Benchmarking Urban Six-Degree-of-Freedom Simultaneous Localization and Mapping

Oliver Wulf
Institute for Systems Engineering (ISE/RTS)
Leibniz Universität Hannover
Appelstraße 9A
D-30167 Hannover, Germany

Andreas Nüchter and Joachim Hertzberg
Institute of Computer Science
University of Osnabrück
Albrechtstraße 28
D-49069 Osnabrück, Germany
e-mail: nuechter@informatik.uni-osnabrueck.de

Bernardo Wagner
Institute for Systems Engineering (ISE/RTS)
Leibniz Universität Hannover
Appelstraße 9A
D-30167 Hannover, Germany

Received 6 September 2007; accepted 31 December 2007

Quite a number of approaches for solving the simultaneous localization and mapping (SLAM) problem exist by now. Some of them have recently been extended to mapping environments with six-degree-of-freedom poses, yielding 6D SLAM approaches. To demonstrate the capabilities of the respective algorithms, it is common practice to present generated maps and successful loop closings in large outdoor environments. Unfortunately, it is nontrivial to compare different 6D SLAM approaches objectively, because ground truth data about the outdoor environments used for demonstration are typically unavailable. We present a novel benchmarking method for generating the ground truth data based on reference maps. The method is then demonstrated by comparing the absolute performance of some previously existing 6D SLAM algorithms that build a large urban outdoor map. © 2008 Wiley Periodicals, Inc.

Journal of Field Robotics 25(3), 148–163 (2008) © 2008 Wiley Periodicals, Inc.
Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/rob.20234

1. THE PROBLEM: DETERMINING MAP QUALITY IN SLAM

Algorithms for solving the robotic simultaneous localization and mapping (SLAM) problem are a key scientific issue in mobile robotics research. Solutions to SLAM are of core importance in providing mobile robots with the ability to operate in true autonomy. SLAM algorithms integrate robot action and sensor readings and exploit the fact that previously mapped areas are recognized.

Popular mapping algorithms work with three-degree-of-freedom (DoF) pose estimates, that is, robot poses are represented as $P = (x, y, \theta_z)$. This choice is appropriate for indoor environments but not for mapping many outdoor environments, which, in general, require using poses in six DoF in order to cope with elevation, pitch, and roll. Therefore, the respective approaches, called 6D SLAM (Nüchter, Surmann, Lingemann, Hertzberg, & Thrun, 2004), consider the six-DoF pose $V = (x, y, z, \theta_x, \theta_y, \theta_z)$ of the mobile robot with three position coordinates and roll, pitch, and yaw angles.

How should results of different SLAM algorithms be compared? An important criterion is clearly the quality of the maps that they would deliver for some test environment. Now what is map quality? Clearly, it is important that a map be consistent, that is, it should contain no artifacts and doublets but all perceivable structures. Ideally, it should be correct, too, that is, it should represent metrical information about the environment in a correct way.

Although it is cumbersome but feasible to determine the correctness of typical planar indoor maps (use a yardstick for making independent measurements), it is practically impossible to get this type of independent ground truth information for large three-dimensional (3D) outdoor scenes. So on the one hand, generating 3D maps of large environments, using 3D or 6D poses, has received much attention recently (e.g., Cole & Newman, 2006; Pfaff et al., 2007; Triebel, Pfaff, & Burgard; 2006). On the other hand, a framework for benchmarking these large experiments is still missing. Well-known SLAM sites offer only data sets, for example, Radish: The Robotics Data Set Repository (Howard & Roy, 2007) or algorithms, for example, OpenSLAM (Stachniss, Frese, & Grisetti, 2007) and no ground truth data. So given a map generated by some SLAM algorithm, we may judge whether the map "looks good," that is, is not obviously inconsistent, but there is no way to measure its quality with respect to independent reference data, let alone to determine its consistency with respect to ground truth. So comparing systematically the maps resulting from different SLAM approaches is futile. Mapping environments by conventional means, that is, using markers and geodetic surveying equipment, ends up with maps containing similar measures of uncertainty, and such guarantees are required by many mapping customers, for example, applications in facility management and architecture or in construction and maintenance of tunnels, mines, and pits. In contrast, current SLAM technology cannot come up with performance measures.

This paper presents a novel method for generating independent reference data for benchmarking outdoor SLAM approaches. In particular, we are aiming at full 6D SLAM approaches. Our procedure makes use of an independently available, accurate environment map (provided by the land registry office), a Monte Carlo localization (MCL) technique that matches sensor data against the reference map, and manual quality control.

To demonstrate the use of the benchmarking method, we apply it for determining the quality of a family of 6D SLAM approaches that have been published previously. To keep this paper self-sufficient, it does include sketches of these approaches; bear in mind, however, that the point of this paper is demonstrating the benchmarking method, not the SLAM approaches. The data used for the respective experiments were recorded using a mobile robot that was steered manually in natural terrain, over asphalt roads, sidewalks, cobblestone, etc., with a fast 3D scanner gauging the environment. A sketch of the demo environment is given in Figure 1.

The next section of this paper points to related work. Then, we describe the sensor system used in this work for generating large 3D maps and the two pairs of mapping algorithms used for demonstrating how to evaluate mapping results. After that, Section 4 presents the MCL-based benchmarking technique. In the end, we use this technique in a demo evaluation based on SLAM results from a data set consisting of 924 3D scans. Section 7 concludes.

2. RELATED WORK

2.1. Ground Truth Experiments

In doing experiments with ground truth reference, researchers aim at measuring the objective performance

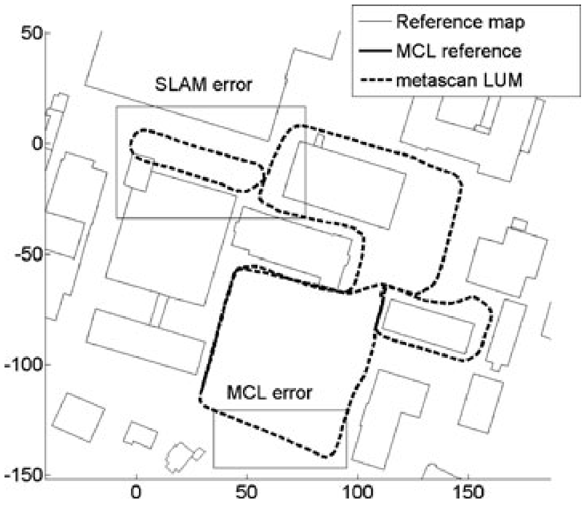

Figure 1. Performance measurement of SLAM algorithms using reference maps and MCL in an urban outdoor environment (University of Hannover campus; 3D scan indices 1–700). Distances are given in meters.

of a dedicated algorithm. On the basis of this benchmark, it is possible to give an experimental proof of the effectiveness of a new algorithm. Furthermore, measuring the performance of an algorithm allows it to be optimized and compared to other existing solutions.

Benchmarking is a common instrument in science. Good examples for successful performance measurements in computer science are available in the computer vision community. Several projects aim at providing image databases for other researchers (Hoover et al., 1996; Torralba, Murphy, & Freeman, 2007). These image databases are supplemented by ground truth images and algorithms that calculate performance metrics. In doing so, the community is able to make progress and to document its progress in fields such as image segmentation and object recognition.

Unfortunately, this kind of performance measurement is not widespread in the robotics community. Even though there are several ways of comparing the performance of robotic algorithms and systems, one basic step is to provide experimental data and results to other research groups. Up to now this is done only by small projects (Howard & Roy, 2007; Stachniss et al., 2007) or by individual researchers. Another way of comparing robotic systems is competitions such as RoboCup (Federation, 2007), ELROB (FGAN, 2007), or the Grand Challenge (DARPA, 2007). This kind of competition allows the level of system integration and the engineering skills of a certain team to be ranked, but it is not possible to measure the performance of a subsystem or a single algorithm.

