

Monte Carlo Localization in Outdoor Terrains using Multilevel Surface Maps

Rainer Kümmerle

Department of Computer Science
University of Freiburg
79110 Freiburg, Germany
kummerl@informatik.uni-freiburg.de

Rudolph Triebel

Autonomous Systems Lab
Swiss Federal Institute of Technology
8092 Zurich, Switzerland
rudolph.triebel@mavt.ethz.ch

Patrick Pfaff

Department of Computer Science
University of Freiburg
79110 Freiburg, Germany
pfaff@informatik.uni-freiburg.de

Wolfram Burgard

Department of Computer Science
University of Freiburg
79110 Freiburg, Germany
burgard@informatik.uni-freiburg.de

Abstract

We propose a novel combination of techniques for robustly estimating the position of a mobile robot in outdoor environments using range data. Our approach applies a particle filter to estimate the full six-dimensional state of the robot and utilizes multilevel surface maps which, in contrast to standard elevation maps, 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. We furthermore describe an active localization approach which actively selects the sensor orientation of the 2D laser range scanner to improve the localization results. To efficiently calculate the appropriate orientation we apply a clustering operation on the particles and only evaluate potential orientations based on these clusters. Experimental results obtained with a mobile robot in large-scale outdoor environments indicate that our approach yields robust and accurate position estimates. The experiments also demonstrate that multilevel surface maps lead to a significantly better localization performance than standard elevation maps. They additionally show that further accuracy is obtained from the active sensing approach.

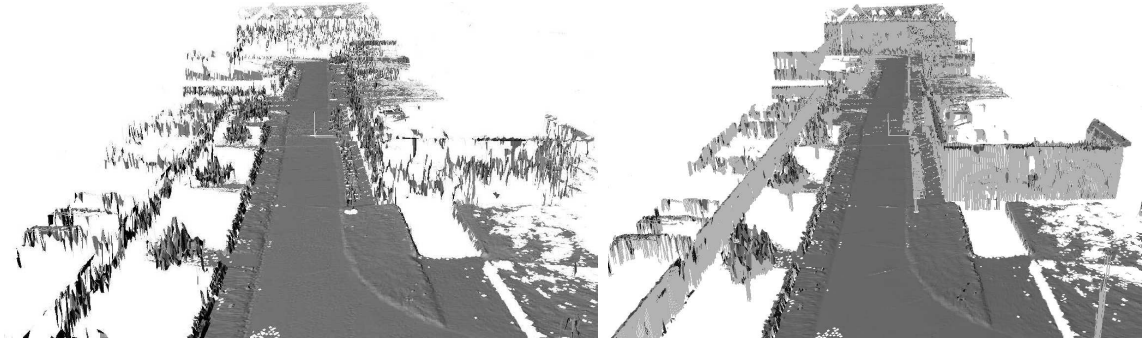


Figure 1: Elevation map (left) and multilevel surface (MLS) map (right) of the Freiburg campus. The MLS map represents vertical structures more accurately and can deal with multiple surfaces that can be traversed by the robot.

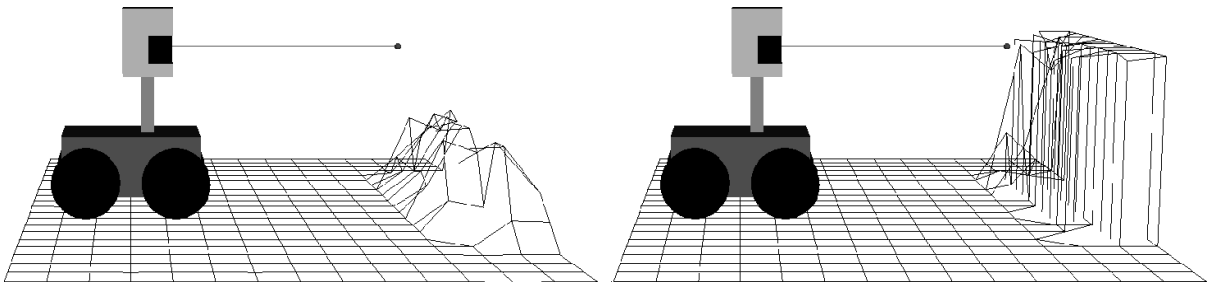


Figure 2: Advantage of the MLS map approach in comparison to the standard elevation maps. In contrast to the MLS map (right) the elevation map (left) lacks the ability to model vertical structures, because it averages over all measured height values. Since the distance of the endpoint of a laser beam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

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 (Bares et al., 1989; Hebert et al., 1989; Lacroix et al., 2002; Parra et al., 1999). The key idea underlying elevation maps is to store the 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, multilevel surface (MLS) maps (Triebel et al., 2006) have been introduced. They can be regarded as an extension of

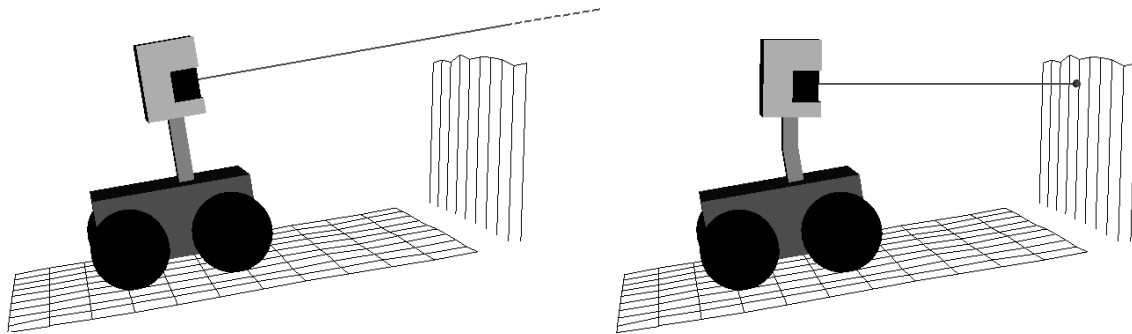


Figure 3: Robot with the standard orientation of the range sensor where the robot is unable to sense the vertical object (left). In contrast to this the autonomously adapted sensor orientation allows the robot to sense the vertical object (right).

