Exploration with Active Loop-Closing for FastSLAM

Cyrill Stachniss Dirk Hähnel Wolfram Burgard

University of Freiburg
Department of Computer Science
D-79110 Freiburg, Germany

Abstract— **Acquiring models of the environment belongs to the fundamental tasks of mobile robots. In the last few years several researchers have focused on the problem of simultaneous localization and mapping (SLAM). Classic SLAM approaches are passive in the sense that they only process the perceived sensor data and do not influence the motion of the mobile robot. In this paper we present a novel and integrated approach that combines autonomous exploration with simultaneous localization and mapping. Our method uses a grid-based version of the FastSLAM algorithm and at each point in time considers actions to actively close loops during exploration. By re-entering already visited areas the robot reduces its localization error and this way learns more accurate maps. Experimental results presented in this paper illustrate the advantage of our method over pervious approaches lacking the ability to actively close loops.**

I. Introduction

In general, the task of acquiring models of unknown environments requires solutions to three sub-tasks, which are mapping, localization and control (see Figure 1). Mapping is the problem of integrating the information gathered with the robot's sensors into a given representation. Localization is the problem of estimating the position of the robot. Finally, the control problem involves the question of how to steer a vehicle in order to efficiently guide it to a desired location.

The diagram also shows the overlapping areas of these sub-tasks. Simultaneous localization and mapping, also called SLAM, is the problem of building a map based on a position estimation and simultaneously localizing the robot within the map constructed so far. Active localization seeks to guide the robot to locations within the map to improve the pose estimation. In contrast to this, exploration approaches focus on guiding the robot efficiently through the environment to build a map. The center area of Figure 1 represents the so-called integrated approaches which simultaneously address mapping, localization and motion control.

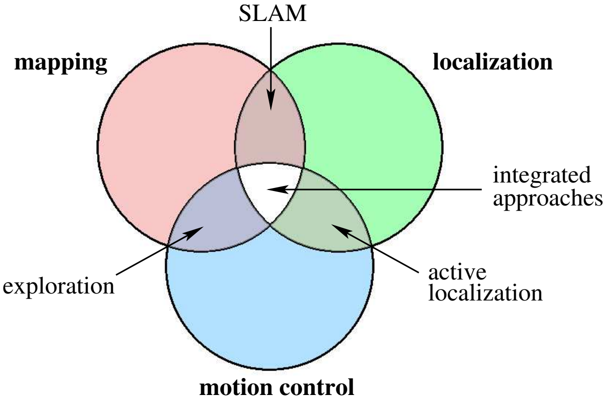

Fig. 1. Sub-tasks that need to be solved by a robot to acquire accurate models of the environment [10]. The overlapping areas represent combinations of these sub-tasks.

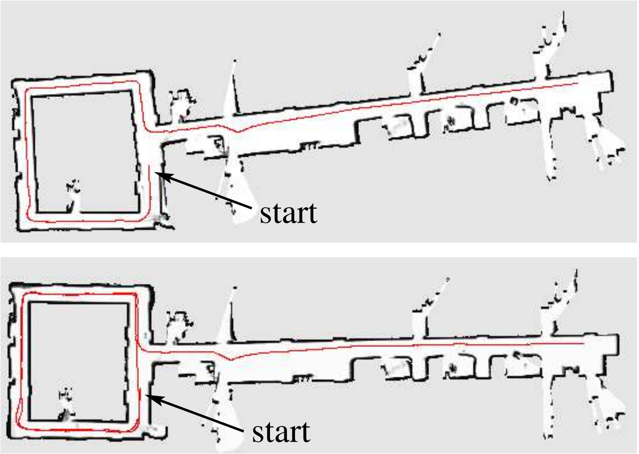

Fig. 2. This figure shows two maps obtained by a real world experiment performed at Sieg Hall, University of Washington. The top image depicts an experiment in which the robot drove around the loop once and then entered the long corridor. As can be seen robot was unable to localize itself correctly before entering the corridor. This leaded to a big error in the orientation of the horizontal corridor. If the robot did active loop-closing and re-visited the loop it typically performed much better (bottom image).

