John J. Leonard
Richard J. Rikoski
Paul M. Newman
Michael Bosse
MIT Department of Ocean Engineering
Cambridge, MA 02139, USA
jleonard@mit.edu
rikoski@mit.edu
pnewman@mit.edu
ifni@graphics.lcs.mit.edu

Mapping Partially Observable Features from Multiple Uncertain Vantage Points

Abstract

In this paper we present a technique for mapping partially observable features from multiple uncertain vantage points. The problem of concurrent mapping and localization (CML) is stated as follows. Starting from an initial known position, a mobile robot travels through a sequence of positions, obtaining a set of sensor measurements at each position. The goal is to process the sensor data to produce an estimate of the trajectory of the robot while concurrently building a map of the environment. In this paper, we describe a generalized framework for CML that incorporates temporal as well as spatial correlations. The representation is expanded to incorporate past vehicle positions in the state vector. Estimates of the correlations between current and previous vehicle states are explicitly maintained. This enables the consistent initialization of map features using data from multiple time steps. Updates to the map and the vehicle trajectory can also be performed in batches of data acquired from multiple vantage points. The method is illustrated with sonar data from a testing tank and via experiments with a B21 land mobile robot, demonstrating the ability to perform CML with sparse and ambiguous data.

KEY WORDS—mapping, navigation, mobile robots

1. Introduction

In this paper we present a generalized framework for feature-based concurrent mapping and localization (CML) that enables mapping of partially observable features from multiple uncertain vantage points. This enables CML to be performed in situations where individual measurements provide weak geometric constraints, such as with wide-beam sonar sensors. The problem of CML, also referred to as simultaneous localization and mapping (SLAM), is stated as follows. Starting from an initial known position, a mobile robot travels through a sequence of positions, obtaining a set of sensor measurements at each position. The goal is to process the sensor data to produce an estimate of the trajectory of the robot while concurrently building a map of the environment.

The key technical difficulty in performing CML is coping with uncertainty (Brooks 1984). For example, Figure 2 shows a SICK laser scanner data and Polaroid sonar data collected by a B21 mobile robot during several back-and-forth traverses of a short corridor (about 60 m total travel length). We can see that the dead-reckoning error of the vehicle becomes intermingled with uncertainty in the values of measurements (noise) and uncertainty in the origins of measurements (spurious reflections, ambiguous associations). These three distinct forms of uncertainty—navigation error, sensor noise, and data association ambiguity—combine to present a challenging data interpretation problem.

The CML problem can be addressed using a variety of different representations, such as evidence grids (Schultz and Adam 1998) and topological models (Kuipers 2000). We advocate the use of a feature-based, probabilistic representation. Smith, Self, and Cheeseman 1987) were the first researchers to cast the problem of feature-based CML using a variable-dimension state estimation formulation. In their approach, the locations of both the robot and a number of objects (geometric features) in the environment are combined into a single state vector in an appropriate parameter space. The locations of the robot and the features are concurrently estimated using recursive state estimation. From the titles of these two papers, we use the term "stochastic mapping" to refer to this type of algorithm.[1] Examples of recent work on CML in robotics that

1. Note that this use of the term stochastic should not be confused with randomized algorithms that themselves are stochastic, such as probabilistic algorithms for motion planning (Kavraki, Svestka, Latombe, and Overmars 1996).

The International Journal of Robotics Research
Vol. 21, No. 10–11, October-November 2002, pp. 943-975,

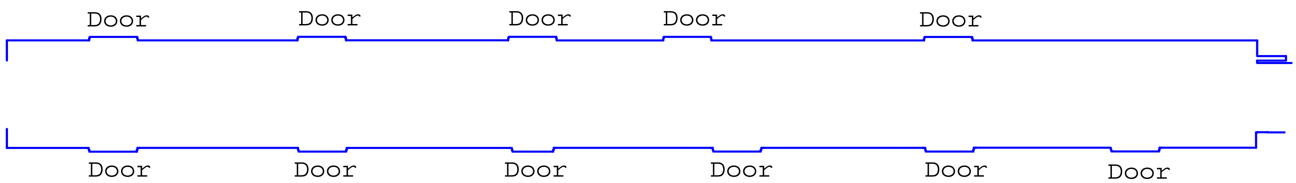


Fig. 1. Hand-measured model of a corridor (total length approximately 25 m).

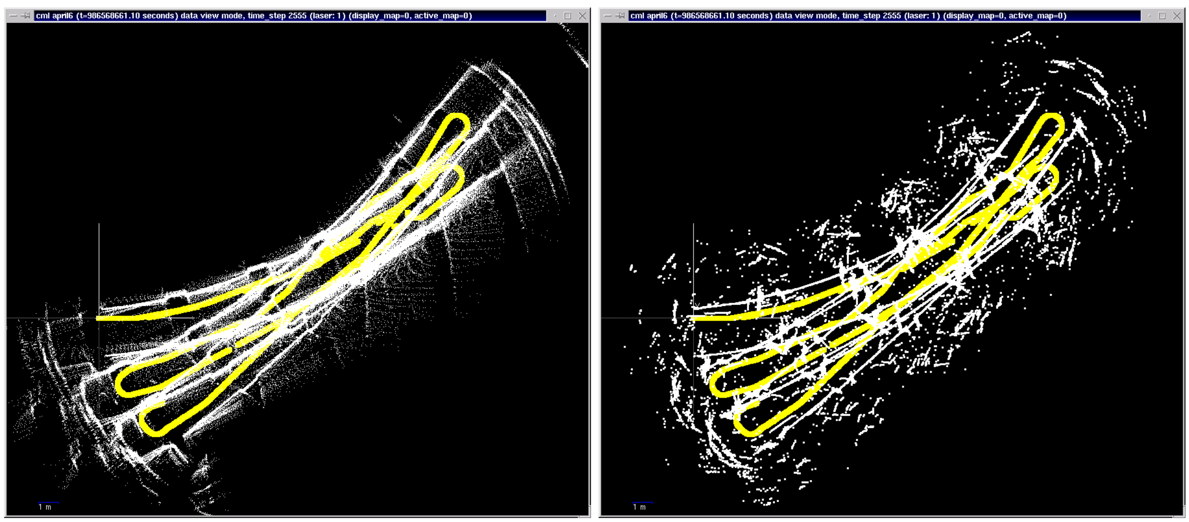


Fig. 2. Laser (left) and sonar (right) data taken with a B21 mobile robot in the corridor shown in Figure 1, referenced to the dead-reckoning position estimate. The vehicle traveled back-and-forth three times following roughly the same path. Each sonar and laser return is shown referenced to odometry. The laser data are slightly smeared during turning due to a latency between the odometry and the laser data.

follow this formulation include Davison using vision (1998), Guivant and Nebot (2001), Castellanos et al. (2000), Dissanayake et al. (1999), and Jensfelt (2001) using laser sensing, and Leonard and Feder (2001), Tardós et al. (2001), and Williams (2001) using sonar. Gibbens et al. have analyzed the closed-form solution of the linear, single-degree-of-freedom CML problem, yielding some insights into convergence. This body of work can be contrasted with other approaches to CML that do not rely on feature-based models, such as Gutmann and Konolige (1999), Thrun (2001) and Choset and Nagatani (2001).

The general problem of CML incorporating data association ambiguity can be cast as a hybrid (mixed continuous/discrete) estimation problem in which tracking and data association are intertwined (Bar-Shalom and Fortmann 1988). General theoretical models for hybrid state estimation, such as multiple hypothesis tracking (Reid 1979; Mori, Chong, Tse, and Wishner 1986), present a staggeringly large computational burden when applied to CML. The navigation error of the platform prevents one from splitting the solution into separate subgroups (known as "clusters" in the tracking literature (Kurien 1990)) corresponding to different parts of the environment (Cox and Leonard 1994). Most implementations of stochastic mapping decouple the discrete (decision-making) and continuous (filtering) parts of the problem. The extended Kalman filter (EKF) is typically used for continuous state estimation (Smith, Self, and Cheeseman 1987; Feder 1999; Guivant and Nebot 2001; Castellanos and Tardós 2000). This is the method used in this paper, but it is not the only state estimation algorithm which could be used. For instance, sequential Monte Carlo algorithms (Doucet, de Freitas, and Gordan 2001; Thrun 2001) could be chosen instead.

Discrete state estimation—making decisions about the origins of measurements—is usually performed in CML with maximum likelihood methods such as "nearest-neighbor gating" (Bar-Shalom and Fortmann 1988). Such methods encounter difficulties when the distance between features in the environment is smaller than the uncertainty in the robot position. Unfortunately, this situation can arise frequently in practice. For example, in the data for the corridor shown in Figure 2, the doors are recessed about 10 cm from the corridor wall. The odometric uncertainty is much larger. In this situation, it is very difficult to associate measurements with features when considered in isolation. A more powerful technique that tests the joint compatibility of multiple sensor measurements, using a branch and bound algorithm, has been developed by Neira and Tardós (2001) and successfully applied to CML using laser data (Newman, Leonard, Neira, and Tardós 2002) and sonar data (Tardós, Neira, Newman, and Leonard 2001). Data association is not the primary focus of this paper, however the topic is intimately linked with the problem of mapping partially observable features. It is desirable to employ methods for perceptual grouping that can reject outliers while finding sets of measurements that, when interpreted together, yield a consistent explanation. In this paper we focus on the state estimation aspects of this problem, describing a representation that allows the output of perceptual grouping routines to be consistently applied for mapping from multiple uncertain vantage points.

In the computer vision community, the analogous problem to CML is structure from motion (SFM) (Faugeras 1993; Taylor and Kriegman 1995; Chiuso, Favaro, Jin, and Soatto 2000; Yagi, Shouya, and Yachida 2000; Hartley and Zisserman 2001). An early approach to SFM that used the EKF was developed by Ayache and Faugeras (1989). This work was one of the first to apply geometric constraints in the state estimation process, such as the fusion of two features in the map that are asserted to be the same. Recently, constraint application has been incorporated in stochastic mapping algorithms by Chong and Kleeman (1997a), Tardós et al. (2001) and Williams et al. (2001). Of recent work in computer vision, the work of Chiuso et al. (2000) and McLauchlan (2000) are the most closely related to the method presented in this paper.

The algorithm of Smith, Self and Cheeseman (1987) for CML uses three models: (1) a robot motion model, (2) a feature mapping model, and (3) a measurement model. We refer to these as the functions $\mathbf{f}(\cdot)$, $\mathbf{g}(\cdot)$, and $\mathbf{h}(\cdot)$, respectively. The robot motion model uses knowledge of the robot's dynamics for state projection. The feature mapping model uses sensor observations to estimate the location of a new geometric feature, so that it may be added to the map. The measurement model predicts observations of mapped features. The basic method assumes that features are stationary, and that there is only one robot, but it can be extended to accommodate dynamic features and/or multiple robots (Fenwick, Newman, and Leonard 2002).

A significant limitation of Smith, Self and Cheeseman (1987) relates to the initialization of new features. The method assumes that the full state of an object can be completely initialized using the measurement data available from a single vehicle position, obtained by a single robot. However, this situation is not satisfied in many important cases of practical interest. In this paper, we present a generalized framework for CML that permits mapping of partially observable features. This can be applied to situations when the measurement data from a single location are insufficient to completely estimate the feature location, such as with wide-beam sonar measurements or angle-only measurements. It also enables composite features, comprised of multiple simple features, to be mapped, such as joining points and lines together to form polygons. The method also has application to mapping by multiple robots.

The issue of estimating partially observable features with measurements obtained from multiple uncertain vantage points is clearly shared in both the CML and SFM problems. State-of-the-art vision algorithms for SFM use bundle adjustment (non-linear least-squares optimization) to concurrently estimate scene structure and camera motion (Triggs, McLauchlan, Hartley, and Fitzgibbon 2000). These techniques typically solve the association problem by using random sample consensus (RANSAC) (Fischler and Bolles 1981) or least medians (Faugeras, Luong, and Papadopoulo 2001) to find groups of measurements that originate from the same point in the scene across multiple images.

While most work has considered the batch SFM problem, there have been some approaches that adopt a recursive approach, which is highly important for navigation applications. Chiuso et al. (2000) have developed a real-time SFM system that can deal with occlusion. In their system, tracked features are not included into the full SFM solution until there is a high certainty that they provide good quality data. McLauchlan has developed the variable state dimension filter (VSDF), which is a hybrid batch/recursive technique that combines the characteristics of the EKF and bundle adjustment (McLauchlan 2000; McLauchlan and Murray 1996; McLauchlan and Murray 1995). The VSDF maintains a dynamic time window of observations and camera motion parameters. We employ a similar idea in this paper. Deans and Hebert (2000) have developed a related method for CML with bearing-only measurements and have performed an experimental investigation of its performance. More recently, Deans (2002) has provided several improvements to the VSDF, including an interpolation scheme that reduces the linearization error and a factorization method that yields computational efficiency.

The structure of this paper is as follows. Section 2 formally defines the problem under consideration. Previous work is reviewed and the problem of mapping with partial observability is formulated. Section 3 describes our new approach to this problem. The key idea is 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, it becomes possible to consistently initialize new map features by combining data from multiple vantage points.

We present two different types of experimental results with the method. In Section 4, we present a series of simplified examples that use manual data association to demonstrate the processes of multi-vantage point initialization and batch measurement processing. The results also demonstrate mapping of composite features and the initialization of a new robot position into a stochastic map. In Section 5, we describe the use of the method within a complete, real-time implementation of CML that uses the Hough transform (Tardós, Neira, Newman, and Leonard 2001) for perceptual grouping. The implementation is being developed to enable autonomous underwater vehicles (AUVs) to perform CML using synthetic aperture sonar (Schmidt 1998). The results in this paper, however, are for a B21 mobile robot navigating in typical indoor environments, such as a corridor, using odometry and Polaroid sonar data. Finally, in Section 6 we provide a further discussion of related research and describe a number of interesting topics for future research.

2. Problem Statement

2.1. General Formulation of the Problem

CML is somewhat unconventional as a state estimation problem for two reasons: (1) data association uncertainty, and (2) variable dimensionality. Initially, the number of features in the environment is unknown and there are no initial location estimates for any features. The initial state vector is restricted to contain only the initial state of the robot. 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. The robot also has to maintain its map, which can incorporate the fusing of two features that are hypothesized to be the same object (Ayache and Faugeras 1989; Chong and Kleeman 1997a) and the deletion of features that are hypothesized to no longer be present (Leonard, Cox, and Durrant-Whyte 1992). In this manner, the number of elements in the stochastic map (and hence the size of the state space) varies through time.

Let us assume 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 $\mathbf{x}_f(k)^T = [\mathbf{x}_{f_1}(k)^T \dots \mathbf{x}_{f_n}(k)^T]^T$ represents the locations of the environmental features. We assume that the robot moves from time k to time $k+1$ in response to a known control input, $\mathbf{u}(k)$, that is corrupted by noise. Let U^k designate the set of all control inputs from time 0 through time k.

The sensors on the robot produce m_k measurements at each step k of discrete time. The set of sensor measurements at time k is designated by $Z(k)$, which is the set $\{\mathbf{z}_j(k) | j = 1 \dots m_k\}$. Let Z^k designate the set of all measurements obtained from time 0 through time k. We assume that each measurement originates from a single feature, or it is spurious. For each measurement $\mathbf{z}_j(k) \in Z(k)$, there is a corresponding assignment index $\mathbf{a}_j$. The value of $\mathbf{a}_j$ is i if measurement $\mathbf{z}_j(k)$ originates from feature i, and it is zero if $\mathbf{z}_j(k)$ is a spurious measurement. Let A^k designate the set of all assignment indices from time 0 through time k. The cardinality of the sets Z^k and A^k are the same. Let n_k designate the number of features that have been measured up through time k (the number of features that have at least one measurement in A^k).

The objective for CML is to compute recursively the probability distribution for the location of the robot and the features and the assignments, given the measurements and the control inputs:

$$p(\mathbf{x}(k), A^k | Z^k, U^{k-1}) = p(\mathbf{x}_r(k), \mathbf{x}_{f_1}(k), \dots, \mathbf{x}_{f_{n_k}}(k), A^k | Z^k, U^{k-1}). \tag{1}$$

Before considering strategies for computing eq. (1), consider first the more restrictive problem of localization and mapping with prior knowledge of all the features and with no data association uncertainty. With perfect knowledge of $A(k)$, we could discard the outliers and combine the remaining measurements of $Z(k)$ into a composite measurement vector $\mathbf{z}(k)$. With prior knowledge of the number of features, and prior state estimates for all features, we are left with a "conventional," fixed-dimension state estimation problem. The general recursive solution applicable for fully non-linear and non-Gaussian systems is well known (Bucy and Senne 1971; Sorenson 1988) and is given by the following two equations

$$p(\mathbf{x}(k)|Z^{k-1}, U^{k-1}) = \int p(\mathbf{x}(k)|\mathbf{x}(k-1), \mathbf{u}(k-1))\, p(\mathbf{x}(k-1)|Z^{k-1}, U^{k-2}) \mathrm{d}\mathbf{x}(k-1) \tag{2}$$

and

$$p(\mathbf{x}(k)|Z^k, U^{k-1}) = c_k\, p(\mathbf{z}(k)|\mathbf{x}(k))\, p(\mathbf{x}(k)|Z^{k-1}, U^{k-1}), \quad k = 1, 2, \dots \tag{3}$$

where $\frac{1}{c_k} = \int p(\mathbf{z}(k)|\mathbf{x}(k))\, p(\mathbf{x}(k)|Z^{k-1}, U^{k-1}) \mathrm{d}\mathbf{x}(k)$. Equation 2 is the Chapman–Komolgorov equation, and represents the use of the dynamic model $p(\mathbf{x}(k)|\mathbf{x}(k-1), \mathbf{u}(k-1))$ for state projection. Equation 3 is Bayes theorem, where $p(\mathbf{z}(k)|\mathbf{x}(k))$ is the measurement model. The direct application of eqs. (2) and (3) entails a computational burden that grows exponentially with the number of features, rendering such application computationally intractable for typical feature-based CML applications in environments with hundreds or more features. Recent work in sequential Monte Carlo methods (Doucet, de Freitas, and Gordan 2001) has achieved successful performance for many challenging non-linear, non-Gaussian state estimation problems; difficulties are encountered, however, in the application of sequential Monte Carlo methods in high-dimensional state spaces (MacCormick 2000).

