2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Towards Mapping of Cities

Patrick Pfaff[*] Rudolph Triebel[*†] Cyrill Stachniss[†*] Pierre Lamon[†] Wolfram Burgard[*] Roland Siegwart[†]

[*]*University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany*
[†]*Eidgenössische Technische Hochschule (ETH), IRIS, ASL, 8092 Zurich, Switzerland*

Abstract— **Map learning is a fundamental task in mobile robotics because maps are required for a series of high level applications. In this paper, we address the problem of building maps of large-scale areas like villages or small cities. We present our modified car-like robot which we use to acquire the data about the environment. We introduce our localization system which is based on an information filter and is able to merge the information obtained by different sensors. We furthermore describe out mapping technique that is able to compactly model three-dimensional scenes and allows us efficient and accurate incremental map learning. We additionally apply a global optimization techniques in order to accurately close loops in the environment. Our approach has been implemented and deeply tested on a real car equipped with a series of sensors. Experiments described in this paper illustrate the accuracy and efficiency of the presented techniques.**

I. I

Building models of the environment is a fundamental task of mobile robots since maps are needed for most high-level robotic applications. In the past, many researchers focused on the problem of learning maps and different techniques have been proposed [5], [10], [11], [18], [20]. Most of the proposed approaches focus on learning models for indoor environments like office spaces. Recently, several groups addressed the problem of learning two and three-dimensional models of outdoor scenes [6], [12], [13], [14], [22].

Since DARPA Grand Challenge [2], the usage of cars instead of classical mobile robots became popular in the research community [1], [23], [25]. Compared to standard robots, cars offer the possibility to travel longer distances, carry more sensors, and thus being more suitable for mapping large areas.

The contribution of this paper is an approach towards mapping of large-scale areas like villages or small cities. We describe our system to learn three-dimensional models of the environment. We apply probabilistic state estimation techniques as well as classification approaches to obtain these models. Our implementation uses a modified Smart car depicted in Figure 1 equipped with a series of sensors, ranging from proximity sensor, GPS, and an inertial measurement unit (IMU).

II. R W

The problem of learning models of the environment has been studied intensively in the past. In the literature, this problem is often referred to as simultaneous localization and mapping (SLAM). Most approaches to map learning generate two-dimensional models from range sensor data. A series of different approaches has been developed to address this problem [4], [5], [7], [9], [10], [11], [18]. Recently, several techniques for acquiring three-dimensional data with rotating 2d range scanners installed on a mobile robot have been developed [8], [26], [27]. Other authors have studied the acquisition of three-dimensional maps from vehicles that are assumed to operate on a flat surface. For example, Thrun *et al.* [21] present an approach that employs two 2d range scanners for constructing volumetric maps. Whereas the first is oriented horizontally and is used for localization, the second points towards the ceiling and is applied for acquiring 3d point clouds.

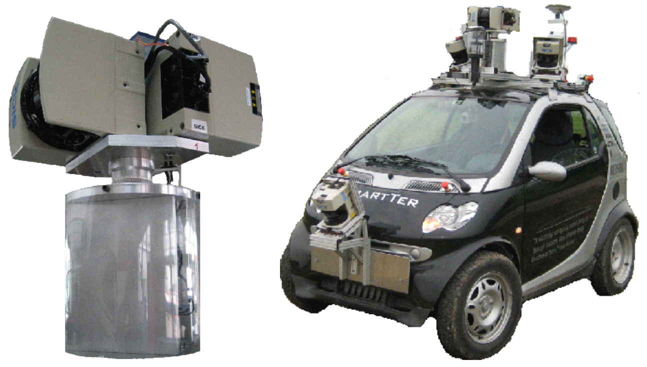
Fig. 1. The left image depicts the vertically mounted SICK LMS laser range finders which are rotated with constant speed by an electric step motor mounted under the lasers. The right image shows our robot. The robot is a standard Smart car. The model is a Smart fortwo coupé passion of the year 2005, which is equipped with a 45 kW engine.

A popular representation for $2\frac{1}{2}$-dimensional maps in robotics are elevation maps [17], [28]. In contrast to that, our approach learns a three dimensional model that can be regarded as an extention to elevation map which is able to store multiple layers for each grid cell [24]. This allows us to model structures like, e.g., bridges, underpasses, and trees in a more accurate way. We present our technique to match individual surface maps into a globally consistent model of the environment using a global error minimization approach. All techniques have been implemented and tested on a real car.

