Trajectory-based Comparison of SLAM Algorithms

Wolfram Burgard Cyrill Stachniss Giorgio Grisetti Bastian Steder
Rainer Kümmerle Christian Dornhege Michael Ruhnke Alexander Kleiner Juan D. Tardós

***Abstract*— In this paper, we address the problem of creating an objective benchmark for comparing SLAM approaches. We propose a framework for analyzing the results of SLAM approaches based on a metric for measuring the error of the corrected trajectory. The metric uses only relative relations between poses and does not rely on a global reference frame. The idea is related to graph-based SLAM approaches, namely to consider the energy that is needed to deform the trajectory estimated by a SLAM approach into the ground truth trajectory. Our method enables us to compare SLAM approaches that use different estimation techniques or different sensor modalities since all computations are made based on the corrected trajectory of the robot. We provide sets of relative relations needed to compute our metric for an extensive set of datasets frequently used in the SLAM community. The relations have been obtained by manually matching laser-range observations to avoid the errors caused by matching algorithms. Our benchmark framework allows the user an easy analysis and objective comparisons between different SLAM approaches.**

I. Introduction

Models of the environment are needed for a wide range of robotic applications including transportation tasks, guidance, and search and rescue. Learning maps has therefore been a major research focus in the robotics community over the last decades. In the literature, the mobile robot mapping problem under pose uncertainty is often referred to as the *simultaneous localization and mapping* (SLAM) or *concurrent mapping and localization* (CML) problem [26]. SLAM is considered to be a complex problem because to localize itself a robot needs a consistent map and for acquiring the map the robot requires a good estimate of its location.

Whereas dozens of different techniques to tackle the SLAM problem have been proposed, there is no gold standard for comparing the results of different SLAM algorithms. In the community of feature-based estimation techniques, researchers often measure the Euclidean or Mahalanobis distance between the estimated landmark location and the true location (if this information is available). As we will illustrate in this paper, comparing results based on an absolute reference frame can have shortcomings. In the area of grid-based estimation techniques, people often use visual inspection to compare maps or overlays with blueprints of buildings. This kind of evaluation becomes more and more difficult as new SLAM approaches show increasing capabilities and thus large scale environments are needed for evaluation. There is a strong need for methods allowing meaningful comparisons of different approaches. Ideally, such a method is capable of performing comparisons between mapping systems that apply different estimation techniques and operate on different sensing modalities. We argue that meaningful comparisons between different SLAM approaches require to have a common performance metric. This metric should allow to compare the outcome of different mapping approaches when applying them on the same dataset.

In this paper, we propose a novel technique for comparing the output of SLAM algorithms. It is based on an idea that is actually similar to the concept of the graph-based SLAM approaches [19], [12], [22]. It uses the energy that is (virtually) needed to deform the trajectory estimated by a SLAM approach into the ground truth trajectory as a quality measure.

We propose a metric that operates only on relative geometric relations between poses along the trajectory of the robot. This is inspired by the fact used in most Rao-Blackwellized particle filter approaches, namely that estimating the map becomes trivial given the robot's trajectory [10], [20], [25]. Our metric enables a user to establish a benchmark for objectively comparing the performance of a mapping system to existing approaches. Our approach allows for making (approximative) comparisons between algorithms even if a perfect ground truth information is not available. This enables us to present benchmarks based on frequently used datasets in the robotics community such as the MIT Killian Court, or the Intel Research Lab dataset. The disadvantage of our method is that it requires manual work to be carried out by a human that knows the topology of the environment. The manual work, however, has to be done only once for a dataset and then allows other researchers to evaluate their methods with low effort. Together with this paper, we provide a web page that hosts such manually matched relations for existing log files [17]. We furthermore provide evaluations for the results of three different mapping techniques, namely scan-matching, SLAM using Rao-Blackwellized particle filter [10], and a maximum likelihood SLAM approach based on the graph formulation [11], [21].

II. Related Work

Learning maps is a frequently studied problem in the robotics literature. SLAM techniques for mobile robots can

All authors are with the University of Freiburg, Dept. of Computer Science, Georges Koehler Allee 79, 79110 Freiburg, Germany except of Juan D. Tardós who is with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, María de Luna 1, E-50018, Zaragoza, Spain.

The authors gratefully thank Mike Bosse, Patrick Beeson, and Dirk Haehnel for providing the MIT Killian Court, the ACES, and the Intel Research Lab datasets. This work has partly been supported by the DFG under contract number SFB/TR-8 and the European Commission under contract numbers FP6-2005-IST-6-RAWSEEDS, FP6-IST-045388-INDIGO, and FP7-231888-EUROPA.

be classified according to the underlying estimation technique. The most popular approaches are extended Kalman filters (EKFs) [18], [23] and its variants [15], sparse extended information filters [8], [28], particle filters [20], [10], least square error minimization approaches [19], [12], [22] and several others. For some applications, it might even be sufficient to learn local maps only [13], [27], [30].

