Ground Truth Evaluation of Large Urban 6D SLAM

Oliver Wulf, Andreas Nüchter, Joachim Hertzberg, and Bernardo Wagner

***Abstract*— In the past many solutions for simultaneous localization and mapping (SLAM) have been presented. Recently these solutions have been extended to map large environments with six degrees of freedom (DoF) poses. To demonstrate the capabilities of these SLAM algorithms it is common practice to present the generated maps and successful loop closing. Unfortunately there is often no objective performance metric that allows to compare different approaches. This fact is attributed to the lack of ground truth data. For this reason we present a novel method that is able to generate this ground truth data based on reference maps. Further on, the resulting reference path is used to measure the absolute performance of different *6D SLAM* algorithms building a large urban outdoor map.**

I. Introduction

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 with real autonomy. SLAM algorithms integrate robot action and sensor readings and exploit the fact that previously mapped areas are recognized. Global optimization methods yield consistent maps. Nevertheless, these consistent maps might be incorrect and therefore ground truth experiments have to be made. This paper presents ground truth experiments using a novel empiricism.

Popular mapping algorithms work with 3DoF pose estimates, i.e., robot poses are represented by three degrees of freedom $P = (x, y, \theta_z)$. For indoor environments this choice is appropriate, but a current trend for mapping outdoor environments are mapping algorithms that represent poses in 6DoF, i.e., *6D SLAM* [16]. These algorithms consider the 6DoF pose $V = (x, y, z, \theta_x, \theta_y, \theta_z)$ of the mobile robot with 3 position coordinates and roll, pitch and yaw angles. Robot motion and localization on natural surfaces must regard these 6 degrees of freedom. Recently, 3D mapping of large environments received much attention, [3], [17], [23]. A framework for benchmarking these large experiments is still missing.

This paper evaluates algorithms and methods for autonomous mapping. A mobile robot, equipped with a fast 3D scanner gages the environment, while it is steered through a large urban environment. The maps generated by online and offline algorithms are compared to odometry based, gyro based and GPS based pose estimates. Ground truth is provided by a Monte Carlo Localization (MCL) using accurate reference maps.

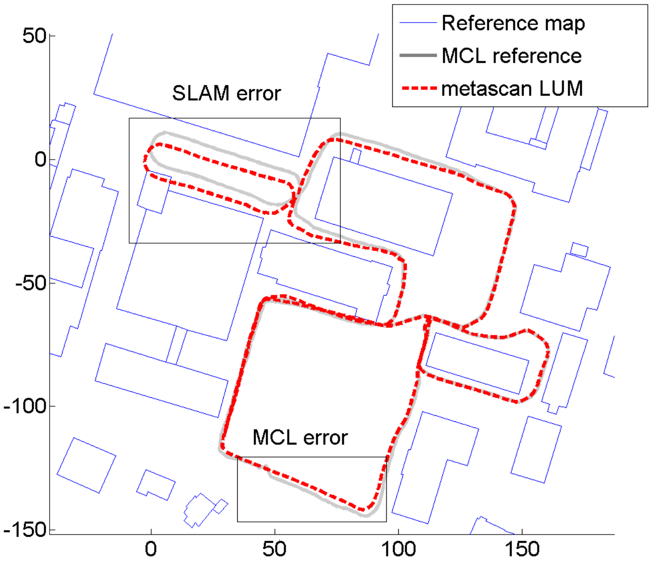

Fig. 1. Performance measurement of SLAM algorithms. Using reference maps and MCL in urban outdoor environments (3D scan index 1 - 700). Distances are given in meter.

A. Ground Truth Experiments

In doing experiments with ground truth reference, researchers aim to measure the objective performance of a dedicated algorithm. Based on this benchmark it is possible to give an experimental prove of the effectiveness of a new algorithm. Furthermore measuring the performance of algorithms allows to optimize the algorithm and to compare it to other existing solutions.

