Closing Multiple Loops while Mapping Features 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 propose an offline feature mapping algorithm capable of identifying and correctly closing multiple loops in cyclic environments (see video). The proposed algorithm iteratively alternates between a Kalman smoother based localization step and a map features recalculation step. The identification of loops is done during the localization step by a hybrid localization algorithm that generates and tracks hypotheses generated each time the robot visits an already mapped area. The main contribution of this paper lies on the ability of the proposed algorithm to exploit information contained within the hypotheses histories in order to calculate correct maps, regardless of the complexity of the environment and the number of loops in the robot's path.**

I. INTRODUCTION

In order to localize itself, a mobile robot needs information about its environment. Creation and maintenance of suitable representations of such information is called mapping and is the purpose of this paper.

Map representations fall in two categories: Grid occupancy maps and feature maps; the former most commonly utilized by discrete state localization [1], [2] algorithms, and the latter by continuous state algorithms [3], [4]. 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 [5], [6]. For continuous approaches, the robot's state vector is usually augmented with the map feature parameters and filtering methods (such as 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 [7]), 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.

As the robot moves and maps features in an unknown environment, errors in both the state and 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. If though, the accumulated error at the end of a long path through unmapped areas is larger than 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 that has already been visited. [8] manually identifies "interesting places" as the robot enters them, while the "global correlation, local registration" method [9] continuously tries to identify already visited areas by correlating measurements with the map.

In our previous work [10] a feature mapping algorithm has been proposed that utilizes the global localization capabilities of a hybrid algorithm [11] in order to identify when the robot revisits an already mapped place. An EM-like iterative algorithm is used to close the loop and to rectify the map: the E step is a Kalman smoother localization step and the M step is a map recalculation step. In this paper we propose an enhancement to this algorithm that makes it capable of simultaneously handling multiple loops. As in [10], the identification of loops is done during the localization step by the hybrid localization algorithm. However, in this paper, a history of all loops identified is kept and each generated hypothesis is tracked separately. The most appropriate path through hypotheses histories is computed just before the Kalman smoother localization step. This permits the algorithm to calculate correct maps, regardless the of complexity of the environment and the number of loops in the robot's path. To the best of our knowledge, this is not achieved by other, contemporary works in the topic.

Experimental results have shown the applicability of the algorithm for modelling complex indoor environments.

II. 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. The feature set utilized by our algorithm is constituted by line segments and corner points. Line segments are extracted by a modified version of the well-known Iterative-End-Point-Fit (IEPF) algorithm [13], while corner points are computed as the intersection points of directly adjacent line segments.

A. 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 robot's 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 $\Sigma_{l_{t+1}}$ 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 [11] 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.

B. 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 [14]. 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:

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

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

where

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

III. 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.

A 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 [2], [6], [15] are used for this purpose.

The method proposed in this paper is based on an algorithm 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. During the E step, the algorithm relies on the parameters that it 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.

A block diagram of the proposed algorithm is depicted in Fig. 1. 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 II-B 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.

Fig. 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 is introduced in order to iteratively correct 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.

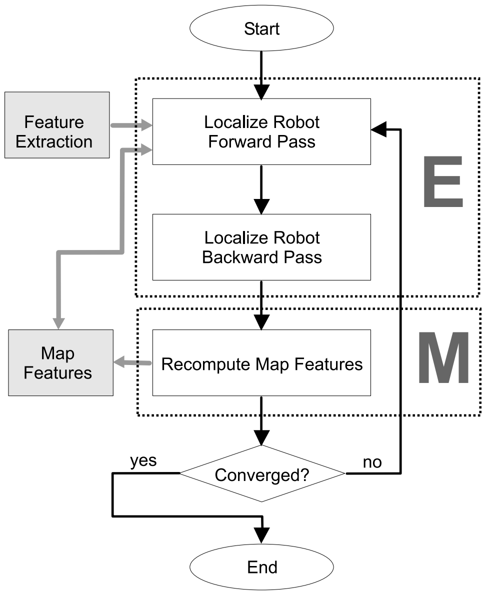

Fig. 1. Flowgram of the Iterative Mapping Algorithm.

