Using Multi-hypothesis Mapping to Close Loops in Complex Cyclic Environments

Haris Baltzakis*
email:xmpalt@ics.forth.gr

Panos Trahanias*†
email:trahania@ics.forth.gr

* Institute of Computer Science
Foundation for Research and Technology – Hellas (FORTH)
P.O.Box 1385, Heraklion, 711 10 Crete, Greece

† Department of Computer Science
University of Crete
P.O.Box 1470, Heraklion, 714 09 Crete, Greece

***Abstract*— This paper describes an off-line, iterative algorithm for simultaneous localization and mapping within large indoor environments. The proposed approach is based on the idea of separately tracking multiple robot pose hypotheses that are generated each time the robot closes a loop by revisiting an already mapped area. Each tracked pose hypothesis corresponds to a separate possible robot path and maintains a separate map. Loop identification is inherently pursued by a hybrid localization algorithm with global localization capabilities. During the first step of the algorithm, all recorded sensor information is sequentially processed, in order to create a set of possible robot paths and their corresponding maps. After all sensor input is processed, the history of all tracked hypotheses is validated in order to select the most probable robot path and the corresponding map. The second step of the algorithm utilizes an EM-like iterative procedure in order to close the loops identified by the first step, compute a correct robot pose history and produce the final map. Experimental results demonstrate that the proposed algorithm facilitates computation of correct maps regardless of the number of loops in the robot's path.**

I. Introduction

For creating feature maps of the environment, the dominant approach implies that all map features as well as the robot states have to be treated as a single multi-dimensional random variable [1]–[3]. According to this approach, often refereed as stochastic mapping, 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. This approach, although optimal in the sense that all available covariance information is utilized, exhibits very severe computational and storing disadvantages [4].

Recent research has focused on scaling this approach to enable mapping of larger environments (i.e. environments with more than a few hundred features). Some of the most well known approaches try to achieve better performance by dividing the environment into a network of submaps and, thus, restricting the kalman filter update step to a subset only of the total features [5]–[8]. In [8], a method for transferring local updates to the global map is provided as well (however, during global updates, the performance remains quadratic to the total number of features). Sparse extended information filters [4], [9] also provide a solution which achieves constant time by applying an iterative algorithm for relaxation equation solving and restricting it to relaxing only a constant number of features per time step.

Another approach to the mapping problem is proposed in [10], where the FastSLAM algorithm is introduced. According to this approach, particle filters are used to track both the pose of the robot and the feature parameters, at the same time. Each particle represents a hypothesis about both the robot pose and the map by maintaining its own separate set of kalman filters to track the features of the map.

Regardless of the mapping algorithm, in order for the robot to be able to handle the continuous uncertainty in the estimated robot poses and the observed features, it must be able to identify observed features and associate them correctly with mapped ones (discrete uncertainty or data association problem [11]). 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.

To compensate with the above mentioned problems, [12] proposed an off-line feature mapping algorithm that employs the global localization capabilities of a multi-hypothesis localization algorithm [13] in order to identify loops in the robot's path. Whenever a loop is detected, an EM procedure is employed to close the loop, eliminate topological inconsistencies and rectify the map. In [14], this procedure is extended to handle the simultaneous presence of multiple loops by postponing the Kalman smoother and the M-phase of the EM algorithm until all data have been processed. According to this approach, each identified loop initiates a new location hypothesis that is tracked separately within the same map. The most appropriate path through hypotheses histories is computed just before the Kalman smoother localization step.

It is a characteristic of this method that all location hypotheses operate within the same map. However, although helping to reduce memory and computational requirements, this leads to multiple copies of the same features during the early iterations of the algorithm which may slow down the convergence process.

Utilizing the same building blocks as in [12], [14], in this paper we propose a mapping algorithm that simultaneously maintains multiple copies of the map. This is achieved by providing each robot pose hypothesis with a separate copy of the map. As in [14], the proposed algorithm consists of two steps. During the first step, all recorded sensor information is sequentially processed, in order to create a set of possible robot paths and their corresponding maps. Loop identification is inherent to this step which is pursued by the hybrid localization algorithm presented in [13]. This algorithm is capable of generating and tracking new robot pose hypotheses each time the robot closes a loop and revisits an already mapped area. A history of all loops identified is kept and each generated hypothesis is tracked separately. Unlike the approach in [14], each hypothesis is tracked within its own copy of the map. After all sensor input is processed, the history of all tracked hypotheses is validated in order to select the most probable robot path and the corresponding map.