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.

However, when a robot attempts to perceive its environment to localize itself, the choice of the direction of the perception can substantially influence the accuracy of the position estimate even when MLS maps are used instead of elevation maps. An example situation is shown in Figure 3. In the left image, the range sensor of the robot is oriented parallel to the ground plane as in most robot scenarios with a 2D sensor setup. This has the effect, that the vertical object shown in the image cannot be sensed by the robot. However, this vertical object might be crucial for localization, because it might allow the robot to reduce its uncertainty. In contrast, the right image shows a robot with a slightly different sensor orientation so that the vertical object can be perceived. Accordingly, the robot can achieve a more accurate position estimate by actively orienting its sensor. This is why the technique is called *active localization*.

In this paper we present a novel combination of techniques to effectively localize a mobile robot in outdoor environments. Our approach uses a particle filter and matches range measurements with multilevel surface maps. We present probabilistic motion and observation models and describe how these models can be utilized in a probabilistic localization scheme. In addition, we present an active localization approach which is able to efficiently determine

appropriate sensor orientations by only considering clustered subsets of the entire particle set. A comparison with standard elevation maps demonstrates that our approach leads to a superior localization performance.

This paper is organized as follows. After discussing related work in the next section, we briefly describe Monte Carlo localization in Section 3. The motion and observation models for localization with MLS maps are presented in Sections 4 and 5. In Section 6, we describe Active Monte Carlo Localization and its application to MLS maps. Finally, in Section 7, we present experimental results illustrating the advantages of our approach 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. (2004) extract predefined features from range scanners and apply a particle filter for localization. Davison and Kita (2001) utilize a Kalman filter for vision-based localization with point features on non-flat surfaces. Recently, Agrawal and Konolige (2006) presented an approach to robot localization in outdoor terrains based on feature points that are tracked across frames in stereo images. Lingemann et al. (2005) 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 (Bares et al., 1989; Hebert et al., 1989; Lacroix et al., 2002; Parra et al., 1999). A probabilistic approach to localize a planetary rover in such elevation maps has been described by Olson (2000). 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. (2007) 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.

In general, the problem of active localization can be described as generating robot actions that particularly aim at improving its position estimate. In the past, this problem has been addressed by several authors. For example, Kaelbling et al. (1996) and Koenig and Simmons (1998) used a partially observable Markov decision process to model actions in the environment. The action selection process chooses the action that minimizes the expected

entropy after the next control action or maximizes the expected total reward, respectively. Jensfelt and Kristensen (2001) applied multi-hypothesis localization to topological maps for active global localization. Davison and Kita (2001) described a vision-based localization in which the robot actively senses the features based on visibility and information gain. Recently, Porta et al. (2005) proposed an entropy-based criterion for action selection within a localization algorithm using a stereo vision system, which allows the robot to recover its location in the initial stages or within a failure recovery procedure more efficiently. Fox et al. (1998) proposed an approach for active localization based on Markov localization using fine-grained grid maps and laser range finder. Their approach is able to increase the efficiency of the localization by minimizing the expected entropy calculated based on the dominant peaks in the given posterior. In contrast to these techniques, this paper focuses on efficient active localization in outdoor terrains represented by MLS maps and given a particle filter implementation of the underlying probabilistic localization approach.

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

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

Here, α is a normalization constant which ensures that $p(\mathbf{x}_t \mid \mathbf{s}_{1:t}, \mathbf{u}_{0:t-1})$ 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* $p(\mathbf{s}_t \mid \mathbf{x}_t)$. 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* (Dellaert et al., 1999). Monte-Carlo localization is a variant of particle filtering (Doucet et al., 2001) where each particle corresponds to a possible robot pose and has an assigned weight $w^{[i]}$. The *belief update* from 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{s}_t is integrated. This is done by assigning a new weight $w^{[i]}$ to each particle according to the sensor model $p(\mathbf{s}_t \mid \mathbf{x}_t)$.

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. To decide when to perform the resampling step, we calculate the

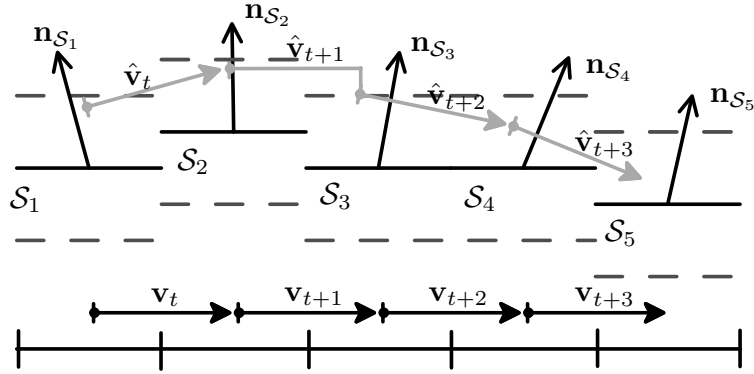


Figure 4: Application of our prediction model to a series of 2D motion vectors (black). They are rotated to estimate the 3D motion vectors (grey). The dashed line indicates the tolerance interval for the z -coordinate.

number N_{eff} of effective particles according to the formula proposed by Doucet et al. (2001)

$$N_{eff} = \frac{1}{\sum_{i=1}^N \left(\widetilde{w}^{[i]} \right)^2}, \quad (2)$$

where $\widetilde{w}^{[i]}$ refers to the normalized weight of sample i and 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 localization and mapping (SLAM) problem by Grisetti et al. (2005).

4 Prediction Model for MLS Maps

In the past, several approaches for realizing the prediction model $p(\mathbf{x}_t \mid \mathbf{u}_{t-1}, \mathbf{x}_{t-1})$ in the context of mobile robot localization using particle filters have been proposed (Dellaert et al., 1999; Eliazar and Parr, 2004). The majority of approaches has been developed for mobile robots operating on a planar surface, an assumption that is typically violated for robots deployed in outdoor environments. To appropriately predict potential positions of the vehicle in such environments, we need to take into account the three-dimensional surface structure. In the remainder of this section, we present an extension of the model proposed by Eliazar and Parr (2004) to the case of five dimensions (x , y , φ (roll), ϑ (pitch), and ψ (yaw)), where the sixth parameter, the height, is given as a constraint specified by the MLS map.

