Monte Carlo Localization in Outdoor Terrains using Multi-Level Surface Maps

Rainer Kümmerle[1], Rudolph Triebel[1,2], Patrick Pfaff[1], and Wolfram Burgard[1]

[1]Department of Computer Science, University of Freiburg, Germany,
{kuemmerl,triebel,pfaff,burgard}@informatik.uni-freiburg.de

[2]Autonomous Systems Lab, Swiss Federal Institute of Technology, Zurich, Switzerland,
rudolph.triebel@mavt.ethz.ch

Summary. In this paper we consider the problem of mobile robot localization with range sensors in outdoor environments. Our approach applies a particle filter to estimate the full six-dimensional state of the robot. To represent the environment we utilize multi-level surface maps which allow the robot to represent vertical structures and multiple levels in the environment. We describe probabilistic motion and sensor models to calculate the proposal distribution and to evaluate the likelihood of observations. Experimental results obtained with a mobile robot in an outdoor environment indicate that our approach can be used to robustly and accurately localize an outdoor vehicle. The experiments also demonstrate that multi-level surface maps lead to a significantly better localization performance than standard elevation maps.

1 Introduction

The problem of mobile robot localization with range sensors in outdoor environments arises whenever GPS signals are missing because of occlusions caused by buildings, bridges, or trees. In such situations, a mobile robot typically has to estimate its position in the environment using its other sensors and a map of the environment. In this paper, we consider the problem of localizing a mobile robot in outdoor environments by matching laser range measurements to a given map of the environment. One of the most popular representations for outdoor environments are elevation maps [3, 9, 10, 13]. The key idea underlying elevation maps is to store the $2\frac{1}{2}$-dimensional height information of the terrain in a two-dimensional grid, which corresponds to a representation of the horizontal surfaces of the environment. Whereas the knowledge about the horizontal surfaces is well suited to support traversability analysis and path planning, it provides only weak support for the localization of the vehicle. Modeling only the horizontal surfaces, namely, means that vertical structures, which are frequently perceived by ground based vehicles, cannot be used to support localization. To avoid this problem, multi-level surface (MLS) maps [16]

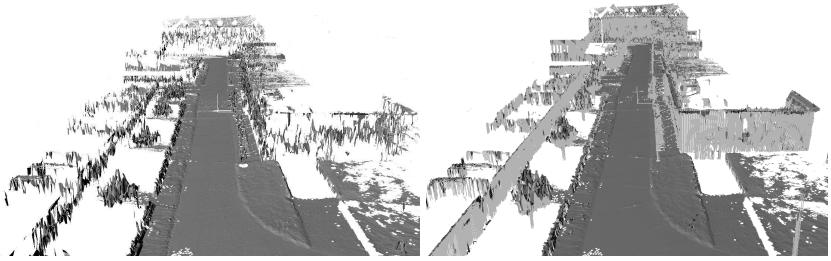

Fig. 1. Elevation Map (left) and multi-level surface (MLS) map (right) of the Freiburg campus. The MLS map represents vertical structures more accurately and can deal with multiple surfaces that can be traversed by the robot.

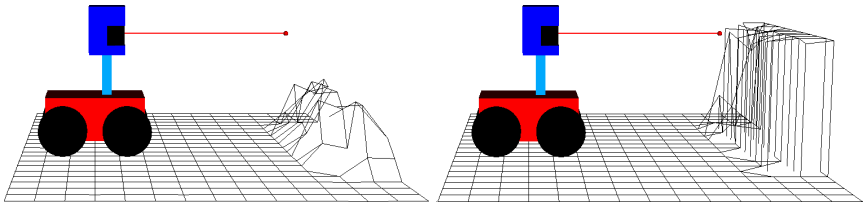

Fig. 2. Advantage of the MLS map approach in comparison to the standard elevation maps. In contrast to the MLS map (right) the elevation map (left) lacks the ability to model vertical structures, because it averages over all measured height values. Since the distance of the endpoint of a laser beam to the closest point in the elevation map can have substantial deviations from the true distance, localization becomes harder.

have been introduced. They can be regarded as an extension of the classical elevation maps as they additionally represent intervals corresponding to vertical objects in the environment. A further disadvantage of elevation maps is that they cannot represent multiple levels. This, for example, is important when mobile robots are deployed in environments with bridges or underpasses.

