Incorporation of Delayed Decision Making into Stochastic Mapping

John J. Leonard and Richard J. Rikoski
MIT Dept. of Ocean Engineering
Cambridge, MA 02139, U.S.A
jleonard@mit.edu and rikoski@mit.edu

Abstract: This paper presents a technique for incorporating delayed decision making into stochastic mapping algorithms for concurrent mapping and localization. The approach explicitly tracks the error correlations between current and previous vehicle states, enabling the initialization of map features using data from multiple time steps and improved data association decision-making. The method is illustrated using data from a ring of Polaroid sonar sensors from a B21 mobile robot, demonstrating the ability to perform CML with sparse and ambiguous data.

1. Introduction

The problem of concurrent mapping and localization (CML) for an autonomous mobile robot is stated as follows: starting from a initial position, a mobile robot travels through a sequence of positions and obtains a set of sensor measurements at each position. The goal is for the mobile robot to process the sensor data to produce an estimate of its position while concurrently building a map of the environment. This paper presents a new technique for performing CML in situations where individual measurements provide weak geometric constraints, making it necessary to initialize new map features using data from multiple vantage points. We apply the method in experiments using a B21 mobile robot and demonstrate accurate mapping of a simple geometric environment using data from a ring of Polaroid sonar sensors.

CML has been a central research topic in the robotics community, due to its theoretical challenges and critical importance for many mobile robot applications [1, 2, 3, 4]. In just the past few years, the research community has made tremendous strides towards the solution of this problem [5, 6, 7, 8], however there are still several critical open issues for future research. 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 [9, 3, 8, 10, 11]. From the titles of two seminal papers on CML by Smith, Self, and Cheeseman [9] and Moutarlier and Chatila [3], we refer to this class of feature-based methods for CML as “stochastic mapping”. 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 differs from other applications of the EKF because the size of the state vector is modified adaptively as new features are added and removed from the map. Initially, no feature locations are known and the state vector is restricted to contain only the initial state of the robot.

Previously published work in stochastic mapping has effectively made two assumptions: (1) there is sufficient information in the set of measurements available from a single robot position to completely and consistently initialize a new feature into the map [9, 3, 8, 10, 11] and (2) a nearest-neighbor gating strategy is sufficient for determination of the correspondence between measurements and features. These two assumptions have limited the robustness of current CML implementations and have restricted the range of environments in which they can be applied.

In this paper, we describe a new extension to the state-of-the-art in CML by presenting a technique for delayed decision making that allows us to perform CML with sparse and noisy sonar data. The key innovations of the current paper are to add past vehicle positions to the state vector and to maintain explicitly estimates of the correlations between current and previous vehicle states. By incorporating past vehicle locations in the state vector, we are able to make improved data association and feature classification decisions and to initialize new map features by consistently combining data from multiple vantage points.

The motivation for the new approach is the following: if the sensor observations available from a single time step do not provide sufficient information to initialize the state estimate of a newly detected feature, then information from multiple vehicle positions must used. To maintain consistent error bounds, correlations between different vehicle locations must be taken into account by the CML algorithm. Further, decisions that are difficult based on the data from a single position (such as the disposition of an individual sonar return) can be made much easier when considered as delayed decisions, using data from multiple vehicle positions.

The structure of this paper is as follows: Section 2 reviews the current state-of-the-art for this type of algorithm, the following section describes our new approach in detail, and in Section 4 experimental results are presented.

2. Stochastic Mapping

Stochastic mapping is a feature-based concurrent mapping and localization algorithm that was first published by Smith, Self, and Cheeseman [9] and Moutarlier and Chatila [3]. The method assumes that there are n features in the environment, and that they are static. The true state at time k is designated by $\mathbf{x}[k] = [\mathbf{x}_r[k]^T \ \mathbf{x}_f[k]^T]^T$, where $\mathbf{x}_r[k]$ represents the location of the robot, and $\mathbf{x}_f[k]^T = [\mathbf{x}_{f_1}[k]^T \ldots \mathbf{x}_{f_n}[k]^T]^T$ represent the locations of the environmental features. Let $\mathbf{z}[k]$ designate the sensor measurements obtained at time k, and Z^k designate the set of all measurements obtained from time 0 through time k.

