To appear in the 2006 IEEE International Conference on Robotics & Automation (ICRA 2006)

SLAM With Sparse Sensing

Kristopher R. Beevers   Wesley H. Huang
Rensselaer Polytechnic Institute, Department of Computer Science
110 8th Street, Troy, NY 12180, U.S.A.
{beevek,whuang}@cs.rpi.edu

Abstract— **Most work on the simultaneous localization and mapping (SLAM) problem assumes the frequent availability of dense information about the environment such as that provided by a laser rangefinder. However, for implementing SLAM in consumer-oriented products such as toys or cleaning robots, it is infeasible to use expensive sensing. In this work we examine the SLAM problem for robots with very sparse sensing that provides too little data to extract features of the environment from a single scan. We modify SLAM to group several scans taken as the robot moves into *multiscans*, achieving higher data density in exchange for greater measurement uncertainty due to odometry error. We formulate a full system model for this approach, and then introduce simplifications that enable efficient implementation using a Rao-Blackwellized particle filter. Finally, we describe simple algorithms for feature extraction and data association of line and line segment features from multiscans, and then present experimental results using real data from several environments.**

I. Introduction

Simultaneous localization and mapping (SLAM) is a fundamental capability for autonomous mobile robots that must be able to navigate within their environment. The SLAM problem has received much attention in the robotics community, and a number of approaches have been demonstrated. However, nearly all the prevalent work on SLAM demands the use of costly, high-fidelity sensors such as laser rangefinders.

As consumer-oriented robotic devices become more widely available, the demand for mapping capabilities in these robots also grows. Robust mapping and navigation will enable many robotic applications such as robotic assistants for the elderly and disabled, security monitoring, and even "smart" toys. However, few consumers would pay the cost that a laser rangefinder would entail for such robots. Furthermore, a laser rangefinder imposes power, size, and computation requirements that may be infeasible for consumer applications.

In this work we present an efficient approach to enable SLAM in indoor environments on robots with much more limited sensing. We eventually plan to implement SLAM using an array of infrared rangefinders that costs less than US$40, several orders of magnitude less than a laser rangefinder.

Using such a limited sensing array demands an approach that is somewhat different from the traditional SLAM sequence of moving, sensing, extracting features, finding correspondences, and then updating the map and pose estimates. To illustrate this, Figure 1a depicts the data density of a single scan of a 180° laser rangefinder. A single such scan can be used to extract meaningful features such as lines or corners, or can be matched directly with the robot's map. In contrast, Figure 1b shows the data density of a scan using only five radially-spaced range sensors, the arrangement we use in our experiments. Clearly, it is much more difficult to extract features from such sparse data.

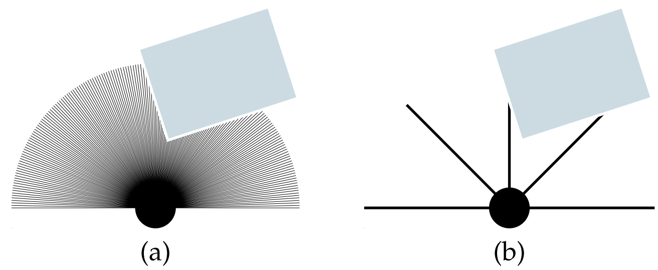


Fig. 1. (a) A typical laser rangefinder scan arrangement, providing 180 degrees of dense coverage in one-degree increments; (b) A sparse sensor, which gives only five range readings in a single scan.

Our approach is to group consecutive sparse scans into *multiscans* so the data from multiple frames can be used to extract good features; the tradeoff is that uncertainty in the robot's motion contributes to noise in feature extraction. This approach requires the pose history to be kept in order to process the multiscan. While others (e.g., [8]) have explicitly stored the pose history in the system state, we instead separate the pose history from the system state so that feature extraction is performed only once per multiscan. In conjunction with a particle filter implementation, our approach yields a reasonably efficient SLAM algorithm for robots with sparse sensing.

The paper proceeds as follows: after a brief discussion of related research, in Section II we discuss a previous approach for applying SLAM using a pose history. We then formulate the multiscan approach, examine its properties, and discuss a particle filtering version in Section III. In Section IV we describe simple methods for extracting line and line segment features from multiscans and finding correspondences. Finally, in Section V, we discuss our implementation of sparse sensing SLAM and the results of experiments on data from the Radish [5] repository.

