An Iterative Approach for Building Feature Maps in Cyclic Environments

Haris Baltzakis and Panos Trahanias

Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), P.O.Box 1385, Heraklion, 711 10 Crete, Greece and Department of Computer Science, University of Crete, P.O.Box 1470, Heraklion, 714 09 Crete, Greece. Email:{xmpalt,trahania}@ics.forth.gr

Abstract

In this paper we deal with one of the fundamental problems for mobile robots, namely their ability to produce accurate representations of their environment. We propose an office mapping algorithm that iteratively alternates between a Kalman smoother based localization step and a map features recalculation step. Moreover, a hybrid algorithm with global localization capabilities is employed in the first step, enabling correct identification of already mapped areas, and, thus, ensuring map correctness in cyclic environments.

1 Introduction

In order to navigate safely and reliably, an autonomous mobile robot must be able to find its position within its environment. For this purpose, creation and maintenance of suitable representations of the environment is necessary.

In robotics literature the tasks associated with these goals are referred as the localization and the mapping task, respectively. These tasks are very closely related in the sense that dealing with either of them requires a solution to the other. In other words, to localize the robot within its environment, a model (map) of the environment should be available, while for utilizing robot perceptual information in order to form a map, the location of the robot within some kind of a coordinate system should be known.

Assuming that a model of the environment is available, various approaches for dealing with the localization task exist, roughly falling into two major categories depending on how the belief of the robot about its location (state) is represented: Discrete state approaches [1, 2] and continuous (Gaussian) state approaches [3, 4]. Each of these two categories exhibit different advantages and disadvantages. More specifically, experiments [5] have shown that discrete state approaches are more robust in the presence of noise and/or unreliable odometry information while continuous state approaches are more efficient with respect to computational efficiency, scalability, and accuracy. Recently, a couple of hybrid approaches have been proposed combining advantages from both families [6, 7, 8, 9].

Map representations utilized by these two classes of algorithms also fall in two categories. Grid occupancy maps and feature maps; the former most commonly utilized by discrete state localization algorithms, and the latter by continuous state algorithms. Algorithms for automatically obtaining such maps are by far more challenging than corresponding localization algorithms since they involve computation of the robot's state as well as the map simultaneously. For discrete approaches, EM-based methods, that iteratively alternate between a localization step and a map parameter learning step have been successfully used in order to produce grid occupancy maps [10, 11]. For continuous approaches, the robot's state vector is usually augmented with the map feature parameters and filtering methods (such us the very commonly used Extended Kalman Filter) are used to simultaneously determine both the state of the robot as well as the map features. The latter approach, often referred as stochastic mapping, in its original form (due to [12]) is optimal in the sense that it is capable of storing and manipulating full covariance information both for the state as well as for the features. However, it exhibits very severe computational and storing disadvantages since the size of the computational information that needs to be updated at each iteration grows with the square of the number of map features.

In this paper we propose a mapping algorithm that utilizes features (namely line segments and corner points) in order to produce maps suitable for continuous state localization algorithms. A main contribution of this paper is the employment of an EM-like procedure for this purpose, that avoids the explicit storage of feature-to-feature covariance information without hazarding performance. A further contribution lies in the utilization of a hybrid localization algorithm [6] that enables correct identification by the robot of already visited areas, which ensures correct mapping of cyclic environments (topological correctness of created maps). Experimental results have shown the applicability of the algorithm for modelling complex indoor environments.

2 Feature Extraction

Line segments and corner points extracted out of laser range measurements constitute the feature set currently utilized by the proposed methodology.

For line segment extraction, a three-stage algorithm has been implemented. Range measurements are initially grouped to clusters of connected points according to their Sphere-of-Influence graph [13]. Clusters are then further-segmented to points belonging to the same line segment by utilization of the well-known Iterative-End-Point-Fit (IEPF) algorithm [14]. IEPF recursively splits a set of points until a distance related criterion is satisfied.

After range points have been clustered into groups of collinear points, a recursive Kalman-filter-based method is used in order to "best-fit" lines to them, minimizing the squared radial distance error from the position of the range measuring device to each point. The exact process is as follows.

Line $l = N((l_f, l_d)^T, \Sigma_l)$, with l_f, l_d implicitly defined through the line equation $x \cos l_f + y \sin l_f = l_d$, is initially computed by its extreme points p_1 and p_n. Recursively, each other point p_t, $(1 < t < n)$, is used in order to update the line estimate l.

