1 Towards Lazy Data Association in SLAM

Dirk Hähnel[1], Sebastian Thrun[2], Ben Wegbreit[2], and Wolfram Burgard[1]

[1] Department of Computer Science, University of Freiburg, Germany
[2] Department of Computer Science, Stanford University, Stanford, CA

Abstract. We present a lazy data association algorithm for the simultaneous localization and mapping (SLAM) problem. Our approach uses a tree-structured Bayesian representation of map posteriors that makes it possible to revise data association decisions arbitrarily far into the past. We describe a criterion for detecting and repairing poor data association decisions. This technique makes it possible to acquire maps of large-scale environments with many loops, with a minimum of computational overhead for the management of multiple data association hypotheses. A empirical comparison with the popular FastSLAM algorithm shows the advantage of lazy over proactive data association.

1.1 Introduction

Simultaneous localization and mapping (SLAM) addresses the problem of a vehicle acquiring a map of its environment while simultaneously localizing itself relative to this map. Most state-of-the-art algorithms approximate posterior distributions over the map and the vehicle pose. In doing so, they accommodate the uncertainty that arises from the robot's sensor noise.

It is widely acknowledged that the SLAM problem consists of a continuous and a discrete component [6,23]. The continuous estimation problem pertains to the location of individual features in the environment and the pose of the robot relative to these features. The discrete aspect of the SLAM problem is the *data association problem* [2,4,14], which is the problem of determining whether or not two features observed at different points in time correspond to one and the same object in the physical world. Data association problems arise when matching two consecutive range scans [13]; or when closing a large cycle in the environment [3,9]. Unfortunately, the number of possible data associations may grow exponentially over time. With unknown data association, the SLAM posterior may (in the worst case) possess exponentially many modes; whereas it commonly contains only a single mode for SLAM problems with known data associations.

The data association problem has been addressed extensively in the SLAM literature [18,22,24]. Most state-of-the-art online data association techniques in SLAM are *proactive*, in the sense that they generate all hypotheses at the time a feature is observed, or a fixed number of time steps thereafter. The most basic example of a proactive strategy is incremental maximum likelihood (ML) data association, which chooses the most likely hypothesis [5]. In more sophisticated approaches, ML data association decisions are made for groups of features at-a-time [22]. Other proactive algorithms generate many data association hypotheses when a feature is observed, and later terminate all but one of them as more sensor data arrives. Two examples of the latter approach are the multi hypothesis Kalman filter (MHT) [2] and particle filter-based algorithms like FastSLAM [10,15,7]. These algorithms are significantly more robust, but at the expense of a much higher computational overhead required for managing many hypothetical maps. The main problem with proactive techniques is computational: In ambiguous situations, multiple data association hypotheses must be generated to ensure that (with high likelihood) the correct association is among them.

This paper seeks to establish a *lazy* data association technique that can "repair" past data association techniques arbitrarily far back into the past. Just as the ML data associator, our approach picks the most likely data association when a feature is observed. However, it differs from ML in that it monitors sensor data to detect whether a different set of data associations (past and present) can yield a map of higher likelihood. When such an opportunity is detected, past data association decisions are revised accordingly. We illustrate our approach in the context of several challenging mapping task, one involving a subterranean vehicle mapping a mine. A comparison with the popular FastSLAM algorithm illustrates the advantages of lazy over proactive data association.

1.2 Preliminaries

1.2.1 SLAM with Known Data Association

We adopt the common probabilistic formulation of the SLAM problem. The vehicle pose at time is denoted ξ_t; we will use $\xi_{1:t}$ to denote the sequence of vehicle poses from time 1 to time t. The environment of the vehicle is composed of N features whose locations will be denoted $\Theta = \theta_1, \ldots, \theta_N$; the vehicle's estimate of Θ is the map. The goal of SLAM is to recover the map of Θ and the vehicle path $\xi_{1:t}$ from sensor measurements and robot controls. Measurements will be denoted by z_t, and controls by u_t. The goal of a SLAM algorithm is to recover the posterior distribution over the map and the vehicle pose $p(\xi_{1:t}, \Theta \mid z_{1:t}, u_{1:t})$.

