Autonomous Robots manuscript No.
(will be inserted by the editor)

On Measuring the Accuracy of SLAM Algorithms

Rainer Kümmerle · **Bastian Steder** · **Christian Dornhege** · **Michael Ruhnke** · **Giorgio Grisetti** · **Cyrill Stachniss** · **Alexander Kleiner**

Received: date / Accepted: date

Abstract In this paper, we address the problem of creating an objective benchmark for evaluating SLAM approaches. We propose a framework for analyzing the results of a SLAM approach based on a metric for measuring the error of the corrected trajectory. This metric uses only relative relations between poses and does not rely on a global reference frame. This overcomes serious shortcomings of approaches using a global reference frame to compute the error. Our method furthermore allows 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 robotics 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 to easily analyze and objectively compare different SLAM approaches.

Keywords SLAM; mapping accuracy; benchmarking

All authors are with the
University of Freiburg, Dept. of Computer Science, Georges Köhler Allee 79, 79110 Freiburg, Germany
Tel.: +49-761-203-8006, Fax: +49-761-203-8007
E-mail: {kuemmerl,steder,dornhege,ruhnke,grisetti,stachnis,kleiner}@informatik.uni-freiburg.de

1 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 in the last decades. Robots that are able to acquire an accurate model of their environment are regarded as fulfilling a major precondition of truly autonomous agents.

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 [Smith and Cheeseman, 1986; Dissanayake *et al*., 2000; Gutmann and Konolige, 1999; Hähnel *et al*., 2003; Montemerlo *et al*., 2003; Thrun, 2001; Leonard and Durrant-Whyte, 1991]. 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. This mutual dependency among the pose and the map estimates makes the SLAM problem hard and requires searching for a solution in a high-dimensional space.

Whereas dozens of different techniques to tackle the SLAM problem have been presented, 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 distance 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. In the community, 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 a common performance measure (metric). This metric should enable the user 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. We aim to establish a benchmark that allows for objectively measuring the performance of a mapping system. We propose a metric that operates only on relative geometric relations between poses along the trajectory of the robot. Our approach allows for making comparisons 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 easily. In this paper, we present manually obtained relative relations for different datasets that can be used for carrying out comparisons. We furthermore provide evaluations for the results of three different mapping techniques, namely scan-matching, SLAM using Rao-Blackwellized particle filter [Grisetti *et al*., 2007b; Stachniss *et al*., 2007b], and a maximum likelihood SLAM approach based on the graph formulation [Grisetti *et al*., 2007c; Olson, 2008].

The remainder of this paper is organized a follows. First, we discuss related work in Section 2 and present the proposed metric based on relative relations between poses along

the trajectory of the robot in Section 3. Then, in Section 4 and Section 5 we explain how to obtain such relations in practice. In Section 6, we briefly discuss how to benchmark if the tested SLAM system does not provide pose estimates. Next, in Section 7 we provide a brief overview of the datasets used for benchmarking and in Section 8 we present our experiments which illustrate different properties of our method and we give benchmark results for three existing SLAM approaches.

2 Related Work

Learning maps is a frequently studied problem in the robotics literature. Mapping techniques for mobile robots can be classified according to the underlying estimation technique. The most popular approaches are extended Kalman filters (EKFs) [Leonard and Durrant-Whyte, 1991; Smith *et al*., 1990], sparse extended information filters [Eustice *et al*., 2005a; Thrun *et al*., 2004], particle filters [Montemerlo *et al*., 2003; Grisetti *et al*., 2007b], and least square error minimization approaches [Lu and Milios, 1997; Frese *et al*., 2005; Gutmann and Konolige, 1999; Olson *et al*., 2006]. For some applications, it might even be sufficient to learn local maps only [Hermosillo *et al*., 2003; Thrun and colleagues, 2006; Yguel *et al*., 2007].

The effectiveness of the EKF approaches comes from the fact that they estimate a fully correlated posterior about landmark maps and robot poses. Their weakness lies in the strong assumptions that have to be made on both, the robot motion model and the sensor noise. If these assumptions are violated the filter is likely to diverge [Julier *et al*., 1995; Uhlmann, 1995].

Thrun *et al*. [2004] proposed a method to correct the poses of a robot based on the inverse of the covariance matrix. The advantage of sparse extended information filters (SEIFs) is that they make use of the approximative sparsity of the information matrix. Eustice *et al*. [2005a] presented a technique that more accurately computes the error-bounds within the SEIF framework and therefore reduces the risk of becoming overly confident.

Dellaert and colleagues proposed a smoothing method called square root smoothing and mapping (SAM) [Dellaert, 2005; Kaess *et al*., 2007; Ranganathan *et al*., 2007]. It has several advantages compared to EKF-based solutions since it better covers the non-linearities and is faster to compute. In contrast to SEIFs, it furthermore provides an exactly sparse factorization of the information matrix. In addition to that, SAM can be applied in an incremental way [Kaess *et al*., 2007] and is able to learn maps in 2D and 3D.

Frese's TreeMap algorithm [Frese, 2006] can be applied to compute nonlinear map estimates. It relies on a strong topological assumption on the map to perform sparsification of the information matrix. This approximation ignores small entries in the information matrix. In this way, Frese is able to perform an update in $\mathcal{O}(\log n)$ where n is the number of features.

An alternative approach to find maximum likelihood maps is the application of least square error minimization. The idea is to compute a network of constraints given the sequence of sensor readings. It should be noted that our approach for evaluating SLAM methods presented in this paper is highly related to this formulation of the SLAM problem.

Lu and Milios [1997] introduced the concept of graph-based or network-based SLAM using a kind of brute force method for optimization. Their approach seeks to optimize the whole network at once. Gutmann and Konolige [1999] proposed an effective way for constructing such a network and for detecting loop closures while running an incremental estimation algorithm. Duckett *et al*. [2002] propose the usage of Gauss-Seidel relaxation to minimize the error in the network of relations. To make the problem linear, they assume

knowledge about the orientation of the robot. Frese *et al*. [2005] propose a variant of Gauss-Seidel relaxation called multi-level relaxation (MLR). It applies relaxation at different resolutions. MLR is reported to provide very good results in flat environments especially if the error in the initial guess is limited.

Olson *et al*. [2006] presented an optimization approach that applies stochastic gradient descent for resolving relations in a network efficiently. Extensions of this work have been presented by Grisetti *et al*. [2007c; 2007a] Most approaches to graph-based SLAM such as the work of Olson *et al*., Grisetti *et al*., Frese *et al*., and others focus on computing the best map and assume that the relations are given. The ATLAS framework [Bosse *et al*., 2003], hierarchical SLAM [Estrada *et al*., 2005], or the work of Nüchter *et al*. [2005], for example, can be used to obtain the constraints. In the graph-based mapping approach used in this paper, we followed the work of Olson [2008] to extract constraints and applied [Grisetti *et al*., 2007c] for computing the minimal error configuration.

Activities related to performance metrics for SLAM methods, such as the work described in this paper, can roughly be divided into three major categories: First, competition settings where robot systems are competing within a defined problem scenario, such as playing soccer, navigating through a desert, or searching for victims. Second, collections of publicly available datasets that are provided for comparing algorithms on specific problems. Third, related publications that introduce methodologies and scoring metrics for comparing different methods.

The comparison of robots within benchmarking scenarios is a straight-forward approach for identifying specific system properties that can be generalized to other problem types. For this purpose numerous robot competitions have been initiated in the past, evaluating the performance of cleaning robots [EPFL and IROS, 2002], robots in simulated Mars environments [ESA, 2008], robots playing soccer or rescuing victims after a disaster [RoboCup Federation, 2009], and cars driving autonomously in an urban area [Darpa, 2007]. However, competition settings are likely to generate additional noise due to differing hardware and software settings. For example, when comparing mapping solutions in the RoboCup Rescue domain, the quality of maps generated using climbing robots can greatly differ from those generated on wheel-based robots operating in the plane. Furthermore, the approaches are often tuned to the settings addressed in the competitions.

Benchmarking of systems from datasets has reached a rather mature level in the vision community. There exist numerous data bases and performance measures, which are available via the Internet. Their purpose is to validate, for example, image annotation [Torralba *et al*., 2007], range image segmentation [Hoover *et al*., 1996], and stereo vision correspondence algorithms [Scharstein and Szeliski, 2002]. These image databases provide ground truth data [Torralba *et al*., 2007; Scharstein and Szeliski, 2002], tools for generating ground truth [Torralba *et al*., 2007] and computing the scoring metric [Scharstein and Szeliski, 2002], and an online ranking of results from different methods [Scharstein and Szeliski, 2002] for direct comparison.

In the robotics community, there are some well-known web sites providing datasets [Howard and Roy, 2003; Bonarini *et al*., 2006] and algorithms [Stachniss *et al*., 2007a] for mapping. However, they neither provide ground truth data nor recommendations on how to compare different maps in a meaningful way.

