Benchmarking the pose accuracy of different SLAM approaches for rescue robotics

Markus Kühn, Rainer Koch, Martin Fees and Stefan May
Faculty of Electrical Engineering, Precision Engineering, Information Technology
Technische Hochschule Nuremberg Georg Simon Ohm
Nuremberg, Germany

Abstract*—Simultaneous Localization and Mapping (SLAM) is essential for a mobile robot. Localizing itself and obtaining information about the environment qualifies the robot to interact with it. For this reason multiple approaches for SLAM exist in the robotics community. For example in the RoboCup Rescue Challenge most teams use the `hector_slam`, `gmapping` or `crsm_slam` approach. Hence it is essential to estimate the robot's pose and map the surrounding environment accurately, the aim of this paper is to describe a benchmark for the objective comparison of the pose accuracy of different approaches. Usually visual inspections of maps is used for this. Furthermore, the first results of this benchmark are outlined in this paper to show that the benchmark is working for the first tests. Additionally, we want to estimate if our approaches `ohm_tsd_slam` is able to compete with the other algorithms. The results show that this is the case for the used dataset. `Ohm_tsd_slam` achieves a better pose accuracy than the `crsm_slam` and is close to the accuracy of `hector_slam`. The `gmapping` approach was not compared as no inertial measurement unit (IMU) was available in the used simulator.

I. Introduction

Every year different robotics competitions are conducted to show the progress of research in mobile robotics. This includes challenges organized by the DARPA [1] or different regional RoboCup competitions and the RoboCup world championships [2]. At RoboCup Rescue competitions teams are able to compare their progress in a challenging arena which simulates a disaster scenario (Figure 1). Here, the main task is to localize victims by detecting vital signs like body heat, motion or breath and mark the location of these victims in a map. Therefore the rescue robot needs to localize itself in the arena and creates a map of it. This problem is usually solved with a Simultaneous Localization and Mapping (SLAM) approach. Such an algorithm uses laser scanners, stereo cameras or 3D cameras to estimate the robot's pose. In our case, the pose describes the position of the robot on a 2D plane and its orientation on this plane. For rescue missions this is reasonable as a 2D map is an intuitive representation for rescuers. The pose is estimated relatively to the origin the mission was started and a map is created with the information from the sensor at the same time. With an increasing focus on searching for and detecting victims autonomously, creating a map and estimating the robots pose gets more important. The reason for this is that tasks for autonomous operation like environment exploration and robot navigation depend on the information from SLAM.

The RoboCup Rescue is a great platform to compare a teams progress in development with other teams annually. Nevertheless, we figured out that it is hard compare different SLAM approaches directly at the competition. Although SLAM is essential at the competition, it is not possible to significantly score points with an accurate SLAM approach. Accordingly the competition results do not represent the quality of the SLAM approaches. However it shows if a SLAM approach worked or not.

For this reason, the aim of this paper is to describe a simple benchmark to compare different SLAM approaches to a given ground truth (or with each other). Therefore a simple error metric according to Burgard et al. [3] was adapted and the first results are proposed in this paper to prove the concept. In particular the benchmark tool is created for the Robot Operating System (ROS) [4]. As far as we know, every team competing at the RoboCup Rescue uses ROS and the analyzed SLAM approaches are available as ROS packages. In detail the approaches `hector_slam`, `crsm_slam` and our own approach `ohm_tsd_slam` were analyzed with the benchmark tool.

Accordingly section II gives an overview about the different SLAM approaches. Furthermore related work on benchmarking SLAM approaches in robotics is outlined in this section, too. Afterwards the important aspects of the benchmark tool are outlined in section III. Finally, the first results for the benchmark tool are summarized in section IV.

II. Related Work

At first, it is necessary to mention that all of the analyzed SLAM approaches use a laser scanner to estimate the robot's pose and obtain a 2D map of the environment. This laser scanner usually scans the environment with a high frequency while covering a huge field of view (e.g. 270°). An example for such a laser scanner is the Hokuyo UTM-30LX which is commonly used at the RoboCup Rescue.

Fig. 1: RoboCup Rescue arena at the RoboCup German Open 2013

Although all approaches use laser scanning, the approaches differ in the way they obtain the map and pose:

- `Hector_slam` was developed by Kohlbrecher et al. [5] and is used by most of the RoboCup Rescue teams. It uses map gradients for pose estimation and leverages the high update rate of the laser scanner. As the results are accurate, it does not use a loop closing algorithm to correct the map. Still an IMU can be used for a more accurate position estimation and mapping.