Under *known data association* and Gaussian noise, this posterior can be estimated using the extended Kalman filter (EKF) [16,21,20]. However, the EKF is inefficient in that its update requires time quadratic in N. A flurry of recent research has led to a number of algorithms that can perform the update in constant time under the assumption of known data association [11,15,26].

1.2.2 Incremental Data Association

Let $c_{1:t}$ be a vector of *correspondence variables*, which are a discrete variable whose values are in $\{1, 2, \ldots, N\}$. If $c_t = n$, the measurement z_t corresponds to the feature θ_n. If $c_t = c_s$ for two different points in time s and t, both measurements detected the same object in the physical world. We now seek to identify the sequence of correspondence variables $c_{1:t}$ that maximize the posterior

$$\hat{c}_{1:t} = \underset{c_{1:t}}{\operatorname{argmax}}\, p(\xi_{1:t}, \Theta \mid z_{1:t}, u_{1:t}, c_{1:t}) \tag{1.1}$$

Unfortunately, the maximization is carried out over t variables; furthermore, these variables interact, and there is an exponential number of values that the combined

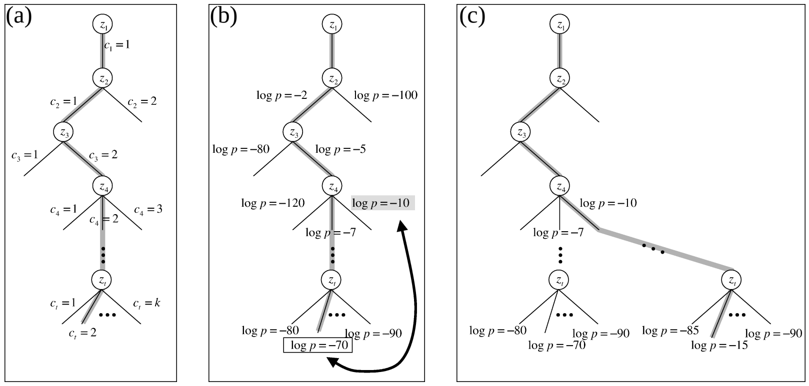

Figure 1.1. (a) The data association tree, whose branching factor grows with the number of landmarks in the map. (b) The proposed algorithm maintains the log-likelihood for the entire frontier of expanded nodes, enabling it to find alternative paths. (c) Improved path.

vector $c_{1:t}$ can take. This exponential complexity makes finding the correct data association difficult.

Incremental ML approaches to SLAM bypass this problem by estimating the t-th data association $\hat{c}_t$ at the time the t-th measurement arrives; and freeze it forever after. This is equivalent to assuming that the optimal setting of the association variables can be found by optimizing one after the other:

$$\hat{c}_t = \underset{\hat{c}_t}{\operatorname{argmax}}\, p(\xi_{1:t}, \Theta \mid z_{1:t}, u_{1:t}, \hat{c}_t, \hat{c}_{1:t-1}) \tag{1.2}$$

where the variables $\hat{c}_{1:t-1}$ are held constant in the optimization on the right-hand side. Figure 1.1a illustrates this approach. Shown there is the tree of possible data association variables, with time 1 at the root and time t at the leaves. The incremental approach greedily follows what appears to be the most likely path at each step, from the root on down. This is illustrated by the path highlighted in gray in Figure 1.1a. Unfortunately, once a wrong choice has been made, the incremental ML approach cannot recover. Moreover, wrong data association decisions introduce errors in the map which, subsequently, can induce more errors in the data association. For this reason, incremental ML data association is considered brittle in SLAM.

1.3 Lazy Data Association

1.3.1 Tree Search

Our approach uses a search procedure for considering alternative data association decisions not just at the present time step, but also for time steps in the past. A simple argument (reminiscent of that underlying the correctness of the A* algorithm [19]) enables us to drastically reduce the number of nodes expended during this search. Figure 1.1b illustrates the basic idea: Our approach maintains not just a single path through the data association tree, but an entire frontier. Every time a node is expanded (e.g., through incremental ML), all alternative outcomes are also assessed and the corresponding likelihoods are memorized. This is illustrated in Figure 1.1b, which depicts the log-likelihood for an entire frontier of the tree. Notice that we chose to represent the likelihood values as log-likelihoods, which is numerically more stable than probabilities.