Fig. 1 depicts examples of an elevation map (left) and the corresponding MLS map (right) of the campus at the University of Freiburg. As can be seen from the images, the MLS map is able to represent the environment more accurately than the elevation map. In the MLS map, objects like trees and walls are represented properly. Another example is shown in Fig. 2. Here the robot is located in front of a wall. In the elevation map, the wall is not represented correctly, because the height values obtained from beams reflected by the wall are averaged which is the typical approach in elevation maps. This can lead to a poor estimate of the measurement likelihood at the particular robot position. In contrast, when the MLS map is used, one obtains a better value of the likelihood, because the wall is modeled correctly.

In this paper we present an approach to use the multi-level surface maps for localization. We present probabilistic motion and observation models and describe how these models can be utilized in a probabilistic localization scheme. We furthermore

evaluate how the localization performance changes when standard elevation maps are used instead of MLS maps.

This paper is organized as follows. After discussing related work in the next section, we briefly describe Monte Carlo localization in Section 3. Our approach to apply the MLS maps is presented in Sections 4 and 5. Finally, in Section 6, we present experimental results illustrating the advantages of applying MLS maps for localization in outdoor environments.

2 Related Work

The problem of localization of mobile robots in outdoor environments with range sensors or cameras has been studied intensively in the past. For example, Adams *et al.* [1] extract predefined features from range scanners and apply a particle filter for localization. Davison and Kita [4] utilize a Kalman filter for vision-based localization with point features on non-flat surfaces. Recently, Agrawal and Konolige [2] presented an approach to robot localization in outdoor terrains based on feature points that are tracked across frames in stereo images. Lingemann *et al.* [11] recently described a method for fast localization in in- and outdoor environments. Their system operates on raw data sets, which results in huge memory requirements. Additionally, they apply a scan-matching routine for localization, which does not facilitate global localization. To reduce the memory requirements of outdoor terrain representations, several researchers applied elevation maps [3, 9, 10, 13]. A probabilistic approach to localize a planetary rover in such elevation maps has been described by Olson [12]. In this system, elevation maps were sufficient to robustly localize the vehicle, mainly because the number of vertical and overhanging objects is negligible in environments like on Mars. However, environments on earth contain many objects like buildings or trees which have vertical or even overhanging surfaces. To address this issue, Pfaff *et al.* [14] extended the elevation map approach by a cell classification. In this approach the individual cells are divided into three classes: cells which have been observed from above, cells with vertical objects, and cells with overhanging objects. This extension has been utilized to improve the data association during the scan matching process and enabled the robot to traverse cells with overhanging objects. In contrast to MLS maps, however, this approach still lacks the ability to store multiple surfaces at one position, which prevents the robot from dealing with situations, in which it has to traverse a bridge and move through the corresponding underpass. The goal of this paper is to develop a probabilistic localization method based on MLS maps and to demonstrate that the more accurate representation of the environment results in improved localization capabilities.

3 Monte Carlo Localization

To estimate the pose $\mathbf{x} = (x, y, z, \varphi, \vartheta, \psi)$ of the robot in its environment, we consider probabilistic localization, which follows the recursive Bayesian filtering scheme. The

key idea of this approach is to maintain a probability density $p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1})$ of the robot's location $\mathbf{x}_t$ at time t given all observations $\mathbf{z}_{1:t}$ up to time t and all control inputs $\mathbf{u}_{0:t-1}$ up to time $t-1$. This posterior is updated as follows:

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1}) = \alpha \cdot p(\mathbf{z}_t \mid \mathbf{x}_t) \cdot \int p(\mathbf{x}_t \mid \mathbf{u}_{t-1}, \mathbf{x}_{t-1}) \cdot p(\mathbf{x}_{t-1})\, d\mathbf{x}_{t-1}. \tag{1}$$

Here, α is a normalization constant ensuring that $p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{0:t-1})$ sums up to one over all $\mathbf{x}_t$. The terms to be described in Eqn. (1) are the *prediction model* $p(\mathbf{x}_t \mid \mathbf{u}_{t-1}, \mathbf{x}_{t-1})$ and the *sensor model* $p(\mathbf{z}_t \mid \mathbf{x}_t)$. One major contribution of this paper is an appropriate computation of these models in the case that an MLS map is given.

