MIT Marine Robotics Laboratory Technical Memorandum 01-4

Robust Mapping and Localization in Indoor Environments using Sonar Data *

Juan D. Tardós, José Neira
Dept. Informática e Ingeniería de Sistemas,
Universidad de Zaragoza
María de Luna 3, E-50015 Zaragoza, Spain
tardos@posta.unizar.es, jneira@posta.unizar.es

Paul M. Newman, John J. Leonard
MIT Dept. of Ocean Engineering
77 Massachusetts Av., Cambridge, MA 02139-4307 USA
pnewman@mit.edu, jleonard@mit.edu

December 5, 2001

* Submitted to International Journal of Robotics Research on December 5, 2001
Abstract

In this paper we describe a new technique for the creation of feature-based stochastic maps using standard Polaroid sonar sensors. The fundamental contributions of our proposal are: (1) a perceptual grouping process that permits the robust identification and localization of environmental features, such as straight segments and corners, from the sparse and noisy sonar data; (2) a map joining technique that allows the system to build a sequence of independent constant-size stochastic maps and join them in a globally consistent and optimal way; (3) a robust mechanism to determine which features in a stochastic map correspond to the same environment feature, allowing the system to update the stochastic map accordingly, and perform tasks such as revisiting and loop closing. We demonstrate the practicality of this approach by building a geometric map of a medium size, real indoor environment, with several people moving around the robot. Maps built from laser data for the same experiment are provided for comparison.

1 Introduction

The problem of concurrent mapping and localization (CML) for an autonomous mobile robot is stated as follows: starting from an initial position, a mobile robot travels through a sequence of positions and obtains a set of sensor measurements at each position. The goal is for the mobile robot to process the sensor data to produce an estimate of its position while concurrently building a map of the environment. The problem of CML, also referred to as SLAM (simultaneous localization and map building), presents a number of difficult issues, including (1) efficient mapping of large-scale environments, (2) correct association of measurements, and (3) robust estimation of map and vehicle trajectory information. The paper presents contributions to each of these three areas.

1.1 Choice of Representation

As with many problems in robotics and artificial intelligence, the issue of choosing a representation is perhaps the key step in developing an effective CML solution. A central requirement is the ability to represent uncertainty (Brooks 1984, Lozano-Pérez 1989). Popular choices for the map representation include grid-based (Elles 1987, Shultz and Adams 1998), topological (Kuipers 2000, Choset and Nagatani 2001), feature-based models (Moutarlier and Chatila 1989, Ayache and Faugeras 1989), and sequential Monte Carlo methods (Thrun 2001, Doucet, de Freitas and Gordon 2001).

This paper adopts a feature-based approach to CML, in which the locations of geometric features in the environment and the position of the vehicle are jointly estimated in a stochastic framework (Smith, Self and Cheeseman 1988, Moutarlier and Chatila 1989). CML is cast as a variable-dimension state estimation problem in which the size of the state space is increased or decreased as features are added or removed from the map. As the robot moves through its
environment, it uses new sensor measurements to perform two basic operations: (1) adding new features to its state vector, and (2) updating concurrently its estimate of its own state and the locations of previously observed features in the environment.

Related recent research by ourselves and others that adopts a similar perspective on the problem can be found in Feder, Leonard and Smith (1999), Castellanos and Tardós (1999), Dissanayake, Newman, Durrant-Whyte, Clark and Csurba (2001) and Guivant and Nebot (2001). Alternative approaches include the work of Lu and Milios (1997a), Thrun (2001) and Gutmann and Konolige (1999). These methods do not need to explicitly associate individual measurements with features in the environment, but seem to rely on the high quality of laser scanner data. Thrun (2001) writes “It is unclear how the performance of our approach degrades with inaccuracy of the sensors. For example, it is unclear if sonar sensors are sufficiently accurate to yield good results”. In this paper, we aim to show that a feature-based, geometric approach to CML can be achieved with sparse and noisy air sonar data.

1.2 The Sonar Mapping Problem

Most would agree that mobile robot navigation and mapping in indoor environments is much more difficult to perform with sonar than with laser data. Figure 1 provides a comparison of data from a SICK laser scanner and a ring of 24 sonar sensors taken at a single position in a typical environment. The lack of information in the sonar data in comparison to the laser data is evident: only half of the Polaroid sensors obtain a return with a high proportion of outliers. As a result, the underlying structure of the scene is less visually apparent to a human observer. Despite the increased difficulty of sonar interpretation, we feel that it is interesting to perform research with sonar for a variety of reasons. From the perspective of cost, laser scanners are much more expensive than sonar sensors. From the perspective of basic science, questions such as the basic mechanisms of bat and dolphin echolocation are highly important (Au 1993). Finally, the fundamental characteristics of high range accuracy and wide beamwidth are shared with many types of sonars of which underwater mapping sonars are a prime example. Our long-term goal is to develop interpretation methods that are applicable to a wide variety of different types of sonars and environments.

The information given by laser sensors in a single scan is quite dense and has good angular precision, usually better than one degree (see figure 1). Feature-based approaches to CML using laser data typically perform the data association task in two steps:

1. A segmentation step, where the laser returns in each scan are grouped and processed to obtain simple geometric features such as lines corresponding to walls (Castellanos, Montiel, Neira and Tardós 1999) or circles corresponding to trees (Guivant and Nebot 2001). Also more complex features as corners or door frames can be easily obtained from laser data (Castel-
Figure 1: Sensor information obtained from a single robot position in a typical environment using a 180 degree SICK laser scanner and a ring of 24 Polaroid sonar sensors. The true environment map, obtained by hand, is shown in dotted lines.
ianos, Neira and Tardós 2001).

2. During the map building process, a second data association step looks for matches between the features obtained from different scans, based on a probabilistic model of the sensor and of the vehicle motion.

Other approaches do not rely on the assumption that a certain type of geometric feature will be present in the environment, and use raw laser data in the map building process. The data association is solved implicitly by computing the robot locations that maximize scan-to-scan (Lu and Milios 1997b, Gutmann and Konolige 1999) or scan-to-map (Thrun 2001) correlations. In these approaches, the density and precision of the laser data is central in achieving robustness in the scan matching process.

In contrast, information given by a ring of sonar sensors is sparse and its angular precision is limited by the large sonar beamwidth, typically in the range of 20 to 30 degrees. In man-made environments, most surfaces are specular reflectors for sonar (Kuc and Siegel 1987). This has two undesirable effects: only surfaces whose perpendicular is inside the sonar beam are usually detected, and many ghost returns are obtained as a result of multiple specular reflections. A rotating sonar scanner has been used by Leonard and Durrant-Whyte (1992) to obtain regions of constant depth (RCDs), allowing to determine with good precision the location of simple features as planes, corners, edges and cylinders. However, from a single scan, the RCDs produced by edges and corners are indistinguishable from the RCDs produced by planes.