Finding the maximum in Equation (1.1) implies that the log-likelihood of the chosen leaf is greater or equal to that of any other leaf at the same depth. Since the log-likelihood decreases monotonically with the depth of the tree, we can guarantee that we indeed found the optimal data association values when the log-likelihood of the chosen leaf is greater or equal to the log-likelihood of any other node on the frontier. Put differently, when a frontier node assumes a log-likelihood greater than the one of the chosen leaf, there might be an opportunity to further increase the likelihood of the data by revising past data association decisions. Our approach then simply expands such frontier nodes. If an expansion reaches a leaf, this leaf is chosen as the new data association; otherwise the search is terminated when the entire frontier possesses values that are all smaller or equal to the one of the chosen leaf. This approach is guaranteed to always maintain the best set of values for the data association variables; however, occasionally it might require substantial search.

1.3.2 Equivalency Constraints

The key missing link is a SLAM representation that lets us efficiently modify data association variables. Our approach effectively implements the idea of global data association, but it does so via a set of auxiliary variables, called *equivalency variables*. Each such variable is of the form $\gamma(t, s)$ where t and s are two different points in time. The equivalency relation $\gamma(t, s)$ holds if and only if both c_t and c_s correspond to the same physical landmark, that is, $c_t = c_s$. Clearly, each assignment of the data association variables $c_{1:t}$ defines a set of such equivalency relationships. Conversely, each set of equivalency relationships constrains the space of all valid data association values. In the limit as all equivalency relations are recovered, the data association can be determined up to a simple index permutation (which can never be recovered, since the index of a feature is arbitrary). The optimization problem defined in Equation (1.1), thus, becomes one of finding equivalency constraints.

Let Γ define an arbitrary set of such pairwise equivalency constraints. Each constraint $\gamma \in \Gamma$ is of the form $\gamma(t, s)$. The goal then is to identify the optimal set of constraints Γ, that is, the set of constraints that maximizes the posterior:

$$\operatorname*{argmax}_{\Gamma} p(\xi_{1:t}, \Theta \mid z_{1:t}, u_{1:t}, \Gamma) \tag{1.3}$$

The notion of equivalency constraints makes it possible to relate two features to each other even though their absolute identity is unknown.

1.3.3 Recovering the Path Posterior under Equivalency Constraints

Our approach for recovering the path posterior is similar in spirit to the Lu/Milios algorithm [12]. The *key* insight is that equivalency constraints can be “translated” into soft constraints that tie together two poses in the path posterior. More specifically, consider the constraint $\gamma(t, s)$. This constraint can be ‘softened’ into a constraint that ties together the location of the feature detected by z_t and the one detected by z_s. This quadratic constraint is of the form

$$[f(z_t, \xi_t) - f(z_s, \xi_s)]^T \; R \; [f(z_t, \xi_t) - f(z_s, \xi_s)] \tag{1.4}$$

Here R is a quadratic penalty, and f is the function that projects the measurement z_t into 3D coordinates, based on the robot pose ξ_t. In general, f is a non-linear projection; however, by approximating it with a first order Taylor expansion we obtain a quadratic constraint of the form

$$\left[A \begin{pmatrix} \xi_t \\ \xi_s \end{pmatrix} - a\right]^T \; R \; \left[A \begin{pmatrix} \xi_t \\ \xi_s \end{pmatrix} - a\right] \tag{1.5}$$

with a Jacobean matrix A and a vector a (indexes omitted for brevity). This quadratic constraint captures the information pertaining to the robot path, which arises from multiple sightings of the same feature according to the data association constraint $\gamma(t, s)$.

Additional constraints for the robot path originate from the robot controls $u_{1:t}$, here written in negative $\log$-form

$$\begin{aligned}
-\log p(\xi_{1:t} \mid u_{1:t}) \;&=\; -\log p(\xi_1) - \sum_{\tau=2}^{t} \log p(\xi_t \mid u_t, \xi_{t-1}) \\
&= \text{const.} \; + \tfrac{1}{2} \sum_{\tau=2}^{t} [\xi_\tau - g(u_\tau, \xi_{\tau-1})]^T \; Q_\tau \; [\xi_\tau - g(u_\tau, \xi_{\tau-1})] \\
&\approx \text{const.} \; + \tfrac{1}{2} \sum_{\tau=2}^{t} \left[B_\tau \begin{pmatrix} \xi_\tau \\ \xi_{\tau-1} \end{pmatrix} - b_\tau\right]^T \; Q_\tau \; \left[B_\tau \begin{pmatrix} \xi_\tau \\ \xi_{\tau-1} \end{pmatrix} - b_\tau\right]
\end{aligned} \tag{1.6}$$