The second step of the algorithm utilizes an EM-like iterative procedure in order to close the loops identified by the first step, compute a correct robot pose history and produce the final map. The proposed algorithm iteratively alternates between a Kalman smoother based localization step (E-step) and a map features recalculation step (M-step). In contrast to contemporary feature mapping algorithms, map features are treated as parameters of the dynamical system according to which the robot's state evolves. This way, the proposed algorithm avoids explicit computation and storage of feature-to-feature and feature-to-state covariance information without hazarding the quality of the produced maps.

Although different in concept, the proposed approach bears similarities with the FastSLAM approach in that neither FastSLAM nor the proposed approach maintain any feature-to-feature covariance information; rather they treat feature measurements as if the robot was certain about its position by the time of each feature measurement. FastSLAM achieves this by having a very large number of robot pose hypotheses (one sample for each possible robot position), each one being certain about its own position and maintaining its own copy of the map. On the other hand, in our approach, map features are computed during the M step of an iterative EM-like algorithm, thus considering robot positions computed during the E-step to be fixed.

Another conceptual similarity lies in the fact that both methods maintain more than one robot pose hypotheses and maps during mapping. However, in FastSLAM each sample represents a single history of exact robot poses (and thus a single possible map). Therefore, the sample set must be large enough so that at least one sample exists that lays close to the correct robot pose at all times. In our approach, hypotheses represent only discrete position uncertainty (usually introduced by loop-closing). Continuous position uncertainty is represented via gaussian distributions around each hypothesis and the requirement for exact assumption of the robot position, needed in order to avoid handling of feature-to-feature and feature-to-pose covariance information, is handled iteratively by EM.

The key idea in this paper lays in letting each robot pose hypothesis generated by the localization algorithm in [14] maintain its own copy of the map. Effectively, this means that "pose hypotheses" become "pose+map hypotheses", allowing the algorithm to eliminate multiple copies of the same features within the same map. The proposed algorithm has been tested extensively with both real and simulated data. All experimental results have shown the applicability of the algorithm for modeling complex indoor environments.

II. Algorithm Overview

In this paper we propose a feature mapping algorithm that consists of two steps. During the first step, all recorded sensor information is sequentially processed, in order to create a set of possible robot paths and their corresponding maps. For this purpose a hybrid localization algorithm [13] is utilized.

At each time step (each laser scan), laser measurements are used in order to extract features. Features are then matched to the existing map in order to localize the robot within it. Features that do not match with the map are used to extend the map.

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 aeady 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. [15] manually identifies "interesting places" as the robot enters them, while the "global correlation, local registration" method [16] continuously tries to identify already visited areas by correlating measurements with the map.

In this paper the identification of loops is inherent to the proposed localization algorithm which takes advantage of the global localization capabilities of the hybrid algorithm [13] 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 hybrid localization algorithm is capable of generating and tracking new robot pose hypotheses each time the robot closes a loop and revisits an already mapped area. Each generated hypothesis gets a copy of the map and is tracked separately. Additionally the hybrid localization assigns and maintains a probability to each of the generated hypotheses depending on whether sensor input continue to verify the generated hypothesis or not.

When all sensor information is processed, the robot's path is determined by tracing back the robot's course among the histories of all robot pose hypotheses. The resulting path and the corresponding map are passed to the next step in order to rectify the map and to close the loops.

The second step of the algorithm utilizes an EM-like iterative procedure in order to close the loops identified by the first step, compute a correct robot pose history and produce the final map. The iterative procedure consists of two different steps, the state estimation step (E step) and the map parameter computation step (M step). The E step is the localization step while the M step is the map features computation step. During the E step, the algorithm relies on the parameters that have already been 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 likelihood of the observations.

The procedure described above is farther analyzed in the following sections.

III. Step A. Forward Multi-Hypothesis Mapping

A hypothesis about the robot's state at time t is modeled 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 the robot's position and orientation, and Σ_{x_t}, the associated 3x3 covariance matrix. The feature set utilized by our algorithm consists of line segments and corner points. Line segments are extracted by a modified version of the well-known Iterative-End-Point-Fit (IEPF) algorithm [17], while corner points are computed as the intersection points of directly adjacent line segments.

At each time step (each laser scan), laser measurements are used in order to extract line segments and corner points according to the procedure described above. Line segments are then matched to the existing map in order to localize the robot. For this purpose an EKF is used that employs sequentially each matched pair of line segments. Features that do not match with the map are used to extend the map.