One avenue for research to overcome difficulties with sonar data is to develop advanced sensor arrays that allow feature discrimination and precise localization from a single vehicle location. For example, Barshan and Kuc (1990) developed an intelligent sonar sensor that was able to disambiguate plane and corner reflectors using amplitude and travel time information. Peremans, Audenaert and Van (1993) developed a trinaural sensor that could measure object curvature. One of the most notable land robot sonars in recent years was developed by Kleeman and Kuc (1995), who were able to achieve remarkably precise localization and classification of targets from a rotating sensor array. This sensor was used in a CML implementation to perform large-scale CML using multiple submaps (Chong and Kleeman 1999a, Chong and Kleeman 1999b). All of these advanced sensors are based on the use of several sonar transducers in a careful configuration with known baselines. The different echoes received by several transducers from a single target allow the determination of the feature type and its precise location. With accurate knowledge of the relative positions of transducers, a variety of powerful array processing algorithms already developed for radar and underwater sonar become possible (Johnson and Dudgeon 1993).

In some ways, the use of multiple transducers in a sonar array is similar to the use of several cameras in classical stereo vision: the use of different viewpoints with precisely calibrated baselines allows the computation of depth information to permit estimation of the 3D structure of the environment (Faugeras 1993). However, a great deal of research in the last decade has shown that, using a single moving camera, it is possible to determine both the camera motion and the
environment structure (Hartley and Zisserman 2000, Faugeras and Luong 2001). Ortin and Montiel (2001) present an interesting example of the use of these techniques to determine the 2D robot motion in an indoor environment using robust techniques to find matchings in the sequence of images. This parallelism with vision techniques raises a fundamental question: is it possible to determine the vehicle motion and the environment structure with a sonar scanner or a sonar ring? In this paper, we aim to pose the problem of interpretation of range-only data from multiple uncertain vantage points, and to present an effective solution for the common scenario of a land robot equipped with a ring of Polaroid sensors.

A somewhat related technique called triangulation-based fusion has been very recently developed by Wijk and Christensen (2000) for point objects only. This approach has been documented with extensive experimental results, including grid-based mapping, continuous localization using Kalman filtering, and absolute relocation using particle filters in a large indoor environment (Wijk 2001). We feel that the Hough transform approach presented in this paper offers advantages over triangulation-based fusion because it can directly identify specular planar reflectors from sonar data, which is vitally important in typical man-made environments with many smooth walls. In addition, Wijk’s approach has yet to be utilized for CML.

The structure of this paper is as follows: Section 2 explores the minimal necessary conditions allowing determination of robot motion and environment structure from range-only sonar data. Section 3 presents a new technique for classification of points and line segments from sonar data acquired from multiple vantage points, using the Hough transform. Section 4 presents a new technique for joining and updating two submaps when performing CML, providing increased computational efficiency and robustness. Section 5 illustrates the application of these techniques to data acquired by a mobile robot in an environment consisting of several loops, with spurious data from people walking near the robot. A side-by-side comparison is provided between the map built from sonar and a map built from laser scanner data. Finally, Section 6 draws conclusions and discusses future research topics.

2 Structure and motion from sonar

In this section we will analyze the conditions under which a sonar scanner or a sonar ring mounted on a vehicle gives enough information to determine both the vehicle motion and the environment structure. Finding a general solution to this problem is an interesting research topic, beyond the scope of this work. Our goal is to determine the potential of using the sonar data to solve the CML problem, where an estimation of robot motion is available through the use of odometry sensors.
2.1 The structure and motion problem

For the purpose of this analysis we will first consider the case where data association is solved, i.e. we know which environment feature has given each sonar return. Assume a robot moves along $k$ locations, and at each location observes $n$ geometric features with some of the sensors in a sonar ring. Let us call $r$ the degrees of freedom (d.o.f) of robot motion and $f$ the d.o.f. that determine each feature location. Without loss of generality, we will use the first robot location as a reference system. The number of unknowns is $r(k - 1)$ for the robot motion, plus $nf$ for the feature locations. Given the high angular uncertainty of the sonar returns, we will only consider as data the $nk$ range values given by the sonar sensor from each robot location to each feature. Provided that no degenerate situations arise, the system is solvable to find robot motion and feature locations iff:

\[ nk \geq r(k - 1) + nf \]  

i.e., iff:

\[ n \geq r \frac{(k - 1)}{(k - f)} \]

or

\[ k \geq \frac{(fn - r)}{(n - r)} \]

In particular, we can also state two simple necessary conditions:

\[ k > f \]
\[ n > r \]

For example, consider the particular case of a robot moving in 2D ($r = 3$), observing the distance to lines or points ($f = 2$). If data association is known, it is possible to compute robot motion and feature location if the robot observes 4 features for at least 5 steps.

Of course, with standard sonar sensors, the difficult part is to solve the data association problem: finding groups of returns coming from the same feature and determining the feature type. When there is enough sensor information (the $> condition in eq. (1) holds), the excess data can be used in a hypothesis-verification scheme to obtain data association, and solve the structure and motion problem.

2.2 The CML problem

In this paper, we will concentrate on solving the CML problem: given a set of robot motions, measured by odometry with some uncertainty, and a set of sonar measurements, solve the data association problem and obtain the structure of the environment together with a better estimation of the robot trajectory.
If the robot motion and data association were perfectly known, from eq. (1),
the condition to solve the structure problem would be:

\[ k \geq f \]  

(2)

However, in order to robustly solve the data association problem, we need
to obtain sensor data from a number of robot positions greater than the limit
defined by eq. (2). When a feature is successfully matched for more than \( f \)
robot locations, the excess data will be used to reduce the uncertainty in robot
and feature location.

When the amount of sensor data successfully associated is large enough to
also satisfy eq. (1), the robot motion could be completely determined from sonar
data. In this case robot uncertainty will approach sensor precision. Otherwise,
the robot uncertainty will only be marginally better than the bare odometric
uncertainty.