Benchmarking is a common scientific instrument. A good example for successful performance measurement in computer science is the computer vision community. There are several projects that aim at providing image data bases to other researchers [11] [22]. 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 like image segmentation and object recognition.

Unfortunately this kind of performance measurement is not widely spread 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 only done by small projects [13] [18] or

Oliver Wulf and Bernardo Wagner are with the Real Time Systems Group (RTS) of the Institute for Systems Engineering, Leibniz Universität Hannover, Germany. Contact: wulf@rts.uni-hannover.de

Andreas Nüchter and Joachim Hertzberg are with the Knowledge-Based Systems Research Group of the Institute of Computer Science, University of Osnabrück, Germany. Contact: nuechter@informatik.uni-osnabrueck.de

individual researchers. Another way of comparing robotic systems are competitions like RoboCup [6], ELROB [7] or the Grand Challenge [4]. With this kind of competitions it is possible to measure the level of system integration and the engineering skills of a certain team, but it is not possible to measure the performance of a subsystem or a single algorithm.

Objective benchmarking of localization and mapping algorithms is only achieved by comparing of experimental results against reference data. The practical problem is the generation of this ground truth data. In computer vision, ground truth data is either available for synthetic images, or needs to be hand labeled. In case of mobile robot navigation one way of gathering ground truth data is the use of precise global positioning systems (RTK-GPS) [10]. Unfortunately, this data is only available in open outdoor environments and not for urban outdoor environments or indoor environments. Another possibility is to use complex external measurement setups.

Another benchmarking method for robotic algorithms comprises simulation. Realistic simulation enables reseachers to perform experiments with defined conditions and to repeat these experiments. However, real life differs from simulation. Experiments, involving sophisticated sensors such as cameras or laser scanners can only be simulated up to a certain level of accuracy, e.g., capturing environments must regard surface properties such as material, local structures and reflexions. Therefore, using real robotic data sets is favored for benchmarking.

With this paper, we present a novel method of gathering ground truth data in indoor and urban outdoor environments. The procedure is making use of a highly accurate environment map (provided by the land registry office), a Monte Carlo Localization that matches sensor data against the reference map and a manual supervision step.

B. State of the Art in Metric Robotic Mapping

1) Planar Mapping: State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion models and uncertain perception models. Through integration of these two distributions with a Bayes filter, e.g., 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, i.e., 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 [20] surveys existing techniques, i.e., maximum likelihood estimation, expectation maximization, extended Kalman filter or (sparsely extended) information filter SLAM. FastSLAM [21] approximates the posterior probabilities, i.e., robot poses, by particles.

SLAM in well-defined, planar indoor environments is considered solved. In principle probabilistic methods are extendable to 6DoF. However, to our knowledge no reliable feature extraction mechanisms nor methods for reducing the computational cost of multihypothesis tracking procedures like FastSLAM (which grows exponentially with the degrees of freedom) have been published.

2) Mapping Environments in 3D.: An emerging research topic is *6D SLAM*, i.e., while mapping the robot pose is represented with six degree of freedom. 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 scan into one coordinate system [16], [19]. Similar experiments have been made by Newman et al. [15]. A current trend in laser based *6D SLAM* is to overcome stop- and go fashion of scan acquisition by rotating or pitching the scanner while moving [3], [23], [24]. In the most recent work Pfaff et al. [17] employ two rotating SICK scanners for data acquisition, odometry, IMU and DGPS positioning, a variant of the iterative closest point (ICP) algorithm and a loop closing procedure to map large urban environments in 3D.

Feature-based *6D SLAM* methods are investigated by Udo Frese, who adapted his fast treemap algorithm to six degrees of freedom [9]. Among the category of feature based *6D SLAM* are the visual SLAM methods, i.e., the MonoSLAM system of Davison et al. [5].

The remainder of the paper is structured as follows: Next, we describe the sensor system for generating large 3D maps and the two pairs of evaluated mapping algorithms. In section III we present the MCL based benchmarking technique. Then we present results from an experiment consisting of 924 3D scans. Section V concludes.