A naive approach to realize an integrated technique could be to combine a SLAM algorithm with an exploration procedure. Since exploration strategies try to explore unknown terrain as fast as possible, they focus on reducing the amount of unseen area and thus avoid repeatedly traveling through known areas. This strategy, however, is suboptimal in the context of the SLAM problem, because the robot typically needs to re-visit places to localize itself again. A good pose estimation is necessary to make the correct data association, i.e., to determine if the current measurements fit into the map built so far. If the robot uses an exploration strategy that avoids multiple visits of the same place, the probability of making the correct associations is reduced. This indicates that combinations of exploration strategies and SLAM algorithms should consider whether it is worth re-entering already covered spaces or to explore new terrain. It can be expected that a system which takes this decision into account can improve the quality of the resulting map.

Figure 2 gives an example that illustrates why an integrated approach doing active place re-visiting provides better results than approaches that do not consider re-entering known terrain during the exploration phase. In the situation shown in the upper image the robot traversed the loop just once. The robot was not able to correctly determine the angle between the loop and the straight corridor, because it did not collect enough data to accurately localize itself. The second map shown in the lower image has been obtained with the approach described in this paper after the robot traveled twice around the loop before entering the corridor. As can be seen from the figure, this reduces the orientation error from approximately 7 degrees (top image) to 1 degree (bottom image). This example illustrates that the capability to actively close loops during exploration allows the robot to reduce its pose uncertainty during exploration and thus to learn more accurate maps.

The contribution of this paper is an integrated algorithm for generating trajectories to actively close loops during SLAM. Our algorithm uses a grid-based version of the FastSLAM algorithm and explicitely takes into account the uncertainty about the pose of the robot during the exploration task. Additionally it avoids that the robot becomes overly confident in its pose when actively closing loops which is a typical problem of particle filters in this context. As a result we obtain more accurate maps compared to combinations of SLAM with greedy exploration.

This paper is organized as follows. After the discussion of related work in the following section, we explain the idea of grid-based FastSLAM, the SLAM algorithm used throughout this work. In Section IV we present our integrated exploration technique. We furthermore describe how to take into account the pose uncertainty and how to actively close loops. Section V then presents experiments carried out on real robots as well as in simulation.

II. Related Work

This paper presents an integrated technique to simultaneous localization, mapping, and exploration. Several previous approaches to SLAM and mobile robot exploration are relevant. In the context of exploration, most of the techniques presented so far focus on generating motion commands that minimize the time needed to cover the whole terrain [1, 9, 17, 18]. Other methods seek to optimize the view-points of the robot to maximize the expected information gain and to minimize the uncertainty of the robot about grid cells [6, 14]. Most of these techniques, however, assume that the location of the robot is known during exploration. In the area of SLAM the vast majority of papers focuses on the aspect of state estimation as well as belief representation and update [2, 3, 4, 7, 8, 11, 12, 15]. These techniques, however, are passive and only consume incoming sensor data without explicitely generating controls.

Recently, some techniques have been proposed which actively control the robot during simultaneous mapping and localization. For example, Makarenko et al. [10] extract landmarks out of laser range scans and use an Extended Kalman Filter to solve the SLAM problem. They furthermore introduce a utility function which trades-off the cost of reaching frontiers with the utility of selected positions with respect to a potential reduction of the pose uncertainty. This approach is similar to the work done by Feder et al. [5] who consider local decisions to improve the pose estimate during mapping. Both techniques, however, rely on the fact that the environment contains landmarks that can be uniquely determined during mapping.

In contrast to this, the approach presented in this paper makes no assumptions about distinguishable landmarks in the environment. It uses raw laser range scans to compute accurate grid maps. It considers the utility of re-entering known parts of the environment and following an encountered loop to reduce the uncertainty of the robot in its pose. This way the resulting maps become highly accurate.

III. Grid-Based FastSLAM

To estimate the map of the environment we use a highly efficient variant of the FastSLAM algorithm [11] which itself is an extension of the Rao-Blackwellized particle filter for simultaneous localization and mapping proposed by Murphy et al. [3]. The key idea of the Rao-Blackwellized particle filter for SLAM is to estimate a posterior $p(x_{1:t} \mid z_{1:t}, u_{0:t-1})$ about potential trajectories $x_{1:t}$ of the robot given its observations $z_{1:t}$ and its odometry measurements $u_{0:t-1}$ and to use this posterior to compute a posterior over maps and trajectories:

$$\begin{aligned} p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) = \\ p(m \mid x_{1:t}, z_{1:t}) p(x_{1:t} \mid z_{1:t}, u_{0:t-1}). \end{aligned} \tag{1}$$

This can be done efficiently, since the quantity $p(m \mid x_{1:t}, z_{1:t}, u_{0:t-1})$ can be computed analytically once $x_{1:t}$ and $z_{1:t}$ are known. To estimate the posterior $p(x_{1:t} \mid z_{1:t}, u_{0:t-1})$ over the potential trajectories FastSLAM uses a particle filter in which an individual map is associated to every sample. Each map is constructed given the observations $z_{1:t}$ and the trajectory $x_{1:t}$ represented by the corresponding particle. During resampling, the weight ω_t of each particle is proportional to the likelihood $p(z_t \mid m, x_t)$ of the most recent observation given the map m associated to this particle and the pose x_t of the corresponding trajectory.