3 Robust sonar grouping

The above analysis suggests that by moving the robot several steps and using
robot odometry, it should be possible to group sonar returns and determine
target type. An example data set obtained with a 24 sonar ring along a simple
motion (first a turn and then a straight motion) is shown in figure 2a. Each
point represents the nominal position of a sonar return, computed along the
central axis of the transducer, using the robot odometry. The dotted lines
represent the ground truth environment map. The walls at both sides of the
robot trajectory clearly show up as lines of sonar returns. However, some other
false lines appear in front of the robot. Usually these false lines arise as a result
of people moving in the robot’s environment, but also from some sonar artifacts
such as sensor cross-talk. The lower left corner appears as an ambiguous arc of
sonar returns. Other corner or edges appear as small lines of returns. Also, many
phantom returns arise from multiple specular reflections. Simple segmentation
techniques using only the position of the nominal sonar returns (Grossmann
and Poli 2001) can successfully find groups of returns coming from the same
target (real or phantom), but they will surely fail to distinguish between real
and false targets and to identify the feature type.

On the contrary, if we use a more sophisticated sensor model, ambiguity can
be significantly reduced. Figure 2b shows the same data set where each return
is depicted with an arc showing the sonar angular uncertainty, i.e. showing
all the possible locations of the piece of object surface actually producing the
sonar return. In this more meaningful representation, walls appear as sets of
sonar arcs tangent to the wall, while point features, such as corners or edges,
appear as sets of arcs intersecting at the feature location. In contrast, false
features produce sets of incoherent sonar arcs and thus can be easily spotted.
The conclusion from this example is compelling and well known: using a careful
sensor model is crucial to adequately interpret sonar returns.
Figure 2: Sonar returns along 40 robot positions, with a reference map superimposed. (a) Corners and walls appear as arcs and lines of sonar returns, respectively. But also false lines can be seen as a result of people walking around the robot, specular reflections and other sonar artifacts. (b) When sonar returns are depicted with arcs showing its angular uncertainty, walls clearly show up as sets of tangent arcs, and corners as set of intersecting arcs.
3.1 Sonar modelling

This work is restricted to man-made indoor environments, and we will use two types of geometric features, extracted from the work of Leonard and Durrant-Whyte (1992): 2D lines to represent walls, and 2D points to represent corners and edges (figure 3). A wall, provided that its surface is smooth, will only produce a sonar return for sensor $S_j$, when the perpendicular to the wall is contained in the sonar emission cone:

$$\theta_{S_j} - \frac{\beta}{2} \leq \theta^P \leq \theta_{S_j} + \frac{\beta}{2}$$

where $\beta$ is the visibility angle for walls. The distance actually given by the sensor $\rho^P$ will correspond to the perpendicular distance from the sensor to the line. Conversely, given the sensor location and the measured distances, all possible lines giving such a return are tangent to the arc depicted in figure 3a.

In a similar way, a point feature like a corner or an edge will produce a sonar return for sensor $S_j$ when it is located inside the sonar emission cone. The distance measured will correspond to the distance between the point and the sensor. All possible points giving the same return are located along the arc depicted in figure 3b, where in this case $\beta$ is the visibility angle for point features. Although the visibility angles for corners or for edges may be different (Leonard and Durrant-Whyte 1992), in this work we do not attempt to distinguish them, and we use the same value of $\beta$ for all point features.

3.2 Sonar data association with Hough Transform

With the sonar model presented above, associating sonar returns to line and point features may be stated as finding groups of sonar arcs all tangent to the same line, and groups of sonar arcs intersecting on the same point, respectively. Given the large amount of spurious data coming from moving people, specular reflections and sonar artifacts, classical robust techniques such as RANSAC (Fischler and Bolles 1981, Hartley and Zisserman 2001) or the Hough transform (Ballard and Brown 1982, Illingworth and Kittler 1988) seem very appropriate.

The Hough transform is a voting scheme where each piece of sensor information accumulates evidence about the presence of certain features compatible with the actual measurement. Voting is performed in a discretized parametric space, known as the Hough space, representing all possible feature locations. The most voted cells in the Hough space should correspond to the features actually present in the environment. By keeping track of the votes, it is very easy to obtain the groups of sensor data coming from each feature detected.

We have found that this technique is particularly well suited to efficiently solve the 2D sonar data association problem, for the following reasons:

- The location of point and line features can be easily described with two parameters, giving a 2D Hough space in which the voting process and the search for maxima can be done quite efficiently.
Figure 3: Model of sonar sensor for (a) line features and (b) point features
The sonar model presented before can be used to restrict the votes generated by each sonar return to be located along the corresponding transformed sonar arc.

Since each sonar return emits a constant number of votes, the whole Hough process is linear with the number of returns processed.

Being a voting scheme, it is intrinsically very robust against the presence of many spurious sonar returns.

One of the key issues of its practical implementation is choosing the parameters defining the Hough space and their quantization (Illingworth and Kittler 1988). In our implementation, the Hough transform is applied to the sonar returns obtained along short trajectories (around 1-2m of total travel), in order to keep the odometry errors small. Lines are represented in a base reference $B$, located in the central robot position, using parameters $\theta^B$ and $\rho^B$ defining the line orientation and its distance to the origin (see figure 3a). In the case of points, we have chosen to use a polar representation relative to the same reference $B$ (see figure 3b), instead of a cartesian representation, because the discretization of this Hough space represents more accurately the precision obtained in point location from range sensors. This also has the advantage that the line and point Hough spaces are very similar. The quantization of the Hough voting table was tuned to approximately match the typical accumulated odometry errors during the trajectory (around 3-5cm in position and 2-5deg in orientation).

With this definition of the Hough space, the basic voting algorithm for lines is shown in figure 4, where $\delta_\theta$ is the angular quantization of the Hough space. The voting algorithm for the case of points can be easily derived from that of lines by changing the geometric definition of $\rho^B$ and $\theta^B$ as illustrated in Figure 3b. Figure 5 (left) shows the resulting voting tables for the example trajectory of figure 2.

Next, the two voting tables are searched for local maxima having a number of votes above a certain threshold. However, in real indoor environments, an
Figure 5: Hough voting tables for lines and points, and groups of sonar returns corresponding to the most voted line and point features
ambiguity usually arises: some groups of sonar returns give maxima in both tables, meaning that they could be interpreted as a line or as a point feature. An typical example can be seen in the figure around \( \rho = 2.1 \) and \( \theta = -150\text{deg} \). This problem can be successfully solved by sorting all maxima by the number of votes received, and using a winner-takes-all strategy. In our example, the point hypothesis wins and gets all votes, discarding the line hypothesis. A different case can be seen at \( \rho = 0.6 \) and \( \theta = 90\text{deg} \), where the line hypotheses wins and takes some votes, but there are still enough sonar returns left to also obtain a point hypothesis. However, there are some singular cases where the ambiguity between point and line cannot be decided: a sonar sensor moving perpendicular to a wall, or in a straight line towards a point would get exactly the same returns. To avoid this problem a further verification is made: to accept a point or line feature we require to have sonar returns taken form sensor positions spread around the point or along the line, respectively.

