Decoupled Stochastic Mapping

John J. Leonard and Hans Jacob S. Feder

***Abstract*—This paper describes decoupled stochastic mapping (DSM), a new computationally efficient approach to large-scale concurrent mapping and localization (CML). DSM reduces the computational burden of conventional stochastic mapping by dividing the environment into multiple overlapping submap regions, each with its own stochastic map. Two new approximation techniques are utilized for transferring vehicle state information from one submap to another, yielding a constant-time algorithm whose memory requirements scale linearly with the number of submaps. The approach is demonstrated via simulations and experiments. Simulation results are presented for the case of an autonomous underwater vehicle (AUV) navigating in an unknown environments with 110 and 1200 features using simulated observations of point features by a forward look sonar. Empirical tests are used to examine the consistency of the error bounds calculated by the different methods. Experimental results are also presented for an environment with 93 features using sonar data obtained in a 3 by 9 by 1 m testing tank.**

***Index Terms*—Autonomous underwater vehicles, mapping, mobile robots, navigation.**

I. Introduction

THE objective of concurrent mapping and localization (CML) is to enable a mobile robot to build a map of an unknown environment while concurrently using that map to navigate. CML has been a central research topic in the field of mobile robotics due to its theoretical challenges and critical importance for many different types of robot applications [1], [5], [22], [24]. A key stumbling block in the development and implementation of new methods for CML has been the map scaling problem—the increase of computational complexity with the size of the operating environment of the mobile robot.

The primary cause of map-scaling complexity in CML is the requirement of representing correlations between the error estimates of the vehicle and all the features in the map [18]. The optimal algorithm that retains all correlations [24] encounters a $\mathcal{O}(n^2)$ computational burden [22], where n is the number of features. (In general the Kalman filter is $\mathcal{O}(n^3)$, but the necessary computations for CML can be reduced to $\mathcal{O}(n^2)$; because the features are assumed to be stationary, the plant model Jacobian matrix $\mathbf{F}$ is sparse [22].) Real-time performance becomes impossible for environments with more than a few hundred features. The correlations arise because the locations of neither the vehicle nor the features are ever known precisely. If a mobile robot uses an observation of an imprecisely known target to update its position, the resulting vehicle position estimate becomes correlated with the feature location estimate. Likewise, correlations are introduced if an observation taken from an imprecisely known position is used to update the location estimate of a geometric feature in the map. Previous work has shown that methods which neglect these correlations will fail [8].

An alternative is to attempt to bound the correlations. Uhlmann and colleagues introduced a tool for CML state estimation called covariance intersection (CI) that produces consistent error estimates in situations where correlation information between states is unknown [15], [27]. The methods for CML implemented with CI maintain consistent error bounds, but may not provide an effective solution because the error bounds can be too conservative.

Notable recent research which has examined the map-scaling complexity issue in CML includes work by Guivant and Nebot [14] and Davison [10], who have implemented strategies for optimization of the computations entailed in the $\mathcal{O}(n^2)$ full covariance solution. These authors have shown methods to reduce the number of times that the full $\mathcal{O}(n^2)$ update must be performed. Newman has analyzed the convergence properties of the covariance matrices in CML and developed techniques for the scaling problem based on the use of relative maps [23].

The idea behind our work is to split the map into multiple globally-referenced submaps, each with its own vehicle track, and to maintain all correlations within a submap. The motivation is to achieve good performance by computing multiple partial solutions in parallel, and to avoid the computational burden that is entailed by computing one complete solution. The key to accomplishing this is the development of appropriate techniques for transitioning from one submap to another.

Several previous researchers have used multiple local maps in a CML algorithm. Betgé-Brezetz *et al.* [4] used multiple local maps to isolate odometry errors. Bulata and Devy introduced a hybrid local map method for incremental construction of feature-based and topological models from laser range data [6]. Chong and Kleeman used multiple local maps to address the issue of map scaling for indoor mobile robots using a novel sonar array that can identify plane and corner features from a single vantage point [9]. Our work is different because all the submaps in our DSM approach are *globally-referenced*. This circumvents the potentially difficult and time-consuming problem of matching multiple local maps.

The motivating application for our work is to enable autonomous underwater vehicles (AUVs) to navigate in unstructured environments without relying on *a priori* maps or acoustic beacons. Navigation is essential for successful operation of underwater vehicles in a range of scientific,

Manuscript received March 1, 2001; revised July 12, 2001. This work was supported in part by National Science Foundation under Grant BES-9733040, in part by the MIT Sea Grant College Program under Grant NA86RG0074 (project RCM-3), in part by the Naval Undersea Warfare Center under Grant N66604-99-M-7235, and in part by the Norwegian Research Council under Grant 109338/410. This paper was presented in part at the Ninth International Symposium on Robotics Research, Salt Lake City, UT, October, 1999.

The authors are with the Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jleonard@mit.edu; hjfeder@alum.mit.edu).

Publisher Item Identifier S 0364-9059(01)09921-6.

commercial, and military applications. Accurate positioning information is vital for the safety of the vehicle and for the utility of the data it collects. The error growth rates of inertial and dead-reckoning systems available for AUVs are usually unacceptable. Surfacing for the global positioning system (GPS) resets is often impossible or undesirable, and the deployment of an acoustic transponder network is often expensive and impractical. Navigation algorithms based on existing maps have been proposed, but sufficiently accurate *a priori* maps are often unobtainable. The goal of our work is to provide methods by which an AUV can build a map of an unknown marine environment and use that map concurrently for navigation.

The stochastic mapping approach assumes that distinctive features of the environment can be extracted from sensor data and represented as points in an appropriate parameter space. Other types of representations are possible and have been employed with success in land and marine robot systems. For example, Thrun [25], [26] has demonstrated highly successful navigation of indoor mobile robots using a combination of grid-based [12] and topological [16] modeling. Our hypothesis is that in many marine environments, salient features can be found and reliably extracted using state-of-the-art processing techniques, enabling a feature-based approach to CML. Feature extraction is not addressed in this paper; our simulations and experiments assume that measurements of point features are provided. In related work, we have demonstrated CML via postprocessing of real data from a US Navy forward looking sonar, for a situation where well-defined point-like objects could be readily detected [17].

The structure of the paper is as follows. Section II introduces our notation and summarizes conventional full covariance stochastic mapping. Section III describes decoupled stochastic mapping (DSM), in which the environment is represented in terms of multiple overlapping submaps. Section IV presents two different approximation methods for transitioning between submaps, cross-map relocation and cross-map updating. These methods enable a constant-time algorithm without introducing complex data association requirements. Section V uses the cross-map vehicle relocation transition technique to describe single-pass DSM, which is a constant-time algorithm for large-scale CML which achieves temporal convergence but not spatial convergence. Section IV describes multipass DSM, which uses a combination of cross-map updating and cross-map relocation to achieve a method that is both spatially and temporally convergent. Sections VII and VIII analyze the performance of the two new methods for scenarios with 110 and 1200 features, respectively. Section IX illustrates the performance of multipass DSM for an experiment with real data from a gantry robot taking data from a testing tank. Finally, Section X discusses our conclusions.

II. Stochastic Mapping

The current state-of-the-art in feature-based approaches to CML is characterized by nearest-neighbor techniques for data association and use of the extended Kalman filter (EKF) for state estimation [7], [11], [22], [24]. From the titles of two seminal papers on CML by Smith, Self, and Cheeseman [24] and Moutarlier and Chatila [22], we refer to this class of feature-based methods for CML as stochastic mapping (SM). Stochastic mapping considers CML as a variable-dimension state estimation problem in which the size of the state space is increased or decreased as features are added or removed from the map. As the robot moves through its environment, it uses new sensor measurements to perform two basic operations: 1) adding new features to its state vector and 2) updating concurrently its estimate of its own state and the locations of previously observed features in the environment.