The motion model for MLS maps reflects systematic errors such as drift, as well as the uncertainty in the execution of an action $\mathbf{u} = (x_u, y_u, \psi_u)$, where (x_u, y_u) is the translation and ψ_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, ψ_v) of the action by applying the probabilistic model (Eliazar and Parr, 2004). 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 MLS map and consists of planar square patches. To adapt the motion vector, we discretize it into segments corresponding to the size of the cells in the MLS map, which usually is 0.1 m

in our current system. 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 sixth parameter – the height value z – we have the constraint that the robot must stay on the ground. Therefore, we adjust the z -value to the height at the current location. This is illustrated in Fig. 4. 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 .

In general, the proposed prediction model is applicable whenever an estimate for the surface and its normal is available. Note that standard elevation maps are able to provide such an estimate, if there is only one height level per map cell, i.e., there is no overhanging object above the drivable surface. Therefore, the same prediction model can be utilized for elevation maps, if the above-stated assumption holds.

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

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

where K is the number of beams in each measurement \mathbf{s} from the laser sensor. In Eqn. (3) and in the following, we drop the index t for convenience. Our sensor model $p(s^k \mid \mathbf{x})$ is based on an approach that has been introduced by Thrun (2001) as *likelihood fields* (LF) or *end point model*. In particular, we formulate the sensor model $p(s^k \mid \mathbf{x})$ for each particular beam as a mixture of three different distributions:

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

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 $p_{rand}(s^k \mid \mathbf{x})$. Maximum range measurements are covered by a point mass distribution $p_{max}(s^k \mid \mathbf{x})$. 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 $p_{hit}(s^k \mid \mathbf{x})$ 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

$$\begin{pmatrix} \tilde{x}^k \\ \tilde{y}^k \\ \tilde{z}^k \end{pmatrix} = \tilde{R}s^k \begin{pmatrix} \cos(\zeta^k) \\ \sin(\zeta^k) \\ 0 \end{pmatrix} + \mathbf{q}, \quad (5)$$

where \mathbf{q} denotes the 3D position of the sensor at time t and \tilde{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 $\hat{\mathbf{p}}^k = (\hat{x}^k, \hat{y}^k, \hat{z}^k)^T$ of the k -th beam end point $\hat{\mathbf{p}}^k$ as follows

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

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 $\hat{\mathbf{p}}^k$ to the closest obstacle is then approximated as the Euclidean distance $d(\hat{\mathbf{p}}^k, \mathcal{P})$ between $\hat{\mathbf{p}}^k$ and \mathcal{P} . This distance can be efficiently calculated by storing all points from \mathcal{P} in a k D-tree (see Bentley (1975)).

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

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

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

5.1 Difference between MLS Maps and Standard Elevation Maps

As mentioned earlier, the major advantage of MLS maps over standard elevation maps is their ability to represent the real environment more accurately. This has a major influence on the parameters of the sensor model that can be used for an optimal localization result. In particular, the variance of the distribution p_{hit} has to be much higher in the case of the elevation map, because it is more likely that a real sensor measurement deviates from the corresponding virtual measurement computed from the map. In other words, the localization algorithm can rely on more accurate information when using MLS maps, because the map representation is closer to the situation in the real world. Using the same sensor variance together with an elevation map would result in an overconfident sensor model, causing the localization algorithm to diverge.

6 Active Monte Carlo Localization

The purpose of our active localization approach is to find the orientation of the laser range finder which reduces the uncertainty of the current pose estimate as much as possible. To achieve this, we apply the greedy approach of Fox et al. (1998). We assume that at a given time step t the robot is able to execute a discrete set of actions \mathcal{A} . The benefit of a sensing action $a \in \mathcal{A}$ can be determined by considering the uncertainty of the posterior $p(\mathbf{x}_{t+1} | a, \mathbf{s}_{t+1})$ after obtaining the sensor measurement \mathbf{s}_{t+1} . The uncertainty of the pose estimate is represented by the entropy

$$h(\mathbf{x}_t) = - \int_{\mathbf{x}_t} bel(\mathbf{x}_t) \log bel(\mathbf{x}_t) d\mathbf{x}_t, \quad (8)$$

where $bel(\mathbf{x}_t)$ is the current posterior about the pose of the vehicle. To determine the optimal direction of the laser range finder, we consider for each potential action a its associated information gain $g_t(a)$ defined by

$$g_t(a) = h(\mathbf{x}_t) - h(\mathbf{x}_{t+1} | a, \mathbf{s}_{t+1}), \quad (9)$$

where $h(\mathbf{x}_{t+1} | a, \mathbf{s}_{t+1})$ is the entropy after the integration of a laser measurement according to the action a . In general, we do not know which range measurement the robot will obtain after changing the sensor orientation according to the action a . Therefore, we have to consider the *expected entropy* by integrating over all possible measurements \mathbf{s}_{t+1} :

$$E[g_t(a)] = h(\mathbf{x}_t) - E[h(\mathbf{x}_{t+1} | a, \mathbf{s}_{t+1})]. \quad (10)$$

Here, $E[h(\mathbf{x}_{t+1} | a, \mathbf{s}_{t+1})]$ defines the expected entropy after the integration of a laser measurement obtained by executing action a . If we take into account that changing the orientation of the range sensor does not change the location of the robot then, according to the reasoning by Fox et al. (1998), the expected entropy is calculated as follows:

$$E[h(\mathbf{x}_{t+1} | a, \mathbf{s}_{t+1})] = - \int_{\hat{\mathbf{s}}} \int_{\mathbf{x}_t} p(\hat{\mathbf{s}} | \mathbf{x}_t) bel(\mathbf{x}_t) \log \frac{p(\hat{\mathbf{s}} | \mathbf{x}_t) bel(\mathbf{x}_t)}{p(\hat{\mathbf{s}} | a)} d\mathbf{x}_t d\hat{\mathbf{s}} \quad (11)$$

