
Gaussian Multi-Robot SLAM

Yufeng Liu and Sebastian Thrun
School of Computer Science
Carnegie Mellon University

Abstract

We present an algorithm for the multi-robot simultaneous localization and mapping (SLAM) problem. Our algorithm enables teams of robots to build joint maps, even if their relative starting locations are unknown and landmarks are ambiguous—which is presently an open problem in robotics. It achieves this capability through a sparse information filter technique, which represents maps and robot poses by Gaussian Markov random fields. The alignment of local maps into a single global maps is achieved by a tree-based algorithm for searching similar-looking local landmark configurations, paired with a hill climbing algorithm that maximizes the overall likelihood by search in the space of correspondences. We report favorable results obtained with a real-world benchmark data set.

1 Introduction

Simultaneous localization and mapping, or SLAM, addresses the problem of acquiring an environment map with one or more moving vehicles [3]. In statistical terms, SLAM is a high-dimensional estimation problem characterized by two major sources of uncertainty, pertaining to the noise in sensing and in motion. Whereas SLAM has mostly been addressed for single robot systems, the problem is particularly challenging for multiple robots that seek to cooperate when acquiring a map. For single-robot SLAM, the “classical” solution is based on the extended Kalman filter, or EKF [13]. EKFs are relatively slow when estimating high-dimensional maps. Recent research has led to a flurry of more capable algorithms, introducing concepts such as hierarchical maps [1, 4, 15], particle filters [7, 8], information filters [17], and junction trees [10] into the SLAM literature.

This paper addresses the topic of multi-robot SLAM, where multiple vehicles seek to build a joint map. The problem is not new: A number of papers addresses the problem under the constraint that the initial pose of all robots relative to each other is known exactly [2, 9, 12] or in approximation [5, 16]. All of those papers sidestep an important data association question: If two robots discover similar maps, are they actually in the same environment, or are these two different parts of the environment that look alike? Clearly, this question is trivially answered if the robots start at the same location. If their initial location is unknown—which is the case addressed in this paper—it creates a challenging data association problem. This problem was addressed in a recent paper [14]: the idea here is that robots continuously attempt to localize themselves in each other’s maps using particle filters. While this is a highly promising approach, it is computationally somewhat expensive, due to the high costs of running K^2 particle filters for K robots.

In this paper, we address the multi-robot SLAM problem from a geometric/Bayesian perspective. Our approach is orthogonal to the one in [14]: Instead of localizing robots in each other’s maps, we compare local maps acquired by the robots. To accommodate the uncertainties in these maps, our approach builds on an algorithm that represents robots maps by sparse Gaussian Markov random fields (GMRFs) [17, 18]. We define fast tree-based matching algorithms

for maps in GMRF representation, which answer questions like: *What is the probability that two maps overlap?* and *what is the most likely overlap?* We define a multi-robot SLAM algorithm that enables teams of robots to identify map overlap, and gradually construct a single large map even under total ignorance with regards to their relative initial location. Experimental results show that our approach works well in large-scale environments.

2 The Multi-Robot SLAM Problem

We are given K mobile robots equipped with environment sensors. The robots operate in an environment populated by N stationary landmarks whose Cartesian location are denoted $Y = y_1 \dots y_N$. Each robot’s pose (i.e., its coordinates and orientation) is a function of time and will be denoted x_t^k ; here t is the time index, and k is the index of the robot. We will collectively denote the set of all robot pose variables at time t by $X_t = x_t^1 \dots x_t^K$.

At each point in time, each robot executes a motion command, denoted u_t^k , which subsequently alters its pose. This pose transition is governed by a function g^k

$$x_{t+1}^k = g^k(x_t^k, u_t^k) + \mathcal{N}(0; G_t^k) \quad (1)$$

where $\mathcal{N}(0; G_t^k)$ denotes a zero-mean Gaussian noise variable with covariance G_t^k . Each robot can also sense relative information to nearby landmarks (e.g., range and bearing). The k -th robot’s measurement vector at time t will be denoted z_t^k . Measurements are governed through the function h^k

$$z_t^k = h^k(x_t^k, Y) + \mathcal{N}(0; Q_t^k) \quad (2)$$

where Q_t^k is the covariance of the measurement noise. The objective of multi-robot SLAM is to estimate a posterior $p(X_t, Y | Z, U)$ over all robot poses X_t and all landmark locations Y from all available data, Z and U . Here Z is the set of all measurements acquired by all robots from time 0 to time t ; U is the set of all controls.

