &= Z_i(k+1) - H_i(X(k+1|k)) \\ W_{G1}(k+1) &= P_{G1L}(k+1|k)[K_{H_{G1}}(k)]^T [S_{G1}(k+1)]^{-1} \\ &\approx W_{G2}(k+1) \end{aligned} \tag{15}$$

where $\hat{X}_{G2}(k+1|k+1)$ is an estimate of $X_{G2}(k+1|k+1)$ from the EKF computation.

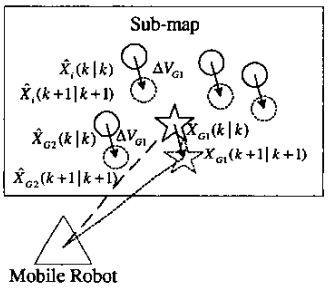


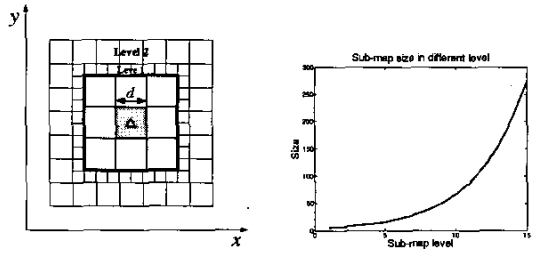
Fig. 3. The update in the sub-map is determined by the representative landmark X_{G1} and its update vector ΔV_{G1}. Every landmark in the sub-map will be shifted by ΔV_{G1}.

This approximation is only valid when the distance to the mobile robot is much larger than the distance between these two landmarks. In the next subsection, this relationship is used to define an error range that will be used to determine an appropriate size of each sub-map.

C. Logarithmic-Map Partitioning

From the last subsection, the error is determined by $\frac{\delta x}{\Delta}$. If an acceptable maximum error of map partitioning is $\varepsilon = \frac{\delta x}{\Delta}$, then we can calculate the acceptable sub-map size from this error ε. The map partitioning is shown in Fig. 4(a), where the sub-map size depends on the distance to the current mobile robot location and ε. The sub-maps have the same size in the same level. At levels $j+1$ and j, we have $\frac{\delta x_{j+1}}{\Delta_{j+1}} = \frac{\delta x_j}{\Delta_j} = \varepsilon$, and $\frac{\delta x_{j+1}}{\delta x_j} = \frac{\Delta_{j+1}}{\Delta_j} \geq 1$ because of $\Delta_{j+1} > \Delta_j$, where Δ_j is the distance from the

mobile robot to the sub-map level j. The area size of the sub-map is $2(\delta x_j)^2$, which means that δx_j determines the width of the sub-map at level j. The growing ratio of the area size is $\left(\frac{\delta x_{j+1}}{\delta x_j}\right)^2 = \left(\frac{\Delta_{j+1}}{\Delta_j}\right)^2 > 1$. Figure 4(b) shows that as the distance increases, the sub-map size increases exponentially, which is, in essence, partitioning the map in logarithmic.



(a) (b)

Fig. 4. (a) Logarithmic-map partitioning. (b) The sub-map size is defined by the sensor range and ε. The sensor range is set to $5m$, $d = 10m$ and $\varepsilon = 0.1$

Since the proposed logarithmic-map partitioning utilizes the traditional EKF approach, its convergence and consistence properties have been proved in [8]. Consider a simple case, if the mobile robot only moves in the local region and there exists only one landmark in each sub-map, then the EKF guarantees that the solution will converge. If there were more than one landmark in each sub-map, the local-region map will always converge because a centroid landmark was being used to represent all the landmarks in that sub-map. As the mobile robot moves out of the local region, it constructs a new converged local map again and again to cover the terrain.