A. Related work

Most recent SLAM work has focused on creating accurate maps using laser rangefinders either to extract features from the environment [12] or to perform scan matching against the robot's map [4]. The particle filtering approach for performing SLAM has recently gained wide acceptance, and both landmark based and scan-matching based techniques have drawn on the strategy of Rao-Blackwellized particle filtering, first introduced in the SLAM literature by Murphy [9].

Relatively little work has examined the SLAM problem under sensing constraints such as sparseness or sensor range limitations. In the topological mapping domain, Huang and Beevers [7], [6] have dealt with both of these constraints and have developed a complete algorithm for tracing a version of the Voronoi diagram of a rectilinear environment under the L_∞ distance metric, using eight radially spaced short-range sensors. It is not immediately clear how the approach can be extended to more general environments.

In work based on techniques similar to our multiscan approach, Zunino and Christensen [16] and Wijk and Christensen [14] have used SONAR data taken from multiple locations to extract features and perform SLAM. Similarly, Leonard *et al.* [8] and Tardós *et al.* [11] have used a ring of 24 SONAR sensors for mapping with an approach we discuss in Section II. In their experiments with feature extraction based on the Hough transform and an accurate SONAR sensor model, they closed loops in several real-world environments using an Extended Kalman Filter (EKF) based mapper.

II. Applying SLAM with a pose history

When features can only be detected from multiple poses, one approach to SLAM is to incorporate the pose history into the state vector. This is the method used by Leonard *et al.* [8] who applied an EKF to the resulting system. In this section, we give a brief description of this approach and how one might apply a particle filter to this system. This sets the context for our simplified approach to the problem, described in Section III.

Suppose we keep the most recent m robot poses in the system state vector. Then, $x(k) = [x_r(k) \quad x_f(k)]$ where $x_r(k) = [x_{t_k} \ x_{t_{k-1}} \ \ldots \ x_{t_{k-m+1}}]$ (with x_{t_k} being the robot pose at time k), and $x_f(k)$ is a vector of n landmarks $[x_{f_1} \ \ldots \ x_{f_n}]$. This fits into the standard system model:

$$x(k) = f(x(k-1), u(k-1)) + v(k) \quad (1)$$

$$z(k) = h(x(k)) + w(k) \quad (2)$$

where $u(k-1)$ is the input, $z(k)$ the measurement, and $v(k)$ and $w(k)$ are system and measurement noise.

A. Using the EKF

Leonard *et al.* [8] apply the EKF to this system, so they maintain an estimate of the state vector as well as its full covariance:

$$P_x = \begin{bmatrix} P_{x_r} & P_{x_r x_f} \\ P_{x_r x_f}^T & P_{x_f} \end{bmatrix} \quad (3)$$

where P_{x_r}, $P_{x_r x_f}$, and P_{x_f} are covariance matrices for the robot pose history and the landmark locations. The key points in applying the EKF to this system are:

- When the state is projected forward, the new robot pose x_{t_k} is inserted in $x_r(k)$ and the oldest pose is discarded.
- Feature extraction is done using the last m pose estimates and the corresponding m measurements.

Since the full covariance (and cross covariance) of the past m robot poses are maintained, feature extraction can account for the uncertainty of (and correlation between) poses from which the sensor measurements were taken.

However, the computation of the full covariance matrix is very expensive (at least $O((m+n)^2)$ complexity), and it must be performed at each time step. Furthermore, EKF-SLAM is unable to represent multiple data association hypotheses. Recently, particle filtering techniques have been used to overcome these limitations in traditional SLAM problems.

B. Using a particle filter

A reasonably straightforward adaptation of the above model to a standard Rao-Blackwellized particle filter is:

1) For each particle $p^i = \{x^i(k-1), \omega^i(k-1)\}$, $i = 1 \ldots N$:
   a) Project the state forward by drawing a new robot pose $x^i_{t_k}$ from the distribution of $v(k)$ centered at $f(x^i_{t_{k-1}}, u(k-1))$, insert $x^i_{t_k}$ in $x^i_r(k)$ and discard the oldest robot pose.
   b) Extract features from the measurements using the last m poses and perform data association.
   c) Update the map and initialize new features.
   d) Compute a new weight $\omega^i(k)$ equal to the likelihood of the data association.
2) If necessary, resample the particles with probabilities proportional to $\omega^i(k)$.