Objective benchmarking of localization and mapping algorithms is achieved only by comparing experimental results against reference data. The practical problem is to generate the ground truth data. In computer vision, ground truth data are either available for synthetic images or need to be hand labeled. In the case of mobile robot navigation, one way of gathering ground truth data is the use of precise global positioning systems (RTK-GPS) (Guivant, Nebot, & Baiker, 2000). Unfortunately, these data are available only for open outdoor environments and not for urban outdoor environments or indoor environments. Another possibility is to use complex external measurement setups.

Simulation yields another benchmarking method for robotic algorithms, enabling researchers to perform experiments in defined conditions and to repeat these experiments. However, real life differs from simulation. Experiments involving sophisticated sensors such as cameras or laser scanners can be simulated only up to a certain level of accuracy, for example, capturing environments must regard surface properties such as material, local structures, and reflections. Therefore, using real robotic data sets is favored for benchmarking in this paper.

2.2. State of the Art in Metric Robotic Mapping

2.2.1. Planar Mapping

The state of the art for metric maps is dominated by probabilistic methods, in which the robot has probabilistic motion models and uncertain perception models. By integration of these two distributions with a Bayes filter, for example, Kalman or particle filter, it is possible to localize the robot. Mapping is often an extension to this estimation problem. Beside the robot pose, positions of landmarks are estimated. Closed loops, that is, a second encounter of a previously visited area of the environment, play a special role here: Once detected, they enable the algorithms to bound the error by deforming the mapped area to yield a topologically consistent model. However, there is no guarantee for a correct model. Several strategies exist for solving

SLAM. Thrun (2002) surveys existing techniques, that is, maximum likelihood estimation, expectation maximization, extended Kalman filter, or (sparsely extended) information filter SLAM. FastSLAM (Thrun, Fox, & Burgard, 2000) approximates the posterior probabilities, that is, robot poses, by particles.

SLAM in well-defined, planar indoor environments is considered solved. In principle, probabilistic methods are extendable to six DoF. However, to our knowledge no reliable feature extraction mechanisms or methods for reducing the computational cost of multihypothesis tracking procedures such as FastSLAM (which grows exponentially with the DoF) have been published.

2.2.2. Mapping Environments in Three Dimensions

Popular 3D environment mapping methods include the use of translated two-dimensional (2D) laser range finders. Thrun et al. (Thrun, Fox, & Burgard, 2000) use two 2D laser scanners for acquiring 3D data; one for planar localization and the second, vertically mounted scanner, for acquiring 3D data while moving. The precision of 3D data points depends, in addition on the precision of the scanner, critically on that pose estimation. These 3D mapping approaches are often used for navigation (Thrun, Montemerlo, & Aron, 2006).

An emerging research topic is 6D SLAM, that is, while mapping, the robot pose is represented with six DoF. In previous work, we used a 3D laser range finder in a stop-scan-match-go-process to create a 3D map of the environment by merging several 3D scans into one coordinate system (Nüchter et al., 2004; Surmann, Nüchter, Lingemann, & Hertzberg, 2004). Similar experiments have been made by Newman et al. (Newman, Cole, & Ho, 2006). A current trend in laser-based 6D SLAM is to overcome the stop-and-go fashion of scan acquisition by rotating or pitching the scanner while moving (Cole & Newman, 2006; Triebel et al., 2006; Wulf, Arras, Christensen, & Wagner, 2004). In their most recent work, Pfaff et al. (2007) employ two rotating SICK scanners for data acquisition, odometry, inertial measurement unit (IMU) and differential GPS (DGPS) positioning, a variant of the iterative closest point (ICP) algorithm, and a loop-closing procedure to map large urban environments in three dimensions.

Feature-based 6D SLAM methods are investigated by Frese (2007), who has adapted his fast treemap algorithm to six DoF, which, however, covers the least-square estimation core and no actual scan-data processing. Other pure 6D SLAM backends such as the tree optimizer by Olson et al. (Olson, Leonard, & Teller, 2006) are becomming available (Grisetti, Grzonka, Stachniss, Pfaff, & Burgard, 2007).

Among the category of feature-based 6D SLAM are the visual SLAM methods, that is, the MonoSLAM system of Davison et al. (Davison, Reid, Molton, & Stasse, 2007).

3. OUR REFERENCE METHODS FOR GENERATING LARGE URBAN 3D MAPS

This section presents the hardware and procedures used in this work as candidates for being benchmarked. They were published earlier, involving coauthors of the present paper; we give sketches here to keep the paper self-sufficient. Remember that the issue in the current paper is the benchmarking method, whose description follows in the next section.

3.1. 3D Range Sensor

The sensor that has been employed for the experiments is a fast 3D laser range scanner, developed at the Leibniz Universität Hannover (Fig. 2). As there is no commercial 3D laser scanner available that meets the requirements of mobile robots, it is common practice to assemble 3D sensors out of standard 2D laser range sensors and additional servo drives.

The specialties of our RTS/ScanDrive are a number of optimizations that are made to allow fast scanning. One mechanical optimization is the slip ring connection for power and data. This connection allows continuous 360-deg scanning without the accelerations and high power consumption that are typical for panning systems. Even more important than the mechanical and electrical improvements is the precise synchronization between the 2D laser data, servo drive data, and the wheel odometry. Owing to this good synchronization, it is possible to compensate systematic measurement errors and to measure accurate 3D point clouds even with a moving robot. Detailed descriptions of these 3D scanning methods and optimizations are published in Wulf and Wagner (2003).

Having the optimizations described above, the limiting factor in building a faster 3D laser scanner is the maximal number of 13,575 (75 × 181) points that can be measured with a SICK LMS 2×× sensor in 1 s. The only way of building faster SICK LMS

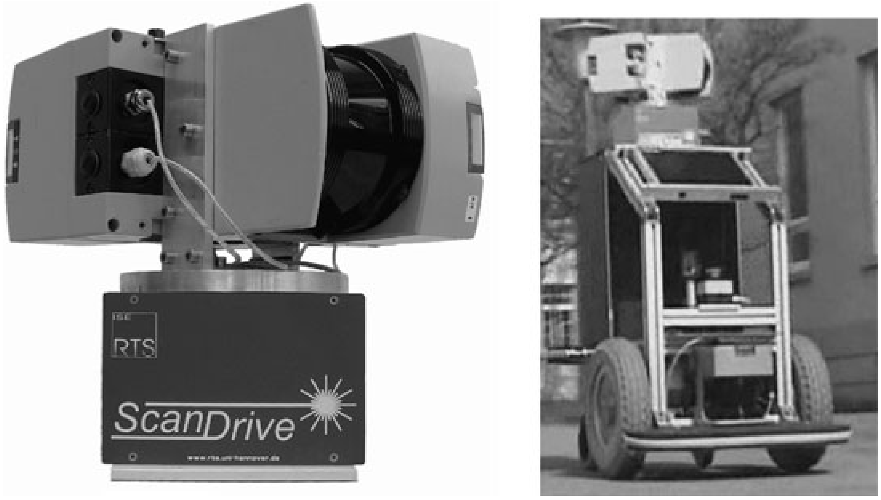

Figure 2. Left: 3D laser range sensor RTS/ScanDriveDuo. It takes a full 3D scan of 32,580 points in 1.2 s. Right: The mobile robot Erika.