Some preliminary steps towards benchmarking navigation solutions have been presented in the past. Amigoni *et al*. [2007] 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. For example, the mapping system should be applied to publicly available data, parameters of the algorithm

should be clearly indicated (and also effects of their variations presented), as well as parameters of the map should be explained. When ground truth data is available, they suggest to utilize the Hausdorff metric for map comparison.

Wulf *et al*. [2008] proposed the idea of using manually supervised Monte Carlo Localization (MCL) for matching 3D scans against a reference map. They suggested that a reference map be generated maps from independently created CAD data, which can be obtained from the land registry office. The comparison between generated map and ground truth has been carried out by computing the Euclidean distance and angle difference of each scan, and plotting these over time. Furthermore, they provided standard deviation and maximum error of the track for comparisons. We argue that comparing the absolute error between two tracks might not yield a meaningful assertion in all cases as illustrated in the initial example in Section 3. This effect gets even stronger when the robot makes a small angular error especially in the beginning of the dataset (and when it does not return to this place again). Then, large parts or the overall map are likely to be consistent, the error, however, will be huge. Therefore, the method proposed in this paper favors comparisons between relative poses along the trajectory of the robot. Based on the selection between which pose relations are considered, different properties can be highlighted.

Balaguer *et al*. [2007] utilize the USARSim robot simulator and a real robot platform for comparing different open source SLAM approaches. They demonstrated that maps resulting from processing simulator data are very close to those resulting from real robot data. Hence, they concluded 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 [Stachniss *et al*., 2007b]. Gaussian noise, however, is typically used in most simulation systems. In addition to that, Balaguer *et al*. do not provide a quantitative measure for comparing generated maps with ground truth. As with many other approaches, their comparisons were carried out by visual inspection.

The paper presented here extends our work [Burgard *et al*., 2009] with a more detailed description of the approach, a technique for extracting relations from aerial images, and a significantly extended experimental evaluation.

3 Metric for Benchmarking SLAM Algorithms

In this paper, 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 at which observations are made. All benchmark computations will be performed on this set.

3.1 A Measure for Benchmarking SLAM Results

Let $x_{1:T}$ be the poses of the robot estimated by a SLAM algorithm from time step 1 to T. Let $x^*_{1:T}$ be the reference poses of the robot, ideally the true locations. A straightforward error

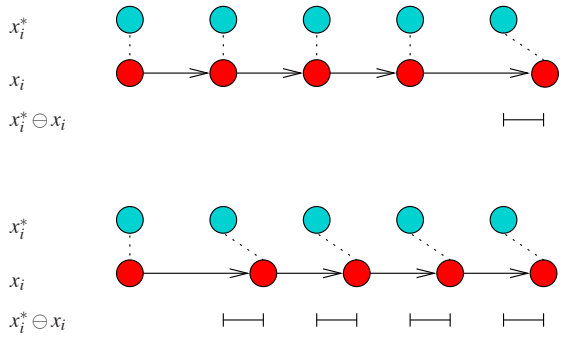


Fig. 1 This figure illustrates a simple example where the metric in Eq. 1 fails. The light blue circles show the reference positions of the robot $\{x_i^*\}$ while the dark red circles show the estimated positions of the robot $\{x_i\}$. The correspondence between the estimated locations and the ground truth is shown with dashed lines, and the direction of motion of the robot is highlighted with arrows. In the situation shown in the upper part, the robot makes a small mistake at the end of the path. This results in a small error. Conversely, in the situation illustrated on the bottom part of the figure the robot makes a small error of the same entity, but at the beginning of the travel, thus resulting in a much bigger global error.

metric could be defined as

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

where $\oplus$ is the standard motion composition operator and $\ominus$ its inverse. Let $\delta_{i,j} = x_j \ominus x_i$ be the relative transformation that moves the node x_i onto x_j and accordingly $\delta_{i,j}^* = x_j^* \ominus x_i^*$. Eq. 1 can be rewritten as

$$\varepsilon(x_{1:T}) = \sum_{t=1}^{T} \left((x_1 \oplus \delta_{1,2} \oplus \ldots \oplus \delta_{t-1,t}) \ominus (x_1^* \oplus \delta_{1,2}^* \oplus \ldots \oplus \delta_{t-1,t}^*) \right)^2 \tag{2}$$

We claim that this metric is suboptimal for comparing the result of a SLAM algorithm. To illustrate this, consider the following 1D example in which a robot travels along a straight line. Let the robot make a translational error of e during the first motion, $\delta_{1,2} = \delta_{1,2}^* + e$, and perfect estimates at all other points in time $\delta_{t,t+1} = \delta_{t,t+1}^*$ for $t > 1$. Thus, the error according to Eq. 2, will be $T \cdot e$, since $\delta_{1,2}$ is contained in every pose estimate for $t > 1$. If, however, we estimate the trajectory backwards starting from x_T to x_1 or alternatively by shifting the whole map by e, we obtain an error of e only. This indicates, that such an error estimate is suboptimal for comparing the results of a SLAM algorithm. See also Figure 1 for an illustration.

In the past, the so-called NEES measure proposed in [Bar-Shalom *et al.*, 2001] as

$$\varepsilon(x_{1:T}) = \sum_{t=1}^{T} (x_t - x_t^*)^T \Omega_t (x_t - x_t^*), \tag{3}$$

has often been used to evaluate the results of a SLAM approach (e.g., [Eustice *et al.*, 2005b]). Here Ω_t represents the information matrix of the pose x_t. The NEES measure, however, suffers from a similar problem as Eq. 1 when computing ε. In addition to that, not all SLAM algorithms provide an estimate of the information matrix and thus cannot be compared based on Eq. 3.

Based on this experience, we propose a measure 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 [1997] — by considering the nodes as masses and connections between them as springs. Thus, our metric is based on the *relative* displacement between robot poses. Instead of comparing x to x^* (in the global reference frame), we do the operation based on δ and δ^* as

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

where N is the number of relative relations and $trans(\cdot)$ and $rot(\cdot)$ are used to separate and weight the translational and rotational components. We suggest that both quantities be evaluated 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.

Our error metric, however, leaves open which relative displacements $\delta_{i,j}$ are included in the summation in Eq. 4. Using the metric and selecting relations are two related but different problems. 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 derives 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.

3.2 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 measure.

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, i.e. its spatial structure is correctly representing the environment. We refer to a map having this property as being locally consistent. Figure 3 illustrates the concept of locally consistent maps which are suited for a robot to carry out navigation tasks.

By selecting the relative displacements $\delta_{i,j}$ used in Eq. 4 for a given dataset, the user can highlight certain properties and thus design a measure 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 additional knowledge (for example, that a corridor has a certain length and is straight) into the benchmark.

4 Obtaining Reference Relations in Indoor Environments

In practice, the key question regarding Eq. 4 is how to determine the *true relative displacements* between poses. Obviously, the true values are not available. However, we can determine close-to-true values by using the information recorded by the mobile robot and the background knowledge of the human recording the datasets, which, of course, involves manual work.

Please note, that the metric presented above is independent of the actual sensor used. In the remainder of this paper, however, we will concentrate on robots equipped with a laser range finders, since they 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 to generate relations. One way would be to temporarily mount a laser range finder on the robot and calibrate it in the robot coordinate frame. If this is not possible, one has to provide a method for accurately determining the relative displacements between two poses from which an observation has been taken that observes the same part of the space.

4.1 Initial Guess

In our work, we propose the following strategy. First, one tries to find 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.

4.1.1 Symeo System

One way for obtaining good initial guesses with no or only very few interactions can be the use of the Symeo Positioning System LPR-B [Symeo GmbH, 2008]. It works similar to a local GPS system but indoors and can achieve a localization accuracy of around 5 cm to 10 cm. The problem is that such a system designed for industrial applications is typically not present at most robotics labs. If available, however, it is well suited for a rather accurate initial guess of the robot's position.

4.1.2 Initial Estimate via SLAM Approaches

In most cases, however, researchers in robotics will have SLAM algorithms at hand that can be used to compute an initial guess about the poses of the robot. In the recent years, several accurate methods have been proposed to serve as such a guess (see Section 2). By manually inspecting the estimates of the algorithm, a human can accept, refine, or discard a match and also add missing relations.

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.