The approach of finding maximum likelihood maps using a graph or network of constraints is strongly related to our approach for evaluating SLAM methods presented in this paper. Lu and Milios [19] introduced the concept of graph-based or network-based SLAM using a kind of brute force method for optimization. Gutmann and Konolige [12] proposed an effective way for constructing such a network and for detecting loop closures while running an incremental estimation algorithm. Olson *et al.* [22] presented an optimization approach that applies stochastic gradient descent for resolving relations in a network efficiently.

Activities related to performance metrics for SLAM methods, as the work described in this paper, can roughly be divided into three major categories: First, competitions where robot systems are competing within a defined problem scenario (such as playing soccer), second, collections of publicly available datasets that are provided for comparing algorithms on specific problem, and third, related publications that introduce methodologies and scoring metrics for comparing different methods.

To perform comparisons between robots, numerous robot competitions have been initiated in the past, evaluating the performance of cleaning robots [6], robots in simulated Mars environments at the ESA Lunar Robot Challenge[7], robots playing soccer or rescuing victims after a disaster at RoboCup, and cars driving autonomously at the DARPA Urban Challenge. However, competition settings are likely to generate additional noise due to differing hardware and software settings. Depending on the competition, approaches are often tuned to the settings addressed in the competitions.

In the robotics community, there exist some well-known web sites providing datasets such as Radish [14] or [3] and algorithms [24] for mapping. However, they neither provide ground truth data nor recommendations on how to compare different maps in a meaningful way. Zivkovic *e*t al. [31] provide a labeled dataset containing information useful for human-robot interaction and Frese [9] provides a dataset of the DLR building with manually obtained ground truth data associations.

Some steps towards benchmarking navigation solutions have been presented in the past. Amigoni *e*t al. [1] presented a general methodology for performing experimental activities in the area of robotic mapping. They suggested a number of issues that should be addressed when experimentally validating a mapping method. If ground truth data is available, they suggest to utilize the Hausdorff metric for map comparison.

Wolf *e*t al. [29] proposed the idea of using manually supervised Monte Carlo Localization (MCL) for matching 3D scans against a reference map. They suggested to generate the reference maps from independently created CAD data, which can be obtained from the land registry office. The comparison between the generated map and the ground truth has been carried out by computing the Euclidean distance and angle difference of each scan, and plotting these over time. We argue here that comparing the absolute error between two tracks might not yield a meaningful assertion.

Balaguer *e*t al. [2] utilize the USARSim robot simulator and a real robot platform for comparing different open source SLAM approaches and they propose that the simulator engine could be used for systematically benchmarking different approaches of SLAM. However, it has also been shown that noise is often but not always Gaussian in the SLAM context [25]. Gaussian noise, however, is typically used in most simulation systems. In addition to that, Balaguer *et al.* do not provide a quantitative metric for comparing generated maps with ground truth. As many other approaches, their comparisons were carried out by visual inspection.

III. Metric for Benchmarking SLAM Algorithms

We propose a metric for measuring the performance of a SLAM algorithm not by comparing the map itself but by considering the poses of the robot during data acquisition. In this way, we gain two important properties: First, it allows us to compare the result of algorithms that generate different types of metric map representations, such as feature-maps or occupancy grid maps. Second, the method is invariant to the sensor setup of the robot. Thus, a result of a graph-based SLAM approach working on laser range data can be compared, for example, with the result of vision-based FastSLAM. The only property we require is that the SLAM algorithm estimates the trajectory of the robot given by a set of poses. All benchmark computations will be performed on this set.

A. Our Metric

Let $x_{1:T}$ be the poses of the robot estimated by a SLAM algorithm from time step 1 to T, $x_t \in SE(2)$ or $SE(3)$. Let $x^*_{1:T}$ be the reference poses of the robot during mapping, ideally the true poses. A straightforward error metric could be defined as

$$\varepsilon(x_{1:T}) \quad = \quad \sum_{t=1}^{T} (x_t \ominus x_t^*)^2, \tag{1}$$

where $\oplus$ is the standard motion composition operator and $\ominus$ its inverse as defined by Lu and Milios [19] or its analogous definition for $SE(3)$, respectively. Thus, $\delta_{i,j} = x_j \ominus x_i$ is the relative transformation that moves the node x_i onto x_j. Let $\delta^*_{i,j}$ be the transformation based on x^*_i and x^*_j accordingly. Eq. 1 can be rewritten as

$$\begin{aligned} \varepsilon(x_{1:T}) \quad = \quad & \sum_{t=1}^{T} \big((x_1 \oplus \delta_{1,2} \oplus \ldots \oplus \delta_{t-1,t}) \\ & \ominus (x_1^* \oplus \delta^*_{1,2} \oplus \ldots \oplus \delta^*_{t-1,t}) \big)^2 . \end{aligned} \tag{2}$$