2× ×–based 3D scanners is to use multiple 2D measurement devices (Triebel et al., 2006). For this reason we present the RTS/ScanDriveDuo in this paper. This 3D scanner makes use of two SICK LMS 291 2D laser scanners. Thus the measurement time for 3D scans with 2-deg horizontal and 1-deg vertical angle resolution is reduced to 1.2 s. In this case one 3D scan measured in 1.2 s consists of 32,580 (180 × 181) 3D points.

3.2. The Mobile Robot Erika

The mobile service robot Erika is built out of the Modular Robotics Toolkit (MoRob-Kit). The overall size (L × W × H) of Erika is 95 × 60 × 120 cm. With its differential drive motors, it is able to drive up to 1.6 m/s in indoor and urban outdoor environments. The battery capacity is designed to supply the electric wheelchair motors, sensors, and a 700-MHz embedded personal computer for at least 2 h or 5 km.

In addition to the 3D laser scanner, the mobile robot is equipped with wheel odometry, a three-axis gyroscope, and a low-cost SiRF III GPS receiver. The measured data of the wheel odometry and the gyroscope are fused to result in the OdometryGyro that is used as the internal sensor for both MCL and SLAM. In contrast to the odometry sensor, the GPS receiver does not influence either the MCL or the SLAM results. It is logged only to have another laser-independent reference.

3.3. 6D SLAM with ICP-Based Scan Matching

We use the well-known ICP algorithm (Besl & McKay, 1992) to calculate the transformation while the robot is acquiring a sequence of 3D scans. The ICP algorithm calculates iteratively the point correspondence. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation (R, t) for minimizing the equation

$$E(R, t) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} ||m_i - (Rd_j + t)||^2, \qquad (1)$$

where N_m and N_d, are the numbers of points in the model set M or data set D, respectively, and $w_{j,i}$ are the weights for a point match. The weights are assigned as follows: $w_{i,j} = 1$, if m_i is the closest point to d_j within a close limit, $w_{i,j} = 0$ otherwise. The assumption is that the point correspondences are correct in the last iteration. In each iteration, the transformation is calculated by the quaternion-based method of Horn (1987).

To digitalize environments without occlusions, multiple 3D scans have to be registered. Consider a robot traveling along a path and traversing $n + 1$ 3D scan poses $V_0, \ldots, V_n$. A first straightforward method for aligning several 3D scans taken from the poses $V_0, \ldots, V_n$ is pairwise ICP, that is, matching the scan taken from pose V_1 against the scan from pose V_0, matching the scan taken from V_2 against the scan from pose V_1, and so on. Here the model set M is formed from the 3D data from pose V_{i-1} and the data set D that of the pose V_i for all $i \in [1, n]$. A second plausible method is to form of all previously acquired 3D scans a so-called metascan and match the last acquired one against this metascan. This method is called metascan ICP. Here, the model set M consists of the union of the 3D scans from the poses $V_0, \ldots, V_{i-1}$ and the data set D that of pose V_i, for all $i \in [1, n]$.

3.4. 6D SLAM with Global Relaxation

Both pairwise ICP and metascan ICP correct the robot pose estimates, but registration errors sum up. SLAM algorithms use loop closing to bound these errors. If two estimated robot poses $V_i = (x_i, y_i, z_i, \theta_{xi}, \theta_{yi}, \theta_{zi})$ and $V_j = (x_j, y_j, z_j, \theta_{xj}, \theta_{yj}, \theta_{zj})$ are close enough, that is, their Euclidean distance falls below a threshold (here 5 m), then we assume that these scans overlap and are matchable. To a graph initially containing the sequence of all poses $(V_0, V_1), (V_1, V_2), \ldots, (V_{n-1}, V_n)$, the edge (V_i, V_j) is added. While processing the scans with pairwise ICP or metascan matching, we detect closed loops using this simple distance criterion. Once detected, a six-DoF graph optimization algorithm for global relaxation based on the method of Lu and Milios (1997) is employed, namely Lu and Milios–style SLAM (LUM). This is a variant of GraphSLAM. Details of the six-DoF optimization, that is, how the matrices have to be filled, can be found in Borrmann, Elseberg, Lingemann, Nüchter, and Hertzberg (in press); thus we give only a brief overview here:

Given a network with $n+1$ nodes $X_0, \ldots, X_n$ representing the poses $V_0, \ldots, V_n$, and the directed edges $D_{i,j}$, we aim at estimating all poses optimally to build a consistent map of the environment. The directed edge $D_{i,j}$ represents the change of the pose $(x, y, z, \theta_x, \theta_y, \theta_z)$ that is necessary to transform one pose V_i into V_j, that is, $V_i = V_j \oplus D_{i,j}$, thus transforming two nodes of the graph. For simplicity, the approximation that the measurement equation is linear is made, that is,

$$D_{i,j} = X_i - X_j. \tag{2}$$

A detailed derivation of the linearization is given in Borrmann et al. (in press). An error function is formed such that minimization results in improved pose estimations:

$$W = \sum_{(i,j)} (D_{i,j} - \bar{D}_{i,j})^T C_{i,j}^{-1} (D_{i,j} - \bar{D}_{i,j}), \tag{3}$$

where $\bar{D}_{i,j} = D_{i,j} + \Delta D_{i,j}$ models random Gaussian noise added to the unknown exact pose $D_{i,j}$. This representation involves resolving the nonlinearities resulting from the additional roll and pitch angles by Taylor expansion. The covariance matrices $C_{i,j}$ describing the pose relations in the network are computed based on the paired closest points (Borrmann et al., in press). The error function equation (3) has a quadratic form and is therefore solved in closed form by sparse Cholesky decomposition. The algorithm optimizes Eq. (3) gradually by iterating the following three steps: First, for every network link the corresponding covariance is computed based on the point correspondences of the scan matching. Then the error function (3) is minimized by solving a linear system of equations. In the third step, the local transformations are applied to the poses, resulting in improved pose estimates. The iterative fashion of the algorithm, including graph reestimation, is the cause of the high quality of the resulting maps (Borrmann et al., in press).

Using the global optimization, two more strategies have been implemented: In pairwise LUM, we use pairwise matching of scans for initially estimating the robot poses. After a loop has been closed, the global relaxation to all previously acquired scans is applied. In metascan LUM, every new scan is initially matched against all previously acquired scans. In both algorithms, global relaxation is started after a closed loop is detected. The relaxation considers all previously acquired scans.

3.5. Mapping Strategies

Figure 3 depicts how the mapping strategies are interleaved. 6D SLAM is the result of a six-DoF ICP algorithm combined with the extension of Lu/Milios Scan Matching to six DoF as global relaxation. The SLAM backend uses fast matrix computations exploiting the sparse structure of the corresponding SLAM graphs (Davis, 2006). Other backends such as Olson's graph optimization, extended to six DoF (Kaess, Ranganathan, & Dellaert, 2007; Olson et al., 2006) or treemap by Frese (2007), might be used as well. Using different paths in Figure 3, the different mapping strategies are created.

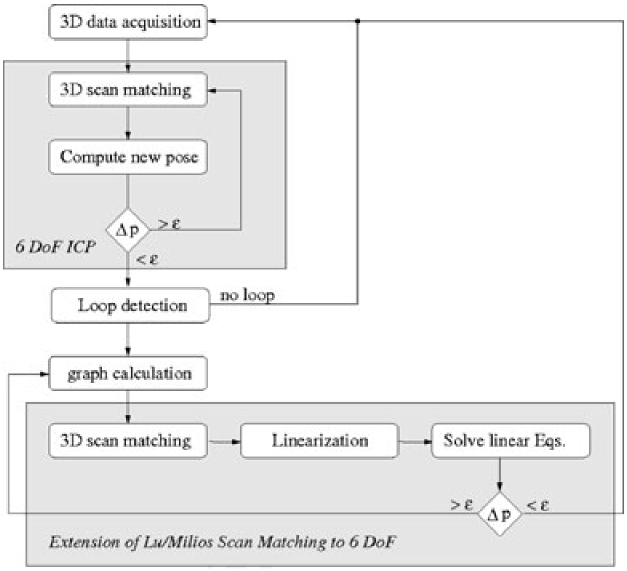

Figure 3. Mapping algorithm overview. Four strategies for mapping can be created from this procedure. For algorithmic details see Borrmann et al. (in press).