Each B_τ is a Jacobean matrix, and b is a vector. The last step of this approximation involves again a Taylor expansion, in which the nonlinear motion model is linearized. Adding Equations (1.5) and (1.6) together leads to a system of quadratic equations in the path variables $\xi_{1:t}$ of the form

$$J := \text{const.} \; + [C \cdot \xi_{1:t} - c]^T \;\; P \;\; [C \cdot \xi_{1:t} - c] \tag{1.7}$$

Here C is a sparse matrix that links together elements in the path vector, and c is a vector. The matrix P defines a Mahalanobis distance composed of the terms R and Q_τ. This quadratic function is (up to a constant) the logarithm of a Gaussian approximation to the posterior over the robot path under the constraint set Γ. The mean of this Gaussian is recovered by setting the first derivative to zero:

$$\frac{\partial J}{\partial \xi_{1:t}} = C^T \; P \;\; [C \cdot \xi_{1:t} - c] \;\; \stackrel{!}{=} \;\; 0 \tag{1.8}$$

The solution to this equality is given by

$$\xi_{1:t} = (C^T \; P \; C)^{-1} \; C^T \; P \; c \tag{1.9}$$

where the matrix in the inversion is high-dimensional but extremely sparse. The covariance is simply the second derivative of J:

$$\frac{\partial^2 J}{\partial \xi_{1:t}^2} = C^T \; P \; C \tag{1.10}$$

The mean and covariance are the estimate of the SLAM posterior at time t. The classical solution to this problem involves the inversion of a spare matrix, which is costly [8]. However, there exists a number of efficient approximations, such as loopy belief propagation [17] and tree-based approximation techniques [27] for approximating these quantities; all of those techniques can exploit an existing solution when modifying the set of equality constraints.

The key insight into this set of quadratic equations is that incrementally adding or removing a constraint can be done computationally very efficiently. This is the direct result of the matrix inversion lemma. Suppose we would like to add a constraint of the form $\gamma(t, s)$. This leads to a local modification of the matrix C and the vector c in (1.7), which involve only ξ_t and ξ_s:

$$\begin{aligned} C' &\longleftarrow C + S \Delta S^T \\ c' &\longleftarrow c + S\delta \end{aligned} \tag{1.11}$$

Here Δ and δ are a low-dimensional matrix, and vector, respectively. The matrix S is a projection matrix for mapping low-dimensional vectors back into high-dimensional spaces. According to the inversion lemma gives us a new solution for the inverse term in (1.9):

$$\begin{aligned} (C'^T \; P \; C')^{-1} &= ([C + S\Delta S^T]^T \; P \; [C + S\Delta S^T])^{-1} \\ &= (C^T \; P \; C + 2S^T \Delta^T SPC + S^T \Delta^T S \; P \; S^T)^{-1} \end{aligned} \tag{1.12}$$

1.3.4 Incorporating Negative Measurement Information

The approach described thus far only accounts for matching points in 3D space. Equality constraint bend the path of the robot relative to the path reconstructed from pure odometry. As a result, the likelihood maximizing set of equality constraints would be the empty set, and each feature would simply be declared a new one. This is because a bent path is less likely under the robot's control variables than a non-bent one.

The problem with the approach thus far is that it does not account for "negative" information. Negative information pertains to situations where a robot fails to see a measurement. Range sensors, which are brought to bear in our implementation, return positive and negative information with regards to the presence of objects in the world. The positive information are object detections. The negative information applies to the space between the detection and the sensor. The fact that the robot failed to detect an object closer than its actual reading provides information about the *absence* of an object within the measurement range.