In figure 5 we can see the point and line features identified, with their corresponding sonar data groups. Despite the large amount of spurious data from moving people, multiple reflections and other sonar artifacts, our method was able to successfully detect two line features corresponding to walls and two point features corresponding to a corner and an edge.

4 Consistent Stochastic Mapping using Local Map Sequencing

The perceptual data grouping technique explained above yields a robust interpretation of the sonar data in terms of point and line features. This allows the solution of the continuous mapping and localization problem using classical stochastic mapping techniques, first introduced by Smith and Cheeseman (1986).

In classical stochastic mapping, the environment information related to a set of elements \( \mathcal{F} = \{F, F_0, F_1, \ldots, F_n\} \) is represented by a map \( \mathcal{M}_\mathcal{F} = (\hat{x}^F, P^F) \), where:

\[
\hat{x}^F = \begin{bmatrix} \hat{x}^F_{F_0} \\ \vdots \\ \hat{x}^F_{F_n} \end{bmatrix} ; P^F = \begin{bmatrix} P_{F_0,F_0} & \cdots & P_{F_0,F_n} \\ \vdots & \ddots & \vdots \\ P_{F_n,F_0} & \cdots & P_{F_n,F_n} \end{bmatrix} \tag{3}
\]

The state vector \( \hat{x}^F_{F_0} \) contains the estimated location of the vehicle (usually represented by reference \( F_0 \)) and of the environment features \( F_1, \ldots, F_n \), all with respect to a base reference \( F \). In the case of the vehicle, its location vector \( \hat{x}^F_{F_0} \) describes the transformation from \( F \) to \( F_0 \). In the case of an environment feature \( f \), the parameters that compose its location vector \( \hat{x}^F_f \) depend on the feature type (see appendix A). Matrix \( P^F_{F_0,F_0} \) is the estimated error covariance of \( \hat{x}^F_{F_0} \).
Two basic operations used in stochastic mapping are transformation inversion and composition, which were represented by Smith et al. (1988) using operators $\ominus$ and $\oplus$:

$$\hat{x}_A^B = \ominus \hat{x}_B^A$$
$$\hat{x}_C^A = \hat{x}_B^A \oplus \hat{x}_C^B$$

In this work, we generalize the $\oplus$ operator to also represent the composition of transformations with feature location vectors, which results in the change of base reference of the feature. The Jacobians of these operations are defined as:

$$J_{\ominus} \{ \hat{x}_B^A \} = \frac{\partial (\ominus \hat{x}_B^A)}{\partial \hat{x}_B^A}$$
$$J_{\oplus} \{ \hat{x}_B^A, \hat{x}_C^B \} = \frac{\partial (\hat{x}_B^A \oplus \hat{x}_C^B)}{\partial \hat{x}_B^A}$$
$$J_{\ominus} \{ \hat{x}_B^A, \hat{x}_C^B \} = \frac{\partial (\hat{x}_B^A \oplus \hat{x}_C^B)}{\partial \hat{x}_C^B}$$

The equations to compute these operations and their Jacobians for references, points and lines in 2D are given in Appendix A.

The classical technique for updating the state vector of a stochastic map and its covariance uses EKF, given that the full covariance matrix must be maintained to assure consistency, updating a full stochastic map of $n$ features is well known to be $O(n^2)$ (Castellanos et al. 1999, Guivant and Nebot 2001). Many recent efforts are concentrated in reducing this computational complexity. Decoupled Stochastic Mapping (Leonard and Feder 2000, Jensfelt 2001), the Local Mapping Algorithm (Chong and Kleeman 1999b), Suboptimal SLAM (Guivant and Nebot 2001), and the Sparse Weight Filter (Julier 2001), reduce the computational complexity of map updating to $O(1)$ the first two and $O(n)$ the others, by obtaining a suboptimal, or pessimistic solution. In contrast, Postponement (Davison 1998, Knight, Davison and Reid 2001) and the Compressed Filter (Guivant and Nebot 2001) delay the global map update, obtaining an optimal solution and strongly reducing the computational cost although it still remains $O(n^2)$.

In this work we propose a new solution, denominated Local Map Sequencing, that obtains a full, consistent and optimal global map from a sequence of independent local maps. First, we use classical stochastic mapping techniques to build local maps along limited portions of the robot trajectory. Then, we join each local map with the global map, find matchings between global and local features, and fuse them to update the global map. The computational complexity of Local Map Sequencing is equivalent to that of the Compressed Filter and Postponement, with the additional advantage of updating local maps where errors remain small, therefore reducing the harmful effects of linearization errors. The mathematical details of Local Map Sequencing are explained next.
4.1 Changing the base reference of a stochastic map

Suppose that we choose to change the base reference of a map $\mathcal{M}_F = (\hat{x}^F, P^F)$ from $F$ to $F_j$, where reference $F_j$ is associated to the vehicle, or to any map feature. Using the composition and inversion operations defined above, the resulting state vector would be:

$$\hat{x}^F = \begin{bmatrix} \hat{x}^F_{F_0} \\ \vdots \\ \hat{x}^F_{F_j} \\ \vdots \\ \hat{x}^F_{F_m} \end{bmatrix} = \begin{bmatrix} \oplus\hat{x}^F_{F_j} \oplus \hat{x}^F_{F_0} \\ \vdots \\ \oplus\hat{x}^F_{F_j} \oplus \hat{x}^F_{F_j} \\ \vdots \\ \oplus\hat{x}^F_{F_j} \oplus \hat{x}^F_{F_m} \end{bmatrix}$$

(4)

To make this operation reversible, we have incorporated the location of the former base reference $F$ with respect to $F_j$ in the new state vector, replacing the location of feature $F_j$. The Jacobian corresponding to this transformation of the state vector would be:

$$J^F = \frac{\partial \hat{x}^F}{\partial \hat{x}^F} = \begin{bmatrix} \mathbf{J}_{00} & \cdots & \mathbf{J}_{0j} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{J}_{jj} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{J}_{mj} & \cdots & \mathbf{J}_{mm} \end{bmatrix}$$

where:

$$\mathbf{J}_{jj} = \mathbf{J}_1 \left\{ \hat{x}^F_{F_j} \right\}$$

$$\mathbf{J}_{ij} = \mathbf{J}_1 \left\{ \oplus\hat{x}^F_{F_j}, \hat{x}^F_{F_i} \right\} \mathbf{J}_2 \left\{ \hat{x}^F_{F_j} \right\} \quad i = 0..m, \: i \neq j$$

$$\mathbf{J}_{ii} = \mathbf{J}_2 \left\{ \oplus\hat{x}^F_{F_j}, \hat{x}^F_{F_i} \right\} \quad i = 0..m, \: i \neq j$$

Thus, the covariance of $\hat{x}^F$ can be obtained as follows:

$$P^F = J^F \cdot P^F \cdot J^F_T$$

(5)

A particular case of this transformation, which produces a stochastic map relative to the vehicle reference ($j = 0$), was proposed by Castellanos and Tardós (1999).