II. Generation of Large Urban 3D Maps

A. Sensor System

The sensor that has been employed for the experiments is a fast 3D laser range scanner, developed at the Leibniz Universität Hannover (see 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° 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. Having 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 [26].

Having these optimizations described above the limiting factor in bulding faster 3D laser scanner is the maximal

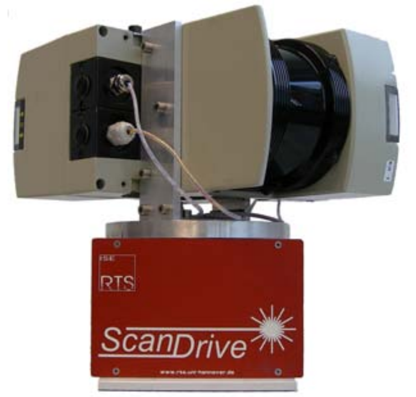

Fig. 2. 3D laser range sensor RTS/ScanDriveDuo. Measuring full 3D scans with 32580 points in 1.2 sec.

number of 13575 (75 × 181) points that can be measured with a SICK LMS 2xx sensor in one second. The only way of building faster SICK LMS 2xx based 3D scanners is the use of multiple 2D measurement devices [23]. For this reason we first present the RTS/ScanDriveDuo with this paper. This 3D scanner makes use of two SICK LMS 291 2D laser scanners. Thus the measurement time for 3D scans with 2° horizontal and 1° vertical angle resolution is reduced to 1.2 sec. In this case one 3D scan measured in 1.2 sec consists of 32580 (180 × 181) 3D points.

In addition to the 3D laser scanner the mobile robot is equipped with wheel odometry, a 3 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 that has got no influence on neither the MCL nor the SLAM results. It is only logged to have another laser independent reference.

B. 6D SLAM with ICP based Scan Matching

We use the well-known Iterative Closest Points (ICP) algorithm [1] 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

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

where N_m and N_d, are the number 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 in the last iteration the point correspondences are correct. In each iteration, the transformation is calculated by the quaternion based method of Horn [12].

To digitalize environments without occlusions, multiple 3D scans have to be registered. Consider a robot travelling 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*, i.e., 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 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 $V_0, ..., V_{i-1}$ and the data set D that of pose V_i, for all $i \in [1, n]$.

C. 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 and V_j are close enough, i.e., their distance falls below a threshold (here: 5 meter) then we assume 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 *meta scan matching*, we detect closed loops using this simple distance criterion. Once detected, a 6DoF graph optimization algorithm for global relaxation based on the method of Lu and Milios [14] is employed, namely Lu and Milios style SLAM (LUM). This is a variant of GraphSLAM. Details of the 6DoF optimization, i.e., how the matrices have to be filled, can be found in [2], thus we give only a brief overview here:

Given a network with $n+1$ nodes $X_0, ..., X_n$ representing the poses $V_0, ..., V_n$, and the directed edges $D_{i,j}$, we aim at estimating all poses optimally to build a consistent map of the environment. For simplicity, the approximation that the measurement equation is linear is made, i.e.,

$$D_{i,j} = X_i - X_j \qquad (2)$$

An error function is formed such that minimization results in improved pose estimations:

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

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 to resolve the non-linearities 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. The error function eq. (3) has a quadratic form and is therefore solved in closed form by Cholesky decomposition in the order of $\mathcal{O}(n^3)$ for n poses. The algorithm optimizes eq. (3) gradually by iterating the follwoing three steps: First, for every network link the corresponding covariance is computed based on the point correspondencies 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.

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.

D. Mapping Stategies

Animations of the four mapping stategies, *pairwise ICP, metascan ICP, pairwise LUM, metascan LUM* are given in the accompanying video and on the following web page: `http://kos.informatik.uni-osnabrueck.de/download/6DSLAMbenchmarking`. Note the maps presented in the video are rotated about 190°.