4.2 Manual Matching Refinement and Rejection

Based on the initial guess about the position 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

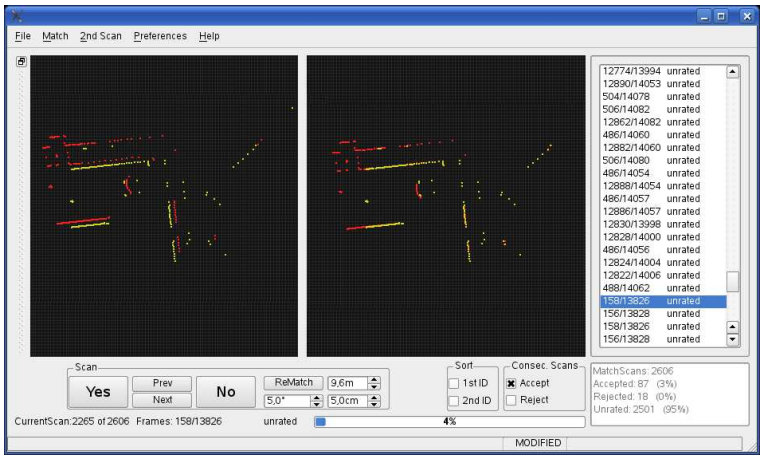


Fig. 2 User interface for matching, accepting, and discarding pairs of observations.

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 the user is able to incorporate his background knowledge about the environment.

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 better. Alternatively, the pair can be discarded.

This approach might sound labor-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 1,700 relations, it took an unexperienced user approximately four hours to extract the relative translations that then served as the input to the error calculation. Figure 2 shows a screen-shot of the user interface used for evaluation.

It should be noted that for the manual work described above some kind of structure in the environment is required. The manual labor might be very hard in highly unstructured scenes.

4.3 Other 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 incorporate knowledge about the global geometry of an environment if this is available. This fact is, for example, illustrated by the black dashed line in Figure 3 that implies a known distance between two poses in a corridor that are not adjacent. Figure 4 plots the corresponding error identified by the relation.

In the experimental evaluation, we will show one example for such additional relations used in real world datasets. In this example, we utilize relations derived from satellite image data.

5 Obtaining Reference Relations in Outdoor Environments

The techniques described in the previous section can be used to obtain a close-to-ground-truth for indoor environments. In outdoor scenarios however, the manual validation of the data is usually less practical due to the reduced structure and the large size. In wide open areas it may be difficult for a human operator to determine whether a potential alignment between laser scans is good or not due to the limited range of the scanner. Furthermore the poor structure of the environment makes this procedure hard even when the laser senses a high number of obstacles.

GPS is commonly used to bound the global uncertainty of a vehicle moving outdoors. Unfortunately, GPS suffers from outages or occlusions so that a robot relying on GPS might encounter substantial positioning errors. Especially in urban environments, GPS is known to be noisy. Even sophisticated SLAM algorithms cannot fully compensate for these errors as there still might be lacking relations between observations combined with large odometry errors that introduce a high uncertainty in the current position of the vehicle.

As an alternative to GPS, it is possible to use aerial images to determine relations close to the ground truth. We investigated this approach in our previous work [Kümmerle *et al.*, 2009] and we show that this solution yields a better global consistency of the resulting map, if we consider the prior information. Satellite images of locations are widely available on the web by popular tools like Google-Earth or Microsoft Live-Earth. This data can be used as prior information to localize a robot equipped with a 3D laser range finder.

The overall approach is based on the Monte-Carlo localization framework [Dellaert *et al.*, 1998]. The satellite images are captured from a viewpoint significantly different from the one of the robot. However, by using 3D scans we can extract 2D information which is more likely to be consistent with the one visible in the reference map. In this way, we can prevent the system from introducing inconsistent prior information.

In the following, we explain how we adapted Monte Carlo Localization (MCL) to operate on aerial images and how to select points from 3D scans to be considered in the observation model of MCL. This procedure returns a set of candidate robot locations $\hat{x}_i$. From those positions, we then select a subset of pairs of locations from which to compute the reference displacements $\hat{\delta}_{i,j}$ to be used in the metric.

5.1 Monte Carlo Localization

To estimate the pose x of the robot in its environment, we consider probabilistic localization, which follows the recursive Bayesian filtering scheme. The key idea of this approach is to maintain a probability density $p(x_t \mid z_{1:t}, u_{0:t-1})$ of the location x_t of the robot at time t given all observations $z_{1:t}$ and all control inputs $u_{0:t-1}$. This posterior is updated as follows:

$$p(x_t \mid z_{1:t}, u_{0:t-1}) = \\ \alpha \cdot p(z_t \mid x_t) \cdot \int p(x_t \mid u_{t-1}, x_{t-1}) \cdot p(x_{t-1})\, dx_{t-1}. \tag{5}$$

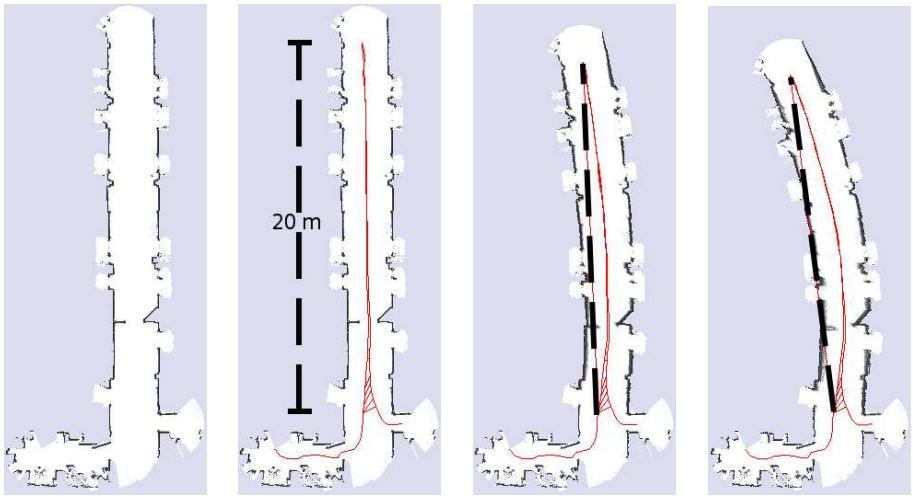


Fig. 3 Example of the performance measure on maps generated using different sensor setups. The relations between close-by positions are determined by a human assisted scan-alignment procedure performed on scans acquired at close-by locations. The long dashed line represents a relation added by manually measuring the relative distance at two locations of the robot: (a) the reference map obtained from the relative measurements, (b) the reference map superimposed with the network of relative measurements, (c) a map obtained by scan matching using a 4 meters range sensor, with the superimposed relation (this map is still usable for navigating a robot), (d) a map obtained by cropping the range of the sensor to 3 meters. Whereas the quality of the rightmost map is visibly decreased, it is also adequate for robot navigation since it preserves a correct topology of the environment (all doorways are still visible) and it correctly reflects the local spatial structure of the corridor. Therefore, it is locally consistent, but not globally consistent as (a). See also Figure 4 for corresponding error plots.

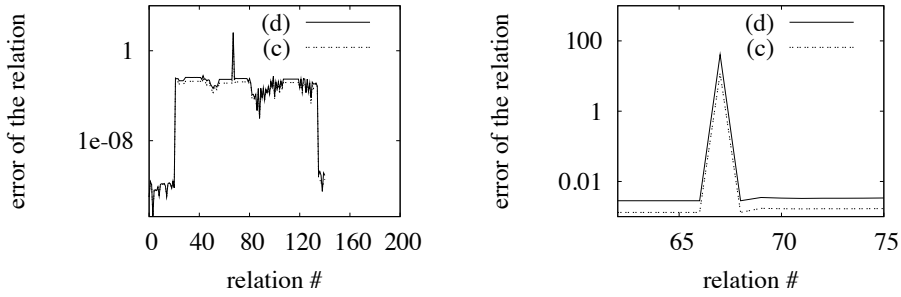
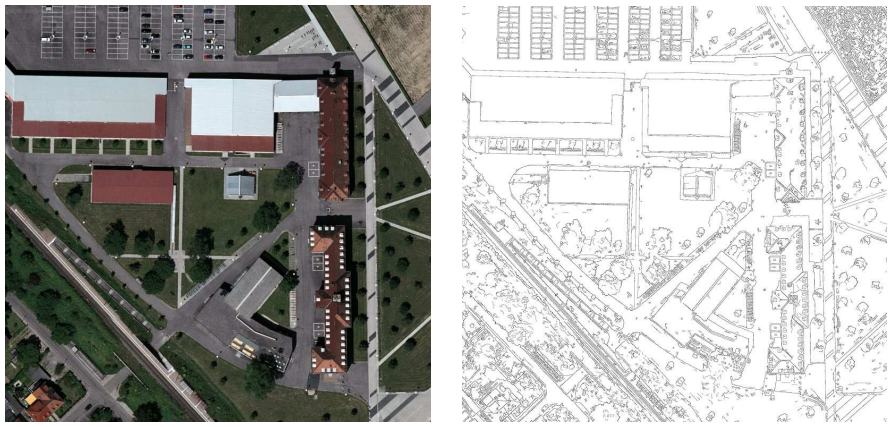


Fig. 4 This figure shows the behavior of the error metric for the maps (c) and (d) in Figure 3. On the left we plot the error introduced by the individual relations. The right plot is a magnification of the left one in the region corresponding to the manually introduced relations marked on the images with the dashed line. This results in a substantial increase of the global ε of SLAM results under comparison.