In the context of autonomous cars, a series of successful systems [1], [23], [25] have been developed due to DARPA Grand Challenge. As a result of this challenge, there exist autonomous cars that reliably avoid obstacles and navigate at comparably high speeds. The focus of the Grand Challenge

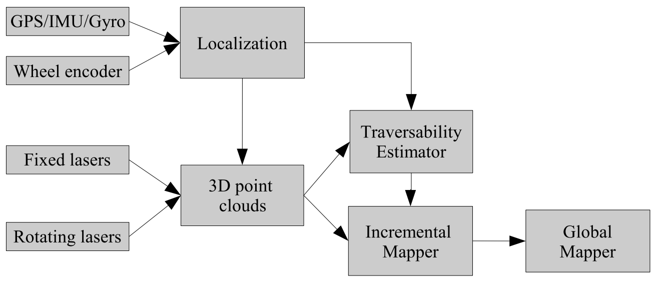

Fig. 2. The information flow between the individual modules.

was to finish the race as quickly as possible whereas certain issues like building consistent large-scale maps of the environment have been neglected since they where not needed for the race.

Even so our Smart car applied similar techniques than the winning vehicle Stanley [23] for following a specified trajectory, we have a different aim compared to the teams participating in the Grand Challenge. Our goal is to learn consistent and accurate three-dimensional models of the large-scale environments.

III. S O

Our instrumented car is equipped with a series of sensors. One group of sensor is used for localization. It consists of the inertial measurement unit, the differential GPS, the optical gyro, and the wheel encoders. The second group of sensors is given by the laser range finders. Three of them point to the front of the car and two are rotating on top of the roof of the car (see Figure 1).

Our software system is based on the modular inter-process communication (IPC) architecture. In this framework, each module can send and receive messages to/from other modules. The diagram in Figure 2 depicts the information flow between the most important modules.

IV. L

Our localization system applies the inverse form of the Kalman filter, i.e., the information filter. This filter has the property of summing information contributions from different sources in the update stage. This characteristic is advantageous when many sensors are involved which is the case in our application. The localization is done in two steps: the state prediction and the state update.

A. State Update

The localization algorithm estimates the state of the vehicle in a fixed navigation frame n which is represented by the north, west, and the altitude. The state vector contains the coordinates (x, y, z) of the vehicle and its three-dimensional orientation (roll ϕ, pitch θ and heading ψ to true north). We define the body frame b as the coordinate system attached to the vehicle. This frame is aligned with the vehicle kinematic axes (forward, left, and altitude) and its origin is placed at the center of the rear axle. The measurements models of the sensors are presented here.

• *Inertial measurement unit* (Crossbow NAV420): This unit provides sensor data with a frequency of 100 Hz that contains the measurements from 3 gyroscopes, 3 accelerometers, a 3D magnetic field sensor, and a GPS receiver. The internal digital signal processor of the unit combines the embedded sensors to provide the filtered orientation of the vehicle (roll, pitch, heading to true north) and the position (latitude, longitude, and altitude). This sensor, however, is not well adapted for ground vehicle driving at low speed. We therefore disabled the GPS and used the unit in angle mode: the unit outputs the filtered roll (ϕ_{imu}), pitch (θ_{imu}) and heading (ψ_{imu}) to magnetic north. This improves the pose estimate when driving at low speed. The measurement model for this sensor is

$$z_{imu} = \begin{bmatrix} \phi_{imu} \\ \theta_{imu} \end{bmatrix}_n = \begin{bmatrix} \phi \\ \theta \end{bmatrix}_n + v_{imu} \tag{1}$$

$$\psi_{imu} = \psi + b_{imu} + v_{himu}, \tag{2}$$

where v denotes the sensor noise and b_{imu} the offset between the heading to true north ψ and the heading measurement of the IMU. The bias b_{imu} is estimated by the filter using the heading measurements of the GPS which provides the heading to true north.