We claim that this metric is suboptimal for comparing the result of a SLAM algorithm. To illustrate this, consider Figure 1. Here, a robot travels along a straight line. Let the robot make perfect pose estimates in general but a rotational

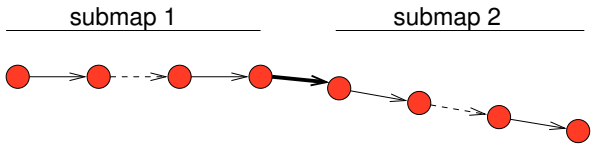

Fig. 1. This figure illustrates a simple example where the metric in Eq. 1 is suboptimal. Consider the robot moves along a straight line and after n poses, it makes a small angular error (bold arrow) but then continues without any further error. Both parts (labeled submap 1 and submap 2) are perfectly mapped and only the connection between both submaps contains an error. According to Eq. 1, the error of this estimates increases with every node added to submap 2 although the submap itself is perfectly estimated. Thus, the error depends on the point in time where the robot made an estimation error without considering that it might not introduce any (further) error.

error somewhere along the line, let us say in the middle. Both submaps (before and after the estimation error) are perfectly mapped. According to Eq. 1, the error of this estimate increases with every node that is added to submap 2 although the submap itself is perfectly estimated. Thus, the error depends on the point in time where the robot made an estimation error without considering that it might not introduce any (further) error. The reason for this is the fact that the metric in Eq. 1 operates on global coordinates and considers the trajectory and thus the map as a rigid body that has to be aligned with the ground truth.

In this paper, we propose to use a metric that considers the deformation energy that is needed to transfer the estimate into the ground truth. This can be done – similar to the ideas of the graph mapping introduced by Lu and Milios [19] – by considering the poses as masses and connections between them as springs. Thus, our metric is based on the *relative* displacement between poses. Instead of comparing x to x^* (in the global reference frame), we do the operation based on δ and δ^* as

$$\varepsilon(\delta) = \frac{1}{N}\sum_{i,j}(\delta_{i,j} \ominus \delta^*_{i,j})^2 \quad (3)$$

$$= \frac{1}{N}\sum_{i,j} trans(\delta_{i,j} \ominus \delta^*_{i,j})^2 + rot(\delta_{i,j} \ominus \delta^*_{i,j})^2, \quad (4)$$

where N is the number of relative relations and $trans(\cdot)$ and $rot(\cdot)$ are used to separate the translational and rotational components. We suggest to provide both quantities individually. In this case, the error (or transformation energy) in the above-mentioned example will be consistently estimated as the single rotational error no matter where the error occurs in the space or in which order the data is processed. Note that this score is a metric since it satisfies the four properties of non-negativity, identity of indiscernibles, symmetry, and triangle inequality.

Our error metric, however, leaves open which relative displacements $\delta_{j,i}$ are included in the summation in Eq. 4. Evaluating two approaches based on a different set of relative pose displacements will obviously result in two different scores. As we will show in the remainder of this section, the set δ (and thus δ^*) can be defined to highlight certain properties of an algorithm.

Note that some researchers prefer the absolute error (absolute value, not squared) instead of the squared one. We prefer the squared one since it comes from the motivation that the metric measures the energy needed to transform the estimated trajectory into ground truth. However, one can also use the metric using the non-squared error instead of the squared one. In the experimental evaluation, we actually provide both values.

It should be noted that the metric presented here also has drawbacks. First, the metric, as we defined it, only evaluates the mean estimate of the SLAM algorithm and does not consider its estimate of the uncertainty. Second, it misses a probabilistic interpretation as the Fisher information would realize, see, for example, Censi's work [4] on the achievable accuracy for range finder-based localization.

B. Selecting Relative Displacements for Evaluation

Benchmarks are designed to compare different algorithms. In the case of SLAM systems, however, the task the robot finally has to solve should define the required accuracy and this information should be considered in the benchmark.

For example, a robot generating blueprints of buildings should reflect the geometry of a building as accurately as possible. In contrast to that, a robot performing navigation tasks requires a map that can be used to robustly localize itself and to compute valid trajectories to a goal location. To carry out this task, it is sufficient in most cases, that the map is topologically consistent and that its observations can be locally matched to the map. A map having these properties is often referred to as locally consistent.

By selecting the relative displacements $\delta_{j,i}$ used in Eq. 4 for a given dataset, the user can highlight certain properties and thus design a benchmark for evaluating an approach given the application in mind.