Stochastic mapping algorithms for CML use the EKF to recursively compute a state estimate $\hat{\mathbf{x}}[k] = [\hat{\mathbf{x}}_r[k]^{\mathrm{T}}\ \hat{\mathbf{x}}_f[k]^{\mathrm{T}}]^{\mathrm{T}}$ at each discrete time step k, where $\hat{\mathbf{x}}_r[k]^{\mathrm{T}}$ and $\hat{\mathbf{x}}_f[k]^{\mathrm{T}} = [\hat{\mathbf{x}}_1[k]^{\mathrm{T}} \ldots \hat{\mathbf{x}}_N[k]^{\mathrm{T}}]^{\mathrm{T}}$ are the estimated robot and feature locations, respectively. Based on assumptions about linearization and data association, this estimate is the approximate conditional mean of $p(\mathbf{x}[k]|\mathbf{Z}^k)$

$$\hat{\mathbf{x}}[k] \approx E\left[p\left(\mathbf{x}[k]|Z^k\right)\right]$$

where Z^k designates the set of all measurement obtained up through time k. Associated with this state vector is an estimated error covariance, $\mathbf{P}[k]$, which represents the errors in the robot and feature locations, and the cross-correlations between these states

$$\mathbf{P}[k] = \begin{bmatrix} \mathbf{P}_{rr}[k] & \mathbf{P}_{rf}[k] \\ \mathbf{P}_{fr}[k] & \mathbf{P}_{ff}[k] \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{rr}[k] & \mathbf{P}_{r1}[k] & \cdots & \mathbf{P}_{rN}[k] \\ \mathbf{P}_{1r}[k] & \mathbf{P}_{11}[k] & \cdots & \mathbf{P}_{1N}[k] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{Nr}[k] & \mathbf{P}_{N1}[k] & \cdots & \mathbf{P}_{NN}[k] \end{bmatrix}. \quad (1)$$

The estimate error covariance matrix $\mathbf{P}[k]$ is an approximation to $\mathcal{P}[k]$, the actual mean squared error of the estimate at time k, which is defined by

$$\mathcal{P}[k] = \mathrm{E}\left[\{\mathbf{x}[k] - \hat{\mathbf{x}}[k]\}\{\mathbf{x}[k] - \hat{\mathbf{x}}[k]\}^{\mathrm{T}}\right]. \quad (2)$$

The state-estimate covariance $\mathbf{P}[k]$ that is associated with the state estimate $\hat{\mathbf{x}}[k]$ will be consistent if the matrix $\mathbf{P}[k] - \mathcal{P}[k]$ is positive definite

$$\mathbf{P}[k] - \mathcal{P}[k] > \mathbf{0} \quad (3)$$

or equivalently

$$\begin{bmatrix} \mathbf{P}_{rr}[k] & \mathbf{P}_{rf}[k] \\ \mathbf{P}_{fr}[k] & \mathbf{P}_{ff}[k] \end{bmatrix} - \begin{bmatrix} \mathcal{P}_{rr}[k] & \mathcal{P}_{rf}[k] \\ \mathcal{P}_{fr}[k] & \mathcal{P}_{ff}[k] \end{bmatrix} > \mathbf{0}. \quad (4)$$

A theoretical proof of consistency is not possible for any stochastic mapping technique that uses nonlinear plant and/or measurement models, or in problems where data association uncertainty (spurious measurements and/or ambiguities in the correspondence of measurements) is present. All practical mobile-robot navigation and mapping applications are characterized by these difficulties, and so empirical consistency tests must be relied upon to assess the validity of a stochastic mapping solution. Consistency in this context means that "the state errors should be acceptable as zero mean and have magnitude

commensurate with the state covariance as yielded by the filter [3]". This can be ascertained using standard statistical tests, such as examination of the square of the normalized estimation error $\{\mathbf{x}[k]-\hat{\mathbf{x}}[k]\}^{\mathrm{T}}\mathbf{P}[k]^{-1}\{\mathbf{x}[k]-\hat{\mathbf{x}}[k]\}$, as illustrated in Section VIII below.

In our implementation, the AUV senses features in the environment through range and bearing measurements. Estimates of feature locations from different vantage points are fused together to build a map, and concurrently, the feature location estimates are used to update the robot position. We denote the vehicle's state by $\mathbf{x}_r = [x_r \quad y_r \quad \phi \quad v]^{\mathrm{T}}$ to represent the vehicle's east position, north position, heading, and speed, respectively. The state of feature i is represented by $\mathbf{x}_i = [x_i \quad y_i]^{\mathrm{T}}$. The dynamic model used in the algorithm simulates an AUV equipped with control surfaces and a single aft thruster for propulsion, moving at a nominal forward speed of 2.5 m/s. The control input $\mathbf{u}$ to the vehicle is given by a change in heading, $\delta\phi$, and speed, δv, of the vehicle to model changes in rudder angle and forward thrust, that is, $\mathbf{u}[k] = [\delta\phi \quad \delta v]^{\mathrm{T}}$. Thus, the dynamic model of the AUV, $\mathbf{f}(\,)$, is given by

$$\mathbf{x}[k+1] = \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k]) + \mathbf{d}_{\mathbf{x}}(\mathbf{u}[k]) \tag{5}$$

where $\mathbf{d}_{\mathbf{x}}(\mathbf{u}[k])$ is a white, Gaussian random process independent of $\mathbf{x}[0]$, with magnitude dependent on the control input $\mathbf{u}[k]$.

The observation model $\mathbf{h}(\,)$ for the system is given by

$$\mathbf{z}[k] = \mathbf{h}(\mathbf{x}[k]) + \mathbf{d}_{\mathbf{z}} \tag{6}$$

where $\mathbf{z}[k]$ is the observation vector of range and bearing measurements. The observation model, $\mathbf{h}(\,)$, defines the (nonlinear) coordinate transformation from state to observation coordinates. Data association is performed using a logic-based approach in innovation space. This approach incorporates sensor uncertainty as well as the uncertainty in the vehicle's state. That is, for all measurements that can potentially be associated with feature i, the innovation, $\mathbf{v}_i$, and the innovation matrix, $\mathbf{S}_i$, are constructed and the closest measurement within the gate defined by the Mahalanobis distance

$$\mathbf{v}_i^{\mathrm{T}}\mathbf{S}_i^{-1}\mathbf{v}_i \leq \gamma \tag{7}$$

is chosen as the most likely measurement of that feature. Under the assumption that the measurement model noise is Gaussian, (7) will have a χ^2-distribution. Thus, for a system of 2 degrees of freedom (the x and y positions of the feature), the true measurement of the feature, if detected, would fall within a $\gamma = 9$ gate with 99% probability.

New feature tracks are incorporated into the system state vector by use of a logic-based track initiator that is similar in spirit to [2], with the difference that the vehicle's position uncertainty is also taken into account. In this approach, all measurements that have not been associated with a track (feature) over the last N time steps are stored. At each time step, a search for clusters with more than $M < N$ measurements are performed. When such a cluster is found, a new track is initiated and the feature is integrated into the system state vector and system error covariance. A cluster is defined as one measurement from each time step that falls within the gate define by (7).

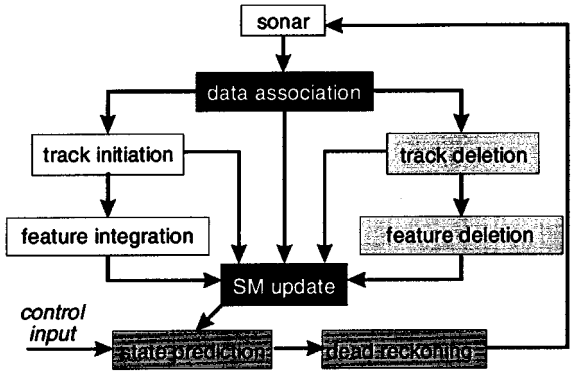

Fig. 1. Overview of stochastic mapping.

Once tracks are initiated, data association is performed by finding the single sonar return closest to each track [2]. It is assumed that a sonar return originates from not more than one feature. After the closest return to each feature is found, this return is gated with the estimated feature position. While the general problem of data association is exponentially complex, in this paper, we assume that nearest neighbor association strategies can be performed in constant time, and the primary source of computational complexity is from the requirement to represent the off-diagonal terms in the covariance matrix $\mathbf{P}$.