3 Sparse Extended Information Filters for Multi-Robot Systems

The classical solution to the SLAM problem is the extended Kalman filter (EKF) [13]. The EKF approximates the posterior $p(X_t, Y | Z, U)$ by a multivariate Gaussian, with mean vector μ_t and covariance Σ_t . Updating this Gaussian is achieved by linearizing g and h at μ_t , and applying the standard Kalman filter equations. Unfortunately, even a single vehicle measurement will generally affect all parameters of the Gaussian. Therefore, updates require time quadratic in $N+K$, which is prohibitively slow when applied in environments with many landmarks (or features). Moreover, the EKF is *not* easily decomposed for decentralized execution on multiple vehicles, as discussed in [9].

In a recent workshop paper [17], we have proposed an efficient SLAM algorithm called *sparse extended information filter*, or SEIF. SEIFs represents the posterior $p(X_t, Y | Z, U)$ by a *sparse* Gaussian Markov random field (GMRF), which makes it possible to (1) represent the entire SLAM posterior in $O(N + K)$ memory, and (2) incorporate new sensor measurements and controls in constant time per update, independent of the size of the map N or the number of robots K . SEIFs were originally proposed for single-robot SLAM; here we show that they can be extended to multi-robot SLAM.

A key idea in SEIFs is to represent the SLAM posterior in the natural parameters of multivariate Gaussians. The natural parameters consist of an information matrix H_t and an information vector b_t , which are defined as follows: $H_t = \Sigma_t^{-1}$ and $b_t = \mu_t^T \Sigma_t^{-1}$. As one easily shows, both representations of Gaussians—moments and natural parameters—are mathematically equivalent (up to a constant factor that is easily recovered):

$$\mathcal{N}(\mathbf{x}; \mu_t, \Sigma_t) \propto \exp -\frac{1}{2}(\mathbf{x} - \mu_t)^T \Sigma_t^{-1} (\mathbf{x} - \mu_t) \propto \exp -\frac{1}{2}\mathbf{x}^T H_t \mathbf{x} + b_t \mathbf{x} \quad (3)$$

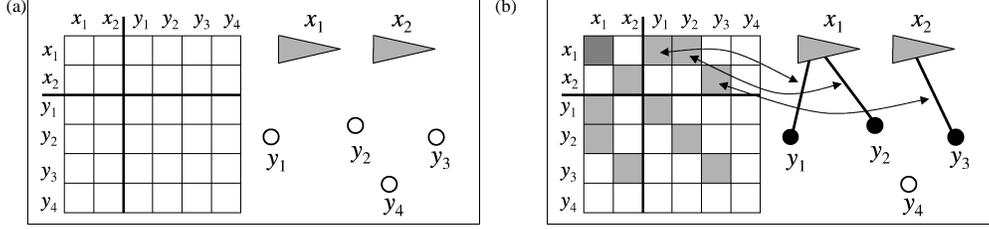


Figure 1: The effect of measurements on the information matrix and the associated network of landmarks: (a) Observing y_1 results in a modification of the information matrix elements H_{x_t, y_1} . (b) Observing y_2 affects H_{x_t, y_2} .

The most important insight in SEIFs is that in SLAM problems, the covariance Σ_t is fully populated whereas the information matrix H_t is dominated by a small number of elements. This insight has opened the door to the development of update rules that maintain a sparse information matrix H_t [17]; for those each update step manipulates only a small number of elements. Furthermore, all updates are additive, which facilitates their distributed (and decentralized) implementation.

Measurement Update. Let \bar{H}_t and \bar{b}_t denote the filter state *before* the measurement update. The parameters H_t and b_t are obtained from \bar{H}_t and \bar{b}_t by virtue of the following additive update equations:

$$H_t = \bar{H}_t + C_t Q_t^{-1} C_t^T \quad \text{and} \quad b_t = \bar{b}_t + (z_t - \hat{Z}_t + C_t^T \mu_t)^T Z^{-1} C_t^T \quad (4)$$

Here Q_t is the joint covariance of the multi-robot measurement error. Q_t is a generalized diagonal matrix composed of the individual measurement error covariances Q_t^1, \dots, Q_t^K . $C_t = \nabla_{X_t Y} h(\mu_t)$ is the Jacobian of the joint measurement function h at μ_t . This Jacobian is naturally sparse: it only involves the robot poses and the observed landmarks. \hat{Z}_t is the predicted joint measurement for all robots. Notice that the update (4) is additive; it is easily decomposed into individual-robot updates, making it amenable to a decentralized implementation.