Equations 2 and 3 assume that the correspondence problem is known. When data association uncertainty (the correspondence problem) is added to the formulation, we are left with a hybrid (mixed continuous/discrete) estimation problem. Mori et al. (1986) published a general recursive non-linear, non-Gaussian algorithm for state estimation with assignment ambiguity. Their solution generalized an earlier linear-Gaussian method by Reid (1979), known as multiple hypothesis tracking (MHT). The solution builds an exponentially growing tree of hypotheses, with each leaf of the tree implementing a different solution to eqs. (2) and (3), based on different hypothesized assignments. Probabilities are assigned recursively to each discrete hypothesis, and pruning is used to restrict the number of hypotheses. While the Mori et al. (1986) solution can accommodate general non-linear, non-Gaussian models, to our knowledge it has never been implemented without simplifying assumptions. Even with the linear-Gaussian assumptions made by Reid's algorithm, the method is exponentially complex due to the combinatorics of discrete decision making. The problem bears some resemblance to object recognition in computer vision (Grimson 1990).

It is unclear how to incorporate variable-dimensionality (initialization of new features based on state estimates for the robot and other features in the map) into the Mori et al. (1986) algorithm. Hence, it is unclear if we can consider the Mori et al. (1986) as the general solution to eq. (1) for the CML problem. Our current opinion is that, because of the interactions between uncertainty and computational complexity, from a general theoretical perspective CML is an "unsolved" problem.

2.2. Linear-Gaussian Approximate Algorithms for CML

The method published in Smith, Self, and Cheeseman (1987) is a linear-Gaussian approximation to the general solution of eqs. (2) and (3). Non-linear functions are linearized via a Taylor series expansion and all probability distributions are approximated by Gaussian distributions. State updates are performed with the EKF. With these approximations, and assuming that data association is known, the computational complexity is reduced to $\mathcal{O}(n^2)$ (Moutarlier and Chatila 1989).

The method recursively computes a state estimate $\hat{\mathbf{x}}(k|k) = [\hat{\mathbf{x}}_r(k|k)^{\mathrm{T}}\ \hat{\mathbf{x}}_f(k)^{\mathrm{T}}]^{\mathrm{T}}$ at each discrete time step k, where $\hat{\mathbf{x}}_r(k|k)^{\mathrm{T}}$ and $\hat{\mathbf{x}}_f(k)^{\mathrm{T}} = [\hat{\mathbf{x}}_{f_1}(k)^{\mathrm{T}} \ldots \hat{\mathbf{x}}_{f_n}(k)^{\mathrm{T}}]^{\mathrm{T}}$ are the robot and feature state estimates, respectively. Based on assumptions about linearization and data association, this estimate is the approximate conditional mean of $p(\mathbf{x}(k)|Z^k, U^{k-1})$:

$$\hat{\mathbf{x}}(k|k) \approx E(\mathbf{x}(k)|Z^k, U^{k-1}). \tag{4}$$

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:

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{rr}(k|k) & \mathbf{P}_{rf}(k|k) \\ \mathbf{P}_{fr}(k|k) & \mathbf{P}_{ff}(k|k) \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{rr}(k|k) & \mathbf{P}_{rf_1}(k|k) & \cdots & \mathbf{P}_{rf_n}(k|k) \\ \mathbf{P}_{f_1r}(k|k) & \mathbf{P}_{f_1f_1}(k|k) & \cdots & \mathbf{P}_{f_1n}(k|k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{f_nr}(k|k) & \mathbf{P}_{f_nf_1}(k|k) & \cdots & \mathbf{P}_{f_nf_n}(k|k) \end{bmatrix}. \tag{5}$$

The method uses three models: a plant model $\mathbf{f}(\cdot)$, a measurement model $\mathbf{h}(\cdot)$, and a feature initialization model $\mathbf{g}(\cdot)$. In this paper we focus on $\mathbf{g}(\cdot)$, presenting a generalized model for feature initialization from multiple uncertain vantage points. The plant model $\mathbf{f}(\cdot)$ is used to make predictions of future vehicle positions based on a control input. The observation model, $\mathbf{h}(\cdot)$, defines the non-linear coordinate transformation from state to observation coordinates. For a more general discussion of these models, see Feder and Leonard (1999) or one of the other references on feature-based CML listed above in Section 1. Before considering the problem of feature initialization in more detail, we now provide a discussion of the data association problem for CML.

2.3. Data Association

To use the models $\mathbf{h}(\cdot)$ and $\mathbf{g}(\cdot)$ properly, stochastic mapping algorithms must make decisions about the origins of measurements. Spurious measurements must be ignored; however, it is often unclear which measurements are spurious. Measurements that are determined to originate from previously mapped features are used via $\mathbf{h}(\cdot)$ to perform a state estimated update. Measurements that are determined to originate from a new feature are used with $\mathbf{g}(\cdot)$ to add the feature to the map.

While there is no mention of the data association problem in Smith, Self and Cheeseman (1987), it is a crucial aspect of the CML problem. The options for data association are rather limited. Powerful tools exist, such as MHT (Reid 1979) or probabilistic data association filter (PDAF) (Bar-Shalom and Fortmann 1988), but the computational burden of these approaches is very high when these techniques are applied to CML. The usual alternative is to employ "nearest-neighbor" gating techniques. For each feature in the 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

$$\boldsymbol{\nu}_{ij}(k)^{\mathrm{T}}\mathbf{S}_{ij}(k)^{-1}\boldsymbol{\nu}_{ij}(k) \leq \gamma, \tag{6}$$

is considered the most likely measurement of that feature

(Bar-Shalom 1988). Such an approach will fail if the features in the environment are too close to one another.

In addition, simply testing the proximity of observations to predicted measurements for previously mapped features provides no indication of when a measurement comes from a new feature. Feature initialization is typically based on looking for several consecutive unexplained measurements that are close to one another, and far from any previously matched features. This policy is referred to as delayed track initiation (Leonard and Durrant-Whyte 1992; Feder, Leonard, and Smith 1999; Dissanayake et al. 2001). In general, there is a trade-off between being more likely to assign a measurement to an old feature, versus using it to initialize a new feature. If we are able to perform feature fusion (Chong and Kleeman 1997), then it is probably better to err on the side of new feature creation. This is the strategy employed in our experiments in Section 5.

A variety of methods for attacking the correspondence problem have been developed in vision, such as RANSAC (Fischler and Bolles 1981). The general idea is to use techniques from robust statistics to find sets of measurements that collectively reinforce one another and yield a single, consistent interpretation. Recently, Neira et al. (2001) have presented a joint compatibility testing method for data association that exploits correlation information when considering potential assignments for groups of measurements. The method succeeds in ambiguous situations when standard nearest-neighbor gating fails. The general policy of looking for consensus among multiple measurements to resolve ambiguity is similar in spirit to RANSAC. Other data association strategies specific to sonar have been proposed; for example, Wijk and Christensen (2000) have recently developed a technique called triangulation-based fusion (TBF) that provides excellent performance for detection of point features from ring sonar data. The TBF method looks for sets of sonar returns obtained from adjacent positions that could all have originated from the same point object, by efficiently computing circle intersection points and applying angle constraints. The method runs in real time and has been successfully used for occupancy grid mapping, model-based localization, and relocation (Wijk and Christensen 2000). CML has also been implemented using the TBF for points features (Zunino and Christensen 2001).

In this paper, we use manual data association in Section 4 to illustrate various new types of feature initialization, and we use a Hough transform voting technique, fully documented in Tardós et al. (2001), to perform initialization of new point and line features when performing real-time CML in Section 5.

2.4. Feature Initialization in Smith, Self, and Cheeseman

The algorithm of Smith, Self and Cheeseman (1987) adds new features to the map using the linear-Gaussian approximation in the following manner. The method assumes 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 $\mathbf{g}(\cdot)$:

$$\hat{\mathbf{x}}_{f_{n+1}}(k) = \mathbf{g}(\hat{\mathbf{x}}(k|k), \mathbf{z}_j(k)). \tag{7}$$

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

$$\hat{\mathbf{x}}_{f_{n+1}}(k) = \mathbf{g}(\hat{\mathbf{x}}(k|k), \mathbf{z}_j(k)) = \begin{bmatrix} x_r + r\cos(\phi + \theta) \\ y_r + r\sin(\phi + \theta) \end{bmatrix}. \tag{8}$$

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

$$\hat{\mathbf{x}}(k|k) \leftarrow \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ \hat{\mathbf{x}}_{f_{n+1}}(k) \end{bmatrix}, \tag{9}$$

$$\mathbf{P}(k|k) \leftarrow \begin{bmatrix} \mathbf{P}_{rr}(k|k) & \mathbf{P}_{rf}(k|k) & \mathbf{P}_{rf_{n+1}}(k|k) \\ \mathbf{P}_{fr}(k|k) & \mathbf{P}_{ff}(k|k) & \mathbf{P}_{ff_{n+1}}(k|k) \\ \mathbf{P}_{f_{n+1}r}(k|k) & \mathbf{P}_{f_{n+1}f}(k|k) & \mathbf{P}_{f_{n+1}f_{n+1}}(k|k) \end{bmatrix}, \tag{10}$$

where

$$\mathbf{P}_{f_{n+1}f_{n+1}}(k|k) = \mathbf{G_x}\mathbf{P}(k|k)\mathbf{G_x}^{\mathrm{T}} + \mathbf{G_z}\mathbf{R}(k)\mathbf{G_z}^{\mathrm{T}}, \tag{11}$$

$$\begin{bmatrix} \mathbf{P}_{f_{n+1}r}(k|k) & \mathbf{P}_{f_{n+1}f}(k|k) \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{f_{n+1}r}(k|k) \\ \mathbf{P}_{f_{n+1}f}(k|k) \end{bmatrix}^{\mathrm{T}} = \mathbf{G_x}\mathbf{P}(k|k). \tag{12}$$

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

3. Mapping Partially Observable Features using an Extended Representation

As mentioned above, the method of Smith, Self and Cheeseman (1987) assumes that 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. To enable CML in situations where this is not the case, we add past vehicle positions to the state vector and maintain explicitly estimates of the correlations between current and previous vehicle states. By incorporating past vehicle locations in the state vector, it becomes possible to make improved probabilistic 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 be used. To maintain consistent error bounds, correlations between different vehicle locations must be taken into account by the CML algorithm. Furthermore, 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. Measurements can also be applied asynchronously, in batches of data from sequences of positions.

To achieve these capabilities, we expand the representation to add a number of previous vehicle locations to the state vector. We refer to past vehicle states that are part of the stochastic map as "trajectory states". We introduce the notation $\mathbf{x}_{t_k}$ (equivalent to $\mathbf{x}_r(k)$) to refer to the true state (pose) of the robot at time $k = t$.

Using trajectory states, the CML problem embodied by eq. (1) is restated in expanded form as the recursive computation of

$$\begin{aligned} p(\mathbf{x}(k), A^k|Z^k, U^{k-1}) = p(&\mathbf{x}_{t_0}, \mathbf{x}_{t_1}, \dots, \mathbf{x}_{t_{k-1}}, \mathbf{x}_r(k), \\ &\mathbf{x}_{f_1}(k), \dots, \mathbf{x}_{f_{n_k}}(k), A^k|Z^k, U^{k-1}). \end{aligned} \tag{13}$$

Expanding the representation in this manner provides a new general framework for feature-based CML.

While the approach is generally applicable using any state estimation framework, in this paper we describe the implementation of the approach in the context of stochastic mapping, yielding a method we refer to as "delayed stochastic mapping". The new method is summarized in Figure 3. The new components of the framework include trajectory state management, perceptual grouping, multiple vantage point initialization, and batch updating.

Each time the vehicle moves, the previous vehicle location is added to the state vector. We introduce the notation $\hat{\mathbf{x}}_{t_i}(k)$ (equivalent to $\hat{\mathbf{x}}_r(i|k)$) 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 $\hat{\mathbf{x}}_t(k) = [\hat{\mathbf{x}}_{t_0}(k)^{\mathrm{T}}\ \ \hat{\mathbf{x}}_{t_1}(k)^{\mathrm{T}}\ \ \hat{\mathbf{x}}_{t_2}(k)^{\mathrm{T}}\ \ \dots\ \ \hat{\mathbf{x}}_{t_{k-1}}(k)]^{\mathrm{T}}$. The complete state vector is

$$\hat{\mathbf{x}}(k|k) = \begin{bmatrix} \hat{\mathbf{x}}_r(k|k) \\ \hat{\mathbf{x}}_t(k) \\ \hat{\mathbf{x}}_f(k) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_r(k|k) \\ \hat{\mathbf{x}}_{t_0}(k) \\ \hat{\mathbf{x}}_{t_1}(k) \\ \hat{\mathbf{x}}_{t_2}(k) \\ \vdots \\ \hat{\mathbf{x}}_{t_{k-1}}(k) \\ \hat{\mathbf{x}}_{f_1}(k) \\ \hat{\mathbf{x}}_{f_2}(k) \\ \hat{\mathbf{x}}_{f_3}(k) \\ \vdots \\ \hat{\mathbf{x}}_{f_{n-1}}(k) \\ \hat{\mathbf{x}}_{f_n}(k) \end{bmatrix}. \tag{14}$$

The associated covariance matrix is

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{rr}(k|k) & \mathbf{P}_{rt}(k|k) & \mathbf{P}_{rf}(k|k) \\ \mathbf{P}_{tr}(k|k) & \mathbf{P}_{tt}(k|k) & \mathbf{P}_{tf}(k|k) \\ \mathbf{P}_{fr}(k|k) & \mathbf{P}_{ft}(k|k) & \mathbf{P}_{ff}(k|k) \end{bmatrix}, \tag{15}$$

or equivalently,

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{rr}(k|k) & \mathbf{P}_{rt_0}(k|k) & \dots & \mathbf{P}_{rt_{k-1}}(k|k) & \mathbf{P}_{rf_1}(k|k) & \dots & \mathbf{P}_{rf_n}(k|k) \\ \mathbf{P}_{t_0r}(k|k) & \mathbf{P}_{t_0t_0}(k|k) & \dots & \mathbf{P}_{t_0t_{k-1}}(k|k) & \mathbf{P}_{t_0f_1}(k|k) & \dots & \mathbf{P}_{t_0f_n}(k|k) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{t_{k-1}r}(k|k) & \mathbf{P}_{t_{k-1}t_0}(k|k) & \dots & \mathbf{P}_{t_{k-1}t_{k-1}}(k|k) & \mathbf{P}_{t_{k-1}f_1}(k|k) & \dots & \mathbf{P}_{t_{k-1}f_n}(k|k) \\ \mathbf{P}_{f_1r}(k|k) & \mathbf{P}_{f_1t_0}(k|k) & \dots & \mathbf{P}_{f_1t_{k-1}}(k|k) & \mathbf{P}_{f_1f_1}(k|k) & \dots & \mathbf{P}_{f_1f_n}(k|k) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{f_nr}(k|k) & \mathbf{P}_{f_nt_0}(k|k) & \dots & \mathbf{P}_{f_nt_{k-1}}(k|k) & \mathbf{P}_{f_nf_1}(k|k) & \dots & \mathbf{P}_{f_nf_n}(k|k) \end{bmatrix}. \tag{16}$$

New trajectory states $\hat{\mathbf{x}}_{t_k}(k) = \hat{\mathbf{x}}_r(k|k)$ are generated at each time step and are added to the state vector:

$$\hat{\mathbf{x}}(k|k) \leftarrow \begin{bmatrix} \hat{\mathbf{x}}_r(k|k) \\ \hat{\mathbf{x}}_{t_0}(k) \\ \hat{\mathbf{x}}_{t_1}(k) \\ \hat{\mathbf{x}}_{t_2}(k) \\ \vdots \\ \hat{\mathbf{x}}_{t_{k-1}}(k) \\ \hat{\mathbf{x}}_{t_k}(k) \\ \hat{\mathbf{x}}_f(k) \end{bmatrix}. \tag{17}$$

The state covariance is expanded as follows

$$\mathbf{P}(k|k) \leftarrow \begin{bmatrix} \mathbf{P}_{rr}(k|k) & \mathbf{P}_{rt_0}(k|k) & \dots & \mathbf{P}_{rt_{k-1}}(k|k) & \mathbf{P}_{rt_k}(k|k) & \mathbf{P}_{rf}(k|k) \\ \mathbf{P}_{t_0r}(k|k) & \mathbf{P}_{t_0t_0}(k|k) & \dots & \mathbf{P}_{t_0t_{k-1}}(k|k) & \mathbf{P}_{t_0t_k}(k|k) & \mathbf{P}_{t_0f}(k|k) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{P}_{t_{k-1}r}(k|k) & \mathbf{P}_{t_{k-1}t_0}(k|k) & \dots & \mathbf{P}_{t_{k-1}t_{k-1}}(k|k) & \mathbf{P}_{t_{k-1}t_k}(k|k) & \mathbf{P}_{t_{k-1}f}(k|k) \\ \mathbf{P}_{t_kr}(k|k) & \mathbf{P}_{t_kt_0}(k|k) & \dots & \mathbf{P}_{t_kt_{k-1}}(k|k) & \mathbf{P}_{t_kt_k}(k|k) & \mathbf{P}_{t_kf}(k|k) \\ \mathbf{P}_{fr}(k|k) & \mathbf{P}_{ft_0}(k|k) & \dots & \mathbf{P}_{ft_{k-1}}(k|k) & \mathbf{P}_{ft_k}(k|k) & \mathbf{P}_{ff}(k|k) \end{bmatrix}, \tag{18}$$

where $\mathbf{P}_{t_kt_i}(k|k) = \mathbf{P}_{rt_i}(k|k)$, $\mathbf{P}_{t_kf}(k|k) = \mathbf{P}_{rf}(k|k)$, and $\mathbf{P}_{t_kt_k}(k|k) = \mathbf{P}_{rr}(k|k)$.