The FastSLAM algorithm used throughout this paper computes grid maps. It applies a scan-matching procedure to compute highly accurate odometry data and uses this corrected odometry in the prediction step of the particle filter [8]. This way the number of particles can be reduced so that maps of even large environments can be constructed online. In the following section we describe how the FastSLAM algorithm for grid maps can be extended to actively close loops during exploration.

IV. Exploration With Active Loop-Closing for FastSLAM

During FastSLAM, whenever the robot explores new terrain, all samples have more or less the same importance weight, since the most recent measurement is typically consistent with the part of the map constructed from the immediately preceding observations. As a result, the uncertainty of the particle filter increases. As soon as it re-enters known terrain, however,

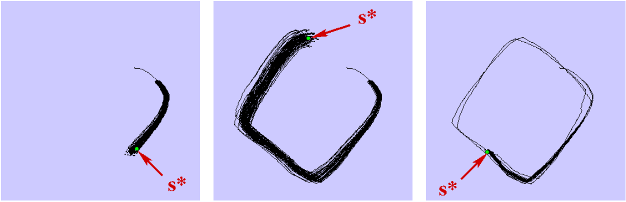

Fig. 3. Evolution of a particle set and the map of the most likely particle s^* at three different time steps. In the left two images the vehicle traveled through unknown terrain, so that the uncertainty increased. In the right image the robot re-entered known terrain so that samples representing unlikely trajectories were depleted.

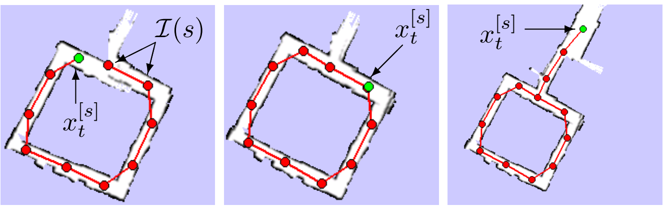

Fig. 4. The red dots and lines in these three image represent the nodes and edges of $\mathcal{G}^{[s]}$. In the left image $\mathcal{I}(s)$ contained two nodes and in the middle image the robot closed the loop until the pose uncertainty is reduced. After this it continued with the acquisition of unknown terrain (right image).

the maps of some particles are consistent with the current measurement and some are not. Accordingly the weights of the samples differ largely. Due to the resampling step the uncertainty about the pose of the robot usually decreases. One typical example is shown in Figure 3. In the left two images the robot explores new terrain and the uncertainty of the sample set increases. In the right image the robot travels through known terrain and unlikely particles have vanished. Note that this effect is much smaller if the robot just moves backward a few meters to re-visit previously scanned areas. This is because each map associated to a particle is generally locally consistent. Inconsistencies mostly arise when the robot re-enters areas explored some time ago. Therefore, visiting places seen further back in the history has a stronger effect on the differences between the importance weights and typically also on the reduction of uncertainty compared to places recently observed.

The key idea of our approach is to identify opportunities for closing loops during terrain acquisition. Here closing a loop means actively re-entering the known terrain and following a previously traversed path. To determine, whether there exists a possibility to close a loop we consider two different representations of the environment. In our current system we associate to each particle s an occupancy grid map $m^{[s]}$ and a topological map $\mathcal{G}^{[s]}$ which both are updated while the robot is performing the exploration task. In the topological map $\mathcal{G}^{[s]}$ the vertices represent positions visited by the robot. The edges represent the trajectory corresponding to the particle s. To construct the topological map we initialize it with one node corresponding to the starting location of the robot. Let $x_t^{[s]}$ be the pose of particle s at the current time step t. We add a new node at $x_t^{[s]}$ to $\mathcal{G}^{[s]}$ if the distance between $x_t^{[s]}$ and all other nodes in $\mathcal{G}^{[s]}$ exceeds a threshold of $c = 2.5m$ or if none of the other nodes in $\mathcal{G}^{[s]}$ is visible from $x_t^{[s]}$:

$$\forall n \in nodes(\mathcal{G}^{[s]}) \quad : \quad \left[dist_{m^{[s]}}(x_t^{[s]}, n) > c \quad \vee \quad not\_visible_{m^{[s]}}(x_t^{[s]}, n) \right]. \qquad (2)$$