Stochastic mapping algorithms for CML use the extended Kalman filter to compute recursively a state estimate $\hat{\mathbf{x}}[k|k] = [\hat{\mathbf{x}}_r[k|k]^T \ \hat{\mathbf{x}}_f[k]^T]^T$ at each discrete time step k, where $\hat{\mathbf{x}}_r[k|k]^T$ and $\hat{\mathbf{x}}_f[k]^T = [\hat{\mathbf{x}}_{f_1}[k]^T \ldots \hat{\mathbf{x}}_{f_n}[k]^T]^T$ are the robot and feature state estimates, respectively. Based on assumptions about linearization and data association, this estimate is the approximate conditional mean of $p(\mathbf{x}[k]|Z^k)$:

$$\hat{\mathbf{x}}[k|k] \approx E[p(\mathbf{x}[k]|Z^k)] \tag{1}$$

Associated with this state vector is an estimated error covariance, $\mathbf{P}[k|k]$, which represents the errors in the robot and feature locations, and the cross-correlations between these states:

$$\mathbf{P}[k|k] = \begin{bmatrix} \mathbf{P}_{rr}[k|k] & \mathbf{P}_{rf}[k|k] \\ \mathbf{P}_{fr}[k|k] & \mathbf{P}_{ff}[k|k] \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{rr}[k|k] & \mathbf{P}_{rf_1}[k|k] & \cdots & \mathbf{P}_{rf_n}[k|k] \\ \mathbf{P}_{f_1 r}[k|k] & \mathbf{P}_{f_1 f_1}[k|k] & \cdots & \mathbf{P}_{f_1 n}[k|k] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{f_n r}[k|k] & \mathbf{P}_{f_n f_1}[k|k] & \cdots & \mathbf{P}_{f_n f_n}[k|k] \end{bmatrix}. \tag{2}$$

The method uses two models, a plant model and a measurement model. The plant model is used to make predictions of future vehicle positions based on a control input. In a two-dimensional implementation, a typical state model for the robot might be: $\mathbf{x}_r = [x_r \ y_r \ \phi \ v]^T$, representing the vehicle's east position, north position, heading, and speed, respectively. The simplest type of features are points, with feature f_i represented by $\mathbf{x}_{f_i} = [x_i \ y_i]^T$. The dynamic model for the motion of the robot is given by

$$\mathbf{x}[k+1] = \mathbf{f}(\mathbf{x}[k], \mathbf{u}[k]) + \mathbf{d}_{\mathbf{x}}(\mathbf{u}[k]), \tag{3}$$

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

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

$$\mathbf{z}[k] = \mathbf{h}(\mathbf{x}[k]) + \mathbf{d}_{\mathbf{z}}, \tag{4}$$

where $\mathbf{z}[k]$ is the vector of sensor measurements (e.g., range and bearing when using sonar). The observation model, $\mathbf{h}(\cdot)$, defines the nonlinear coordinate transformation from state to observation coordinates. The stochastic process $\mathbf{d}_{\mathbf{z}}$, is assumed to be white, Gaussian, and independent of $\hat{\mathbf{x}}[0]$ and $\mathbf{d}_{\mathbf{x}}$, and has covariance $\mathbf{R}$. Given these assumptions, an extended Kalman filter (EKF) is employed to estimate the state $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$ given the measurements.

Data association is the process of determining the origin of sensor measurements. A decision must be made for each new measurement to determine if (1) it originates from one of the features currently in the map, (2) it originates from a new feature, or (3) it is spurious. In general, the data association problem is exponentially complex [12], and no general solution that can run in real-time has been published. Most published implementations of CML have used variations of "nearest-neighbor" gating techniques. For each feature in the vehicle state vector, predicted range and angle measurements are generated and are compared against the actual measurements using a weighted statistical distance in measurement space. For all measurements $\mathbf{z}_j[k]$ that can potentially be associated with feature $\hat{\mathbf{x}}_{f_i}[k]$, the innovation, $\boldsymbol{\nu}_{ij}[k]$, and the innovation covariance, $\mathbf{S}_{ij}[k]$, are constructed and the closest measurement within the gate defined by the Mahalanobis distance

$$\boldsymbol{\nu}_{ij}[k]^T \mathbf{S}_{ij}[k]^{-1} \boldsymbol{\nu}_{ij}[k] \leq \gamma, \tag{5}$$

is considered the most likely measurement of that feature [12].