For example, by adding only known relative displacements between nearby poses based on visibility, a local consistency is highlighted. In contrast to that, by adding known relative displacements of far away poses, for example, provided by an accurate external measurement device or by background knowledge, the accuracy of the overall geometry of the mapped environment is enforced. In this way, one can incorporate the knowledge into the benchmark that, for example, a corridor has a certain length and is straight.

IV. Obtaining Reference Relations

In practice, the key question regarding Eq. 4 is how to determine the *true relative displacements* between poses. Obviously, the true values are available only in simulation. If ground truth information is available, it is trivial to derive the exact relations. However, we can also determine close-to-true values by using the information recorded by the mobile robot and the background knowledge of the human recording the datasets. This, of course, involves manual work, but from our perspective it is the best method for obtaining such relations if no ground truth is available.

Please note, that the metric proposed above is independent of the actual sensor used. In the remainder of this paper, however, we will concentrate on laser range finders which are probably the most popular sensors in robotics at the moment. To evaluate an approach operating on a different sensor

modality, one has two possibilities. Either one temporarily mounts a laser range finder on the robot (if this is possible) or has to provide a method for accurately determining the relative displacement between two poses from which an observation has been taken that observes the same part of the space.

A. Initial Guess

In our work, we propose the following strategy. First, one seeks for an initial guess about the relative displacement between poses. Based on the knowledge of the human, a wrong initial guess can be easily discarded since the human "knows" the structure of the environment. In a second step, a refinement is proposed based on manual interaction.

In most cases, researchers in robotics will have SLAM algorithms at hand that can be used to compute such an *initial guess*. By manually inspecting the estimates of the algorithm, a human can accept or discard a match. It is important to note that the output is not more than an initial guess and it is used to estimate the visibility constraints which will be used in the next step.

B. Manual Matching Refinement and Scan Rejection

Based on the initial guess about the pose of the robot for a given time step, it is possible to determine which observations in the dataset should have covered the same part of the space or the same objects. For a laser range finder, this can easily be achieved. Between each visible pair of poses, one adds a relative displacement into a candidate set.

In the next step, a human processes the candidate set to eliminate wrong hypotheses by visualizing the observation in a common reference frame. This requires manual interaction but allows for eliminating wrong matches and outliers with high precision. Since we aim to find the best possible relative displacement, we perform a pair-wise registration procedure to refine the estimates of the observation registration method. It furthermore allows the user to manually adjust the relative offset between poses so that the pairs of observations fit perfectly. Alternatively, the pair can be discarded.

This approach might sound work-intensive but with an appropriate user interface, this task can be carried out without a large waste of resources. For example, for a standard dataset with 1700 relations, it took an unexperienced user approximately four hours to extract the relative translations that then served as the input to the error calculation.

C. Adding Additional Relations

In addition to the relative transformations added upon visibility and matching of observations, one can directly incorporate additional relations resulting from other sources of information, for example, given the knowledge about the length of a corridor in an environment. By adding a relation between two poses – each at one side of the corridor – one can easily incorporate knowledge about the global geometry of an environment if this is available and a quality to evaluate. In most cases, however, such information is not available. See [16] for an example using aerial imagery.

V. Benchmarking of Algorithms without Trajectory Estimates

A series of SLAM approaches estimate the trajectory of the robot as well as a map. However, in the context of the EKF, researchers often exclude an estimate of the full trajectory to lower the computational load.

We see two solutions to overcome this problem: (a) depending on the capabilities of the sensor, one can recover the trajectory as a post processing step given the feature locations and the data association estimated by the approach. This procedure could be quite easily realized by a localization run in the built map with given data association (the data association of the SLAM algorithm). (b) in some settings this strategy can be difficult and one might argue that a comparison based on the landmark locations is more desirable. In this case, one can apply our metric as well operating on the landmark locations instead of based on the poses of the robot. In this case, the relations $\delta^*_{i,j}$ can be determined by measuring the relative distances between landmarks using, for example, a highly accurate measurement device and a triangulation based on the landmark locations for defining the relations.

The disadvantage of this approach is that the data association between estimated landmarks and ground truth landmarks is not given. Depending on the kind of observations, a human can manually determine the data association for each observation of an evaluation datasets as done by Frese [9]. This, however, might get intractable for SIFT-like features obtained with high-framerate cameras. Note that all metrics measuring an error based on landmark locations require such a data association as given. Furthermore, it becomes impossible to compare significantly different SLAM systems using different sensing modalities. Therefore, we recommend the first option to evaluate techniques such as the EKF.

VI. Datasets for Benchmarking

To validate the metric, we selected a set of datasets representative for different kinds of environments from the publicly available datasets. We extracted relative relations between robot poses using the methods described in the previous sections by manually validating every single observation between pairs of poses.