The growth of the state vector in this manner increases the computational burden as $\mathcal{O}(n^2)$, so caution must be taken. The

1. **while** active mission **do**
2. $\hat{\mathbf{x}}(k|k-1) = \mathbf{f}(\hat{\mathbf{x}}(k-1|k-1), \mathbf{u}(k))$ {state projection}
3. $\mathbf{P} = \mathbf{F_x P F_x^T} + \mathbf{Q}$ {covariance projection}
4. $\hat{\mathbf{z}}(k) = \mathbf{h}(\hat{\mathbf{x}}(k|k-1))$ {sensor prediction}
5. $(\mathbf{a}, \neg\mathbf{a}) \leftarrow (\mathbf{z}(k), \hat{\mathbf{z}}(k))$ {data association}
6. $\mathbf{S} = \mathbf{H_x P H_x^T} + \mathbf{R}$ {innovation covariance}
7. $\mathbf{K} = \mathbf{P H_x}^{\mathrm{T}} \mathbf{S}^{-1}$ {Kalman gain}
8. $\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(\mathbf{z_a} - \hat{\mathbf{z}}_\mathbf{a})$ {Kalman state update}
9. $\mathbf{P} = \mathbf{P} - \mathbf{K S K}^{\mathrm{T}}$ {Kalman covariance update}
10. $\hat{\mathbf{x}}(k|k) \leftarrow \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ \mathbf{g}(\hat{\mathbf{x}}(k|k), \mathbf{z}_{\neg\mathbf{a}}) \end{bmatrix}$ {mapping state}
11. $\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P G_x^T} \\ \mathbf{G_x P} & \mathbf{G_x P G_x^T} + \mathbf{G_z R G_z^T} \end{bmatrix}$ {mapping covariance}
12. $k = k + 1$
13. **end while**

1. **while** active mission **do**
2. $\hat{\mathbf{x}}(k|k-1) \leftarrow \begin{bmatrix} \mathbf{f}(\hat{\mathbf{x}}_r(k-1|k-1), \mathbf{u}(k)) \\ \hat{\mathbf{x}}(k-1|k-1) \end{bmatrix}$ {state augmentation with projection}
3. $\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{F_x P F_x^T} + \mathbf{Q} & \mathbf{F_x P} \\ \mathbf{P F_x}^{\mathrm{T}} & \mathbf{P} \end{bmatrix}$ {covariance augmentation with projection}
4. $\hat{\mathbf{z}}(k) = \mathbf{h}(\hat{\mathbf{x}}(k|k-1))$ {sensor prediction}
5. $(\mathbf{a}, \neg\mathbf{a}) \leftarrow (\mathbf{z}(k), \hat{\mathbf{z}}(k))$ {data association}
6. $\mathbf{S} = \mathbf{H_x P H_x^T} + \mathbf{R}$ {innovation covariance}
7. $\mathbf{K} = \mathbf{P H_x}^{\mathrm{T}} \mathbf{S}^{-1}$ {Kalman gain}
8. $\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(\mathbf{z_a} - \hat{\mathbf{z}}_\mathbf{a})$ {Kalman state update}
9. $\mathbf{P} = \mathbf{P} - \mathbf{K S K}^{\mathrm{T}}$ {Kalman covariance update}
10. $\hat{\mathbf{x}}(k|k) = \begin{bmatrix} \hat{\mathbf{x}}(k|k) \\ \mathbf{g}(\hat{\mathbf{x}}(k|k), \mathbf{z}_{\neg\mathbf{a}}) \end{bmatrix}$ {mapping state}
11. $\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P G_x^T} \\ \mathbf{G_x P} & \mathbf{G_x P G_x^T} + \mathbf{G_z R G_z^T} \end{bmatrix}$ {mapping covariance}
12. Contract the state **x** and covariance **P** to remove unnecessary trajectory states and associated terms in the covariance matrix
13. $k = k + 1$
14. **end while**

Fig. 3. Comparison of conventional stochastic mapping (top) and delayed stochastic mapping (bottom). The notation is summarized as follows: **x** is the state vector; **P** is the covariance; **F**, **G**, and **H** are the Jacobians of their respective non-linear functions; **Q** and **R** are the propagation and measurement covariance matrices; **u** is the control input; **z** are the observations; **a** labels associated observations; ¬**a** labels unmatched observations.

new problem of trajectory state management is introduced. Old vehicle trajectory states and associated terms in the covariance need to be deleted once all the measurements from a given time step have been either processed or discarded. In practice, we have seen excellent performance by keeping the number of trajectory states restricted to a fixed size of 40. With a fixed window size, the process of adding past states is similar to a fixed-lag Kalman smoother (Anderson and Moore 1979).

3.1. Perceptual Grouping

The next step in the framework is to apply a perceptual grouping algorithm to examine collectively the entire set of data that came from the current and past vehicle positions currently in the map. Instead of being forced to make an instantaneous decision about the origins of current measurements, delayed decision making is now possible. In general, a wide variety of perceptual grouping algorithms are possible, such as the RANSAC (Fischler and Bolles 1981) and least median (Faugeras, Luong, and Papadopoulo 2001) methods that have been successfully employed in vision. The subject of delayed decision making is very broad in scope, and in this paper we focus on the state estimation aspects of the problem only. For now, we assume that some perceptual grouping algorithm has been employed to classify and associate measurements. Further discussion of perceptual grouping is contained below in Section 5.

3.2. Feature Initialization using Data from Multiple Vantage Points

Given decisions about the origins of measurements from multiple vantage points, the next step is to use the trajectory states and associated correlation information to initialize new features using measurements from multiple vantage points. Suppose that a perceptual grouping method has produced a set of candidate measurements, Z_C, from a set of vehicle positions that are currently contained in the set of active trajectory states of the stochastic map. Initialization can then be performed by picking a minimal subset Z_S of "seed" features from Z_C that are sufficient to estimate the state of the new feature. Let X_S be the set of vehicle positions for the measurements in Z_S. Then, the location of the new feature can be computed as

$$\hat{\mathbf{x}}_{f_{n+1}} = \mathbf{g}(X_S, Z_S). \tag{19}$$

For example, consider the initialization of a new point feature in two dimensions, using two range measurements, r_1 and r_2, taken at time steps k_1 and k_2. The function $\mathbf{g}(\cdot)$ represents a solution for the intersection of two circles. The algebra is very simple and is reviewed in Appendix A. The beamwidth of the sonar sensor can be used as an angle constraint to rule out multiple solutions (Leonard and Durrant-Whyte 1992; Wijk and Christensen 2000). The covariance for the new feature is initialized in a similar fashion as shown above in eqs. (10)–(12), except that the Jacobian matrix $\mathbf{G_x}$ will contain additional non-zero terms corresponding to the trajectory states and the Jacobian matrix $\mathbf{G_z}$. The direct differentiation of the multiple vantage point initialization function can be cumbersome. It is possible, however, to compute the initialization Jacobians by inverting the Jacobian of the composite function of minimal observations, as follows

$$\mathbf{G_z} = \mathbf{H_y}^{-1}, \tag{20}$$

and

$$\mathbf{G_x} = -\mathbf{H_y}^{-1}\mathbf{H_x}, \tag{21}$$

where $\mathbf{y} = \mathbf{x}_{f_{n+1}}$ is the state estimate for the new feature. This is possible because $\mathbf{H_y}$ is a square matrix with full rank.

An alternative is to compute the Jacobians numerically (Durrant-Whyte, Julier, and Ulhmann 1996); this is the approach used for the results in this paper. For initialization of a line feature from two positions with two sonar measurements, the function $\mathbf{g}(\cdot)$ represents the solution for the common tangents of two circles. The general procedure is the same if the feature initialization function $\mathbf{g}(\cdot)$ is a function of measurements from more than two time steps, for example to initialize a cylinder using three range values.

3.3. Batch Updating

Once a new feature is initialized, the map can be simultaneously updated using all other previously obtained measurements that can be associated with the new feature (measurements that are elements of Z_C but not elements of Z_S). This update is performed using a composite measurement vector, consisting of measurements obtained for different time steps. We call this procedure a "batch update". It allows the maximum amount of information to be extracted from all past measurements. Empirically, we have observed that the EKF can be less prone to divergence when batch updates are performed.

To update the robot history and features using a batch of measurements, a measurement vector $\mathbf{z_B}$ is constructed from an appropriate temporal observation set:

$$\mathbf{z}_B = \begin{bmatrix} \mathbf{z}(k-\delta_0) \\ \mathbf{z}(k-\delta_1) \\ \mathbf{z}(k-\delta_2) \\ \vdots \\ \mathbf{z}(k-\delta_n) \end{bmatrix}. \tag{22}$$

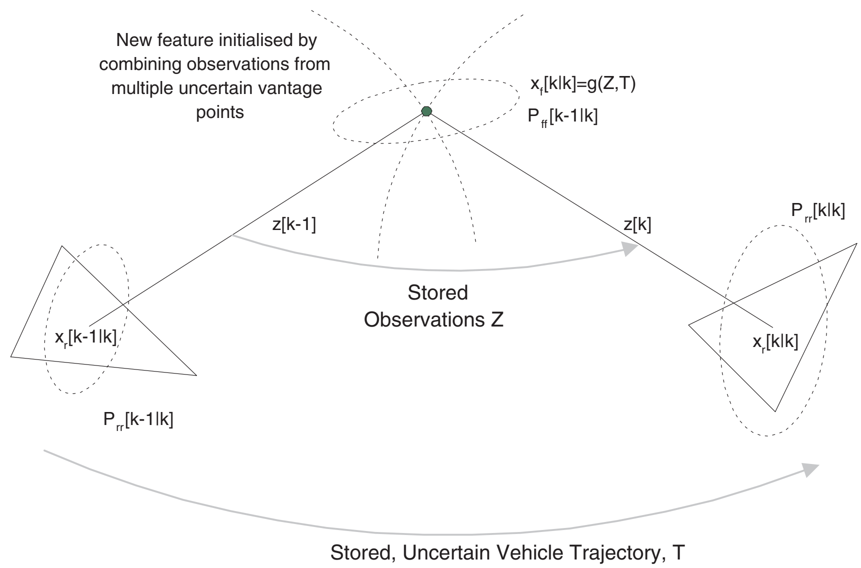


Fig. 4. Illustration of initialization from multiple vantage points.

An appropriate predicted measurement vector is constructed:

$$\hat{\mathbf{z}}_B = \begin{bmatrix} \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_0), \hat{\mathbf{x}}_f(k)) \\ \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_1), \hat{\mathbf{x}}_f(k)) \\ \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_2), \hat{\mathbf{x}}_f(k)) \\ \vdots \\ \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_n), \hat{\mathbf{x}}_f(k)) \end{bmatrix}. \tag{23}$$

This leads to an innovation which includes observations from across multiple time steps:

$$\nu = \begin{bmatrix} \mathbf{z}(k-\delta_0) - \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_0), \hat{\mathbf{x}}_f(k)) \\ \mathbf{z}(k-\delta_1) - \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_1), \hat{\mathbf{x}}_f(k)) \\ \mathbf{z}(k-\delta_2) - \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_2), \hat{\mathbf{x}}_f(k)) \\ \vdots \\ \mathbf{z}(k-\delta_n) - \mathbf{h}(\hat{\mathbf{x}}_r(k-\delta_n), \hat{\mathbf{x}}_f(k)) \end{bmatrix}. \tag{24}$$

Similarly, the measurement Jacobian $\mathbf{H}_x$ is constructed, as is the measurement covariance $\mathbf{R}$:

$$\mathbf{H}_{\mathbf{x}} = \begin{bmatrix} \mathbf{H}_{\mathbf{x}_r(k-\delta_0),\mathbf{x}_f(k)} \\ \mathbf{H}_{\mathbf{x}_r(k-\delta_1),\mathbf{x}_f(k)} \\ \mathbf{H}_{\mathbf{x}_r(k-\delta_2),\mathbf{x}_f(k)} \\ \vdots \\ \mathbf{H}_{\mathbf{x}_r(k-\delta_n),\mathbf{x}_f(k)} \end{bmatrix} \tag{25}$$

$$\mathbf{R} = \tag{26}$$

$$\begin{bmatrix} \mathbf{R}(k-\delta_0) & 0 & 0 & \dots & 0 \\ 0 & \mathbf{R}(k-\delta_1) & 0 & \dots & 0 \\ 0 & 0 & \mathbf{R}(k-\delta_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{R}(k-\delta_n) \end{bmatrix}.$$

Using these components and the standard Kalman update equations, all robot trajectory positions and all features that are currently in map are concurrently updated.

3.4. Composite Initialization

In general, the feature initialization function can use *any* of the information in the state estimation $\hat{\mathbf{x}}$ for the stochastic map, including the locations of other previously mapped features as well as as previous vehicle states. The new feature location can be a function of one or more previous mapping features, $\hat{\mathbf{x}}_{f_i}$, one or more measurements, and the robot state estimate corresponding to these measurements. For example, we can initialize a line that passes through a point feature $\hat{\mathbf{x}}_{f_i}$ and is tangent to one sonar return $\mathbf{z}_j(k)$. In this case, the feature initialize function is of the form:

$$\hat{\mathbf{x}}_{f_{n+1}} = \mathbf{g}(\hat{\mathbf{x}}_{f_i}, \hat{\mathbf{x}}_r(k), \mathbf{z}_j(k)). \tag{27}$$

Alternatively, we can initialize a point that lies at the intersection of a line currently in the map and a new sonar return. The equations for these two initialization scenarios are described in Appendix A 1.3. We can also initialize a new feature without any measurements, for example, hypothesizing the constraint that a new point feature exists at the intersection of two line segments currently in the map. Examples with real data for several of these scenarios are given below in Section 4.

3.5. Extension to Mapping by Multiple Robots

Cooperative stochastic mapping with perfect communications requires adding robots to the state vector (Fenwick, Newman, and Leonard 2002). Although there are numerous technical issues to overcome, from a state estimation perspective it is relatively straightforward to perform feature initialization using data from multiple robots. By using trajectory states, such initializations can occur on a delayed basis, removing the requirement for the two robots to sense the features in question at precisely the same time. It is also possible for one robot to "map" the location of another robot (add the other robot to its state vector), through processing measurements of common features (assuming that association is known). A simplified example of how this process can work is shown below in Section 4.

4. Examples Using Manual Association

We now present several illustrations of the concepts presented above using real sonar data sets with manual data association. The first experiment uses 500 kHz underwater sonar data acquired in a testing tank, and the second experiment uses data from a ring of 24 Polaroid sonar sensors. Both experiments use manually-guided data association strategies that exploit *a priori* knowledge of the environment. While both environments are highly simplified, they are useful in illustrating the state estimation process for mapping from multiple uncertain vantage points. Fully automatic data association is used in the experiments in Section 5, as part of an integrated system that can perform CML in real-time using odometry and wide-beam sonar measurements.

4.1. Tank Experiment

An experiment was conducted using a robotic gantry to emulate the motion and sensing of an underwater vehicle. A 500 KHz binaural sonar was used (Kuc 1996; Au 1993). To show mapping of partially observable features, bearing information was discarded. Two objects were placed in the tank, a metal triangle and a point-like object (a fishing bobber). The gantry was moved through two trajectories, one to the left and one to the right of the objects, emulating cooperative CML by two vehicles. All processing was post-processing. Data association was done by hand since it is not the focus of this paper. The manually-associated returns used for feature initialization are labeled in Figures 7 and 10 and are listed in Table 1. The initialization strategies used for each feature and for the position of the second robot are listed in Table 2. We consider the set of measurements from the right side of the objects to originate from "robot 1" and the measurements from the left side to be from "robot 2".