Measurements that do not gate with any existing feature become candidates for the initialization of new features. Previously published methods have assumed that the state of the new feature, $\hat{\mathbf{x}}_{f_{n+1}}[k]$ can be computed using the measurement data available from a single vehicle position, using a feature initialization function $\mathbf{g}(\cdot)$:

$$\hat{\mathbf{x}}_{f_{n+1}}[k] = \mathbf{g}(\hat{\mathbf{x}}[k|k], \mathbf{z}_j[k]). \tag{6}$$

For example, for a sensor providing range and bearing measurements, $\mathbf{z}_j[k] = [r\ \theta]$, the feature initialization function for a point $\mathbf{g}(\cdot)$ takes the following form:

$$\hat{\mathbf{x}}_{f_{n+1}}[k] = \mathbf{g}(\hat{\mathbf{x}}[k|k], \mathbf{z}_j[k]) = \begin{bmatrix} x_r + r\cos(\phi+\theta) \\ y_r + r\sin(\phi+\theta) \end{bmatrix}. \tag{7}$$

The new feature is integrated into the map by expanding the state vector $\hat{\mathbf{x}}[k|k]$ and covariance $\mathbf{P}[k|k]$ as shown below:

$$\hat{\mathbf{x}}[k|k] \leftarrow \begin{bmatrix} \hat{\mathbf{x}}[k|k] \\ \hat{\mathbf{x}}_{f_{n+1}}[k] \end{bmatrix}, \tag{8}$$

$$\mathbf{P}[k|k] \leftarrow \begin{bmatrix} \mathbf{P}_{rr}[k|k] & \mathbf{P}_{rf}[k|k] & \mathbf{P}_{rf_{n+1}}[k|k] \\ \mathbf{P}_{fr}[k|k] & \mathbf{P}_{ff}[k|k] & \mathbf{P}_{ff_{n+1}}[k|k] \\ \mathbf{P}_{f_{n+1}r}[k|k] & \mathbf{P}_{f_{n+1}f}[k|k] & \mathbf{P}_{f_{n+1}f_{n+1}}[k|k] \end{bmatrix}, \tag{9}$$

where

$$\mathbf{P}_{f_{n+1}f_{n+1}}[k|k] = \mathbf{G_x}\mathbf{P}[k|k]\mathbf{G_x}^T + \mathbf{G_z}\mathbf{R}[k]\mathbf{G_z}^T, \tag{10}$$

$$\begin{bmatrix} \mathbf{P}_{f_{n+1}r}[k|k] & \mathbf{P}_{f_{n+1}f}[k|k] \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{f_{n+1}r}[k|k] \\ \mathbf{P}_{f_{n+1}f}[k|k] \end{bmatrix}^T = \mathbf{G_x}\mathbf{P}[k|k], \tag{11}$$

$\mathbf{G_x}$ is the Jacobian of $\mathbf{g}$ with respect to the state vector and $\mathbf{G_z}$ is the Jacobian of $\mathbf{g}$ with respect to the measurement.

3. Incorporation of Delayed Decision Making

The key idea of our new approach is to expand the representation to add a number of previous vehicle locations to the state vector. We refer to these states as trajectory states. Each time the vehicle moves, the previous vehicle location is added to the state vector. We introduce the notation $\hat{\mathbf{x}}_{t_i}[k]$ to refer to the estimate of the state (position) of the robot at time i given all information up to time k. The complete trajectory of the robot for time step 0 through time step $k-1$ is given by the vector