(a) (b)

Fig. 5 **(a)** A Google Earth image of the Freiburg campus. **(b)** The corresponding Canny image. Despite the considerable clutter, the structure of the buildings and the vertical elements are clearly visible.

Here, α is a normalization constant which ensures that $p(x_t \mid z_{1:t}, u_{0:t-1})$ sums up to one over all x_t. The terms to be described in Eq. 5 are the prediction model $p(x_t \mid u_{t-1}, x_{t-1})$ and the sensor model $p(z_t \mid x_t)$. One contribution of this work is an appropriate computation of the sensor model in the case that a robot equipped with a 3D range sensor operates in a given birds-eye map.

MCL is a variant of particle filtering [Doucet *et al.*, 2001] where each particle corresponds to a possible robot pose and has an assigned weight $w^{[i]}$. The belief update from Eq. 5 is performed according to the following two alternating steps:

1. In the prediction step, we draw for each particle with weight $w^{[i]}$ a new particle according to $w^{[i]}$ and to the prediction model $p(x_t \mid u_{t-1}, x_{t-1})$.
2. In the correction step, a new observation z_t is integrated. This is done by assigning a new weight $w^{[i]}$ to each particle according to the sensor model $p(z_t \mid x_t)$.

Furthermore, the particle set needs to be re-sampled according to the assigned weights to obtain a good approximation of the pose distribution with a finite number of particles.

So far, we have described the general framework of MCL. In the next section, we will describe the sensor model for determining the likelihood $p(z_t \mid x_t)$ of perceiving the 3D scan z_t from a given robot position x_t within an aerial image. For convenience, we will drop the time index t in the remainder of this section.

5.2 Sensor Model for 3D Range Scans in Aerial Images

The task of the sensor model is to determine the likelihood $p(z \mid x)$ of a reading z given the robot is at pose x. In our current system, we apply the so called endpoint model or likelihood fields [Thrun *et al.*, 2005]. Let z^k be the endpoints of a 3D scan z. The endpoint model computes the likelihood of a reading based only on the distances between a scan point $\hat{z}^k$ re-projected onto the map according to the pose x of the robot and the point in the map $\hat{d}^k$ which is closest to $\hat{z}^k$ as:

$$p(z \mid x) = f(\|\hat{z}^1 - \hat{d}^1\|, \ldots, \|\hat{z}^k - \hat{d}^k\|). \tag{6}$$

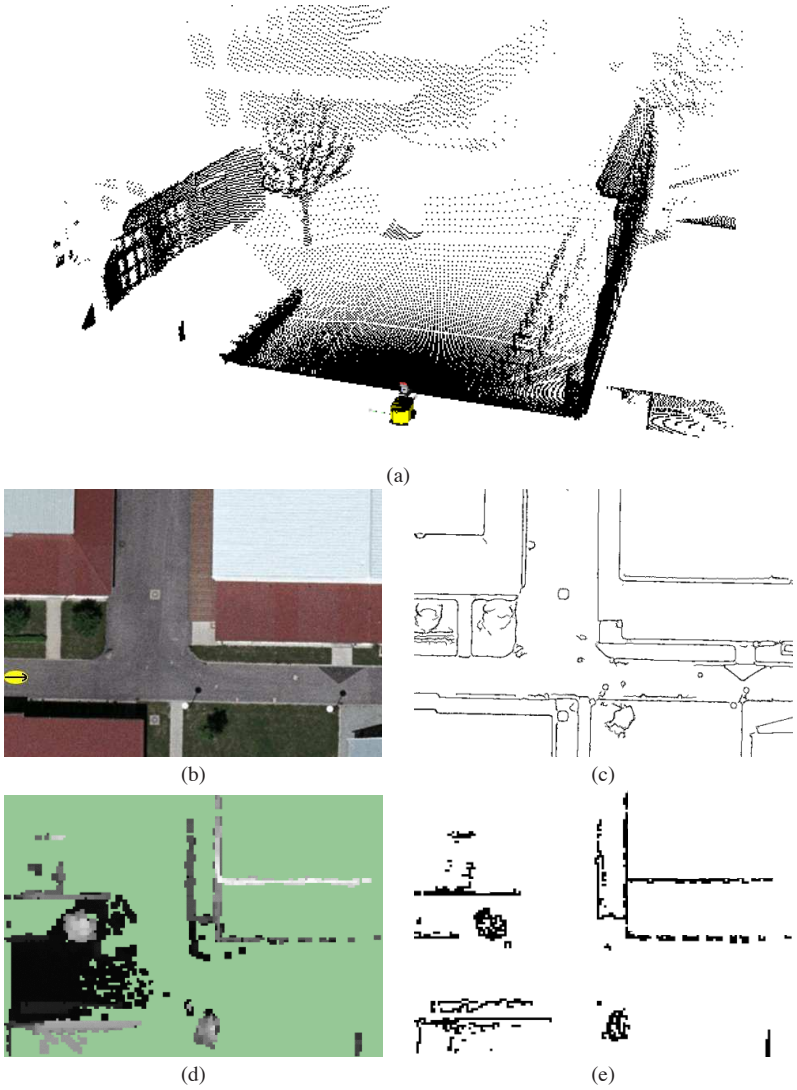


Fig. 6 **(a)** A 3D scan represented as a point cloud. **(b)** The aerial image of the scene. **(c)** The Canny edges extracted from (b). **(d)** A view from the top, where the gray value represents the maximal height per cell. The darker the color the lower the height. **(e)** Extracted height variations from (d).

If we assume that the beams are independent and the sensor noise is normally distributed we can rewrite Eq. 6 as

$$f(\|\hat{z}^1 - \hat{d}^1\|, \ldots, \|\hat{z}^k - \hat{d}^k\|) \propto \prod_j e^{\frac{(\hat{z}^j - \hat{d}^j)^2}{\sigma^2}}. \tag{7}$$

Since the aerial image only contains 2D information about the scene, we need to select a set of beams from the 3D scan, which are likely to result in structures that can be identified

and matched in the image. In other words, we need to transform both the scan and the image into a set of 2D points which can be compared via the function $f(\cdot)$.

To extract these points from the image we employ the standard Canny edge extraction procedure [Canny, 1986]. The idea behind this is that if there is a height gap in the aerial image, there will often also be a visible change in intensity in the aerial image and we assume that this intensity change is detected by the edge extraction procedure. In an urban environment, such edges typically correspond to borders of roofs, trees, fences or other structures. Of course, the edge extraction procedure returns a lot of false positives that do not represent any actual 3D structure, like street markings, grass borders, shadows, and other flat markings. All these aspects have to be considered by the sensor model.

A straightforward way to address this problem is to select a subset of beams z^k from the 3D scan which will then be used to compute the likelihood. The beams which should be considered are the ones which correspond to significant variations along the z direction of the 3D scan. For vertical structures, a direct matching between the extracted edges and the measurements of a horizontal 2D laser range scanner can be performed, as discussed by Früh and Zakhor [2004]. If a 3D laser range finder is available, we also attempt to match variations in height that are not purely vertical structures, like trees or overhanging roofs. This procedure is illustrated by the sequence of images in Figure 6.

In the current implementation, we considered variations in height of 0.5 m and above as possible positions of edges that could also be visible in the aerial image. The positions of these variations relative to the robot can then be matched against the Canny edges of the aerial image in a point-by-point fashion, similar to the matching of 2D-laser scans against an occupancy grid map. Additionally, we employ a heuristic to detect when the prior is not available, i.e., when the robot is inside of a building or under overhanging structures. This is based on the 3D perception. If there is a ceiling which leads to range measurements above the robot no global relations from the localization are integrated, since we assume that the area the robot is sensing is not visible in the aerial image.

Figure 7 shows an example trajectory estimated with this technique (in red) and the GPS positions (in blue). As can be seen, the estimates are more accurate than the GPS data. Thus the improved guess facilitates the manual verification of the data. Note that the approach presented here is used to obtain the candidate relations for outdoor datasets. A human operator has to accept or decline all relations found by the approach.

6 Benchmarking for 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. To facilitate evaluation one could store the current pose for each processing step and use it to build a trajectory. This would lead to good results if only local accuracy is considered. However, global corrections appearing later in run-time are not represented correctly.