4.2 Local Map Joining

The map joining mechanism can be applied to any two maps, provided that these two conditions hold:

1. The two maps are statistically independent.
2. A reference common to both maps has been identified.

Assume that we have two statistically independent stochastic maps \( \mathcal{M}_F = (\tilde{x}_F^F, P_F^F) \), involving features \( \mathcal{F} = \{ F, F_0, F_1, \ldots, F_n \} \), and \( \mathcal{M}_E = (\tilde{x}_E^E, P_E^E) \), involving features \( \mathcal{E} = \{ E, E_0, E_1, \ldots, E_m \} \). Assume that we have identified a common reference in both maps: \( F_i = E_j \). These two hypotheses allow us to consistently join the information contained in \( \mathcal{M}_F \) and \( \mathcal{M}_E \) in a full stochastic map \( \mathcal{M}_{F+\mathcal{E}} = (\tilde{x}_{F+\mathcal{E}}^F, \mathcal{P}_{F+\mathcal{E}}^F) \), relative to reference \( F \). This can be done by first changing the base reference of map \( \mathcal{M}_E \) from \( E \) to \( E_j \), using eqs. (4) and (5), and then obtaining the joint state vector \( \tilde{x}_{F+\mathcal{E}}^F \) as follows:

\[
\tilde{x}_{F+\mathcal{E}}^F = \begin{bmatrix}
\tilde{x}_F^F \\
\tilde{x}_E^F \\
\vdots \\
\tilde{x}_F^{E_1} \oplus \tilde{x}_E^{E_j} \\
\cdots \\
\tilde{x}_F^{E_1} \oplus \tilde{x}_E^{E_m}
\end{bmatrix}
\]

(6)

In order to determine the covariance \( \mathcal{P}_{F+\mathcal{E}}^F \) of the state vector \( \tilde{x}_{F+\mathcal{E}}^F \), we need the Jacobians:

\[
\mathcal{J}_{F+\mathcal{E}}^F = \frac{\partial \tilde{x}_{F+\mathcal{E}}^F}{\partial \tilde{x}_F^F} = \begin{bmatrix}
\mathcal{J}_1 \\
\mathcal{J}_2
\end{bmatrix}
\]

where:

\[
\mathcal{J}_1 = \begin{bmatrix}
0 & \cdots & \mathcal{J}_1 \{ \tilde{x}_F^F, \tilde{x}_E^{E_j} \} & \cdots & 0 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0 & \cdots & \mathcal{J}_1 \{ \tilde{x}_F^F, \tilde{x}_E^{E_m} \} & \cdots & 0
\end{bmatrix}
\]

\[
\mathcal{J}_2 = \begin{bmatrix}
\mathcal{J}_2 \{ \tilde{x}_F^F, \tilde{x}_E^{E_j} \} & \cdots & 0 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0 & \cdots & \mathcal{J}_2 \{ \tilde{x}_F^F, \tilde{x}_E^{E_m} \}
\end{bmatrix}
\]

The covariance \( \mathcal{P}_{F+\mathcal{E}}^F \) of the joint map is given by:

\[
\mathcal{P}_{F+\mathcal{E}}^F = \begin{bmatrix}
\mathcal{P}_F^F & \mathcal{P}_F^F \mathcal{J}_1^T \\
\mathcal{J}_1 \mathcal{P}_F^F & \mathcal{J}_1 \mathcal{P}_F^F \mathcal{J}_1^T + \mathcal{J}_2 \mathcal{P}_E^E \mathcal{J}_2^T
\end{bmatrix}
\]

(7)
Obtaining vector $\mathbf{x}'_{\mathcal{F},\mathcal{E}}$ with eq. (6) is an $O(m)$ operation. Given that the number of non-zero elements in $\mathbf{J}_1$ and $\mathbf{J}_2$ is $O(m)$, obtaining matrix $\mathbf{P}'_{\mathcal{F},\mathcal{E}}$ with eq. (7) is an $O(nm + m^2)$ operation. Thus when $n \gg m$, map joining is linear with $n$.

Map joining is a general procedure to consistently convey the information of several independent stochastic maps, a very useful tool in many situations. Some of its applications are:

- **Robot relocation.** In this problem, also known as the First Location problem, or the Kidnapped Robot problem, a global stochastic map of the environment where the vehicle is navigating is available, but the location of the vehicle is completely unknown. In this case, the vehicle can build a local map of the environment, which will be statistically independent from the global map. If we identify a feature in the local map whose location is available in the global map, we can join the two maps. The result will be a global map which includes the vehicle location.

- **Multirobot map building.** We can deploy a team of vehicles to independently build maps of different areas of an environment. Whenever we can determine that two vehicles have included the same environment feature in their maps, we can join their maps into a full map that also includes the location of one vehicle with respect to the other.

- **Local map sequencing.** In this paper we concentrate on showing that map joining can be used to obtain a full consistent stochastic map of any size from a sequence of independent local stochastic maps of constant size.

### 4.3 Matching and Fusion after Map Joining

The map resulting from map joining is statistically consistent, but may not be optimal: it may contain features that, coming from different local maps, correspond to the same environment feature. To eliminate such duplications and obtain an optimal map we need a data association algorithm to determine correspondences, and a feature fusion mechanism to update the global map. Determining correspondences between features in two successive local maps is a fairly simple problem, because the relative location uncertainty is small. Classical techniques, such as the Nearest Neighbor (Castellanos et al. 1999, Feder et al. 1999), will successfully solve this problem. However, to find matchings in loop closing situations, a more robust method is required. We use the Joint Compatibility Branch and Bound algorithm (Neira and Tardós 2001), which provides the largest set of correspondences that are guaranteed to be jointly consistent, taking all feature correlations into account. Experiments will show the superiority of this algorithm in loop closing situations.