Animations of the four mapping strategies, pairwise ICP, metascan ICP, pairwise LUM, and metascan LUM, are given in the accompanying video (available as a supplementary file at http://www.interscience.wiley.com/jpages/1556-4959/suppmat/) and on the following web page: http://kos.informatik.uni-osnabrueck.de/download/6DSLAMbenchmarking/. Note that the maps presented in the video are rotated about 190 deg. Note that the data set contains 928 3D scans and the total trajectory has a length of more than 1 km.

We did choose our own SLAM algorithm set for carrying out the proposed benchmark over an existing SLAM algorithm (Stachniss et al., 2007), because it is often the case that parameters of the algorithm require being adjusted for performing well on a specific data set.

4. THE BENCHMARKING TECHNIQUE

This paper introduces a new benchmarking technique for SLAM algorithms. The benchmark is based on the final SLAM results and a reference position that is obtained independently of the SLAM algorithm under test.

As highly accurate RTK-GPS receivers cannot be used in urban outdoor environments, we present a technique that is based on surveyed maps as they can be obtained from German land registry offices. The process of generating these ground truth reference positions can be divided into a MCL step that matches the sensor data to the highly accurate map and a manual quality control step to validate the MCL results.

As the SLAM algorithm under test and the MCL algorithm use the same sensor data, the SLAM results and the reference positions are not completely independent. On the other hand, global localization algorithms and incremental localization and mapping algorithms work differently. Incremental mapping algorithms like odometry and SLAM can suffer from accumulating errors and drift effects. However, pure localization algorithms eliminate these errors by continuously matching to an accurate given map. Therefore, the remaining error of the manually supervised reference position is at least an order of magnitude smaller then the discussed SLAM errors.

4.1. Reference Map

As part of their geo information system, the German land registration offices feature surveyed data of all buildings within Germany. The information about these building is stored in vector format in the so-called Automatisierte Liegenschaftskarte (ALK). The vector format contains lines that represent the outer walls of solid buildings. Each line is represented by two points with northing and easting coordinates in a Gauss–Krueger coordinate system. The upper error bound of all points stored in the ALK is specified to be 4 cm. Up to now, no further details about doors, windows, or balconies are available.

If no reference map is at hand, one can produce one from calibrated areal photographs or satellite images. The accuracy of such maps is usually in the range of 10 cm.

4.2. MCL

The MCL is a commonly used localization algorithm that is based on particle filtering (Fox, Thrun, Burgard, & Dellaert, 2001). As the theory of MCL is well understood, we focus on the sensor model that is used to match the 3D sensor data to the 2D reference map with this paper.

The key problem of matching a 3D laser scan to a 2D map is solved by using a method called Virtual 2D Scans (Wulf, Arras, Christensen, & Wagner, 2004).

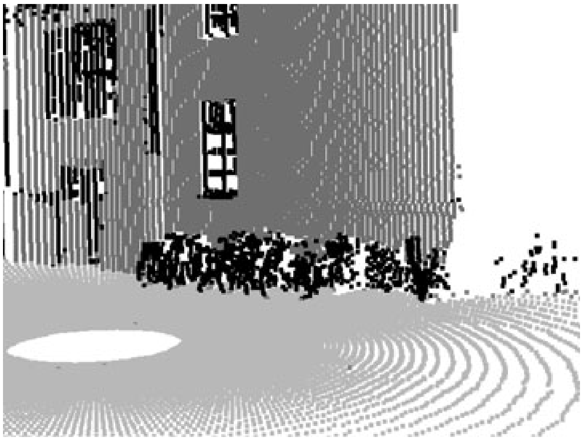

Figure 4. Classified 3D point cloud of an urban scene, featuring a building and bicycles in front (ground points, light gray; plain vertical surface points, dark gray; others, black).

The method splits up into two steps. The first step reduces the number of points in the 3D point cloud. The reduction step is based on the assumption that the reference map presents plain vertical walls. For this reason all 3D measurement points that do not belong to plain vertical surfaces need to be removed (Figure 4). Wulf, Brenneke, and Wagner (2004) describe a sequence of 3D segmentation and classification algorithms that is used to do this reduction in urban outdoor environments. By this means, the ground floor, vegetation, and small objects are removed from the 3D data. Measurement points on the outer walls of buildings and on other unmapped vertical obstacles remain. Thus, cars, people, and so on are removed in urban scenes, because the scanner usually acquires data points above these objects and the Virtual 2D Scan contains data points on buildings.

Having this reduced 3D point cloud, the second step of the Virtual 2D Scan method is a parallel projection of the remaining 3D points onto the horizontal plane. After this projection the z coordinate contains no information and can be removed. By this means, the Virtual 2D Scan has the same data format as a regular 2D scan. Thus it can be used as input data of a regular 2D MCL algorithm. To reduce the computational complexity of the successive MCL algorithm, the remaining measurement points are randomly downsampled. Experimental results show that fewer than 100 measurement points are needed for sufficient localization. Thus the average 3D point cloud with about 30,000 measurement points is reduced to a Virtual 2D Scan with only 100 points without losing information that is needed for localization in urban outdoor environments.

Owing to the 2D nature of the reference map and the used 2D MCL algorithm, only the three-DoF pose $P^{\mathrm{REF}} = (x, y, \theta_z)$ of the robot can be estimated. There is no reference information on the robot's elevation z. Furthermore, the roll and pitch components θ_x θ_y of the six-DoF robot pose cannot be estimated with this 2D method. These angles need to be measured and compensated with a gyro unit before the generation of the Virtual 2D Scans.

4.3. Manual Quality Control

As described above, the MCL is used to generate the reference path from the given map and sensor readings. But the output of the MCL cannot be stated as ground truth in general. Like all measured signals, the reference path has an error. The absolute error depends on the error of the reference map, the sensor errors, and the convergence and precision of the MCL algorithm. To be able to use the reference path as ground truth in our benchmark, the absolute error of the reference path needs to be at least one order of magnitude smaller than the error of the SLAM algorithms under test.

To ensure the quality of the reference path, a manual quality control step is integrated into the benchmark procedure. In this step a human supervisor monitors the sensor data, the particle distribution, and the matching results to decide whether the calculated reference path can be used as ground truth. The manual quality control decides which parts of the path are used for benchmarking, but the manual supervisor is not taking any influence on the sensor data of the MCL algorithm.

First, the sensor data need to be checked for a sufficient number of landmarks, namely, walls as given in the reference map. If there are not enough landmarks in the environment and thus in the Virtual 2D Scans, the MCL results depend only on odometry and are therefore inaccurate.

Second, the numerical condition of the particle filter needs to be monitored. As a particle filter presents only a sampled belief, an efficient distribution of the finite number of particles is essential for correct operation and precise results.

If these two criteria are fulfilled, the remaining error is white noise with zero mean and a small standard deviation depending on the map, the 3D sensor, and the discretization of the particle filter.

4.4. Benchmark Criteria

Up to this point, the MCL positions and SLAM positions are given in different coordinate systems. The MCL positions are given in the global Gauss–Krueger coordinate system of the reference map, and the SLAM positions are given in a local coordinate system that is centered in the robot's start position. To be able to compare the positioning results, the SLAM positions need to be transformed into the global coordinate system based on the known start position.

