

A Scalable Hybrid Multi-Robot SLAM Method for Highly Detailed Maps.

Max Pfingsthorn¹, Bayu Slamet², and Arnaud Visser²

¹ Jacobs University Bremen**, Campus Ring 1, 28759 Bremen, Germany

² Universiteit van Amsterdam, 1098 SJ Amsterdam, the Netherlands

Abstract. Recent successful SLAM methods employ hybrid map representations combining the strengths of topological maps and occupancy grids. Such representations often facilitate multi-agent mapping. In this paper, a successful SLAM method is presented, which is inspired by the *manifold* data structure by Howard et al. This method maintains a graph with sensor observations stored in vertices and pose differences including uncertainty information stored in edges. Through its graph structure, updates are local and can be efficiently communicated to peers. The graph links represent known traversable space, and facilitate tasks like path planning. We demonstrate that our SLAM method produces very detailed maps without sacrificing scalability. The presented method was used by the UvA Rescue Virtual Robots team, which won the Best Mapping Award in the RoboCup Rescue Virtual Robots competition in 2006.

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a vital technology for autonomous mobile robots. Using a SLAM algorithm, the robot can keep track of its location by maintaining a map of the physical environment and an estimate of its position on that map. This provides a spatial context for the interpretation of current and past observations and enables higher level reasoning, control, and coordination. The map delivered may also provide an intuitive representation with which the robot can convey its findings to humans. For this purpose the visualization of the map may be augmented with any kind of extra information that the robot is able to infer.

As described by Thrun [1], SLAM algorithms can be roughly classified according to the map representation and the employed estimation technique. A very popular map representation is the occupancy grid [2]. Grid-based approaches typically require a large amount of memory, however they are able to represent the environment at arbitrary resolution and thereby have the potential to be highly detailed.

Graph-based representations on the other hand primarily map the topology of the environment. This results in a much more compact description and makes graph-based maps an attractive alternative when scalability to multiple robots is

** formerly International University Bremen

a concern. The low memory requirement of graph-based maps allows for low-cost information sharing between multiple robots, e.g. via a wireless network. Also, as topological graphs provide a direct description of the free-space regions and their interconnectedness, they significantly facilitate path planning algorithms. A disadvantage of such maps is that localization is limited to the nearest node due to the lack of more detailed information. Additionally, this absence of low-level geometric information precludes the rendering of detailed visualizations.

Given the individual strengths and shortcomings, researchers have tried to combine multiple representations in hybrid approaches [3–7]. These are potentially as scalable as topological approaches, and at the same time provide the same geometric detail as grid maps.

In this paper we present a hybrid SLAM method that combines grid-based and topological representations. The underlying data structure exhibits both desired properties: Our method produces highly detailed maps without sacrificing scalability. As part of our research presented in this paper, we participated in the Rescue Virtual Robots competition in the RoboCup World Championships of 2006 [8] where we demonstrated the presented approach. Our system supported teams of up to 8 robots that searched the computer-simulated emergency site [9, 10] and jointly constructed a map of the environment. Our maps ranked highest after being evaluated on a number of aspects as described in [8] and earned the Best Mapping Award.

2 The Data Structure

The data structure the presented method is built on is inspired by the *manifold* concept conceived by Howard et al.[7]. It is a layered data structure with a topological organization at the global level and small detailed metric maps at the local level. Sensor observations are not integrated into small grid maps as done in other current approaches [5], but kept as raw data in the nodes for later processing. Edges represent known connections between poses where these sensor observations were made and contain a transformation between these two poses. Additionally, uncertainty information about these transformations are stored in the edges.

Such a data structure is a very natural observation-centric formulation of a map. It also explicitly includes the important factorization that was the key insight used in FastSLAM [11]: Sensor observations are conditionally independent given the robot path. The observations are kept explicitly connected to the easily correctable path encoded in the graph.

Effectively, the sensor data stored in a node is a self-contained piece of information. This means that node poses can be updated without much computational overhead, which is in direct contrast to grid-based maps. Also, the usual slight update of pose differences does not invalidate associated sensor readings or the general map integrity. The map stays usable for many algorithms, like path planners.

Formally, the map m consists of a set of local sensor observations $\{\pi\}$, and a set of links $\{\phi\}$, with

$$\phi = \{i, j, \delta, \Sigma\}$$

where i and j are the indices of the two observations which are linked, δ is the estimated pose difference, and Σ is that estimation’s covariance matrix.

For the multi-robot scenario, similar benefits apply. The displacement information present in the links is independent of any global coordinate frame, so collaborating robots can easily exchange parts of the graph without further processing.