IV. Analysis of Computational Complexity

We shall focus on three major computational components: logarithmic-map partitioning, execution of EKF in the local region, and the updating of centroid-landmark in each sub-map. Logarithmic-map partitioning assigns a sub-map number to each landmark and selects one centroid landmark to represent each sub-map. This process only requires $O(N)$. After the map partitioning, the number of landmarks used in the EKF computation is reduced to a logarithm-based function of N. Thus, the computational requirement in EKF is reduced to $O((N_L + \log N)^2)$, which is equal to $O({N_L}^2 + (\log N)^2)$. To further reduce the real-time computational requirement, Compressed Extended Kalman Filter (CEKF) is utilized in our algorithm. In CEKF, real-time computational requirement in the local region is bounded by $O({N_L}^2)$. When the mobile robot leaves the local region, it updates with a full SLAM algorithm in CEKF with $O({N_L}^2 + (\log N)^2)$. This is because logarithmic-map partitioning reduces the number of landmarks used in the EKF. Before the next map partitioning, as shown in Fig. 3, the locations of all the landmarks in the sub-maps are updated. The covariance matrix data are also updated in each landmark. This procedure will require $O(N)$. Thus, the total computational requirement is $O(N)$ for one local map construction.

Assume that the mobile robot explores the unknown terrain with a constant velocity and the landmarks are uniformly distributed in the terrain. The exploration time of the mobile robot through the whole terrain will be $O(N)$. Hence, the time in which the mobile robot needs to construct the whole map is $O(N^2)$.

V. Computer Simulation Results

Computer simulations were performed on the proposed SLAM algorithm and the original full SLAM algorithm. The simulation parameters are set as following: velocity $1m/s$, mobile robot process noise (std. dev.) $0.01m/s$, sensor range $3m$, $d = 6m$, sensor noise (std. dev.) $0.05m$, partitioning error parameter $\varepsilon = 0.2$, map size $20m \times 20m$ with 50 landmarks and a sampling period of $0.1s$. To construct a complete map, the mobile robot path is planned as a scanline to cover the whole area. In each time step, visible landmarks are selected to generate the observations.

Figure 5 shows the relative location error between the mobile robot and landmark X_1 and landmark X_2, respectively by using the proposed algorithm. Figure 5 is compared with Fig. 6, which is generated by the full SLAM algorithm. It shows the convergence of localization in the proposed algorithm. Figure 7 shows the landmark location error. Spikes occurred at the first time when the mobile robot entered a new local region with a new map partitioning. The spikes disappeared when the mobile robot re-entered explored local regions. When the map is fully explored, there are 50 landmarks in the full SLAM algorithm. In the proposed algorithm, at most 27 landmarks existed in the EKF computations which dramatically reduced the computational complexity. The generated final map of the terrain is shown in Fig. 8.

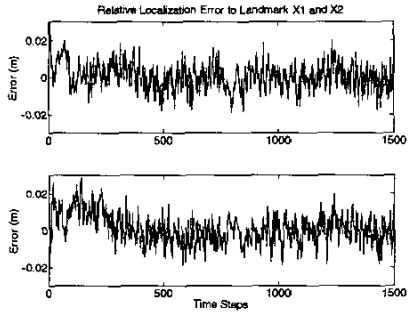


Fig. 5. The relative localization error between the mobile robot and the landmarks. The upper one is to landmark X_1 with $\sigma = 0.0148$, and the lower one is to landmark X_2 with $\sigma = 0.0136$.

VI. Conclusions

In this paper, we proposed a logarithmic-map partitioning algorithm that partitions the map into several sub-maps and one local area. The sub-map size is based on the distance from the mobile robot to the sub-map, and in each sub-map, a centroid landmark is selected to represent it in

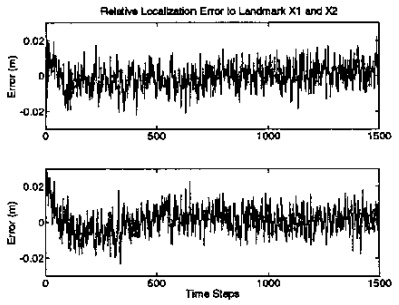


Fig. 6. The relative localization error between the mobile robot and the landmarks in the full SLAM algorithm. The upper one is to landmark X_1 with $\sigma = 0.0117$, and the lower one is to landmark X_2 with $\sigma = 0.0127$.