$\hat{\mathbf{x}}_t[k] = [\hat{\mathbf{x}}_{t_0}[k]^T \ \hat{\mathbf{x}}_{t_1}[k]^T \ \hat{\mathbf{x}}_{t_2}[k]^T \ \ldots \ \hat{\mathbf{x}}_{t_{k-1}}[k]]^T$. The complete state vector is:

$$\hat{\mathbf{x}}[k|k] = \begin{bmatrix} \hat{\mathbf{x}}_r[k|k] \\ \hat{\mathbf{x}}_t[k] \\ \hat{\mathbf{x}}_f[k] \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_r[k|k] \\ \hat{\mathbf{x}}_{t_0}[k] \\ \hat{\mathbf{x}}_{t_1}[k] \\ \hat{\mathbf{x}}_{t_2}[k] \\ \vdots \\ \hat{\mathbf{x}}_{t_{k-1}}[k] \\ \hat{\mathbf{x}}_{f_1}[k] \\ \hat{\mathbf{x}}_{f_2}[k] \\ \hat{\mathbf{x}}_{f_3}[k] \\ \vdots \\ \hat{\mathbf{x}}_{f_{n-1}}[k] \\ \hat{\mathbf{x}}_{f_n}[k] \end{bmatrix}. \tag{12}$$

The associated covariance matrix is:

$$\mathbf{P}[k|k] = \begin{bmatrix} \mathbf{P}_{rr}[k|k] & \mathbf{P}_{rt}[k|k] & \mathbf{P}_{rf}[k|k] \\ \mathbf{P}_{tr}[k|k] & \mathbf{P}_{tt}[k|k] & \mathbf{P}_{tf}[k|k] \\ \mathbf{P}_{fr}[k|k] & \mathbf{P}_{ft}[k|k] & \mathbf{P}_{ff}[k|k] \end{bmatrix}, \tag{13}$$

or equivalently,

$$\mathbf{P}[k|k] = \begin{bmatrix} \mathbf{P}_{rr}[k|k] & \mathbf{P}_{rt_0}[k|k] & \ldots & \mathbf{P}_{rt_{k-1}}[k|k] & \mathbf{P}_{rf_1}[k|k] & \ldots & \mathbf{P}_{rf_n}[k|k] \\ \mathbf{P}_{t_0r}[k|k] & \mathbf{P}_{t_0t_0}[k|k] & \ldots & \mathbf{P}_{t_0t_{k-1}}[k|k] & \mathbf{P}_{t_0f_1}[k|k] & \ldots & \mathbf{P}_{t_0f_n}[k|k] \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{t_{k-1}r}[k|k] & \mathbf{P}_{t_{k-1}t_0}[k|k] & \ldots & \mathbf{P}_{t_{k-1}t_{k-1}}[k|k] & \mathbf{P}_{t_{k-1}f_1}[k|k] & \ldots & \mathbf{P}_{t_{k-1}f_n}[k|k] \\ \mathbf{P}_{f_1r}[k|k] & \mathbf{P}_{f_1t_0}[k|k] & \ldots & \mathbf{P}_{f_1t_{k-1}}[k|k] & \mathbf{P}_{f_1f_1}[k|k] & \ldots & \mathbf{P}_{f_1f_n}[k|k] \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{f_nr}[k|k] & \mathbf{P}_{f_nt_0}[k|k] & \ldots & \mathbf{P}_{f_nt_{k-1}}[k|k] & \mathbf{P}_{f_nf_1}[k|k] & \ldots & \mathbf{P}_{f_nf_n}[k|k] \end{bmatrix}. \tag{14}$$

New trajectory states are added to the state vector each time step by defining a new trajectory state $\hat{\mathbf{x}}_{t_k}[k] = \hat{\mathbf{x}}_r[k|k]$ and adding this to the state vector:

$$\hat{\mathbf{x}}[k|k] \leftarrow \begin{bmatrix} \hat{\mathbf{x}}_r[k|k] \\ \hat{\mathbf{x}}_{t_0}[k] \\ \hat{\mathbf{x}}_{t_1}[k] \\ \hat{\mathbf{x}}_{t_2}[k] \\ \vdots \\ \hat{\mathbf{x}}_{t_{k-1}}[k] \\ \hat{\mathbf{x}}_{t_k}[k] \\ \hat{\mathbf{x}}_f[k] \end{bmatrix}. \tag{15}$$