III. 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 can not be used in urban outdoor environments, we present a technique that is based on surveyed maps as they can be obtained from the German land registry offices. The process of generating this ground truth reference positions can be divided into a Monte Carlo Localization step that matches the sensor data to the highly accurate map and a manual supervision 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. But 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. For this reason the remaining error of the manually supervised reference position is at least an order of magnitude smaller then the discussed SLAM errors.

A. Reference Map

As part of their geo information system (GIS) the German land registration offices features 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 there are no further details about doors, windows or balconies available.

B. Monte Carlo Localization

The Monte Carlo Localization (MCL) is a commonly used localization algorithm that is based on particle filtering [8]. 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* [24]. 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. A sequence of 3D segmentation and classification algorithms that is used to do this reduction in urban outdoor environments is described in [25]. 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.

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 got 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 down sampled. Experimental results show that less than 100 measurement points are needed for sufficient localization.

Due to the 2D nature of the reference map and the used 2D MCL algorithm it is only possible to estimate the 3DoF pose $P^{REF} = (x, y, \theta_z)$ of the robot. There is no reference information on the robots height z. And the roll and pitch components θ_x θ_y of the 6DoF robot pose can not 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*.

C. Manual Supervision

Unlike MCL algorithms used in fully autonomous navigation the generation of reference positions needs manual supervision. Even though the human supervisor is not able to identify the absolute accuracy of the estimated MCL position, it is possible to check the conditions that are needed for proper operation. If all of these conditions are fulfilled the MCL algorithm is able to find the true position of the robot in global coordinates.

There are several conditions that need to be checked to attest proper operation:

At first the sensor data needs to be checked for a sufficient number of landmarks. Namely, walls as they are given in the reference map. In case of an open area without landmarks in the surrounding of the robot, occluded landmarks or insufficient *Virtual 2D Scans* the MCL results only depend on odometry and are therefore not accurate.

The second step is to supervise the numerical condition of the particle filter. As a particle filter only presents a sampled belief an efficient distribution of the finite number of particles is essential for correct operation. For this reason the human supervisor needs to make sure that enough particles are located around the true position. The estimated position can be corrupt if particles are located around more than one maximum or around wrong local maxima.

Finally, the human supervisor can valuate the overall soundness of the localization and mapping results. For this reason it is necessary to display the reference map with overlaid sensor data. As the sensor data is transformed with the MCL results, fatal matching errors can be detected by the supervisor.

D. 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 robots start position. To be able to compare the positioning results it is necessary to transform the SLAM positions 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 based on the 2D Euclidean distance between the SLAM and MCL position

$$e_i = \sqrt{(x_i^{SLAM} - x_i^{REF})^2 + (y_i^{SLAM} - y_i^{REF})^2}. \qquad (4)$$

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

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

As the MCL position has got only 3DoF, the robots elevation, roll and pitch angle can not 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

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

and the error maximum

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

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

IV. Experimental Results

A. Experimental Setup

The presented experiment has been 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 5 small and large loops. On this path 924 full 3D scans have been 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 have been stored with a data rate of 10 Hz. And the position fixes of a low-cost GPS have been logged with 1 Hz.

B. Ground Truth Data

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 5806400 m northing and 3548500 m easting is subtracted from all Gauss-Krueger coordinates. This offset corresponds to the position 52°23′58″ north, 9°42′41″ east in WGS84 coordinates.

The MCL reference positions are calculated online on the Pentium III 700 MHz processor included in the 3D sensor. The particle filter runs with 200 samples and a generous estimate of the sensor variance of 30cm. This estimate includes the sensor range error, errors from scanning while moving and map uncertainties. The localization results are plotted as a solid gray line in Fig. 1.

The result of the offline manual observation is that the MCL positions can be used as reference positions for 3D scan indexes 1 to 197 and 242 to 924. On the other hand positions corresponding to 3D scan indexes 198 to 241 can not be used as there are not enough landmarks visible to the 3D sensor (MCL error box in Fig. 1). Due to that particles diverge and the calculated position follows the drifting odometry. Starting with 3D scan 138 the *Virtual 2D Scan* contains new landmarks and thus the MCL converges quickly to the true position.