In some situations, it is useful to incorporate a track deletion capability as part of the data association process. In general, track deletion will be necessary for mapping in dynamic environments (where features might move or disappear [19]). For the current paper, we utilized a simple track deletion strategy that checks for consistency of the estimated features. That is, if the 95% error bounds for a feature estimate overlap by more than a certain percentage, (set at 30%), with another feature's 95% error bounds, the feature with the smallest error bound is deleted. This causes the system to remove tracks for features that are physically too close to each other for reliable data association. In general, track deletion must be performed with caution as good tracks may be erroneously discarded.

Fig. 1 shows a flow-chart representation of the structure of the augmented stochastic mapping algorithm. We will refer to the implementation of the stochastic mapping algorithm with the addition of logic-based track initiation and deletion and nearest-neighbor data association as augmented stochastic mapping (ASM).

III. Decoupled Mapping

The full covariance SM algorithm computes a single state estimate and associated covariance matrix which contain *all* features encountered thus far during a mission, encountering $\mathcal{O}(n^2)$ computational complexity. This makes it impossible to apply SM in situations with more than a few hundred features. In decoupled stochastic mapping, we compute multiple partial solutions, each consisting of a state vector for the vehicle and a subset of the features in the environment $\hat{\mathbf{x}}^i[k]$ and an associated estimate error covariance matrix $\mathbf{P}^i[k]$. For each solution, there is an associated submap region

$\mathcal{R}^i = \{x^i_{\min}, x^i_{\max}, y^i_{\min}, y^i_{\max}\}$. The number of feature estimates associated with submap is N^i. Each submap contains state estimates for only a subset of the N features of the environment. The submap regions are chosen such that the union of the submap regions completely cover the portion of the environment in which the vehicle travels. In this paper, the physical size of all submaps are the same, and their physical location is implicitly determined *a priori*. The size is set based on an assumed knowledge of the density of features, so that N^i is kept to a reasonable value and real-time performance is possible. More advanced, adaptive strategies for map location and sizing could be utilized. In the strategy employed here, a new submap is created and initialized when the vehicle moves out of the current submap. The submaps overlap slightly in order to prevent excessive map switching.

At any given time, one submap is the "active" submap, and the other submaps are "inactive". The active submap is the earliest created submap region that contains the current estimated vehicle position. New submaps are created when the vehicle exits the active submap and enters a region for which no submap exists. When the vehicle enters a region for which one or more submaps exist, the earliest created submap that contains the new vehicle position becomes the new active submap, and vehicle state estimate information is transitioned from the previously active submap to the newly active submap.

The concept of creating a new, globally referenced submap when the computational cost of the current map has become sufficiently high is relatively straightforward. But one challenging issue to consider is: can we reuse old submaps and thus be able to move between submaps in a manner such that consistent error bounds are maintained and temporal convergence is achieved? A second, more difficult issue is to consider spatial convergence: can the information gained from mapping one submap be passed on and exploited to achieve global reduction of errors?

In evaluation of the various possible strategies for transitioning between submaps, a variety of criteria need to be considered.

- **Computation time**: how does the computational burden scale with the number of features in the environment?
- **Theoretical consistency**: can the error covariances computed by the algorithm be shown mathematically to be consistent?
- **Empirical consistency**: do the error covariances computed by the algorithm pass tests of consistency based on the distribution of normalized squared errors [2] of the estimates?
- **Temporal convergence**: within a given submap region, do the error bounds converge, or slowly increase with time?
- **Spatial convergence**: do the error bounds obtained in different submaps converge to the same level, or do the errors increase for submaps that are further from the origin?
- **Data association complexity**: do the methods incur a high degree of data association complexity when transitions between submaps occur?

Based on these criteria, we have examined a wide range of transition strategies. In this paper, we next present two new approximation techniques, cross-map relocation and cross-map updating, that enable large-scale CML to be performed with constant-time algorithms. The next two sections describe these two techniques.

IV. Approximation Techniques for Transitioning Between Submaps

To illustrate the two-map transition strategies, let us assume that the vehicle is currently in submap A and is entering submap B, which already exists. Let $\hat{\mathbf{x}}^A_r[k]$ designate the submap A robot state estimate and $\hat{\mathbf{x}}^B_r[k]$ designate the submap B robot state estimate. Let us consider an environment which consists of two sets of features, designated by f and F. Let f designate the set of features that are mapped in submap A and let F designate the set of features that are mapped in submap B. In general, some of the features f mapped in submap A might also be contained in the set of features F that are contained in submap B. The state estimate of the vehicle and all the features of submap A are described by $\hat{\mathbf{x}}^A[k]$ and covariance by $\mathbf{P}^A[k]$, in analogy with (1)

$$\mathbf{P}^A[k] = \begin{bmatrix} \mathbf{P}^A_{rr}[k] & \mathbf{P}^A_{rf}[k] \\ \mathbf{P}^A_{fr}[k] & \mathbf{P}^A_{ff}[k] \end{bmatrix}. \tag{8}$$

Similarly, let $\hat{\mathbf{x}}^B[k]$ and $\mathbf{P}^B[k]$ define submap B

$$\mathbf{P}^B[k] = \begin{bmatrix} \mathbf{P}^B_{rr}[k] & \mathbf{P}^B_{rF}[k] \\ \mathbf{P}^B_{Fr}[k] & \mathbf{P}^B_{FF}[k] \end{bmatrix}. \tag{9}$$

Suppose that the robot leaves submap B at time step j. At this time, the state estimate for the robot is $\hat{\mathbf{x}}^B_r[j]$ and the robot state estimate covariance is $\mathbf{P}^B_{rr}[j]$. The state estimate for the features in submap B is $\hat{\mathbf{x}}^B_F[j]$ and the associated covariance matrix is $\mathbf{P}^B_{FF}[j]$. The joint state estimate, $\hat{\mathbf{x}}^B[j]$, for the robot and the set of features F is

$$\hat{\mathbf{x}}^B[j] = \begin{bmatrix} \hat{\mathbf{x}}^B_r[j] \\ \hat{\mathbf{x}}^B_F[j] \end{bmatrix} \tag{10}$$

and the associated joint covariance matrix is

$$\mathbf{P}^B[j] = \begin{bmatrix} \mathbf{P}^B_{rr}[j] & \mathbf{P}^B_{rF}[j] \\ \mathbf{P}^B_{Fr}[j] & \mathbf{P}^B_{FF}[j] \end{bmatrix}. \tag{11}$$

From time steps $j+1$ through $k-1$, no state estimates for submap B are computed. During this time, submap A is active. At time step k, where $k > j$, the robot transitions from submap A back into submap B. At this point, the state estimates for the robot and the feature locations in submap A are given by

$$\hat{\mathbf{x}}^A[k] = \begin{bmatrix} \hat{\mathbf{x}}^A_r[k] \\ \hat{\mathbf{x}}^A_f[k] \end{bmatrix} \tag{12}$$

and the associated joint-covariance matrix for submap A at time k is

$$\mathbf{P}^A[k] = \begin{bmatrix} \mathbf{P}^A_{rr}[k] & \mathbf{P}^A_{rf}[k] \\ \mathbf{P}^A_{fr}[k] & \mathbf{P}^A_{ff}[k] \end{bmatrix}. \tag{13}$$

A. Cross-Map Vehicle Relocation

The cross-map relocation step transitions the vehicle state information from submap A back into submap B as follows:

$$\hat{\mathbf{x}}^B[k] = \begin{bmatrix} \hat{\mathbf{x}}_r^A[k] \\ \hat{\mathbf{x}}_F^B[j] \end{bmatrix} \tag{14}$$

with associated covariance matrix

$$\begin{aligned} \mathbf{P}^B[k] &= \begin{bmatrix} \mathbf{P}_{rr}^B[k] & \mathbf{P}_{rF}^B[k] \\ \mathbf{P}_{Fr}^B[k] & \mathbf{P}_{FF}^B[k] \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{P}_{rr}^A[k] + \mathbf{P}_{rr}^B[j] & \mathbf{P}_{rF}^B[j] \\ \mathbf{P}_{Fr}^B[j] & \mathbf{P}_{FF}^B[j] \end{bmatrix}. \end{aligned} \tag{15}$$

The vehicle state estimate in submap B at time k is obtained by using the current vehicle state estimate from submap A and the feature state estimate from submap B from time step j. The current vehicle covariance from submap A is added to the vehicle covariance for submap B from time j, and the vehicle-to-feature correlation and feature covariance terms for submap B are left unchanged.