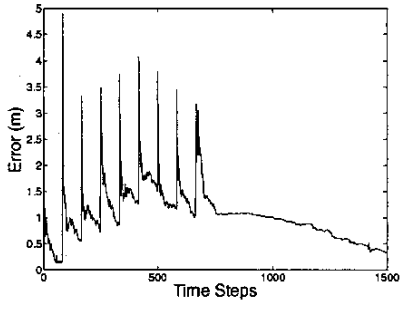


Fig. 7. The estimated error of the 50 landmarks of the proposed SLAM algorithm. Spikes occurred at the first time when the mobile robot entered a new local region with a new map partitioning. The spikes disappeared when the mobile robot re-entered explored local regions.

EKF. We analyzed the effect of map-partitioning in the EKF and also illustrated how to determine the sub-map size by a pre-defined error bound. The proposed SLAM algorithm provides a suboptimal but efficient solution to the SLAM problem, and the computational requirement of constructing a map reduces from $O(N^3)$ to $O(N^2)$. Our computer simulation result verified that the proposed SLAM algorithm is convergent and consistent. Future work will include experimentation of the proposed SLAM algorithms on our Pioneer 3-DX mobile robots with sonar sensors.

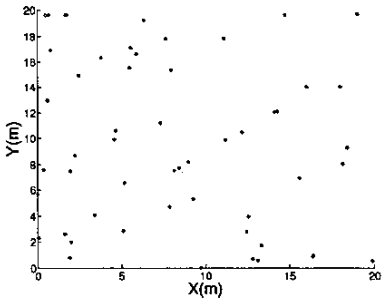


Fig. 8. The 20m×20m map with 50 landmarks. The mark '*' indicates the real location of the landmarks and 'o' indicates the estimated location of the landmarks by the proposed SLAM algorithm.

REFERENCES

[1] R. C. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Uncertainty in Artificial Intelligence 2*, pp. 435–461, 1988.

[2] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[3] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Machine Learning*, vol. 31, no. 1-3, pp. 29–53, 1998.

[4] J. W. Fenwick, P. M. Newman, and J. J. Leonard, "Cooperative concurrent mapping and localization," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, USA, May 2002, pp. 1810–1817.

[5] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *8th International Conference on Advanced Robotics*, July 1997, pp. 193–200.

[6] S. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneous localisation and map building," in *IEEE International Conference on Robotics and Automation*, April 2000, pp. 1793–1798.

[7] J.-H. Kim and S. Sukkarieh, "Airborne simultaneous localisation and map building," in *IEEE International Conference on Robotics and Automation*, vol. 1, Sept. 2003, pp. 406–411.

[8] M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[9] H. Feder, J. Leonard, and C. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robotics Research*, vol. 18, no. 7, pp. 650–668, July 1999.

[10] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE Int. Workshop on Intelligent Robots and Systems*, 1991, pp. 1442–1447.

[11] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map building algorithm for real-time implementation," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 242–257, June 2001.

[12] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "Efficient simultaneous localisation and mapping using local submaps," in *Australian Conference on Robotics and Automation*, 2001, pp. 128–134.

[13] P. M. Newman and J. J. Leonard, "Consistent, convergent and constant time slam," in *International Joint Conference on Artificial Intelligence*, Acapulco Mexico, 2003.

[14] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *IEEE International Conference on Robotics and Automation*, 2003.

[15] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Ng., "Simultaneous mapping and localization with sparse extended information filters," in *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, Eds., Nice, France, 2002.

[16] M. A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *International Joint Conference on Artificial Intelligence*, G. Gottlob and T. Walsh, Eds., San Francisco, CA, 2003, pp. 1157–1164.

[17] J. Leonard and H. Feder, "Decoupled stochastic mapping," *IEEE J. of Ocean Engineering*, vol. 26, pp. 561–571, Oct. 2001.

[18] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

[19] J. Neira and J. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 890–897, Dec 2001.

[20] J.-S. Gutmann and C. Schlegel, "Amos: comparison of scan matching approaches for self-localization in indoor environments," in *Proceedings of the First Euromicro Workshop on Advanced Mobile Robot*, G. Gottlob and T. Walsh, Eds., San Francisco, CA, Oct. 1996, pp. 61–67.