• *Car sensors*: The measurements taken by the car sensors are reported with a frequency of 100 Hz and are accessible via the CAN bus of the vehicle. The car provides the motor temperature, gas pedal position, steering wheel angle, wheel velocities, engine RPM, and some further status information. For localization, we use the velocity $\dot{x}_{odo}$ of the car from the CAN bus. Unlike a flight vehicle, the motion of a wheeled vehicle on the ground is governed by nonholonomic constraints. Under ideal conditions, there is no motion normal to the ground surface and no side slip: they can be written respectively as $\dot{z}_{odo} = 0$ and $\dot{y}_{odo} = 0$. In practice, these constraints are often violated. Thus, as in [3], we use zero mean Gaussian noise to model the extent of constraint violation. The measurement model for the odometry is then expressed as

$$z_{odo} = \begin{bmatrix} \dot{x}_{odo} \\ 0 \\ 0 \end{bmatrix}_b = \left[C_b^n\right]^T \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_n + v_{odo}, \tag{3}$$

where C_b^n is the matrix for transforming velocities expressed in the reference frame b of the car into the navigation frame n. The observation noise covariance is obtained using

$$R_{odo} = C_b^n \cdot diag\left\{\sigma_{enc}^2, \sigma_{vy}^2, \sigma_{vz}^2\right\} \left[C_b^n\right]^T, \tag{4}$$

where σ_{enc}^2 is the variance of the car velocity and σ_{vy}^2,σ_{vz}^2 are the amplitude of the noise related to the constraints.

• *Differential GPS system* (Omnistar Furgo 8300HP): This device provides the latitude, longitude, and altitude together with the corresponding standard deviation and the standard NMEA messages with a frequency of 5 Hz. In case the sensor receives the GPS drift correction signal, the unit changes automatically into the high precision GPS mode. When no correction signal is available, the device outputs standard GPS information. We use the WGS-84 standard to convert the GPS coordinates in Cartesian coordinates (x, y, z) expressed in a local navigation frame n. The heading to true north ψ is also provided by that unit in the RMC message. The measurement model for the GPS is

$$z_{gps} = \begin{bmatrix} x_{gps} \\ y_{gps} \\ z_{gps} \\ \psi_{gps} \end{bmatrix}_n = \begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix}_n + v_{gps}. \tag{5}$$

In order to reject the erroneous fixes caused by satellite constellation and multi-path change, we use the following gating function [19]

$$z^T(k) \cdot S^{-1} \cdot z(k) \leq \gamma, \tag{6}$$

where S is the innovation covariance of the observation. The value of γ is set to reject innovations exceeding the 95% threshold.

• *Optical gyroscope* (KVH DSP3000): This fiber optic gyroscope can measure very low rotation rates with a frequency of 100 Hz. It is possible to use it as a heading sensor for a comparably long period of time by integrating the angular rate (the unit provides the integrated angle). Contrary to compasses, the integrated heading is not sensitive to earth magnetic field disturbances. Finally, this unit offers much better accuracy than mechanical gyro and is not sensitive to shocks because it contains no moving parts. The measurement model for the optical gyro is

$$z_{opt} = \psi_{opt} = \psi + b_{opt} + v_{opt}, \tag{7}$$

where b_{opt} is the angular offset between the heading to true north ψ and the actual measurement of the gyro.

B. Prediction model

We apply a standard prediction model for the car which has the following form

$$\mathbf{x}_{k+1} = \begin{bmatrix} F_x & \dots & & 0 \\ \vdots & F_y & & \\ & & F_z & \vdots \\ 0 & & \dots & I_{5x5} \end{bmatrix} \cdot \mathbf{x}_k + w_k. \tag{8}$$

The state vector $\mathbf{x}$ contains the position and velocity expressed in the navigation frame n, the orientation of the vehicle represented by the three angles roll ϕ, pitch θ, yaw ψ, and the two biases b_{imu} and b_{opt}:

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \phi & \theta & \psi & b_{imu} & b_{opt} \end{bmatrix}^T \tag{9}$$

The position of the vehicle at time $k+1$ is predicted using the position and velocity at time k. This takes the form of a first order process written as

$$F_{x,y,z} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}_k, \tag{10}$$

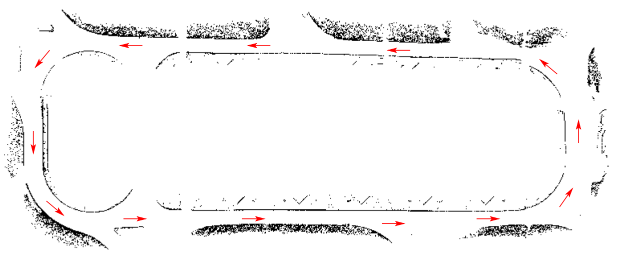

Fig. 3. Obtained traversability map using the fixed sick laser range finders. Black refers to non traversable cells and the red/gray arrows illustrate the trajectory taken by the car.