Figure 1 illustrates the measurement update and its effect H_t for a simple example involving two robots (and assuming known data association). Here robot 1 senses landmarks y_1 and y_2 , and robot 2 senses y_3 . The matrix C_t is zero except for the state variables that influence these measurements. Consequently, the measurement update manipulates the off-diagonal elements $H_{x_t^1, y_1}$, $H_{x_t^1, y_2}$, and $H_{x_t^2, y_3}$ and its symmetrical counterparts, H_{y_1, x_t^1} , H_{y_2, x_t^1} , and H_{y_3, x_t^2} , and the corresponding elements on the main diagonal. Graphically, these updates are equivalent to adding links in the GMRF between x^1 and y_1 , x^2 and y_1 , and x^2 and y_3 , as illustrated.

To implement this motion update on actual robots, it is necessary to determine the identity of observed landmarks. This data association decision is commonly made on a maximum likelihood basis [3], allowing for the provision that an observed landmark has never been seen before. The calculation of the likelihood of a measurement under a data association hypothesis is achieved by computing the marginal distribution $p(x_t^k, y_n | Z, U)$ of the robot pose x_t^k and the landmark y_n in question. This marginal is approximated by the following Gaussian:

$$\begin{aligned} \Sigma_{t:n} &= S_{x^k, y_n}^T (S_{x^k, y_n, Y_n^+}^T H_t S_{x^k, y_n, Y_n^+})^{-1} S_{x^k, y_n} \\ \mu_{t:n} &= \mu_t S_{x^k, y_n} \end{aligned} \quad (5)$$

Here S_{x^k, y_n} is a projection matrix that extracts from the state vector the k -th robot pose and the coordinates of the n -th landmark. This expression can be tightly approximated in constant time for each landmark under consideration, regardless of the size of the map N and the number of robots K .

Motion Update. Motion updates manipulate the information in less obvious ways, but they are still additive and local. The equations for the general case are somewhat involved (see [17] for a full derivation). Here S_X is again a projection matrix, which extracts all robot pose

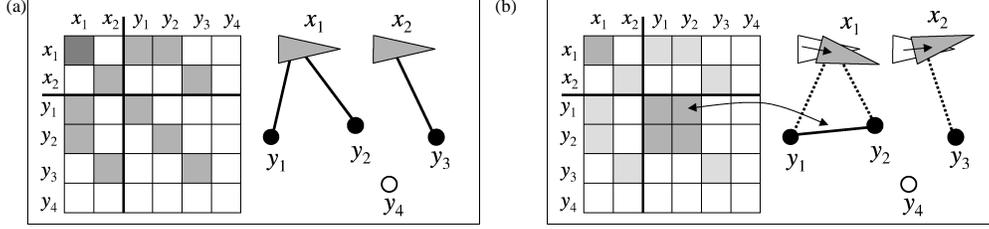


Figure 2: The effect of motion on the information matrix and the associated network of landmarks: (a) before motion, and (b) after motion. Motion updates introduce new links (or reinforce existing links) between any two active landmarks, while weakening the links between the robot and those landmarks. This step introduces links between pairs of landmarks.

variables $X = x^1 \dots x^K$ form the full state vector.

$$\begin{aligned}
 \bar{H}_t &= H'_{t-1} - \Delta H_t \\
 \bar{b}_t &= b_{t-1} - \mu_{t-1}^T (\Delta H_t + H'_{t-1} - H_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
 \text{with } H'_{t-1} &= \Psi_t^T H_{t-1} \Psi_t \\
 \Delta H_t &= H'_{t-1} S_X [U_t^{-1} + S_X^T H'_{t-1} S_X]^{-1} S_X^T H'_{t-1} \\
 \Psi_t &= I - S_X (I + [S_X^T A_t S_X]^{-1})^{-1} S_X^T
 \end{aligned} \tag{6}$$

This update relies once again on linearization using Taylor series expansion, where $A_t = \nabla_X g(u_t, \mu)$. As before, this update is additive, and it is easily decomposed into individual additive updates for each robot, hence can be executed decentrally. In [17], we prove that each robot can perform this update in constant time if H_t is sparse, regardless of the number of landmarks N . This proof generalizes to the multi-robot case.