The gantry operates in a tank that is 10 m long by 3 m wide by 1 m deep. The mechanism provides ground-truth good to a few millimeters. Simulated speed and heading measurements were generated and used for dead-reckoning. Initially, the sensor dead-reckoned through a trajectory of 11 positions, as shown in Figure 5. Upon completing this trajectory, the robot had a state vector and covariance matrix which contained only robot states, one estimate for each position. In Figure 6, the correlation coefficients for the x components of the trajectory are plotted. Each line represents the correlations between one time step and all other saved time steps. Because this is a short dead-reckoned trajectory, each curve has only one maxima; more complex trajectories may have numerous local maxima.

The assumed range measurement standard deviation was 3 cm for each measurement. The added process noise had a standard deviation of 1 cm per time step in x and y and 2 degrees per time step in heading.

The data processing was performed as follows. First, state projection was performed and trajectory states were created for robot 1 for time steps 1–11, without any measurements being processed. The three-σ error bounds for the dead-reckoned (x, y) trajectory of robot 1 are shown in Figure 5. The correlation coefficients between the x coordinates of the robot trajectory states are plotted in Figure 6.

Having an entire trajectory of positions, robot 1 starts to construct its map. Because the robot uses a range measurements only, features must be observed from multiple vantage points to be mapped. By combining Returns 1 and 5 (labeled in Figure 7), the robot initializes the point object at the bottom of its map (Figure 7). Similarly, by intersecting two more arcs (Returns 2 and 6), the bottom corner of the triangle is mapped. The equations for arc intersection are given in Appendix A 1.1. Next, Return 4 is used in conjunction with the estimated location of the bottom corner of the triangle to add the right wall of the triangle to the map. Finally, by intersecting Return 3 with the estimated line corresponding to the right wall, the top corner is added to the map. Using these observations, the point object and the side of the triangle are initialized (Figure 8).

The wall is represented in the map by an infinite line with two parameters, ρ and θ, which are the angle and offset of the normal with respect to the origin. The dashed lines in Figures 7 and 8 show the extension of the estimated line.

After initializing the features, all other observations are used for a batch update of the newly initialized features (Figure 9). Mapping and navigation are improved substantially. Robot 1 obtained 29 total measurements. Of these, six were used for initialization and 23 were used batch updating.

Next, robot 1 tries to use information from robot 2. We assume that there is no *a priori* information for the initial location of robot 2, and that hence robot 2 must be initialized into the map using shared measurements of features seen by both robots.

Table 1. Details for the Ten Manually-selected Returns Used for Feature Initialization

Return number	Robot	Time Step	Odometry X (m)	Odometry Y (m)	Range (m)
1	1	1	0.0	0.0	1.0036
2	1	1	0.0	0.0	1.8058
3	1	3	0.0045	0.4992	1.8380
4	1	8	−0.0399	1.7637	0.9504
5	1	11	−0.0235	2.537	2.7205
6	1	11	0.0235	2.5373	1.4192
7	2	1	(unused)	(unused)	0.9773
8	2	1	(unused)	(unused)	1.7851
9	2	2	(unused)	(unused)	1.0054
10	2	2	(unused)	(unused)	1.5692

Table 2. Method of Initialization for Features

Feature	Initialization Method
Point object	Return 1 and Return 5
Lower right vertex of triangle	Return 2 and Return 6
Right plane of triangle	Lower right vertex of triangle and Return 4
Upper right vertex of triangle	Right plane of triangle and Return 3
Position 1 for robot 2	Point object and Returns 7 and 8
Position 2 for robot 2	Point object and Returns 9 and 10

Table 3. Comparison of Hand-measured and Estimated Feature Locations for Points Features (in meters)

	Hand measured		CML estimated	
Feature	x	y	x	y
Point object	−1.0	0.0	−1.0054	0.0109
Lower right vertex of triangle	−1.0	1.5	−0.99	1.5045
Upper right vertex of triangle	−1.0	2.05	−0.9922	2.0837
Left vertex of triangle	−1.4763	1.77	−1.4908	1.7846

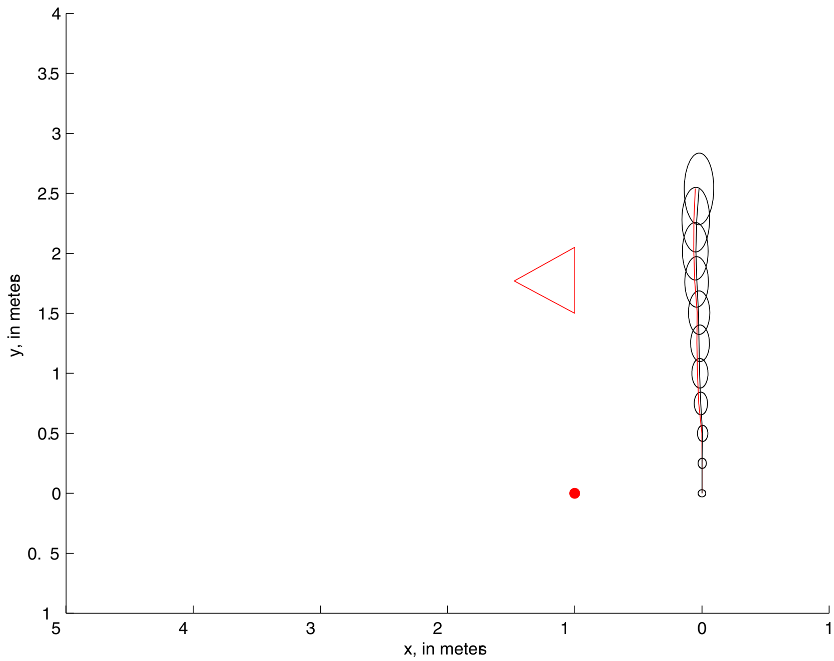


Fig. 5. Dead-reckoned trajectory of robot 1. The robot started at the bottom and moved upwards. The ellipses represent the 99% confidence interval for each position. The triangle is an aluminum sonar target, and the small filled circle at the position $x = -1.0$ and $y = 0.0$ is a fishing bobber, which served as a "point" object.

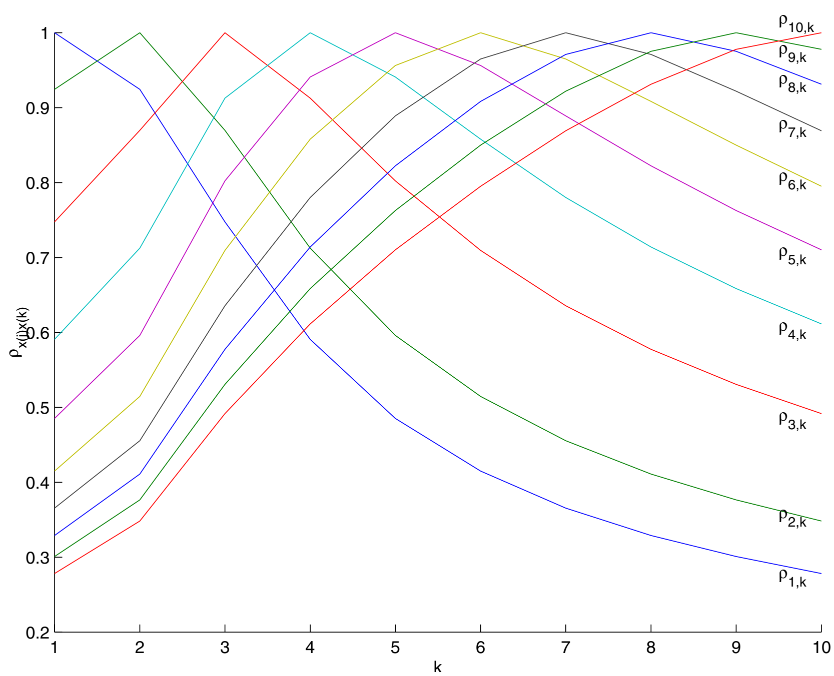


Fig. 6. Correlation coefficients between the x components of the trajectory of robot 1. Each line represents the correlations between a specific time step and all other time steps.

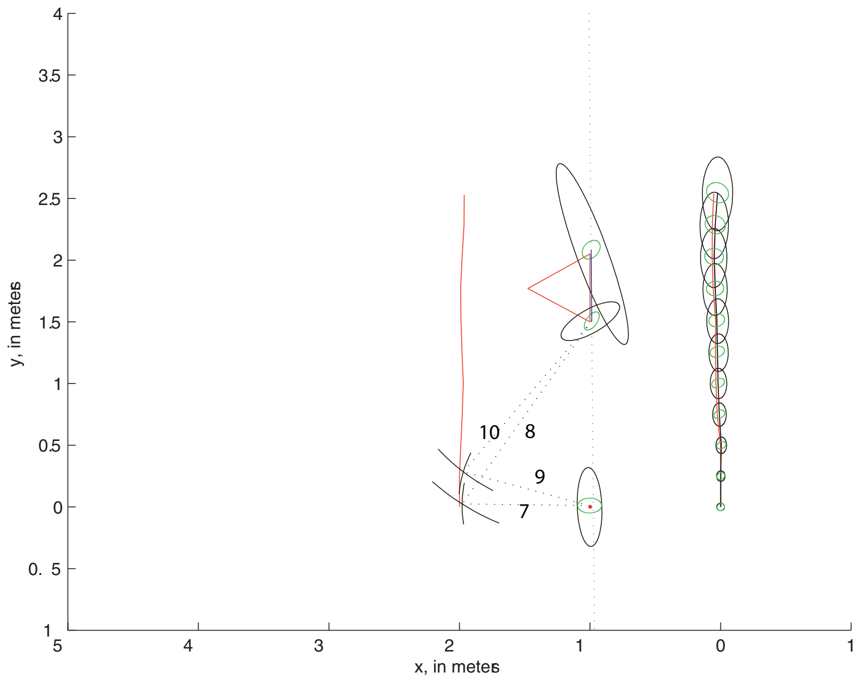


Fig. 7. Set of observations used to initialize the side of the triangle and the point object. Two observations are needed to initialize the point target, two more are needed to initialize the corner of the triangle. Given the constraint of the corner, only one measurement was needed to map the wall. Given the wall, only one more measurement was needed to map the top corner of the triangle.

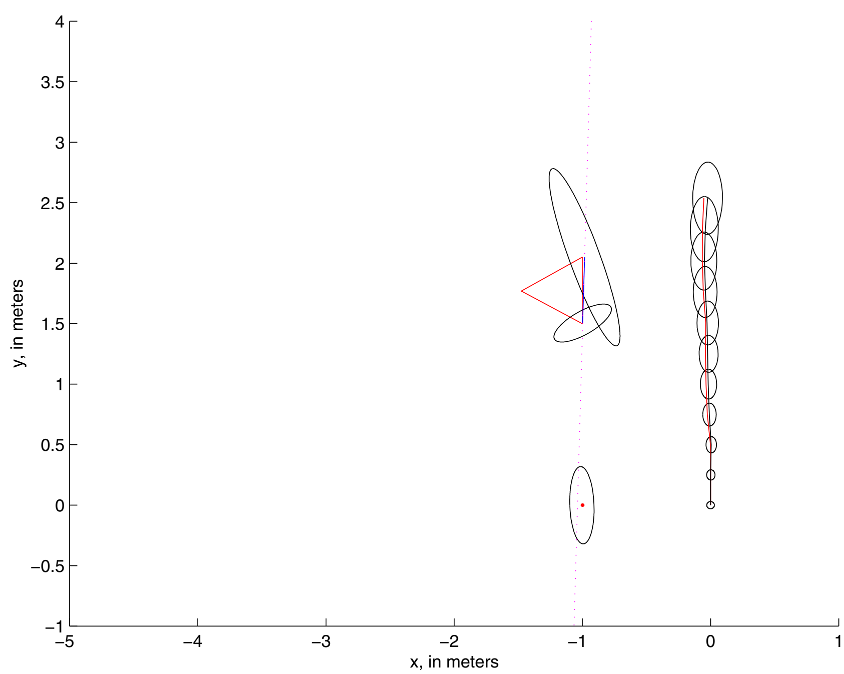


Fig. 8. Initial map. 99% confidence intervals for the corners and the point object are shown.

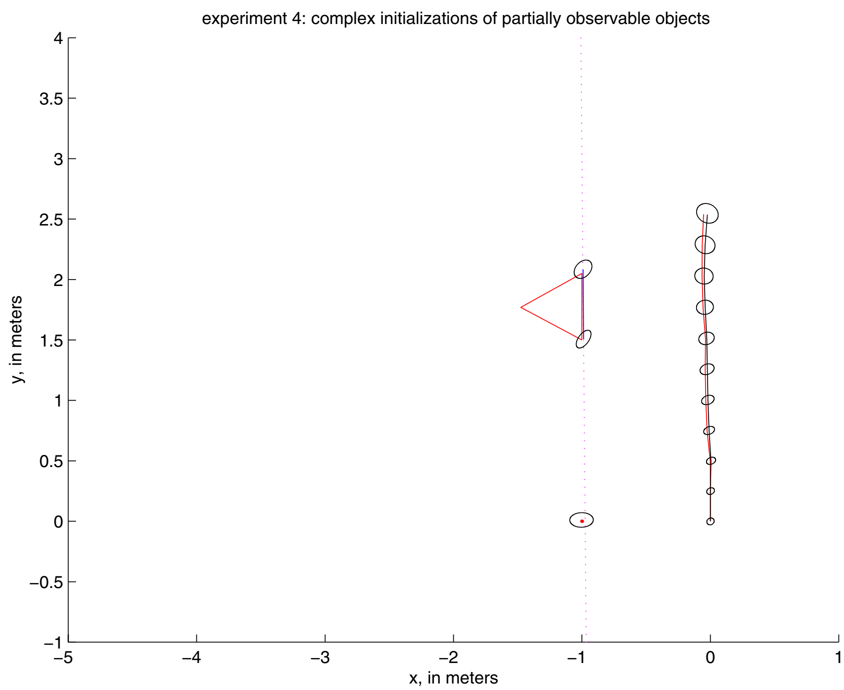


Fig. 9. Map after a batch update. Note the improved confidence intervals for the features and the robot.

It is determined that robot 2 has observed features in the map of robot 1. From the ranges to the point object and the bottom corner of the triangle, (Returns 7–10 as labeled in Figure 10) the first two positions of robot 2 can be observed. Those two positions are initialized into the map, and their initialization function is of the form $\mathbf{g}(\mathbf{x}_{f_{r1}}, \mathbf{z}_{r2})$, meaning that robot 2 is added to the map of robot 1 using robot 2's observations of features that have been previously mapped by robot 1 (Figure 11).

Next, using the first two estimated positions for robot 2, the initial heading and velocity of robot 2 are mapped. Using this information, along with the control inputs, a dead-reckoned trajectory for robot 2 is established (Figure 12). Because neither compass nor velocity observations are available, and because the initial estimates of velocity and heading for robot 2 are imprecise, the trajectory is imprecise and has large error bounds.

Having a dead-reckoned trajectory for robot 2 and its measurements, robot 1 then maps the (otherwise unobservable) back side of the triangle and performs a batch update to get an improved map and an improved estimate of where robot 2 traveled (Figure 13) . Robot 2 obtained 22 total measurements. Of these, four measurements were used to initialize the position of robot 2 in the stochastic map of robot 1. Two measurements were used in initializing features on the left side of the triangle, and the remaining 16 returns were used in batch updating.

The error bounds for robot 2 do not exhibit the growth profile that is normally seen. Normally, since the robot starts with an initial position estimate and then moves, the uncertainty grows with time. In this case, since the second robot is mapped and localized with respect to previously mapped features, the smoothed estimate of its trajectory has the least uncertainty in the middle (Figure 14).

The next experiment is for data from a simple "box" environment made of plywood, demonstrating the process of multiple vantage point initialization, batch updating, and composite feature initialization. The data association and feature modeling techniques utilize the *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. The input data consists of 600 sonar returns acquired from a sequence of 50 positions that form one-and-a-half loops around the inside of the box. The vehicle started in the lower left corner facing upward.

The data processing proceeded as follows. First, state projection was performed and trajectory states were created for time steps 1–50, without any measurements being processed.

At each processing cycle, a new vehicle trajectory state was added to the system state vector, using eqs. (17) and (18). The dead-reckoned vehicle trajectory is shown in Figure 16(b). After 50 cycles, a manually-guided search strategy was performed to find nine returns that were used to initialize nine features (the four corners and four walls of the box and a

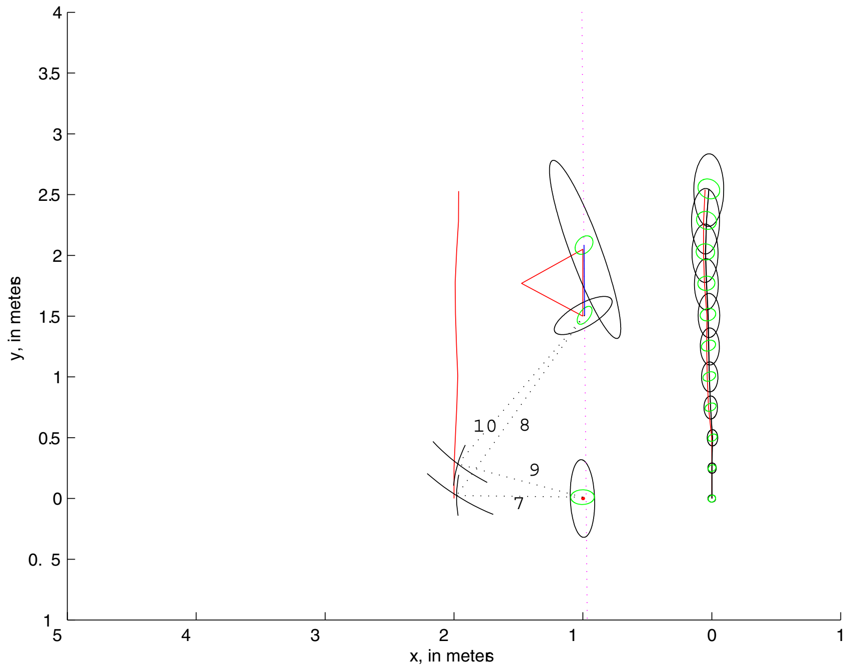


Fig. 10. Adding in the second robot. The true trajectory for robot 2 is the line on the left. Observations of the bottom corner of the triangle and of the fishing bobber are reversed to find the first two vantage points.