The state covariance is expanded as follows:

$$\mathbf{P}[k|k] \leftarrow \begin{bmatrix} \mathbf{P}_{rr}[k|k] & \mathbf{P}_{rt_0}[k|k] & \ldots & \mathbf{P}_{rt_{k-1}}[k|k] & \mathbf{P}_{rt_k}[k|k] & \mathbf{P}_{rf}[k|k] \\ \mathbf{P}_{t_0 r}[k|k] & \mathbf{P}_{t_0 t_0}[k|k] & \ldots & \mathbf{P}_{t_0 t_{k-1}}[k|k] & \mathbf{P}_{t_0 t_k}[k|k] & \mathbf{P}_{t_0 f}[k|k] \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{P}_{t_{k-1} r}[k|k] & \mathbf{P}_{t_{k-1} t_0}[k|k] & \ldots & \mathbf{P}_{t_{k-1} t_{k-1}}[k|k] & \mathbf{P}_{t_{k-1} t_k}[k|k] & \mathbf{P}_{t_{k-1} f}[k|k] \\ \mathbf{P}_{t_k r}[k|k] & \mathbf{P}_{t_k t_0}[k|k] & \ldots & \mathbf{P}_{t_k t_{k-1}}[k|k] & \mathbf{P}_{t_k t_k}[k|k] & \mathbf{P}_{t_k f}[k|k] \\ \mathbf{P}_{fr}[k|k] & \mathbf{P}_{ft_0}[k|k] & \ldots & \mathbf{P}_{ft_{k-1}}[k|k] & \mathbf{P}_{ft_k}[k|k] & \mathbf{P}_{ff}[k|k] \end{bmatrix}, \tag{16}$$

where $\mathbf{P}_{t_k t_i}[k|k] = \mathbf{P}_{rt_i}[k|k]$, $\mathbf{P}_{t_k f}[k|k] = \mathbf{P}_{rf}[k|k]$, and $\mathbf{P}_{t_k t_k}[k|k] = \mathbf{P}_{rr}[k|k]$. The growth of the state vector in this manner increases the computational burden, however it is straightforward to delete old vehicle trajectory states and associated terms in the covariance, once all the measurements from a given time step have been either processed or discarded.

This process of adding past states is similar to a fixed-lag Kalman smoother [13]. In a fixed-lag smoother, states exceeding a certain age are automatically removed. In our approach, states are added and removed based on the data processing requirements of the stochastic mapping process. Unlike the fixed-lag smoother, states are not necessarily removed in the order in which they are added.

With the addition of prior vehicle states to the state vector, it now becomes possible to initialize new features using measurements from multiple time steps. For example, consider the initialization of a new feature using two measurements, $\mathbf{z}[k_1]$ and $\mathbf{z}[k_2]$, taken at time steps k_1 and k_2. The state of the new feature can be computed using a feature initialization function involving data from multiple time steps:

$$\hat{\mathbf{x}}_{f_{n+1}} = g(\hat{\mathbf{x}}_{t_{k_1}}[k], \hat{\mathbf{x}}_{t_{k_2}}[k], [\mathbf{z}[k_1]^T \ \mathbf{z}[k_2]^T]^T). \tag{17}$$

For example, in two-dimensions if each measurement is a range-only sonar measurement, then the function $\mathbf{g}(\cdot)$ represents a solution for the intersection of two circles. The covariance for the new feature is initialized in a similar fashion as shown above in Equations 9 to 11, except that the Jacobian matrix $\mathbf{G}_{\mathbf{x}}$ will contain additional non-zero terms corresponding to the trajectory states and the Jacobian matrix $\mathbf{G}_{\mathbf{z}}$. The procedure is the same if the feature initialization function $\mathbf{g}(\cdot)$ is a function of measurements from more than two time steps. New feature initialization can also be performed using non-linear least squares [14] performed on many measurements, instead of using an explicit function $\mathbf{g}(\cdot)$.

To provide improved stability, the addition of new features to the state vector can be delayed to occur only when the initializing Jacobians indicate that the new feature estimate is well-conditioned. By examining the different possible initialization sets and choosing the Jacobian with the smallest values, the most stable initialization can be determined. In addition, one can incorporate an adaptive motion control step to direct the robot to move to a better vantage point that will yield a more stable initialization. By considering second-order derivatives, the robot can determine the optimal direction to move in order to obtain data that will yield the most stable initialization of a new feature.

Once a new feature is initialized, the map can be updated using all other previously obtained measurements that can be associated with the new feature. We call this procedure a ”batch update”. It allows the maximum amount of information to be extracted from all past measurements.