For the implementation of the described filtering scheme, we use a sample-based approach which is commonly known as *Monte Carlo localization* [5]. Monte-Carlo localization is a variant of particle filtering [6] where each particle corresponds to a possible robot pose and has an assigned weight w_i. The *belief update* from Eqn. (1) is performed by the following two alternating steps:

1. In the **prediction step**, we draw for each particle with weight w_i a new particle according to w_i and to the prediction model $p(\mathbf{x}_t \mid \mathbf{u}_{t-1}, \mathbf{x}_{t-1})$.
2. In the **correction step**, a new observation $\mathbf{z}_t$ is integrated. This is done by assigning a new weight w_i to each particle according to the sensor model $p(\mathbf{z}_t \mid \mathbf{x}_t)$.

Furthermore, the particle set needs to be re-sampled according to the assigned weights to obtain a good approximation of the pose distribution with a finite number of particles. However, the resampling step can remove good samples from the filter which can lead to particle impoverishment. Thus, we need to find a criterion for deciding when to perform the resampling step. In this paper, we calculate the number N_{eff} of effective particles according to the formula proposed by Doucet *et al.* [6]

$$N_{\mathit{eff}} = \frac{1}{\sum_{i=1}^{N} \left(\tilde{w}_i^2\right)}, \tag{2}$$

where $\tilde{w}_i$ refers to the normalized weight of sample i. In particular, we only resample if N_{eff} drops below the threshold of $\frac{N}{2}$ where N is the number of samples. In the past, this approach has already successfully been applied in the context of the simultaneous mapping and localization (SLAM) problem [8].

4 Prediction Model for MLS Maps

The prediction model $p(\mathbf{x}_t \mid \mathbf{u}_{t-1}, \mathbf{x}_{t-1})$ we use is based on an approach introduced by Eliazar *et al.* [7]. It reflects systematic errors such as drift, as well as the uncertainty in the execution of an action $\mathbf{u} = (x_u, y_u, \theta_u)$, where (x_u, y_u) is the translation and θ_u the rotation angle. To incorporate this 2D motion into our 3D map we proceed as follows. First, we obtain a possible outcome (x_v, y_v, θ_v) of the action by applying the probabilistic model. Then, we adapt the motion vector $\mathbf{v} = (x_v, y_v)$ to the shape of the 3D surface traversed by the robot. This surface is obtained from the given

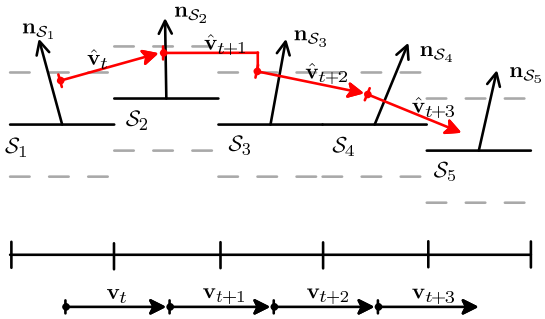

Fig. 3. Application of our prediction model to a series of 2D motion vectors (black). They are rotated to estimate the 3D motion vectors (red). The dashed line indicates the tolerance interval for the z-coordinate.