Having the trusted MCL reference P^{REF} and the SLAM results V^{SLAM} in the same coordinate system, it is possible to calculate objective performance metrics based on position differences. The first metric is based on the 2D Euclidean distance between the SLAM and MCL position:

$$e_i = \sqrt{\left(x_i^{\text{SLAM}} - x_i^{\text{REF}}\right)^2 + \left(y_i^{\text{SLAM}} - y_i^{\text{REF}}\right)^2}. \quad (4)$$

The second metric is based on the difference between the SLAM and MCL orientation:

$$e_{\theta,i} = \left|\theta_{z,i}^{\text{SLAM}} - \theta_{z,i}^{\text{REF}}\right|. \quad (5)$$

As the MCL position is given in three DoF only, the robot's elevation and roll and pitch angles cannot be tested.

To compare the performance of different SLAM algorithms on the same data set, it is possible to calculate scores like the standard deviation

$$\sigma = \sqrt{\frac{1}{n+1}\sum_{i=0}^{n} e_i^2}, \quad (6)$$

and the error maximum

$$e_{\max} = \max e_i. \quad (7)$$

Of course, these statistical tests can be done analogously on the orientation errors $e_{\theta,i}$, resulting in the scores (σ_θ and $e_{\theta,\max}$).

5. EXPERIMENTAL RESULTS

5.1. Experimental Setup

The presented experiment was carried out at the campus of the Leibniz Universität Hannover. The experimental robot platform that was used to collect the data was manually driven on the 1.242-km path closing a total of five small and large loops. On this path 924 full 3D scans were collected at an average robot speed of 4 km/h and a maximum speed of 6 km/h. In addition to the 3D laser data, wheel odometry and fused wheel/gyro odometry were stored with a data rate of 10 Hz. The position fixes of a low-cost GPS were logged with 1 Hz.

5.2. The Reference Path

The section of the ALK that is used as the reference map contains 28 buildings represented by 413 line segments. To avoid huge coordinate numbers, a constant offset of 5,806,400 m northing and 3,548,500 m easting is subtracted from all Gauss–Krueger coordinates. This offset corresponds to the position 52°23′58″ north, 9°42′41″ east in world geodetic system 1984 (WGS84) coordinates.

The reference path is calculated with a MCL based on 3D laser data and wheel/gyro odometry. The particle filter with 200 samples runs in parallel with the data acquisition and logging threads on a Pentium III 700-MHz processor. The localization results are plotted as a solid gray line in Fig. 1.

For the manual quality control the MCL results can be displayed offline. As a result of the manual quality control, it came out that the reference path for 3D scan index 198–241 cannot be used for benchmarking as not enough landmarks are visible in the 3D laser scan. As can be seen in Fig. 1 (MCL error box), the MCL results drift off from the true position. Starting with index 236 on the 3D scan and thus the Virtual 2D Scan, new landmarks appear. As of index 242 the MCL is converged to the true position again. Thus, the manual quality control decides that the reference path from index 1 to 197 and 242 to 924 can be used as ground truth in the benchmarking process. The remaining error of the reference path is estimated to be white noise with zero mean and a standard deviation of 10 cm.

5.3. Mapping Results

5.3.1. Mapping with Internal Sensors and GPS

Because any sensor is inaccurate, the maps generated using internal sensors for pose estimation are of limited quality, as has been demonstrated many times before. For odometry and the gyro-based localization, the errors for orientation and position are potentially unbounded. However, because paths usually contain left and right turns, these errors partially balance. The GPS shows problems close to buildings, where the

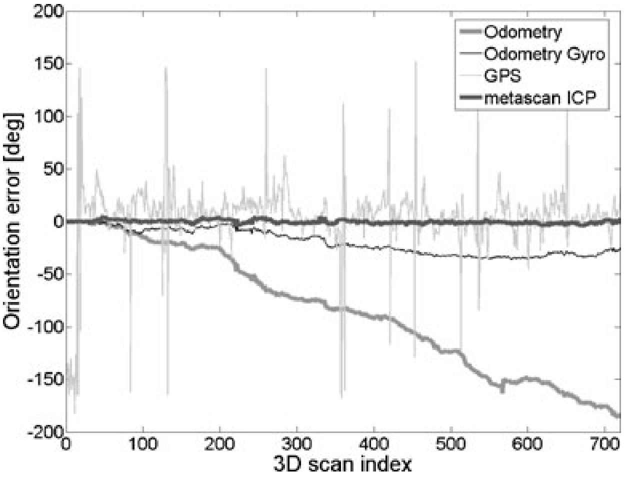

Figure 5. Orientation errors. Comparing internal sensors measurements, GPS headings, and metascan ICP matching with orientations computed by MCL localization. The x axis represents the 3D scan index, roughly corresponding to the position at the robot path. The orientation errors were computed using Eq. (5).

orientation is poorly estimated and the position error reaches its maximal value. Figure 5 shows the orientation errors of the internal sensors in comparison to ICP scan matching.

5.3.2. Mapping with ICP

Mapping with ICP was done using two different methods, namely pairwise ICP and metascan ICP. The latter method outperforms pairwise ICP because it considers all previously acquired 3D scans leading to slower error accumulation. Figure 6 shows the scan matching errors in comparison to methods that use explicit loop closure that are described next.

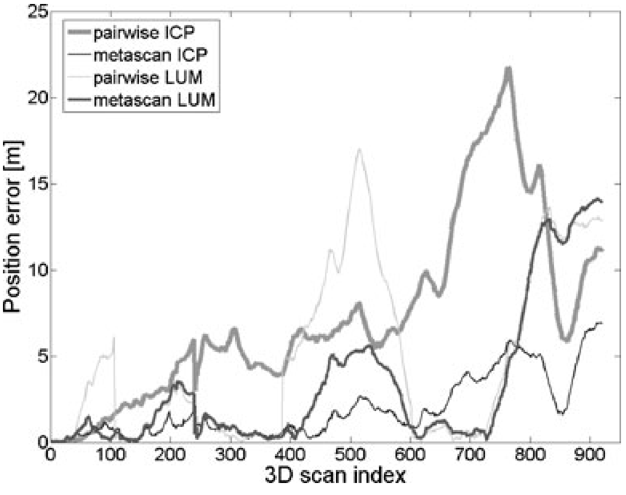

Figure 6. Position errors. Comparison of different mapping strategies. The position errors were computed using Eq. (4).

5.3.3. Mapping with ICP and Global Relaxation

The performances of the methods pairwise LUM and metascan LUM were also evaluated. As expected, loop closing reduces the position error at the positions where the loop is closed to approximately zero, for example, Figure 6 at scan index 100, where the first loop was closed, and at the indices 300–400 and 600–700. At these locations, the Lu/Milios-style SLAM methods outperform the pairwise ICP and metascan ICP methods and the absolute error is close to zero. However, because global relaxation produces consistent maps, the error between the loop closures might be larger than the one obtained without global relaxation. Having consistent maps does not necessarily imply correct maps. Pairwise LUM, and metascan LUM may also fail, if the loop cannot be closed. This case occurs in our experiment in the final part of the trajectory, that is, when the scan index is greater than 700 (cf. Figures 6 and 7). This last loop was not detected by the threshold method described in Section 3.4.