The intuitive appeal of this strategy is that, if the error bounds for the features in submap B were consistent at time j, and the features are static, then the feature covariance $\mathbf{P}_{FF}^B[k]$ in submap B at time k will also be consistent. If the error bounds for the vehicle in submap A at time k are consistent, then $\mathbf{P}_{rr}^B[k]$, the vehicle covariance for the vehicle in submap B, will also be consistent. However, the complete submap $\mathbf{P}^B[k]$ is consistent only under the assumption that the vehicle-to-feature correlation terms have not changed from time steps $j+1$ through $k-1$. Empirical consistency tests must be relied upon to verify the impact of this approximation. See [20] for a detailed discussion of this issue.

An alternative strategy that avoids this approximation about the cross-correlation terms is to perform state projection and dead-reckoning sensor measurement updates in submap B at each time step during the period when it is inactive. For the linear-Gaussian case with perfect data association, provably consistent error bounds can be obtained at the expense of either an $\mathcal{O}(n)$ algorithm or a great increase in data association complexity. In related research, we have examined a variety of nonapproximate alternative transition strategies, with encouraging results.

B. Cross-Map Vehicle Updating

The goal of cross-map updating is to bring more accurate vehicle estimates from submaps that were created earlier to submaps that were created later, to facilitate spatial convergence. (If each submap is given an integer ID based on the sequence in which it is initiated, then submaps that are created later will have a higher ID number than submaps that were created earlier. Hence we refer to submaps created earlier as "lower" submaps, and submaps created later as "higher" submaps.) It consists of two steps, (1) de-correlation (denoted by k^-) and (2) EKF updating (denoted by k^+). First, the vehicle state estimate for submap B is randomized, the vehicle covariance for submap B is greatly inflated, and the feature covariance for submap B is doubled

$$\begin{aligned} \hat{\mathbf{x}}^B[k^-] &\leftarrow \begin{bmatrix} \phi^B \\ \mathbf{x}_f^B[k] \end{bmatrix} \\ \mathbf{P}^B[k^-] &\leftarrow \begin{bmatrix} \mathbf{P}_{rr}^B[j] + \mathbf{\Phi}^B & \mathbf{P}_{rf}^B[j] \\ \mathbf{P}_{fr}^B[j] & 2\mathbf{P}_{ff}^B[j] \end{bmatrix} \end{aligned}$$

where $\boldsymbol{\phi}^B$ designates a random value uniformly distributed over the region defining submap B and $\mathbf{\Phi}^B$ designates a covariance much larger than the size of submap B. Second, the vehicle state estimate from submap A, $\hat{\mathbf{x}}_r^A[k]$, is used as a measurement $\mathbf{z}$, with covariance $\mathbf{P}_{rr}^A[k]$, in an EKF to update the vehicle position in submap B. This is summarized by the following equations:

$$\begin{aligned} \mathbf{K} &= \mathbf{P}^B[k^-]\mathbf{H}^{\mathrm{T}} \left(\mathbf{H}\mathbf{P}^B[k^-]\mathbf{H}^{\mathrm{T}} + \mathbf{P}_{rr}^A[k]\right)^{-1} \\ \hat{\mathbf{x}}^B[k^+] &\leftarrow \hat{\mathbf{x}}^B[k^-] + \mathbf{K}\left(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}^B[k^-]\right) \\ \mathbf{P}^B[k^+] &\leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^B[k^-](\mathbf{I} - \mathbf{K}\mathbf{H})^{\mathrm{T}} + \mathbf{K}\mathbf{P}_{rr}^A[k]\mathbf{K}^{\mathrm{T}} \end{aligned}$$

where $\mathbf{H}$ is the 4 by $(4+2N)$ matrix $[\mathbf{I} \quad \mathbf{0}]$.

To explain the intuition behind this strategy, it is useful to first consider several strategies which are unsuccessful for one reason or another. The first idea one might consider is to simply replace the vehicle state estimate for submap B with the state estimate from submap A, $\hat{\mathbf{x}}_r^B[k] \leftarrow \hat{\mathbf{x}}_r^A[k]$, and to replace the corresponding vehicle covariance submatrix, $\mathbf{P}_{rr}^B[k] \leftarrow \mathbf{P}_{rr}^A[k]$. The problem with this approach is that the submap B covariance matrix can no longer be guaranteed to be positive definite; such a replacement violates the physical meaning of a covariance matrix and thus violates the consistency of the submap.

A second idea that overcomes this would be to use the machinery of the EKF to perform the update—that is, to use the vehicle location estimate from submap A as a measurement of the vehicle location in an EKF that uses the vehicle covariance in submap A, $\mathbf{P}_{rr}^A[k]$, as the assumed measurement noise in the Kalman filter update equations for submap B. While this strategy maintains positive semi-definiteness, it violates the independence assumptions of the Kalman filter and produces overconfident estimates of both the vehicle locations and the features. Because the vehicle location estimates $\hat{\mathbf{x}}_r^A[k]$ and $\hat{\mathbf{x}}_r^B[k]$ are not independent, using $\hat{\mathbf{x}}_r^A[k]$ as a measurement of the vehicle position results in some information being used twice. In addition, due to the correlations between the vehicle and the features in submap B, performing a vehicle update measurement in this way also updates the locations of the feature states in submap B, and greatly reduces their error covariances, even though no new information about the locations of these features has been obtained.

To amend this, we can negate the effect of using vehicle location information twice by setting $\hat{\mathbf{x}}_r^B[k]$ to a completely random position within the submap and inflating the vehicle covariance $\mathbf{P}_{rr}^B[k]$ by a large amount. After the Kalman update, the resulting vehicle state estimate is effectively replaced by $\hat{\mathbf{x}}_r^B[k]$, its covariance becomes $\mathbf{P}_{rr}^A[k]$ and the correlation terms $\mathbf{P}_{rf}^B[k]$ between the vehicle and features are greatly reduced. The assumption is that the great inflation of the vehicle state estimate

before the update serves to "de-correlate" the vehicle estimate in submap B from its feature location estimates.

We are still left, however, with the fact that the map B feature states have been updated and the corresponding features covariances greatly reduced, despite the addition of no new information about their locations. This can be rectified by adding the *a priori* map B feature covariance $\mathbf{P}_{ff}^{B}$ to the *a posteriori* feature covariance after the Kalman update. The resulting covariance must be positive semi-definite and the updated feature covariance $\mathbf{P}_{ff}^{B}[k^{-}]$ can be shown to bound the a priori feature covariance $\mathbf{P}_{ff}^{B}[j]$. Because the Kalman gain is independent of $\mathbf{P}_{ff}^{B}$, it is equivalent to multiply $\mathbf{P}_{ff}^{B}$ by two before the Kalman update.

In this way, improved vehicle location information is transferred from submap A to submap B, and subsequent measurements of the features in submap B allow the accuracy of submap B to be improved. This technique is an approximation because, even though consistency can be demonstrated for the vehicle and feature covariances after the update, if they were consistent before, the consistency of $\mathbf{P}^{B}[k^{-}]$ cannot be guaranteed. In a similar manner to cross-map relocation, assumptions must be made about the cross-correlation terms. Again, empirical consistency tests will be vital to verify the performance of an implementation of the technique, as shown in our simulation results below.

