

Map Merging for Distributed Robot Navigation

Kurt Konolige, Dieter Fox[§], Benson Limketkai, Jonathan Ko[§], Benjamin Stewart[§]

Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
konolige@ai.sri.com

[§]Computer Science and Engineering
University of Washington
Seattle, Washington 98195
fox@cs.washington.edu

Abstract

A set of robots mapping an area can potentially combine their information to produce a distributed map more efficiently than a single robot alone. We describe a general framework for distributed map building in the presence of uncertain communication. Within this framework, we then present a technical solution to the key decision problem of determining relative location within partial maps.

1 Introduction

Coordinated exploration of an unknown environment is one of the most fundamental problems in multirobot operation. As autonomous exploration and map-making becomes increasingly robust on single robots, the next challenge is to extend these techniques to large teams of robots; our current project aims to field 100 robots. A general solution must consider some difficult issues, including limited communication between robots, no assumptions about relative start locations of the robots, and dynamic assignment of processing tasks. In this paper, we first describe a novel approach that addresses these issues, and then identify and concentrate on a key technical challenge, *map merging*.

1.1 A Distributed Approach

We propose a novel, distributed approach to the coordinated exploration problem. Each robot is capable of mapping the environment on its own, and is also capable of communicating intermittently with other robots. When robots are in communication, they can share information and coordinate their exploration activities.

Our distributed approach to mapping and exploration is enabled by pair-wise relations between robots. Each pair of robots can have four different types of interactions: none, hypothesis generation, hypothesis verification, and coordination. At each point in time, the state of the multi-robot system can be summarized by a graph structure where the nodes are individual robots and edges represent the current interaction between two robots (see Figure 1).

We will now briefly discuss the different types of interactions.

1. **No interaction**: The robots are not within communication range (no arc between nodes in Figure 1).
2. **Hypothesis generation** (dotted edges): The robots can communicate but don't know their relative locations. In this stage, one of the two robots receives sensor data from the other robot and estimates their relative location using its own map. Which of the two robots adopts the estimation role depends on available computational resources.
3. **Hypothesis verification** (dashed edges): Robots can communicate and verify a location hypothesis determined in the hypothesis generation phase. In the case of uncertainty, the robots can try to meet. If the robots don't meet at the expected location, the hypothesis is rejected and they continue with the hypothesis generation phase. Otherwise, the robots can establish their relative positions, combine their maps, and coordinate their exploration efforts.
4. **Coordinated exploration** (solid edges): Once the robots determined their relative locations, they can share their maps and perform coordinated exploration. A nice feature of this interaction type is transitivity, *i.e.* if robot i and j can share their maps, and robot j and k can share their maps, then all three robots can build a combined map. Hence, robots in this interaction mode form *exploration clusters* in which they can coordinate their actions (indicated by the gray areas in Figure 1). Each cluster determines one robot responsible for data combination and robot coordination (dark nodes). All information is frequently spread throughout the cluster, and direct communication between all robot pairs within the cluster is not required; nor is a central information repository or mission control.

1.2 Technical Approach

We are at the beginning of a multi-year effort in building a robot network that conforms to this design. To implement the individual parts of our architecture, we rely on existing, well established techniques whenever possible. The key components are:

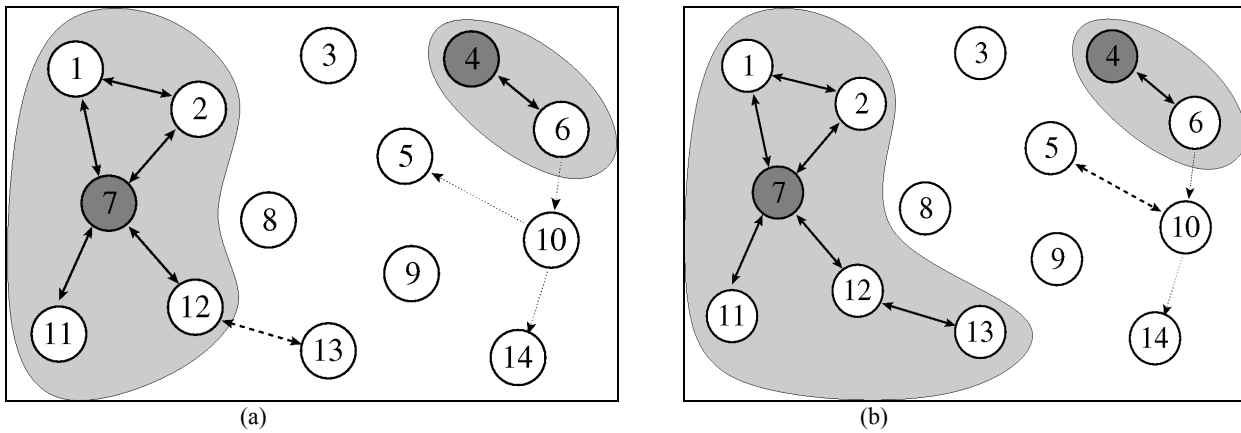


Figure 1 Dynamic communication / interaction graph at two points in time. Note that this graph illustrates different interactions between robots, not spatial relations. Robots are shown as circles. Solid edges indicate coordinated exploration of robots; dashed lines indicate that two robots currently navigate to a meeting point so as to verify a hypothesis for their relative positions; and dotted lines show communication between robots without valid location hypotheses. (a) Robots 1, 2, 7, 11, and 12 already established their relative locations and coordinate exploration. For this exploration cluster, robot 7 was chosen to perform data combination and exploration coordination. Robots 12 and 13 do not yet know their relative locations. They are currently moving to a meeting point so as to verify a location hypothesis. Robot 10 can communicate with robots 5, 6, and 14, but no good location hypothesis has been generated so far. (b) Robots 12 and 13 moved to the meeting point and detected each other. As a result, robot 13 is integrated into the exploration cluster. Robot 10 determined a hypothesis for robot 10's relative location. Robot 10 accepted the meeting point and they both move to this location.

Individual mapping and exploration Individual robots can map and explore an area, using laser rangefinder and visual information. [Yamauchi 1998, Thrun et al. 2000, Gutmann and Konolige 2000].

Coordinated mapping and exploration Individual robots can determine areas that have not been explored, and form plans to explore them efficiently [Burgard et al. 2000, Simmons et al. 2000].

Localization Robots are able to localize themselves within maps, using efficient probabilistic techniques, especially Markov Localization [Fox et al. 2000, Fox et al. 2001].

However, these techniques do not by themselves solve the problem of merging maps made by independently working robots, which is the key enabling technology for the distributed exploration problem. Consider the situation in Figure 2. Two robots are exploring the large environment, and have made partial maps (upper left). Can these two maps be merged? They share a general geometric configuration, and many features. But, they came from different parts of the environment, and have no overlap. The main focus of this paper is to experimentally determine the reliability of matching partial maps on the basis of geometric information they contain.

1.3 Previous Work

We have mentioned work on Markov Localization. An offshoot of this research is a solution to the *globalization* problem, that is, determining location when the initial position within a map is unknown [Fox et al. 2001]. A simple approach to map merging is to use globalization to find the position of one robot within the other's map. However, because Markov Localization makes the strong assumption that the robot is *somewhere* within the map, globalization can easily lead to false positive. For example, the two wings of the map in Figure 2 look somewhat alike. We have been able to fully (and falsely) globalize a robot restricted to the horizontal wing, using sensor data from the vertical wing.