Corner points are computed at the intersection points of directly adjacent line segments. A corner point is defined as $c \sim N((c_x, c_y)^T, \Sigma_c)$ where $(c_x, c_y)^T = I(f_1, d_1, f_2, d_2)$ is the intersection of lines $l_1 \sim N((f_1, d_1)^T, \Sigma_{l_1})$ and $l_2 \sim N((f_2, d_2)^T, \Sigma_{l_2})$, and Σ_c is the related covariance matrix.

3 Localization

Robot's state at time t is modelled as a Gaussian distribution $x_t \sim N(\mu_{x_t}, \Sigma_{x_t})$, where $\mu_{x_t} = (x_t, y_t, \theta_t)^T$ is the mean value of robot's position and orientation, and Σ_{x_t}, the associated 3x3 covariance matrix.

3.1 Kalman filtering

Line segments extracted by the procedure described above are matched to an a-priori known set of line segments (map). An EKF is then used, that employs sequentially each matched pair, in order to track the robots state over time.

The transition model of the Kalman filter, that is, the function used to project state estimates forward in time (prediction step) is given according to

$$\mu_{x^-_{t+1}} = Exp(F(\mu_{x_t}, \alpha_t)) \tag{1}$$

$$\Sigma_{x^-_{t+1}} = \bigtriangledown F_x \Sigma_{x_t} \bigtriangledown F_x^T + \bigtriangledown F_\alpha \Sigma_{\alpha_t} \bigtriangledown F_\alpha^T \tag{2}$$

where Exp is the expectation operator, F the transition function and $\bigtriangledown F_x$ and $\bigtriangledown F_\alpha$ denote the Jacobians of F with respect to μ_{x_t} and the robot action at time t, α_t.

Using this predicted state, a known line segment l is predicted as

$$l^-_{t+1} = H(\mu_{x^-_{t+1}}) \tag{3}$$

where $H(x)$ is the function that converts a map line segment into robot's frame coordinates. The difference between the predicted line segment l^-_{t+1} and the measured line segment l_{t+1} is the measurement residual (Kalman Innovation) and can be written as

$$r_{t+1} = l_{t+1} - l^-_{t+1} \tag{4}$$

$$\Sigma_{r_{t+1}} = \bigtriangledown F_{x^-_{t+1}} \Sigma_{x^-_{t+1}} \bigtriangledown F^T_{x^-_{t+1}} + \Sigma_{l_{t+1}} \tag{5}$$

where Σ_{l_t+1} is the measured line segment covariance and $\bigtriangledown F_{x^-_{t+1}}$ is the Jacobian of F, obtained by linearizing about the state prediction x^-_{t+1}.

The Kalman gain is computed as

$$K_{t+1} = \Sigma_{x^-_{t+1}} \bigtriangledown F_{x^-_{t+1}} \Sigma^{-1}_{r^-_{t+1}} \tag{6}$$

and, finally, the update to the state prediction is:

$$\mu_{x_{t+1}} = \mu_{x^-_{t+1}} + K_{t+1} r_{t+1} \tag{7}$$

$$\Sigma_{x_{t+1}} = \Sigma_{x^-_{t+1}} - K^T_{t+1} \Sigma_{r_{t+1}} K^T_{t+1} \tag{8}$$

The success of any Kalman filtering method for localization tasks relies heavily on the correct matching of features. In this paper, for matching features we utilize the method described in [6] which is based on a dynamic programming string-search algorithm. The algorithm exploits information contained in the spatial ordering of the features, while, its dynamic programming implementation, furnishes it with computational efficiency.

3.2 Kalman Smoothing

As we have seen in the previous section, the extended Kalman filter provides a means of estimating the robot state at time instant $t \leq T$ given all observations up to time t. For off-line mapping, however, observations up to time T are available. The problem of estimating variables given both past and future observations is denoted as "smoothing". A very popular method for performing smoothing is the Rough-Tung-Striebel smoother [15]. The algorithm consists of two steps. The first step (forward step) is the Extended Kalman Filter forward recursions, as described in the previous paragraph. For each time t estimates for the mean μ_{x_t} and covariance Σ_{x_t} are computed using equations (1)-(8). The second step is a backward recursion starting at time $t = T$ and recursively estimating maximum a-posteriori estimates for $\mu_{x_{t|T}}$ and $\Sigma_{x_{t|T}}$ as:

$$\Sigma_{x_{t|T}} = \Sigma_{x_t} + S[\Sigma_{x_{t+1|T}} - \Sigma_{x^-_{t+1}}] \quad (9)$$

$$\mu_{x_{t|T}} = \mu_{x_t} + S[\mu_{x_{t+1|T}} - \Sigma_{x^-_{t+1}}]S^T \quad (10)$$

where

$$S = \Sigma_{x_t} \bigtriangledown {F_x}^T \Sigma^{-1}_{x^-_{t+1}} \quad (11)$$

4 Mapping

For estimating environmental features and appending them on the map or refining already mapped feature estimates, according to robot's measurements. the robot's state, at the time the measurements were taken, must be known. Hence the problem of mapping expands to the problem of simultaneously estimating both the robot's pose and the map features. Unfortunately, at any time instant t only a small subset of map features is visible and thus directly related to the robot's measurements. Reduction of the problem to individually updating only visible features leads to incorrect results because it does not take into account probabilistic relations among the visible map features and the ones that are not visible.

The theoretically correct treatment implies that all map features and the robot state have to be treated as a single complex random variable [12]. According to this approach, mapped features are augmented to the state vector as soon as they get detected. Both the augmented state vector and the corresponding covariance matrix that reflects state-to-feature and feature-to-feature covariance information have to be updated after each measurement. However, as the number of features grows, the algorithm becomes computationally intractable since the storage and computational needs grow with the square of the number of features. Many complicated schemes have been proposed to reduce this computational burden.

Instead of augmenting all map features to the state vector, an alternative solution is to treat map features as parameters of the dynamical system according to which the robot's state evolves. The problem can be reformulated as to simultaneously determine the state and the parameters of a dynamical system; that is, a learning problem. Variations of the EM algorithm [16, 11, 2] are used for this purpose.

The EM algorithm consists of two steps. The E step which is a state estimation step and the M step that is a parameter estimation step. During the E step, the algorithm relies on the parameters that is has already computed during the previous iterations and tries to estimate the current state as though the parameters were correct. During the M state the algorithm uses the computed state and tries to recompute the parameters in order to maximize the overall probability of the states given the observations and the parameters.

5 The Iterative Algorithm

In this paper we propose a method that resembles the EM algorithm in the sense that it consists of two different steps, that state estimation step (E step) and the map parameter computation step (M step). The E step is the localization step while the M step if the map features computation step.

A block diagram of the proposed algorithm is depicted in fig.5. During the E-step, the algorithm localizes the robot using all the available measurements. To achieve this, Kalman and the Rough-Rung-Striebel equations described in section 3 are utilized in order to provide maximum a-posteriori estimates of the robot states. During the M-step, the algorithm recalculates the mapped features. The procedure is iterated until convergence is achieved (no significant changes are made to the map features) or a maximum number of iterations is reached.

Figure 2 demonstrates the operation of the proposed iterative algorithm in a simple artificial environment. As the robot moves through unexplored environment mapping features (fig.2a and fig.2b), both position and map errors accumulate. As soon as the robot recognizes an already visited area (fig.2b) the EM algorithm iteratively corrects both position and map estimates. The result is depicted in fig.2c. In all three figures the current belief of the robot about its state is displayed as a 95% isoprobability ellipse, magnified, for displaying purposes, by a factor of 10. Modes of past position estimates are displayed in fig.2c as dots, while modes of corrected position estimates are depicted as crosses.

The EM algorithm presented above can be invoked either at the end or at selected intermediate points of a mapping session. In the next section a novel method is presented for automatically detecting loops in the map and initiate the EM algorithm accordingly.

5.1 The Hybrid Model and Closing Loop Detection

As the robot moves and maps features in an unknown environment, errors in both the state end the mapped features tend to increase with time. However, when an already mapped area is revisited, the robot should be able to correct its state and eliminate the accumulated error. The Kalman Smoother redistributes the error among the previous state estimates and the iterative algorithm described previously corrects the map.