Having two tools for passing vehicle state information between submaps, cross-map vehicle relocation and cross-map vehicle updating, we now move on to define two concurrent mapping and localization algorithms that utilize these methods. The first one is single-pass decoupled stochastic mapping, which is suitable for missions in which maximum accuracy is desired after only a single transit through the environment. The second method is labeled multipass decoupled stochastic mapping. Multipass DSM is more suitable when the vehicle will be able to make repeated passes through its environment, improving the accuracy of its map with each pass.

V. Single-Pass DSM Algorithm

The purpose of single-pass decoupled stochastic mapping is to achieve the highest accuracy possible based on one traversal of the environment. For this reason, new submaps are initialized with some of the feature state estimates for the previously active submap. This slows the growth of position uncertainty from one submap to another.

These features are called *correspondence features*. In the experimental results in this paper, the correspondence features are chosen based on their proximity to the new submap, to slow the growth of spatial errors. Because these features are known more accurately, the vehicle is able to produce a more accurate map than without the existence of these features. As a result, the uncertainty grows more slowly than if these features were not included when creating a submap.

To be more precise, let us assume that we are moving out of submap A and are about to create submap B. From submap A we have $\mathbf{P}^{A}[k]$ and $\hat{\mathbf{x}}^{A}[k]$, which for clarity, can be separated into the states and estimate for the vehicle, denoted by subscript r, correspondence features, denoted by subscript F, and the remaining features, denoted by subscript f. That is

$$\mathbf{P}^{A}[k]=\begin{bmatrix}\mathbf{P}_{rr}^{A}[k] & \mathbf{P}_{rF}^{A}[k] & \mathbf{P}_{rf}^{A}[k]\\ \mathbf{P}_{Fr}^{A}[k] & \mathbf{P}_{FF}^{A}[k] & \mathbf{P}_{Ff}^{A}[k]\\ \mathbf{P}_{fr}^{A}[k] & \mathbf{P}_{fF}^{A}[k] & \mathbf{P}_{ff}^{A}[k]\end{bmatrix}$$
$$\hat{\mathbf{x}}^{A}[k]=\begin{bmatrix}\hat{\mathbf{x}}_{r}^{A}[k]\\ \hat{\mathbf{x}}_{F}^{A}[k]\\ \hat{\mathbf{x}}_{f}^{A}[k]\end{bmatrix}. \tag{16}$$

Then, map B is created from $\mathbf{P}^{A}[k]$ and $\hat{\mathbf{x}}^{A}[k]$ in the following way:

$$\mathbf{P}^{B}[k]\leftarrow\begin{bmatrix}\mathbf{P}_{rr}^{A}[k] & \mathbf{P}_{rF}^{A}[k]\\ \mathbf{P}_{Fr}^{A}[k] & \mathbf{P}_{FF}^{A}[k]\end{bmatrix}\quad \hat{\mathbf{x}}^{B}[k]\leftarrow\begin{bmatrix}\hat{\mathbf{x}}_{r}^{A}[k]\\ \hat{\mathbf{x}}_{F}^{A}[k]\end{bmatrix}. \tag{17}$$

A block diagram of the single-pass decoupled stochastic mapping algorithm is shown in Fig. 2.

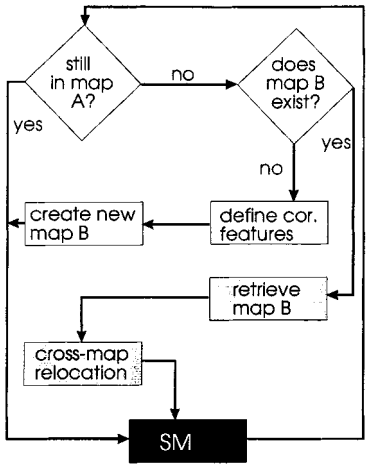

Fig. 2. Structure of the single-pass decoupled stochastic mapping algorithm. The stochastic mapping (SM) algorithm structure is outlined in Fig. 1.

VI. Multi-Pass DSM Algorithm

The purpose of multipass decoupled stochastic mapping is to have a method where the global error of a submap can be reduced by revisiting the submap. The reduction of global uncertainty of a submap is obtained by utilizing cross-map vehicle updating. Again, it should be noted that this can *only* be performed to a submap that was created *later* in time. For the instances when the vehicle travels back to a submap that was created earlier in time, cross-map vehicle relocation is employed just as in single-pass DSM. In order for cross-map vehicle updating to be applicable, correspondence features are not initialized. Thus, if the vehicle is moving out of submap A to create submap B, submap B is only created from the vehicle state estimate of submap A. That is, using the same notation as in (16)

$$\mathbf{P}_{rr}^{B}[k]\leftarrow\mathbf{P}_{rr}^{A}[k],\quad \hat{\mathbf{x}}_{r}^{B}[k]\leftarrow\hat{\mathbf{x}}_{r}^{A}[k]. \tag{18}$$

As there are no correspondence features used in multipass DSM, the feature states between submaps are independent in the sense that their estimates do not share any measurements. However, there will likely be features in adjacent maps that are

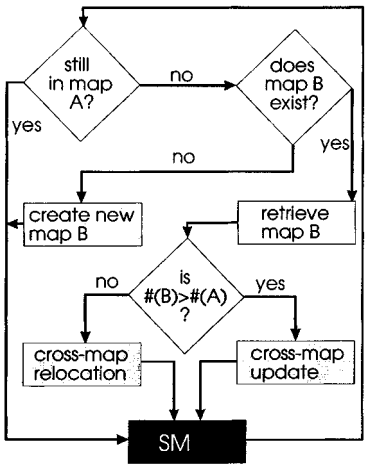

Fig. 3. Algorithm structure for the multipass DSM method. In contrast with single-pass DSM, two different map transition strategies are used. Cross-map vehicle relocation is used when transitioning from maps created later to maps created earlier, and cross-map vehicle updating is used when transitioning from maps created earlier to maps created later. Also, in multipass DSM, the submap regions do not overlap, although different submaps can possess estimates of the same environmental features that are independent (in the sense that they do not share any measurements).

represented in both submaps. It should be noticed that the feature states of a map will be (weakly) dependent on all earlier maps because a map is always initiated with the vehicle estimate and state from the earlier map upon creation of the later map. For this reason, cross-map vehicle updating can only be performed to a later map, and not the reverse. A block diagram of the multipass DSM algorithm is shown in Fig. 3.

VII. Experimental Comparison Between DSM and Full Covariance Stochastic Mapping

The simulation parameters are based on the characterization of an AUV equipped with a forward look electronically scanned sonar. The AUV is assumed to be given a differential GPS reset at the start of the mission, thus providing an initial position uncertainty of a few meters. In each scenario, false measurements are generated by assuming that the number of spurious returns has a poisson distribution with an expected value of λ. The range and angle of the spurious returns are uniformly distributed over the field of view of the sonar. The probability of detection of features is set to $P_D = 0.9$. Sonar measurements and dead reckoning measurements are obtained at 1 Hz. In these simulations, the physical boundaries of the submaps were set to be 525 by 525 meter squares. This size was chosen so that, given the feature density, the number of features in each submap would generally be less than 50. Further, they are placed so that the boundaries overlap by 25 m. This was done to prevent excess transitions between submaps when the vehicle moves along the boundary between two submaps. This also prevents the generation of excess submaps for missions in which the vehicle travels close to the boundaries of the survey area.

To compare each DSM method with the full covariance SM algorithm, simulations were performed for a scenario with 110 features randomly distributed over a 1– km by 1-km area, in the presence of clutter and dropouts. The desired path of the AUV and the true feature locations are shown in Fig. 4.

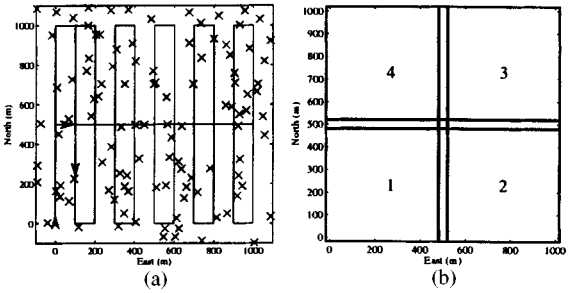

Fig. 4. (a) The desired survey path of the vehicle and the location of the 110 randomly distributed point features (crosses). The vehicle starts at $(0, 0)$ meters and follows the path of the arrows. (b) The submap partition of the survey area as generated by the two DSM algorithms.