As a challenging indoor corridor-environment with a non-trivial topology including nested loops, we selected the MIT Killian Court dataset and the dataset of the ACES building at the University of Texas, Austin. As a typical office environment with a significant level of clutter, we selected the dataset of building 079 at the University of Freiburg, the Intel Research Lab dataset, and a dataset acquired at the CSAIL at MIT. To give a visual impression of the corresponding environments, Figure 2 illustrates maps obtained by executing state-of-the-art SLAM algorithms. All datasets, the manually verified relations, and map images are available online [17].

VII. Experimental Evaluation

This evaluation is designed to illustrate the properties of our method. We selected three popular mapping techniques

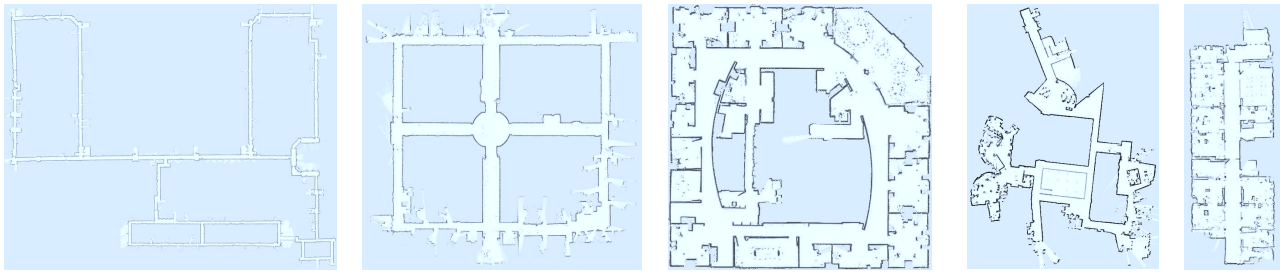

Fig. 2. Maps obtained by the reference datasets used to validate our metric. From left to right: MIT Killian Court, ACES Building at the University of Texas, Intel Research Lab Seattle, MIT CSAIL Building, and building 079 University of Freiburg.

and processed the datasets discussed in the previous section. We provide the obtained scores from the metric for all combinations of SLAM approach and dataset. This will allow other researchers to compare their own SLAM approaches against our methods using the provided benchmark datasets.

A. Evaluation of Existing Approaches

In this evaluation, we considered the following mapping approaches and present results obtained with these techniques using the datasets briefly described in the previous section.

First, we applied incremental scan matching as a kind of baseline approach. Scan matching, here using the approach of Censi [5], computes an open loop maximum likelihood trajectory of the robot incrementally, by matching consecutive scans.

Second, we used GMapping which is a mapping system based on a Rao-Blackwellized Particle Filter (RBPF) for learning grid maps. We used the RBPF implementation described in [10] and available online [24]. It estimates the posterior over maps and trajectories by means of a particle filter. Each particle carries its own map and a hypothesis of the robot pose within that map.

Third, we selected an approach that addresses the SLAM problem by graph optimization. The idea is to construct a graph out of the sequence of measurements. Every node of the graph is labeled with a robot pose and the measurement taken at that pose. Then, a least square error minimization approach is applied to obtain the most-likely configuration of the graph. The approach described by Olson [21] is used to determine constraints and the optimizer TORO available online [24] and described in [11] is applied.

For our evaluation, we manually extracted the relations for all datasets mentioned in the previous section (the data is available online). We then carried out the mapping approaches and used the corrected trajectory to compute the error according to our metric. Note, that the error computed according to our metric (as well as for most other metrics too) can be separated into two components: a translational error and a rotational error. Often, a "weighting-factor" is used to combine both error terms into a single number. In this evaluation, however, we provide both terms separately for a better transparency of the results.

We processed all benchmark datasets mentioned in Section VI using the algorithms listed above. A condensed view of each algorithm's performance is given by the averaged

TABLE I

QUANTITATIVE RESULTS OF DIFFERENT APPROACHES/DATASETS.
[1] SCAN MATCHING HAS BEEN APPLIED AS A PREPROCESSING STEP.