Graphically, the motion update is shown in Figure 2. As the robots move, the update equations reduce the strength of all elements in the H_t matrix in the row/column of the robot pose vector. In the GMRF representation, this amounts to weakening all links directly connected to the pose of a moving robot, by shifting weight to the between-landmark links of landmarks linked to the robot, as sketched in Figure 2b. From this diagram, it is easy to see that motion updates can be decomposed into individual-robot updates, hence can be run decentrally.

Sparsification. Regular information filters as described so far generate densely populated information matrices, despite the fact that those matrices are dominated by a linear number of elements [17]. To enforce sparseness, which is necessary for the multi-robot decomposition, SEIFs include an approximation step called *sparsification*, which removes elements in the H -matrix between a robot and a landmark without introducing new links. This step is reminiscent of the *arc removal* technique known from Bayes networks [11], with the time required for removing an arc being independent of the size of the GMRF. The sparsification is graphically illustrated in Figure 3. By removing arcs between the robot and specific landmarks, the fan-in and fan-out of every node in the GMRF can be controlled, and the resulting matrix H_t remains sparse.

The update equations are also somewhat involved, and their derivation can be found in [17]. As before, we denote a robot pose by x^k . All landmarks connected to this robot are collectively called Y^+ . Y^0 is the landmark whose link to the robot is being removed in the sparsification step, and Y^- refers to all other landmarks. The outcome of this step is given by \tilde{H}_t and \tilde{b}_t .

$$\begin{aligned}
 \tilde{H}_t &= H_t - H'_t L'_t \\
 \tilde{b}_t &= b_t - b'_t L'_t S_{Y^0} S_{Y^0}^T + (\mu_t^T \tilde{H}_t - b_t) S_{x^k, Y^+} S_{x^k, Y^+}^T \\
 \text{with } H'_t &= S_{x^k, Y^+, Y^0} S_{x^k, Y^+, Y^0}^T H_t S_{x^k, Y^+, Y^0} S_{x^k, Y^+, Y^0}^T \\
 b'_t &= b_t S_{x^k, Y^+, Y^0} S_{x^k, Y^+, Y^0}^T \\
 L'_t &= [S_{x^k} (S_{x^k}^T H_t S_{x^k})^{-1} S_{x^k}^T + S_{Y^0} (S_{Y^0}^T H_t S_{Y^0})^{-1} S_{Y^0}^T]
 \end{aligned} \tag{7}$$

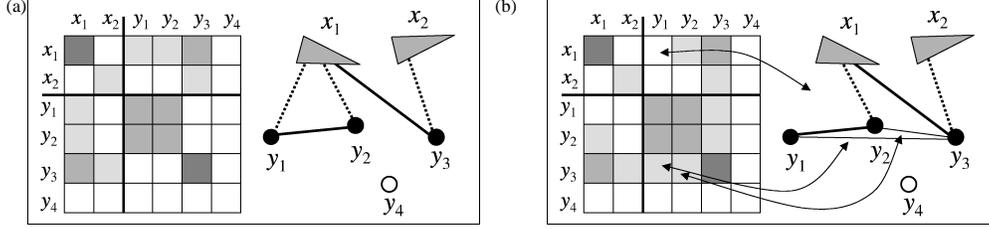


Figure 3: Sparsification: A landmark is deactivated by eliminating its link to the robot. To compensate for this change in information state, links between active landmarks and/or the robot are also updated.

$$-S_{x^k, Y^0} (S_{x^k, Y^0}^T H_t S_{x^k, Y^0})^{-1} S_{x^k, Y^0}^T H_t' \quad (8)$$

This update is also additive and local. It leaves the values of all poses of robots other than x_k and all landmarks Y untouched.

We note that the original SEIF algorithm also utilizes an amortized Gauss-Seidel algorithm for recovering the mode μ_t from the information state [17]. This mode is needed at various places of the SEIF update, e.g., for the linearization of the measurement and motion equations.

4 Multi-Robot SLAM

The two key properties that makes SEIF amenable to decentralized, multi-robot mapping are additivity and locality. *Additivity* enables multiple robots to integrate their information by adding increments; addition is commutative and associative, hence the resulting update schemes can cope with network latencies (see [9] for a related paper that exploits additivity in multi-robot SLAM). *Locality* ensures that all updates performed by a robot are confined to its own pose and landmarks previously detected by this robot (which is *not* the case for the much more popular EKF!). As a result, in multi-robot SEIF, each robot can maintain its own local map and posterior estimate.