Note that each particle contains its own pose history, so collectively the particles sample the space of the previous m pose histories. This approach avoids the expensive computation of covariance matrices, but potentially, many particles would be required to adequately sample the pose history space. Furthermore, feature extraction is required for every particle because of their unique pose histories, and this computation is also fairly expensive.

III. SLAM using multiscans

Our approach is based on grouping sparse scans from m consecutive robot poses into a *multiscan* $\mathbf{z}(k) = [z(k) \ z(k-1) \ \ldots \ z(k-m+1)]$. We formulate a system model in which a SLAM update is performed only after each m steps, reducing the required computation. A further simplification enables a particle filter implementation where features are extracted only once per multiscan (instead of once per particle per multiscan).

A. System model

As before, the state vector $x(k) = [x_r(k)\ x_f(k)]$, but now $x_r(k)$ contains only the single pose x_{t_k}. Our system model is:

$$x(k) = F(x(k-m), \mathbf{u}(k-m)) \quad (4)$$

$$\mathcal{Z}(k) = g(\mathbf{z}(k), \mathbf{x}(k)) \quad (5)$$

where $\mathbf{u}(k-m)$ is the vector of inputs $[u(k-1) \ldots u(k-m)]$.

The system function F is defined as:

$$F(x(k-m), \mathbf{u}(k-m)) = [x_{t_k}\ \ x_f(k-m)]^T \quad (6)$$

where the pose x_{t_k} is modeled recursively as:

$$x_{t_k} = f(x_{t_{k-1}}, u(k-1)) + v(k) \quad (7)$$

The pose $x_{t_{k-m}}$ is taken from the state vector $x(k-m)$ after the previous SLAM update.

The function g computes a feature vector $\mathcal{Z}(k)$ containing the parameters of features extracted from the multiscan $\mathbf{z}(k)$, which is acquired from the intermediate robot poses $\mathbf{x}(k) = [x_{t_k} \ldots x_{t_{k-m+1}}]$. Each scan $z(k)$ in a multiscan is modeled by:

$$z(k) = h(x_{t_k}, x_f) + w(k) \quad (8)$$

where $w(k)$ is the measurement noise and x_f is the feature vector from the most recent SLAM update.

B. Approach

We apply a Rao-Blackwellized particle filter to this system, assuming the landmarks are independent when conditioned on the robot's trajectory:

1) For m time steps: move and collect sparse scans.
2) Extract features by considering the data from all m scans simultaneously as a single *multiscan*.
3) For each particle $p^i = \{x^i(k{-}m), \omega^i(k{-}m)\}$, $i = 1 \ldots N$:
   a) Project the pose forward for each motion by drawing samples for intermediate poses:
      - For $i = (k-m+1)$ to k,
        - Draw $v \sim V(i)$
        - Let $x_{t_i} \leftarrow f(x_{t_{i-1}}, u(i-1)) + v$
   b) Find correspondences between extracted features and the landmark locations $x_f^i(k-m)$.
   c) Update map and initialize new features.
   d) Compute a new weight $\omega^i(k)$ equal to the likelihood of the data association.
4) If necessary, resample the particles with probabilities proportional to $\omega^i(k)$.

C. Feature extraction

Because our features are extracted from sensor readings taken from different poses, both the measurement noise and the odometry error contribute to uncertainty in the extracted features. In our particle filter implementation, however, we do not maintain a pose history of the intermediate states for each particle. Instead, we use the expected intermediate pose history $\bar{\mathbf{x}}(k) = [\bar{x}_{t_k} \ \ldots\ \bar{x}_{t_{k-m+1}}]$, calculated from the odometry as:

$$\bar{x}_{t_k} = f(\bar{x}_{t_{k-1}}, u(k-1)) \quad (9)$$

Without loss of generality, we assume $\bar{x}_{t_{k-m}}$ is the origin.

We perform feature extraction using this expected pose history, and then transform the features for each particle p^i so that the pose $\bar{x}_{t_k}$ coincides with $x_{t_k}^i$ to find correspondences. While this approach does not precisely account for the correlation between robot movement and feature parameters and increases feature extraction uncertainty, it avoids performing feature extraction on an intermediate pose history for every particle.