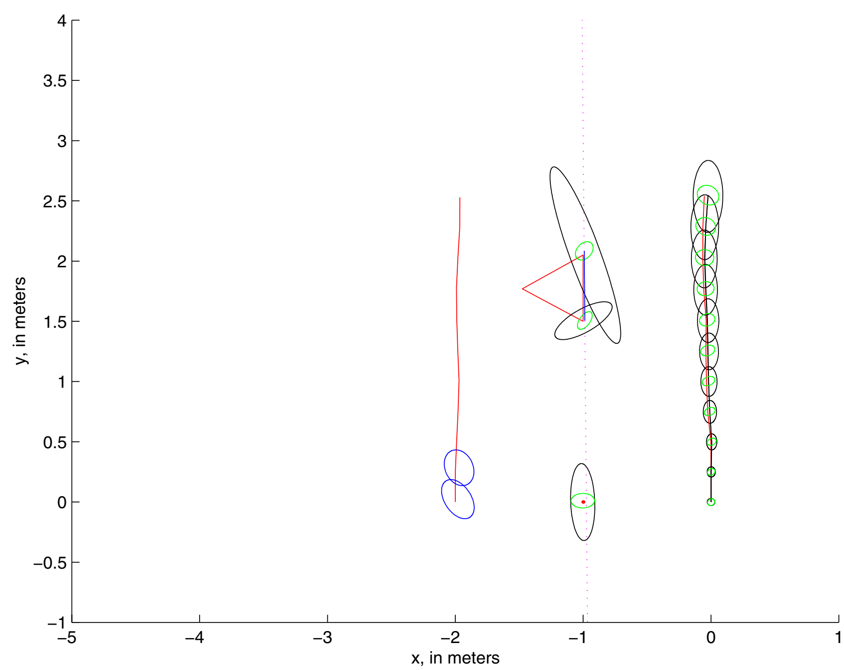


Fig. 11. The first two positions for robot 2 are mapped. Using these two positions, it is possible to estimate the initial heading and velocity.

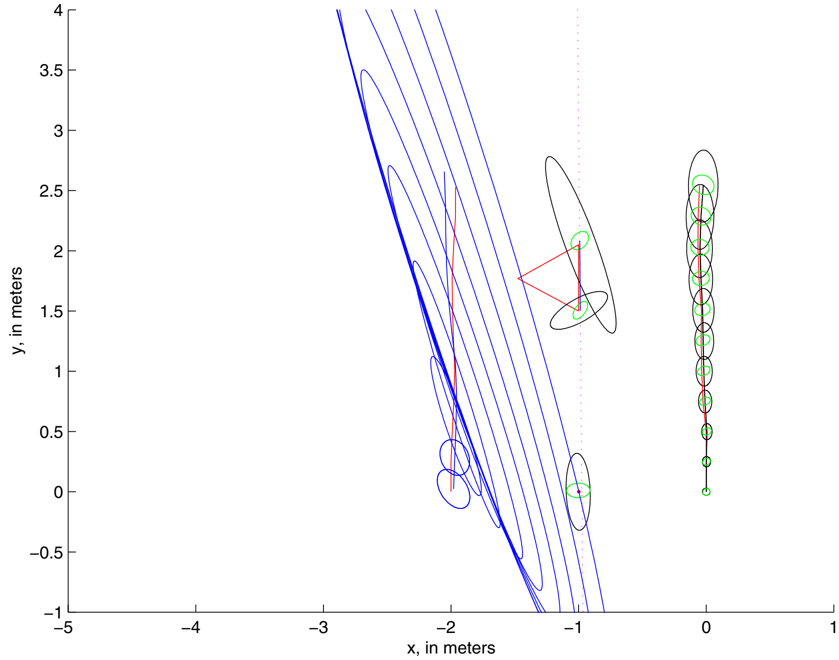


Fig. 12. Using the estimated initial heading and velocity for robot 2, along with its control inputs, a dead-reckoned trajectory is constructed. With poor initial estimates of heading and velocity, and no compass or velocity measurements for updates, the trajectory is very imprecise.

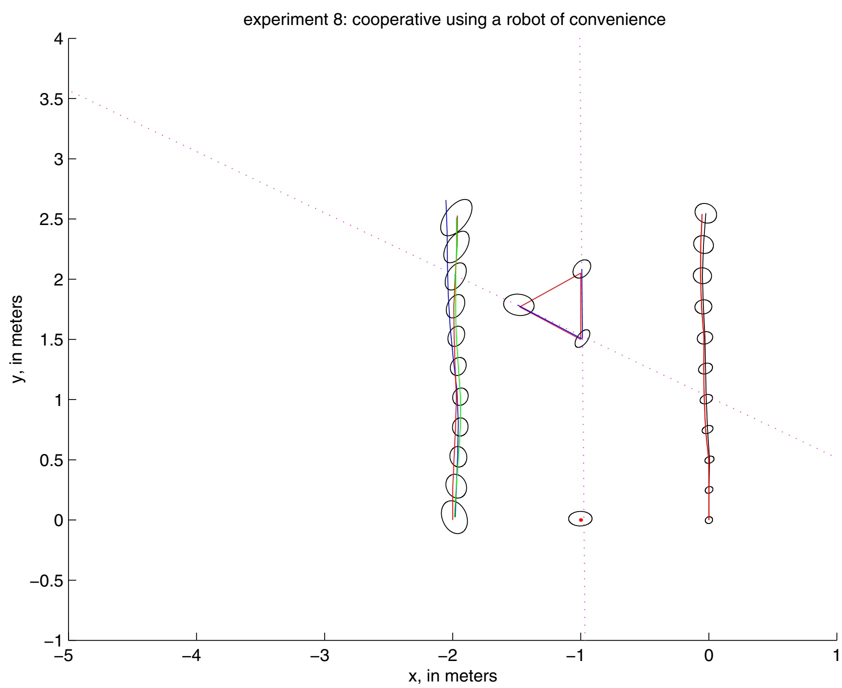


Fig. 13. Using information from robot 2, robot 1 is able to map the back side of the triangle, which it otherwise could not observe. After the batch update, its estimate of robot 2 improves considerably.

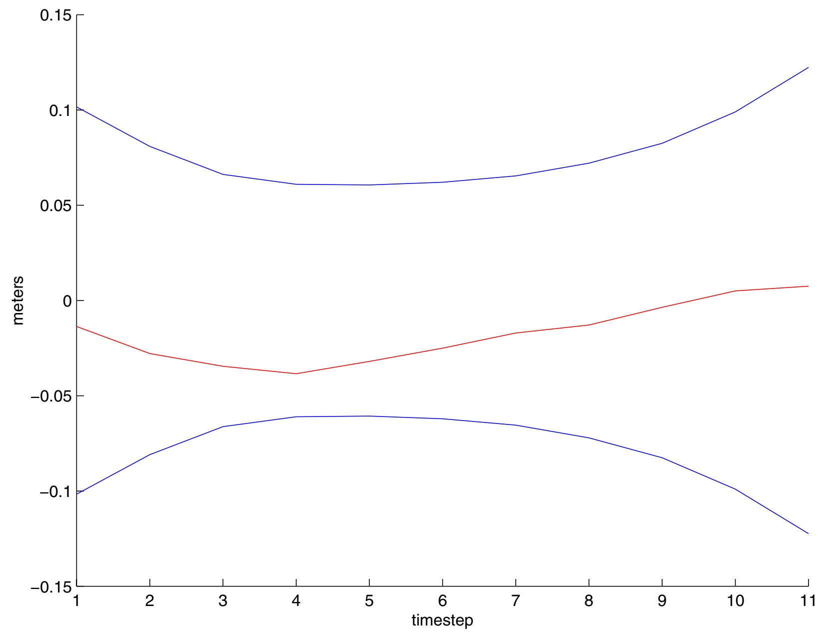
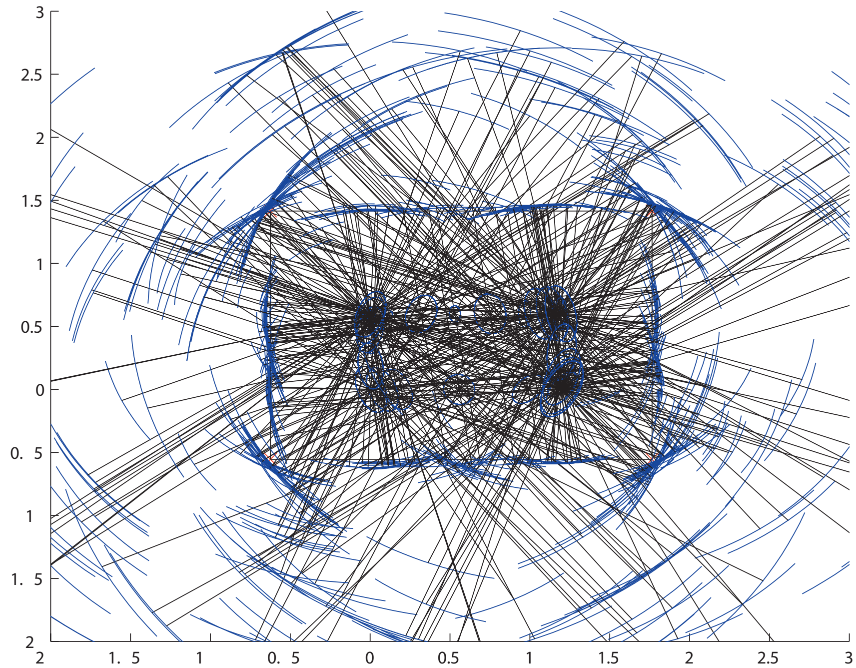
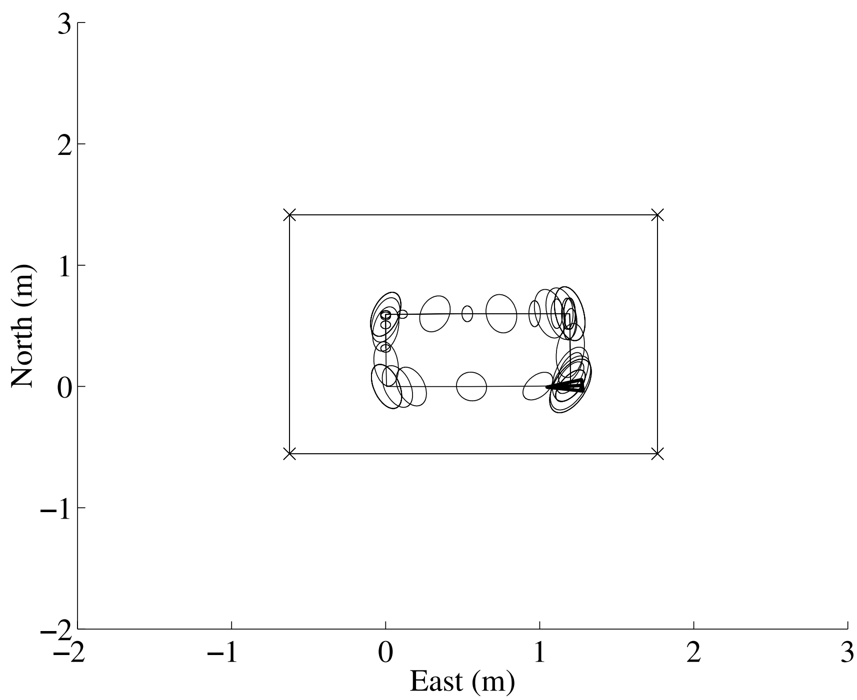


Fig. 14. Error in the smoothed estimate and three-σ confidence interval for the x position of robot 2. The minimum uncertainty is in the middle of the trajectory due to forward/backward smoothing.

Fig. 15. B21 mobile robot in the plywood box.



(a)



(b)

Fig. 16. (a) Set of all measurements processed, from 50 vehicle positions. 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 three-σ error ellipses.

prominent "crack" on the bottom wall). The nine returns and the nine features are each labeled in Figure 17, and details of initialization sequence are shown in Tables 4 and 5. Azimuth information from the sonar returns was not used for gating but not for estimation. The processing proceeded as follows. First, Returns 1 and 2 were used to initialize Corner 1 using a two position range-only initialization (circle intersection). Next, Return 3 was used in conjunction with the state estimate for Corner 1 to initialize Plane 1, and Return 4 was used in conjunction with the state estimate for Corner 1 to initialize Plane 2. After this, Return 5 was used in conjunction with the state estimate for Plane 1 to initialize Corner 2, and Return 6 was used in conjunction with the state estimate for Plane 2 to initialize Corner 3. Likewise, Return 7 was used in conjunction with the state estimate for Corner 2 to initialize Plane 3, and Return 8 was used in conjunction with the state estimate for Corner 3 to initialize Plane 4. Next, Return 9 and Plane 4 were used to initialized the crack on Plane 4. Finally, the state estimates for Plane 3 and Plane 4 were used to initialize the final feature, Corner 4.

After the initializations, nine constrained features (shown in Figure 17) were mapped using nine range measurements (shown in Figure 18). Once these features were initialized, nearest-neighbor gating was performed between all of the remaining sonar measurements and the newly initialized map features. A total of 217 of the original 600 measurements were uniquely matched to one of the nine features shown in Figure 19(a). Finally, Figure 19(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. Integrated Concurrent Mapping and Localization Results

The framework described above in Section 3 has been implemented as part of an integrated framework for real-time CML, which incorporates delayed state management, perceptual grouping, multiple vantage point initialization, batch updating, and feature fusion. Figure 20 gives an overview of the flow of information in the system.

For these experiments, a fixed size of 40 trajectory time steps was utilized. Every 40 time steps, perceptual grouping was performed using the sonar returns from the past 40 time steps. In general, a wide variety of strategies for making delayed data association decisions are possible within this framework. In this paper, we do not attempt to describe a single definitive decision-making policy, rather our goal is to illustrate the process with a few representative examples with sonar data. For the results reported here, we use a Hough transform technique documented in Tardós et al. (2001). A brief summary of the technique is as follows (Leonard, Newman, Rikoski, Neira, and Tardós 2001).

The data from a standard ring of Polaroid sonar sensors can be notoriously difficult to interpret. This leads many researchers away from a geometric approach to sonar mapping. However, using a physics-based sensor model, the geometric constraints provided by an individual sonar return can be formulated (Leonard and Durrant-Whyte 1992). Each return could originate from various types of features (point, plane, etc.) or could be spurious. For each type of feature, there is a limited range of locations for a potential feature that are possible. Given these constraints, the Hough transform (Ballard and Brown 1982) can be used as a voting scheme to identify point and planar features. More details on this technique are contained in Tardós et al. (2001). The method is similar in spirit to the TBF method of Wijk and Christensen (Wijk and Christensen 2000), but can also directly identify specular planar reflectors from sonar data, which is vitally important in typical man-made environments with many smooth walls.

The output from the Hough transform gives sets of measurements with a high likelihood to originate from a single point or plane feature. Each candidate set from the Hough typically contains between 10 and 40 sonar returns hypothesized to originate from a new feature. For each candidate set, two returns are chosen to serve as "seed" features for the initialization, to be used in the function $\mathbf{g}(\cdot)$, and the remaining returns are used in a batch update. The first of the two "seed" measurements is chosen to be the return in the candidate set that originates from the earliest vehicle position from the set of trajectory states. The other seed measurement is chosen to be the earliest return that, when combined with the first seed measurement, achieves a sufficient minimum baseline for feature initialization (typically 0.6 m). Once a new feature is initialized, it is discarded if it has too small a baseline. To successfully distinguish doors from the walls in the corridor experiment shown in Figure 1, a minimum valid line length of 1.2 m is used for adding a feature into the map. (This restriction can be removed if the joint compatibility method of Neira and Tardós (2001) is applied.)

For state estimation, we have a choice between two basic strategies: (1) attempt to match individual measurements to pre-existing features, or (2) use measurements exclusively for new feature initialization and batch updating, followed by feature fusion with previously mapped features to obtain error reduction. A hybrid strategy that mixes both policies is also possible. While we have had good success with either (1) only, (2) only, or a mix of both, in this paper we focus on option (2), namely new feature mapping followed by feature fusion; see Ayache and Faugeras (Ayache and Faugeras 1989), Chong and Kleeman (Chong and Kleeman 1997a), Tardós et al. (2001) and Williams et al. (2001) for further discussion of feature fusion. To determine when features should be fused together, we use the Mahalanobis distance and nearest neighbor.