Next, we explain the details of feature matching and fusion for global map updating. For simplicity, let $\mathbf{x}'_{\mathcal{F},\mathcal{E}}$ and $\mathbf{P}'_{\mathcal{F},\mathcal{E}}$ represent the state and covariance of the joined map. Let us consider also that references $F_0$ and $E_0$ represent the vehicle location in each corresponding map. Let $\mathcal{H} = \{j_1, \ldots, j_m\}$
be a hypothesis that pairs each feature \( E_i \) coming from the local map with a feature \( F_j \) coming from the global map. When \( j_i = 0 \), feature \( E_i \) is considered new. Each feature \( E_i \) and its corresponding feature \( F_j \) will be related by an \textit{implicit function} (Ayache and Faugeras 1988) of the form:

\[
f_{ij_i}(x) = 0
\]  

Function \( f_{ij_i} \) imposes a constraint on the location of features \( E_i \) and \( F_j \). Note that in this formulation the location of \( E_i \) and \( F_j \) are both already part of the state. Thus, the state estimation can be updated using the update equations of the EKF without measurement noise. Since \( f_{ij_i} \) is usually non-linear, linearization around the current estimation is necessary:

\[
f_{ij_i}(x) \simeq h_{ij_i} + H_{ij_i}(x - \hat{x})
\]

where:

\[
h_{ij_i} = f_{ij_i}(\hat{x})
\]

\[
H_{ij_i} = \left. \frac{\partial f_{ij_i}}{\partial x} \right|_{(\hat{x})} = \begin{bmatrix} 0 & \cdots & H_{F_{j_i}} & \cdots & H_{E_i} & \cdots & 0 \end{bmatrix}
\]

\[
H_{F_{j_i}} = \left. \frac{\partial f_{ij_i}}{\partial x_{F_{j_i}}} \right|_{(x_{F_{j_i}})}
\]

\[
H_{E_i} = \left. \frac{\partial f_{ij_i}}{\partial x_{E_i}} \right|_{(x_{E_i})}
\]

Vector \( h_{ij_i} \) represents the innovation of the pairing between \( E_i \) and \( F_j \). The joint implicit function of hypothesis \( \mathcal{H} \) is \( f_{\mathcal{H}}(x) = 0 \), where:

\[
f_{\mathcal{H}}(x) = \begin{bmatrix} f_{1,\mathcal{H}}(x) \\ \vdots \\ f_{m,\mathcal{H}}(x) \end{bmatrix} \simeq h_{\mathcal{H}} + H_{\mathcal{H}}(x - \hat{x})
\]

\[
h_{\mathcal{H}} = f_{\mathcal{H}}(\hat{x}) = \begin{bmatrix} h_{1,\mathcal{H}} \\ \vdots \\ h_{m,\mathcal{H}} \end{bmatrix}
\]

\[
H_{\mathcal{H}} = \left. \frac{\partial f_{\mathcal{H}}}{\partial x} \right|_{(\hat{x})} = \begin{bmatrix} H_{1,\mathcal{H}} \\ \vdots \\ H_{m,\mathcal{H}} \end{bmatrix}
\]

The validity of \( \mathcal{H} \) can be determined using an innovation test on the joint innovation \( h_{\mathcal{H}} \) as follows:
\[
D^2_{\ell} = h^T_{\ell} \left( H_{\ell} P H^T_{\ell} \right)^{-1} h_{\ell} < \chi^2_{d, \alpha}
\]

The value of $\alpha$ is the desired confidence level, and $d = \dim (f_{\ell})$. This innovation test is used inside the Joint Compatibility Branch and Bound to search for the hypothesis with the largest set of compatible matchings. Once this hypothesis has been found, is can be used to obtain estimate $\tilde{x}_k$ of the state vector and its covariance $P_k$, by applying the EKF as follows:

\[
\tilde{x}_k = \hat{x}_{k-1} - K_k h_{\ell}
\]
\[
P_k = (I - K_k H_{\ell}) P_{k-1}
\]

where:

\[
K_k = P_{k-1} H^T_{\ell} \left( H_{\ell} P_{k-1} H^T_{\ell} \right)^{-1}
\]

Notice that the minus sign in the state update equation appears because we use an implicit measurement equation (8). Once the matching constraints have been applied, the corresponding matching features become fully correlated, with the same estimation and covariance. Thus, one of them can be eliminated.

4.4 Local Map Sequencing

Map joining is used in Local Map Sequencing to obtain a full consistent stochastic map of any size by joining a sequence of local maps of constant size. In order
to satisfy the conditions under which map joining can be used, we can proceed as follows: assume that at step $i$, using the current robot location as base reference, say $B_i$, we perform a limited motion, acquiring sensorial information, and build a standard stochastic map of limited size $M^{B_i}_{F_i}$, which includes the final robot location $R_i$ and the set of perceived features $F_i$; in step $i+1$ we start a new map $M^{B_{i+1}}_{F_{i+1}}$ using the current vehicle location as base reference (fig. 6). If no sensorial information has been shared to build both maps, vectors $\tilde{x}^{B_i}_{F_i}$ and $\tilde{x}^{B_{i+1}}_{F_{i+1}}$ are guaranteed to be statistically independent. Additionally, we know that the last vehicle position in map $i$, is the base reference of map $i + 1$: i.e. $B_{i+1} = R_i$. This correspondence gives us the second condition that allows us to join both maps into map $M^{B_i}_{F_{i+1}}$. After $n$ steps, we have a full stochastic map $M^{B_i}_{F_{i:n}}$, with the initial vehicle position as base reference.

In order to determine the computational cost of Local Map Sequencing, assume that a vehicle navigates in an environment of $n$ features. Assume that each feature is perceived by the vehicle from $k$ different locations. In full stochastic mapping, the cost related to this feature is $O(n)$ when it is included in the map, plus $O(n^2)$ for map updating each time it is re-observed, giving a total cost of $O((k - 1)n^2)$. In contrast, using Local Map Sequencing, assume that each feature appears in $q$ different local maps. The cost of including and updating the feature in each local map does not increase with $n$, and therefore it is negligible. Its inclusion in the full stochastic map will cost $O(n)$ when the first local map where the feature appears is joined, plus $O(n^2)$ for each full map update where the feature reappears, giving a total cost of $O((q - 1)n^2)$. Therefore, Local Map Sequencing cuts processing time by a factor of $\frac{k-1}{q-1}$. In our experiments, each feature has been observed from a mean of $k = 77$ locations, in a mean of $q = 2.3$ local maps, giving an asymptotic speedup factor of around 58.5.