Now the optimal action \hat{a} which results in the maximum information gain over all actions from an action set \mathcal{A} can be determined as follows:

$$\hat{a} = \operatorname{argmax}_{a \in \mathcal{A}} E[g_t(a)] = \operatorname{argmax}_{a \in \mathcal{A}} h(\mathbf{x}_t) - E[h(\mathbf{x}_t | a, \mathbf{s}_{t+1})]. \quad (12)$$

Unfortunately, the calculation of the expected entropy $E[h(\mathbf{x}_t | a, \mathbf{s}_{t+1})]$ is highly complex as it requires to integrate over all potential measurements and additionally needs to be carried out for all potential positions or poses contained in the current particle set. As the first operation is exponential in the dimensionality of the measurement, one in practice only considers subsets of measurements. In our current implementation, we only consider a single measurement which is obtained by a ray-casting operation carried out depending on the position \mathbf{x} and the action a . Because this measurement has a very high likelihood compared to all other potential measurements, this quantity can be assumed to dominate the integral.

Performing a ray casting operation for each particle of our particle set still imposes high computational demands. To reduce the required computation time, we only simulate the

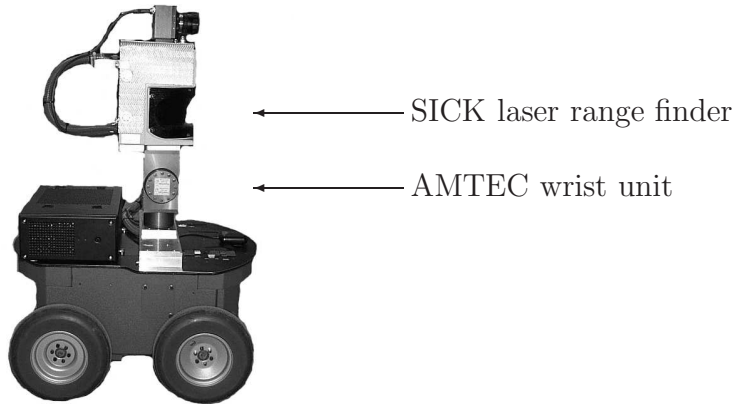


Figure 5: Robot Herbert used to carry out the experiments.

range beams on a subset of particles, which is calculated by a clustering operation. This is motivated by the fact that typically the particles are located in a small number of areas with high probability. Each cluster of particles represents such an area. To cluster the particles, we apply a technique known as QT-Clustering (Heyer et al., 1999). This method allows us to specify the maximal extent of a cluster beforehand and thus ensures that the centroid of each cluster accurately represents the whole cluster. In our current implementation the maximal diameter of a cluster is set to 1 m. The clustering algorithm yields a cluster set $\mathcal{K} = \bigcup_{j=1}^J (\mathbf{m}_j, \mathcal{I}_j)$, where \mathbf{m}_j is the center of mass and \mathcal{I}_j is an index set of the particles contained in cluster j , where each particle is a member of exactly one cluster.

The range measurement generated for a subset is weighted according to the sum of the normalized weights of the particles contained in the subset. Following this approximation and also considering the underlying particle filter implementation, the expected entropy (11) is estimated as follows:

$$E[h(\mathbf{x}_{t+1} | a, \mathbf{s}_{t+1})] \approx - \sum_{j=1}^{|\mathcal{K}|} \sum_{i=1}^N w(\mathcal{I}_j) p(\mathbf{s}_{\mathbf{m}_j}^a | \mathbf{x}^{[i]}) \cdot w^{[i]} \log \frac{p(\mathbf{s}_{\mathbf{m}_j}^a | \mathbf{x}^{[i]}) \cdot w^{[i]}}{p(\mathbf{s}_{\mathbf{m}_j}^a | a)}, \quad (13)$$

where $w(\mathcal{I}_j)$ refers to the sum of the normalized weights of the particles contained in cluster j . The term $\mathbf{s}_{\mathbf{m}_j}^a$ stands for the measurement obtained by the ray casting operation whose simulated laser beams originate from the center of mass \mathbf{m}_j with a sensor orientation according to the action a .

Based on Eqn. (13) we can efficiently calculate an approximation of the future expected entropy for action a . By plugging this into Eqn. (12) we are able at any point in time to select the best action \hat{a} to be executed next. In our practical experiments described below we never found any evidence that these approximations yield results that are worse than a passive approach with a fixed sensor orientation.

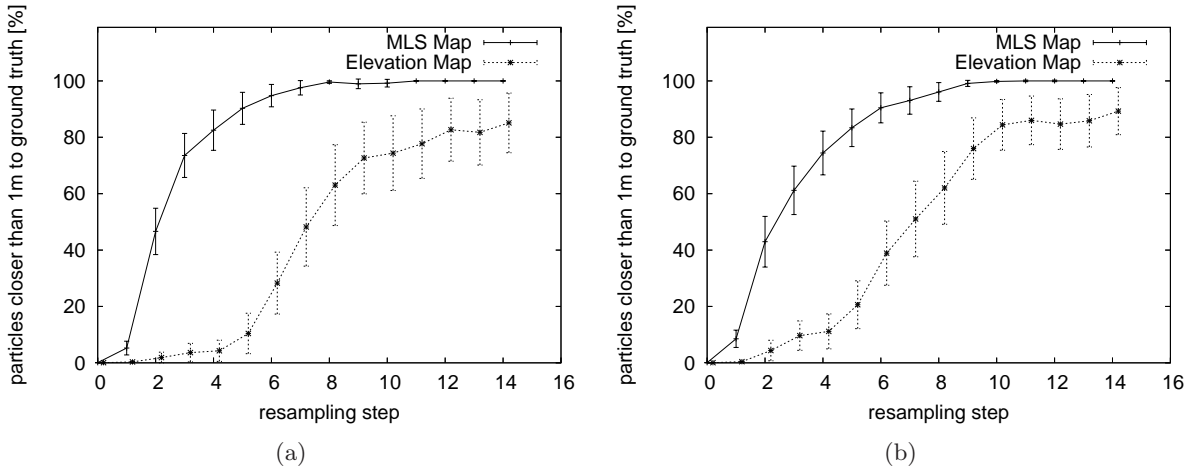


Figure 6: Convergence of the particles to the estimated ground truth of the robot with (a) 500,000 and (b) 1,000,000 particles. Whereas the x -axes depict the number of resampling steps, the y -axes show the percentage of particles that are closer than 1 m to the estimated ground truth.