To illustrate the performance of the implementation, we present results from two different simplified settings: one experiment with two point objects only (cylinders of known

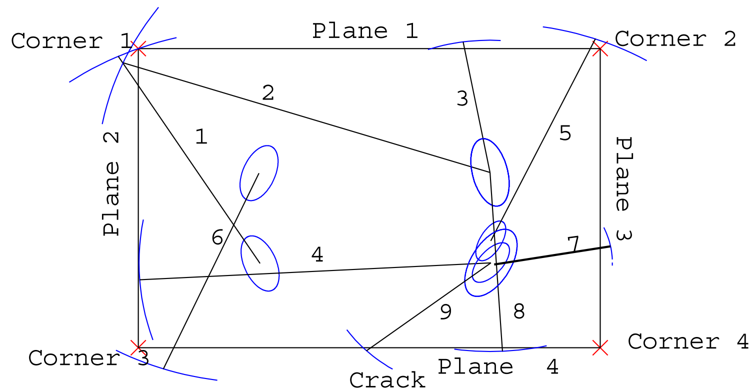


Fig. 17. Nine measurements used to initialize nine new features, starting with the corner in the upper left of the figure, and building in both directions around the room, closing the box in the lower right-hand corner. Three-σ error ellipses are shown for the dead-reckoned vehicle positions for each of the returns.

Table 4. Details for the Nine Manually-selected Returns Used for Feature Initialization

Return Number	Time Step	Odometry X (m)	Odometry Y (m)	Odometry Heading (deg) (deg)	Range (m)	Azimuth (deg)
1	27	0.0058	0.0002	−3.8264	1.5486	−0.3927
2	44	1.1949	0.5997	−7.8248	2.0303	4.3197
3	44	1.1949	0.5997	−7.8248	0.8706	3.2725
4	49	1.1998	0.0044	−8.9705	1.8202	−0.3927
5	16	1.1990	0.1490	−1.5643	1.4352	2.7489
6	36	0.0004	0.5950	−6.2571	1.3806	−1.9635
7	21	1.1995	0.0063	−3.1013	0.6280	3.2725
8	42	1.1949	0.5997	−7.1450	1.1788	−0.6545
9	50	1.1998	0.0044	−9.3471	0.8658	0.6545

Table 5. Method of Initialization for the Nine Features, and Comparison of Hand-measured and Actual Locations for the Four Corners of the Box

Feature	Initialization Method	Hand measured x	Hand measured y	CML estimated x	CML estimated y
Corner 1	Returns 1 and 2	−0.6240	1.4153	−0.6564	1.4287
Plane 1	Corner 1 and Return 3				
Plane 2	Corner 1 and Return 4				
Corner 2	Plane 1 and Return 5	1.7652	1.4153	1.7838	1.4605
Corner 3	Plane 2 and Return 6	−0.6240	−0.5550	−0.6050	−0.6243
Plane 3	Corner 2 and Return 7				
Plane 4	Corner 3 and Return 8				
Crack	Plane 4 and Return 9				
Corner 4	Plane 3 and Plane 4	1.7652	−0.5550	1.7652	−0.5550

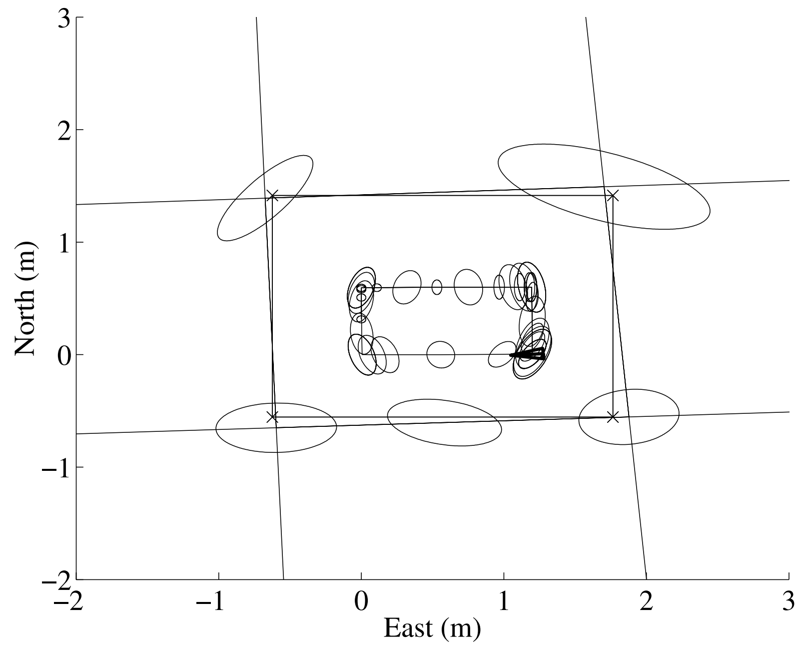


Fig. 18. State estimates and three-σ error ellipses for the nine initialized features.

radii), and one experiment in the corridor shown in Figures 1 and 2. Videos of the replay of data processing for these two experiments are accessible at www.ijrr.org.

The method has also been implemented running in real time under manual control. To our knowledge, this is the first successful feature-based CML implementation using sonar sensing for which the robot was continually in motion and the CML output was generated in real-time. (Chong and Kleeman (1997a) implemented sonar-based mapping with a high-performance sonar array that stopped to perform mechanical scanning for each data acquisition cycle.) The method uses the standard Polaroid sonar array on the B21 robot and can be readily ported to any B21 mobile robot. Such a result has not been achieved before because it has not been possible without the expanded representation accounting for temporal correlations, described in Section 3.

The method presented in this paper has also been used extensively in two other experimental systems. With sonar, using RANSAC for perceptual grouping and the ATLAS framework for scalable mapping (Bosse et al. 2002), we have mapped a large portion of the MIT campus and demonstrated closure of large loops, using only sonar and odometry data. A representative result is shown in Figure 25. We have also extended the framework presented in this paper to achieve robust three-dimensional local mapping from omnidirectional video data (Bosse et al. 2002).

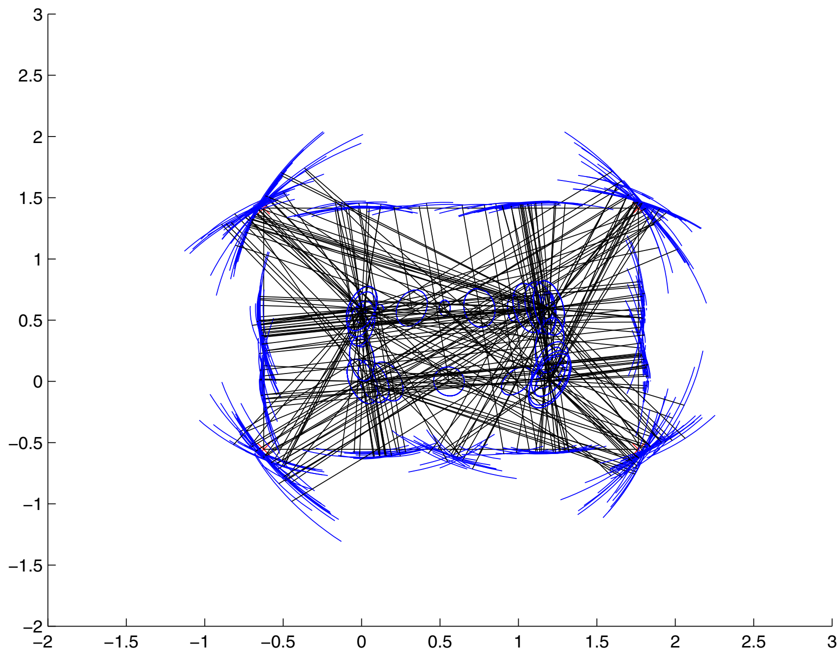
6. Conclusion

In this paper we have described a generalized framework for CML that incorporates temporal as well as spatial correlations, allowing features to be initialized from multiple uncertain vantage points. The method has been applied to Polaroid sonar data from a B21 mobile robot, demonstrating the ability to perform CML with sparse and ambiguous data. These experiments illustrate 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 and the ambiguity of individual measurements is high.

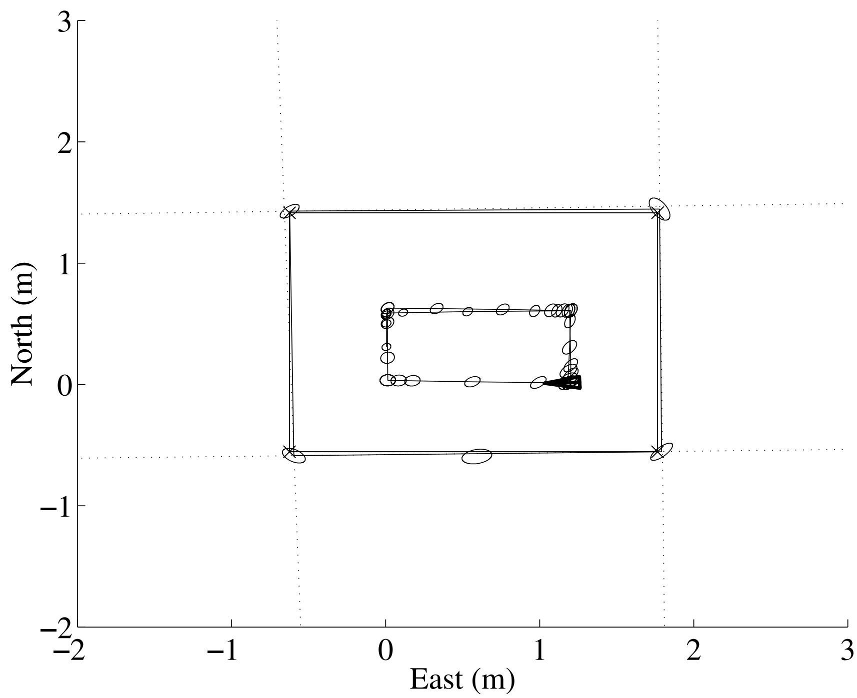
6.1. Related Research

The notion of incorporating segments of the robot trajectory in the state vector (instead of just the current robot state) employed in this paper is similar in some respects to the work of Thrun (2001) and Gutmann and Konolige (1999), which also use the vehicle trajectory as one of the key elements of the map representation. In our work, we only save partial segments of the vehicle trajectory, on an "as-needed" basis to resolve data association and feature modeling ambiguity. We believe that it is possible to pose the problem of stochastic mapping without features, using only trajectory states. The basic update operation would be to correlate the observed sensor data from one position with that observed at another position, and to formulate a measurement update function $\mathbf{h}(\cdot)$ that involves only trajectory states. For example, Carpenter and Medeiros (2001) have reported CML results using multibeam sonar images. Fleischer has employed smoothing to good effect in a stochastic framework for undersea video mosaicking (Fleischer 2000).

The methods of Thrun (2001) and Gutmann and Konolige (1999) can compute position offsets for the robot by correlat-



(a)



(b)

Fig. 19. (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.

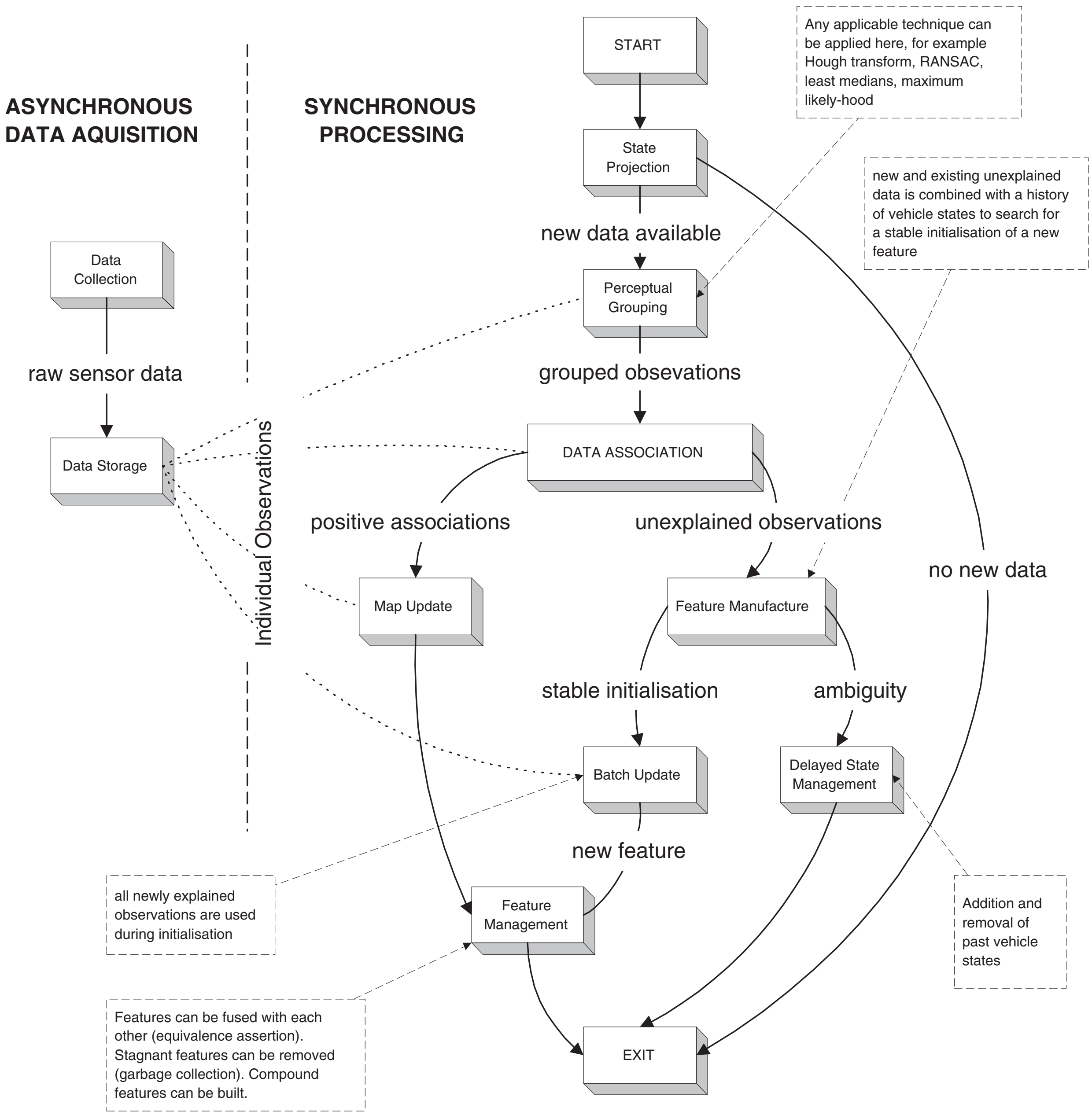


Fig. 20. Information flow for cycle integrated real-time CML incorporating delayed state management, perceptual grouping, multiple vantage point initialization, batch updating, and feature fusion.

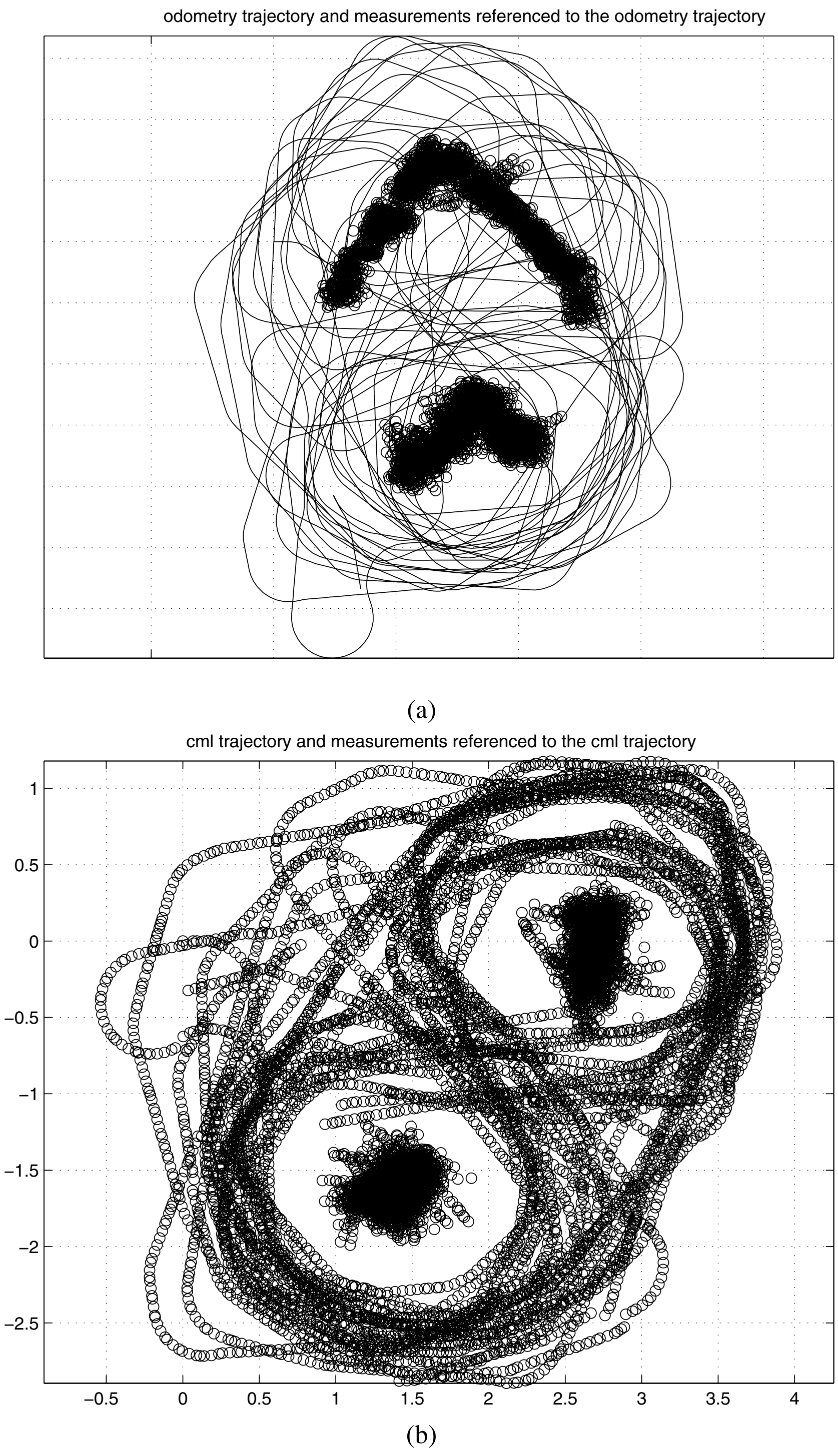


Fig. 21. (a) Raw sonar data for experiment with two point objects, referenced to odometry. (b) Sonar returns matched to the two features, referenced to the CML estimated trajectory. The experiment was 50 min long. The vehicle moved continuously under manual control at a speed of 0.1 m per second, making about 15 loops of the two cylinders. (See Extension 1.)