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 operating on the landmark locations instead of based on the

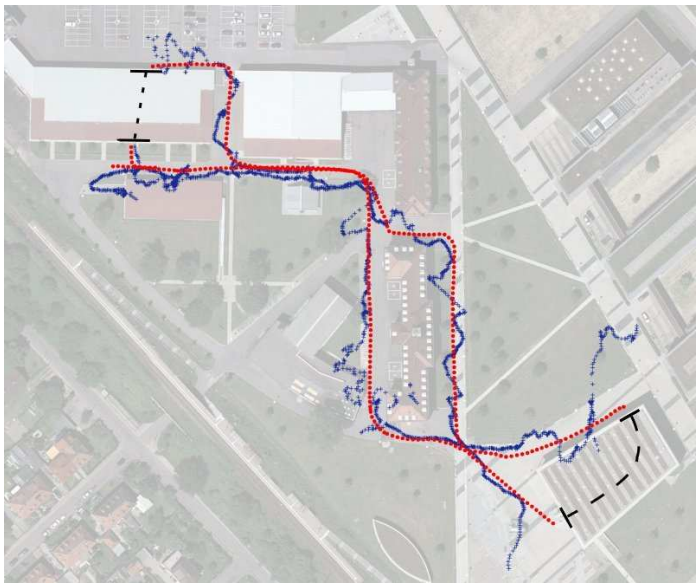


Fig. 7 Trajectory estimated using satellite images versus GPS data overlaid on the image of the ALU-FR campus.

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.

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 [2008]. This, however, might get intractable for SIFT-like features obtained with high frame rate 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 would recommend the first option to evaluate techniques such as EKF.

7 Datasets for Benchmarking

To validate the metric, we selected a set of datasets representing 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 [1] (Infinite Corridor) and the dataset of the ACES building at the University of Texas, Austin [2]. 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 [3], and a dataset acquired at the CSAIL at

[1] Courtesy of Mike Bosse

[2] Courtesy of Patrick Beeson

[3] Courtesy of Dirk Haehnel

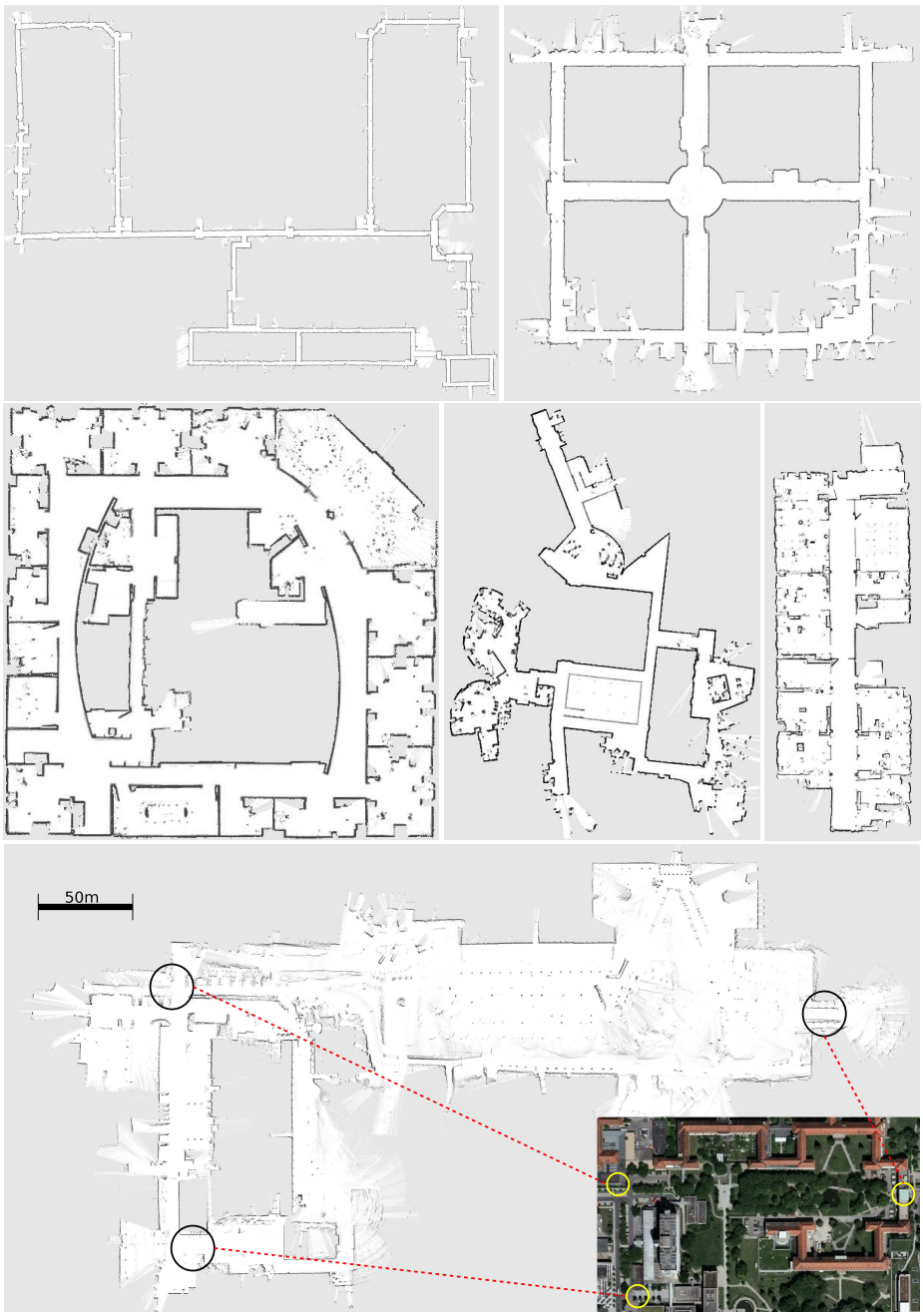


Fig. 8 Maps obtained by the reference datasets used to validate our metric. From top to bottom and left to right: MIT Killian Court (Boston), ACES Building (Austin), Intel Research Lab (Seattle), MIT CS Building (Boston), building 079 University of Freiburg, and the University Hospital in Freiburg. The depicted map of the University Hospital was obtained by using the background information extracted from the satellite images.

MIT. For addressing outdoor environments, we recorded a new dataset at the park area of the University Hospital, Freiburg. To give a visual impression of the scanned environments, Figure 8 illustrates maps obtained by executing state-of-the-art SLAM algorithms [Grisetti *et al*., 2007b; 2007c; Olson, 2008]. All datasets, the manually verified relations, and map images are available online at:
`http://ais.informatik.uni-freiburg.de/slamevaluation/`

8 Experimental Evaluation

This evaluation is designed to illustrate the properties of our method. We selected three popular mapping techniques, namely scan matching, a Rao-Blackwellized particle filter-based approach, and a graph-based solution to the SLAM problem and processed the datasets discussed in the previous section.

We provide the scores obtained 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. In addition, we also present sub-optimally corrected trajectories in this section to illustrate how inconsistencies affect the score of the metric. We will show that our error metric is well-suited for benchmarking and this kind of evaluation.

8.1 Evaluation of Existing Approaches using the Proposed Metric

In this evaluation, we considered the following mapping approaches:

Scan Matching: Scan matching is the computation of the incremental, open loop maximum likelihood trajectory of the robot by matching consecutive scans [Lu and Milios, 1994; Censi, 2006]. In small environments, a scan matching algorithm is generally sufficient to obtain accurate maps with a comparably small computational effort. However, the estimate of the robot trajectory computed by scan matching is affected by an increasing error which becomes visible whenever the robot reenters in known regions after visiting large unknown areas (loop closing or place revisiting).

Grid-based Rao-Blackwellized Particle Filter (RBPF) for SLAM: We use the RBPF implementation described in [Grisetti *et al*., 2007b; Stachniss *et al*., 2007b] which is available online [Stachniss *et al*., 2007a]. 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. The approach uses an informed proposal distribution for particle generation that is optimized to laser range data. In the evaluation presented here, we used 50 particles. Note that a higher number of samples may improve the performance of the algorithm.

Graph Mapping: This approach computes a map by means of graph optimization [Grisetti *et al*., 2007c]. The idea is to construct a graph out of the sequence of measurements. Every node in the graph represents a pose along the trajectory taken by the robot and the corresponding measurement obtained at that pose. Then, a least square error minimization approach is applied to obtain the most-likely configuration of the graph. In general, it is non-trivial to find the constraints, often referred to as the data association problem. Especially in symmetric environments or in situations with large noise, the edges in the

Table 1 Quantitative results of different approaches/datasets on the translation error as well as the corresponding standard deviation and the maximum error. [1] scan matching has been applied as a preprocessing step to improve the odometry.