However, the fusion of local maps is tricky because of two difficulties: first, each robot maintains its own local coordinate system. As a result, transformation of one robot’s coordinate system into another are *non-linear*, which is at odds with the linear SEIF representation (in particular, it is invalid to simply add the local maps!). Second, a complex data association problem has to be solved when fusing maps, namely that of establishing correspondence between landmarks in the robots’ local maps. Prior work [9] side-steps this issue by assuming knowledge of (1) the initial poses and (2) unique landmark signature; under these highly restrictive assumptions, addition is a viable way to fuse maps. Neither of these assumptions are made here.

4.1 Fusing Maps Acquired by Multiple Robots

Let $\langle H_t^k, b_t^k \rangle$ and $\langle H_t^j, b_t^j \rangle$ two local estimates (maps and vehicle poses) acquired by two different vehicles, k and j . To fuse these maps, we need two pieces of information: a relative coordinate transformation between these two maps (translation and rotation), and a correspondence list, that is, a list of pairs of landmarks that correspond to each other in the different maps.

Suppose we are given the translation d and the rotation matrix r that specify the coordinate transformation from the j -th to the k -th robot’s coordinate system—we will discuss our approach to finding d and r further below. Coordinates y in the j -th robot’s coordinate system are mapped into the k -th coordinate system via the linear equation $y^{k \leftarrow j} = r y + d$. This transformation is easily extended to the filter variables $\langle H_t^j, b_t^j \rangle$

$$H_t^{k \leftarrow j} = R^T H_t^j R \quad b_t^{k \leftarrow j} = (b_t^j + H_t^j D^T) R^T \quad (9)$$

where R and D are matrices that extend r and d to the full dimension of the posterior maintained by the j -th robot:

$$R = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & r & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} \alpha \\ d \\ \vdots \\ d \end{pmatrix} \quad (10)$$

Notice the special provision for the robot's heading direction, which is the very first element in the state vector. The heading simply changes by the angle of the rotation between both maps, denoted α in (10).

To see the correctness of (9), we recall that the parameters $\langle H_t^j, b_t^j \rangle$ define a Gaussian over the j -th robot pose and map $\mathbf{x}_t^j = (x_t^j, Y)^T$. This gives us the following derivation:

$$\begin{aligned} p(\mathbf{x}_t^j | Z^j, U^j) &\propto \exp -\frac{1}{2} (R \mathbf{x}_t^j - D - \mu_t^j)^T \Sigma_t^{j,-1} (R \mathbf{x}_t^j - D - \mu_t^j) \\ &\propto \exp -\frac{1}{2} \mathbf{x}_t^{j,T} R^T \Sigma_t^{j,-1} R \mathbf{x}_t^j - (\mu_t^j + D)^T \Sigma_t^{j,-1} R^T \mathbf{x}_t^j \\ &= \exp -\frac{1}{2} \mathbf{x}_t^{j,T} R^T H_t^j R \mathbf{x}_t^j - (b_t^j + H_t^j D^T) R^T \mathbf{x}_t^j \end{aligned} \quad (11)$$

The key observations here are that the alignment takes time linear in the state vector (and not cubic as would be the case for EKFs). More importantly, the sparseness is preserved by this update step. The reader may also notice that the transformation can be applied to subsets of features (e.g., a local map), thanks to the sparseness of H_t^j . In such a case, one has to include the Markov blanket of the variables of interest.

After the alignment step, both maps are expressed in the same coordinate system. The joint information state is obtained by concatenating both information matrices and both information states. The correspondence list is then incorporated into this joint map by *collapsing* the corresponding rows and columns of the resulting information matrix and vector. The following example illustrates the operation of collapsing feature 2 and 4 in the filter, which would occur when our correspondence list states that landmark 2 and 4 are identical:

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} \rightarrow \begin{pmatrix} h_{11} & & h_{12}+h_{14} & h_{13} \\ h_{21}+h_{41} & h_{22}+h_{42}+h_{24}+h_{44} & h_{23}+h_{43} & \\ h_{31} & h_{32}+h_{34} & h_{33} & \\ & & & b_3 \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2+b_4 \\ b_3 \end{pmatrix}$$

Collapsing the information state exploits the additivity of the information state. The viability of a data fusion hypothesis is finally assessed by computing the *likelihood* of the data after fusing two maps. This is achieved by plugging the fused map into the original two Gaussians defining each vehicles' local maps, and by multiply the resulting probabilities. This calculation plays an important role in our algorithm's decision to accept or reject a fusion hypothesis. Technically, this operation involves a recovery of the state, and the evaluation of two Gaussians (one per robot). The mathematics for doing so are straightforward and omitted for brevity.