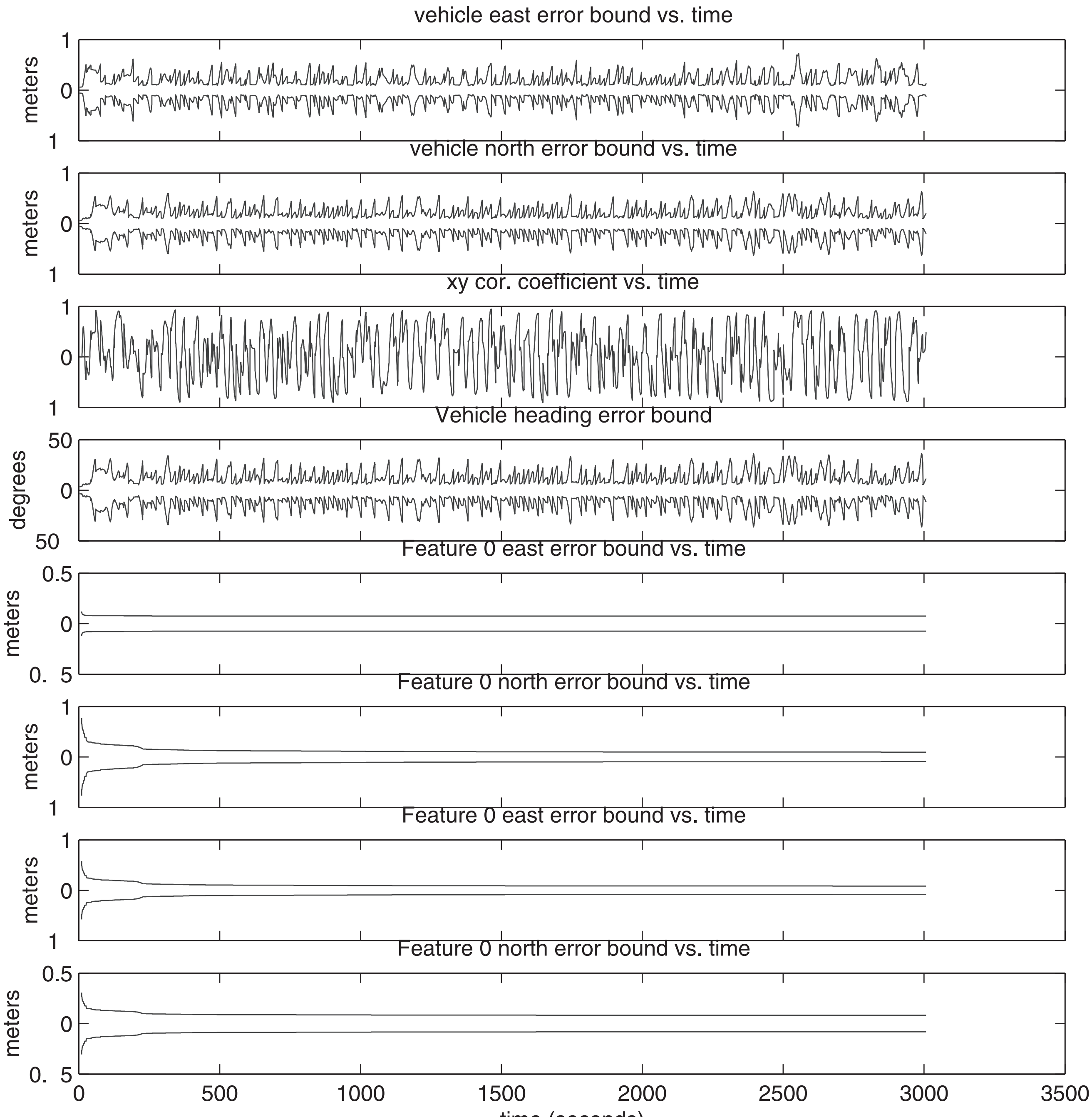


Fig. 22. Estimated error bounds for the experiment: top plots, three-σ bounds for x and y of the vehicle; next plot, x–y correlation coefficient; next plot, three-σ bounds for vehicle heading; bottom four plots, three-σ bounds for the x and y locations of the two features. There is no ground-truth for this experiment, however the vehicle returned to within a few inches of the start position, commensurate with the CML algorithm state estimation error.

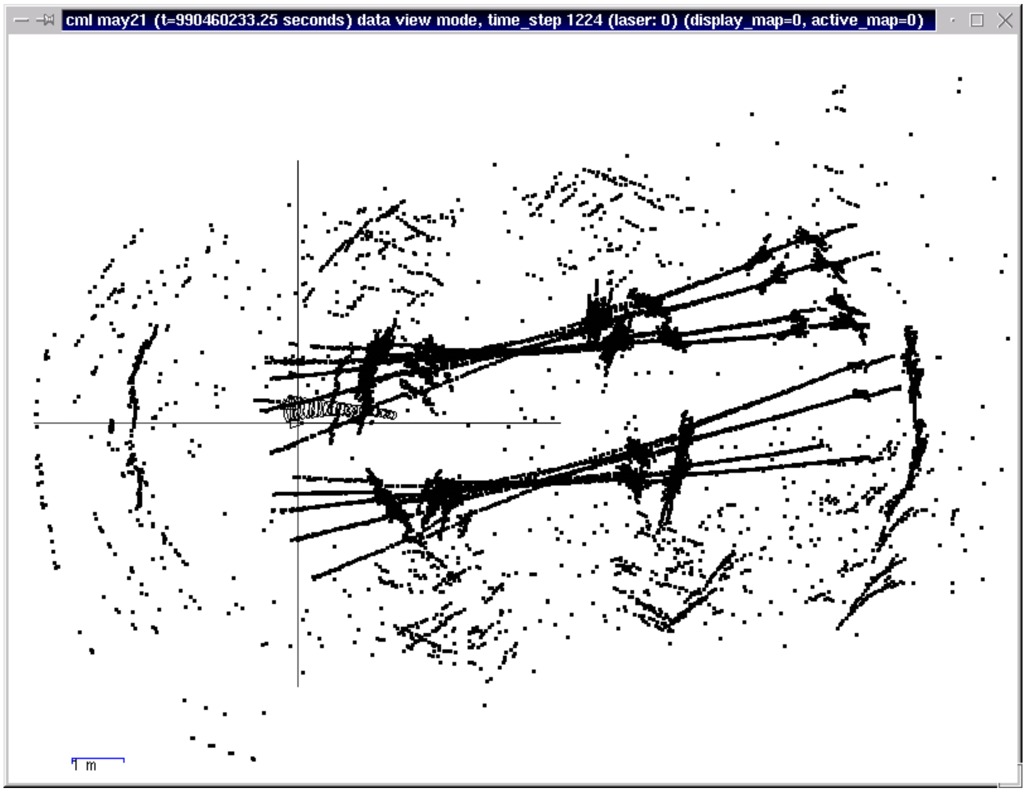


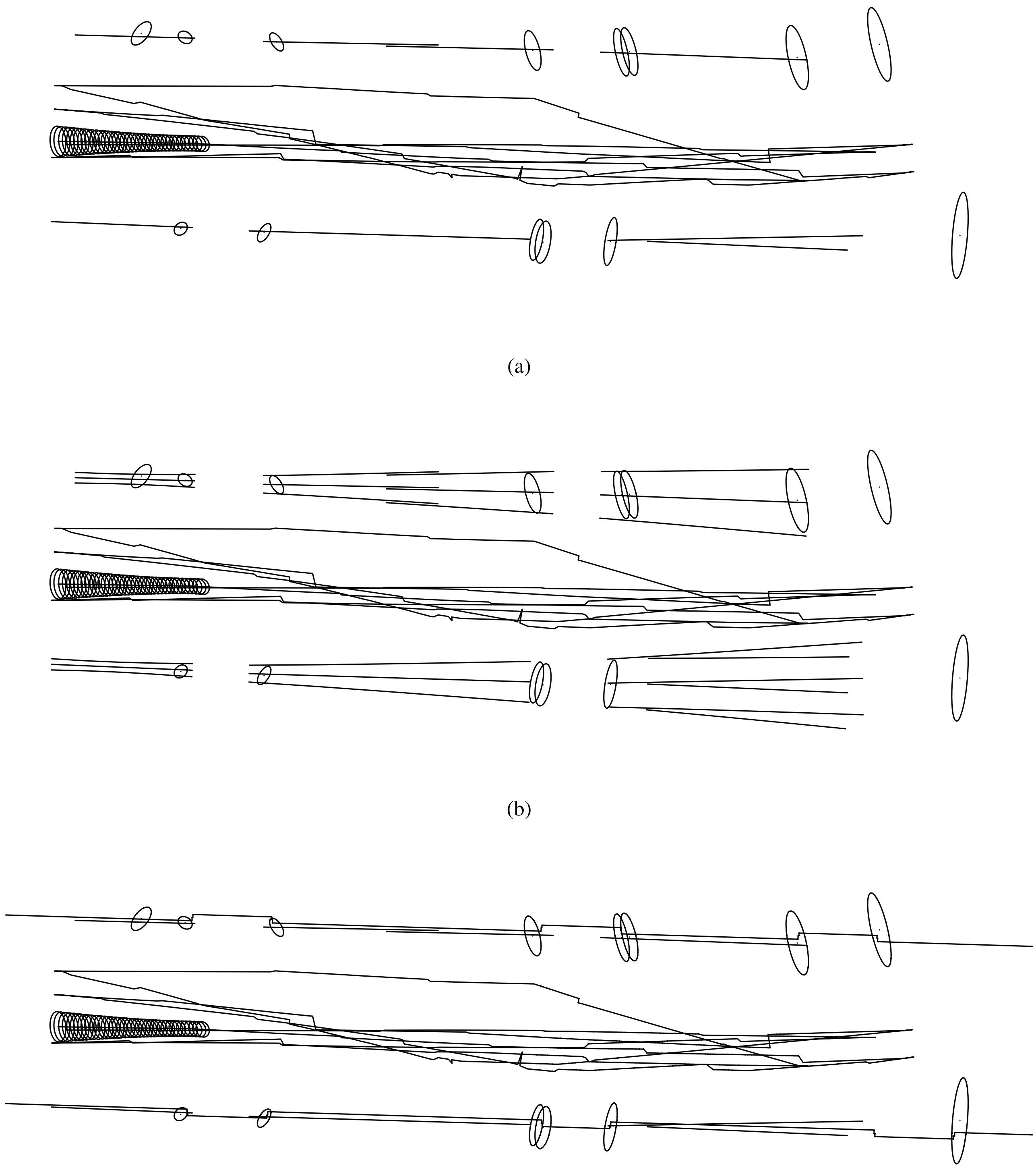
Fig. 23. Raw data for corridor experiment, referenced to odometry.

ing the current laser scans with another previously obtained laser scan. A benefit of this type of approach is that the data association problem does not need to be solved for individual sensor measurements. Very impressive experimental results have been obtained with both approaches. With sonar, the raw data is usually too noisy and ambiguous for these correlation-based approaches to work.

Recent work in feature-based CML has shown the importance of maintaining spatial correlations between the state estimates for different features, in order to maintain consistent error bounds (Castellanos and Tardós 2000; Dissanayake et al. 1999). The representation of spatial correlations results in an $\mathcal{O}(n^2)$ growth in computational cost (Moutarlier and Chatila 1989), where n is the number of features in the environment. This has motivated techniques to address the map scaling problem through spatial and temporal partitioning (Davison 1998; Leonard and Feder 2001; Guivant and Nebot 2001). In the current paper we have not addressed the map scaling problem, however the paper provides a framework for increasing the reliability of local map building. This is anticipated to greatly expand the range of environments in which CML can be successfully performed. For a given new type of environment, it is essential to establish reliable local mapping before considering the large-scale mapping problem.

An alternative to achieve the sonar mapping results presented here is to use a custom sonar array that can classify and initialize geometric primitives from a single vantage point. The state-of-the-art in this area is exhibited by the work of Kleeman et al. (Kleeman 2001; Heale and Kleeman 2001; Chong and Kleeman 1997a; Kleeman and Kuc 1993). For example, Chong and Kleeman (Chong and Kleeman 1997) have used custom advanced sonar arrays to very good effect in testing large-scale CML algorithms; since this is a scanning sonar, the robot has to stop and scan at each location. However, more recently, Heale and Kleeman (2001) have demonstrated a small, multi-element sensor that performs rapid classification to enable mapping while moving.

Nonetheless, attempting to perform CML with the standard ring of Polaroid sensors is an interesting and important problem from both a practical and a basic science perspective. The challenges of range-only interpretation explicitly capture many important uncertainty management problems posed by CML. The fundamental essence of sonar as a range-only sensor providing only sparse information is maintained in a manner that can be applied to alternative, more general situations, such as multi-robot mapping. Much further research is necessary to extend the approach to complex environments, such as the mapping of underwater terrain; however, we anticipate the generalized framework for CML presented here to be broadly applicable in a variety of environments.



(a)

(b)

(c)

Fig. 24. (a) CML estimated trajectory for corridor scene and estimated map consisting of points and line segments. Three-σ error bounds are shown for the location of points. (b) Same plot as in (a), but with three-σ error bounds for lines added. (c) Same plot as in (a), but with hand-measured model overlaid. (See Extensions 2 and 3.)

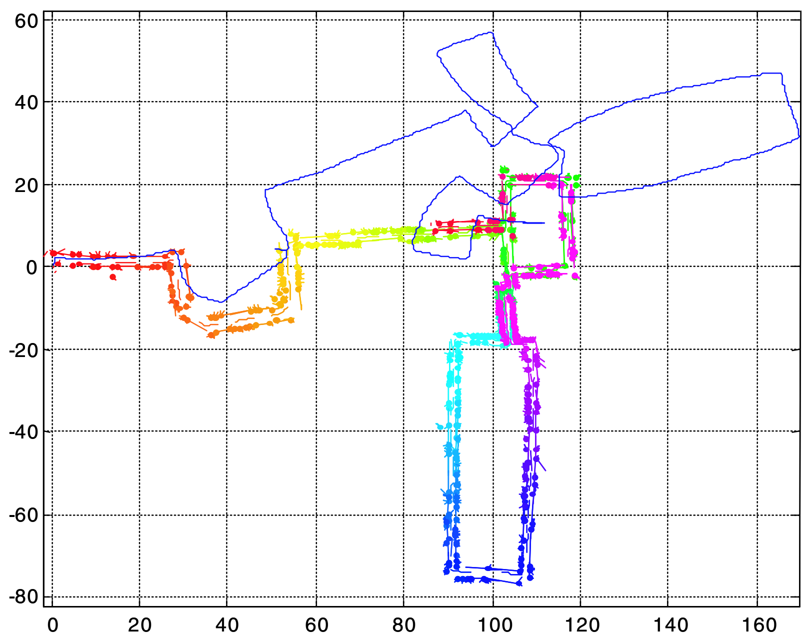


Fig. 25. Map produced from B21 sonar data for several corridors of the MIT campus, created using the state estimation framework described in Section 3 combined with RANSAC for perceptual grouping and the ATLAS framework for large-scale mapping (Bosse et al. 2002). The mission was 50 min in duration and the vehicle traveled a distance of 481 m, with a peak velocity of 0.3 m per second and an average velocity of 0.163 m per second. (See Extensions 4 and 5.)

6.2. Future Work

A number of important topics warrant consideration in future work. In the results presented in this paper, new features were mapped using an initialization function $\mathbf{g}(\cdot)$ that determined the location of a new map feature location as an explicit function of measurements from multiple uncertain vantage points. An alternative is to use non-linear least squares performed on many measurements, such as in a similar manner to bundle adjustment (Triggs et al. 2000). Faugeras summarizes the necessary formulae for computing the covariance matrix to accompany the state estimate derived from the least-squares optimization (Faugeras 1993), and this can be readily incorporated into the stochastic map.