D. Innovation covariance

We consider $\mathcal{Z}$ to be the measurement for SLAM updates, and thus the innovation for a measurement lies in the feature parameter space. The innovation covariance S is required for both maximum likelihood data association and to update landmark locations. To simplify our explanation, assume that all features in $\mathcal{Z}$ are associated with landmarks whose parameters are in $x_\ell(k) = Mx(k)$, where M is a matrix that simply selects the appropriate landmarks from $x(k)$. The innovation is then $\nu = \mathcal{Z}(k) - x_\ell(k)$, and its covariance is:

$$S = J_g P_{(\mathbf{z},\mathbf{x})} J_g^T + M P_{x(k)} M^T \quad (10)$$

The covariance of landmark parameters from $P_{x(k)}$ is readily available, but the covariance of the multiscan and pose history is more complicated:

$$P_{(\mathbf{z},\mathbf{x})} = \begin{bmatrix} P_{\mathbf{z}} & P_{\mathbf{zx}} \\ P_{\mathbf{zx}}^T & P_{\mathbf{x}} \end{bmatrix} \quad (11)$$

Feature extraction can be performed using the full $P_{(\mathbf{z},\mathbf{x})}$. A further approximation that yields acceptable results is to represent $P_{(\mathbf{z},\mathbf{x})}$ using only the block diagonal portions of $P_{\mathbf{z}}$ and $P_{\mathbf{x}}$ (i.e, assuming that measurements are independent and that although pose uncertainty compounds, multiscan poses are independent.)

For complicated feature extraction methods, J_g, the Jacobian of the feature extraction with respect to the multiscan, is difficult to compute analytically. However, it is well-known that maximum likelihood estimation gives good estimates for the covariance of the parameters being estimated, even for an approximately specified model such the block diagonal version of $P_{(\mathbf{z},\mathbf{x})}$ [13]. Thus, by using maximum likelihood estimation as a feature extractor, we can obtain a good approximation to the measurement covariance S without computing J_g. We use this approach to extract lines from multiscan data with a procedure described in Section IV.

E. Co-dependence on odometry

Notice that there is an issue of independence in the multiscan formulation. The robot's odometry is used twice, first to update the pose estimate, and second to

Algorithm 1 EXTRACT-LINES($\mathbf{x}, \mathbf{z}, P_{(\mathbf{z},\mathbf{x})}$)

Compute Cartesian points from $\mathbf{x}$ and $\mathbf{z}$ and covariance of points from $P_{(\mathbf{z},\mathbf{x})}$
Cluster points using a threshold based on the range of each reading
for each cluster C **do**
  Compute $\text{argmax}_{p_i, p_j \in C} ||p_i - p_j||$
  Perform IEPF using p_i, p_j as temporary segment endpoints to split C into subclusters
end for
for each cluster C with enough supporting points **do**
  $\ell = [r\ \theta] \leftarrow$ Maximum Likelihood line parameters for points in C
  $P_\ell \leftarrow$ covariance of ℓ returned by ML estimator
  Add line ℓ with covariance P_ℓ to L
end for
return L

extract features. The multiscan approach makes an implicit assumption that the co-dependence of the robot's pose and the measurements can be ignored. For this to be true, m, the number of readings per multiscan, must be kept small enough that the odometry error experienced over the course of the multiscan is small. One strategy is to make m a function of the pose uncertainty.

IV. FEATURE EXTRACTION AND DATA ASSOCIATION

We have implemented sparse sensing SLAM with two types of features: lines parameterized by distance r and angle θ to the origin; and line segments, which explicitly include extent information in the representation. Line features have been used for SLAM on a number of occasions, e.g. [15], and methods for performing line extraction and data association with lines are readily available [10]. Fewer attempts have been made to perform line segment SLAM. Brunskill and Roy [2] recently described the use of probabilistic PCA to extract line segment parameters and compute covariance information for use in data association. Since our focus is on SLAM with very little data and minimal computation, we have employed simpler techniques.

A. Line features

Before line parameters can be estimated, the data from a multiscan must be segmented. Since our data are sparse and from different poses, the simplest segmentation methods cannot be employed because they assume ordered measurements from a single pose. We instead take all the Cartesian points from the multiscan and apply an agglomerative clustering algorithm with a threshold distance computed adaptively based on measurement range. This is similar to the technique used in an adaptive breakpoint detector [1] for a radial range scan. The resulting clusters are further split using an iterative endpoint filter (IEPF) [3]. Maximum likelihood line parameters are then estimated for each cluster using weights based on the covariance of each point. Lines with too few supporting points are discarded. Algorithm 1 shows the pseudocode for the line extraction procedure.