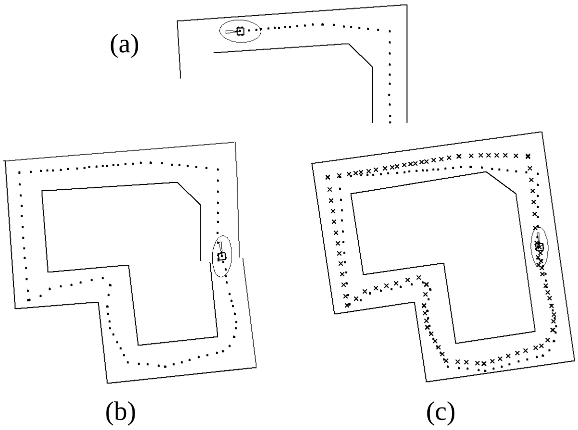

Fig. 2. Operation of the proposed iterative mapping algorithm in a simulated environment.

IV. ALGORITHM FOR MULTIPLE-LOOP CLOSING

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 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. [8] manually identifies “interesting places” as the robot enters them, while the “global correlation, local registration” method [9] 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 [11] 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.

The mapping algorithm starts with only one hypothesis h_1 that is responsible to perform the initial mapping function. Whenever new line segments and corner points are observed they are appended to the map. A time index corresponding to this particular instant of time is stored along with the feature itself for later use by the iterative loop closing and map rectification algorithm. Already mapped line segments are used to perform robot localization. During this initial mapping session, no new hypotheses are generated; mapping is performed with a single tracker, the one corresponding to the initial hypothesis h_1.

After the initial mapping session is finished the iterative EM-like algorithm described in the previous section is initialized in order to close the loops and rectify the map. The iterative algorithm starts with a forward localization step initialized at the starting position of hypothesis h_1, used to perform the initial mapping function described in the previous paragraph. During this step hypotheses are dynamically generated by matching corner points extracted from robot's measurements as described in section II with corner points that already exist in the map. Hypotheses that are not verified by observation sequences, eventually become less probable and finally disappear. At each time instant only map features with time indexes corresponding to past time instants are utilized for localization (that is, features that have been mapped in the future time instants are not used for localization). This prevents the algorithm from utilizing erroneous features (duplicate features created due to incorrect loop closing) for localization.

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, since observations confirm its validity, while penalizing the validity of all other hypotheses.

During the forward localization step, all hypotheses created are tracked individually and the corresponding robot states are stored. After the forward localization step is over, the algorithm tries to find the dominant path through the stored hypotheses histories. This step is equivalent to detecting the loops in the robot path. Hypotheses histories stored during the forward localization process contain all needed information to extract the correct path.

Let the correct state of the robot at time instant t be given by the state of the tracker corresponding to hypothesis $h_c(t)$. The problem is transformed to finding the sequence of hypotheses $\{h_c(t), 1 \leq t \leq T\}$, and the following algorithm is utilized for this purpose.

Step 1. Fix the final state of the robot at the position of the most probable hypothesis h_T at time T (that is, set $h_c(T) = h_T$).

Step 2. Set the current hypothesis h at the position of the initial hypothesis h_1. That is, set $h = h_c(1) = h_1$

Step 3. Track forward the current hypothesis h by setting $h_c(t) = h$ as long as it remains the most probable one. When another hypothesis h' becomes more probable at time instant t' then examine if h' remains more probable than h for at least a threshold time t_{thres} (fixed to about 10 iterations of the localization algorithm). If this criterion is satisfied or if $h' = h_T$ then track backwards h' until the time instant t'' that h' was created by setting $h_c(t) = h'$ for $t'' \leq t \leq t'$. Set the current hypothesis $h = h'$

Step 4. If $t = T$ then exit else repeat step 3.

After the path $h_c(t)$ through localization hypotheses has been determined for all t, $1 \leq t \leq T$ then the robot poses are tracked backwards by running the Kalman smoother described in section II-B.

Using the smoothed robot poses the map is adjusted and the localization step is once more introduced for another iteration of the algorithm until convergence is achieved. During this last step of the algorithm (map recalculation step) no new map features are inserted into the map (only existing features are recomputed).