If, though, the accumulated error at the end of a long path through unmapped areas is larger than

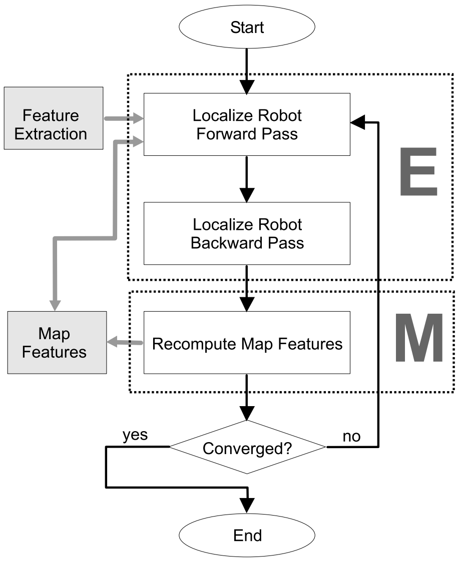


Figure 1: *Flogramm of the Iterative Mapping Algorithm.*

what the feature matching algorithm can handle, the robot will fail to recognize that the measured features already exist in the map and will try to reinsert them. This results in a topologically incorrect map, i.e. a map were the same features appear with multiple entries.

Various methods have been proposed to handle such cases and to identify when the robot enters a place it has already visited. [17] manually identifies "interesting places" as the robot enters them, while the "global correlation, local registration" method [18] continuously tries to identify already visited areas by correlating measurements with the map.

In this paper we propose a method that takes advantage of the global localization capabilities of the hybrid algorithm presented in [6] in order to identify when the robot revisits an already mapped place. Based on a switching state-space model, the hybrid localization algorithm assumes multiple Kalman trackers assigned to multiple hypotheses about the robot's state while letting discrete Markovian dynamics handle the probabilistic relations among these hypotheses.

Hypotheses are dynamically generated by matching corner points extracted from robot's measurements as described in section 2 with corner points that already exist in the map. Hypotheses that are not verified by observation sequences, eventually become less probable and finally disappear.

The mapping algorithm starts with only one hypothesis that is responsible to perform the mapping function as described in the previous section (the dominant hypothesis). Whenever a corner point appears

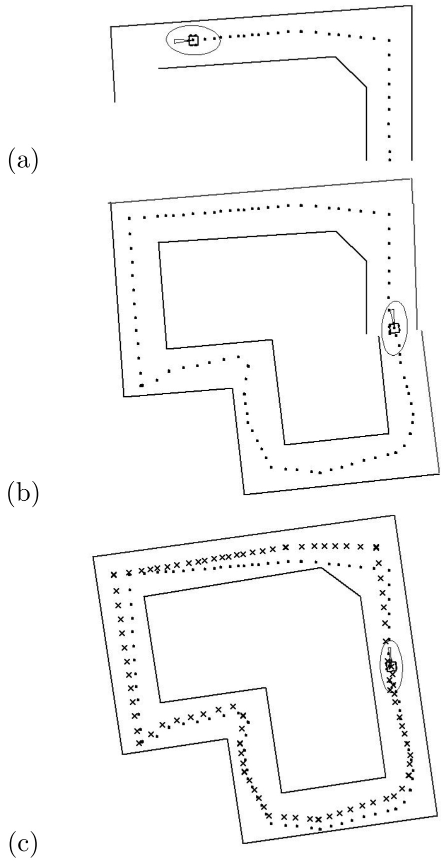


Figure 2: *Operation of the proposed Iterative Mapping Algorithm in a simulated environment*

in the robot's measurements, new hypotheses will be created at positions corresponding to similar corner points in the map but they will eventually vanish if observation sequences do not confirm their validity.

Suppose that a robot enters a previously mapped area and fails to recognize it due to large localization errors introduced by a long journey through completely unmapped areas. As soon as the robot sensors measure a corner point that is already mapped somewhere in the map, a hypothesis will be generated at the correct position of the robot. Eventually this newly created hypothesis will gain probability against the dominant hypothesis, since observations confirm its validity, while penalizing the validity of the dominant hypothesis. As soon as its probability becomes larger that the probability of the dominant hypothesis, it replaces the later, and the iterative algorithm described at the previous section corrects the map and the state estimates.

6 Experimental Results

The probabilistic framework proposed in this paper has been assessed using a variety of test data acquired by a robotic platform of our laboratory, namely an iRobot-B21r, equipped with a SICK-PLS laser range finder. Extensive tests have also been performed with simulated data for various environments and varying odometry and range measuring resolution and accuracy.