3 Single-Robot SLAM

Given the above data structure, the main challenge is to gather the information necessary to construct the links ϕ .

Like in most current SLAM methods, a laser range scanner is used as the main source of mapping information. Odometry measurements are not used, the method exclusively relies on scan matching to estimate a robot’s displacement. Thus, no explicit motion model must be developed. The corresponding covariance matrix can be computed either by sampling as described in [12] or directly from the scan matching result [13, 14].

As long as the uncertainty is low enough, the information in the map is used to get an estimate of the current location. When new sensor data arrives, the scan matching algorithm is used to compare the current range scan to the scan stored at the current node. This results in a new location estimate. At the moment the uncertainty for the scan matching operation increases, a new node is created to store the current scan and a new link is created containing the latest displacement estimation.

We chose the Weighted Scan Matcher (WSM) by Pfister et al. [14], which belongs to the Iterative Closest Point (ICP) family. The IDC scan matcher by Lu and Milius [15] belongs to the same family and has been the most popular ICP scan matcher. However, WSM is known to outperform IDC significantly in both accuracy and speed, given that dense range scans are available [14]. In [16], we produced similar results in an extensive set of experiments specific to the simulated environment at the RoboCup Rescue Virtual Robots competition. In addition, computing the covariance matrix of the displacement given by WSM is easy.

Given the data structure and a means to acquire the needed displacement estimate, a naïve implementation would result in incremental SLAM. Without any further processing, a new laser range scan could be matched against the previous one (or a set of previous ones) and added to the graph. However, this often results in accumulated error, as shown in Figure 1a.

The main manifestations of this error are inconsistencies in overlapping map regions where the robot travelled in a loop. Overlaps in our map representation can be detected via feature extraction and comparison on a node by node basis, by mutual observations of robots in a multi-agent setting, or by re-observing

artificial or natural landmarks. Due to error accumulation, the two position estimates of the overlapping regions are not likely to project to the same global coordinates.

In the USARSim [9,10] simulator used in the RoboCup Rescue Virtual Robots competition, victims and robot-placable RFIDs were uniquely identifiable. This made them perfect unique landmarks, and thus a great and reliable shortcut to loop detection.

Loop-Closing is the main way to reduce the accumulated error in the map. A new link is inserted into the graph by computing the displacement between the two overlapping nodes. The inconsistency introduced by the new link has to be resolved by incorporating this new information into the graph as a whole.

Re-matching each node to its respective neighbors incrementally, starting from the new link, produces good results and has been used in the literature before [7]. Figure 1b shows the corrected map using this approach.

It is also possible to close loops by optimizing the estimated global positions of graph nodes such that the estimated displacements stored in the links apply best to all nodes. Here, the probability distribution formed by the initial pose estimate δ and its covariance matrix Σ can be used to evaluate how good a certain global positioning fits a given link, as shown by Olson et al. [17].

4 Multi-Robot SLAM

A graph-based map can be very easily shared with and communicated to collaborating robots. Only new nodes and corresponding links have to be transferred over a network connection to communicate map updates completely. New data only needs to be appended to the graph and connected to the right nodes. Scan matching does not have to be performed again.

Keeping multiple disconnected maps in memory is problematic in other successful SLAM methods. In the context of the graph however, this is trivial. We allow the graph to contain multiple disconnected components, one for each robot. Similarly, it is possible to start a new disconnected component when a robot loses track of its location, for example after falling down stairs. This is also an attractive solution to the “kidnapped robot problem”.

The main challenge is merging the disconnected partial maps in a meaningful way. Incidentally, this process is very similar to loop closing. The same techniques can be used to detect overlaps in two disconnected maps. Map merging is done by computing a transformation between the two overlapping regions once by scan matching. Subsequently, one of the two maps is transformed as a rigid body and moved so that the two overlapping parts fit. Optionally, a loop closing operation may be run to refit the two maps for improved accuracy. An example is shown in the bottom of Figure 1.