The ability to perform a batch update using many previous measurements provides a facility for making delayed data association decisions. If there is ambiguity about the correspondence between measurements and features, decisions can be postponed until additional information becomes available. Feature extraction is also simplified. The initialization of complex features in situations with high ambiguity can be greatly simplified by considering a batch of data obtained at multiple time steps.

4. Experimental Results

To demonstrate the approach, we present the application of the method to ring sonar data from a B21 mobile robot, shown in Figure 1. Figures 2 to 4 show the results of an experiment performed in a simple 6-foot by 8-foot room made of plywood. The data association and feature modeling techniques used in this experiment utilized a priori knowledge of the structure of the box, namely that each corner of the box was created by two walls, and that each wall was bounded at each end by a corner. More generic and powerful “anytime” data association techniques are in development and should be applicable to more complex scenarios. These results are reported here only as a means of illustrating how delayed decision making and batch reprocessing can be used for CML in a situation where individual measurements are sparse and provide weak constraints.

The data processing for this experiment proceeded as follows. First, the vehicle dead-reckoned around the room for 50 timesteps, collecting the data shown in Figure 2(a), but not initializing any features. At each time step, a new vehicle trajectory state was added to the system state vector, using Equations 15 and 16. The dead-reckoned vehicle trajectory is shown in Figure 2(b). After fifty time steps, a search was performed on all the sonar returns acquired to find a large subset of returns originating from a single point in the room. Because of the structure of the box, this point can safely be assumed to be one of the corners of the box, since they are visible from the largest number of sensing locations. From this subset of returns, the pair of returns obtained that provided the most stable initialization of a new point feature was selected, and used to initialize a new point feature. This feature corresponded to the upper left corner of the box, and serves as the “seed” feature for reconstruction of the box.

Next, a search was performed to find sonar returns that were tangent to lines drawn from the initial corner, corresponding to the two walls that comprise the corner. The initialization for each wall was based on a single measurement and the constraint that the wall passes through the corner. Subsequently, searches were performed to find a sonar returns consistent with new point features that lay close to the new walls. The chosen sonar returns were used, together with the state estimate for the respective walls, to initialize new point features corresponding to the upper right corner and lower left corner of the box. This process was continued around the box until the bottom right corner was initialized without any measurements as the intersection of two walls. An additional point feature, a crack on one of the walls, was discovered and mapped from a single measurement as a point constrained to lie on the wall. After the

Figure 1: B21 mobile robot in the plywood box.

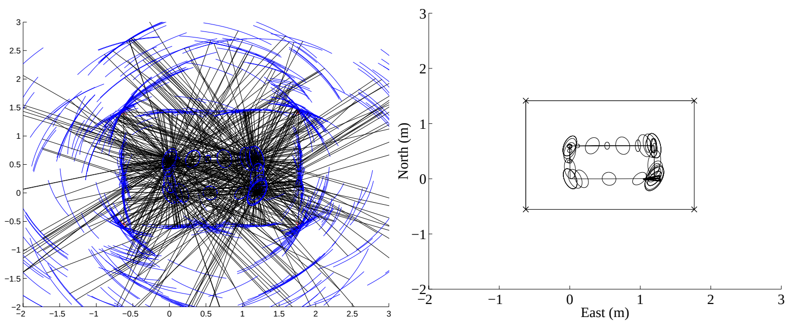
initializations, nine constrained features (shown in Figure 3(a)) were mapped using nine range measurements (shown in Figure 3(b)).

Once these features were initialized, nearest-neighbor gating was performed between all of the remaining sonar measurements and the newly initialized map features. All the measurements that were uniquely matched to one of the nine features are shown in Figure 4(a). Finally, Figure 4(b) shows the result when all these measurements are applied in a single batch update, resulting in a dramatic reduction in the uncertainty ellipses for the estimated feature locations and in the complete trajectory of the vehicle.