The proposed algorithm starts with only one hypothesis that is responsible to perform the initial mapping function. Whenever new line segments and corner points are observed they are appended to the map. New hypotheses are dynamically generated by matching corner points extracted from the robot's measurements with already mapped corner points. Hypotheses that are not verified by observation sequences, eventually become less probable and finally disappear. By the time a hypothesis is created, it gets a copy of the map that was used in order to create it.

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 close ("close" is defined later in the next paragraph) in the map(s) of one or more hypotheses, one or more hypothesis will be generated at the corresponding positions of the robot. Eventually one of the newly created hypotheses will gain probability, since observations will continue to confirm its validity, while penalizing the validity of all other hypotheses.

In order to keep the number of hypotheses small, only hypotheses that are close to the true robot position are created for each corner point that is measured. That is, when a corner point is measured, all maps (of existing hypotheses) are searched in order to find matching corner points. Each matching corner point yields a candidate child hypothesis to be created. However only hypotheses that correspond to the robot positions laying not far (within three standard deviations, in our implementation) from a validation pose, computed according to odometry measurements, are finally created. To compute the validation pose, which is different for each candidate hypothesis, we assume a very pessimistic odometry model and we use it to accumulate all odometry measurements starting from the last time that this specific corner point was observed within the history of its ancestor hypotheses.

During the forward localization step, all hypotheses created are tracked individually within their own maps and the corresponding robot states are stored. After the forward localization step is over, the algorithm determines the robot's path by tracing back the histories of all robot pose hypotheses. This step is equivalent to detecting the loops in the robot path and is achieved by starting from the hypothesis that has the larger final probability and by tracing back in time its robot pose history, jumping to the history of its parent when reaching the time instant that the traced hypothesis was created.

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 $h_c(T) = h_T$). Set $t_{end} = T$.

Step 2. Track the current hypothesis $h_c(t_{end})$ backwards until the time instant $t = t_{start}$ that $h_c(t_{end})$ was created by setting $h_c(t) = h_c(t_{end})$ for $t_{start} \leq t \leq t_{end}$.

Step 3. If $t = t_{start} = 0$ then exit

Step 4. Set the current hypothesis $h_c(t)$ for $t = t_{start} - 1$ as the hypothesis whose map was used in order to create the previous hypothesis $h_c(t_{end})$. Set $t_{end} = t_{start} - 1$.

Step 5. Goto Step 2.

The path $h_c(t)$ through localization hypotheses that has been determined via the previous procedure for all t, $1 \leq t \leq T$ as well as the corresponding map (the map of $h_c(T)$) are used to initialize the EM algorithm described in the next section.

IV. Step B. Iterative Map Rectification and Loop Closing

For estimating environmental features and appending them on the map or refining already mapped feature estimates, according to the 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 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 [12] 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.

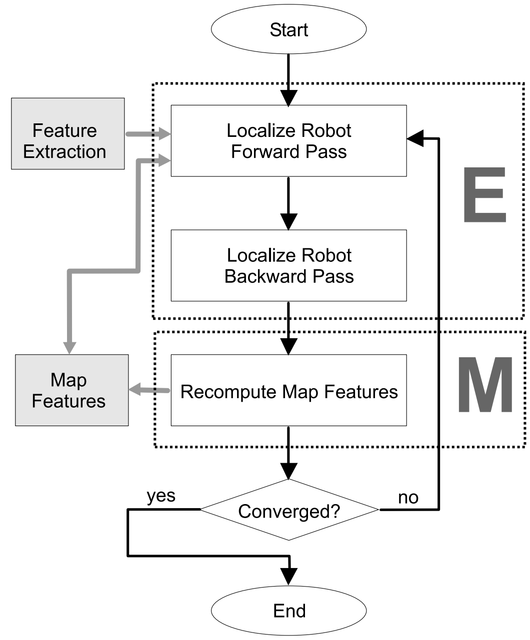

Fig. 1. Flowgram of the Iterative Mapping Algorithm.

V. Experimental Results

The proposed algorithm 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 2, 3 and 4 demonstrate the operation of the proposed mapping algorithm in a simulated environment. Figure 2 demonstrates the forward mapping process (Step A). In Fig. 2c, the robot, after mapping a large portion of the artificial environment, re-enters an already mapped area. By this time, the accumulated error in both its position and the map 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 re-inserted in the map as if the robot was still mapping unvisited areas. However a new pose hypothesis is generated as soon as the robot observes an already mapped corner point (bottom right corner point). At this point, a new copy of the map is created and passed to this hypothesis in order to be tracked separately. By the time the first step of the algorithm finishes (Fig. 2d), six possible robot pose hypotheses exist, each one carrying its own pose history and map.