Some research has directly addressed the question of merging maps, when *a priori* information restricts the merge location. [Gutmann and Konolige 1999] describe a technique for "closing the loop" in cyclic environments,

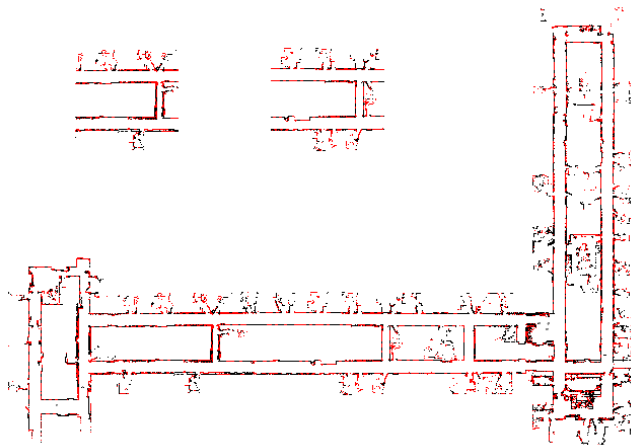


Figure 2 An office map, and two partial maps extracted by robot exploration. Are the partial maps the same? Do they overlap?

when a robot returns to an area of previous exploration via a loop. [Thrun et al. 2000] implement a similar facility using particle filter methods. Both these systems sidestep the basic question of when the merging decision can be reliably made. A different approach is taken in [Thrun et al. 1998a, 1998b]. Instead of an irrevocable decision about merging partial maps, the information about possible merges is assimilated in a probabilistic way as the maps are being built, so that many possible merges can be simultaneously explored. While elegant theoretically, the implementation via EM techniques tends to be impractical, and does not produce good maps.

The SLAM community has generally ignored questions of data association, but some recent work in [Newman et al.] uses features and geometric constraints to merge partial maps. They concentrate on the search problem for finding matches, while here we examine the decision problem for reliable matching. We also note that these segment-based methods, unlike scan matching techniques, have not been tried in large, difficult indoor environments.

2 Map Merging as a Decision Problem

When two robots i and j exchange map information, they must determine whether they are in each other's partial maps. We have a decision problem that can be phrased by robot i as:

(1) *Am I at location \mathbf{x} in robot j 's map?*

Once the decision is made to combine maps, it is difficult to undo, and so it is important that the decision (1) be reliable. On the other hand, if decisions are *not* made when i is indeed within j 's map, then the map construction will be inefficient. So we need a decision method that is both highly specific and highly sensitive.

This question is similar to a decision problem for localization within a given map:

(2) *Am I at location \mathbf{x} in the localization map?*

In Markov localization [Fox et al. 2001, Thrun et al. 2001], standard Bayesian techniques are used to find the most likely place for the robot, given that the robot must be somewhere in the map. But Markov localization does not address the decision problem (2), which can be important for planning. Since (2) is a particular form of (1), the results from this paper are also applicable to decision problems in localization, which have not received attention in the literature.

2.1 Decision Problem Formulation

We formulate the decision problem (1) in terms of likelihood over positions of the robot, both in the partial map and in any other possible map that the robot could be in. Formally:

$$(3) \quad L(\mathbf{x}) > \alpha \quad \text{and} \quad \text{MIN}_{\mathbf{x}' \in E} \left[\frac{L(\mathbf{x})}{L(\mathbf{x}')} \right] > \beta$$

where \mathbf{x} is the map position of the decision, $L(\mathbf{x})$ is the likelihood of being at \mathbf{x} , and E is the set of possible maps that the robot could be in. There are two thresholds for the decision: the absolute likelihood has to be large (α is close to 1.0), and the likelihood relative to all other possible places should be high (β larger than 1.0).

To understand this decision equation, consider a complex of several buildings, in which the offices are given unique tags. A robot seeing the tag *EK288* would have a high likelihood of being at the office EK288, and a very low likelihood of being anywhere else, because the tags are unique; the minimum ratio is very high. Conversely, a robot in front of an office, but not seeing the tag, would have a high likelihood of being in front of *any* office, and while the absolute likelihood $L(\mathbf{x})$ would be high, the minimum ratio would be close to 1.