- `Gmapping` is optimized for long range laser scanners and does SLAM with a Rao-Blackwellized Particle Filter [6], [7]. In comparison to `hector_slam` the update rate of the robot pose is much slower. For this reason it is necessary to use an IMU with this solution to get good results.

- `Crsm_slam` computes the robot's pose globally with a scan matching between obtained laser scans and the complete map. It only uses a laser scanner and works without an IMU. Additionally, it filters the laser scans for critical information for a faster pose estimation. The algorithms used for this approach are proposed by Tsardoulis and Petrou [8].

- `ohm_tsd_slam` was developed by the Team AutonOHM and generalizes the KinectFusion approach [9]. It uses only a laser scanner and needs no additional IMU. Furthermore the representation as a truncated signed distance (TSD) grid is drift-reduced and hence no loop closing is necessary [10]. For pose estimation it uses the widespread Iterative Closest Point (ICP) algorithm [11] that can be combined with a Random Sample and Consensus (RANSAC) based matching [12] for enhanced robustness.

Second, discussions on benchmarking SLAM approaches were conducted between 2005 and 2010 at workshops of major robotics conferences like the ICRA and IROS. General methodologies to evaluate SLAM approaches were described by Amigioni et al. [13] to give an alternative to the common visual inspection for robot mapping. Further, the logical separation of a benchmark into a problem and a solution was formalized by Fontana et al. [14]. In particular important aspects like the usage of ground truth information and the parametrization of SLAM approaches were outlined. Furthermore, the Hausdorff metric is suggested for comparing the quality of maps [13], [15]. On the other hand, the comparison of relative pose changes is a metric for benchmarking the accuracy of the robot's pose. This is better than using absolute robot poses for evaluation because the relative error does not accumulate an error over time [3], [15], [16]. Another important point is that noise in a SLAM context usually is Gaussian [16]. Hence simulators modelling Gaussian noise are sufficient for comparing different approaches most of the time.

Third, the aspect of distributing datasets for the comparison of different projects is a key factor for all benchmarking solutions. Especially the RAWSEEDS project [14] or websites like "SLAM Benchmarking" [17] or "The robotics data set repository (radish)" [18] provide datasets that are commonly used to test and benchmark progress in SLAM research.

This paper builds on this expertise in the community to create a simple benchmarking tool for the RoboCup Rescue community. In the future this will enable the teams to compare different SLAM approaches.

III. The Benchmarking Tool

This section outlines the important aspects of the benchmarking tool. Therefore we follow the logical separation of a benchmark into a problem and solution according to [14].

In this case the benchmark problem describes the task interesting for evaluation and a metric to compare the results. Further, the different SLAM approaches are the benchmark solutions. Both parts are considered by the developed benchmark tool and results are generated by using different benchmark solutions to solve the same benchmark problem.

A. Benchmarking Problem and Metric

For the formulation of the benchmarking problem three parts are necessary:

First of all, the task of the benchmark is to analyze the accuracy of the robot's pose estimation. Although this means the resulting map is not analyzed directly, we follow the statement that creating a map becomes easy as soon we have a good estimate of the robots pose [3].

Moreover, the datasets used for the benchmark are the second part of the benchmark problem. In this case, the benchmark is conducted on a dataset obtained with the `stdr_simulator` that is available for ROS. The dataset was created while the robot was driving around in the simulator in a RoboCup Rescue like scenario for about five minutes. The ground truth of the robot and the data from the laser scanner were recorded during this time. This makes it possible to use the exact same dataset for the evaluation of every SLAM approach. The dataset was recorded with a emulated laser scanner with the characteristics of a Hokuyo UTM-30LX

Third, it is necessary to describe the metric of the benchmark. Therefore a pose estimate is formulated as the vector $\vec{p}_t = (x \quad y \quad \theta)^{\mathrm{T}}$ which contains the robot's movement in the x- and y-direction in the plane and the orientation as a rotation angle around the z-axis for the moment t. The same information is provided by the ground truth vector $\vec{g}_t$. With this information we can calculate the relative error between two moment moments in time, similar to Burgard et al. [3]:

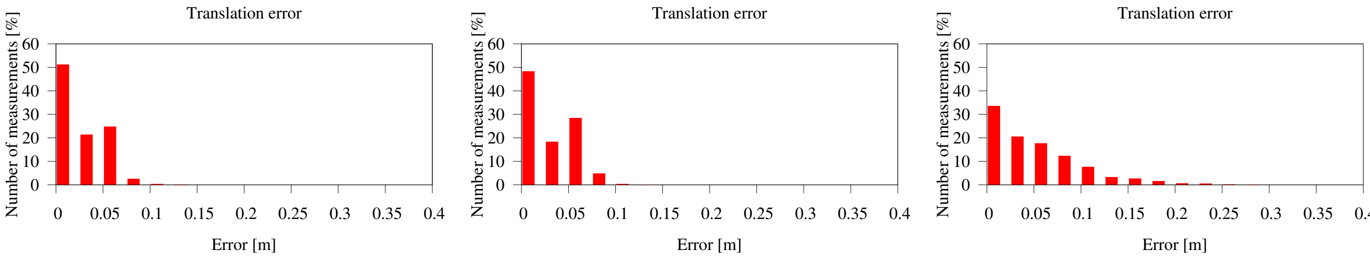
$$\Delta\vec{p} = \vec{p}_{t-1} - \vec{p}_t \tag{1}$$

$$\Delta\vec{g} = \vec{g}_{t-1} - \vec{g}_t \tag{2}$$

$$\epsilon = ||\Delta\vec{g} - \Delta\vec{p}|| \tag{3}$$

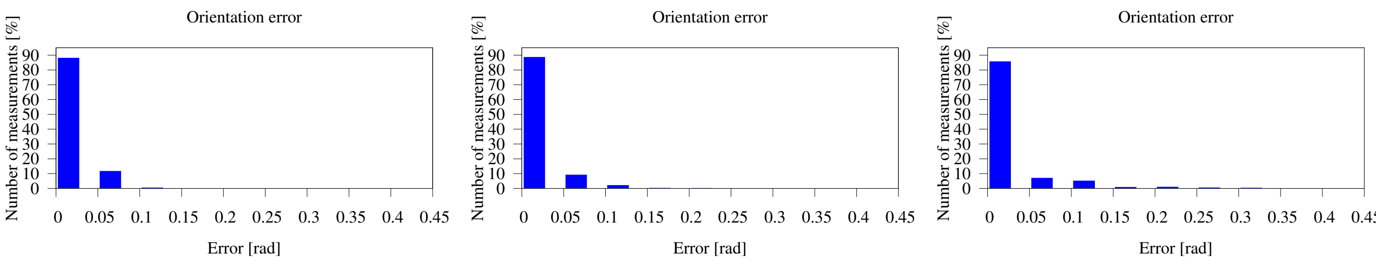
$\Delta\vec{p}$ and $\Delta\vec{g}$ describe the change for the pose estimate and the ground truth. The relative error between two estimates is described as the absolute difference between $\Delta\vec{g}$ and $\Delta\vec{p}$, denoted as the relative error ϵ between two pose updates. After this, the error in orientation and translation is provided separately.

At this point it is very important to think about the update frequency of the pose estimate and the ground truth. Depending



(a) Result of `hector_slam` (b) Result of `ohm_tsd_slam` (c) Result of `crsm_slam`

Fig. 2: Relative error for the translation over time.



(a) Result of `hector_slam` (b) Result of `ohm_tsd_slam` (c) Result of `crsm_slam`

Fig. 3: Relative error for the orientation over time.

on the algorithm the robot's pose is updated with a frequency between 5 Hz and 50 Hz. The benchmark deals with these differences by using the actual ground truth information of the moment the pose information was updated. ROS is very helpful at this point as the `tf` package tracks the ground truth and pose information for 15 seconds into the past by default. This is important to avoid problems due to scheduling issues on the computer. Because the benchmark tool does not need to process pose updates in real time. Nevertheless it is necessary to provide the ground truth with a higher frequency as this information is interpolated within two updates of the ground truth. The reason for this is that the update of the robot's pose is not synchronized with the update of the ground truth. Only for a high update frequency relative errors due to timing issues have a small influence on the benchmark results.

B. Benchmark Solutions

All SLAM approaches that can provide the robot's pose for the described benchmark problem are possible benchmark solutions. In our case, ROS and its `tf` package are a key factor for the approaches that were introduced in section II. `Tf` provides an interface which standardizes the output of the algorithms and simplifies the development of the benchmarking tool. `Hector_slam`, `gmapping`, `crsm_slam` and `ohm_tsd_slam` provide the pose information in the same way. All use a transformation matrix between a local map coordinate system and a dynamic robot coordinate system. The ground truth information from the benchmark problem is also provided as a transformation between the same local map frame and a second robot coordinate system. This last information is provided by the dataset.