Translational error m (abs) / m^2 (sqr)	Scan Matching	RBPF (50 part.)	Graph Mapping
Aces			
Eq. 4 using absolute errors	0.173 ± 0.614	0.060 ± 0.049	0.044 ± 0.044
Eq. 4 using squared errors	0.407 ± 2.726	0.006 ± 0.011	0.004 ± 0.009
Maximum absolute error of a relation	4.869	0.433	0.347
Intel			
Eq. 4 using absolute errors	0.220 ± 0.296	0.070 ± 0.083	0.031 ± 0.026
Eq. 4 using squared errors	0.136 ± 0.277	0.011 ± 0.034	0.002 ± 0.004
Maximum absolute error of a relation	1.168	0.698	0.229
MIT Killian Court			
Eq. 4 using absolute errors	1.651 ± 4.138	0.122 ± 0.386[1]	0.050 ± 0.056
Eq. 4 using squared errors	19.85 ± 59.84	0.164 ± 0.814[1]	0.006 ± 0.029
Maximum absolute error of a relation	19.467	2.513[1]	0.765
MIT CSAIL			
Eq. 4 using absolute errors	0.106 ± 0.325	0.049 ± 0.049[1]	0.004 ± 0.009
Eq. 4 using squared errors	0.117 ± 0.728	0.005 ± 0.013[1]	0.0001 ± 0.0005
Maximum absolute error of a relation	3.570	0.508[1]	0.096
Freiburg bldg 79			
Eq. 4 using absolute errors	0.258 ± 0.427	0.061 ± 0.044[1]	0.056 ± 0.042
Eq. 4 using squared errors	0.249 ± 0.687	0.006 ± 0.020[1]	0.005 ± 0.011
Maximum absolute error of a relation	2.280	0.856[1]	0.459
Freiburg Hospital			
Eq. 4 using absolute errors	0.434 ± 1.615	0.637 ± 2.638	0.143 ± 0.180
Eq. 4 using squared errors	2.79 ± 18.19	7.367 ± 38.496	0.053 ± 0.272
Maximum absolute error of a relation	15.584	15.343	2.385
Freiburg Hospital, only global relations (see text)			
Eq. 4 using absolute errors	13.0 ± 11.6	12.3 ± 11.7	11.6 ± 11.9
Eq. 4 using squared errors	305.4 ± 518.9	288.8 ± 626.3	276.1 ± 516.5
Maximum absolute error of a relation	70.9	65.1	66.1

graph may be wrong or imprecise and thus the resulting map may yields inconsistencies. In our current implementation of the graph mapping system, we followed the approach of Olson [2008] to compute constraints.

For our evaluation, we manually extracted the relations for all datasets mentioned in the previous section. The manually extracted relations are available online, see Section 7. We then carried out the mapping approaches and used the corrected trajectory for computing the error according to our metric. Please 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, see, for example, [Pfaff *et al.*, 2006]. In our evaluation, however, we provide both terms separately for a better transparency of the results.

We processed all benchmark datasets from Section 7 using the algorithms listed above. A condensed view of each algorithm's performance is given by the averaged error over all relations. In Table 1, we give an overview on the translational error of the various algorithms, while Table 2 shows the rotational error. Comparing two algorithms can be done by comparing the values given in the tables, namely the maximum error as well as the average error. It can be seen that the more advanced algorithms (Rao-Blackwellized particle filter

Table 2 Quantitative results of different approaches/datasets on the rotational error as well as the corresponding standard deviation and the maximum error. [1] scan matching has been applied as a preprocessing step to improve the odometry.

Rotational error *deg* (abs) / deg^2 (sqr)	Scan Matching	RBPF (50 part.)	Graph Mapping
Aces			
Eq. 4 using absolute errors	1.2 ± 1.5	1.2 ± 1.3	0.4 ± 0.4
Eq. 4 using squared errors	3.7 ± 10.7	3.1 ± 7.0	0.3 ± 0.8
Maximum absolute error of a relation	12.1	7.9	3.5
Intel			
Eq. 4 using absolute errors	1.7 ± 4.8	3.0 ± 5.3	1.3 ± 4.7
Eq. 4 using squared errors	25.8 ± 170.9	36.7 ± 187.7	24.0 ± 166.1
Maximum absolute error of a relation	4.5	34.7	6.4
MIT Killian Court			
Eq. 4 using absolute errors	2.3 ± 4.5	0.8 ± 0.8[1]	0.5 ± 0.5
Eq. 4 using squared errors	25.4 ± 65.0	0.9 ± 1.7[1]	0.9 ± 0.9
Maximum absolute error of a relation	21.6	7.4[1]	5.4
MIT CSAIL			
Eq. 4 using absolute errors	1.4 ± 4.5	0.6 ± 1.2[1]	0.05 ± 0.08
Eq. 4 using squared errors	22.3 ± 111.3	1.9 ± 17.3[1]	0.01 ± 0.04
Maximum absolute error of a relation	26.3	18.2[1]	0.8
Freiburg bldg 79			
Eq. 4 using absolute errors	1.7 ± 2.1	0.6 ± 0.6[1]	0.6 ± 0.6
Eq. 4 using squared errors	7.3 ± 14.5	0.7 ± 2.0[1]	0.7 ± 1.7
Maximum absolute error of a relation	9.9	6.4[1]	5.4
Freiburg Hospital			
Eq. 4 using absolute errors	1.3 ± 3.0	1.3 ± 2.3	0.9 ± 2.2
Eq. 4 using squared errors	10.9 ± 50.4	7.1 ± 42.2	5.5 ± 46.2
Maximum absolute error of a relation	27.4	28.0	29.6
Freiburg Hospital, only global relations (see text)			
Eq. 4 using absolute errors	6.3 ± 5.2	5.5 ± 5.9	6.3 ± 6.2
Eq. 4 using squared errors	66.1 ± 101.4	64.6 ± 144.2	77.2 ± 154.8
Maximum absolute error of a relation	27.3	35.1	38.6

and graph mapping) usually outperform scan matching. This is mainly caused by the fact that scan matching only locally optimizes the result 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. It should also be noted that for the outdoor dataset (Freiburg hospital), the RBPF mapper was not able to close the large loop and therefore was substantially worse than the graph mapper.

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 can also provide insights as to why it fails. Inspecting those situations in correlation with the map helps to understand the properties of algorithms and gives valuable insights on its capabilities. For three datasets, a detailed analysis using these plots is presented in Section 8.2 to Section 8.4. The overall analysis provides the intuition that our metric is well-suited for evaluating SLAM approaches.

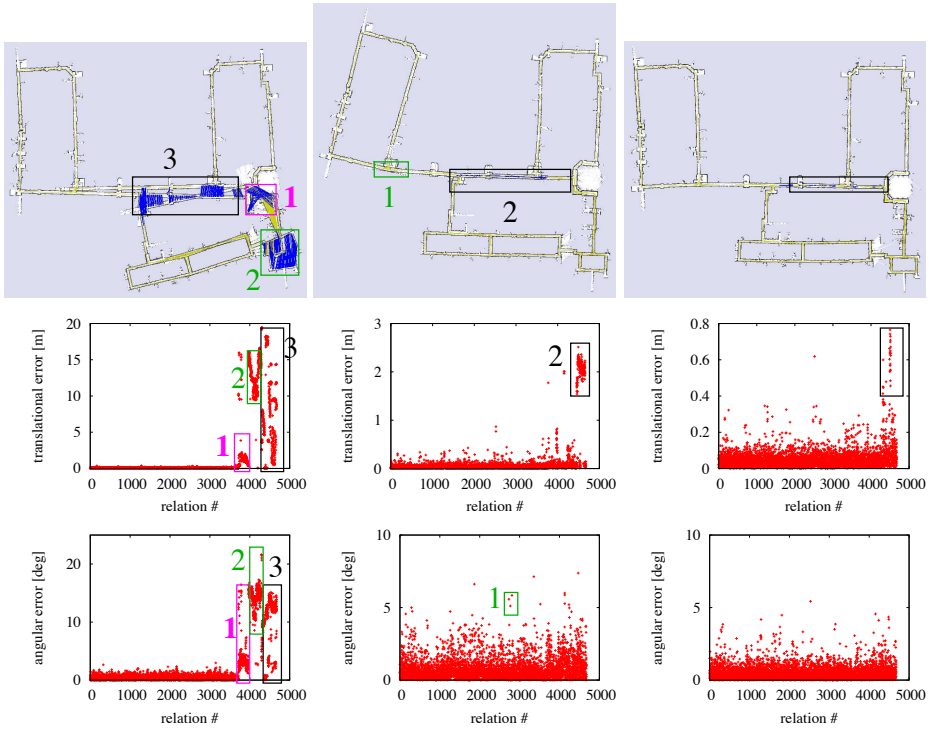


Fig. 9 This figure illustrates our metric applied to the MIT Killian Court dataset. The reference relations are depicted in light yellow, while the relations marked in the plots are shown in dark blue. The left column shows the results of pure scan-matching, the middle column the result of a RBPF-based technique with 50 samples, and the right column shows the result of a graph-based approach. The regions marked in the map correspond to regions in the error plots having high error. Due to its inability of dealing with loop closures scan matching has a high error when revisiting known regions. However, the absence of significant structure along the corridors for scan registration is an issue for both the graph-based and the RBPF approach. All in all, the graph-based approach outperforms the other methods.