Figure 3 depicts all the six final hypotheses of Fig. 2d along with their robot pose histories and their map copies. The initial hypothesis is the one depicted in Fig. 3a. The one finally selected by the algorithm to be most probable one is the hypothesis of Fig. 3b (which happens to be the hypothesis generated at Fig. 2c).

The operation of the iterative loop closing algorithm (second step) is depicted in Fig. 4. The initial robot positions, as calculated by the first step of the algorithm, are depicted as dots, while the final, rectified robot positions are depicted as crosses. As can be easily observed the algorithm very quickly succeeds in converging to a correct (both topologically and metrically) map.

A step by step demonstration of the operation of the proposed algorithm is given with a real example in Fig. 5. In Fig. 5a, the robot, after mapping a large cyclic area, revisits a place that it has already visited and a new hypothesis is generated. The robot's path and the corresponding map, as the robot continues mapping, according to this new hypothesis, are depicted in Figures 5b,c,d. The robot after mapping another large area, closes another loop (Fig. 5d). For comparison, Fig. 5e depicts the path of the first (initial) hypothesis at this time instant (and the map that would have been produced if no other hypotheses had been generated). On the contrary, Fig. 5f depicts the robot's path and the map, according to the most probable hypothesis at this time instant which is also the time instant that all sensor input has been processed (finish of first step of the algorithm). As shown in the picture, there are two

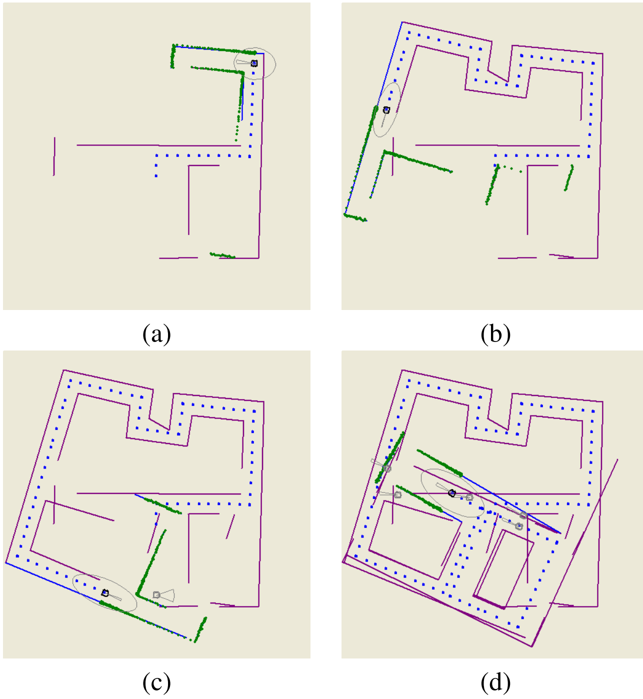

Fig. 2. Forward mapping and hypothesis generation (Step A).

"jumps" within the robot's path. These jumps correspond to an equal number of loops that have been detected. The final, rectified map with the loops closed by the iterative algorithm (step b) is depicted in Fig. 5g.

VI. Conclusions

In this paper we proposed an off-line 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.

The key concept of this paper lays in tracking multiple robot pose hypotheses that are generated by the localization algorithm each time the robot closes a loop and revisits an already mapped area. A history of all identified loops is kept and each generated hypothesis is maintained separately along with a separate copy of the map. The most appropriate path through hypotheses histories is computed and used to initialize an EM-like iterative procedure used to close the loops and rectify the map. This permits the algorithm to calculate correct maps, regardless of the 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 field.