4.2 Finding Good Alignments

The previous section provided a method for evaluating the goodness of a map fusion candidate, but left open how such candidates are found. Finding good candidates for fusing maps is essentially a hybrid search problem, involving continuous (alignment) and discrete (correspondence) variables.

Our approach performs this search in two states. First, it searches for corresponding pairs of local landmark configurations in different maps. In particular, our approach identifies for each landmark in each map all triplets of three adjacent landmarks that fall within a small radius (a similar grouping was used in [15]). The relative distances and angles in these triplets are then memorized in an SR-tree, to facilitate efficient retrieval. Using these SR-trees, similar local configurations are easily identified in different maps by searching the tree. Correspondences found in this search serve as a starting hypotheses for the fusion process; they are also used to

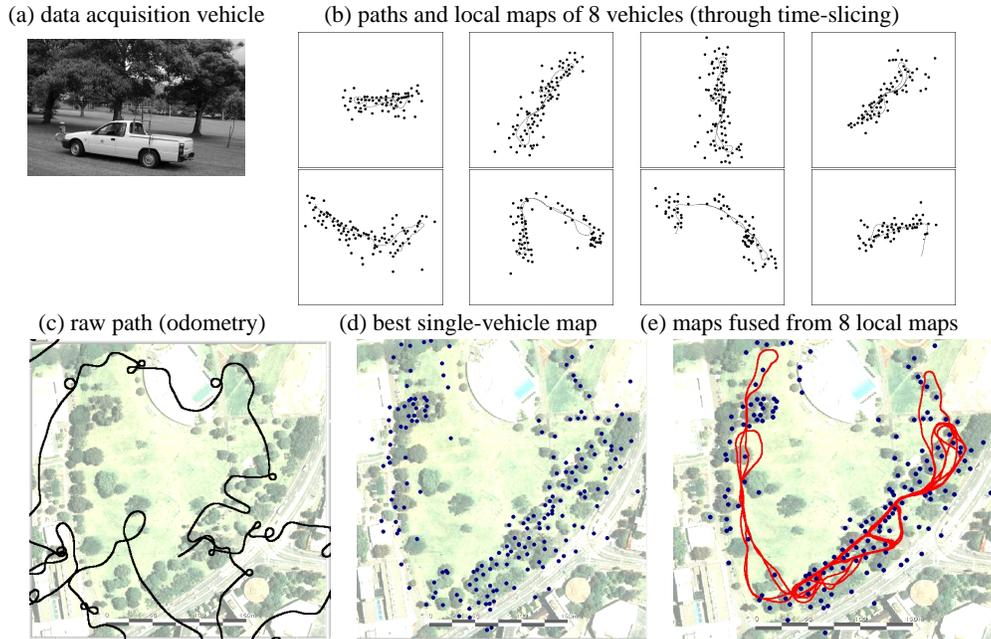


Figure 4: (a) The vehicle; (b) 8 local maps obtained by splitting the data into 8 disjoint sequences; (c) the raw odometry; (d) the path and map recovered by single-vehicle SLAM; (e) the multi-robot result, obtained using the algorithm described in this paper.

recover the alignment parameters, the rotation r and the shift d (using the obvious geometric laws).

When an appropriate candidate has been identified, the space of possible data associations is searched recursively, by assuming and un-assuming correspondences between landmarks in the different maps. The search is guided by two opposing principles: The reduction of the overall map likelihood that comes from equating two landmarks, and the increase in likelihood that results from the fact that if there were really two separate landmarks, both robots should have detected both of them (and not just one). To calculate the latter, we employ a sensor model that characterizes “negative” information (not seeing a landmark).

In general, the search for the optimal correspondence is NP-hard. However, in all our experiments with real-world data we found hill climbing to be successful in every single instance. Hill climbing is extremely efficient; we suspect it to be in $O(N \log N)$ for maps of size N . In essence, it associates nearby landmarks if, as a result, the overall likelihood increases. Once the search has terminated, a fusion is finally accepted if the resulting reduction of the overall likelihood (in logarithmic form) is offset by the number of collapsed landmarks times a constant; this effectively implements a Bayesian MAP estimator with an exponential prior over the number of landmarks in the world.