5. Conclusion

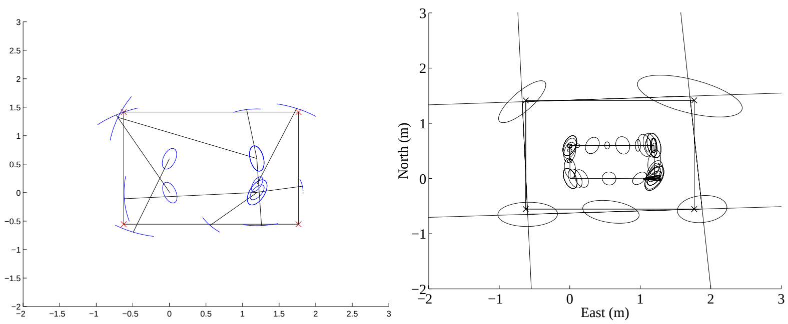
This paper has described a technique for incorporating delayed decision making into stochastic mapping algorithms for CML. The approach enables the initialization of features using data from multiple time steps and improved data association decision-making. The method has been applied using data from a ring of Polaroid sonar sensors from a B21 robot, demonstrating the ability to perform CML with sparse and ambiguous data. The experiment shown above is quite simple, and utilized a priori knowledge of the environmental structure to solve the data association problem. However, the experiment provides one illustration of the benefits of adding past vehicle positions to the state vector, enabling stochastic mapping to be performed in situations where the state of a feature can only by partially observed from a single vehicle position. This approach should allow the development of robust CML implementations that use more complex objects as map features.

Work in progress is applying the technique in combination with decoupled stochastic mapping [10] to larger-scale and more complex experiments using both land and underwater robots.



(a) (b)

Figure 2: (a) Set of all measurements acquired over 50 time steps. Each sonar return is shown as a circular arc, with rays drawn from the center of the dead-reckoned robot position to the center of each arc. (b) Dead-reckoned vehicle trajectory, with 3-σ error ellipses.



(a) (b)

Figure 3: (a) Nine measurements used to initialize nine new features, starting with the corner in the upper left of the figure, and building in both directions around the room, closing the box in the lower right hand corner. (b) State estimates and 3-σ error ellipses for the nine initialized features.

Acknowledgements

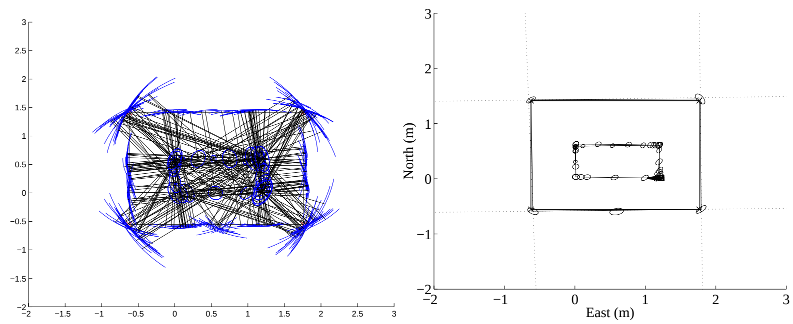
This research has been funded in part by the Henry L. and Grace Doherty Assistant Professorship in Ocean Utilization, NSF Career Award BES-9733040, the MIT Sea Grant College Program under grant NA86RG0074 (project RCM-3), and the US Navy International Programs Office.

References

[1] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 138–145, 1985.

[2] W. K. Stewart. *Multisensor Modeling Underwater with Uncertain Information*. PhD thesis, Massachusetts Institute of Technology, 1988.

[3] P. Moutarlier and R. Chatila. An experimental system for incremental environment mod-



(a) (b)

Figure 4: (a) Sonar measurements that uniquely gated with the nine initialized features, to be used in the batch update. (b) Feature location estimates, vehicle trajectory, and error ellipses after the batch update.

eling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics*, Montreal, June 1989.

[4] N. Ayache and O. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Trans. Robotics and Automation*, 5(6):804–819, 1989.

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

[6] J-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2000.

[7] J. A. Castellanos and J. D. Tardos. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, Boston, 2000.

[8] M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. An experimental and theoretical investigation into simultaneous localization and map building. In *Sixth International Symposium on Experimental Robotics*, pages 265–274, March 1999.

[9] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[10] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D Koditschek and J. Hollerbach, editors, *Robotics Research: The Ninth International Symposium*, pages 169–176, Snowbird, Utah, 2000. Springer Verlag.

[11] 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. Robotics and Automation*, 15(5):948–952, 1999.

[12] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[13] B. D. O. Anderson and J. B. Moore. *Optimal filtering*. Englewood Cliffs, N.J.: Prentice-Hall, 1979.

[14] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.