To evaluate the effect of a new constraint on the overall likelihood of the data, our approach evaluates both types of information: positive and negative. Both types

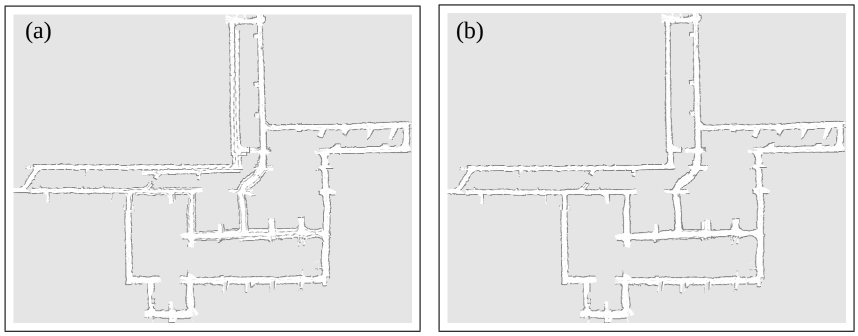

Figure 1.2. (a) Mine map with incremental ML scan matching and (b) using our lazy data association approach. The map is approximately 250 meters wide, and acquired without odometry information.

are obtained by calculating the pairwise (mis)match of two scans under their pose estimate. In our implementation, the log-likelihood of each measurement is obtained by superimposing a scan onto a local occupancy grid map build by another scan. In doing so, it is straightforward to determine the probability of a measurement in a way that incorporates both the positive and the negative information.

1.4 Experimental Results

We have implemented a version of our algorithm complex enough to permit testing in realistic setting with large-scale data sets. Our first data set was acquired in an abandoned mine [1,25], using a vehicle equipped with a laser range finder. The vehicle does not provide any odometry or controls information; and no values for $u_{1:t}$ are available. A further difficulty arises from the absence of well-defined "landmarks" in the mine. When faced with raw $O(10^9)$ laser measurements, the number of possible data association variables is beyond what can be handled computationally. To make this problem tractable, we modified the basic algorithm in a number of ways. Instead of using all poses in the optimization, our approach acquires local occupancy maps of approximately five meters length, assuming that within a local map, the incremental ML data association techniques works reliable (which in practice it does). As a result, we only have to align a few hundred local maps, making the problem computationally tractable. Further, we have not yet implemented the most efficient version of our algorithm (e.g., we are not using Equation (1.12)), which makes our implementation slower than real-time.

The left panel of Figure 1.2a depicts the result of incremental ML data association, which is equivalent in our case to regular incremental scan matching. Clearly, certain corridors are represented doubly in this map, illustrating the shortcomings of the ML approach. The right panel, in comparison, shows the result of our approach. Clearly, this map is more accurate than the one generated by the incremental ML approach. Its diameter is approximately 250 meters wide, and the floor of the mine is highly uneven.

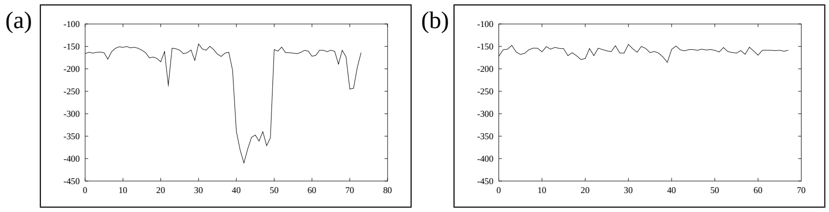

Figure 1.3. (a) Log-likelihood of the actual measurement, as a function of time. The lower likelihood is caused by the wrong assignment. (b) Log-likelihood using our approach, which recursively fixed false data association hypotheses. The success of our approach is manifested by the lack of a distinct dip.

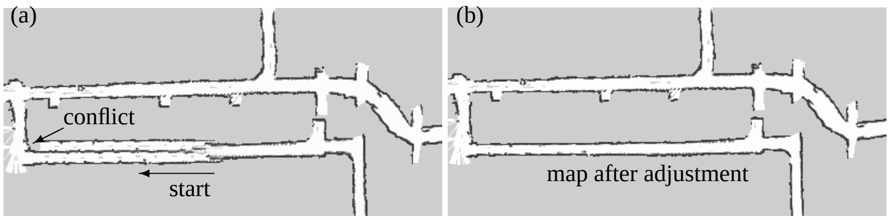

Figure 1.4. Example of our lazy data association technique: (a) When closing a large loop, the robot first erroneously assumes the existence of a second, parallel hallway. However, this model leads to a gross inconsistency as the robot encounters a corridor at a right angle. At this point, our approach recursively searches for improved data association decisions, arriving on the map shown in diagram (b).