Maximum likelihood data association and map updates for line features are straightforward since the parameters of lines can be directly compared and merged.

B. Line segment features

We have also implemented SLAM with line segment features. An advantage of segment features is that they explicitly encode extent information which can be useful in data association. In order to extract segments from multiscan data, we employ the same procedure as for line features, and then project the clustered points onto their associated lines. The two extremal projected points p_i and p_j are used as the segment parameters, and the parameter covariance is computed based on the covariances of the line parameters and of the projected endpoints, i.e. $P_s = J_\ell P_\ell J_\ell^T + J_{p_i} P_{p_i} J_{p_i}^T + J_{p_j} P_{p_j} J_{p_j}^T$

A complication of line segment features is that it is impossible to directly compare the parameters of two segments for data association since they may actually represent different portions of the same feature. A simple solution is to "extend" the segments so that their endpoints can be directly compared. We extend two segments so that their projections onto the angular bisector are the same. Care must be taken to update the endpoint covariances accordingly. Unfortunately the extension procedure represents a complicated function with convoluted Jacobians, so updating the covariance is hard. A simple approximation is to assume that the lengths each endpoint is extended, d_0 and d_1, are known parameters of a function that transforms a single segment, i.e., $E(s, d_0, d_1)$, which has a simpler Jacobian $J_s = \partial E / \partial s$ that can be used to transform the segment covariance. In practice, this simplification works reasonably well and is much easier to compute.

V. RESULTS

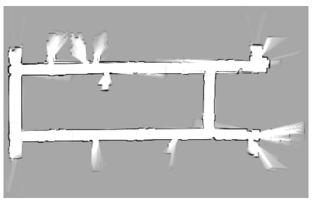
We have implemented our sparse sensing SLAM approach and tested it on a variety of datasets. Our Rao-Blackwellized particle filter is based mainly on Fast-SLAM 1.0 [12]; we also use the adaptive sampling approach described by Grisetti, Stachniss, and Burgard [4].

Most aspects of our implementation have been discussed in previous sections. In our feature extraction, a simple weighted least squares estimator was used instead of a full maximum likelihood approach, for efficiency. Also, the experiments presented here with line segment features estimated covariance using only the two segment endpoints rather than the full data.

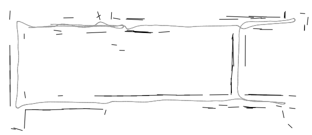
Our experiments used data from Radish [5], an online repository of SLAM datasets. Most of the available datasets use scanning laser rangefinders with 1° spacing. In order to test SLAM with sparse sensing, we simply

TABLE I
EXPERIMENT STATISTICS

	USC SAL	CMU NSH	Stanford
Dimensions	39m × 20m	25m × 25m	64m × 56m
Particles	100	600	1000
Sensing range	5 m	3 m	5 m
Path length	122 m	114 m	517 m
Path rotation	450 rad	133 rad	495 rad
Scans per multiscan	50	40	18
Total multiscans	89	118	1151
Avg. MS translation	1.37 m	0.97 m	0.45 m
Avg. MS rotation	3.80 rad	1.12 rad	0.43 rad
Feature type	Lines	Segments	Segments
Num. landmarks	88	168	750



(a)



(b)

Fig. 2. The USC SAL Building, second floor. Dataset courtesy of Andrew Howard. (a) Occupancy data for the corrected map. (b) The corrected line landmark map (black) and trajectory (gray). The landmark map shows only the original observed extent of the line features.

discarded most of the data in each scan, using only the five range measurements at $0°, 45°, 90°, 135°$, and $180°$. Additionally, we restricted the maximum sensor range to at most 5 m, and in some cases less.

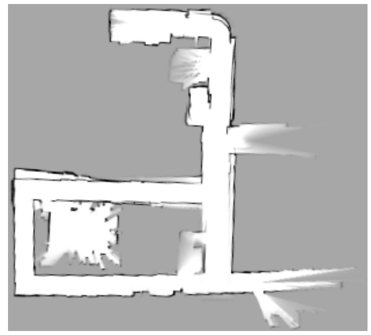
A. Experiments

Figures 2, 3, and 4 show the results of sparse sensing SLAM on several datasets. (The occupancy grid images were generated using corrected trajectories from sparse sensing SLAM, but with the original dense laser data for better clarity.) Detailed statistics for these datasets are shown in Table I.