where T denotes the sampling period (10 ms). All the other elements of the state vector are predicted as simple Gaussian processes. The covariance matrix Q_k associated to the state prediction process is represented as

$$Q_k = G_k \cdot q_k \cdot G_k^T, \tag{11}$$

where q_k is a diagonal matrix containing the variances of the individual elements of the state vector

$$q_x = diag\left\{ \sigma_x^2 \quad \sigma_y^2 \quad \sigma_z^2 \quad \sigma_\phi^2 \quad \sigma_\theta^2 \quad \sigma_\psi^2 \quad \sigma_{b_{imu}}^2 \sigma_{b_{opt}}^2 \right\}. \tag{12}$$

Finally, the matrix mapping the noise covariance q_k to the process covariance Q_k is written as

$$G_k = \begin{bmatrix} g_x & \dots & & 0 \\ \vdots & g_y & & \vdots \\ & & g_z & \\ 0 & & \dots & diag_{5x5}(T) \end{bmatrix}_k, \tag{13}$$

where

$$g_{x,y,z} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}. \tag{14}$$

All in all, this information filter framework allows us to robustly and efficiently integrate the information from the different sensor into a pose estimate of the car. The pose information is provided with a high frequency and with small delays only. This is important for online control of the car.

V. T E

Whenever driving with a robot car, a central issue is to identify the obstacle-free terrain. Without a reliable estimation of the traversable area, autonomous car driving is nearly impossible. This paper does not focus on autonomous navigation, the estimation of the traversability, however, is regarded as a mapping task and therefore also addressed in this work.

The car is equipped with five SICK laser range finders whereas two are mounted on a rotating unit and three are fixed (compare Figure 1). We currently use the three static laser range finders in order to estimate the traversability of the area in front of the car. Given a laser range observation, we first compute the end points of the individual beams.

We then add the 3d points to the cells of a local two-dimensional grid map according to the x, y-coordinate of the beam. We then parse the cells and compute the mean and variance of the z-values for each cell. The decision if a cell is locally traversable can be done based on these two values. When adding the data of multiple laser range finders into a single grid, it is likely to get a series of obstacles at locations where actually no obstacle is located. This phenomenon is also described by Thrun *et al.* [23] as phantom obstacles. These phantom obstacles are caused by small errors in the pitch estimate of the location of the car, between the individual laser range scans. Therefore, we compute the traversability estimate individually for each scan and merged the independently estimated traversability values into a common grid structure. We found that this yields good results when moving on streets as well as on unpaved roads and avoids phantom obstacles. An example for a resulting traversability estimate is shown in Figure 3.

VI. M

During the mapping process, we create globally consistent maps using the inputs of the localization module and the five laser range finders mounted on the robot. We use multi-level surface maps (MLS maps) as proposed in our previous work [24]. MLS maps store in each cell of a discrete grid the height of the surface in the corresponding area. In contrast to elevation maps, MLS maps allow us to store multiple surfaces in each cell of the grid. In the remainder of this paper, these surfaces a referred to as patches. This representation enables a mobile robot to model environments with structures like bridges, underpasses, buildings or mines. Additionally, they enable the robot to represent vertical structures.

The localization technique described in Section IV works well for navigation issues. However, applying mapping with known poses based on this pose estimate usually results in globally inconsistent maps. In practice, this typically becomes apparent when the robot encounters a loop, i.e., when it returns to a previously visited place. To achieve the goal of globally consistent maps it is needed to associate the data which is acquired when the robot reaches the same place of the environment at different times. To achieve this, we build local MLS maps and apply the ICP algorithm to iteratively find constraints between poses and to solve this data association problem. This is described in detail in the reminder of this section. After the map matching and loop closing process the local MLS maps can be merged to one global consistent MLS map.

A. Data Acquisition and Local Map Building

During the data acquisition process, we collect three-dimensional points which corresponds directly to the sensed environment. The data is collected while our robot is moving continously through the environment using the five SICK laser range finders. As explained before, three of them are mounted in a fixed position and provide data points about the environment in front of our robot. Additionally, we mounted two laser range finders in vertical direction on a rotating plate on the top of the car. Figure 1 depicts the two lasers and the electric step motor. During data acquisition the step motor rotates the two laser range finders with a constant frequence of 0.37 Hz. Due to this configuration the rotating lasers provide data points which correspond to the environment in all directions around the robot. To build a local MLS map, we now use the data points acquired during a complete 360 degree turn by the rotating lasers. This setup is well-suited to build 3d maps of the environment. Furthermore, we add the data points which are acquired with the three fixed lasers during this period of time. Figure 4 depicts an example of a single local MLS map. From this point on, we discard the point clouds and perform all computations based on the local MLS maps. The example shows a typical scene of an urban environment with street lights and trees. Note that the data of all five SICK laser range finders are used for mapping. For estimating the traversable area in front of the car, however, only the three static sensor are used due to the comparable slow rotation of the rotating laser sensors.