This example illustrates some key aspects of the decision problem. The problem is easy if we have features with good discrimination power, such as the office tag; if we can add such tags to the environment, we have solved the problem. But, in general we won't have the benefit of unique single features; rather, we will have to judge how likely it is that a set of features the robot encounters uniquely determines its position, i.e., estimate the value of $L(\mathbf{x}')$ over all environments, for some current set of features.

In this paper we approach these issues from an experimental standpoint. We look at a number of features for matching, including raw range scan information, geometric features, and combinations of features. Using statistical methods, we characterize the suitability of these features for map merging. Finally, we briefly discuss how to efficiently implement map merging.

2.2 Methodology

Our model of map merging is as follows. An *environment* consists of a set of range scans registered correctly into a map. We use 4 large maps of office buildings at widely differing locations: SRI, UW, and CMU. The maps have a total corridor length of approximately 600 meters.

A robot constructs a *partial map* as it explores the environment. The local environment around the robot is called a *patch*. Figure 4 shows a typical office environment, and a set of sample patches. In this paper, we try to solve the decision problem using only a single patch. To make the map merging problem more challenging, we used small patches, consisting of 15 scans taken about 0.5 meters apart. These patches are just on the edge of having enough information to enable reliable merging. Longer patches, of course, make the decision problem easier: in the limit, one could use the whole map. There is also the possibility of using sets of patches for a more reliable decision.

Features are configurations of scan points. For this paper, we consider three basic features: *doors*, *junctions*, and *corners*. These features are marked in the patches at

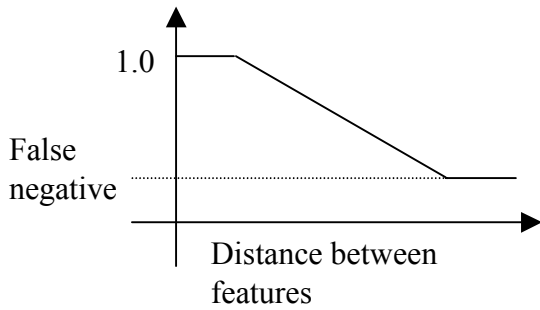


Figure 3 Likelihood model for feature matching. There is a small dead band for short distances, then a linear dropoff.

the right in Figure 4. The doors here are closed, and are somewhat difficult to distinguish from the hallway wall. A patch contains from 2 to 8 features in the data set used. The features were extracted by hand.

Likelihoods are calculated by placing the patch onto an environment at a pose \mathbf{x} . The likelihood $L(\mathbf{x})$ is a function of matching the patch range points against the map, and matching patch features against map features. Because it is geometrically based, the likelihood function automatically takes into account the spatial relationship among features. Details are in the next subsection.

Patches are classified (by hand) into one of a number of *classes*. Each class is a population of patches with certain characteristics in common, e.g., *corridors with no junctions*. The classes considered are:

1. Corridor scans (no junctions)
2. Corridor scans with junctions
3. Single corner scans
4. Multiple corner / junction scans
5. Corridor scans + door features
6. Junction scans + door and junction features
7. Single corner scans + door and corner features
8. Multiple corner scans + door and corner features

Note that the first four classes do not contain extracted features, while the last four do. For each class, we determine the likelihood ratio of Equation (3) over all test

environments, and check the decision against ground truth for various values of β (we fix the value of α as a filter). The results are summarized in *ROC (Receiver Operating Characteristic) curves*, which measure the tradeoff between sensitivity and specificity of the decision method for a class.