Closing loops and merging partial maps can be delayed without impacting the continuity of the map. Only some collaboration has to be sacrificed for this significant reduction in immediate computational requirement. Some robots may

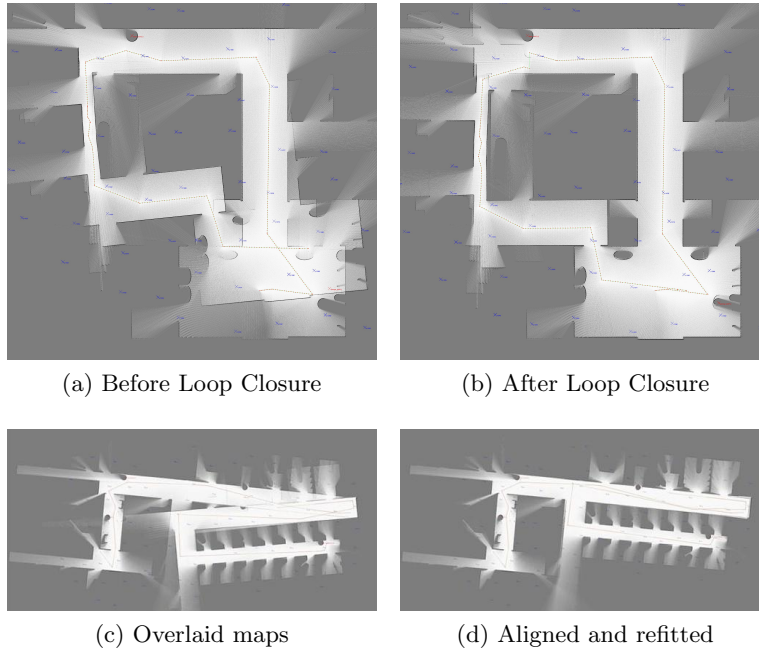


Fig. 1: Loop Closing and Map Merging

explore the same area twice or may take less efficient routes through the environment if maps are not merged immediately. However, map integrity is never at risk. This is interesting for Urban Search and Rescue as deferring these expensive operations saves time that can be used to further explore the emergency site. It is an open research question to decide when, and if, such operations should take place.

During the RoboCup Rescue Virtual Robots 2006 competition, and preliminary trials in the lab, our method was able to map a simulated area with up to 8 concurrently running robots. This high scalability in the multi-robot setting is mainly due to the above mentioned trivial map updates and deferrability of costly mapping operations.

5 Results

In this section, we show sample maps generated by our presented method for both simulated and real data. Simulated data was taken from the USARSim simulator used in the RoboCup Rescue Virtual Robots competition in 2006 [8–10]. The real robot data comes from a publicly available data set from the IROS 2006 Workshop “From sensors to human spacial concepts” [18].

5.1 RoboCup Rescue Virtual Robots Competition 2006

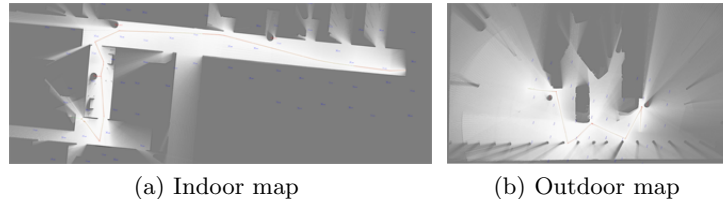


Fig. 2: Some maps produced during the competition.

During the competition the maps provided by the participating teams were judged on their quality. The quality score had two components, a metric quality which was scored automatically and a topological quality which was assessed manually. The basis for the topological quality was how well the map corresponded to the actual environment, supplemented with bonuses for the utility of the map for a first responder. Utility increased with the ability for a person to determine which areas had been searched, where hazards may be located, and where victims were found. The more additional information was depicted on the map, the better the utility.

During the three qualification rounds (denoted by Q1-Q3) we managed to qualify for the semi-finals (denoted with Semi1 and Semi2). In these last two rounds our team of robots produced the maps displayed in Figure 2. A jury assessed the topological quality of our maps. In the competition, we achieved the maximum quality score by displaying very accurate and highly detailed maps.

The metric map quality was based on how well some artificial landmarks were localized. Our method achieved a root mean square (RMS) localization error of $0.2m$ and $0.02m$.

Subsequently, our team received the Best Mapping Award from the Virtual Robots competition as a special recognition of the high quality of our maps.

5.2 Results on Real World Data

The Cogniron data set [18] has been published on the Radish website [19]. The data-set has been acquired using a Nomad Scout robot in a home environment. The maps produced by our algorithm for this data-set are based exclusively on the laser range data, which were recorded using a SICK LMS-200 laser scanner.

Two maps generated from this data-set are shown in Figure 3. The odometry information was not used when constructing the maps. It should also be noted that our loop closure algorithm only works with landmark observations that could be acquired in the USARSim simulator. We have not implemented any detection algorithm for other landmarks or loops in general, so the maps presented were generated without explicitly closing any loops.

The Cognirion data set also includes the sensor logs for a run in which the robot traversed three loops. Despite these loops, and still without using odometry data or explicit closing loops, a highly consistent map is produced which is of comparable quality to the one of the former run.