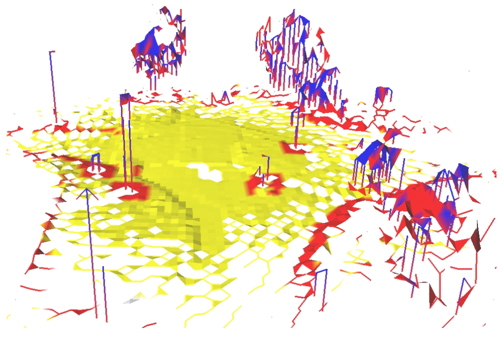
Fig. 4. Example of a single local MLS map. The example shows a typical scene of an urban environment with street lights and trees.

B. Map Matching

In addition to the traversability analysis described in Section V, we can identify vertical objects based on the 3d data. As a result, every patch in the MLS map is labeled as 'traversable', 'non-traversable', and 'vertical'. The labels are used in the ICP-based map matching process to obtain a more robust and accurate registration.

ICP seeks to find a rotation matrix R and a translation vector $\mathbf{t}$ that minimizes an error function computed based on the two maps we aim to match. We integrate the labels of the individual patches into the ICP error function in order to improve the matching result. We only consider matches between patches of the same label.

Let $\mathbf{u}$ be the vertical patches, $\mathbf{v}$ the traversable, and $\mathbf{w}$ the non-traversable ones of the first map. The cells of the second map are indicated by primed variables. We can define the following error function:

$$e(R, \mathbf{t}) = \underbrace{\sum_{c=1}^{C_1} d(\mathbf{u}_{i_c}, \mathbf{u}'_{j_c})}_{\text{vertical objects}} + \underbrace{\sum_{c=1}^{C_2} d(\mathbf{v}_{i_c}, \mathbf{v}'_{j_c})}_{\text{traversable}} + \underbrace{\sum_{c=1}^{C_3} d(\mathbf{w}_{i_c}, \mathbf{w}'_{j_c})}_{\text{non-traversable}}. \quad (15)$$

In this equation, d is the Mahalanobis distance and the indices i_c and j_c indicate the correspondence between the patches. Minimizing $e(R, \mathbf{t})$ as well as the computation of the correspondences is iterated within the ICP algorithm.

In practical experiments [16], we found that matching only patches with the same label leads to more robust and accurate map estimates. Furthermore, the ICP algorithm converges faster due to the smaller number of potential correspondences.

C. Loop Closing

The ICP-based scan matching technique described above works well for the registratering robot poses into one global reference frame. However, the individual scan matching processes result in small residual errors which accumulate over time and usually result in globally inconsistent maps. In practice, this typically becomes apparent when the robot encounters a loop, i.e., when it returns to a previously visited place. Accordingly, techniques for calculating globally consistent maps are necessary. Therefore, we apply an approach that combines the ideas of Lu and Milios [10] and Olson's algorithm [15] to globally correct the map. This approach applies error minimization via stochastic gradient descent on the whole vector of poses and yields accurate map estimates given a set of constraints between poses.

VII. E

A. Localization

Our localization system has been extensively tested and provides accurate pose estimates in a robust manner when moving though urban environments. A typical result obtained with our smart car is depicted in Figure 5. The figure represents the estimated trajectory of the car overlayed on the ortho-photo of the EPFL campus.

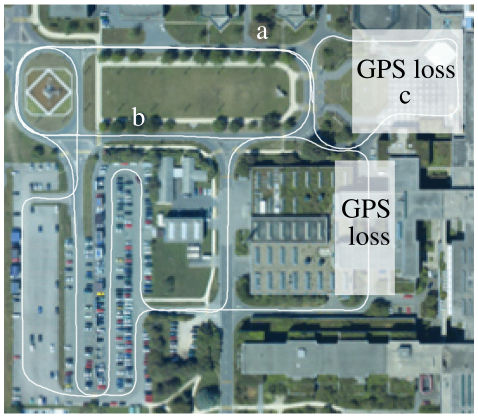

Fig. 5. Overlay of the estimated trajectory and the ortho-photo of the EPFL campus. The zones where the GPS was not available are highlighted. The total traveled distance is around 2300 m. The labels (a), (b), and (c) identify areas which are later on referred to by Figure 6 and 7.