7 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 5). The MLS maps used in these experiments have been generated using the approach by Triebel et al. (2006). The resolution of these maps is 10 cm per grid cell.

7.1 Passive Localization

The experiments presented within this subsection are designed to investigate if the MLS map approach facilitates mobile robot localization and whether it yields better localization results than obtained with elevation maps. We analyze the performance of our approach in the context of two different localization tasks, which are *position tracking* where the initial pose of the robot is known and *global localization* without any prior knowledge about the robot's position. To evaluate the performance of our algorithm, we compare the results with a ground truth localization. As this ground truth is in general impossible to obtain for real-world data sets, we approximate it with the robot positions obtained from a tracking experiment with a large number of particles. The number of particles is chosen large enough, such that no larger number of particles would result in a change of the obtained robot positions. In our experiments, this number was found to be 100,000. We also note that during all experiments on passive localization, the laser was directed horizontally in a fixed position.

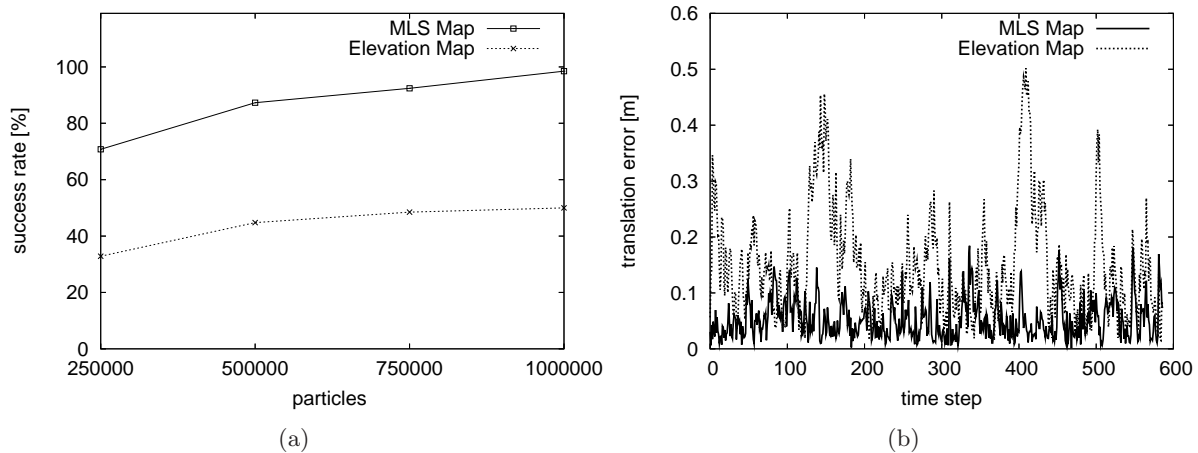


Figure 7: The left image depicts the number of successful localizations after 15 resampling steps for the two different map representations for particle numbers from 250,000 up to 1,000,000. The right image shows the average localization error over all particles for a tracking experiment with 1,000 particles. In average the use of the MLS maps leads to smaller errors.

7.1.1 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. The experiments have been carried out in an outdoor environment on the campus of the University of Freiburg. A top view of the map of this environment is depicted in Figure 10. The map covers an area of approximately 195 m by 146 m. To obtain the data used in this set of experiments, we steered the robot along a trajectory that traverses all regions of the map. Figure 6 depicts the convergence of the particles to our estimated ground truth position of the robot with 500,000 and 1,000,000 particles, respectively. 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 ground truth position. Shown are the evolutions of these numbers when the Monte Carlo localization 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 needs to have a higher variance, as discussed in Sec. 5.1. For the results depicted within this section, the sensor noise variance was chosen so that the algorithm performs best on the respective map representation. With both particle numbers, a t-test revealed 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 7(a) 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 is more reliably achieved on the MLS map than on the standard elevation map.

7.1.2 Position 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 m. Figure 10 depicts a top view of the MLS map of our test environment. The dark grey line shows the estimated robot poses whereas the light grey line indicates the odometry data. Figure 7(b) 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. Using 1,000 particles each filter update requires 40 milliseconds in the case of the MLS maps and 35 milliseconds for standard elevation maps.

One advantage of using MLS maps compared to elevation maps is the possibility to represent overhanging objects, e.g., trees and underpasses, as well as multiple levels per cell. To investigate this property of the MLS maps with respect to a localization task, we steered the robot along a 540 m long loop through our campus environment. Figure 8 depicts a top view of the environment. The traversed trajectory guides the robot through an underpass, up a ramp, and over the bridge corresponding to the underpass. Figure 9 depicts the trajectory as it is estimated by our technique in real time using 1,000 particles. While driving along this trajectory the robot traverses several cells of the map multiple times, i.e., the (x, y) coordinates are equal whereas z – the height coordinate – differs when driving on different surface levels of the environment. The average translation error while driving along the trajectory is 0.04 m and the average rotation error is less than one degree per rotation axis. A standard elevation map is unable to accurately represent the underpass and the bridge. Therefore, tracking the position of the robot using an elevation map of this environment fails, as it is unable to represent the different levels per cell.

7.2 Active Localization

The experiments presented within this subsection are designed to investigate if the active localization approach facilitates mobile robot localization and whether it improves localization performance. In our current implementation, the active selection of the preferred direction of the scan increases the runtime per update by 25%.