2.3 Likelihood Functions

Likelihood $L(\mathbf{x})$ measures the probability of seeing a patch (and its associated features) at the position \mathbf{x} . The likelihood is a combination of the scan match likelihood and the feature match likelihood:

$$(4) L(\mathbf{x}) = L_s(\mathbf{x}) \cdot \prod_i L_i(\mathbf{x})$$

where i is an index over all the features of the patch. The likelihood $L(\mathbf{x})$ is given by the sensor response function $p(r | \mathbf{x}, E)$, the probability that we would see the map patch r from the robot pose \mathbf{x} , given the environment. As shown in [Konolige 1999], the sensor response can be approximated by a correlation operator. A regular grid is imposed on the map area, and for each cell we calculate the probability $p(r_i)$ of the map patch impinging on the cell and $p(E_i)$ of the map impinging on the cell. The correlation operator is:

$$(5) \sum_i p(r_i)p(m_i)$$

Map smearing is employed as an efficient means of approximating uncertainty in the scan readings, and errors in the map.

Feature likelihoods are computed using a simple model that assumes a linear degradation in the match with distance and angle to the nearest similar feature (Figure 3). A small false negative response takes care of the situation in which the robot does not see the feature.

We deliberately chose a very simple and conservative (non-sensitive) feature model, to test the robustness of adding just simple features to the matching process.

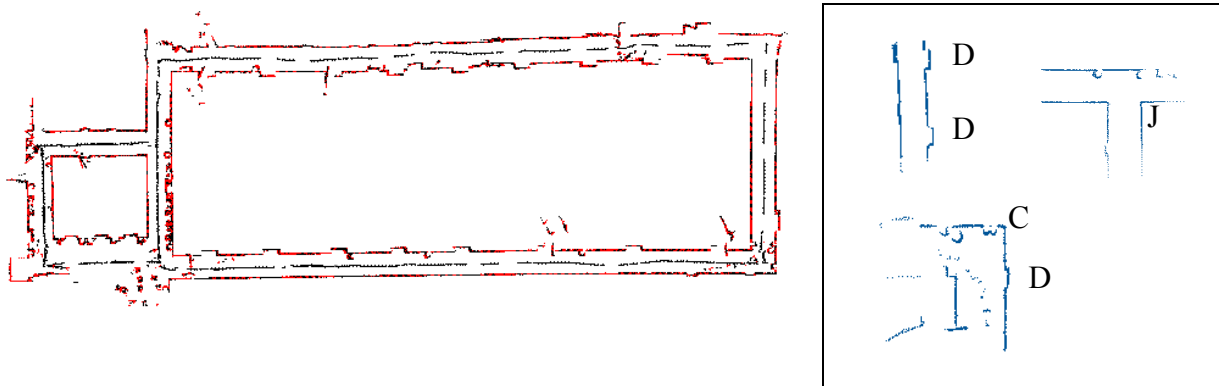


Figure 4 An office environment (~60m long) created from laser range scans, and three patches of 15 range scans extracted from the map: a corridor, a junction, and a corner. Where do they belong in the map? Short patches tend to have ambiguous geometric matches.

3 Experimental Results

For each of the classes mentioned in Section 2.2, we extracted ROC curves, based on varying the parameter β . The results vary dramatically for range scans vs. features.

3.1 Raw Scan Matching

Figure 5 shows the ROC curves for raw scan matching, over four different classes of scans. An ideal discriminator would have a curve with maximum area, i.e., it would lie along the left and top sides of the graph.

Corridors, as expected, have the weakest discrimination, and there are most chance of false positives. Even in different buildings, corridors tend to be about the same width, and there are relatively few features along their length to help distinguish them, using raw range matching. Given the relatively coarse grid resolution (200 mm), there were even some cases in which the highest response was from the wrong location.

The best discrimination, in terms of both sensitivity and selectivity, comes from corner scans that also contain nearby junctions. But even here, using just 4 sample environments, it is easy to make a wrong decision: there is a significant false positive rate, even for low recognition rates.

Looking at the data in terms of Equation (3), the “knee” of the curve occurs at a β of 1.10, which means there is very little margin in setting a robust value for this parameter.