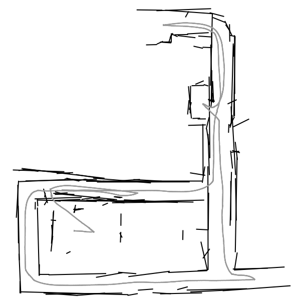
1) USC SAL Building: (Figure 2) This dataset consists of a primary loop and several small excursions. The robot closed the loop properly in its line-based landmark map despite only a small overlap between the loop's start and end. Notice that some slight aliasing exists in the occupancy grid map: this was a common occurrence since most features have high uncertainty due to the use of odometry to augment the sparse sensing.

2) CMU Newell-Simon Hall: (Figure 3) Because of the maximum sensing range of 3 m for this experiment, the fairly large initial loop (bottom) could not be closed until after the robot finished exploring the upper hallway.

3) Stanford Gates Building: (Figure 4) This long dataset has three large and several small loops. Figure 4a shows the uncorrected data, which exhibits significant error, particularly with respect to the robot's orientation. At several points the robot spins in place, a difficult situation for sparse sensing because rotation dramatically increases the pose uncertainty and decreases the scan density due to high rotational velocities.



(a)



(b)

Fig. 3. A partial map of Newell-Simon Hall Level A at CMU. Dataset courtesy of Nicholas Roy. (a) Occupancy data for the corrected map. (b) The corrected line segment map (black) and trajectory (gray).

Note that some landmarks are spurious, a result of poor data association due to large uncertainty. No detection of spurious landmarks was implemented, so these landmarks remained in the map. Again, although there is some aliasing in the results, the environment's structure and the robot's trajectory were properly recovered.

B. Discussion

These results show that performing SLAM in large indoor environments is feasible even with minimal sensing. All of our tests used only five sensors with restricted range, but even large loops were closed correctly. However, there are tradeoffs to using such limited sensing:

- More particles are required since the parameters of landmarks are more uncertain due to the use of odometry to augment sensing.
- The success of SLAM is sensitive to the amount of pose uncertainty accumulated during a multiscan.
- The size of multiscans (m) is a parameter that must be determined, either by selecting a constant or computing a value based on pose uncertainty. Choosing m is a complex problem given all of the factors — error models, environment complexity, robot behaviors, etc. — that affect SLAM.
- Computationally-motivated approximations (such as those made in extracting features and computing the innovation covariance) can lead to poor data association, as exhibited by the existence of spurious landmarks.

VI. CONCLUSIONS

In this paper, we have presented a strategy for implementing particle filtering SLAM on a robot with very sparse sensing. Rather than performing feature extraction on every frame of scan data, we group the data into "multiscans" consisting of several consecutive frames of data. Feature extraction, data association, and state updates are then performed based on multiscan data as if it were data from a single scan. We formally specified a system model for this approach that maintains multiscan information separately from the system state, allowing