Figure 3 demonstrates the operation of the algorithm in a simulated, corridor-like environment. In fig.3a, the robot, after mapping a large and complicated corridor structure, reaches an already mapped area. Hypotheses are continuously created and pruned according to the observations. The dominant hypothesis (the one that actually performs the mapping function) is marked with a tick. Failing to recognize the observed features as already mapped features, the algorithm incorrectly reinserts them on the map. However, as soon as an already mapped corner point is observed, new hypotheses are generated (fig.3b) . Among the newly created hypotheses, the one that corresponds to the correct position of the robot, eventually gains probability. By the time it reaches its position in fig.3c, the correct hypothesis replaces the dominant hypothesis and the EM algorithm is initiated in order to correct the map. Figure 3d shows the corrected map and the position estimates. Modes of corrected position estimates are displayed, in fig.3d, as crosses as opposed to initial position estimates which are displayed as dots.

Figure 4 demonstrates the operation of the algorithm in a complex real environment of approximately 2500 square meters. The robot, after travelling through a long rectangular corridor structure, fails to match observed features with already mapped features (fig.4a, fig.4b). However, very soon the dominant hypothesis is superseded by the hypothesis corresponding to the correct robot position (fig.4c) and the iterative algorithm successfully restores the topology of the map (fig.4d).

7 Conclusions

In this paper we proposed an offline feature mapping methodology based on kalman smoother and an EM-like iterative algorithm. Moreover, exploiting the global localization capabilities of a hybrid localization algorithm, we have furnished the proposed methodology with the ability for loop closing in cyclic environments.

We have demonstrated the capabilities of the proposed methodology with both artificial and real data. In all our experiments the algorithm has always been able to converge to correct maps.

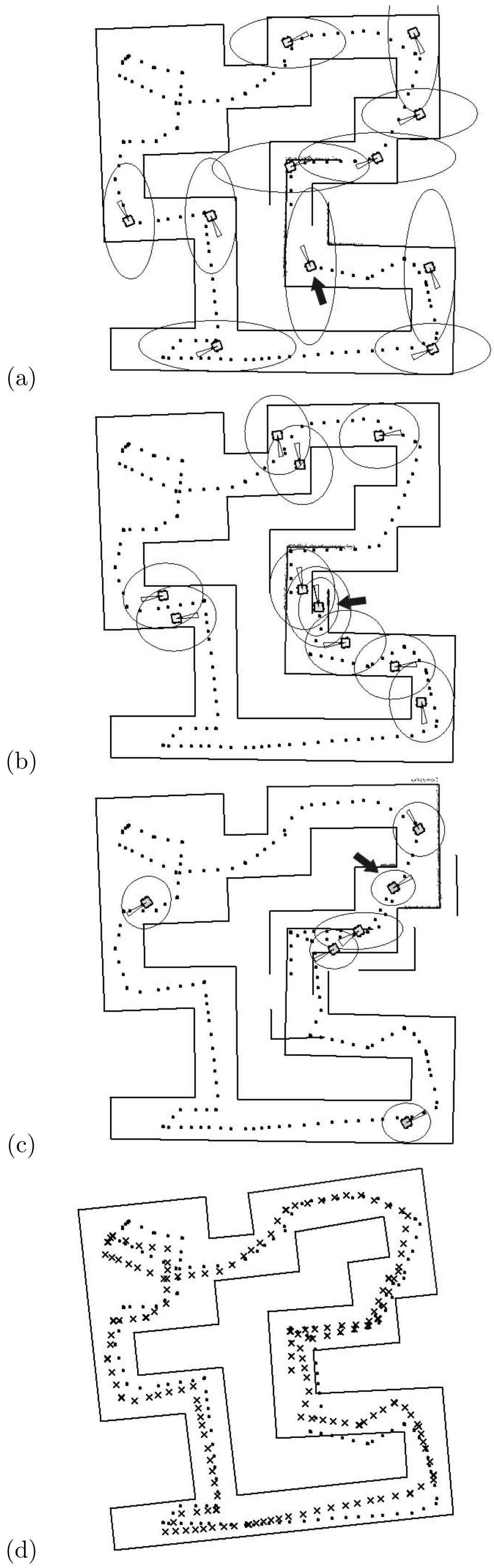
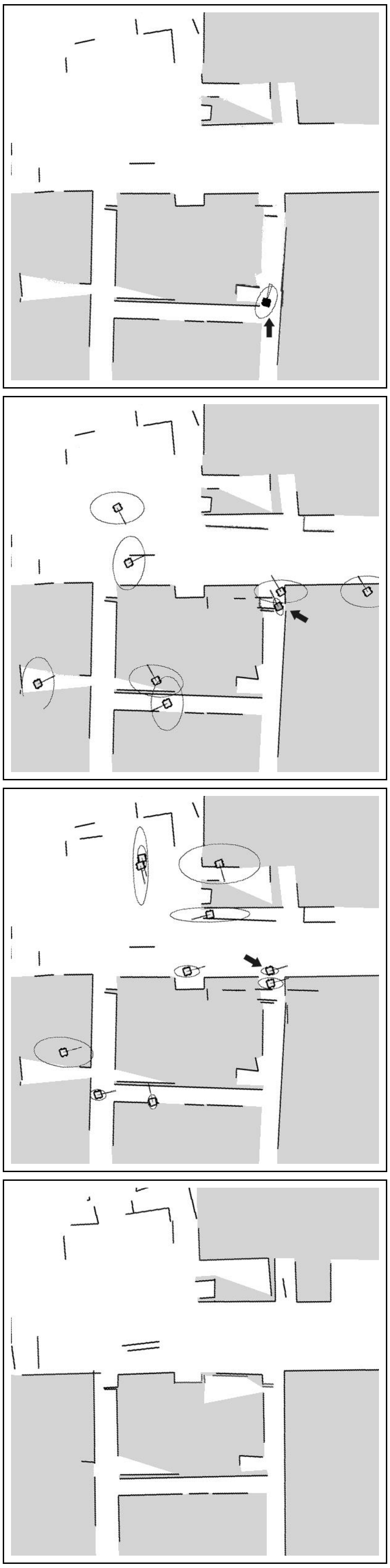