MLS map and consists of planar square patches. To adapt the motion vector, we discretize it into segments of length c, which is the cell size of the MLS map, in our case 0.1 m. For each segment, we determine the corresponding surface patch $\mathcal{S}$ and rotate the segment according to the orientation (φ_S, ϑ_S) of the patch, where φ_S is the rotation about the x-axis and ϑ_S the rotation about the y-axis. The patch orientation is computed from the normal vector $\mathbf{n}_S$ of the patch $\mathcal{S}$, which in turn is obtained by fitting a plane into the local vicinity of $\mathcal{S}$. The normal vector computation is done beforehand and constitutes an extension to the framework of MLS maps. In general, it is not robust against noise and small errors in the MLS map, which results in an uncertainty of the patch orientation. In our approach, we model this uncertainty by adding Gaussian noise to the orientation parameters φ_S and ϑ_S. Thus, our prediction model expresses the uncertainty in 5 out of 6 position parameters – x, y and ψ by the 2D motion model and φ and ϑ by our 3D extension. For the last one – the height value z – we have the constraint that the robot must stay on the ground. Therefore, we adjust the z-value manually whenever it is too high or too low. This is illustrated in Fig. 3. Finally, after concatenating all transformed motion vector segments, we obtain a new 3D motion vector $\hat{\mathbf{v}}$ which is added to the current estimate of the robot position $\mathbf{x}_{t-1}$ to obtain a new position estimate $\mathbf{x}_t$.

5 Endpoint Sensor Model for MLS Maps

In our sensor model, we treat each beam independently and determine the likelihood of a whole laser scan by factorizing over all beams. Thus, we have

$$p(\mathbf{z} \mid \mathbf{x}) = \prod_{k=1}^{K} p(z^k \mid \mathbf{x}) \tag{3}$$

where K is the number of beams in each laser measurement $\mathbf{z}$. In Eqn. (3) and in the following, we drop the index t for convenience. Our sensor model $p(z^k \mid \mathbf{x})$ is

based on an approach that has been introduced by Thrun [15] as *likelihood fields* (LF) or *end point model*. In particular, we formulate the sensor model $p(z^k \mid \mathbf{x})$ for each particular beam as a mixture of three different distributions:

$$p(z^k \mid \mathbf{x}) = \alpha_{hit} p_{hit}(z^k \mid \mathbf{x}) + \alpha_{rand} p_{rand}(z^k \mid \mathbf{x}) + \alpha_{max} p_{max}(z^k \mid \mathbf{x}), \tag{4}$$

where p_{hit} is a normal distribution $\mathcal{N}(0, \sigma^2)$ that models situations in which the sensor detects an obstacle. Random measurements are modeled using a uniform distribution $p_{rand}(z^k \mid \mathbf{x})$. Maximum range measurements are covered by a point mass distribution $p_{max}(z^k \mid \mathbf{x})$. These three distributions are weighted by the non-negative parameters α_{hit}, α_{rand}, and α_{max}, which sum up to one. The values for α_{hit}, α_{rand}, α_{max}, and σ^2 used in our current implementation have been determined empirically.

In the end point model, the probability $p_{hit}(z_k \mid \mathbf{x})$ only depends on the distance d^k between the end point of the k-th laser beam and the closest obstacle in the map. Thus, the physical property of the laser beam is ignored, because the model just uses the end point and does not consider the beam characteristic of the laser. Therefore, we need to calculate the global coordinates for a beam end point. If we denote the angle of the k-th beam relative to the zero angle with ζ^k, then the end point $\tilde{\mathbf{p}}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{z}^k)^T$ of that beam in the robot's own coordinate frame is calculated as

$$\begin{pmatrix} \tilde{x}^k \\ \tilde{y}^k \\ \tilde{z}^k \end{pmatrix} = \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} + Rz^k \begin{pmatrix} \cos(\zeta^k) \\ \sin(\zeta^k) \\ 0 \end{pmatrix}, \tag{5}$$

where $(\hat{x}, \hat{y}, \hat{z})^T$ denotes the position of the sensor at time t and R is a rotation matrix that expresses the 3D sensor orientation in the robot's coordinate frame. For a given robot pose $\mathbf{x} = (x, y, z, \varphi, \vartheta, \psi)$ at time t we can compute the global coordinates $\mathbf{p}^k = (x^k, y^k, z^k)^T$ of the k-th beam end point $\mathbf{p}^k$ as follows