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

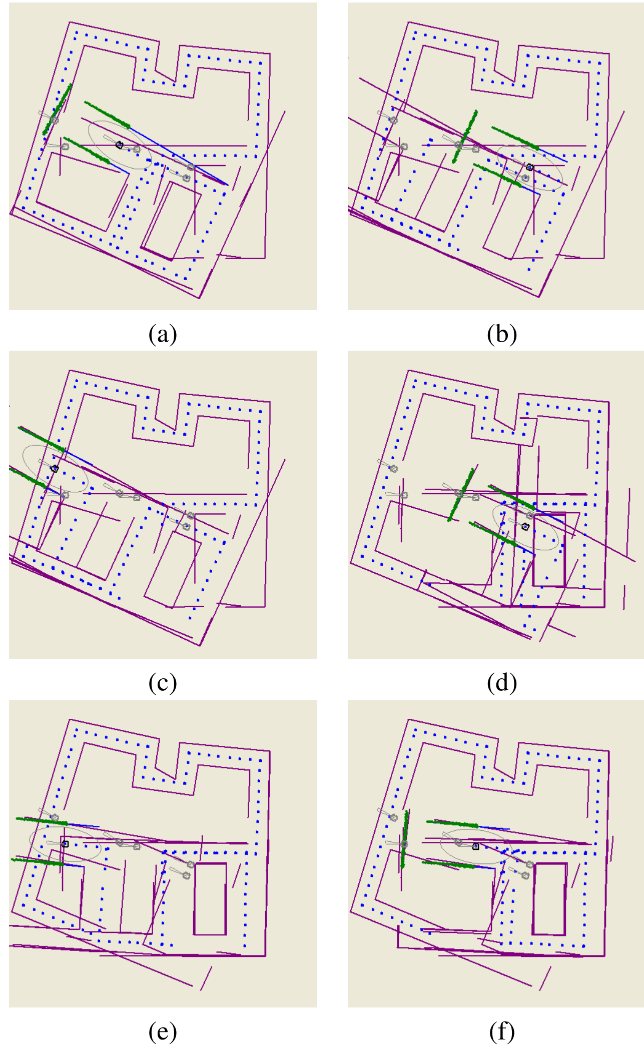

Fig. 3. All six final hypotheses of Fig. 2d, along with their robot pose histories and their map copies.

The methodology presented in this paper can also be applied to the more challenging problem of multi-robot mapping. Our current research efforts focus towards this goal.

References

[1] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proc. International Symposium of Robotics Research*, pages 467–474, Santa Cruz, California, USA, August 1987.

[2] P. Moutarlier and R Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *Proc. International Symposium of Robotics Research*, Tokyo, Japan, August 1989.

[3] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proc. IEEE/RSJ Intlernational Conference on Intelligent Robotics and Systems (IROS)*, pages 1442–1447, Osaka, Japan, November 1991.

[4] U. Frese and G. Hirzinger. Simultaneous localization and mapping - a discussion. In *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics*, Seatle, Washington, USA, August 2001.

[5] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Proc. International Symposium of Robotics Research*, Salt Lake City, Utah, October 1999.

[6] K. Chong and L. Kleeman. Feature-based mapping in real, large scale environments using an ultrasonic array. *International Journal Robotics Research*, 18(1):3–19, January 1999.

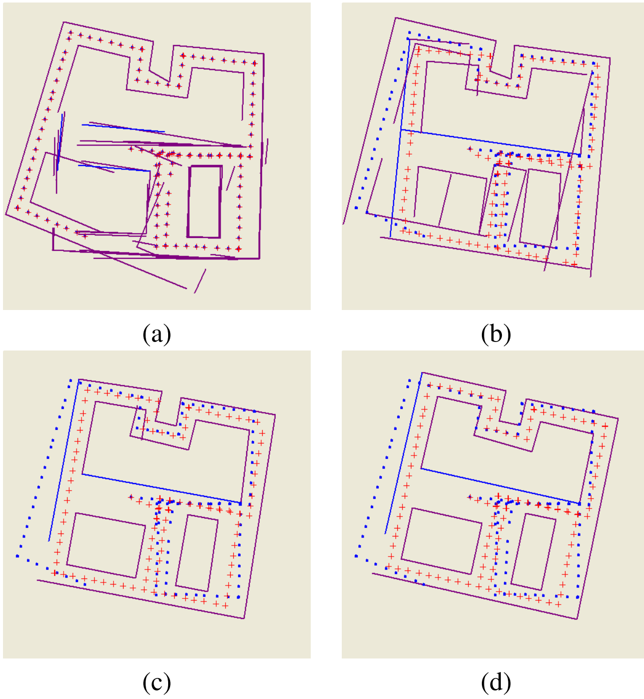

Fig. 4. Iterative loop closing and map rectification procedure: (a) initial map, (b) iteration 4, (c) iteration 10, (d) iteration 17. The initial robot positions, as calculated by the first step of the algorithm, are depicted as dots, while the final, rectified, robot positions are depicted as crosses.

[7] J. J. Leonard and H. J. S. Feder. Decoupled stochastic mapping. *IEEE Journal of Oceanic Engineering*, pages 561–571, October 2001.

[8] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Trans. Robotics and Automation*, 17:242–257, 2001.