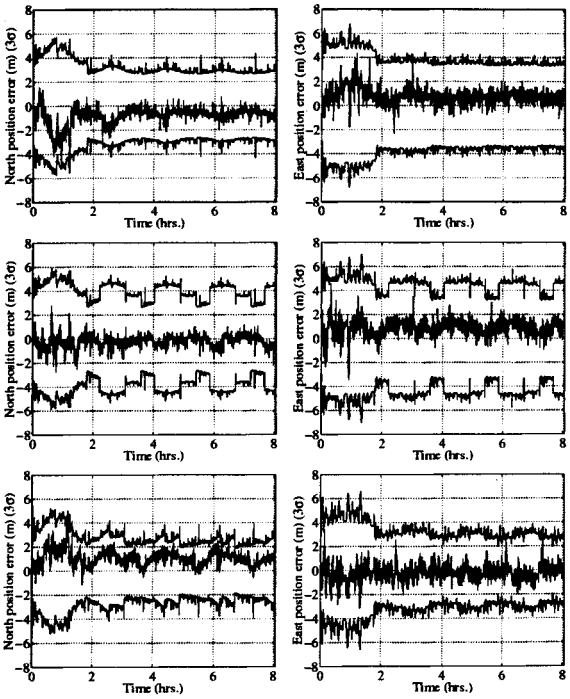

Fig. 5. Errors and 3σ bounds produced by full covariance SM (top), single-pass DSM (middle), and multipass DSM (bottom) surveys of the region shown in Fig. 4. The error is the difference between the true and the estimated vehicle position.

The top two plots of Fig. 5 show the position errors of the vehicle versus time and the 3σ error bounds for full covariance SM. On the first pass through the environment, the uncertainty grows as the vehicle is furthest from the origin, and then decreases when the vehicle returns close to the origin. On subsequent traversals, the error bounds are reduced.

For each of the two DSM algorithms, the survey area is partitioned into four submaps. Each submap bounds a 525 m by 525 m square region. Fig. 4 shows the location of the submaps as generated by the DSM algorithm. The numbers signify the order in which submaps were created.

The middle plots of Fig. 5 show the position errors for single-pass DSM. As with the full covariance algorithm, the position uncertainty of the vehicle grows as the distance from the starting point increases. Further, after the first pass through the survey path, the full covariance SM and the single-pass DSM results look very similar and achieve close to the same error bounds. The crucial difference between the methods

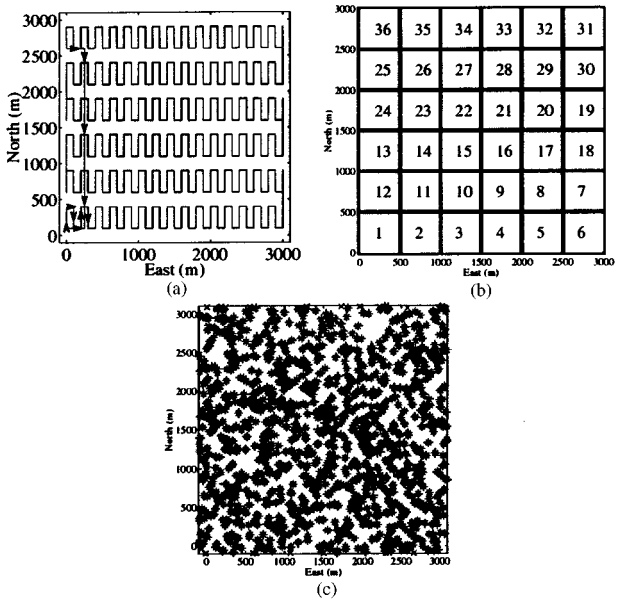

Fig. 6. (a) The desired survey path of the vehicle in the 3 km by 3 km survey area with 1200 features. (b) The partition of the survey area into 36 submaps. (c) The true feature positions (marked by '$\times$') and the estimated feature positions (marked by '$+$') and 3σ error ellipses for the long duration multipass DSM run.

is that SM estimates the correlations between all features, while single-pass DSM only estimates the correlations within submaps. The full covariance method is able to exploit all the correlations and thus reduce the global error at all locations. Single-pass DSM is unable to reduce the global uncertainty of submaps below the uncertainty upon creation of the submap. This can be seen from the "steps" in the north and east 3σ bounds after the completion of the first pass through the survey area (that is, after the first 2 h of the mission).

The bottom plots of Fig. 5 show the position errors of the vehicle versus time and the 3σ bounds for the survey performed by multipass DSM. The multipass errors resemble the results for full covariance SM more than the results from single-pass DSM. Clearly, the vehicle does better after the first pass through the survey area (that is, after about 2 hours) than before. Thus, the algorithm is capable of reducing the *global* error everywhere and not only locally in the submaps, as for single-pass DSM. However, one can see that the error bounds are slightly smaller than those of full covariance SM. This raises the concern that multipass DSM might become somewhat overconfident. This issue is discussed again below for an example with 36 submaps, where the consistency of the error bounds is improved when the path followed by the vehicle is varied.

VIII. Large-Scale Simulation Results

Next, we will demonstrate results using single-pass DSM and multipass DSM for surveying a large-scale environment with 1200 features for a mission duration of over 100 hours, sampling at a rate of 1 Hz. Fig. 6 shows the desired path of the AUV through the 3 km by 3 km survey area, the partition of the survey area into submaps, and true and estimated positions of the features in the survey area for the multipass DSM simulation shown in Fig. 6.

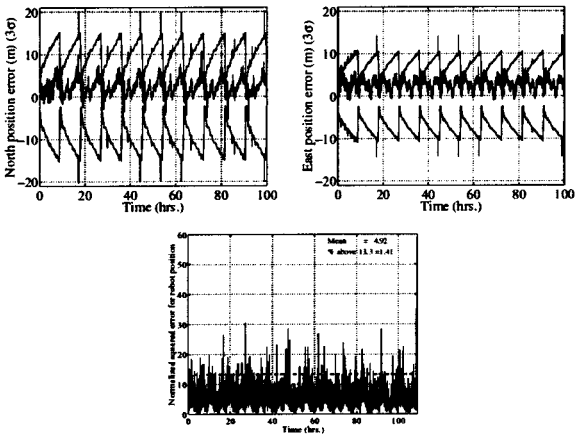

Fig. 7. Position errors, 3σ bounds (top), and normalized mean squared error (bottom) for a 36 submap single-pass DSM survey of the area given in Fig. 6. The horizontal dashed line drawn at the value 13.3 in the bottom plot indicates the 99% error level [3].

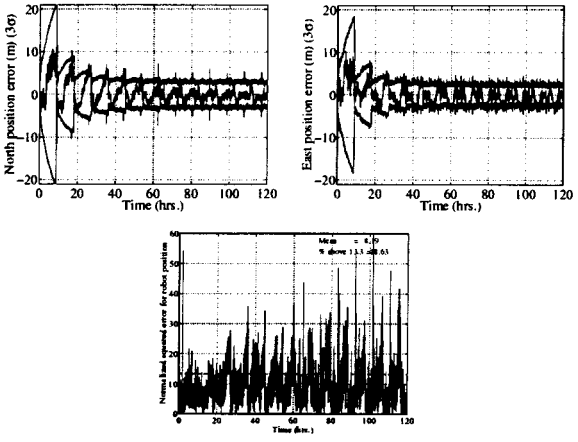

Fig. 8. Position errors, 3σ bounds (top), and normalized mean squared error (bottom) for a 36 submap multipass DSM survey of the area given in Fig. 6. The horizontal dashed line drawn at the value 13.3 in the bottom plot indicates the 99% error level [3].

Fig. 7 shows plots of the position errors of the vehicle versus time and the 3σ bounds when using single-pass DSM for the survey area of Fig. 6. In this simulation, the vehicle completed 11 laps of the survey path. The position uncertainty of each submap grows as a function of submap number.