$$\begin{pmatrix} x^k \\ y^k \\ z^k \end{pmatrix} = R(\varphi, \vartheta, \psi) \begin{pmatrix} \tilde{x}^k \\ \tilde{y}^k \\ \tilde{z}^k \end{pmatrix} + \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{6}$$

where $R(\varphi, \vartheta, \psi)$ denotes the rotation matrix for the given Euler angles φ, ϑ, and ψ. In MLS maps, obstacles are represented as *vertical surface patches*, which can be seen as vertical segments of occupied space. Unfortunately, there is no efficient way to find the closest of all vertical segments to a given beam end point. Therefore, we use an approximation by uniformly sampling a set $\mathcal{P}$ of 3D points from all vertical patches. The distance d^k of the k-th beam end point $\mathbf{p}^k$ to the closest obstacle is then approximated as the Euclidean distance $d(\mathbf{p}^k, \mathcal{P})$ between $\mathbf{p}^k$ and $\mathcal{P}$. This distance can be efficiently calculated by storing all points from $\mathcal{P}$ in a kD-tree.

Equations. (5) and (6) describe a 3D transform $T(z^k; \mathbf{x})$ of the measurement z^k at position $\mathbf{x}$. Using this and the fact that p_{hit} is Gaussian, we can compute p_{hit} as

$$p_{hit}(z^k \mid \mathbf{x}) \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{d(\mathbf{p}^k, \mathcal{P})}{\sigma}\right)^2\right), \tag{7}$$

where $\mathbf{p}^k = T(z^k; \mathbf{x})$. Plugging this into Eqn. (4) and the result into Eqn. (3), we obtain the entire sensor model.

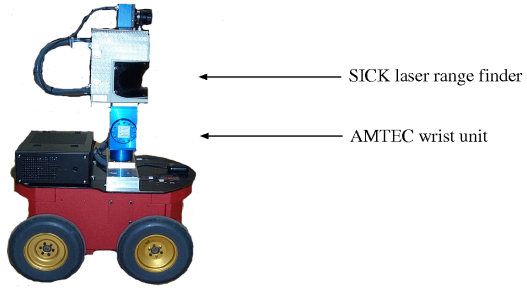

Fig. 4. Robot Herbert used for the experiments.

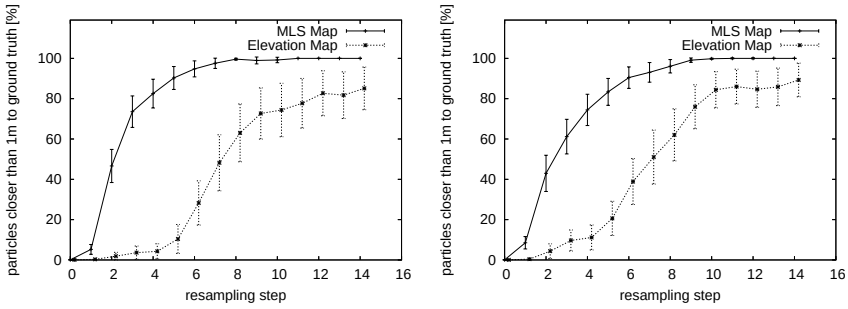

Fig. 5. Convergence of the particles to the true position of the robot with (a) 500,000 and (b) 1,000,000 particles. The *x*-axes depict the number of resampling steps, while the *y*-axes show the percentage of particles that are closer than 1*m* to the true position.

6 Experimental Results

The sensor and prediction models have been implemented in a particle filter algorithm and evaluated on real data acquired with a mobile robot. The robot is a Pioneer II AT system equipped with a SICK LMS laser range scanner and an AMTEC wrist unit, which is used as a pan/tilt device for the laser (see Figure 4). During the experiments described in this section, the laser was directed horizontally. The experiments are designed to investigate if the MLS map approach facilitates mobile robot localization and whether it yields better localization performance than the elevation maps.