For that reason results from 3D scan indexes 198 to 241 are not considered in the following analysis.

C. Mapping Results

1) Mapping with Internal Sensors and GPS: Since all sensors are 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 error for orientation and postion are potentially unbounded. However, since paths usually contain left and right turns, these errors partially balance. The GPS shows problems close to buildings, where the orientation is poorly estimated and the position error reaches its maximal value. Fig. 3 shows the orientation errors of the internal sensors in comparison to ICP scan matching.

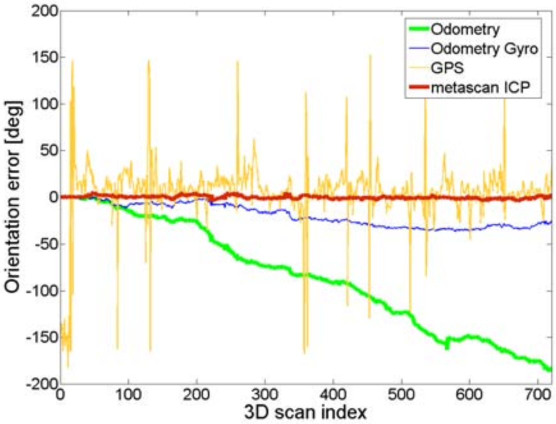

Fig. 3. Orientation errors. Comparing internal sensors measurments, GPS headings and *metascan ICP* matching with orientations computed by MCL localization. The x-axis represents the 3D scan index, roughly corresponding to the position at the robot path.

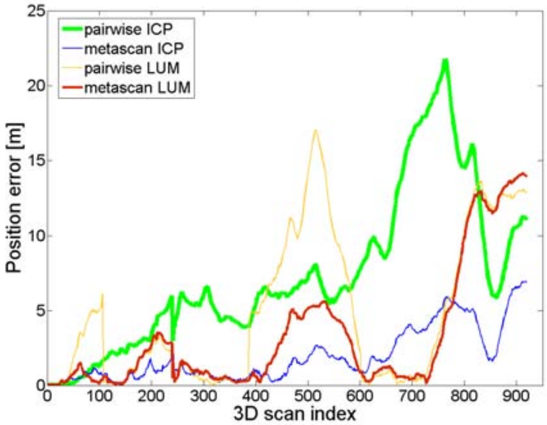

Fig. 4. Position errors (Euclidean distance to MCL localization). Comparison of different mapping strategies.

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* since it considers all previously acquired 3D scans leading to slower error accumulation. Fig. 4 shows the scan matching errors in comparison to methods using explicit loop closure that are described next.

3) Mapping with ICP and Global Relaxation: The performance of the methods *pairwise LUM, metascan LUM* have also been evaluated. As expected, loop closing reduces the position error at the positions, where the loop is closed to approximately zero, e.g., Fig. 4 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. However, *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, i.e., when the scan index is greater than 700 (cf. Fig. 4 and Fig. 6). This last loop was not detected by the threshold method described in section II-C.

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

Finally, tables I and II compare all localization/mapping methods. Fig. 5 shows the final map generated with *metascan LUM*. The left part contains the first 720 3D scans that have been matched correctly, whereas the right part contains all scans including the errors, due to the undetected loop. Fig. 7 shows a 3D view of the scene including two close-up views.

D. Computational Requirements

Of the compared mapping methods only the internal sensor based and the *pairwise ICP* are online capable. *Pairwise ICP* using an octree based point reduction and kd-tree search are perfomed in less than 1.2 sec. 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 sec. for matching scan No. 920 with all previous ones, i.e., matching 32580 against 29 Mio. points.

Pairwise LUM and *metascan LUM* spend additional time on computing the point correspondences for scans represented by the nodes in the graph. Due to the iteration required by our GraphSLAM algorithm, both methods are not online capable [2]. The total map processing time was 207 min and 371 min, respectively. The largest portion of the computing time was spent by calculating closest points.