Whenever a new node is added, we also add an edge from this node to the most recently visited node. To determine whether or not a node is visible from another node we perform a ray-casting operation in the occupancy grid $m^{[s]}$.

Figure 4 depicts such a graph for one particular particle during different phases of an exploration task. In each image, the topological map $\mathcal{G}^{[s]}$ is printed on top of metric map $m^{[s]}$. To motivate the idea of our approach we would like to refer the reader to the left image of this figure. Here the robot was almost closing a loop. This can be detected by the fact that the length of the shortest path between the current pose of the robot and previously visited locations in the topological map $\mathcal{G}^{[s]}$ was large, where as it was small in the grid-map $m^{[s]}$.

Thus, to determine whether or not a loop can be closed we compute for each sample s the set $\mathcal{I}(s)$ of positions of interest, which contains all nodes that are close to current pose $x_t^{[s]}$ of particle s based on the grid map $m^{[s]}$ but are far away given the topological map $\mathcal{G}^{[s]}$ of s:

$$\mathcal{I}(s) \quad = \{x_{t'}^{[s]} \in nodes(\mathcal{G}^{[s]}) \mid dist_{m^{[s]}}(x_{t'}^{[s]}, x_t^{[s]}) < c_1 \wedge dist_{\mathcal{G}^{[s]}}(x_{t'}^{[s]}, x_t^{[s]}) > c_2\}. \quad (3)$$

Here $dist_{\mathcal{M}}(x_1, x_2)$ is the length of the shortest path from x_1 to x_2 given the representation $\mathcal{M}$. The distance between two nodes in $\mathcal{G}^{[s]}$ is given by the length of the shortest path between both nodes, whereas the length of a path is computed by the sum over the lengths of the traversed edges between these nodes. The terms c_1 and c_2 are constants that must satisfy the constraint $c_1 < c_2$. In our current implementation the values of these constants are $c_1 = 6m$ and $c_2 = 20m$.

If $\mathcal{I}(s) \neq \emptyset$ there exist so-called shortcuts from $x_t^{[s]}$ to the positions in $\mathcal{I}(s)$. These shortcuts represent edges that would close a loop in the topological map $\mathcal{G}^{[s]}$. The left image of Figure 4 illustrates a situation in which a robot encounters the opportunity to close a loop since $\mathcal{I}(s)$ contains two nodes. The key idea of our approach is to use such shortcuts whenever the uncertainty of the robot in its pose becomes too large. The robot then re-visits portions of the previously explored area and this way reduces the uncertainty in its position.

To determine the most likely movement allowing the robot to follow a previous path of a loop, one in principle has to integrate over all particles and consider all potential outcomes of that particular action. Since this would be too time consuming for online-processing we consider only the particle s^* with the highest accumulated importance weight:

$$s^* \quad = \quad \underset{s}{\operatorname{argmax}} \sum_{i=1}^{t} \log \omega_i^{[s]}. \qquad (4)$$

Here $\omega_i^{[s]}$ is the weight of sample s at time step i. If $\mathcal{I}(s^*) \neq \emptyset$

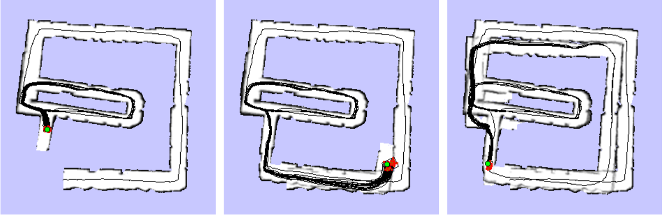

Fig. 5. The particle depletion problem: a robot traveled through the inner loop several times (left image). After this the diversity of hypotheses about the trajectory outside the inner loop had decreased too much (middle image) and the robot is unable to close the outer loop correctly (right image).

we choose the node x_{t_e} from $\mathcal{I}(s^*)$ which is closest to $x_t^{[s^*]}$:

$$x_{t_e} = \underset{x \in \mathcal{I}(s^*)}{\operatorname{argmin}}\, dist_{m^{[s^*]}}(x_t^{[s^*]}, x). \qquad (5)$$

In the sequel x_{t_e} is denoted as the *entry point* at which the robot has the possibility to close a loop. t_e corresponds to the last time the robot was at the node x_{t_e}.