5 Experimental results

Experiments were carried out using a B21 mobile robot equipped with a SICK laser scanner and a ring of 24 Polaroid sensors (the enclosure sensors were used). The robot carried out a guided trajectory in the 'Compton' Gallery, a 12m×12m exhibition gallery at MIT. Several people were visiting the gallery during the experiment. The total trajectory of the robot, which included several loops, was around 101m, lasting around 18 minutes, with an average speed of 9.2cm/s. During the robot motion, the laser sensor and the sonar ring were acquiring range scans at average frequencies of 2.28Hz and 3.72Hz, respectively. Figure 7a shows the raw sonar returns obtained during the first 10 minutes of the trajectory. Two important facts should be noticed in the figure:

- Although the environment structure is still perceptible, there are huge amounts of spurious data coming from moving people, specular reflections and other sonar artifacts.

- Robot odometry suffered a severe drift, probably due to the carpeted floor
Figure 7: (a) Sonar returns obtained during the first 10 minutes of the experiment. (b) Lines and points detected using the Hough transform.
of the Compton Gallery. In the upper right corner, after a travel of some 35m, the accumulated error is about 3.9m and 21deg; when the robot is closing the loop, in the lower left corner, the error is about 2.4m and 22deg.

The grouping scheme described in section 3 was applied to the sonar returns obtained along short sequences of 40 robot steps, with a separation of 5cm between them. The lines and points detected are shown in figure 7b. The small dots along the robot trajectory represent the locations where the Hough transform was applied. Most lines detected correspond to the walls, while point features correspond to corners, edges and picture frames and other objects placed on the walls. The majority of spurious sonar returns were successfully rejected by our grouping scheme.

For every sequence of 40 steps, a local stochastic map with points and lines was built, using conventional techniques (Castellanos et al. 1999). The data associations given by the Hough transform were also stochastically verified by testing the joint compatibility between the returns and their corresponding map features (Neira and Tardós 2001). Each local map was built relative to the robot location at the beginning of the sequence.

Using the techniques described in section 4, a global stochastic map was built by joining every new local map as it was available. Then, the largest set of jointly compatible pairings between the new features coming from the local map and those of the global map was obtained (Neira and Tardós 2001). This set of pairings was used to update the global map. An example of these map joining and updating processes is shown in figure 8. In a first step (figure 8a), a local map with four features P1, P2, S3 and S4, is joined with a second local map containing two features S5 and S6. The first robot depicted represents the estimated robot location at the end of the first local map, and the second corresponds to the estimated robot location after the joining process. Once the correspondences between segments S3 and S5, and segments S4 and S6 have been established, segment fusion is carried out using the EKF equations in section 4.3, and the estimations of the robot and feature locations are updated (figure 8b).

The sonar grouping, local map building, map joining, matching and updating processes were repeated along the robot trajectory, obtaining a growing global map. The robustness of our approach is exemplified by the loop closing operation shown in figure 9. Due to the big odometry errors accumulated, simple data association algorithms, such as the Nearest Neighbor, would incorrectly match the signaled point with a point feature previously observed in the pillar. Accepting an incorrect matching will cause the EKF to diverge, obtaining an inconsistent map. On the other hand, our joint compatibility algorithm takes into account the relative location between the point and the segment (Neira and Tardós 2001) and has no problem in finding the right associations. The result is a consistent and more precise global map.

The final global map obtained at the end of the experiment is shown in figure 10a. For comparison purposes, the laser scans obtained every 5cm along
Figure 8: (a) A local map with four features, P1, P2, S3 and S4, is joined with another local map with two features, S5 and S6. (b) Resulting map after fusing S3 with S5, and S4 with S6. Uncertainty is depicted by the 2σ bounds for point and segment tip locations.
Figure 9: Global stochastic maps in a loop closing situation. (a) Before loop closing, a local map with features signaled with an arrow has been joined to the global map. (b) The two features have been correctly matched with the corner and the lower wall, and the global map has been updated.
the same trajectory were processed to extract straight segments, and the same map joining approach was used to build a segment map of the environment (figure 10b). Comparing both figures, it is clear that the laser sensor is able to obtain a more detailed map. The laser map suffers from a small magnification error, probably due to a systematic error in range measurements. The compensation of this error with a suitable calibration technique would also produce a map more precise than the sonar map. Nevertheless, the sonar map obtained is consistent with ground truth and is complete and precise enough to be reliably used for navigation purposes. To further analyze the consistency of our approach, we obtained a ground truth solution for the robot trajectory by matching the segments obtained from laser and the true environment map, measured by hand. Figure 11 shows the errors in robot location, relative to this ground truth, during the sonar map building process. We can clearly see that most of the time, errors remain inside their 2σ uncertainty bounds.

Regarding computational efficiency, all techniques presented run faster than real-time for this environment. In our actual implementation, typical computing times on a Pentium III at 600MHz, for each sequence of 40 steps (about 2 m travel) are: 300ms for sonar grouping, 1.5s for local map building, and up to 1s for fusing the local map with the global map (joining, matching and updating). The first two processes use local information and their computing times are independent of environment size. The computing time for map fusion grows with the number of features in the global map. Although its asymptotic complexity is quadratic, the moderate size of the final map in our experiment (63 features) makes the increase to be almost linear, dominated by the matching operation. For bigger environments, the quadratic cost of the global map update would become dominant.

6 Conclusion

This paper has presented several new techniques for mobile robot navigation and mapping with sonar. Our results include a new method for detection of points and line segments from sonar data and new techniques for joining and combining two stochastic maps. Results have been presented for the implementation of these methods using data from a standard ring of Polarcid sonars mounted on a B21 mobile robot. The results are compared to a map produced from laser scanner data for exactly the same environment. To our knowledge, this constitutes the first published side-by-side comparison of laser and sonar maps produced via CML.

Many early implementations of navigation and mapping algorithms that used a first-order representation of uncertainty suffered the criticism of “brittleness” (Lozano-Pérez 1989). When incorrect decisions are made concerning the origins of measurements, dramatically erroneously results can be produced. These issues motivated the development of methods for sensor data interpretation that can make delayed decisions, such as multiple hypothesis tracking (Cox and Leonard 1994), to achieve better classification of measurements. The meth-
Figure 10: Global maps obtained using (a) sonar and (b) laser. Small dots along the robot trajectory represent the locations where map joining and updating was performed.
Figure 11: Errors in robot location during the sonar map building process. Dotted lines represent $2\sigma$ uncertainty bounds.
ods presented in this paper can effectively make delayed decisions about the assignment of individual measurements to achieve robustness despite the difficulties inherent in sonar data. The voting scheme used to aggregate data from multiple vantage points, acts analogously to methods in computer vision such as RANSAC that use the principle of consensus to resolve data association ambiguity. We believe that in the future it should be possible to generalize the concept behind the voting scheme used in this paper to provide a more generic capability for computing structure-from-motion from acoustic sensors in more complex environments (i.e., three-dimensional and underwater).