V. CONCLUSION AND FUTURE WORK

Benchmarking of algorithms and research in experimental methodology are topics that get more and more important 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 Monte Carlo Localization working on surveyed reference maps. Having this reference positions it is possible to calculate objective benchmark scores that can be used to improve and compare 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

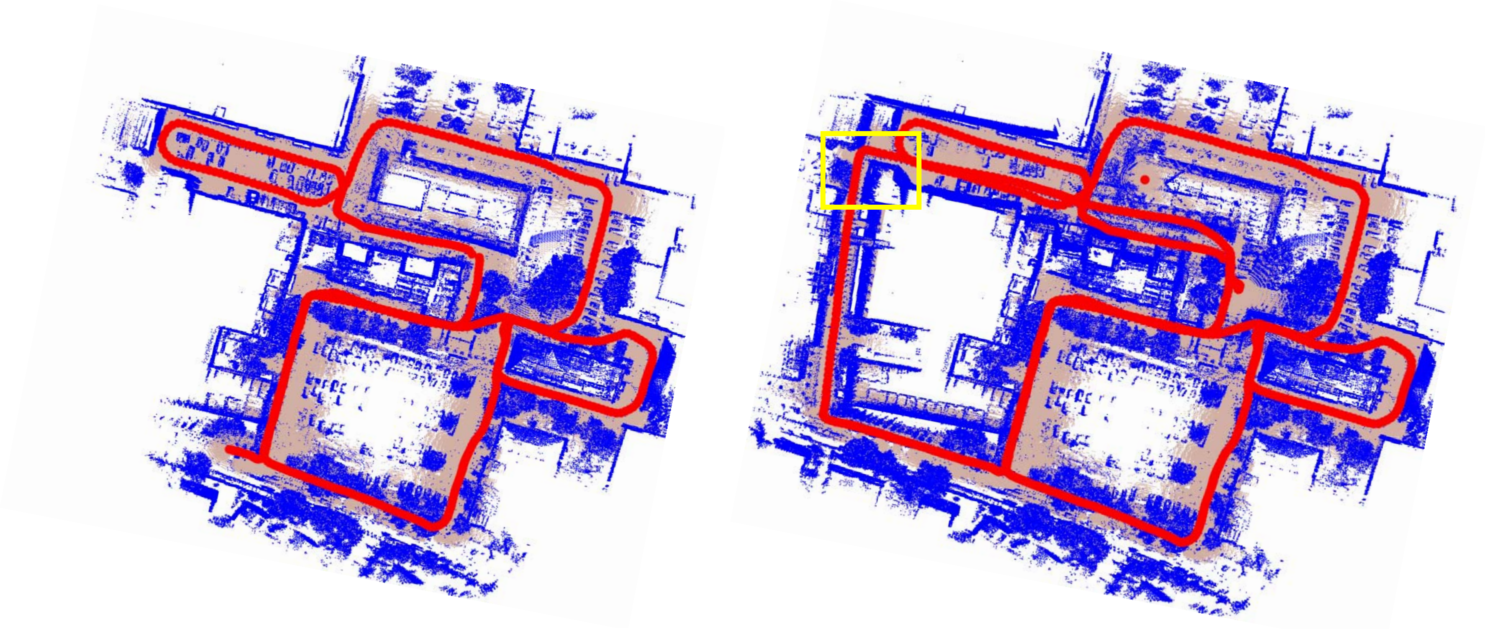

Fig. 5. Final 3D map using the metascan LUM strategy. 3D points are classified as ground (gray) and object points (blue). The trajectory is denoted in red. Left: Registration of the first 720 3D laser scans into a common coordinate system. Global relaxation leads to a consistent map. Right: The accumulated elevation errors on the remaining path (3D Scan 700 to end) prevents loop closing (yellow rectanlge). Due to that parts of the map are inconsistent. A detailed view of the yellow rectangle is provided in Fig 6.