The first set of experiments for active localization is designed to evaluate the performance of the active localization approach during a position tracking task. To obtain the data, we steered along a 284 m long loop on our campus. Figure 10 depicts a top view of the MLS map. The dark grey line shows the localized robot poses. The light grey line shows the pure odometry. The left image of Figure 11 depicts the average localization error for a tracking experiment with 1,000 particles. As can be seen from the figure, our active approach achieves the same performance as the passive approach as long as the laser range finder is not tilted more than 3° downwards. This is due to the fact that at higher tilt angles the robot perceives less vertical objects and therefore misses the map features that are important for a correct

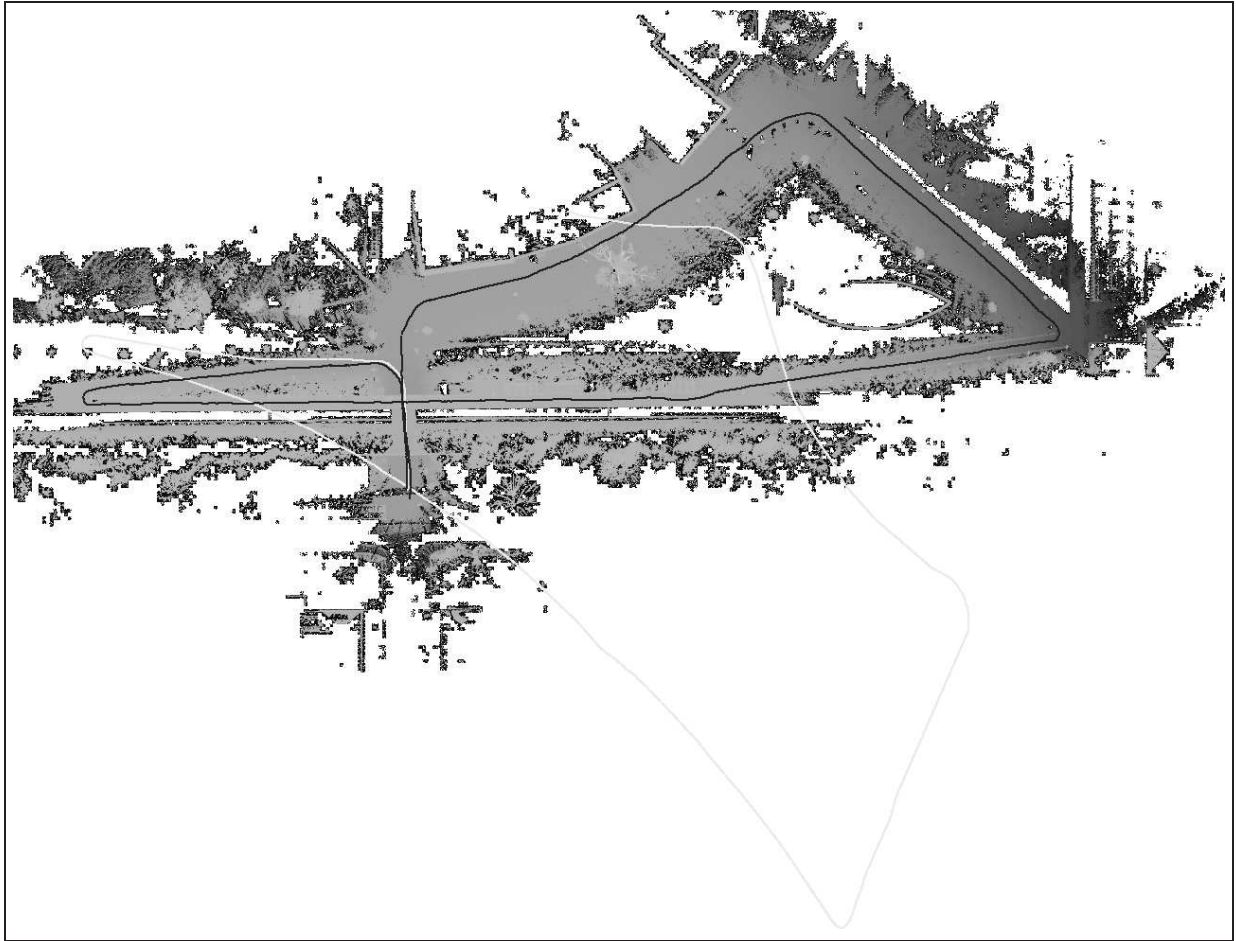


Figure 8: MLS map used for the localization experiments. The area represented by this map spans approximately 300 m by 150 m. Whereas the dark grey line shows the estimated robot poses, the light grey line indicates the odometry data. The traversed trajectory has a length of 540 m.

position estimate. The right image depicts the frequency of the tilt angles chosen by our active approach. As can be seen from the histogram our active approach prefers upwards to downwards orientations of the range sensor, whereas an upward orientation corresponds to a negative tilt angle configuration. Thus, the active approach enables the robot to sense the important vertical features of the environment and avoids to obtain range measurements from the ground.

Additionally we also carried out experiments, in which we evaluated the convergence of the active localization approach during global localization in our campus environment. The map spans approximately 195 m by 146 m. Figure 12 depicts the convergence of the particles to the true position of the robot with 500,000 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. The figure shows the evolution of these numbers for the passive and active Monte Carlo localization. A t-test revealed that it is significantly better to apply the active approach than the passive one for the global localization task.

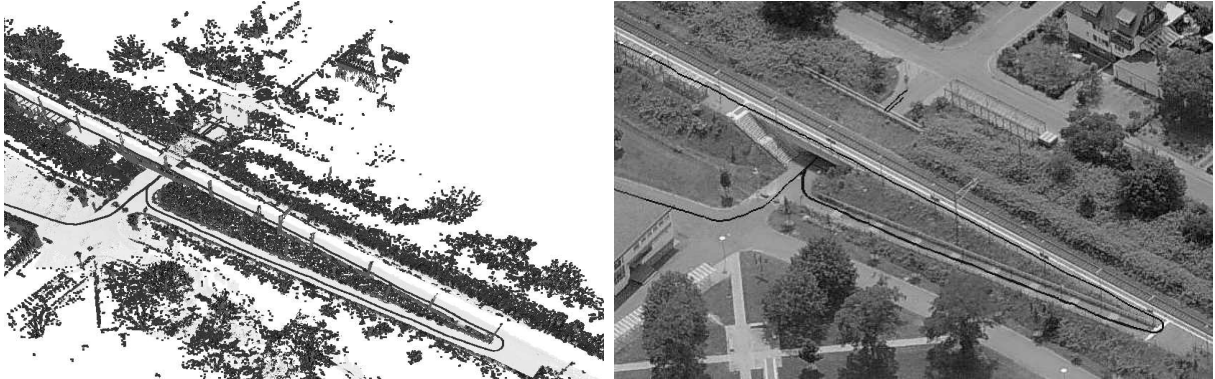


Figure 9: The left image depicts the trajectory (black) as it is estimated by our localization method projected into a rendered model of the MLS map. The right image shows an overlay of the same trajectory on an aerial image of the same region. In this experiment the robot had to traverse an underpass and then move over the corresponding bridge.