Finally, Tables 1 and 2 compare all localization/mapping methods. Figure 8 shows the final map generated with metascan LUM. The left-hand part contains the first 720 3D scans that were matched correctly, whereas the right-hand part contains all scans including the errors due to the

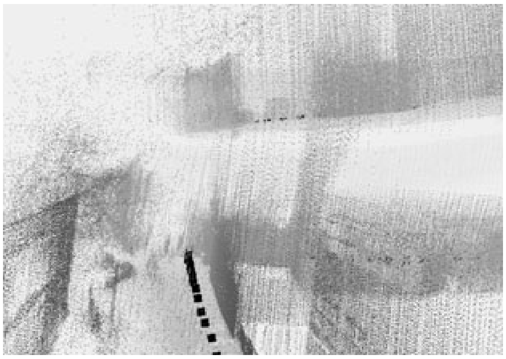

Figure 7. 3D view of the problematic loop closure in Figure 8 (right, black rectangle). Loop closing is not possible due to accumulated elevation errors.

Table I. Position errors (m).

Method	σ	$e_{\max}$
Odometry	55.1	261.2
OdometryGyro	64.7	250.1
GPS	5.8	95.1
Pairwise ICP	5.2	21.8
Metascan ICP	1.6	6.6
Pairwise LUM	4.9	17.0
Metascan LUM	3.8	13.8

Table II. Orientation errors (deg).

Method	σ_θ	$e_{\theta,\max}$
Odometry	77.2	256.6
OdometryGyro	15.1	56.7
GPS	27.3	171.0
Pairwise ICP	6.3	17.7
Metascan ICP	2.4	11.8
Pairwise LUM	5.2	22.8
Metascan LUM	4.3	21.2

undetected loop. Figure 9 shows a 3D view of the scene including two close-up views for trajectories given in Figure 10.

5.4. Computational Requirements

Of the compared mapping methods, only the internal sensor–based method and the pairwise ICP are online capable. Pairwise ICP using an octree-based point reduction and *k*d-tree search are performed in less than 1.2 s using standard computing hardware. In metascan ICP, mapping the computing time for closest point calculations increases with the number of scans; therefore, the scan matching time increases to 11.2 s for matching scan no. 920 with all previous ones, that is, matching 32,580 against 29 million points.

Pairwise LUM and metascan LUM spend additional time on computing the point correspondences for scans represented by the nodes in the graph. Owing to the iteration required by our GraphSLAM algorithm, the two methods are not online capable (Borrmann et al., in press). The total map processing times were 207 and 371 min, respectively. The largest portion of the computing time was spent in calculating closest points.

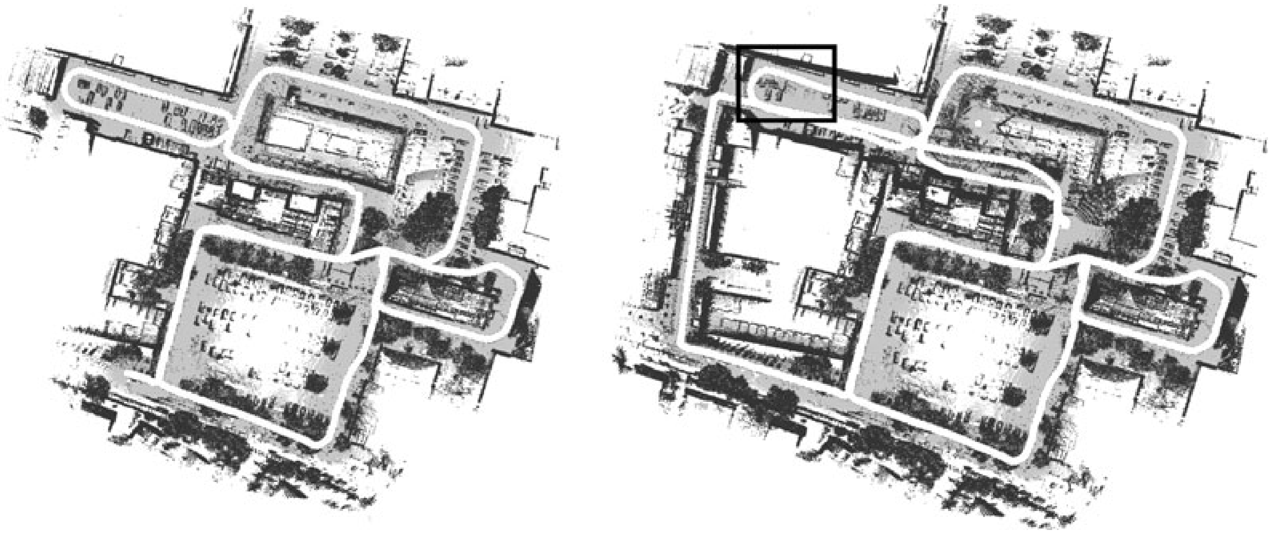

Figure 8. Final 3D map using the metascan LUM strategy. 3D points are classified as ground (light gray) and object points (dark gray). The trajectory is denoted in white. Left: Registration of the first 720 3D laser scans into a common coordinate system. Global relaxation leads to a consistent map. Right: The accumulated elevation errors on the remaining path (3D Scan 700 to end) prevent loop closing (black rectangle). Because of that, parts of the map are inconsistent. A detailed view of the black rectangle is provided in Figure 7.