6.1 Global Localization

The first set of experiments is designed to evaluate the performance of the MLS map approach in the context of a global localization task. Figure 5 depicts the convergence of the particles to the true position of the robot with 500,000 and 1,000,000 particles. Whereas the x-axis corresponds to the resampling step, the y-axis shows the number of particles in percent that are closer than 1 m to the true position, which has been

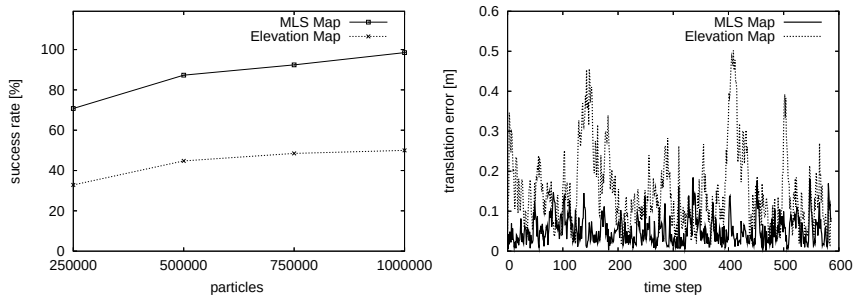

Fig. 6. The left image depicts the number of successful localizations after 15 resampling steps for the two different map representations for particle numbers from 250,000 up to 1,000,000. The right image shows the average localization error over all particles for a tracking experiment with 1,000 particles. In average the use of the MLS maps leads to smaller errors.

computed by a tracking experiment with 100,000 particles. Shown are the evolutions of these numbers when the MCL is applied on standard elevation maps and on MLS maps. Note that the elevation map does not reach 100%. This is due to the fact that the sensor model for the standard elevation map relies on a highly smoothed likelihood function, which is good for global localization but does not achieve maximal accuracy during tracking. The application of a more peaked sensor model in the case of the standard elevation map would lead to much higher divergence rates. In both cases, a t-test showed that it is significantly better to apply the MLS maps than the standard elevation maps for the global localization task. Experiments with 250,000 and 750,000 particles showed the same behavior. Figure 6 shows the number of successful localizations for the two different map representations and for different numbers of particles. Here, we assumed that the localization was achieved when every particle differed by at most 1 m from the true location of the robot. We can see that the global localization performs more robust on the MLS map than on the standard elevation map.

6.2 Tracking

We also carried out experiments, in which we analyzed the accuracy of the MLS map approach in the context of a position tracking task. To obtain the corresponding data set, we steered along a loop in our campus environment. The traversed trajectory has a length of 284 meters. Figure 7 depicts a top view of the MLS map of our test environment. The blue / dark grey line shows the localized robot poses. The yellow / light grey line shows the pure odometry. Figure 6 depicts the average localization error for a tracking experiment with 1,000 particles. As can be seen from the figure, the MLS map approach outperforms the standard elevation map approach. The tracking experiments have been computed online on a standard PC with an AMD Athlon 64 3200+ processor. In the practical experiments we found that the use of the MLS maps results in a computational overhead of no more than 10% compared to elevation maps.