Trans. error m / m^2	Scan Matching	RBPF (50 part.)	Graph Mapping
Aces (abs)	0.173 ± 0.614	0.060 ± 0.049	0.044 ± 0.044
Aces (sqr)	0.407 ± 2.726	0.006 ± 0.011	0.004 ± 0.009
Intel (abs)	0.220 ± 0.296	0.070 ± 0.083	0.031 ± 0.026
Intel (sqr)	0.136 ± 0.277	0.011 ± 0.034	0.002 ± 0.004
MIT (abs)	1.651 ± 4.138	0.122 ± 0.386[1]	0.050 ± 0.056
MIT (sqr)	19.85 ± 59.84	0.164 ± 0.814[1]	0.006 ± 0.029
CSAIL (abs)	0.106 ± 0.325	0.049 ± 0.049[1]	0.004 ± 0.009
CSAIL (sqr)	0.117 ± 0.728	0.005 ± 0.013[1]	0.0001 ± 0.0005
FR 79 (abs)	0.258 ± 0.427	0.061 ± 0.044[1]	0.056 ± 0.042
FR 79 (sqr)	0.249 ± 0.687	0.006 ± 0.020 [1]	0.005 ± 0.011

Rot. error deg / deg^2	Scan Matching	RBPF (50 part.)	Graph Mapping
Aces (abs)	1.2 ± 1.5	1.2 ± 1.3	0.4 ± 0.4
Aces (swr)	3.7 ± 10.7	3.1 ± 7.0	0.3 ± 0.8
Intel (abs)	1.7 ± 4.8	3.0 ± 5.3	1.3 ± 4.7
Intel (sqr)	25.8 ± 170.9	36.7 ± 187.7	24.0 ± 166.1
MIT (abs)	2.3 ± 4.5	0.8 ± 0.8[1]	0.5 ± 0.5
MIT (sqr)	25.4 ± 65.0	0.9 ± 1.7[1]	0.9 ± 0.9
CSAIL (abs)	1.4 ± 4.5	0.6 ± 1.2[1]	0.05 ± 0.08
CSAIL (sqr)	22.3± 111.3	1.9 ± 17.3[1]	0.01 ± 0.04
FR 79 (abs)	1.7 ± 2.1	0.6 ± 0.6[1]	0.6 ± 0.6
FR 79 (sqr)	7.3 ± 14.5	0.7 ± 2.0[1]	0.7 ± 1.7

error over all relations. In Table I (top) we give an overview on the translational error of the various algorithms, while Table I (bottom) shows the rotational error. As expected, it can be seen that the more advanced algorithms (Rao-Blackwellized particle filter and graph mapping) usually outperform scan matching. This is mainly caused by the fact, that scan matching only optimizes the result locally and will introduce topological errors in the maps, especially when large loops have to be closed. A distinction between RBPF and graph mapping seems difficult as both algorithms perform well in general. On average, graph mapping seems to be slightly better than a RBPF for mapping.

To visualize the results and to provide more insights about the metric, we do not provide the scores only but also plots showing the error of each relation. In case of high errors in a block of relations, we label the relations in the maps. This enables us to see not only where an algorithm fails, but might also provide insights why it fails. Inspecting those situations in correlation with the map helps to understand the properties of an algorithm and gives valuable insights on its capabilities. For two datasets, a detailed analysis using these

plots is presented in the following sections.

B. MIT Killian Court

In the MIT Killian Court dataset (also called the infinite corridor dataset), the robot mainly observed corridors with only few structures that support accurate pose correction. The robot traverses multiple nested loops – a challenge especially for the RBPF-based technique. We extracted close to 5000 relations between nearby poses that are used for evaluation. Figure 3 shows three different results and the corresponding error distributions to illustrate the capabilities of our method. Regions in the map with high inconsistencies correspond to relations having a high error. The absence of significant structure along the corridors results in a small or medium re-localization error of the robot in all compared approaches. In sum, we can say the graph-based approach outperforms the other methods and that the score of our metric reflects the impression of a human about map quality obtained by visually inspecting the mapping results (the vertical corridors in the upper part are supposed to be parallel).

C. Freiburg Indoor Building 079

The building 079 of the University of Freiburg is an example for a typical office environment. The building consists of one corridor which connects the individual rooms. Figure 4 depicts the results of the individual algorithms (scan matching, RBPF, graph-based). In the first row of Figure 4, the relations having a translational error greater than 0.15 m are highlighted in dark blue.

In the left plot showing the scan matching result, the relations plotted in blue are generated when the robot revisits an already known region. These relations are visible in the corresponding error plots (Figure 4 first column, second and third row). As can be seen from the error plots, these relations with a number greater than 1000 have a larger error than the rest of the dataset. The fact that the pose estimate of the robot is sub-optimal and that the error accumulates can also be seen by the rather blurry map and that some walls occur twice. In contrast to that, the more sophisticated algorithms, namely RBPF and graph mapping, are able to produce consistent and accurate maps in this environment. Only very few relations show an increased error (illustrated by dark blue relations).

D. Summary of the Experiments

Our evaluation illustrates that the proposed metric provides a ranking of the results of mapping algorithms that is likely to be compatible with a ranking made by humans. Inconsistencies yield increased error scores since in the wrongly mapped areas the relations obtained from manual matching are not met. By visualizing the error of each constraint as done in the plots in this section, one can identify regions in which algorithms fail and we believe that this helps to understand where and why different approaches have problems to build accurate maps.