Figure 3: *Operation of the proposed algorithm in an simulated corridor-like environment*

References

[1] A.R. Cassandra, L.P. Kaelbling, and J.A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile robot navigation. In *IEEE/RSJ Intl. Conf. Intell. Robots and Systems*, 1996.



(a)

(b)

(c)

(d)

Figure 4: *Operation of the proposed algorithm in a real environment*

[2] S. Thrun et al. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *Int. J. Robotics Research*, 19(11):972–999, 2000.

[3] J.A. Castellanos, J.M. Martínez, J. Neira, and J.D. Tardós. Simultaneous map building and localization for mobile robots: A multisensor fusion approach. *Intl. Conf. Robotics and Autom.*, pp.1244–1249,1998.

[4] K.O. Arras, N. Tomatis, and R. Siegwart. Multisensor on-the-fly localization using laser and vision. *IEEE/RSJ Intl. Conf. Intell. Robots and Syst.*, 2000.

[5] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. *IEEE/RSJ Intl. Conf. Intell. Robots and Systems*, 1998.

[6] H. Baltzakis and P. Trahanias. Hybrid mobile robot localization using switching state-space models. *Intl. Conf. Robotics and Autom.*, May 2002.

[7] H. Baltzakis and P. Trahanias. A hybrid framework for mobile robot localization: Formulation using switching state-space models. *Autonomous Robots (submited).*

[8] P. Jensfelt and S. Kristensen. Active global localisation for a mobile robot using multiple hypothesis tracking. *IJCAI Workshop on adaptive spatial representations of dynamic environments*, Sweden, 1999.

[9] S.I. Roumeliotis and G.A. Bekey. Bayesian estimation and kalman filtering: A unified framework for mobile robot localization. *Intl. Conf. Robotics and Autom.*, pp.2985–2992, San Francisco, CA, 2000.

[10] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53, 1998.

[11] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping with mobile robots using EM. *Intl. Conf. Machine Learning*, 1999.

[12] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. I. J. Cox and G. T. Wilfong, eds, *Autonomous Robot Vehicles.*, pp.167–193. Springer-Verlag, 1990.

[13] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, 1973.

[14] T.S. Michael and T. Quint. Sphere of influence graphs in general metric spaces. *Mathematical and Computer Modelling*, 29:45–53, 1994.

[15] H. Rauch, F. Tung, and C. Striebel. Maximum likelihood estimates of linear dynamic systems. *Amer. Inst. Aeron. & Astronautics J*, 3(8):1445–50, 1965.

[16] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:185–197, 1977.

[17] D. Fox, W. Burgard, and S. Thrun. Probabilistic methods for mobile robot mapping. *IJCAI Workshop on adaptive spatial representations of dynamic environments*, 1999.

[18] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. *IEEE Intl. Symp. Comput. Intell. in Robotics & Autom.,* CA, 2000.