8 Conclusions

In this paper, we presented a novel combination of techniques for robust Monte-Carlo localization of a mobile robot in outdoor-terrains represented by multilevel surface maps. We presented appropriate prediction and observation models and introduced an approach to efficiently calculate the preferred direction of the range scanner. The approach has been implemented and tested on a real robot in a urban environment with a non-flat structure and multiple levels.

Experimental results carried out on real data demonstrated that our approach allows the robot to reliably and accurately estimate its position. The experiments furthermore show that multilevel surface maps are better suited for the task of position tracking and global localization than standard elevation maps. The slightly increased runtime introduced by the use of multilevel surface maps is compensated by a significantly higher robustness and accuracy. This makes multilevel surface maps useful for three-dimensional environments and outdoor terrains where no GPS signal is available. Additionally, we presented an approach to efficiently calculate the preferred orientation of the laser range finder to improve the localization results. To speed up the calculation, we spatially cluster the particles and use the resulting clusters to evaluate potential orientations. In experiments obtained with a robot in a real outdoor environment we found that the active control of the direction of the scanner further improves the localization results.

Acknowledgments

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.

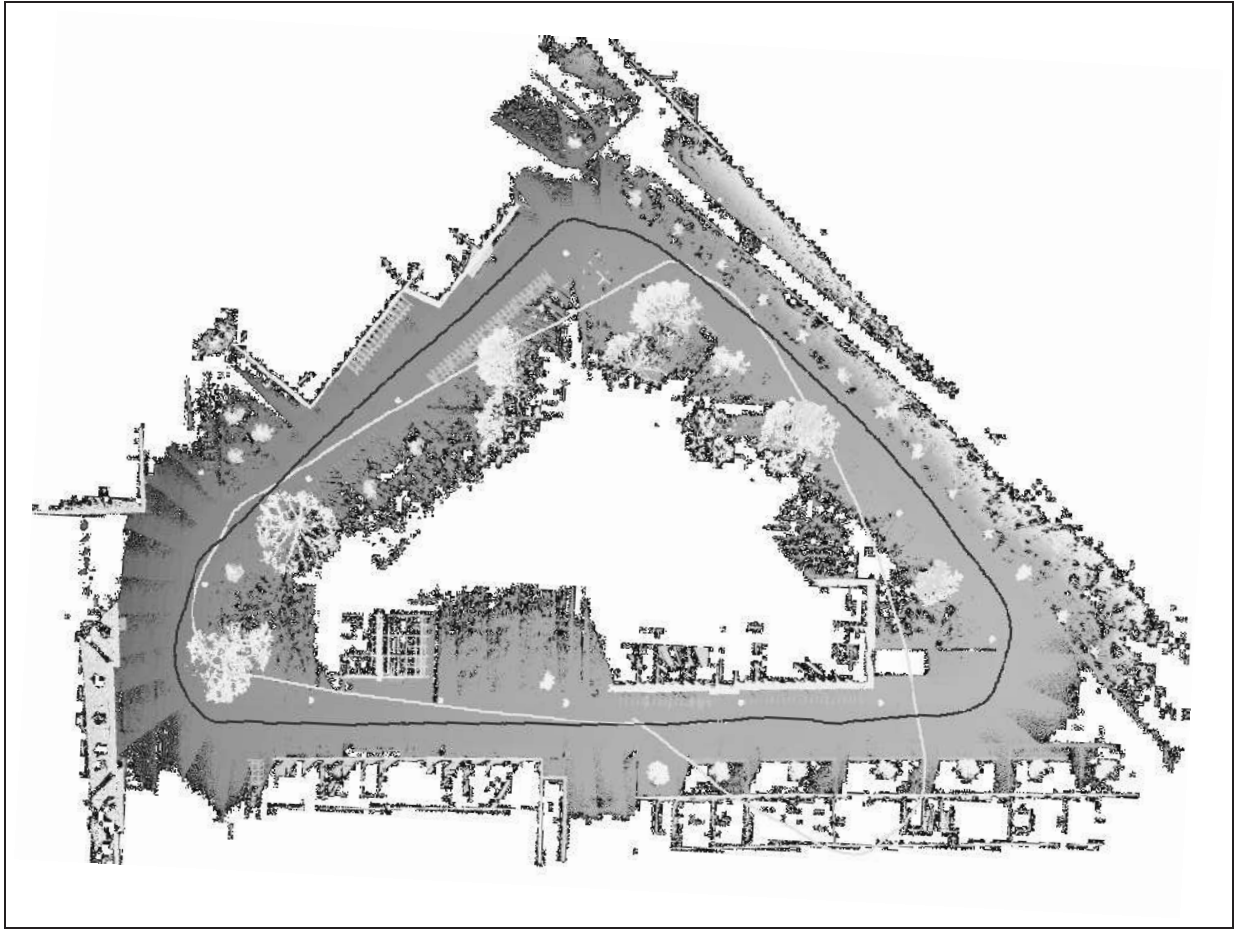


Figure 10: MLS map used for the localization experiments. The area represented by this map spans approximately 195 m by 146 m. The dark grey line shows the localized robot poses. The light grey line shows the pure odometry. The traversed trajectory has a length of 284 m.