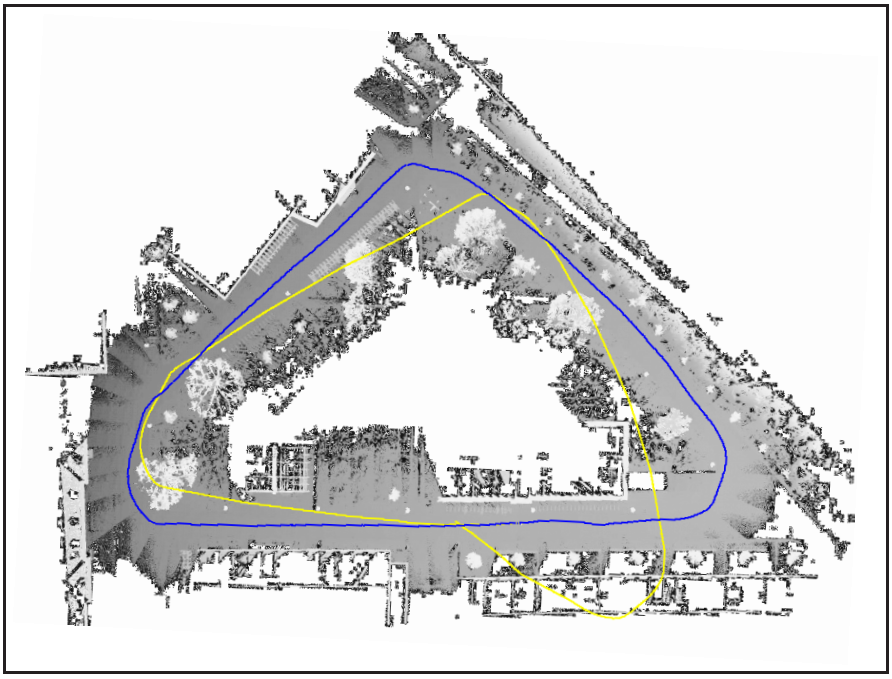

Fig. 7. MLS map used for the localization experiments. The area represented by this map spans approximately 195 by 146 meters. The blue / dark grey line shows the localized robot poses. The yellow / light grey line shows the pure odometry. The traversed trajectory has a length of 284 meters.

7 Conclusions

In this paper, we presented an approach to Monte-Carlo localization (MCL) using multi-level surface (MLS) maps. We applied the main concepts of MCL to this new 3D map representation by adapting the prediction model and the sensor model to MLS maps. Furthermore, we showed in experiments that MLS maps are better suited for the task of local and global localization than standard elevation maps. The slightly increased runtime due to the higher accuracy of MLS maps is compensated with a significantly faster convergence of the particle filter. This makes MLS maps useful for outdoor localization in cases in which no GPS signal is available.

Acknowledgment

This work has partly been supported by the German Research Foundation (DFG) within the Research Training Group 1103 and under contract number SFB/TR-8 and by the EC under contract number FP6-IST-027140-BACS and FP6-IST-034120, Action line: Micro/Nano Based Sub-Systems.

References

1. M. Adams, S. Zhang, and L. Xie. Particle filter based outdoor robot localization using natural features extracted from laser scanners. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2004.
2. M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *International Conference on Pattern Recognition (ICPR)*, 2006.
3. J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. R. L. Whittaker. Ambler: An autonomous rover for planetary exploration. *IEEE Computer Society Press*, 22(6):18–22, 1989.
4. A Davison and N. Kita. 3d simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai*. IEEE Computer Society Press, December 2001.
5. F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1999.
6. A. Doucet, N. de Freitas, and N. Gordan, editors. *Sequential Monte-Carlo Methods in Practice*. Springer Verlag, 2001.
7. A. Eliazar and R. Parr. Learning probabilistic motion models for mobile robots. In *Proc. of the International Conference on Machine Learning (ICML)*, 2004.
8. 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.
9. M. Hebert, C. Caillas, E. Krotkov, I.S. Kweon, and T. Kanade. Terrain mapping for a roving planetary explorer. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 997–1002, 1989.
10. S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury and; M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: Functions and integration. *International Journal of Robotics Research*, 21(10-11):917–942, 2002.
11. K. Lingemann, H. Surmann, A. Nüchter, and J. Hertzberg. High-speed laser localization for mobile robots. *Journal of Robotics & Autonomous Systems*, 51(4):275–296, 2005.
12. C.F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, 2000.
13. C. Parra, R. Murrieta-Cid, M. Devy, and M. Briot. 3-d modelling and robot localization from visual and range data in natural scenes. In *1st International Conference on Computer Vision Systems (ICVS)*, number 1542 in LNCS, pages 450–468, 1999.
14. P. Pfaff, R. Triebel, and W. Burgard. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *International Journal of Robotics Research*, 2007.
15. S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.
16. 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.