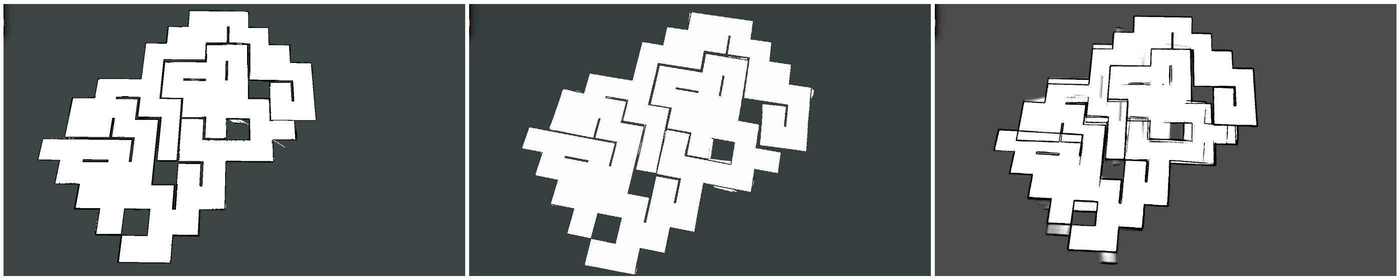
With the pose information from a SLAM approach and the dataset, the benchmark tool can calculate the error according to the metric in equations 1, 2 and 3. The relative errors for every update of the robot pose are calculated while the dataset is played back. Afterwards, all errors are exported in a ".csv"-file and the overall error statistics like in Table I are calculated. Now, this information is used to create diagrams for the comparison of the different benchmark solutions / SLAM approaches.

IV. Experimental Results

Due to the lack of an IMU in the simulator all approaches except of `gmapping` were benchmarked as explained in section III. While the dataset was played back, a single approach computed the pose information. All benchmarks ran on an Intel Core i5-2410M CPU.

The relative pose error is seperated into the error in translation and orientation. First, the errors are visualized as histograms with different error classes for translation and orientation. Figure 2 shows the translational error for the dataset for all three approaches. Each block summarizes an error of 0.025 m width. Furthermore, the error for the orientation is displayed in Figure 3. Here, every block has a width of 0.05 radian (2.86 degree). Additionally, the error statistics for the different approaches are outlined in Table I and II.

`Hector_slam` achieves the best results of all approaches. For the leftmost error class the relative frequency is the highest for both, the translation and orientation. This means



(a) Map created with `hector_slam`. (b) Map created with `ohm_tsd_slam`. (c) Map created with `crsm_slam`.

Fig. 4: Maps created for the simulated RoboCup Rescue dataset created with the `stdr_simulator`.

`hector_slam` estimates the pose with an maximum error of 0.0025 m in over 50 percent of all measurements. The `ohm_tsd_slam` achieves results with a similar quality. For example, the translational maximum error of both approaches differs only by about two millimetres as depicted in Table I. Nevertheless, Figure 2 shows that the `ohm_tsd_slam` is not always as good as `hector_slam`. This is displayed by a higher relative frequency for error classes representing higher errors. Additionally, it is pointed out by the mean and standard deviation values in Table I which are higher for the `ohm_tsd_slam`. For the orientation error the histograms show mostly equal results. The statistics in Table II show this to. The mean value differs only by 0.003 radian. Anyway, the maximum error is two times higher than the maximum error of `hector_slam`.

In comparison to these algorithms, the `crsm_slam` achieves results with lower accuracy. For the translational error, the error distribution is wider than the error distributions of the other approaches. This is the same for the orientation but the effect is much smaller in this case. The error statistics in Tables I and II show there still is a significant difference.

V. Conclusion

The aim of this paper was to describe a simple benchmark and initially compare frequently used SLAM approaches of the RoboCup Rescue community.

Summarizing the benchmark results for the different algorithms, `hector_slam` achieves the highest pose accuracy. In comparison, `ohm_tsd_slam` performs worse. Nevertheless, it has a higher pose accuracy than `crsm_slam` which has the worst results.