Figs. 8 and 10 show the position errors of the vehicle versus time when using multipass DSM for two different survey paths. In Fig. 8, the vehicle follows the survey path indicated in Fig. 6, whereas in Fig. 10 the vehicle follows an alternating survey path that rotates the path given in Fig. 6 by 90 degrees after each complete circuit of the environment. This path is shown in Fig. 9.

The multipass DSM shows a considerable improvement over single-pass DSM in the long run as the survey area is revisited. However, during the first pass through the survey area, the maximum uncertainty when using multipass DSM is more than 30% higher than the result when using single-pass DSM. Single-pass DSM should be used when the survey area is to be traversed only once and multipass DSM should be used if one anticipates multiple traversals of the environment.

The normalized squared state errors [3] for the vehicle state estimates are also shown in Figs. 7, 8 and 10. The normalized

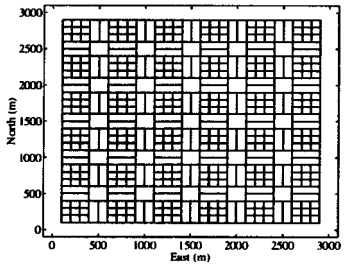

Fig. 9. Alternating survey path for second 36 submap multipass DSM survey. The path given in Fig. 6 is rotated by 90 degrees after each complete circuit of the environment. The first two cycles through the environment result in the following submap transition sequence: 1, 2, 3, . . ., 36, 25, 24, 13, 12, 1, 12, 13, 24, 25, 36, 35, 26, 23, 15, 11, 2, 3, 10, 15, 22, 27, 34, 33, 32, 29, 20, 17, 8, 5, 6, 7, 18, 19, 30, 31.

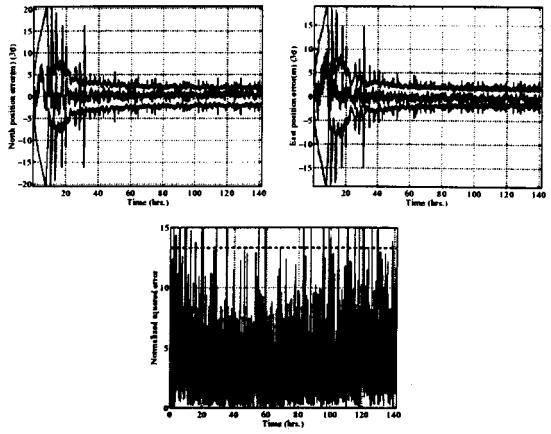

Fig. 10. Position errors, 3σ bounds (top), and normalized mean squared error (bottom) for a 36 submap multipass DSM survey of an environment with 1200 features with the same conditions as in Fig. 8, except using the alternating survey strategy shown in Fig. 9. With the alternating transition sequence, improved normalized squared errors are obtained.

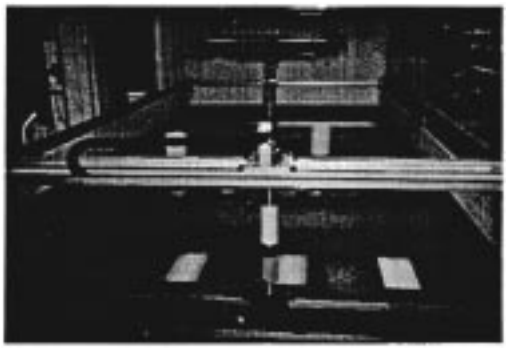

Fig. 11. The testing tank and robotic positioning system used in the underwater sonar experiments. 93 fishing bobbers were randomly positioned in the tank and used as features for testing the DSM algorithm.

squared errors are reasonably well-behaved for the single-pass DSM run. However, Fig. 8 indicates when the same repetitive survey path is used with multipass DSM, the amount of normalized squared errors falling outside the 99% error bounds is unacceptably high. However, this situation improves tremendously when the vehicle follows an alternating survey path, as illustrated by the mission shown in Fig. 10. When the vehicle is able to observe each feature from many different survey directions, the normalized errors are very well-behaved. Further research is

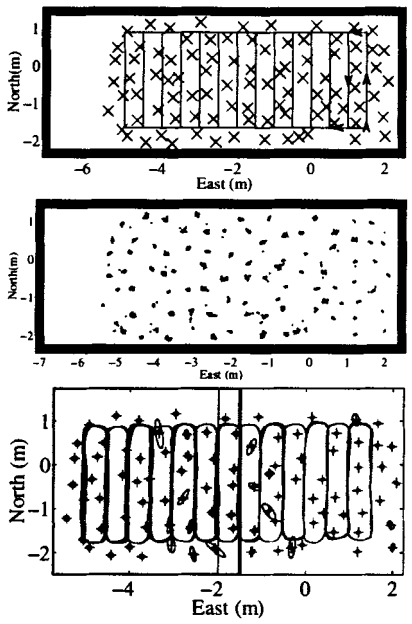

Fig. 12. *Top:* Desired path and the location of 93 features for the experiment. *Middle*: All sonar returns processed during the experiment, referenced to the true sensor location. (Returns originating from the tank walls were discarded.) *Bottom*: Actual path of the sensor and estimated feature locations with 3σ error ellipses.

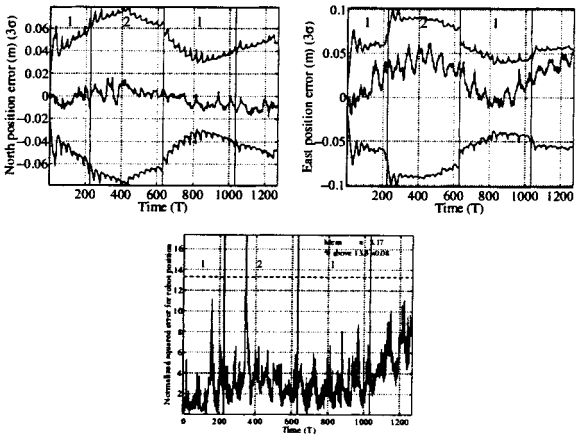

Fig. 13. Position errors, 3σ bounds (top) and normalized mean squared error (bottom) for the robot position for multipass DSM experiment in the testing tank.

necessary to characterize the effect of the vehicle trajectory on bias accumulation.

IX. Testing Tank Experiment

We now present a simple multipass DSM experiment for further investigation of the approach. The parameters for the mission were chosen so as to simulate an AUV scaled down by a factor of 100. A 500-kHz mechanically scanned sonar was mounted on a robotic positioning system and scanned over a $\pm 40°$ sector at each sensing location. Each scan took approximately 2 min. The testing tank is shown in Fig. 11.

In the experiment, 93 fishing bobbers were used as features and were randomly placed in the testing tank as shown by the crosses in Fig. 12. The sonar returns from the tank walls were

discarded by time gating. The sonar trajectory was set to perform a lawn-mower path starting at the lower right corner of the tank and moving toward the left. The estimated result from the DSM algorithm was compared to the true position of the sonar as obtained from position encoders on the robotic positioning system. The entire mission lasted about 1250 time steps. The resulting position estimate errors and the normalized squared state errors for the experiment are shown in Fig. 13.

X. Conclusion

This paper has presented a new, computationally efficient method for large-scale CML and demonstrated its performance through simulations and experiments. The single-pass and multipass DSM algorithms yield performance that is comparable to full covariance stochastic mapping, while maintaining constant computational requirements. Further work is necessary to explore the limits of the approximations employed by the cross-map relocation and cross-map updating submap transition strategies. For example, the normalized squared errors of multipass DSM are too large when a repetitive survey path is followed (Fig. 8), but are quite good when an alternating survey path is followed (Fig. 10). The results are encouraging, however, because they demonstrate successful stochastic mapping on a scale an order of magnitude larger (in terms of number of features) than any results previously published using a stochastic mapping approach. Using either method, the CPU and memory requirements are independent of the size of the map. The hard disk storage requirements of the method scale as $\mathcal{O}(mn)$ where m is the number of features states per submap and n is the total number of features.