We anticipate that the techniques presented here for joining and combining multiple submaps in CML will be useful in the development of new computationally efficient methods for large-scale CML. By performing the operations of data association and feature initialization in local submaps, sensitivity to linearization errors in the state estimation process is reduced. Indeed, it is possible to pose the problem such that all operations for creation of local submaps could be performed without Kalman filtering, for example using nonlinear least-squares minimization techniques that have effectively been employed for the structure from motion problem in computer vision (Hartley and Zisserman 2001).

In the example in this paper, all maps were combined relative to a single, globally-referenced coordinate frame. In situations with sufficiently large angular errors, it will be impossible to consistently reference all submaps to a common reference frame using a first-order representation of uncertainty. For this situation, we envision that effective solutions can be developed using a representation consisting of a network of different coordinate frames, each related through a sequence of approximate transformations. The techniques of joining and combining submaps are expected to be invaluable as a means of building and maintaining the components of the map.

Acknowledgements

This research has been funded in part by the National Science Foundation under Career Award BES-9733040, the National Oceanic and Atmospheric Administration via the MIT Sea Grant College Program under grant NA86RG0074 (project RCM-3), the Ministerio de Educación y Cultura of Spain, grant PR2000-0104, and the Spain-US Commission for Cultural, Educational and Scientific Exchange (Fulbright), grant 20079.

Appendix A: Transformations, points and lines in 2D

Transformations

In 2D, the location of a reference $B$ relative to a reference $A$ (or transformation from $A$ to $B$) can be expressed using a vector with three d.o.f.:
\[
x^A_B = \begin{bmatrix} x_1 \\
y_1 \\
\phi_1 \end{bmatrix}
\]

The location of A relative to B is computed using the inversion operation:

\[
x^B_A = \ominus x^A_B = \begin{bmatrix} -x_1 \cos \phi_1 - y_1 \sin \phi_1 \\
x_1 \sin \phi_1 - y_1 \cos \phi_1 \\
-\phi_1 \end{bmatrix}
\]

The Jacobian of transformation inversion is:

\[
J_\ominus \{x^A_B\} = \begin{bmatrix} -\cos \phi_1 & -\sin \phi_1 & -x_1 \sin \phi_1 - y_1 \cos \phi_1 \\
\sin \phi_1 & -\cos \phi_1 & x_1 \cos \phi_1 + y_1 \sin \phi_1 \\
0 & 0 & -1 \end{bmatrix}
\]

Let \(x^B_C = [x_2, y_2, \phi_2]^T\) be a second transformation. The location of reference C relative to A is obtained by the composition of transformations \(x^A_C = x^A_B \oplus x^B_C\):

\[
x^C_A = x^A_B \oplus x^B_C = \begin{bmatrix} x_1 + x_2 \cos \phi_1 - y_2 \sin \phi_1 \\
y_1 + x_2 \sin \phi_1 + y_2 \cos \phi_1 \\
\phi_1 + \phi_2 \end{bmatrix}
\]

The Jacobians of transformation composition are:

\[
J_1 \oplus \{x^A_B, x^B_C\} = \begin{bmatrix} 1 & 0 & -x_2 \sin \phi_1 - y_2 \cos \phi_1 \\
0 & 1 & x_2 \cos \phi_1 - y_2 \sin \phi_1 \\
0 & 0 & 1 \end{bmatrix}
\]

\[
J_2 \oplus \{x^A_B, x^B_C\} = \begin{bmatrix} \cos \phi_1 & -\sin \phi_1 & 0 \\
\sin \phi_1 & \cos \phi_1 & 0 \\
0 & 0 & 1 \end{bmatrix}
\]

**Point features**

If we consider a point feature \(P\), we can represent its location with respect to reference \(B\) in cartesian coordinates:

\[
x^B_P = \begin{bmatrix} x_2 \\
y_2 \end{bmatrix}
\]

The composition operation to obtain the location of the point with respect to reference \(A\) is as follows:

\[
x^A_P = x^A_B \oplus x^B_P = \begin{bmatrix} x_1 + x_2 \cos \phi_1 - y_2 \sin \phi_1 \\
y_1 + x_2 \sin \phi_1 + y_2 \cos \phi_1 \end{bmatrix}
\]

The Jacobians for transforming points are:
\[ J_1 \oplus \{ x^A_B, x^B_B \} = \begin{bmatrix} 1 & 0 & -x_2 \sin \phi_1 - y_2 \cos \phi_1 \\ 0 & 1 & x_2 \cos \phi_1 - y_2 \sin \phi_1 \end{bmatrix} \]
\[ J_2 \oplus \{ x^A_B, x^B_B \} = \begin{bmatrix} \cos \phi_1 & -\sin \phi_1 \\ \sin \phi_1 & \cos \phi_1 \end{bmatrix} \]

Line features

The location of a line feature \( L \) relative to reference \( B \) can be represented using the perpendicular distance from the origin of reference \( B \) to the line, and the line orientation:

\[ x_L^B = \begin{bmatrix} \rho_2 \\ \theta_2 \end{bmatrix} \]

The composition operation to obtain the location of the line with respect to reference \( A \) is as follows:

\[ x_L^A = x_A^B \oplus x_L^B = \begin{bmatrix} x_1 \cos (\phi_1 + \theta_2) + y_1 \sin (\phi_1 + \theta_2) + \rho_2 \\ \phi_1 + \theta_2 \end{bmatrix} \]

The Jacobians for transforming lines are:

\[ J_1 \oplus \{ x^A_B, x_L^B \} = \begin{bmatrix} \cos (\phi_1 + \theta_2) & \sin (\phi_1 + \theta_2) & 0 & 0 \\ 0 & 0 & -x_1 \sin (\phi_1 + \theta_2) + y_1 \cos (\phi_1 + \theta_2) & 0 \end{bmatrix} \]
\[ J_2 \oplus \{ x^A_B, x_L^B \} = \begin{bmatrix} 1 & 0 & -x_1 \sin (\phi_1 + \theta_2) + y_1 \cos (\phi_1 + \theta_2) \\ 0 & 1 \end{bmatrix} \]

References


Lu, F. and Milios, E.: 1997a, Globally consistent range scan alignment for environment mapping.