During the experiment, the car drove in areas where the GPS quality was bad or not available, for example along narrow alleys bordered with trees, close to buildings, or in an underground parking lot. However, the localization algorithm was able to cope with GPS faults and provided accurate positioning estimation, such as depicted in Figure 6.

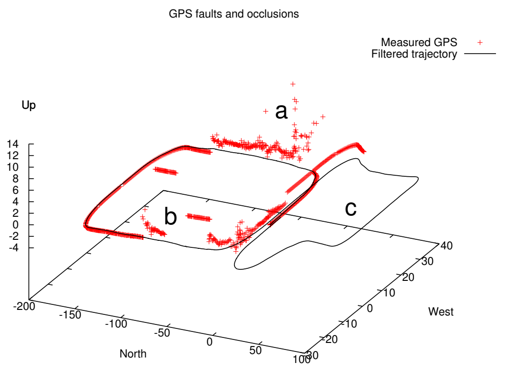

Fig. 6. This graph represents a part of the trajectory depicted in Figure 5. In this urban environment, the GPS signal is disturbed by many objects (trees, buildings, etc.) and GPS faults are of high amplitude (several meters in the horizontal plane and up to 16 m vertically). The localization algorithm was able to reject erroneous GPS fixes and to provide accurate estimations. The labels a,b and c mark areas where GPS is of poor quality (a), (b) or unavailable (c).

The uncertainty associated to the pose estimation mainly depends on the quality of the GPS fixes. As depicted in Figure 7, the standard deviation is low when differential GPS is available (~3 cm) but increases as soon as fixes are unavailable (up to 60 cm).

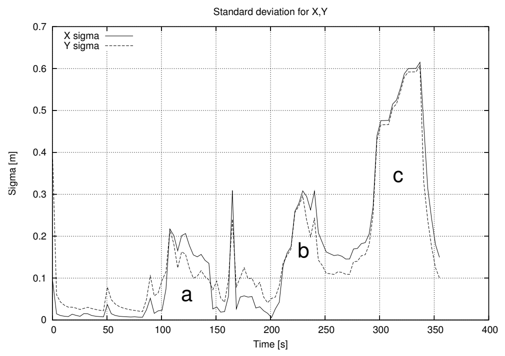

Fig. 7. Standard deviation along the north (x) and west axis (y) for the trajectory depicted in Figure 6. The standard deviation increases when GPS quality is poor and decreases as soon as it gets better. The labels (a), (b) and (c) corresponds to the zones marked in Figure 5.

B. Mapping

To acquire the data, we steered our robotic car depicted in Figure 1 over streets of the EPFL campus. The goal of these experiments is to demonstrate that our representation yields a significant reduction of the memory requirements compared to a point cloud representation, while still providing highly accurate maps. Additionally, they show that our representation is well-suited for global pose estimation and loop closure. Furthermore, the experiments show the necessity of the loop closing procedure. Figure 8 show the resulting map of a dataset acquired along a 2.3 km trajectory. Figure 9 shows a cutout of two MLS maps from that dataset. The left image depicts the resulting MLS Map when only