Of the alternatives to the EKF that have been applied to CML, most are not computationally feasible for large-scale environments. A variety of techniques have been developed which employ numerical techniques to compute $p(\mathbf{x}[k]|\mathbf{Z}^k)$, the probability density function of the actual state conditioned on all the measurements obtained up through time k. Such methods can entail prohibitive computational requirements for moderately sized problems. The current state-of-the-art in this type of approach is represented by Thrun's work on real-time CML using particle filtering [25] using laser scanning data on land mobile robots. Julier and Uhlmann have used Covariance Intersection to perform CML with over 100 000 features [15]. CI, however, can be overly conservative. In the future, it would be interesting to perform a side-by-side experimental comparison of CI and DSM.

Any method based on Kilman filtering can be expected to encounter difficulties when angular errors grow very large, causing difficulties with linearization. We expect that the DSM technique should perform well in situations with small angular errors and a relatively high density of features (at least a few features visible at any given time). For situations with very poor proprioceptive sensors and a sparse environment, large angular errors would accumulate and it would be very difficult to maintain consistent, globally referenced error bounds. In this case, an approach using multiple local maps might be the only option. Empirical testing with careful ground-truth information will be necessary to define the limits of the approach for a given type of vehicle and sensor suite. Gibbens *et al.* [13] have presented a closed-form solution for the single-degree of freedom CML problem with linear models and observation of every feature at every time step. This solution can provide insights into some of the convergence properties of CML for a given combination of vehicle, sensors, and environment.

It is anticipated that the general strategy of temporal and spatial partitioning can be useful in the investigation of more complex issues of the concurrent mapping and localization problem, such as map maintenance in dynamic environments, improved resolution of data association ambiguity, relocation, recovery from errors, and cooperative mapping and navigation by teams of mobile robots.

The problem of reliably extracting features from sensor data remains an important outstanding issue for the realization of CML in more complex and natural environments. In related research, we have developed a technique called delayed decision making that can be useful for performing CML with composite features and in situations with high data association ambiguity [21]. The results in this paper considered only point features. Some interesting issues arise when considering long linear features (such as corridors or trenches) that would span multiple submaps. In our approach, features that are near the boundaries between submaps get mapped multiple times, in different submaps. In general, the question of how to combine the information across different submaps, while maintaining consistent error bounds, remains open.

References

[1] N. Ayache and O. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 804–819, Dec. 1989.

[2] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York: Academic, 1988.

[3] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Norwood, MA: Artech House, 1993.

[4] S. Betgé-Brezetz, P. Hébert, R. Chatila, and M. Devy, "Uncertain map making in natural environments," in *Proc. IEEE Int. Conf. Robotics and Automation*, Apr. 1996, pp. 1048–1053.

[5] R. A. Brooks, "Aspects of mobile robot visual map making," in *Proc. 2nd Int. Symp. Robotics Research*, Tokyo, Japan, 1984, pp. 287–293.

[6] H. Bulata and M. Devy, "Incremental construction of landmark-based and topological model of indoor environments by a mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, Apr. 1996, pp. 1054–1060.

[7] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardos, "The SPmap: A probabilistic framework for simultaneous localization and map building," *IEEE Trans. Robot. Automat.*, vol. 15, no. 5, pp. 948–952, 1999.

[8] J. A. Castellanos, J. D. Tardos, and G. Schmidt, "Building a global map of the environment of a mobile robot: The importance of correlations," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1997, pp. 1053–1059.

[9] K. S. Chong and L. Kleeman, "Sonar Based Map Building in Large Indoor Environments," Department of Electrical and Computer Systems Engineering, Monash University, Victoria, Australia, Tech. Rep. MECSE-1997-1, 1997.

[10] A. J. Davison, "Mobile Robot Navigation Using Active Vision," Ph.D., Dept. of Eng. Sci., Univ. of Oxford, 1998.

[11] M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," University of Sydney, Sydney, Australia, Tech. Rep. ACFR-TR-01-99, Mar. 1999.

[12] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. Robot. Automat.*, vol. RA-3, no. 3, pp. 249–265, June 1987.

[13] P. W. Gibbens, M. W. M. G. Dissanayake, and H. F. Durrant-Whyte, "A closed form solution to the single degree of freedom simultaneous localization and the map building (slam) problem," in *PROC. IEEE Int. Conf. Decision and Control (CDC)*, Sydney, Australia, Dec. 2000, pp. 191–196.

[14] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implementation," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 242–257, June 2001.

[15] S. J. Julier and J. K. Uhlmann, "Real time distributed map building in large environments," in *Sensor Fusion and Decentralized Control in Robotic Systems III, Photonics EAST*. Bellingham, WA: SPIE, 2000.

[16] B. J. Kuipers and Y. Byun, "A robot exploration and mapping strategy based on a semantic heirarchy of spatial representations," *J. Robot. Autonom. Syst.*, vol. 8, pp. 47–63, 1991.

[17] J. J. Leonard, R. N. Carpenter, and H. J. S. Feder, "Stochastic mapping using forward look sonar," in *Proc. Int. Conf. Field and Service Robotics*, Pittsburgh, PA, Aug. 1999, pp. 69–74.

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

[19] ——, *Directed Sonar Sensing for Mobile Robot Navigation*. Boston, MA: Kluwer, 1992.

[20] J. J. Leonard and H. J. S. Feder. (1999) Decoupled Stochastic Mapping, Tech. Rep. 99-1. Marine Robotics Laboratory, Massachusetts Institute of Technology, Cambridge, MA. [Online] http://oe.mit.edu/~jleonard/pubs/tr9901.ps

[21] J. J. Leonard and R. Rikoski, "Incorporation of delayed decision making into stochastic mapping," in *Experimental Robotics VII*, D. Rus and S. Singh, Eds. New York: Springer-Verlag, 2001, Lecture Notes in Control and Information Sciences.

[22] P. Moutarlier and R. Chatila, "Stochastic multisensory data fusion for mobile robot location and environment modeling," in *Proc. 5th Int. Symposium on Robotics Research*, Tokyo, 1989, pp. 207–216.

[23] P. M. Newman, "On the Structure and Solution of the Simultaneous Localization and Mapping Problem," Ph.D., Australian Center for Field Robotics,Univ. of Sydney, 1999.

[24] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proc. 4th Int. Symp. Robotics Research*, 1987.

[25] S. Thrun, "An online mapping algorithm for teams of mobile robots," Int. J. Robotics Res., vol. 20, no. 5, pp. 335–363, 2001, to be published.

[26] S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Machine Learning*, vol. 31, pp. 29–53, 1998.

[27] J. K. Uhlmann, S. J. Julier, and M. Csorba, "Nondivergent simultaneous map building and localization using covariance intersection," *Nav. Control Technol. Unmanned Syst. II*, 1997.

John J. Leonard received the B.S.E. degree in electrical engineering and science,and the D.Phil. degree in engineering science, from the University of Pennsylvania, Philadelphia, PA, and the University of Oxford, Oxford, U.K. , in 1987, and 1994, respectively.

He is an Associate Professor of Ocean Engineering at Massachusetts Insitute of Technology (MIT), Cambridge, MA. He has been a member of the MIT faculty since 1996. From 1991 to 1996, he was a Postdoctoral Fellow and Research Engineer at the MIT Sea Grant College Program and a member of the Autonomous Underwater Vehicles Laboratory. His research addresses the problem of sensor data fusion in marine robotics, with emphasis on the problems of sonar perception and navigation for autonomous underwater vehicles.

Dr. Leonard is an Associate Editor of the IEEE JOURNAL OF OCEANIC ENGINEERING and the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION.

Hans Jacob S. Feder received the of S.B. degree in mechanical engineering, and the Ph.D. degree in mechanical engineering, both from Massachusetts Insitute of Technology (MIT), Cambridge, in 1995, and 1999, respectively. His research is in the area of estimation theory with application to the problem of navigation for autonomous underwater vehicles.