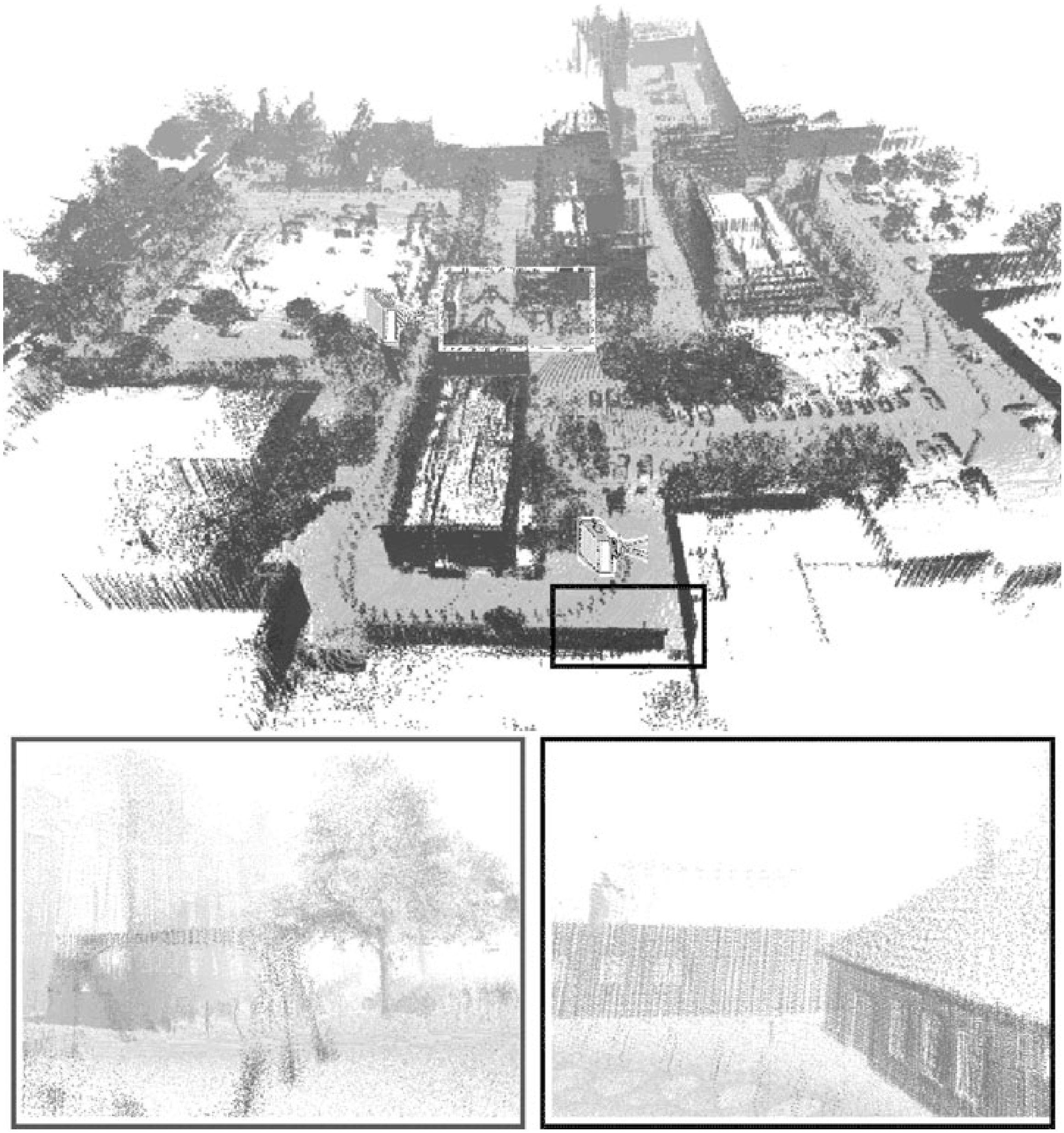

Figure 9. 3D view corresponding to Figure 8, left. Two close-up views are provided, and the corresponding camera positions are shown. The trajectories are given in Figure 10.

6. JUSTIFICATION OF THE RESULTS

To validate our experimental methodology, that is, to generate ground truth reference positions using MCL as described, we match the acquired 3D scans with a 3D map generated from the 2D reference map. For this, the 2D map is extrapolated as a 3D map, cuboids representing the boundaries of the buildings. Figure 11 shows the final map with the point clouds representing the buildings.

This 3D map, which is a 2D map extended by some height, is used for comparison using the following three strategies:

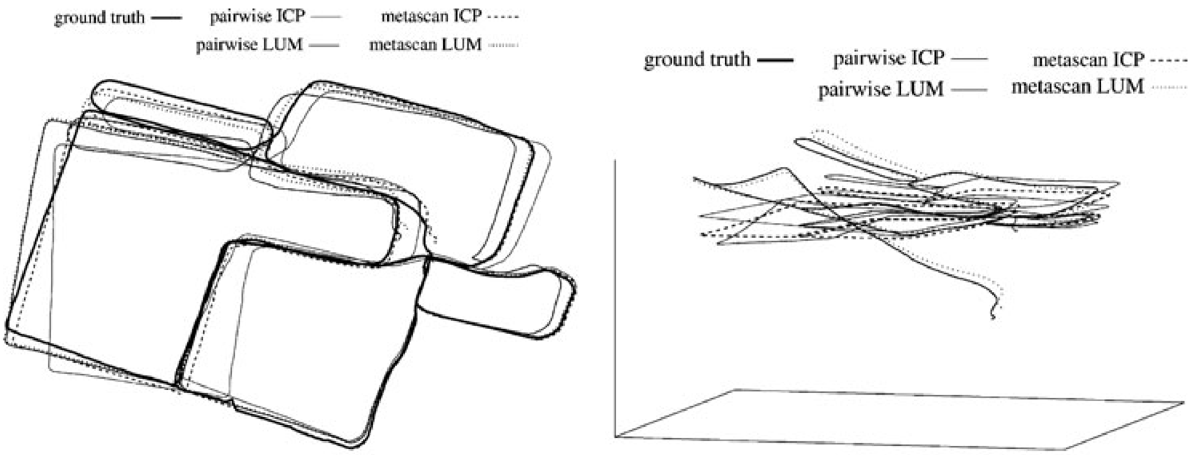

Figure 10. Computed trajectories in comparison. Right: Top view. Left: 3D view. Owing to the accumulated height error, the last loop closing is not possible and the computed error score of the SLAM algorithms with global relaxation is larger than with metascan ICP (cf. Table 1).

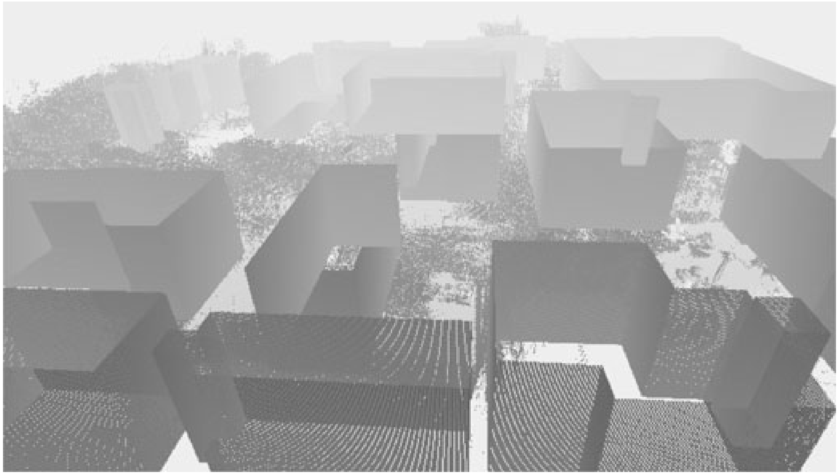

Figure 11. Mapped 3D scans overlaid with 3D cuboids based on the 2D reference map.

1. ICP is used for matching every single 3D scan with the point cloud based on the 2D map. As it turns out, this method can be applied only to the first 200 scans, because the map does not cover the whole robot path. In comparison, MCL successfully deals with this problem by applying the motion model to the particles until scan matching is possible again. This method is referred to as map ICP (first part).
2. ICP is used for matching a 3D scan with the metascan consisting of the 3D points from the map and all previous acquired and registered 3D scans. The method will be called metascan map ICP.
3. The previous method is used with the extension that points classified as ground are not included, that is, only the dark gray points are used for computing point correspondences and the transformation. Thus it

is called metascan map ICP w/o ground. It is expected that this restriction results in better ICP convergence.

Figures 12 and 13 compare the additional localization/mapping methods. Tables 3 and 4 give quantitative results. It turns out that these justification methods behave similarly to MCL and produce comparable results, that is, the MCL trajectory differs only by statistical noise from the trajectories produced by ICP scan matching using a 3D point cloud derived from the map.

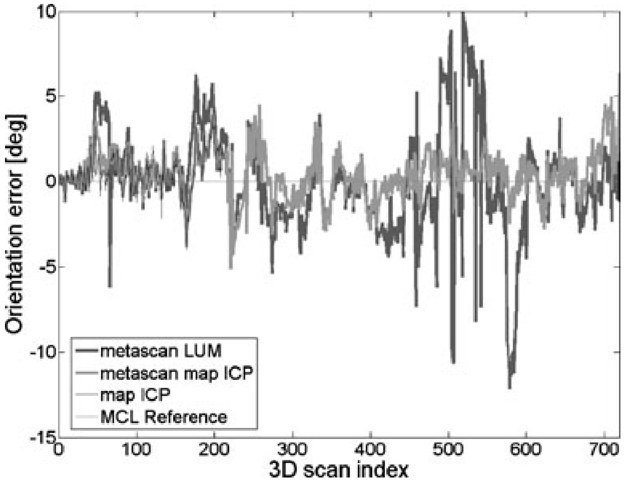

Figure 12. Orientation errors. Comparing metascan LUM, metascan map ICP, and map ICP (first part) with orientations computed by MCL localization. The x axis represents the 3D scan index, roughly corresponding to the position at the robot path.