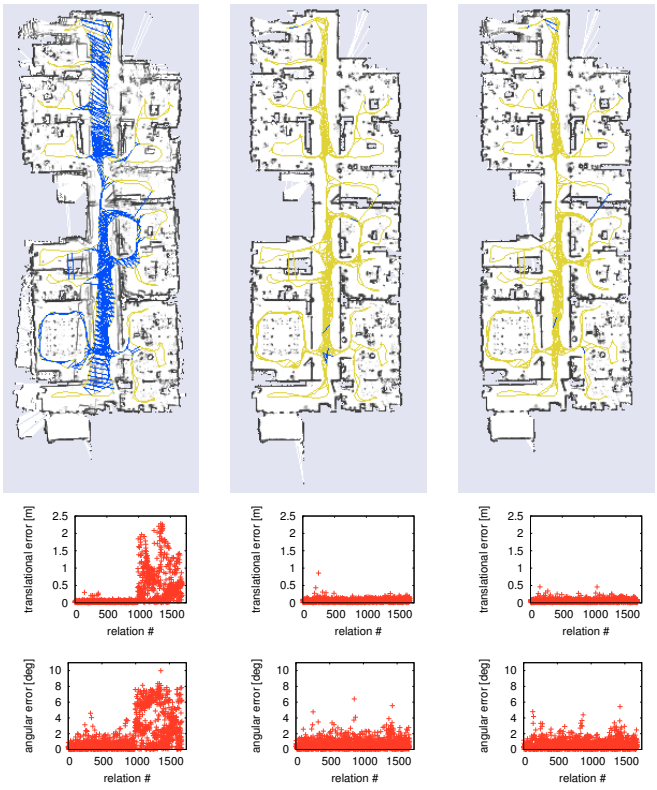

Fig. 4. This figure shows the Freiburg Indoor Building 079 dataset. Each column reports the results of one approach. Left: scan-matching, middle: RBPF and right a graph based algorithm. Within each column, the top image shows the map, the middle plot is the translational error and the bottom one is the rotational error.

VIII. Conclusion

In this paper, we presented a framework for comparing the results of SLAM approaches with the goal to create objective benchmarks. We proposed a metric for measuring the error of a SLAM system based on the corrected trajectory. Our metric uses only relative relations between poses and is motivated by the energy needed to transform an estimate into ground truth. This overcomes serious shortcomings of approaches using a global reference frame to compute the error. The metric even allows the comparison of SLAM approaches that use different estimation techniques or different sensor modalities. In addition to the proposed metric, we provide robotic datasets together with relative relations between poses for benchmarking. These relations have been obtained by manually matching observations and yield a high matching accuracy. Finally, we provide an error analysis for three mapping systems using the metric and datasets. We believe that our results are a valuable benchmark for SLAM researchers since we provide a framework that allows for objectively and comparably easily analyzing the results of SLAM systems.

References

[1] F. Amigoni, S. Gasparini, and M. Gini. Good experimental methodologies for robotic mapping: A proposal. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

[2] B. Balaguer, S. Carpin, and S. Balakirsky. Towards quantitative comparisons of robot algorithms: Experiences with SLAM in simulation and real world systems. In *IROS 2007 Workshop*, 2007.