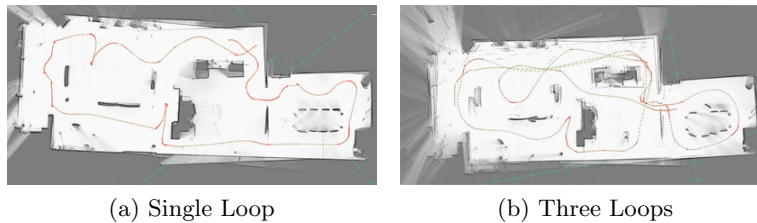


Fig. 3: Corrected maps of the Cognirion data-set. The maps show very high detail by preserving small obstacles such as chairs and vases.

6 Conclusion and Future Work

We presented a hybrid SLAM approach that combines grid-based and topological maps. Our approach thereby exhibits the best of both: Highly detailed maps are learned without limiting scalability to many robots.

Our system was used in the computer-simulated emergency sites of the Rescue Virtual Robot competition at the RoboCup World Championships of 2006, where it supported teams of up to 8 robots. The maps produced by our system earned the Best Mapping Award at this competition. Our experiments with real data show that these results carry over to real-world situations.

The presented work can be intuitively extended into 3D in the future. The graph structure, as well as all other algorithmic details presented, would remain unchanged, only a 3D scan matcher is needed.

Further research should investigate how current exploration algorithms can be adapted to work well with graph-based maps. This includes efficient ways to compute important information like frontiers [20]. Explicitly including meeting points in the plans to facilitate the active detection of overlapping regions and loops would significantly improve the resulting combined maps. Existing algorithms for grid maps might still apply with slight modifications.

References

1. Thrun, S.: Robotic Mapping: A Survey. In Lakemeyer, G., Nebel, B., eds.: Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann (2002)
2. Moravec, H.: Sensor fusion in certainty grids for mobile robots. *AI Magazine* **9** (1988) 61–74

3. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* **99(1)** (1998) 21–71
4. Tomatis, N., Nourbakhsh, I., Siegwart, R.: Hybrid simultaneous localization and map building: Closing the loop with multi-hypotheses tracking. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington DC, USA, May 11 - 15. (2002)
5. Bosse, M., Newman, P., Leonard, J., Teller, S.: An Atlas Framework for Scalable Mapping. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. (2003)
6. Lisien, B., Morales, D., Silver, D., Kantor, G., Rekleitis, I., Choset, H.: The hierarchical atlas. *IEEE Transactions on Robotics and Automation* **21** (2005) 473–481
7. Howard, A., Sukhatme, G.S., Matarić, M.J.: Multi-robot mapping using manifold representations. *Proceedings of the IEEE - Special Issue on Multi-robot Systems* (2006)
8. Balakirsky, S., Scrapper, C., Carpin, S., Lewis, M.: Usarsim: providing a framework for multi-robot performance evaluation. In: *Proceedings of PerMIS 2006*. (2006)
9. Carpin, S., Wang, J., Lewis, M., Birk, A., Jacoff, A.: High fidelity tools for rescue robotics: Results and perspectives. In: *Proceedings of the 2005 RoboCup Symposium*. (2005)
10. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: Bridging the gap between simulation and reality in urban search and rescue. In: *RoboCup 2006: Robot Soccer World Cup X*. LNAI, Springer (2006)
11. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: Fastslam: A factored solution to the simultaneous localization and mapping problem. In: *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, AAAI (2002)
12. Grisetti, G., Stachniss, C., Burgard, W.: Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transaction on Robotics* **23(1)** (2007) 34–46
13. Lu, F., Milios, E.: Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots* **4** (1997) 333–349
14. Pfister, S.T., Kriechbaum, K.L., Roumeliotis, S.I., Burdick, J.W.: A Weighted Range Sensor Matching Algorithm for Mobile Robot Displacement Estimation. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. (2002)
15. Lu, F., Milios, E.: Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems* **18** (1997) 249–275
16. Slamet, B., Pflingsthor, M.: ManifoldSLAM: a Multi-Agent Simultaneous Localization and Mapping System for the RoboCup Rescue Virtual Robots Competition. Master's thesis, Universiteit van Amsterdam (December 2006)
17. Olson, E., Leonard, J., Teller, S.: Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. (2006) 2262–2269
18. Zivkovic, Z.: *IEEE/RSJ IROS 2006 Workshop: 'From sensors to human spatial concepts'* (2006)
19. Radish: The robotics data set repository (-)
20. Yamauchi, B.: A frontier based approach for autonomous exploration. In: *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, July 10-11, 1997. (1997)