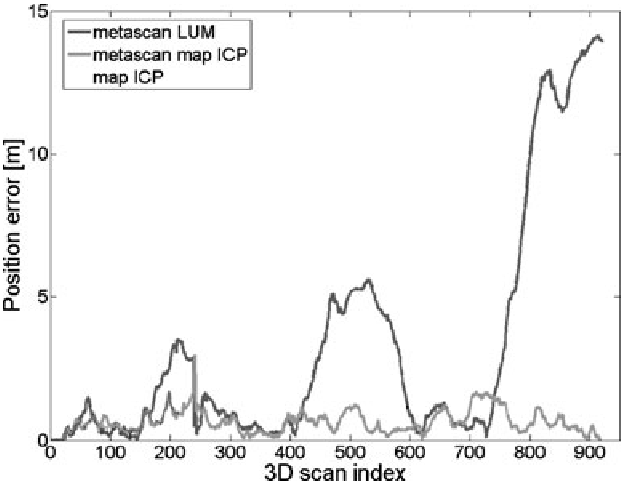

Figure 13. Position errors (Euclidean distance to MCL localization). Comparing metascan LUM, metascan map ICP, and map ICP (first part).

Table III. Position errors (m).

Method	σ	$e_{\max}$
Map ICP (first part)	0.3	1.77
Metascan map ICP	0.38	1.79
Metascan map ICP w/o ground	0.38	2.98

Table IV. Orientation errors (deg).

Method	σ_θ	$e_{\theta,\max}$
Map ICP (first part)	1.24	4.00
Metascan map ICP	1.40	6.37
Metascan map ICP w/o ground	1.38	6.32

7. CONCLUSION AND FUTURE WORK

Benchmarking of algorithms and research in experimental methodology are topics that are gaining increasing importance in robotics. Thus this paper presents a novel evaluation method for SLAM in urban outdoor environments. The evaluation is based on a comparison of the final SLAM results and ground truth reference positions. In our case, these reference positions are generated with a manually supervised MCL working on surveyed reference maps. Given these reference positions, it is possible to calculate objective benchmark scores that can be used for improving and comparing algorithms. This evaluation technique is demonstrated with experimental data and four different 6D SLAM strategies. The experiment that contains 924 full 3D scans on a 1.2-km path was carried out on the campus of the Leibniz Universität Hannover.

Needless to say, much work remains to be done, because some limitations of the system remain: Using MCL, it is possible only to estimate a three-DoF pose. Therefore, we plan to acquire a 3D scan by an airborne laser scanner as an additional source of information for estimating six-DoF reference poses. Furthermore, in areas lacking features, that is, wide open spaces with no buildings, we plan to complement our system with GPS. However, this switch might not be straightforward.

Future work will include two aspects: First, research in robotic benchmarking techniques needs emphasis. This conviction will be shared with the robotics community. To this end, we plan to cooperate with the Radish: The Robotics Data Set Repository (Howard & Roy, 2007) and the OpenSLAM (Stachniss et al., 2007) projects. The material presented in this paper, that is, the 3D scans, odometry, 6D SLAM, and MCL pose estimates can currently be accessed at http://kos.informatik.uni-osnabrueck.de/3Dscans/. Moreover, we will address loop closing as a crucial and challenging problem in SLAM (cf. Figure 7).

ACKNOWLEDGMENTS

The authors thank Dorit Borrmann, Jan Elseberg, and Kai Lingemann (University of Osnabrück) for joint development of the Lu/Milios-style GraphSLAM with six DoF. Furthermore, we acknowledge Raj Madvahan (National Institute of Standards and Technology) for putting the benchmarking issue into our research focus.

REFERENCES

Besl, P., & McKay, N. (1992). A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239–256.

Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., & Hertzberg, J. (in press). Globally consistent 3D mapping with scan matching. Journal of Robotics and Autonomous Sytems.

Cole, D. M., & Newman, P. M. (2006). Using laser range data for 3D SLAM in outdoor environments (pp. 1550–1563). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06), Orlando, FL.

DARPA (2007). www.darpa.mil/grandchallenge/.

Davis, T. A. (2006). Direct methods for sparse linear systems. Philadelphia: SIAM.

Davison, A. J., Reid, I., Molton, N., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), 29(6), 1052–1067.

Federation, T. R. (2007). http://www.robocup.org/.

FGAN (2007). http://www.elrob2006.org/.

Fox, D., Thrun, S., Burgard, W., & Dellaert, F. (2001). Particle filters for mobile robot localization (pp. 419–516). In A. Doucet, N. de Freitas, & N. Gordon (Eds.), Sequential Monte Carlo methods in practice. Springer.

Frese, U. (2007). Efficient 6-DOF SLAM with treemap as a generic backend (pp. 4814–4819). In Proceedings of the IEEE Internation Conference on Robotics and Automation (ICRA '07), Rome, Italy.

Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P., & Burgard, W. (2007). Efficient estimation of accurate maximum likelihood maps in 3d (pp. 3472–3478). In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '07), San Diego, CA.

Guivant, J., Nebot, E., & Baiker, S. (2000). Autonomous navigation and map building using laser range sensors in outdoor applications. Journal of Robotic Systems, 17(10), 565–583.

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

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A, 4(4), 629–642.

Howard, A., & Roy, N. (2007). http://radish.sourceforge.net/.

Kaess, M., Ranganathan, A., & Dellaert, F. (2007). iSAM: Fast incremental smoothing and mapping with efficient data association (pp. 2486–2493). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07), Rome, Italy.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. Autonomous Robots, 4(4), 333–349.

Newman, P. M., Cole, D. M., & Ho, K. (2006). Outdoor SLAM using visual appearance and laser ranging (pp. 1180–1187). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06), Orlando, FL.

Nüchter, A., Surmann, H., Lingemann, K., Hertzberg, J., & Thrun, S. (2004). 6D SLAM with an application in autonomous mine mapping (pp. 1998–2003). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04), New Orleans, LA.

Olson, E., Leonard, J. J., & Teller, S. (2006). Fast iterative optimization of pose graphs with poor initial estimates (pp. 2262–2269). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06), Orlando, FL.

Pfaff, P., Triebel, R., Stachniss, C., Lamon, P., Burgard, W., & Siegwart, R. (2007). Towards mapping of cities (pp. 4807–4813). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07), Rome, Italy.

Stachniss, C., Frese, U., & Grisetti, G. (2007). http://www.openslam.org/.

Surmann, H., Nüchter, A., Lingemann, K., & Hertzberg, J. (2004). 6D SLAM: A preliminary report on closing the loop in six dimensions. In Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV '04), Lisbon, Portugal.

Thrun, S. (2002). Robotic mapping: A survey (pp. 1–35). In G. Lakemeyer & B. Nebel (Eds.), Exploring artificial intelligence in the new millennium. San Francisco: Morgan Kaufmann.

Thrun, S., Fox, D., & Burgard, W. (2000). A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping (pp. 321–326). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00), San Francisco, CA.

Thrun, S., Montemerlo, M., & Aron, A. (2006). Probabilistic terrain analysis for high-speed desert driving (pp. 161–168). In Proceedings of robotics: Science and systems. Cambridge, MA.

Torralba, A., Murphy, K. P., & Freeman, W. T. (2007). http://web.mit.edu/torralba/www/database.html/.

Triebel, R., Pfaff, P., & Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing (pp. 4176–4182). In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06), Beijing, China.

Wulf, O., Arras, K. O., Christensen, H. I., & Wagner, B. (2004). 2D mapping of cluttered indoor environments by means of 3D perception (pp. 4204–4209). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04), New Orleans, LA.

Wulf, O., Brenneke, C., & Wagner, B. (2004). Colored 2D maps for robot navigation with 3D sensor data (pp. 2991–2996). In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '04), Sendai, Japan.

Wulf, O., & Wagner, B. (2003). Fast 3D-scanning methods for laser measurement systems (pp. 312–317). In Proceedings of the International Conference on Control Systems and Computer Science, Bucharest, Romania.