Figure 1.3a illustrates the log-likelihood of the most recent measurement (not the entire path), which drops significantly as the map becomes inconsistent. At this point, our approach engages in searching alternative data association values. It quickly finds the “correct” one and produces the map shown in Figure 1.2b. The area in question is shown in Figure 1.4, illustrating the moment at which the likelihood takes its dip. The log-likelihood of the measurement for our approach is shown in Figure 1.3b.

We also compared lazy data association with a popular proactive one, the FastSLAM algorithm [10,15,7]. Our comparison is based on the implementation in [10], which addresses mapping with laser range finders (instead of idealized point features). The data set was gathered in a large indoor environment, using a Pioneer 2 robot equipped with a laser range-finder. To make this problem difficult, the robot first traversed a small loop a number of times, before closing a larger loop. This is shown in Figure 1.5a. Figure 1.5b shows the result for the incremental ML approach, which is implemented here as an incremental scan matching algorithm. FastSLAM’s map is shown in Figure 1.5c, for 100 particles. Both of these maps show inconsistencies in the upper left corner. Our approach produces the map in Figure 1.5d, which is significantly more accurate.

When mapping the small cycle, FastSLAM runs out of particles. This is shown in Figure 1.6, which plots the particles (including their paths) before and after closing the small cycle on the right. While in principle, the problem can be reduced by using larger particle sets, eventually such a deprivation takes place, posing intrinsic limits on FastSLAM's ability to map large environments with many cycles. This is a fundamental problem inherent in all proactive approaches; FastSLAM is among the most robust proactive approaches in the present literature.

1.5 Conclusion

This paper described a new algorithm for data association in SLAM. In essence, our approach searches the combinatorial tree of possible data association decisions. The search is lazy: only when an alternative assignment shows promise will it be evaluated. To implement this efficiently, our approach condenses maps into graphical representation, and employs equality constraints for alleged data associations. Using linear algebra techniques, these constraints can be added or removed efficiently. We have evaluated our approach using some of the most challenging data sets in our possession, and have consistently found that it produces accurate maps, even if for maps with many large cycles.

Acknowledgments

The research has been sponsored by DARPA's MARS Program (contracts N66001-01-C-6018 and NBCH1020014), which is gratefully acknowledged. The authors also acknowledge an inspiring discussion with Peter Cheeseman who suggested to explore lazy data association techniques.

References

1. C. Baker, Z. Omohundro, S. Thayer, W. Whittaker, M. Montemerlo, and S. Thrun. A case study in robotic mapping of abandoned mines. In *Proceedings of FSR-2003*.
2. Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
3. M. Bosse, P. Newman, M. Soika, W. Feiten, J. Leonard, and S. Teller. An atlas framework for scalable mapping. In *Proceedings of ICRA-2003*.
4. J.C. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.
5. G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. Working notes ICRA'2000 Workshop W4: Mobile Robot Navigation and Mapping, 2000.
6. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001.
7. A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Proceedings of IJCAI-03*.
8. Anshul Gupta, George Karypis, and Vipin Kumar. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Transactions on Parallel and Distributed Systems*, 8(5):502–520, 1997.
9. J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of CIRA-00*.