V. 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.

Figures 3 demonstrates the operation of the algorithm in a simulated environment. Figures 3(a-e) demonstrate the forward mapping process. In Fig. 3b, the robot, after mapping a large portion of the artificial environment, exits the environment area and enters an area where no features exist. The lack of features forces the robot to relay only on odometry for localization and thus leads to large localization errors. As soon as the robot reenters the environment area (Fig. 3c), the accumulated odometry error is too large for the robot to be able to match detected features with already mapped features and correct its position. Hence, the robot localization error is not corrected and all newly detected figures are reinserted in the map as if the robot was still mapping unvisited areas. In Fig. 3d the robot exits the artificial environment and enters the area with no features for one more time. Once again, by the time the robot reenters the mapped area (Fig. 3e), it fails to recognize the observed features as already mapped features and the algorithm incorrectly reinserts

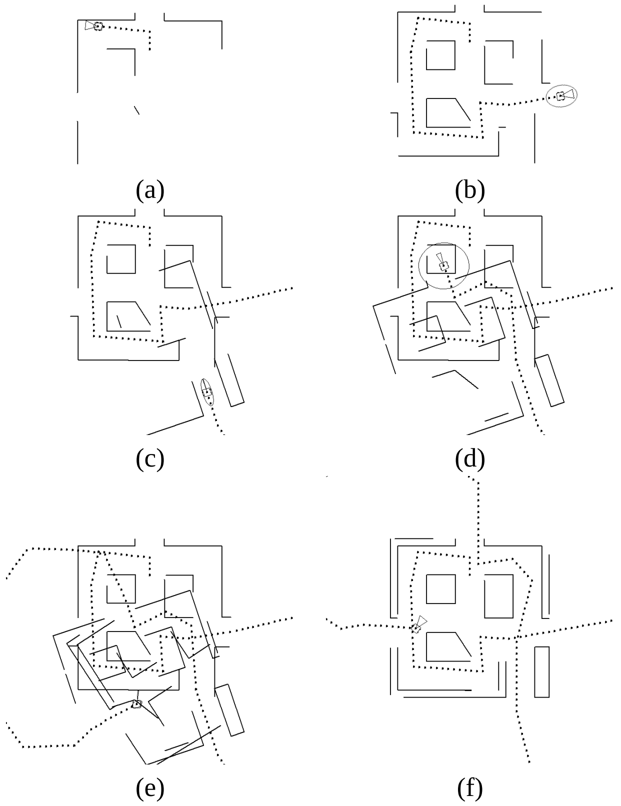

Fig. 3. Map construction.(a-e):Forward mapping, (f):Final rectified map

them on the map. Fig. 3(f) shows the corrected map and robot pose estimates.

Fig. 4 demonstrates the operation of the mapping 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). The final, rectified map computed by the proposed iterative algorithm is shown in Fig. 5 in a side-to-side comparison with the architectural plan of the mapped area (the best available ground truth information). As can be easily observed, the resulting map is both topologically and metrically correct.

VI. CONCLUSIONS

In this paper we proposed an offline feature mapping algorithm capable of identifying and correctly closing multiple loops in cyclic environments. Loop identification is facilitated by a hybrid localization algorithm that tracks hypotheses generated each time the robot visits an already mapped area. Loop closing is done by an EM-like algorithm that iteratively alternates between a Kalman smoother localization step and a map features recalculation step.