3.2 Feature Matching

When features are added to the matching process, the ROC curves change dramatically (Figure 6). With the exception of the corridor and single corner patches, the response is perfect for certain thresholds, with all patches correctly placed, and no false positives. Although it is not shown in the graph, the robustness of the threshold

parameter β also improves. For values over 2, there are essentially no false positives; while for values under 5, almost all true positives are present.

It is still difficult to match corridors, even with door features identified. In fact, there are several cases in which the wrong match again has a higher response than the correct one. The placement of doorways along a corridor is thus not a very strong discriminator in these indoor environments, apparently because they occur in repeating patterns. On the other hand, door features around junctions and hall corners are much less ambiguous, and lead to good recognition rates.

4 Implementation

In previous work, map merging was possible in the constrained situation of a common starting point, because it limited the search for matches [Gutmann and Konolige 1999]. But in the unconstrained case, correlation of a patch over a partial map is inefficient, and can take minutes on current PC hardware.

The use of features in matching allows for efficient implementations of map merging. Instead of matching the whole patch at every possible pose x , we select poses on the basis of inexact matching of the features in the patch. There are efficient methods for selecting such regions [Grimson 1990]. As a special case, the use of features can help to efficiently solve the *globalization problem*, in which a robot is initially unaware of its position within a given map. [Thrun et al. 2001] present a *Mixture-MCL* model that solves the globalization problem efficiently, by placing sample points at places where the robots sensors determine it is likely to be. As they note, the necessary inverse sensor model is difficult to compute for range scans; but using features can help to solve this problem [Newman et al. 2002; Arras et al. 2001].

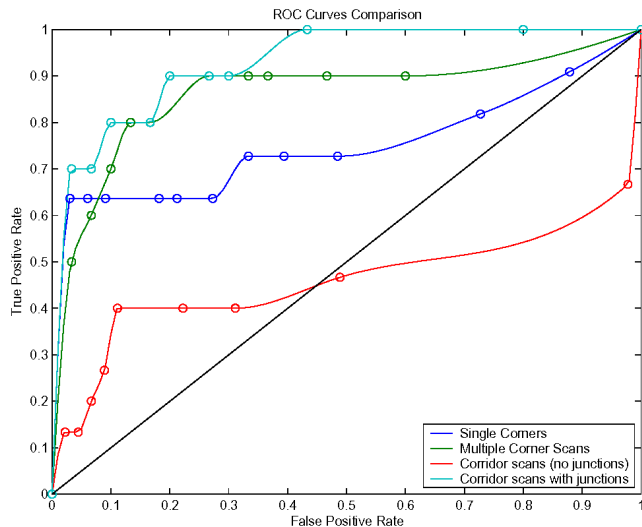


Figure 5 ROC curve for range scan matching, over all four classes of range scans.

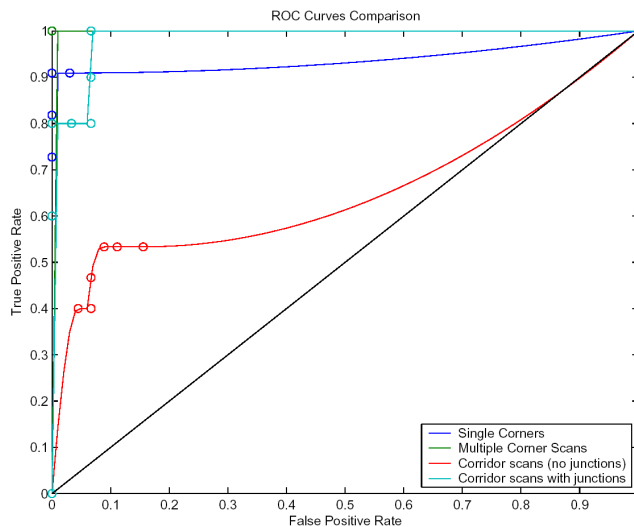


Figure 6 ROC curve for combined range scan and feature matching, over four classes of range scans.