To determine whether or not the robot should activate the loop-closing behavior our system constantly monitors the uncertainty $\mathcal{H}(t)$ about the robot's pose at the current time step. The necessary condition for starting the loop-closing process is the existence of an entry point x_{t_e} and that $\mathcal{H}(t)$ exceeds a given threshold. Once the loop-closing process has been activated, the robot approaches x_{t_e} and then follows the path taken after arriving previously at x_{t_e}. During this process the uncertainty in the pose of the vehicle typically decreases, because the robot is able to localize itself in the map built so far and unlikely particles vanish.

We furthermore have to define a criterion for deciding when the robot actually has to stop following a loop. A first attempt could be to introduce a threshold and to simply stop the trajectory following behavior as soon as the uncertainty becomes smaller than a given threshold. This criterion, however, can be problematic especially in the case of nested loops. Suppose the robot encounters the opportunity to close a loop that is nested within an outer and so far unclosed loop. If it eliminates all of its uncertainty by repeatedly traversing the inner loop, particles necessary to close the outer loop may vanish. As a result, the filter diverges and the robot fails to build a correct map (see Figure 5). To remedy this so-called particle depletion problem [16] we introduce a constraint on the uncertainty of the robot. Let $\mathcal{H}(t_e)$ denote the uncertainty of the posterior when the robot visited the entry point last time. Then the new constraint allows the robot to re-traverse the loop only as long as its current uncertainty $\mathcal{H}(t)$ exceeds $\mathcal{H}(t_e)$. If the constraint is violated the robot resumes its frontier-based exploration process. The idea of this constraint is to avoid the depletion of relevant particles during the loop-closing process.

To better illustrate the importance of this constraint consider the following example: a robot moves from place A to place B and then repeatedly observes B. While it is mapping B it does not get any further information about A. Since each particle represents a whole trajectory of the robot also hypotheses representing ambiguities about A will vanish when reducing potential uncertainties about B. Our constraint avoids the depletion of particles representing ambiguities about A by aborting the loop-closing behavior at B as soon as the uncertainty drops below the uncertainty stemming from A.

Algorithm 1 The loop-closing algorithm

```
Compute $\mathcal{I}(s^*)$
if $\mathcal{I}(s^*) \neq \emptyset$ then begin
    $\mathcal{H} \leftarrow \mathcal{H}(t_e)$
    $path \leftarrow x_t^{[s^*]} \cdot shortest\_path_{\mathcal{G}^{[s^*]}}(x_{t_e}, x_t^{[s^*]})$
    while $\mathcal{H}(t) > \mathcal{H} \ \wedge \ \mathcal{H}(t) > threshold$ do
        $robot\_follow(path)$
end
```

Finally we have to describe how we actually measure the uncertainty in the position estimate. The typical way of measuring the uncertainty of a posterior is to calculate the entropy. In the case of multi-modal distributions, however, the entropy does not consider the distance between the different modes. In our experiments we figured out that we obtain better results if we use the volume expanded by the samples instead of the entropy of the posterior. We therefore calculate the pose uncertainty by determining the volume of the oriented bounding box around the particle cloud. A good approximation of the minimal oriented bounding box can be obtained efficiently by a principal component analysis.

As long as the robot is localized well enough or no loop can be closed, we use a frontier-based exploration strategy [1] to choose target points for the robot. In our current system we determine frontiers based on the map of the most likely particle s^*. Here a frontier is any known cell that is an immediate neighbor of an unknown, unexplored cell [18].

A precise formulation of the loop-closing strategy is given by Algorithm 1. In our implementation this algorithm runs as a background process that is able interrupt the frontier-based exploration procedure. An application of this algorithm in a simulation run is illustrated in Figure 4.