[9] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and Ng. A.Y. Simultaneous mapping and localization with sparse extended information filters. In *Proc. International Workshop on the Algorithmic Foundations of Robotics*, Nice, France, December 2002.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. National Conference on Artificial Intelligence (AAAI)*, 2002.

[11] U. Frese and T. Duckett. A multigrid approach for accelerating relaxation-based slam. In *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 39–46, Acapulco, Mexico, August 2003.

[12] H. Baltzakis and P. Trahanias. An iterative approach for building feature maps in cyclic environments. In *Proc. IEEE/RSJ Intlernational Conference on Intelligent Robotics and Systems (IROS)*, pages 576–581, Lausanne, Switzerland, September 2002.

[13] H. Baltzakis and P. Trahanias. Hybrid mobile robot localization using switching state-space models. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 366–373, Washington DC, USA, May 2002.

[14] H. Baltzakis and P. Trahanias. Closing multiple loops while mapping features in cyclic environments. In *Proc. IEEE/RSJ Intlernational Conference on Intelligent Robotics and Systems (IROS)*, pages 717–723, Las Vegas, USA, October 2003.

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

[16] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation, (CIRA)*, pages 318–325, Monterey, CA, November 1999.

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

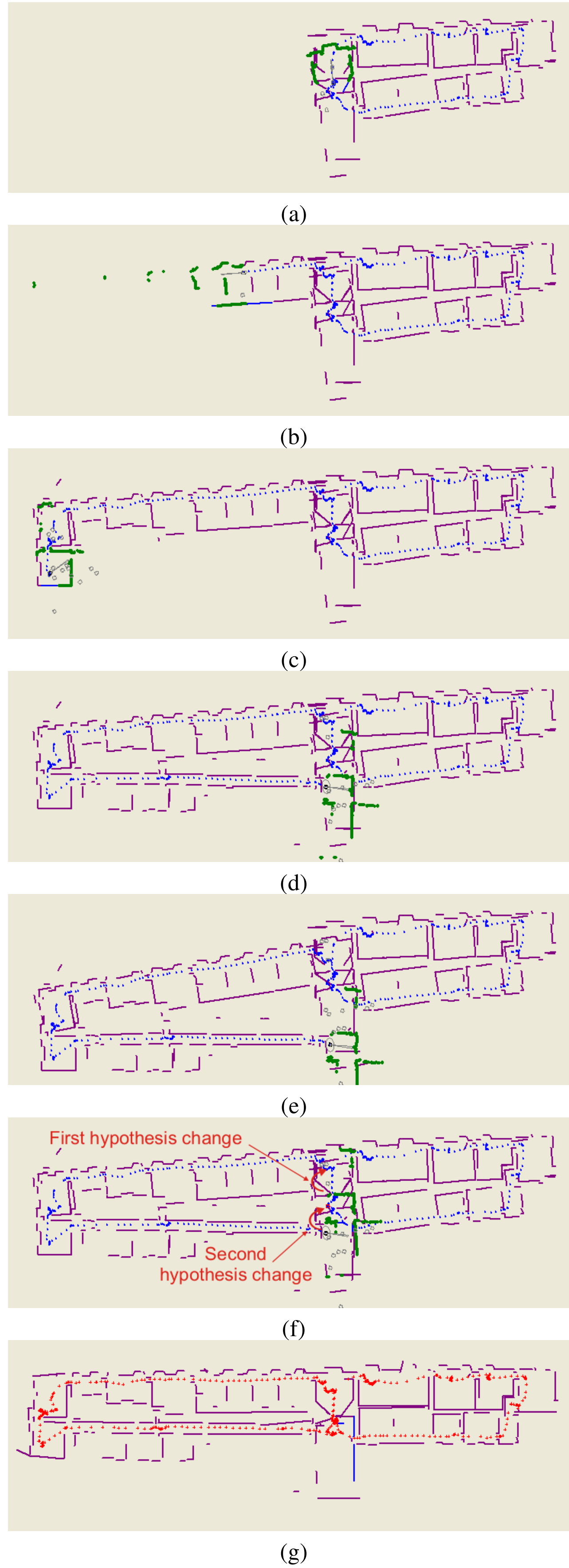

Fig. 5. Operation of the algorithm in a real environment: (a) Step A: Initial hypothesis after first loop, (b,c,d) Step A: Most probable hypothesis after first loop, (e) Step A: Initial hypothesis after second loop, (f) Step A: Most probable hypothesis after second loop, (g) Step B:Final Rectified map.