8.2 MIT Killian Court

The MIT Killian Court dataset has been acquired in a large indoor environment, where the robot mainly observed corridors lacking structures that support accurate pose correction. The robot traverses multiple nested loops – a challenge especially for the RBPF-based technique. We extracted close to 5,000 relations between nearby poses that are used for evaluation. Figure 9 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).

8.3 Freiburg Indoor Building 079

The building 079 of the University of Freiburg is an example for an indoor office environment. The building consists of one corridor which connects the individual rooms. Figure 10 depicts the results of the individual algorithms (scan matching, RBPF, graph-based). In the first row of Figure 10, the relations having a translational error greater than 0.15 m are highlighted in 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 10 first column, second and third row). As can be seen from the error plots, the relations with a number greater than 1,000 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).

8.4 Freiburg University Hospital

This dataset consists of 2D and 3D laser range data obtained with one statically mounted SICK scanner and one mounted on a pan-tilt unit. The robot was steered through a park area that contains a lot of bushes and which is surrounded by buildings. Most of the time, the robot was steered along a bike lane with cobble stone pavement. The area is around 500 m by 250 m in size.

Figure 11 depicts the three mapping results, one obtained with scan matching (left), one with the RBPF (middle), and one with the graph mapper (right). The quality of all maps is lower than the quality of the map depicted in Figure 8. The reason for that is that while building the map in Figure 8, we also used the satellite image data which was not available for the algorithms under evaluation.

Based on the error plots in Figure 11 as well as the overall score depicted in the tables, we can see that graph mapping outperforms the RBPF and scan matching. The RBPF was not able to close the large loop and therefore performed similar to scan matching. However, note that in most parts of the map, the results of the scan matcher and RBPF are comparable to the one of graph mapping. Significant differences can be observed in the areas labeled as 1 and 3. Here, the two approaches fail to build a consistent map which is the reason for the significantly higher overall error.

In the area labeled as 4, the results of all algorithms yield matching errors. In that area, the robot makes a 180 degree turn and looks towards the open park area where almost no structure that would allow for pose correction is visible. Therefore, none of the tested algorithms was able to build a perfect map here.

Note that for this dataset, we present two alternative sets of relations. One using only local relations based on the sensor range. In addition, we provide a set where the relations are generated for pairs of randomly sampled poses. This set should be used if global consistency is desired. A comparison between the two data sets can be seen in Figure 12. The histograms count relations based by the difference in the time indices of the connected poses. As can be seen from the left image, using local relations based on the sensor range leads to a peaked histogram, since the relations only cover a small time frame. Additional minor peaks occur if the robot re-visits a region. In contrast, the set of relations used to evaluate the global

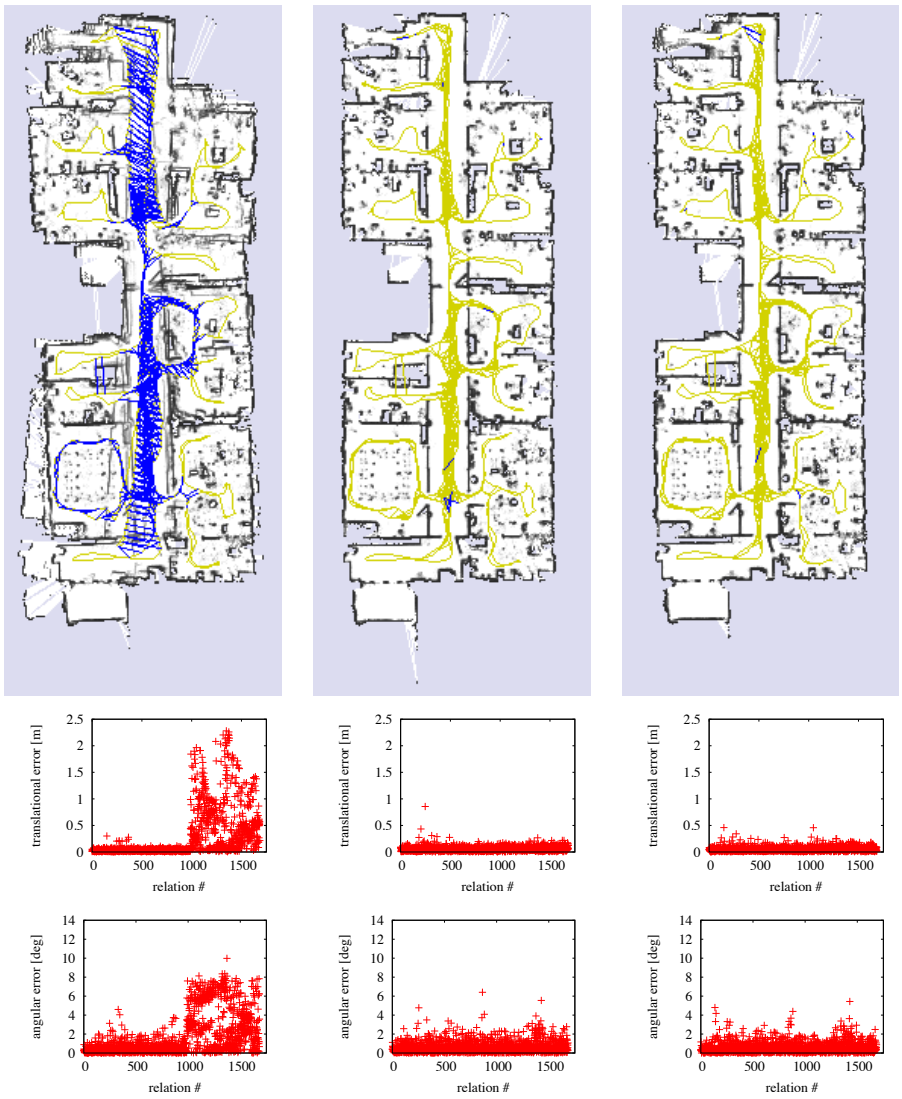


Fig. 10 This figure shows the Freiburg Indoor Building 079 dataset. Each column reports the results of one approach. Left: scan-matching, middle: RBPF and right a graph based algorithm. Within each column, the top image shows the map, the middle plot is the translational error and the bottom one is the rotational error.

consistency of the map is less peaked. Here, the relations uniformly sub-sample all available pairwise combinations of robot poses.

8.4.1 Utilizing Additional Relations

To illustrate that it is possible to incorporate additional relations as claimed in Section 4.3, we added in a further experiment the satellite image data which was used to obtain the close-to-true pose information for the Freiburg hospital. These additional relations favor