A. Handling Multiple Nested Loops

Note that our loop-closing technique can also handle multiple nested loops. During the loop-closing process the robot follows its previously taken trajectory to re-localize. It does not leave this trajectory until the termination criterion, described in previous section, is fulfilled. Therefore it never starts a new loop-closing process before the current one is completed. A typical example with multiple nested loops is shown in Figure 6. In the situation depicted in the left image the robot starts with the loop-closing process for the inner loop. After completing this loop it moves to the second inner one and again starts the loop-closing process. Since our algorithm considers the uncertainty at the entry point it keeps enough variance in the filter to close the outer loop. In general, the quality of the solution and whether or not the overall process succeeds depends on the number of particles used. Since determining this quantity is an open research problem the number of particles has to be defined by the user in our current system.

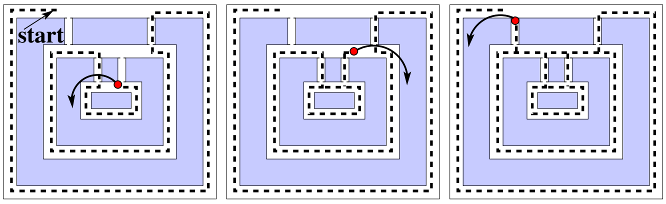

Fig. 6. Active Loop-Closing in an environment with multiple nested loops.

V. EXPERIMENTS

Our approach has been implemented and evaluated in a series of real world and simulation experiments. For the real world experiments we used an iRobot B21r robot and an ActivMedia Pioneer II robot. Both are equipped with a SICK laser range finder. For the simulation experiments we used the real-time simulator of the Carnegie Mellon Robot Navigation Toolkit (CARMEN) [13]. This simulator generates realistic noise in the odometry and laser range sensor data.

The experiments described in this section are designed to illustrate that our approach can be used to actively learn accurate maps of large indoor environments. Furthermore, they demonstrate that our integrated approach yields better results than an approach without active loop-closing. Additionally, we analyze how the active termination of the loop-closure influences the result of the mapping process.

A. Real World Exploration

The first experiment was carried out to illustrate that our current system can effectively control a mobile robot to actively close loops during exploration. To perform this experiment we used a Pioneer II robot to explore the main lobby of the Department for Computer Science at the University of Freiburg. The size of this environment is 51m times 18m. Figure 7 depicts the final result obtained by a completely autonomous exploration run using our active loop-closing technique. It also depicts the trajectory of the robot, which has an overall length of 280m. The robot decided four times to re-enter a previously visited loop in order to reduce the uncertainty in its pose. Figure 7 also shows the corresponding entry points as well as the positions where the robot left the loops ("exit points"). In this experiment the FastSLAM routine used 250 particles. As can be seen the resulting map is quite accurate.

B. Active Loop-Closing vs. Frontier-Based Exploration

The second experiment was carried out to compare our algorithm with a standard exploration strategy that does not consider loop closing actions. The lower image of Figure 2 shows the map obtained with a B21r robot in the Sieg Hall at the University of Washington using our algorithm. To eliminate the influence of measurement noise and different movements of the robot we removed the data corresponding to the second loop traversal from the recorded data file and used this data as input to our FastSLAM algorithm. This way we simulated the behavior of a greed exploration strategy which forces the robot to directly enter the corridor after returning to the starting location in the loop. As can be seen from the upper image of Figure 2, an approach that does not actively re-enter the loop fails to correctly estimate the angle between the loop and the corridor which should be oriented horizontally in that figure. Whereas the angular error is 7 degrees with the standard approach it is only 1 degree with our method. Both maps correspond to the particle with the highest accumulated importance factor.

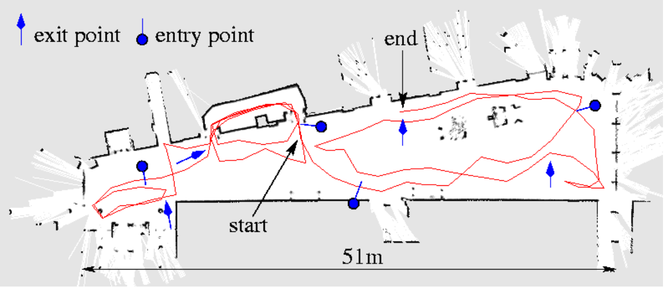

Fig. 7. This image shows the resulting map of an exploration experiment carried out using a Pioneer II robot equipped with a laser range scanner in the entrance hall of the Department for Computer Science at the University of Freiburg. Also shown is the path of the robot as well as entry and exit points where the robot started and stopped the active loop-closing process.