5 Experimental Results

Systematic experiments were conducted using a real-world data set. This data [3], which is generally recognized as a primary benchmark data set in the SLAM community [6], was acquired by a single vehicle operating in a park in Victoria Park in Sydney. Figure 4a shows the vehicle, along with its raw odometry (Figure 4c), illustrating that odometry alone is extremely erroneous. Features in this map correspond to tree stems extracted from the vehicle’s laser range finder. SEIF, when applied to the single-vehicle SLAM problem, generates the path shown in Figure 4d. This map matches other published results in accuracy [3, 7].

To perform a multi-robot experiment, we simply cut the data into $K = 8$ disjoint sequences, each emulating the data collected by a different robot. Figure 4b shows the resulting local

maps for each robot. Our new multi-robot SLAM algorithm results in the map shown in Figure 4e. The exact error of this map is unknown, because the exact location of the trees in Victoria Park environment are not known. However, it appears that the error of our approach is approximately twice that of the single vehicle approach, in the worst case. This does not surprise, as the data is cut into very small chunks, making it more difficult to recover a joint map. The important finding here is that our approach indeed is able to identify the correct alignments between the different local maps, which is a challenging problem given the small size of the local maps (Figure 4e). To our knowledge, this is the first time an algorithm succeeded in fusing local maps into a single global map, without relative knowledge of the robots' poses and identifiable landmarks (not even the approach in [14] generates a single map).

6 Conclusion

We have proposed a new Bayesian approach to the multi-robot simultaneous localization and mapping problem, which enables teams of robots to acquire a single environment map. Our approach extends work on single-robot SLAM by techniques for establishing correspondence between maps gathered by multiple robots. Empirically, we find our approach to be highly reliable in its ability to identify overlapping maps and fusing them into a single map. We conjecture that our approach is unprecedented in its ability to build a single map from data acquired with multiple platforms, without initial knowledge of the relative pose of each robot and with landmark ambiguity.

Our approach generalizes the work in [9] to the SLAM problem with unknown relative locations and with non-unique landmarks. It also goes beyond the work in [14] in that maps can change shape in the map fusion process, thanks to a representation of a full posterior over maps (instead of the maximum likelihood map only). However, we believe that fusing ideas in [14] and the present paper will ultimately solve multi-robot mapping problems that are presently beyond the state-of-the-art.

References

- [1] M. Bosse, P. Newman, M. Soika, W. Feiten, J. Leonard, and S. Teller. An atlas framework for scalable mapping. *ICRA-03*.
- [2] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. *ICRA-00*.
- [3] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the SLAM problem. *Transactions of Robotics and Automation*, in press.
- [4] J. Guivant and E. Nebot. Optimization of the SLAM algorithm for real time implementation. *Transactions of Robotics and Automation*, May 2001.
- [5] K. Konolige, J.-S. Gutmann, D. Guzzoni, R. Ficklin, and K. Nicewarner. A mobile robot sense net. *SPIE 3839*, 1999.
- [6] J. Leonard, J.D. Tardós, S. Thrun, and H. Choset, editors. *Notes ICRA workshop (W4)*, 2002.
- [7] 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. *IJCAI-03*.
- [8] K. Murphy. Bayesian map learning in dynamic environments. *NIPS-99*.
- [9] E. Nettleton, H. Durrant-Whyte, P. Gibbens, and A. Goktoğan. Multiple platform localisation and map building. In *Sensor Fusion and Decentralised Control in Robotic Systems III*: 4196, 2000.
- [10] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. *IJCAI-03*.
- [11] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
- [12] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. *AAAI-00*.
- [13] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Int. J. Robotics Research*, 5(4), 1986.
- [14] B. Stewart, J. Ko, D. Fox, and K. Konolige. A hierarchical bayesian approach to mobile robot map structure estimation. *UAI-03*.
- [15] J.D. Tardós, J. Neira, P.M. Newman, and J.J. Leonard. Robust mapping and localization in indoor environments using sonar data. *Int. J. Robotics Research*, 21(4), 2002.
- [16] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. *ICRA-00*.
- [17] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A.Y. Ng. Simultaneous mapping and localization with sparse extended information filters. *Proc. of the Int. Workshop on Algorithmic Foundations of Robotics*, 2002.
- [18] Y. Weiss and W.T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10), 2001.