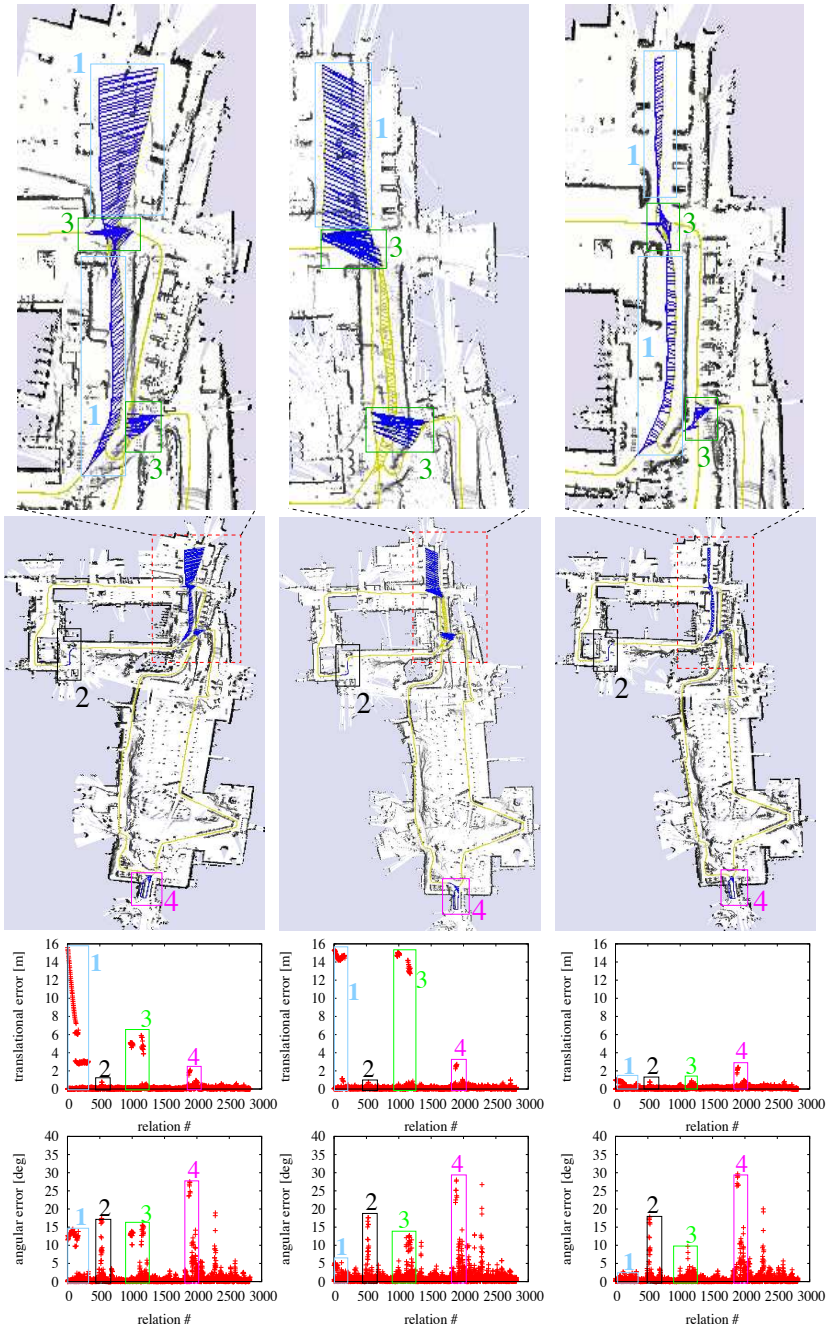


Fig. 11 Maps and error plots of the Freiburg University Hospital. Each column reports the results of one approach. Left: scan-matching, middle: RBPF and right a graph based algorithm. The second row depicts the created maps. The first row shows close-ups of areas in these maps. The error plots in the middle are regarding translation and on the bottom regarding rotation. There are corresponding areas marked in the plots and the maps.

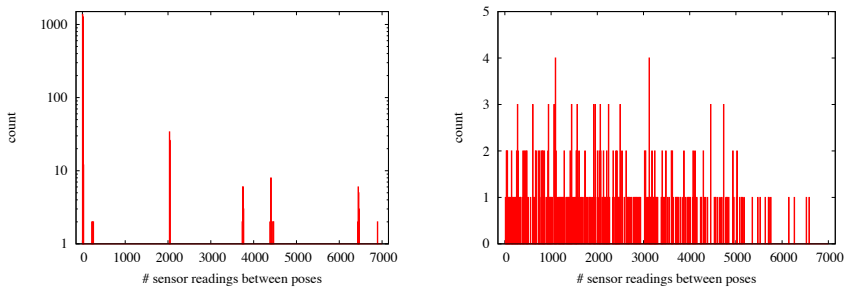


Fig. 12 Comparison of local relations with global relations based on their histograms. The abscissa shows the number of sensor readings between the two positions in a relation (a bin size of 10 was chosen). On the left side the histogram for the local relation set is shown, while the right side displays the global relations.

approaches that are able to generate global consistency as it is desired for robots that, for example, build blueprints.

The resulting scores for such a setting are given in the last rows of Table 1 and Table 2, respectively. As expected, the error in case of the evaluation including global relations is higher.

8.5 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 obtained from visual inspection 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 relation 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.

We furthermore encourage authors to evaluate their algorithms based on multiple datasets and not just using a single in order to illustrate the generality of the method and not being optimized for a single dataset.

9 Conclusion

In this paper, we have presented a framework for analyzing the results of SLAM approaches that allows for creating 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 does not rely on a global reference frame. This overcomes serious shortcomings of approaches using a global reference frame to compute the error. The metric even allows for comparing 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. We present relations for self-recorded datasets with laser range finder data as well as for a set of log-files that are frequently used in the SLAM community to evaluate approaches. In addition, we provide

an error analysis for three mapping systems including two modern laser-based SLAM approaches, namely a graph-based approach as well as system based on a Rao-Blackwellized particle filter. We believe that our results are a valuable benchmark for SLAM researchers since we provide a framework that allows for objectively and comparably easy analyzing the results of SLAM systems.

Acknowledgments

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, FP7-231888-EUROPA, and FP6-IST-045388-INDIGO. 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.

References

[Amigoni *et al*., 2007] 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.

[Balaguer *et al*., 2007] 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.

[Bar-Shalom *et al*., 2001] Y Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Application to Tracking and Navigation*. Jonh Wiley and Sons, 2001.

[Bonarini *et al*., 2006] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos. Rawseeds a project on SLAM benchmarking. In *Proceedings of the IROS'06 Workshop on Benchmarks in Robotics Research*, 2006. available online at http://www.robot.uji.es/EURON/pdfs/Lecture Notes IROS06.pdf.

[Bosse *et al*., 2003] M. Bosse, P.M. Newman, J.J. Leonard, and S. Teller. An ALTAS framework for scalable mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1899–1906, Taipei, Taiwan, 2003.

[Burgard *et al*., 2009] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. D. Tardós. A comparison of slam algorithms based on a graph of relations. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009. To appear.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

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

[Darpa, 2007] Darpa. Darpa Urban Challenge, 2007. http://www.darpa.mil/grandchallenge/.

[Dellaert *et al*., 1998] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Leuven, Belgium, 1998.

[Dellaert, 2005] F. Dellaert. Square Root SAM. In *Proc. of Robotics: Science and Systems (RSS)*, pages 177–184, Cambridge, MA, USA, 2005.

[Dissanayake *et al*., 2000] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1009–1014, 2000.

[Doucet *et al*., 2001] A. Doucet, N. de Freitas, and N. Gordan, editors. *Sequential Monte-Carlo Methods in Practice*. Springer Verlag, 2001.

[Duckett *et al*., 2002] T. Duckett, S. Marsland, and J. Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287 – 300, 2002.

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

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

[Estrada *et al*., 2005] C. Estrada, J. Neira, and J.D. Tardós. Hierachical SLAM: Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.

[Eustice *et al*., 2005a] 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.

[Eustice *et al*., 2005b] R. Eustice, M. Walter, and J.J. Leonard. Sparse extended information filters: Insights into sparsification. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 641–648, Edmonton, Cananda, 2005.

[Frese *et al*., 2005] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):1–12, 2005.

[Frese, 2006] U. Frese. Treemap: An $o(logn)$ algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2):103–122, 2006.

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

[Früh and Zakhor, 2004] C. Früh and A. Zakhor. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, 60:5–24, 2004.

[Grisetti *et al*., 2007a] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3D. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, 2007.

[Grisetti *et al*., 2007b] 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.

[Grisetti *et al*., 2007c] 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.

[Gutmann and Konolige, 1999] 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.

[Hähnel *et al*., 2003] 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 Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 206–211, 2003.

[Hermosillo *et al*., 2003] 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.

[Hoover *et al*., 1996] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. K. Bowyer, D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.

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

[Julier *et al*., 1995] 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.

[Kaess *et al*., 2007] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast incremental smoothing and mapping with efficient data association. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

[Kümmerle *et al*., 2009] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard. Large scale graph-based SLAM using aerial images as prior information. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[Leonard and Durrant-Whyte, 1991] 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.

[Lu and Milios, 1994] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 935–938, 1994.

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

[Montemerlo *et al*., 2003] 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.

[Nüchter *et al*., 2005] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6d SLAM with approximate data association. In *Proc. of the 12th Int. Conference on Advanced Robotics (ICAR)*, pages 242–249, 2005.

[Olson *et al*., 2006] 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.

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

[Pfaff *et al*., 2006] P. Pfaff, W. Burgard, and D. Fox. Robust monte-carlo localization using adaptive likelihood models. In H.I. Christiensen, editor, *European Robotics Symposium 2006*, volume 22 of *STAR Springer tracts in advanced robotics*, pages 181–194. Springer-Verlag Berlin Heidelberg, Germany, 2006.

[Ranganathan *et al*., 2007] A. Ranganathan, M. Kaess, and F. Dellaert. Loopy sam. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2007.

[RoboCup Federation, 2009] RoboCup Federation. RoboCup Competitions, 2009. http://www.robocup.org.

[Scharstein and Szeliski, 2002] D. Scharstein and R. Szeliski. Middlebury stereo vision page, 2002. http://www. middlebury. edu/stereo.

[Smith and Cheeseman, 1986] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[Smith *et al*., 1990] 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.

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

[Stachniss *et al*., 2007b] 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.

[Symeo GmbH, 2008] Symeo GmbH. http://www.symeo.de, 2008.

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

[Thrun *et al*., 2004] 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.

[Thrun *et al*., 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

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

[Torralba *et al*., 2007] A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: the open annotation tool, 2007. http://labelme.csail.mit.edu/.

[Uhlmann, 1995] J. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. PhD thesis, University of Oxford, 1995.

[Wulf *et al*., 2008] 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.

[Yguel *et al*., 2007] 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.