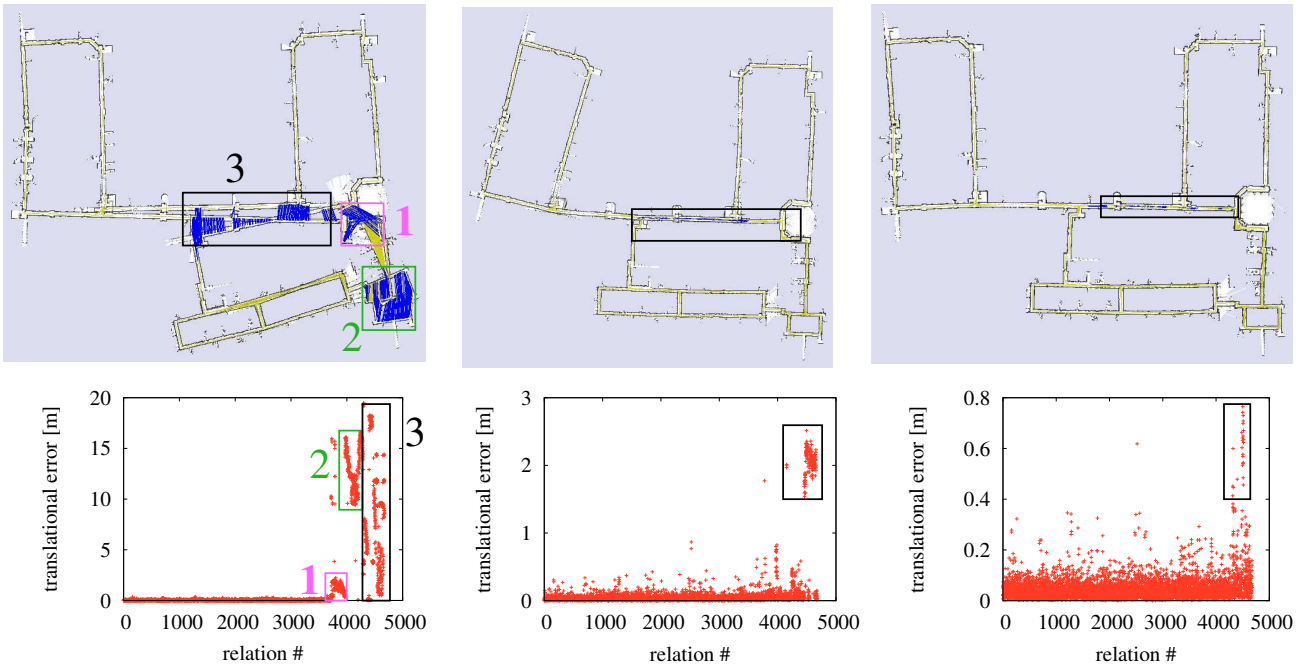

Fig. 3. The MIT Killian Court dataset. The reference relations are depicted in light yellow. The left column shows the results of scan-matching, the middle column the result of a GMapping using 50 samples, and the right column shows the result of a graph-based approach. The regions marked in the map (boxes and dark blue relations) correspond to regions in the error plots having high error. The rotational error is not plotted due to space reasons.

[3] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos. Rawseeds a project on SLAM benchmarking. In *Proc. of the IROS WS on Benchmarks in Robotics Research*, 2006.

[4] A. Censi. The achievable accuracy for range finder localization. *IEEE Transactions on Robotics*. Under review.

[5] A. Censi. Scan matching in a probabilistic framework. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2291–2296, 2006.

[6] EPFL and IROS. Cleaning Robot Contest, 2002. http://robotika.cz/competitions/cleaning2002/en.

[7] ESA. Lunar robotics challenge, 2008. http://www.esa.int/esaCP/SEM4GKRTKMF_index_0.html.

[8] R. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2428–2435, 2005.

[9] U. Frese. Dlr spatial cognition data set. http://www.informatik.uni-bremen.de/agebv/en/DlrSpatialCognitionDataSet, 2008.

[10] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23:34–46, 2007.

[11] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.

[12] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.

[13] J. Hermosillo, C. Pradalier, S. Sekhavat, C. Laugier, and G. Baille. Towards motion autonomy of a bi-steerable car: Experimental issues from map-building to trajectory execution. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.

[14] A. Howard and N. Roy. Radish: The robotics data set repository, standard data sets for the robotics community, 2003. http://radish.sourceforge.net/.

[15] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. of the American Control Conference*, pages 1628–1632, 1995.

[16] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. On measuring the accuracy of SLAM algorithms. *Autonomous Robots*, 2009. Condtionally accepted for publication.

[17] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. Slam benchmarking webpage. http://ais.informatik.uni-freiburg.de/slamevaluation, 2009.

[18] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4):376–382, 1991.

[19] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[20] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1151–1156, 2003.

[21] E. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2008.

[22] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2262–2269, 2006.

[23] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial realtionships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.

[24] C. Stachniss, U. Frese, and G. Grisetti. OpenSLAM.org – give your algorithm to the community. http://www.openslam.org, 2007.

[25] C. Stachniss, G. Grisetti, N. Roy, and W. Burgard. Evaluation of gaussian proposal distributions for mapping with rao-blackwellized particle filters. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[26] S. Thrun. An online mapping algorithm for teams of mobile robots. *Int. Journal of Robotics Research*, 20(5):335–363, 2001.

[27] S. Thrun and colleagues. Winning the darpa grand challenge. *Journal on Field Robotics*, 2006.

[28] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Int. Journal of Robotics Research*, 23(7/8):693–716, 2004.

[29] O. Wulf, A. Nüchter, J. Hertzberg, and B. Wagner. Benchmarking urban six-degree-of-freedom simultaneous localization and mapping. *Journal of Field Robotics*, 25(3):148–163, 2008.

[30] M. Yguel, C.T.M. Keat, C. Braillon, C. Laugier, and O. Aycard. Dense mapping for range sensors: Efficient algorithms and sparse representations. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.

[31] Z. Zivkovic, O. Booij, B. Krose, E.A. Topp, and H.I. Christensen. From sensors to human spatial concepts: An annotated data set. *IEEE Transactions on Robotics*, 24(2):501–505, 2008.