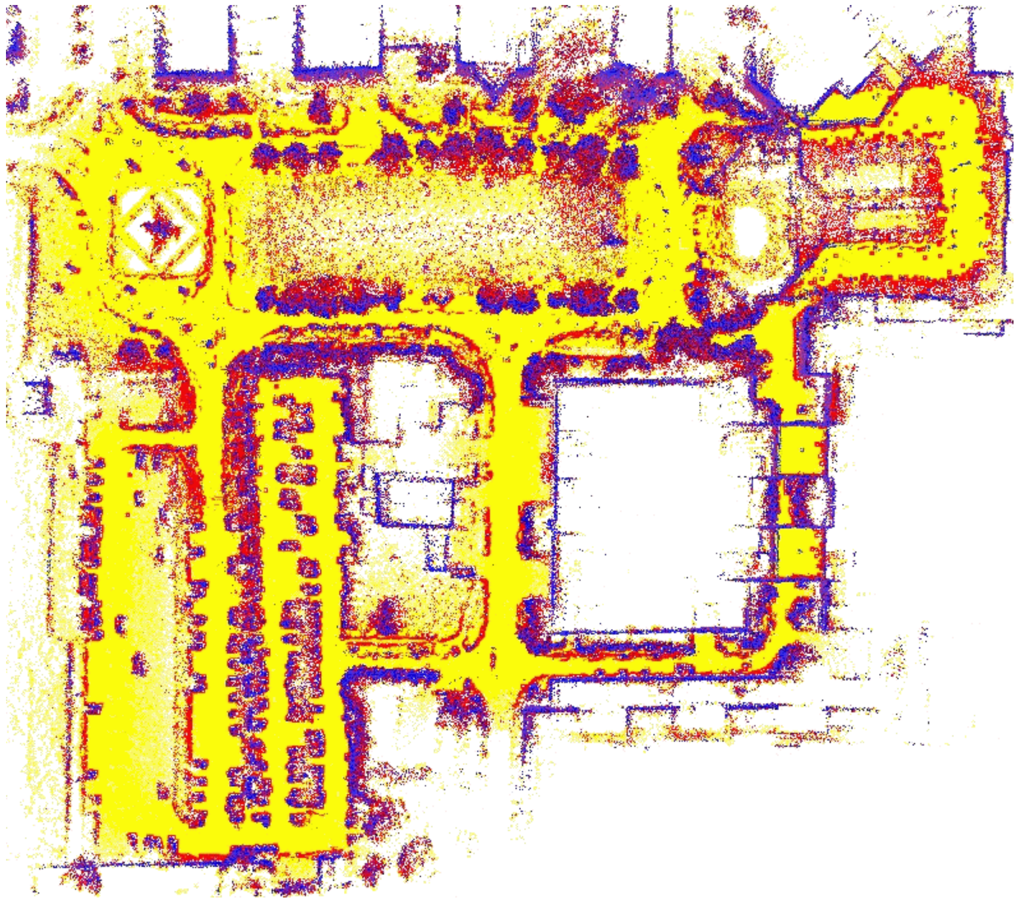

Fig. 8. Top view of the resulting MLS map with a cell size of 50cm x 50cm. The yellow/light gray surface patches are classified as traversable.The area scanned by the robot spans approximately 300 by 250 meters. During the data acquisition, the robot traversed five nested loops with a length of approximately 2,300m.

local map matching is applied. The right image displays the same part of the MLS map where we additionally applied our loop closing algorithm.

In this experiment, we acquired 374 local point clouds consisting of 68,162,000 data points. The area scanned by the robot spans approximately 300 by 250 meters. During the data acquisition, the robot traversed five nested loops with a length of approximately 2,300m. Figure 8 shows a top view of the resulting MLS map with a cell size of 50cm x 50cm. The yellow/light gray surface patches are classified as traversable. It requires 55 MB to store the computed map, where 34% of 300,000 cells are occupied. Compared to this the storage of the 68,162,000 data points requires 1,635 MB. The scan matching between the local MLS maps has been computed online during the data acquisition on a 2GHz dual core laptop computer. The loop closing step of our mapping algorithm is computed offline when the robot finished the data acquisition. In our current approach, the computation time for the optimization of the shown data set is approximately 15 minutes.

VIII. C

In this paper, we presented our approach towards mapping of large-scale areas like villages or cities. We presented the setup of our modified car and the techniques applied to learn accurate models of the environment and localize the vehicle in the world. Our map representation can be seen as an extention of elevation maps which are able to store different surfaces in the environment. In order to learn these maps, we present our pose estimation technique as well as an approach to match sub-maps in order to correct the poses based on the proximity sensors. In order to accurately close loops, we apply a least square minimization approach. As a result, we obtain high quality three-dimensional models. All techniques have been implemented and tested using a real car equipped with different types of sensors. The experiments presented in this paper, show the result of real world data obtained with this robot.

A

This work has partly been supported by the EC under contract number FP6-IST-027140-BACS, FP6-2005-IST-5-muFly, and by the German Science Foundation (DFG) within the Research Training Group 1103 and under contract number SFB/TR-8. The authors would like to thank Sascha Kolksi, Frederic Pont, and all other members of the *SMART-Team* who contributed to this work.