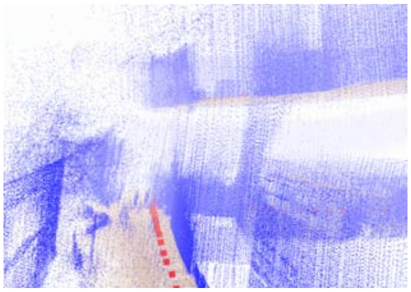

Fig. 6. 3D view of the problematic loop closure in Fig. 5 (right, yellow rectangle). Loop closing is not possible due to accumulated elevation errors.

path was carried out on the campus of the Leibniz Universität Hannover.

Needless to say that much work remains to be done. Future work will be done on two aspects: First, research in robotic benchmarking techniques needs to be emphasized. And second this ideas need to be spread out in the robotics community. To this end, we plan to cooperate with the *Radish: The Robotics Data Set Repository* [13] and the OpenSLAM [18] project.

VI. Acknowledgments

The authors would like to thank Dorit Borrmann, Jan Elseberg and Kai Lingemann (University of Osnabrück) for joint development of the Lu/Milios style GraphSLAM with 6 degrees of freedom.

References

[1] P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 – 256, February 1992.

[2] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally Consistent 3D Mapping with Scan Matching. *Journal of Robotics and Autonomous Sytems*, 2007, (accepted).

[3] D. M. Cole and P. M. Newman. Using Laser Range Data for 3D SLAM in Outdoor Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, Orlando, Florida, U.S.A., May 2006.

[4] DARPA. `www.darpa.mil/grandchallenge/`.

[5] A. J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6), June 2007.

[6] The RoboCup Federation. `http://www.robocup.org/`.

[7] FGAN. `http://www.elrob2006.org/`.

[8] D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization, 2001.

[9] U. Frese. Efficient 6-DOF SLAM with Treemap as a Generic Backend. In *Proceedings of the IEEE Internation Conference on Robotics and Automation (ICRA '07)*, Rome, Italy, April 2007.

[10] J. Guivant, E. Nebot, and S. Baiker. Autonomous navigation and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 2000.

[11] 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.

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

[13] A. Howard and N. Roy. `http://radish.sourceforge.net/`.

[14] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333 – 349, October 1997.

[15] P. M. Newman, D. M. Cole, and K. Ho. Outdoor SLAM using Visual Appearance and Laser Ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, Florida, U.S.A., April 2006.

[16] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an Application in Autonomous Mine Mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1998 – 2003, New Orleans, USA, April 2004.

[17] P. Pfaff, R. Triebel, C. Stachniss, P. Lamon, W. Burgard, and R. Siegwart. Towards Mapping of Cities. In *Proceedings of the IEEE Internation Conference on Robotics and Automation (ICRA '07)*, Rome, Italy, April 2007.

[18] C. Stachniss, U. Frese, and G. Grisetti. `http://www.openslam.org/`.