Another interesting topic is adaptive feature initialization and the integration of CML with closed-loop trajectory control. Strategies should be developed for controlling the robot during the initialization of a feature, and in selecting which measurements to use for performing the initialization. We can incorporate an adaptive motion control step to direct the robot to move to a better vantage point that will improve the information available to the robot in attempting to initialize a feature. 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. It would be interesting to couple this back to control the robot's motion for data acquisition.

It is currently unclear how well the technique will perform in situations with extremely poor dead-reckoning. As long as the linearization assumptions of the EKF are satisfied, then the state estimation framework presented here should be expected to provide consistent initialization despite dead-reckoning error, by maintaining correlations between past and current vehicle states. However, we expect any approach that uses the EKF to fail when very large angular errors are encountered. In addition, the task of perceptual grouping is reliant to some extent on reasonably accurate dead-reckoning. Future work is necessary to quantify these relationships over a range of different operating conditions (e.g., environment with sparse features).

In ongoing research, we are extending the framework presented in this paper to enable large-scale autonomous exploration of unknown environments (Bosse et al. 2002; Newman, Bosse, and Leonard 2002). The approach is also being extended to enable real-time cooperative navigation by multiple AUVs (Fenwick, Newman, and Leonard 2002).

Appendix A: Functions for Initialization of Features from Multiple Vantage Points

In this appendix we describe functions for initialization of point and line measurements from multiple positions.

1.1. Initializing a Point from Two Range Measurements

Observing the range from the robot to a point defines a circle which the point must lie upon. Two observations define two circles, the intersection of which defines two points. The ambiguity is resolved through the use of additional information, usually a third range measurement or a beamwidth constraint. If the robot observes a point feature (x, y) from vantage points (x_1, y_1) and (x_2, y_2), measuring ranges r_1 and r_2, two circles can be defined:

$$(x - x_1)^2 + (y - y_1)^2 - r_1^2 = 0 \tag{28}$$

$$(x - x_2)^2 + (y - y_2)^2 - r_2^2 = 0. \tag{29}$$

The solution can be computed as follows

$$x_p = \frac{\mp (y_2 - y_1)\,\gamma}{d^2} - \frac{\left(r_2^2 - r_1^2\right)(x_2 - x_1)}{d^2} + \frac{x_1 + x_2}{2} \tag{30}$$

$$y_p = \frac{\pm (x_2 - x_1)\,\gamma}{d^2} - \frac{\left(r_2^2 - r_1^2\right)(y_2 - y_1)}{d^2} + \frac{y_1 + y_2}{2} \tag{31}$$

where

$$\gamma = \sqrt{\left((r_2 + r_1)^2 - d^2\right)\left(d^2 - (r_2 - r_1)^2\right)} \tag{32}$$

and

$$d^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 . \tag{33}$$

If γ is imaginary, then the circles do not intersect; see also Wijk and Christensen (2000).

1.2. Initializing a Line from Two Range Measurements

There can be as many as four lines which are tangent to two circles. Considering only the cases where the two circles are tangent to the same side of the line, the two solutions are

$$\theta = \arctan \frac{\pm (x_2 - x_1)\,\gamma}{\mp (y_2 - y_1)\,\gamma - (x_2 - x_1)(r_2 - r_1)} \tag{34}$$

$$\rho = \frac{\mp (x_1 y_2 - x_1 y_1)\,\gamma + (y_2 - y_1)(r_1 y_2 - r_2 y_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{35}$$

where

$$\gamma = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 - (r_2 - r_1)^2}, \tag{36}$$

and θ designates the angle of the normal of the line and ρ is the perpendicular offset of the line from the origin. When γ is imaginary then the circles are concentric and do not have a cotangent.

1.3. Initializing a Line from a Range Measurement and a Colinear Point

Initializing the line which is tangent to circle (x_1, y_1, r_1) and passes through point (x_2, y_2) is equivalent to finding the line which is tangent to two circles when one of the circles has zero radius. There are two solutions. The two results, (ρ_1, θ_1) and (ρ_2, θ_2), are

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{37}$$

$$\theta_{pc} = \arctan 2\,(y_2 - y_1, x_2 - x_1) \tag{38}$$

$$\beta = \arccos\left(\frac{r}{d}\right) \tag{39}$$

$$\theta_1 = \alpha + \beta \tag{40}$$

$$\theta_2 = \alpha - \beta \tag{41}$$

$$\begin{bmatrix} x_{c1} \\ y_{c1} \end{bmatrix} = \begin{bmatrix} x_1 + r_1 \cos\theta_1 \\ y_1 + r_1 \sin\theta_1 \end{bmatrix} \tag{42}$$

$$\begin{bmatrix} x_{c2} \\ y_{c2} \end{bmatrix} = \begin{bmatrix} x_1 + r_1 \cos\theta_2 \\ y_1 + r_1 \sin\theta_2 \end{bmatrix} \tag{43}$$

$$\alpha_1 = \sqrt{x_{c1}^2 + y_{c1}^2} \tag{44}$$

$$\alpha_2 = \sqrt{x_{c2}^2 + y_{c2}^2} \tag{45}$$

$$\beta_1 = \arctan 2\,(y_{c1}, x_{c1}) \tag{46}$$

$$\beta_2 = \arctan 2\,(y_{c2}, x_{c2}) \tag{47}$$

$$\rho_1 = \alpha_1 \cos(\theta_1 - \beta_1) \tag{48}$$

$$\rho_2 = \alpha_2 \cos(\theta_2 - \beta_2) . \tag{49}$$

1.4. Initializing a Point from a Line and a Range Measurement

A robot position and a range define a circle (x, y, r). Provided that the circle intersects the line (ρ, θ), we want to find the intersection points. First, we find the distance from the center of the circle to the line, which is defined as

$$d = \left|\rho - x \sin(\theta) - y \cos(\theta)\right| . \tag{50}$$

This is the first leg of a right triangle. The hypotenuse is the range measurement. The angle between the two is

$$\beta = \arccos\left(\frac{d}{r}\right). \tag{51}$$

Knowing the bearing to the line is, by definition, θ, the two intersections are therefore

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x + \rho\cos(\theta - \beta) \\ y + \rho\sin(\theta - \beta) \end{bmatrix} \tag{52}$$

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x + \rho\cos(\theta + \beta) \\ y + \rho\sin(\theta + \beta) \end{bmatrix}. \tag{53}$$

Appendix B: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Sonar experiment with two objects
2	Video	Measurement processing for corridor experiment
3	Video	Mapped features for corridor experiment
4	jpg	Map and odometry trajectory for large-scale experiment
5	jpg	Perceptual grouping output for large-scale experiment using RANSAC

Acknowledgments

This research has been funded in part by NSF Career Award BES-9733040, the MIT Sea Grant College Program under grant NA86RG0074 (project RCM-3), and the Office of Naval Research under grant N00014-97-0202. The authors would like to thank J.D. Tardós and J. Neira for helpful discussions and the contribution of the Hough transform feature classification technique to the results reported in Section 5.

References

Anderson, B. D. O., and Moore, J. B. 1979. *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall.

Au, W. 1993. *The Sonar of Dolphins*. New York: Springer-Verlag.

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

Ballard, D., and Brown, C. 1982. *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall.

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

Bosse, M., Newman, P., Leonard, J., and Teller, S. 2002a. An atlas framework for scalable mapping. Technical Report Marine Robotics Laboratory Technical Memorandum 02-04, Massachusetts Institute of Technology.

Bosse, M., Rikoski, R., Leonard, J., and Teller, S. September 2002b. Vanishing points and 3d lines from omnidirectional video. In *Int. Conf. on Image Processing*, Rochester, NY.

Brooks, R. A. 1984. Aspects of mobile robot visual map making. In *2nd Int. Symp. Robotics Research*, Tokyo, Japan, pp. 287–293. MIT Press.

Bucy, R. S., and Senne, K. D. 1971. Digital synthesis of non-linear filters. *Automatica* 7:287–298.

Carpenter, R. N. 2001. Concurrent mapping and localization and map matching on autonomous underwater vehicles. In *IEEE Oceans*, Hawaii, pp. 380–389.

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

Chiuso, A., Favaro, P., Jin, H., and Soatto, S. 2000. 3-d motion and structure from 2-d motion causually integrated over time: Implementation. In *6th European Conference on Computer Vision*.

Chong, K. S., and Kleeman, L. 1997a. Sonar based map building in large indoor environments. Technical Report MECSE-1997-1, Department of Electrical and Computer Systems Engineering, Monash University.

Chong, K. S., and Kleeman, L. 1997b. Sonar feature map building for a mobile robot. In *Proc. IEEE Int. Conf. Robotics and Automation*.

Choset, H., and Nagatani, K. April 2001. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Trans. Robotics Automation* 17(2):125–137.

Cox, I. J., and Leonard, J. J. April 1994. Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence* 66(2):311–344.

Davison, A. J. 1998. *Mobile Robot Navigation Using Active Vision*. Ph. D. thesis, University of Oxford.

Deans, M. May 2002. Maximally informative statistics for localization and mapping. In *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1824–1829.

Deans, M., and Hebert, M. December 2000. Experimental comparison of techniques for localization and mapping using a bearings only sensor. In *Proc. ISER '00 7th International Symposium on Experimental Robotics*.

Dissanayake, M. W. M. G., Newman, P., Durrant-Whyte, H. F., Clark, S., and Csorba, M. March 1999. An experimental and theoretical investigation into simultaneous localization and map building. In *6th Int. Symposium on Experimental Robotics*, pp. 265–274.

Dissanayake, M. W. M. G., Newman, P., Durrant-Whyte, H. F., Clark, S., and Csorba, M. June 2001. A solution to the simultaneous localization and map building (slam) problem. *IEEE Trans. Robotics Automation* 17(3):229–241.

Doucet, A., de Freitas, N., and Gordan, N., eds. 2001. *Sequential Monte Carlo Methods in Practice*. Berlin: Springer-Verlag.

Durrant-Whyte, H. F., Julier, S., and Uhlmann, J. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Translations on Automatic Control* 45(3):477–482, March 2000.

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

Faugeras, O., Luong, Q.-T., and Papadopoulo, T. 2001. *The Geometry of Multiple Images*. MIT Press.

Feder, H. J. S. 1999. *Simultaneous Stochastic Mapping and Localization*. Ph. D. thesis, Massachusetts Institute of Technology.

Feder, H. J. S., Leonard, J. J., and Smith, C.M. July 1999. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research* 18(7):650–668.

Fenwick, J., Newman, P., and Leonard, J. May 2002. Collaborative concurrent mapping and localization. In *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, pp. 1810–1817.

Fischler, M. A., and Bolles, R. C. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24(6):381–395.

Fleischer, S. D. 2000. *Bounded-Error Vision-Based Navigation of Autonomous Underwater Vehicles*. Ph. D. thesis, Stanford University.

Grimson, W. E. L. 1990. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press. (With contributions from T. Lozano-Perez and D. P. Huttenlocher).

Guivant, J., and Nebot, E. June 2001. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Trans. Robotics Automation* 17(3):242–257.

Gutmann, J.-S., and Konolige, K. 1999. Incremental mapping of large cyclic environments. In *Int. Symposium on Computational Intelligence in Robotics and Automation*.

Hartley, R. I., and Zisserman, A. 2001. *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press.

Heale, A., and Kleeman, L. 2001. Fast target classification using sonar. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pp. 1446–1451.

Jensfelt, P. 2001. *Approaches to Mobile Robot Localization in Indoor Environments*. Ph. D. thesis, Royal Institute of Technology, Stockholm, Sweden.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. August 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robotics Automation* 12:566–580.

Kleeman, L. 2001. Advanced sonar sensing. In R. Jarvis and A. Zelinsky, eds., *Robotics Research: The 10th Int. Symposium*, Lorne, Australia. Berlin: Springer Verlag.

Kleeman, L., and Kuc, R. 1993. Mobile robot sonar for target localization and classification. Technical Report ISL-9301, Intelligent Sensors Laboratory, Yale University.

Kuc, R. 1996. Fusing binaural sonar information for object recognition. In *IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 727–735.

Kuipers, B. J. 2000. The spatial semantic hierarchy. *Artificial Intelligence*.

Kurien, T. 1990. Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*, pp. 43–83. Boston: Artech House.

Leonard, J. J., Cox, I. J., and Durrant-Whyte, H. F. August 1992. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research* 11(4):286–298.

Leonard, J. J., and Durrant-Whyte, H. F. 1992. *Directed Sonar Sensing for Mobile Robot Navigation*. Boston: Kluwer Academic Publishers.

Leonard, J. J., and Feder, H. J. S. 2001. Decoupled stochastic mapping. *IEEE J. Ocean Eng.* 26(4):561–571.

Leonard, J. J., Newman, P., Rikoski, R., Neira, J., and Tardós, J. 2001. Towards robust data association and feature modeling for concurrent mapping and localization. In R. Jarvis and A. Zelinsky, eds., *Robotics Research: The 10th Int. Symposium*, Lorne, Australia. Berlin: Springer Verlag.

MacCormick, J. (2000). *Probabilistic modelling and stochastic algorithms for visual localisation and tracking*. Ph. D. thesis, University of Oxford.

McLauchlan, P. F. 2000. A batch/recursive algorithm for 3d scene reconstruction. In *Int. Conf. Computer Vision and Pattern Recognition*, Volume 2, Hilton Head, SC, pp. 738–743.

McLauchlan, P. F. and Murray, D. W. 1995. A unifying framework for structure and motion recovery from image sequences. In *Proc. Int. Conference on Computer Vision*, Boston, MA, pp. 314–320.

McLauchlan, P. F. and Murray, D. W. January 1996. Active camera calibration for a head-eye platform using the variable state-dimension filter. *IEEE Trans. Pattern Anal. Mach. Intell.* 18(1):14–22.

Mori, S., Chong, C., Tse, E., and Wishner, R. May 1986. Tracking and classifying multiple targets without a priori identification. *IEEE Trans. Automatic Control* 31(5).

Moutarlier, P., and Chatila, R. 1989. Stochastic multi-sensory data fusion for mobile robot location and environment modeling. In *5th Int. Symposium on Robotics Research*, pp. 207–216.

Neira, J., and Tardós, J. 2001. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics Automation* 17(6):890–897.

Newman, P., Bosse, M., and Leonard, J. 2002. Autonomous feature-based exploration. Technical Report Marine Robotics Laboratory Technical Memorandum 02-03, Massachusetts Institute of Technology.

Newman, P., Leonard, J., Neira, J., and Tardós, J. May 2002. Explore and return: Experimental validation of real time concurrent mapping and localization. In *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, pp. 1802–1809.

Reid, D. B. December 1979. An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control* 24(6):843–854.

Schmidt, H. 1998. GOATS-98 — AUV network sonar concepts for shallow water mine countermeasures. Technical Report SACLANTCEN SR-302, SACLANT Undersea Research Centre.

Schultz, A. C. and Adams, W. 1998. Continuous localization using evidence grids. In *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2833–2839.

Smith, R., Self, M., and Cheeseman, P. 1987. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research*. MIT Press.

Sorenson, H. W. 1988. Recursive estimation for nonlinear dynamic systems. In J. C. Spall, ed., *Bayesian Analysis of Time Series and Dynamic Models*, pp. 127–166. New York: Marcel Dekker.

Tardós, J., Neira, J., Newman, P. M., and Leonard, J. J. 2001. Robust mapping and localization in indoor environments using sonar data. Technical Report Marine Robotics Laboratory Technical Memorandum 2001-4, MIT. See http://oe.mit.edu/~jleonard/pubs/tardos_etal_ijrr_2002.pdf

Taylor, C. J., and Kriegman, D. J. November 1995. Structure and motion from line segments in multiple images. *IEEE Trans. Pattern Anal. Mach. Intell.* 17(11):1021–1032.

Thrun, S. May 2001. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research* 20(5):335–363.

Triggs, W., McLauchlan, P., Hartley, R., and Fitzgibbon, A. 2000. Bundle adjustment for structure from motion. In B. Triggs, A. Zisserman, and R. Szeliski, eds., *Vision Algorithms: Theory and Practice*, Vol. 1883 of Lecture Notes in Computer Science, pp. 298–372. Berlin: Springer-Verlag.

Wijk, O., and Christensen, H. December 2000. Triangulation based fusion of sonar data with application in robot pose tracking. *IEEE Trans. Robotics Automation* 16(6):740–752.

Williams, S. 2001. *Efficient solutions to autonomous navigation and mapping problems*. Ph. D. thesis, University of Sydney.

Williams, S., Dissanayake, G., and Durrant-Whyte, H. June 2001. Constrained initialisation of the simultaneous localisation and mapping algorithm. In *Field and Service Robot 2001*, Finland, pp. 309–314. ISBN 952-5183-13-0.

Yagi, Y., Shouya, K., and Yachida, M. May 2000. Environmental map generation and ego-motion estimation in a dynamic environment for an omni-directional image sensor. In *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3493–3498.

Zunino, G. and Christensen, H. I. July 2001. SLAM in realistic environments. In M. Devy, ed., *Symposium on Intelligent Robotic Systems*, Toulouse, France.