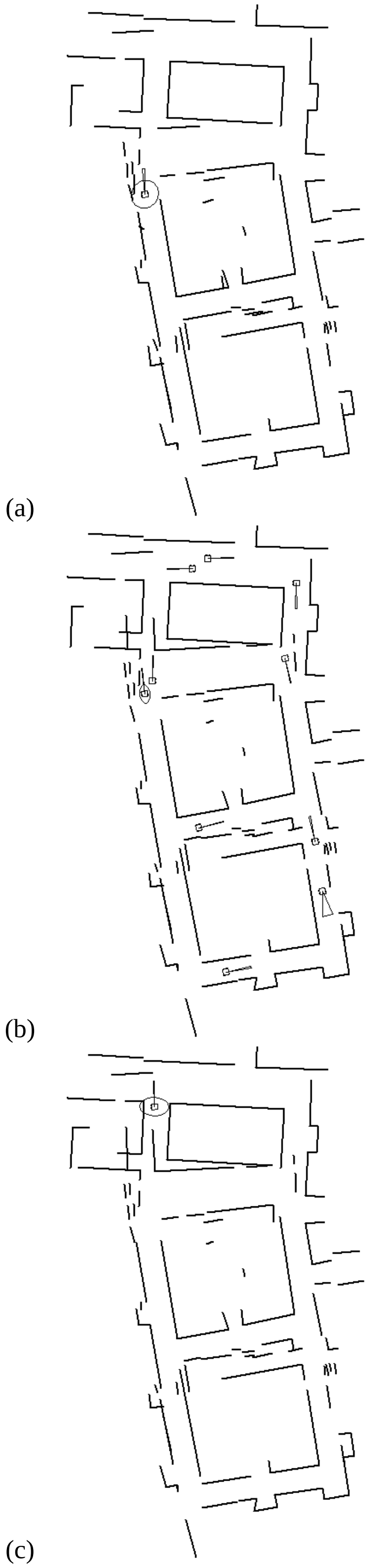

Fig. 4. Operation of the proposed mapping algorithm in a real environment.

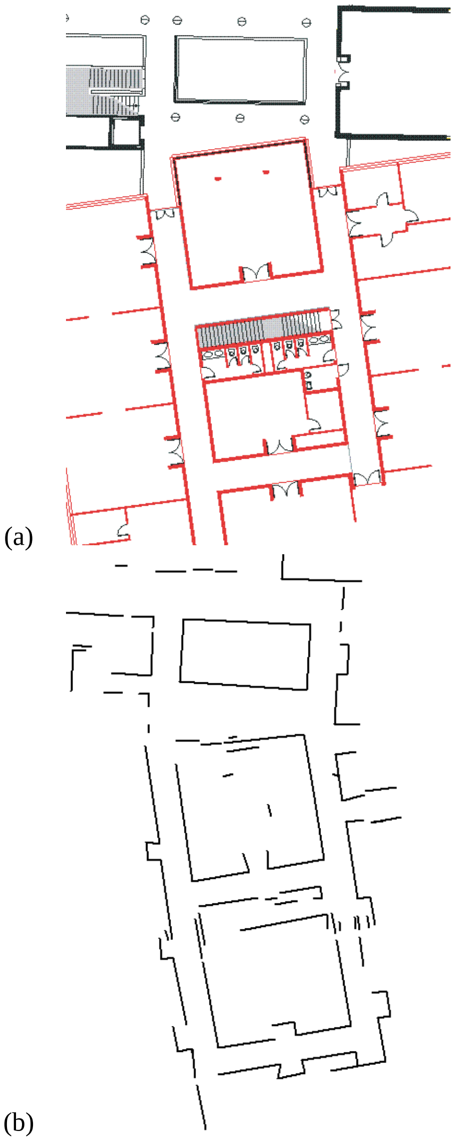

Fig. 5. Comparison of the computed map with ground truth. (a) Architectural plan of the mapped area, (b) Map created by the proposed algorithm.

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.

VII. ACKNOWLEDGMENTS

This work has partly been supported by the by the IST Programme of Commission of the European Communities under contract numbers IST-1999-12643 and IST-2000-29456. The authors furthermore would like to thank the members of the IST-project TOURBOT for helpful comments and fruitful discussions.

VIII. REFERENCES

[1] A.R. Cassandra, L.P. Kaelbling, and J.A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile robot navigation. In *IROS96*, 1996.

[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. *ICRA98*, 1998.

[4] K.O. Arras, N. Tomatis, and R. Siegwart. Multisensor on-the-fly localization using laser and vision. *IROS00*, 2000.

[5] 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.

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

[7] 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.

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

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

[10] H. Baltzakis and P. Trahanias. An Iterative Approach for Building Feature Maps in Cyclic Environments. *IROS02*,2002.

[11] H. Baltzakis and P. Trahanias. Hybrid mobile robot localization using switching state-space models. *ICRA02*, May 2002.

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

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

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

[15] 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.