Approach	**Mean**	**Deviation**	**Min**	**Max**
hector_slam	0.025	0.024	~ 0	0.147
ohm_tsd_slam	0.031	0.028	~ 0	0.148
crsm_slam	0.054	0.039	~ 0	0.280

TABLE I: Statistics for the relative error in translation [m]

Approach	**Mean**	**Deviation**	**Min**	**Max**
hector_slam	0.009	0.015	~ 0	0.103
ohm_tsd_slam	0.012	0.019	~ 0	0.202
crsm_slam	0.023	0.028	~ 0	0.333

TABLE II: Statistics for the relative error in orientation [rad]

If we compare our experience from different RoboCup Rescue competitions and visual inspection these results are reasonable. The maps from the different algorithms in Figure 4 show this too. Especially if we compare the map obtained by `hector_slam` to the map from `crsm_slam`, we can see inaccuracies in the third map. These inaccuracies of mapping the walls are related to the smaller accuracies in pose estimation. Another fact that underlines the quality of `hector_slam` is that the RoboCup Rescue community already uses this approach extensively.

This leads to two key results of this paper. First, our `ohm_tsd_slam` approach is able to compete with *hector_slam*. Second, visual inspection and our experience resemble the results of the benchmark tool. The benchmark worked for this first tests.

Nevertheless, the benchmark has to be used and tested in the future to proof its correctness. Furthermore it will be extended by other metrics to rate the pose accuracy in detail. This possibly includes metrics for map quality, too. Finally, this will help to increase the reliability and comparability of different SLAM approaches.

References

[1] "DARPA - Defense Advanced Research Projects Agency." [Online]. Available: http://www.darpa.mil/

[2] "The RoboCup Federation." [Online]. Available: http://www.robocup.org/

[3] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kummerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardos, "A comparison of SLAM algorithms based on a graph of relations," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS*, 2009, pp. 2089–2095.

[4] M. Quigley, K. Conley, B. P. Gerkey, J. FAust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Mg, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[5] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," *9th IEEE International Symposium on Safety, Security, and Rescue Robotics. SSRR*, pp. 155–160, 2011.

[6] G. Grisetti, C. Stachniss, and W. Burgard, "Improving Grid Based SLAM with Rao Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in *Proc. IEEE International Conference on Robotics and Automation. ICRA*, 2005, pp. 2432–2437.

[7] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–36, 2007.

[8] E. Tsardoulias and L. Petrou, “Critical rays scan match SLAM,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 72, no. 3-4, pp. 441–462, 2013.

[9] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, “KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera,” in *Proc. 24th ACM Symposium on User Interface Software and Technology. UIST*, 2011.

[10] S. May, R. Koch, P. Koch, R. Koch, C. Merkl, C. Pfitzner, and A. Nüchter, “A Generalized 2D and 3D Multi-Sensor Data Integration Approach based on Signed Distance Functions for Multi-Modal Robotic Mapping,” in *19th International Workshop on Vision, Modeling and Visualization . VMV 2014*, Darmstadt, Germany 2014, 2014.

[11] P. J. Besl and N. D. McKay, “A Method for Registration of 3D-Shapes,” *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 14, no. 2, pp. 239–256, 1992.

[12] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Communications of the ACM*, vol. 24, pp. 381 – 395, 1981.

[13] F. Amigoni, S. Gasparini, and M. Gini, “Good Experimental Methodologies for Robotic Mapping: A Proposal,” in *Proc. IEEE International Conference on Robotics and Automation. ICRA*, 2007, pp. 4176 – 4181.

[14] G. Fontana, M. Matteucci, and D. G. Sorrenti, “The RAWSEED proposal for representation-independent benchmarking of SLAM,” 2008. [Online]. Available: http://rawseeds.eridia.it/home/wp-content/uploads/2008/12/rss08tasklevelbenchmarking.pdf

[15] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, “On measuring the accuracy of SLAM algorithms,” *Journal of Autonomous Robots*, vol. 27, no. 4, pp. 387–407, 2009.

[16] B. Balaguer, S. Carpin, and S. Balakrisky, “Towards Quantitative Comparisons of Robot Algorithms: Experiences with SLAM in Simulation and Real World Systems,” in *Proc. Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems. IROS*, 2007.

[17] A. Kleiner, B. Steder, C. Dornhege, C. Stachniss, G. Grisetti, M. Ruhnke, R. Kümmerle, and W. Burgard, “slam benchmarking.” [Online]. Available: http://kaspar.informatik.uni-freiburg.de/~slamEvaluation

[18] A. Howard and N. Roy, “The robotics data set repository (radish).” [Online]. Available: http://radish.sourceforge.net/