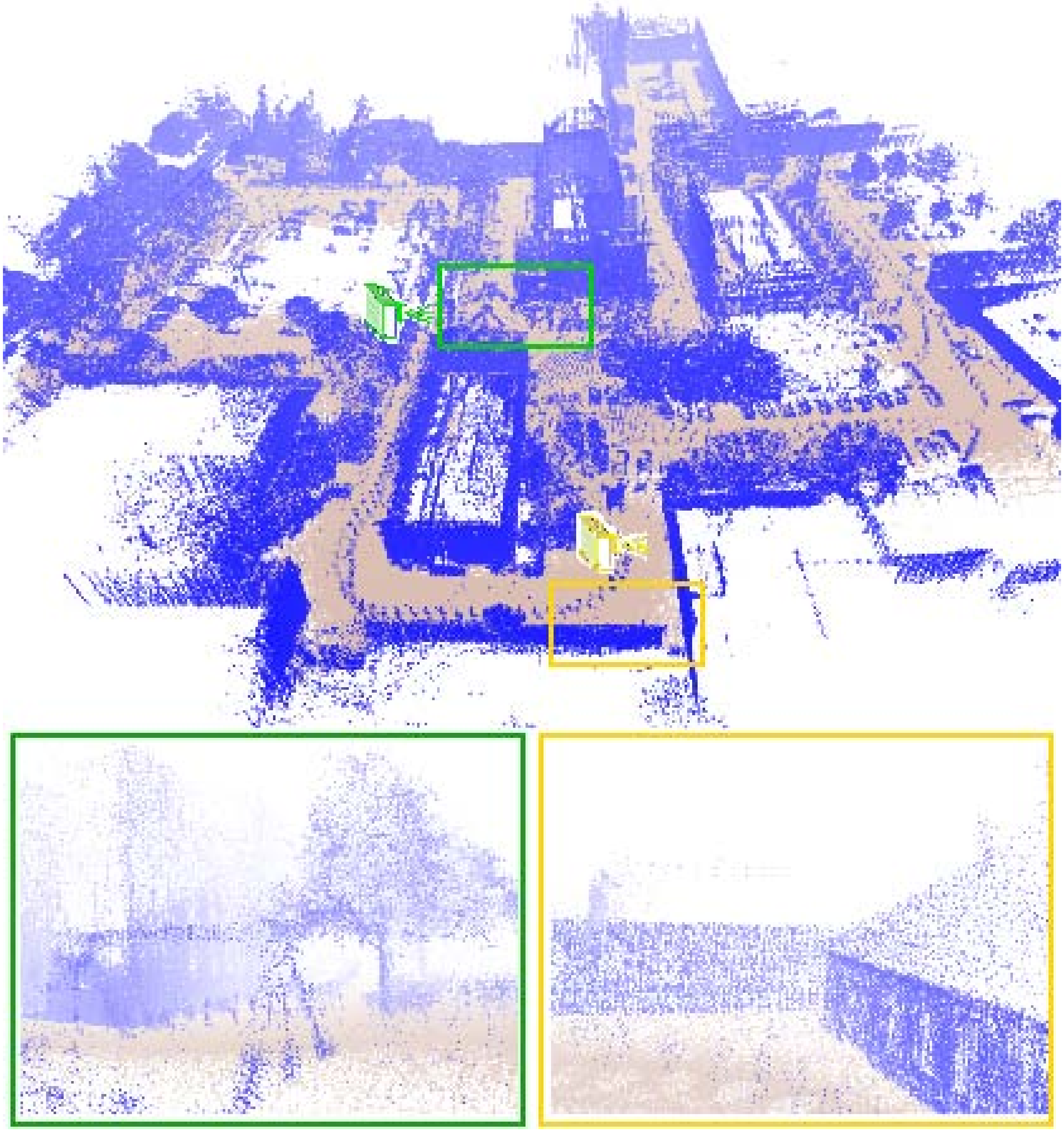

Fig. 7. 3D view corresponding to Fig. 5, left. Two close-up views are provided and the corresponding camera positions are shown.

[19] H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 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)*, Lisabon, Portugal, July 2004.

[20] S. Thrun. Robotic Mapping: A Survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

[21] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, CA, USA, April 2000.

[22] A. Torralba, K. P. Murphy, and W. T. Freeman. `http://web.mit.edu/torralba/www/database.html/`.

[23] R. Triebel, P. Pfaff, and W. Burgard. Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, Beijing, China, October 2006.

[24] O. Wulf, K. O. Arras, H. I. Christensen, and B. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, New Orleans, USA, April 2004.

[25] O. Wulf, C. Brenneke, and B. Wagner. Colored 2D Maps for Robot Navigation with 3D Sensor Data. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS '04)*, Sendai, Japan, September 2004.

[26] O. Wulf and B. Wagner. Fast 3D-Scanning Methods for Laser Measurement Systems. In *Proceedings of the International Conference on Control Systems and Computer Science*, Bucharest, Romania, July 2003.