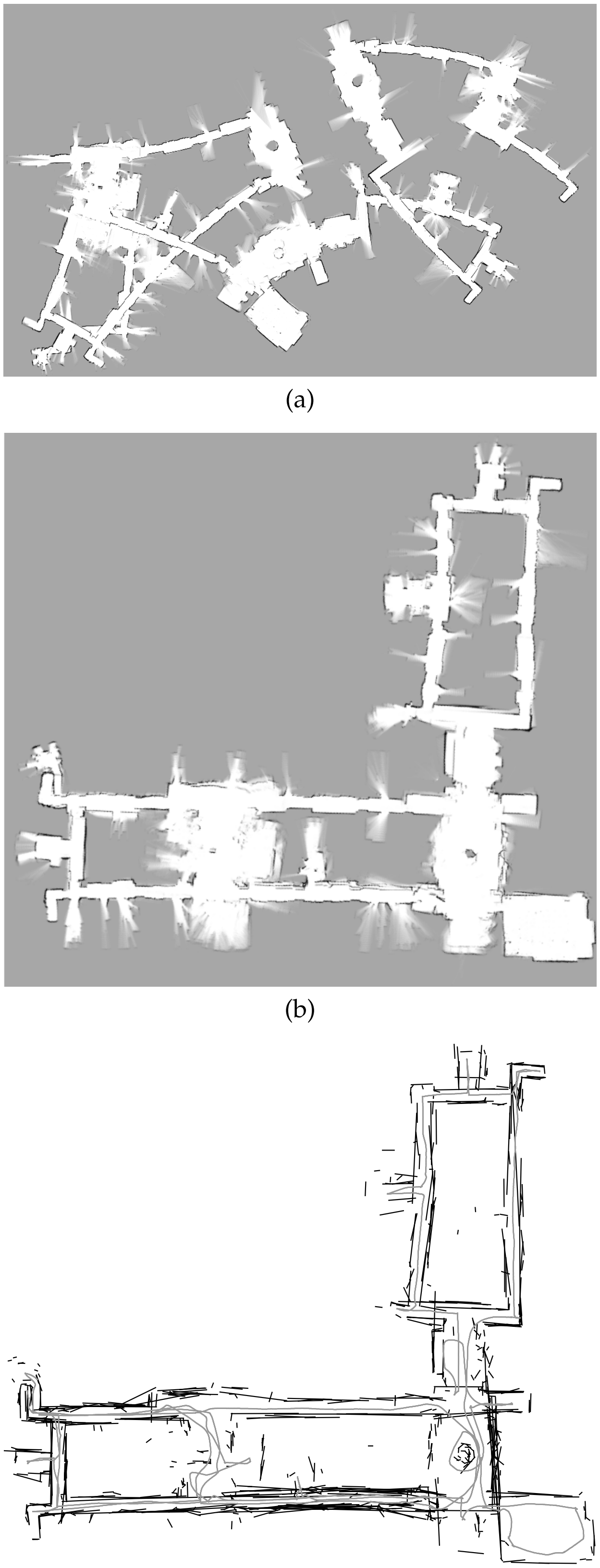


Fig. 4. The Gates Building at Stanford, first floor. Dataset courtesy of Brian Gerkey. (a) Raw uncorrected data (not to scale with corrected data). (b) Occupancy data for the corrected map. (c) The corrected line segment landmark map (black) and trajectory (gray).

the efficient application of particle filtering methods for SLAM. We then discussed properties of the innovation covariance for features extracted from multiscans, and presented a simplified representation for which the computation costs are significantly reduced. Finally, we described simple approaches for extracting line and line segment features from multiscan data and performing data association.

In our experiments using measurements from only five one-dimensional range sensors, the robot was able to close loops and recover the correct map and trajectory in several large real environments. While the experiments uncovered several tradeoffs to using limited sensing, the success of our approach with real data shows that it is possible to implement SLAM with sparse sensors.

VII. Acknowledgments

This work was supported in part by the NSF through award IIS–9983642.

References

[1] G. Borges and M.-J. Aldon. Line extraction in 2D range images for mobile robotics. *Journal of Intelligent and Robotic Systems*, 40:267–297, 2004.

[2] E. Brunskill and N. Roy. SLAM using incremental probabilistic PCA and dimensionality reduction. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 344–349, 2005.

[3] R. Duda and P. Hart. *Pattern classification and scene analysis*. Wiley, New York, USA, 1973.

[4] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 2443–2448, 2005.

[5] A. Howard and N. Roy. The Robotics Data Set Repository (Radish), 2003.

[6] W. H. Huang and K. R. Beevers. Complete topological mapping with sparse sensing. Technical Report 05-06, Rensselaer Polytechnic Institute, Troy, NY, March 2005.

[7] W. H. Huang and K. R. Beevers. Topological mapping with sensing-limited robots. In M. Erdmann et al., editors, *Algorithmic Foundations of Robotics VI*, pages 235–250. Springer, 2005.

[8] J. Leonard, R. Rikoski, P. Newman, and M. Bosse. Mapping partially observable features from multiple uncertain vantage points. *Intl. Journal of Robotics Research*, 21(10):943–975, October 2002.

[9] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems*, volume 12, pages 1015–1021. MIT Press, 2000.

[10] S. Pfister, S. Roumeliotis, and J. Burdick. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003.

[11] J. Tardós, J. Neira, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using SONAR data. *Intl. Journal of Robotics Research*, 21(4):311–330, April 2002.

[12] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 2004. to appear.

[13] H. White. Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–26, January 1982.

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

[15] D. Yuen and B. MacDonald. Line-base SMC SLAM method in environments with polygonal obstacles. In *Proc. of the Australasian Conf. on Robotics and Automation*, 2003.

[16] G. Zunino and H. Christensen. Navigation in realistic environments. In M. Devy, editor, *9th Intl. Symp. on Intelligent Robotic Systems*, Toulouse, France, 2001.