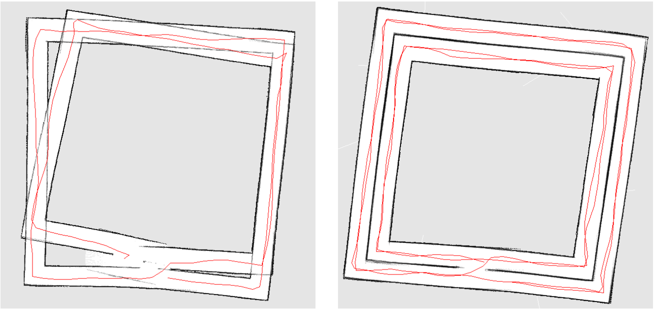

Fig. 8. This figure depicts an environment with two large loops. The outer loop has a length of over 220m. The left image show the resulting map of a trajectory in which the robot drove through the loops only once. In the second run the robot visited every loop twice and obtained a highly accurate map (see right image).

A further experiment that illustrates the advantage of place re-visiting is shown in Figure 8. The environment used in this simulation run is 80 times 80 meters and contains two large nested loops with nearly featureless corridors. The left image shows the result of the frontier-based approach which traversed each loop only once. Since the robot is not able to correct the accumulated pose error, the resulting map contains large inconsistencies and two of the corridors are mapped onto each other. Our approach, in contrast, first revisits the outer loop before entering the inner one (see right image). Accordingly, the resulting map is quite accurate.

C. A Quantitative Analysis

To quantitatively evaluate the advantage of the loop-closing behavior we performed a series of simulation experiments in an environment similar to the Sieg Hall. We performed 20 experiments, 10 with active loop-closing and 10 without. After completing the exploration task we measured the average error in the relative distances between positions lying on the

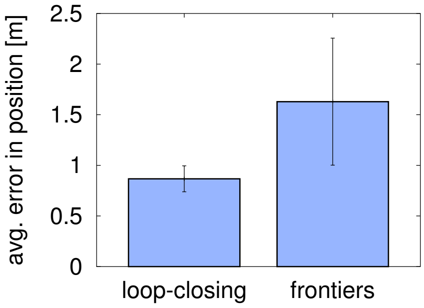

Fig. 9. This figure compares our loop-closing strategy with a pure frontier-based exploration technique. The left bar in this graph plots the average error in the pose of the robot obtained with our loop-closing strategy. The right one shows the average error if a frontier-based approach was used. As can be seen our technique significantly reduces the distances between the estimated positions and the ground truth (confidence intervals do not overlap).

resulting estimated trajectory and the ground truth provided by the simulator. The results are depicted in Figure 9. As can be seen the active loop-closing behavior significantly reduces the error in the position of the robot.

D. Importance of the Termination Criterion

In this final experiment we analyze the importance of the constraint that terminates the active loop-closing behavior as soon as the current uncertainty $\mathcal{H}(t)$ of the belief drops under the uncertainty $\mathcal{H}(t_e)$ of the posterior when the robot was at the entry point last time.

In this simulated experiment the robot had to explore an environment containing two nested loops (see Figure 10 (d)). In one case we simply used a constant threshold to determine whether or not the loop-closing behavior should be stopped. In the second case we applied the additional constraint that the uncertainty should not become smaller than $\mathcal{H}(t_e)$.

Figure 5 shows the map of the particle with the highest accumulated importance weight obtained with our algorithm using a constant threshold instead of considering $\mathcal{H}(t_e)$. In this case the robot repeatedly traversed the inner loop (left image) until its uncertainty was reduced below a threshold. After three and a half rounds it decided to again explore unknown terrain, but the diversity of hypotheses had decreased too much (middle image). Accordingly the robot was unable to accurately close the outer loop (right image). We repeated this experiment several times and in no case the robot was able to correctly map the environment. In contrast to that, our approach using the additional constraint always generated an accurate map. One example run is shown in Figure 10. Here the robot stopped the loop-closing after traversing half of the inner loop. In both cases we used 80 particles.

As this experiment illustrates, the termination of the loop-closing is important for the convergence of the filter and to obtain accurate maps in environments with several (nested) loops. Note that similar results in principle can also be obtained without this termination constraint if the number of particles is dramatically increased. Since exploration is an online problem and since every particle carries its own map it is of utmost importance to keep the number of particles as small as possible. Therefore our approach also can be regarded as a contribution to limit the number of particles during FastSLAM.