5 Conclusions

Map merging is an interesting and difficult problem, which has not enjoyed the same attention that localization and map building have. It has strong connections to the problem of *globalization*, but differs in that there is always the possibility that two partial maps have no overlap.

The experimental evidence clearly shows the power of extracted features to help in deciding map merging. Small patches, when matched purely on the basis of range information, lack the specificity necessary to make good decisions about merging, even when their sensitivity is sacrificed. Adding geometric features that are commonly found in office environments improves the ROC curves dramatically, and makes it possible to select reliable merges.

The question remains of how to determine good features, and how to extract them. There is a large literature on learning features and landmarks for navigation (e.g., [Fleischer and Marsland, 2002; Thrun, 1998c]) that can be exploited. These algorithms typically perform extraction and learning simultaneously. Alternatively, we could use a set of pre-determined features that are known to have significance, and train or program an extraction routine.

Experimental validation over large environments, long times, and many robots is important for robotics research, as many methods are proposed to solve similar problems. Our project, one of the first large-scale mobile robot coordination efforts, is a contribution in this direction.

6 References

[Arras et al. 2001] K.O. Arras, N. Tomatis, B. Jensen, R.Y. Siegwart. Multisensor on-the-fly localization: Precision and reliability for applications. *Robotics and Autonomous Systems*, 34(2-3), pp.131-143, 2001.

[Burgard et al. 2000] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. *ICRA 2000*.

[Fleischer and Marsland, 2002] J. Fleischer and S. Marsland. Learning to Autonomously Select Landmarks for Navigation And Communication. In *Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, 2002.

[Fox et al. 2000] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325-344, 2000.

[Fox et al. 2001] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In Doucet et al., eds., *Sequential Monte Carlo in Practice*. Springer-Verlag, 2001.

[Grimson, 1990] Grimson, W. E. L. : *Object Recognition by Computer: The Role of Geometric Constraints*. The MIT Press, Cambridge, Mass., 1990.

[Gutmann and Konolige 1999] Gutmann, J. S. and K. Konolige, "Incremental Mapping of Large Cyclic Environments." In *CIRA 99*, Monterey, California, 1999.

[Konolige 1999] Konolige, K. "Markov localization using correlation." In *Proceedings International Joint Conference on Artificial Intelligence (IJCAI' 99)*, Stockholm, 1999.

[Konolige 2002] K. Konolige, D. Guzzoni, and K. Nicewarner. A Multiagent System for Multirobot Mapping and Exploration. *Proceedings of the Workshop on Distributed Robotics*, Washington DC, 2002.

[Newman et al., 2002] P.M. Newman, J.J. Leonard, J. Neira and J.D. Tardós: "Explore and Return: Experimental Validation of Real Time Concurrent Mapping and Localization". *IEEE Int. Conf. Robotics and Automation*, May, 2002.

[Simmons et al. 2000] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. *AAAI 2000*.

[Thrun et al. 1998a] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29-53, 1998. Also appeared in *Autonomous Robots* 5:253-271.

[Thrun et al., 1998b] S. Thrun, J.-S. Gutmann, D. Fox, W. Bugard, and B. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proceedings of the 15th National Conference of the American Association for Artificial Intelligence (AAAI'98)*, July 1998.

[Thrun, 1998c] S. Thrun. Bayesian Landmark Learning for Mobile Robot Localization. *Machine Learning* 33(1), 1998.

[Thrun et al. 2000] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and {3D} mapping. *ICRA 2000*.

[Thrun et al. 2001] S. Thrun, D. Fox, W. Burgard, F. Dellaert. Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence* 128(1-2), 2001.

[Yamauchi 1998] B. Yamauchi. Frontier-based exploration using multiple robots. *Proc. of the Second International Conference on Autonomous Agents*, 1998.