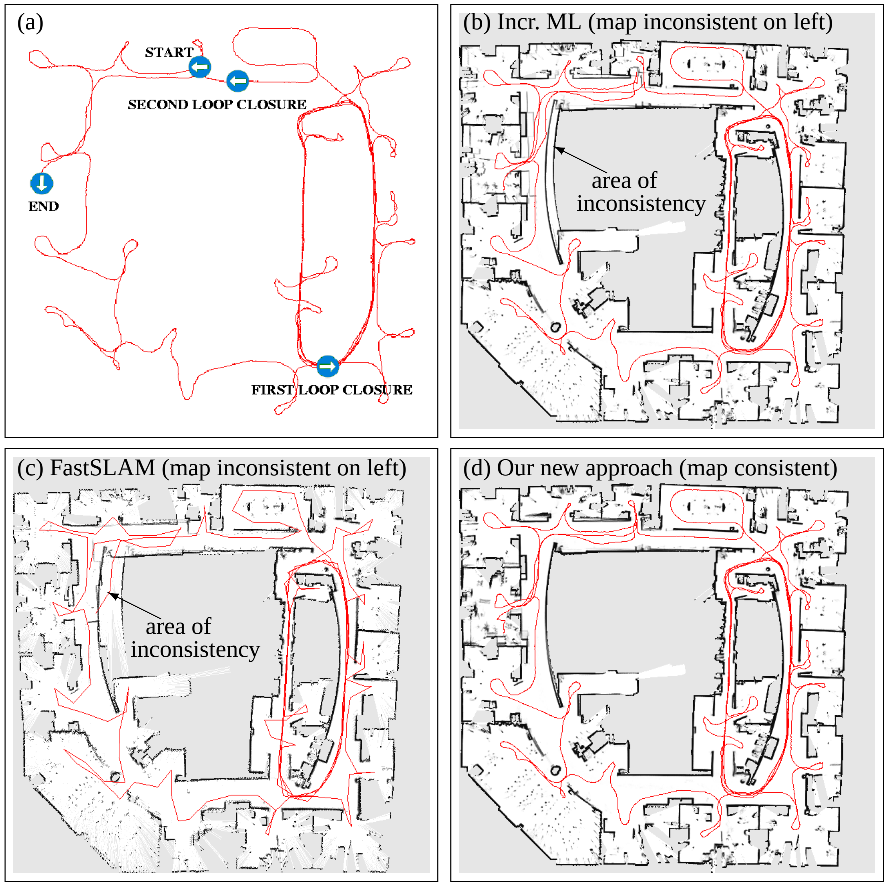

Figure 1.5. (a) Path of the robot. (b) Incremental ML (scan matching) (c) FastSLAM. (d) Our approach.

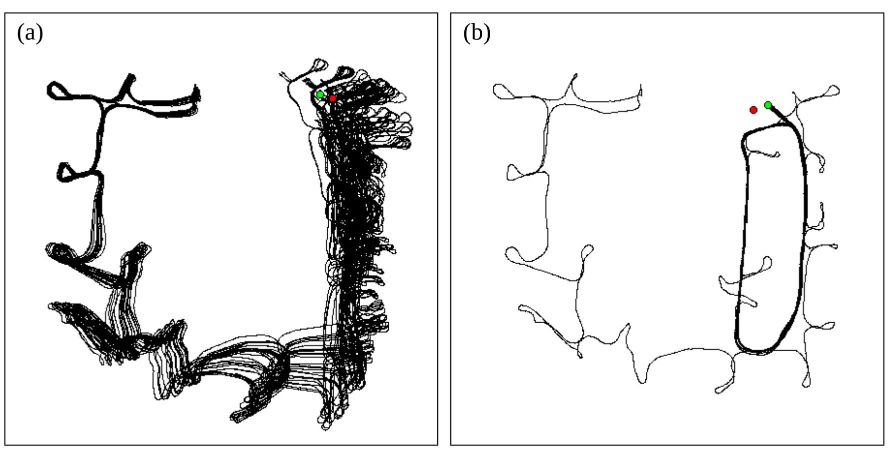

Figure 1.6. The problem with FastSLAM is particle deprivation: (a) Particle paths before closing the small loop on the right and (b) after closing it.

10. D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proceedings of IROS-03*.
11. J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Proceedings of ISRR-99*.
12. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
13. F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1998.
14. M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proceedings of ICRA-03*.
15. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of IJCAI-03*.
16. P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *Proceedings of ISER-89*.
17. K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of UAI-99*.
18. J. Neira and J.D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.
19. N. J. Nilsson. *Principles of Artificial Intelligence*. Springer Publisher, 1982.
20. R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, 167–193. Springer, 1990.
21. R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.
22. J.D. Tardós, J. Neira, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using sonar data. TR TM 2001-04, MIT Marine Robotics Lab, 2001.
23. S. Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
24. S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.
25. S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Proceedings of ICRA-03*.
26. S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A.Y. Ng. Simultaneous mapping and localization with sparse extended information filters. In J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, *Proceedings of WAFR-02*.
27. M. J. Wainwright. *Stochastic processes on graphs with cycles: geometric and variational approaches*. PhD thesis, MIT, 2002.