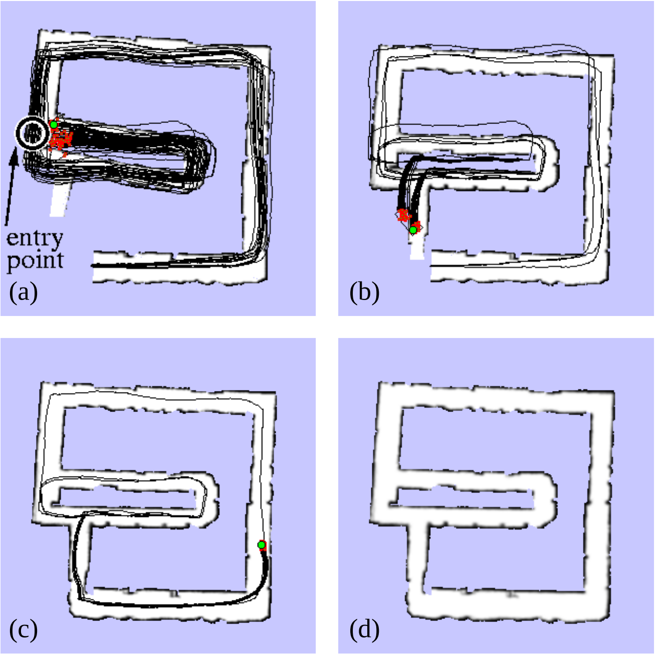

Fig. 10. These images depict snapshots of our loop-closing strategy. The robot explored the terrain and detected an opportunity to close a loop in order to reduce its uncertainty (a). It then traversed parts of the inner loop until its uncertainty $\mathcal{H}(t)$ did not exceed the uncertainty $\mathcal{H}(t_e)$ of the posterior when the robot at the entry point anymore. It then turned back and left the loop to explore new terrain (b). After this, enough hypotheses are left to correctly close the outer loop (c) and (d). In contrast to that, a system considering only a constant threshold criterion fails to map the environment correctly as depicted in Figure 5.

VI. Conclusion

In this paper we presented a novel approach for active loop-closing during autonomous exploration. We combined a Rao-Blackwellized particle filter for localization and mapping with a frontier-based exploration technique extended by the ability to actively close loops. Our algorithm forces the robot to traverse previously visited loops again and this way reduces the uncertainty in the pose estimation. As a result, we obtain more accurate maps compared to standard combinations of SLAM algorithms with exploration techniques.

One general problem of FastSLAM is that the number of particles needed to build an accurate map is not known in advance. Even our technique does not provide tools to estimate this quantity but it produces better maps with a given number of particles compared to a naive combination of frontier-based exploration with FastSLAM. The major restrictions of our algorithm are similar to those of FastSLAM, e.g, there are no means to recover from divergence without a complete re-run of the whole algorithm. Such issues are subject of future research.

Acknowledgment

This work has partly been supported by the German Science Foundation (DFG) under contract number SFB/TR-8 (A3) and by the EC under contract number IST-2000-29456. The authors would like to thank all people who are involved in the development of CARMEN, especially Nicholas Roy. Furthermore we would like to thank Luis Montesano for the fruitful discussions and help during the real world experiments.

REFERENCES

[1] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2000.

[2] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *ICRA'2000 Workshop on Mobile Robot Navigation and Mapping*, 2000.

[3] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russel. Rao-blackwellized partcile filtering for dynamic bayesian networks. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[4] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultainous localization and mapping without predetermined landmarks. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2003.

[5] H. Feder, J. Leonard, and C. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7), 1999.

[6] R. Grabowski, P. Khosla, and H. Choset. Autonomous exploration via regions of interest. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[7] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.

[8] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[9] S. Koenig and C. Tovey. Improved analysis of greedy mapping. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[10] A. A. Makarenko, S. B. Williams, F. Bourgoult, and F. Durrant-Whyte. An experiment in integrated exploration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

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

[12] K. Murphy. Bayesian map learning in dynamic environments. In *Neural Info. Proc. Systems (NIPS)*, 1999.

[13] N. Roy, M. Montemerlo, and S. Thrun. Perspectives on standardization in mobile robot programming. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[14] C. Stachniss and W. Burgard. Exploring unknown environments with mobile robots using coverage maps. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2003.

[15] S. Thrun. An online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 2001.

[16] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR380, Cambridge University Engineering Department, August 2000.

[17] G. Weiß, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 1994.

[18] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents*, 1998.