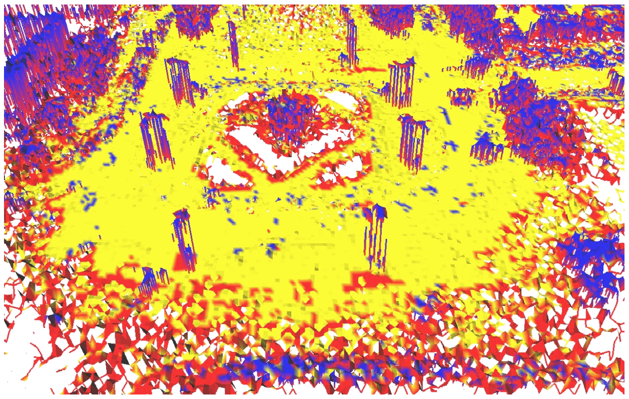
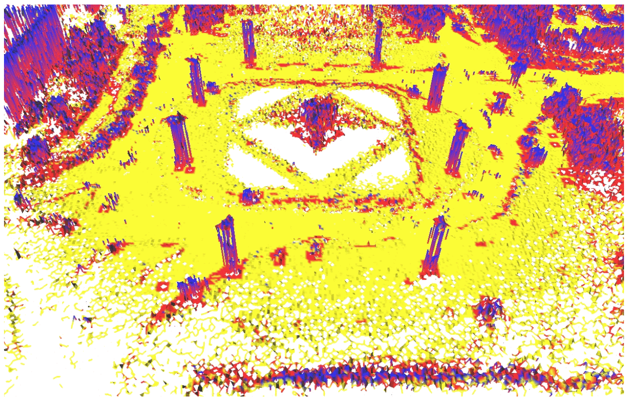

Fig. 9. This figure depicts the lower left corner of the MLS Map shown in Figure 8. The left image illustrates the resulting MLS Map when only local map matching is applied. The right image displays the same part of the MLS map where we additionally applied our loop closing algorithm. The inconsistencies can be seen by the vertical poles in the figure. Furthermore, several traversable patches have been misclassified as non traversable (red/dark gray) due to the misalignment of the maps.

R

[1] L.B. Cremean, T.B. Foote, J.H. Gillula, G.H. Hines, D. Kogan, K.L. Kriechbaum, J.C. Lamb, J. Leibs, L. Lindzey, C.E. Rasmussen, A.D. Stewart, J.W. Burdick, and R.M. Murray. Alice: An information-rich autonomous vehicle for high-speed desert navigation. *Journal of Field Robotics*, 2006. Submitted for publication.

[2] DARPA. Darpa grand challenge rulebook. Website, 2004. http://www.darpa.mil/grandchallenge05/Rules 8oct04.pdf.

[3] G. Dissanayake, S. Sukkarieh, and H. Durrant-Whyte. The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE Transactions on Robotics and Automation*, 17(5), 2001.

[4] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultainous localization and mapping without predetermined landmarks. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1135–1142, Acapulco, Mexico, 2003.

[5] 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, Barcelona, Spain, 2005.

[6] 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.

[7] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2443–2448, Barcelona, Spain, 2005.

[8] P. Kohlhepp, M. Walther, and P. Steinhaus. Schritthaltende 3D-Kartierung und Lokalisierung für mobile inspektionsroboter. In *18. Fachgespräche AMS*, 2003. In German.

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

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

[11] 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, Acapulco, Mexico, 2003.

[12] M. Montemerlo and S. Thrun. A multi-resolution pyramid for outdoor robot terrain perception. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2004.

[13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to simultaneous localization and mapping. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 593–598, Edmonton, Canada, 2002.

[14] 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.

[15] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006.

[16] Pfaff P. and Burgard W. An efficient extension of elevation maps for outdoor terrain mapping. In *Proc. of the Int. Conf. on Field and Service Robotics (FSR)*, pages 165–176, Port Douglas, QLD, Australia, 2005.

[17] S. Singh and A. Kelly. Robot planning in the space of feasible actions: Two examples. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1996.

[18] 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.

[19] S. Sukkarieh, E.M. Nebot, and H. Durrant-Whyte. A high integrity imu/gps navigation loop for autonomous land vehicle application. *IEEE Transactions on Robotics and Automation*, 15(3), 1999.

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

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

[22] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Taipei, Taiwan, 2003.

[23] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the darpa grand challenge. *Journal of Field Robotics*, 2006. To appear.

[24] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[25] C. Urmson. *Navigation Regimes for Off-Road Autonomy*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2005.

[26] J. Weingarten and R. Siegwart. 3d slam using planar segments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[27] O. Wulf, K-A. Arras, H.I. Christensen, and B. Wagner. 2d mapping of cluttered indoor environments by means of 3d perception. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 4204–4209, New Orleans, 2004.

[28] C. Ye and J. Borenstein. A new terrain mapping method for mobile robot obstacle negotiation. In *Proc. of the UGV Technology Conference at the 2002 SPIE AeroSense Symposium*, 1994.