References

- Adams, M., Zhang, S., and Xie, L. (2004). Particle filter based outdoor robot localization using natural features extracted from laser scanners. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.
- Agrawal, M. and Konolige, K. (2006). Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *International Conference on Pattern Recognition (ICPR)*.
- Bares, J., Hebert, M., Kanade, T., Krotkov, E., Mitchell, T., Simmons, R., and Whittaker, W. R. L. (1989). Amblor: An autonomous rover for planetary exploration. *IEEE Computer Society Press*, 22(6):18–22.
- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517.
- Davison, A. and Kita, N. (2001). 3D simultaneous localisation and map-building using active

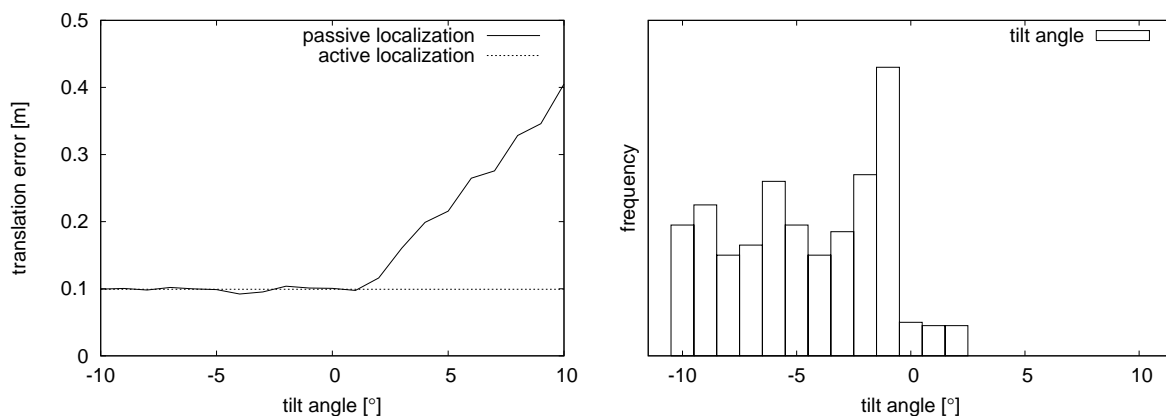


Figure 11: Localization error of the passive approach using different tilt angles and of the active approach, respectively (left). The right image depicts the frequency of the tilt angles chosen by the active approach.

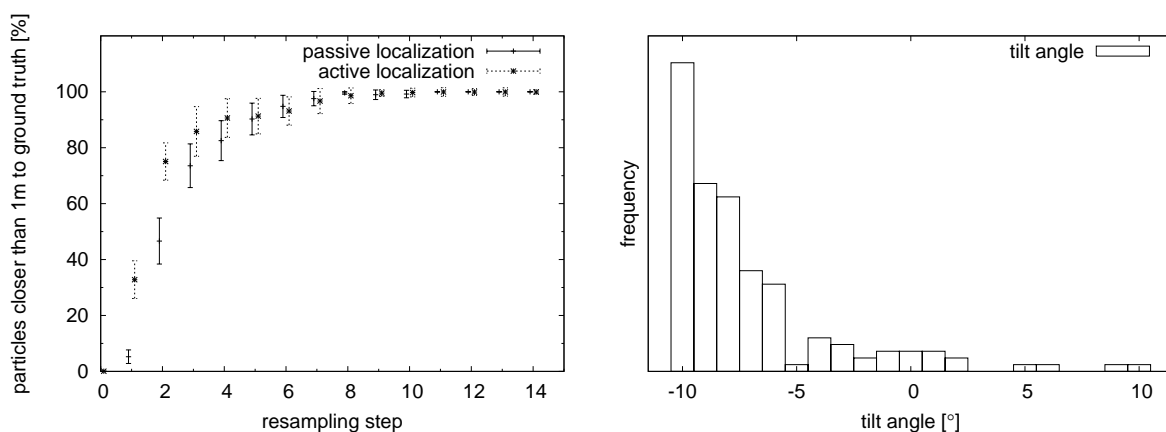


Figure 12: The left image depicts the convergence of the particles to the true position of the robot with 500,000 particles. The right image shows the frequency of the tilt angles chosen by the active approach during the global localization task.

vision for a robot moving on undulating terrain. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Kauai*. IEEE Computer Society Press.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte Carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.

Doucet, A., de Freitas, N., and Gordan, N., editors (2001). *Sequential Monte-Carlo Methods in Practice*. Springer Verlag.

Eliazar, A. and Parr, R. (2004). Learning probabilistic motion models for mobile robots. In *Proc. of the International Conference on Machine Learning (ICML)*.

Fox, D., Burgard, W., and Thrun, S. (1998). Active markov localization for mobile robots. *Journal of Robotics & Autonomous Systems*, 25:195–207.

Grisetti, G., Stachniss, C., and Burgard, W. (2005). 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.
- Hebert, M., Caillias, C., Krotkov, E., Kweon, I., and Kanade, T. (1989). Terrain mapping for a roving planetary explorer. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 997–1002.
- Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999). Exploring expression data: Identification and analysis of coexpressed genes. *Genome Res.*, 9(11):1106–1115.
- Jensfelt, P. and Kristensen, S. (2001). Active global localisation for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5):748–760.
- Kaelbling, L. P., Cassandra, A. R., and Kurien, J. A. (1996). Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Osaka, Japan.
- Koenig, S. and Simmons, R. (1998). Xavier: A robot navigation architecture based on partially observable markov decision process models. In Kortenkamp, D., Bonasso, R., and Murphy, R., editors, *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, pages 91–122. MIT Press.
- Lacroix, S., Mallet, A., Bonnafous, D., Bauzil, G., Fleury, S., Herrb, M., and Chatila, R. (2002). Autonomous rover navigation on unknown terrains: Functions and integration. *International Journal of Robotics Research*, 21(10-11):917–942.
- Lingemann, K., Surmann, H., Nüchter, A., and Hertzberg, J. (2005). High-speed laser localization for mobile robots. *Journal of Robotics & Autonomous Systems*, 51(4):275–296.
- Olson, C. (2000). Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66.
- Parra, C., Murrieta-Cid, R., Devy, M., and Briot, M. (1999). 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.
- Pfaff, P., Triebel, R., and Burgard, W. (2007). An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *International Journal of Robotics Research*, 26(2):217–230.
- Porta, J. M., Verbeek, J. J., and Kröse, B. J. A. (2005). Active appearance-based robot localization using stereo vision. *Autonomous Robots*, 18(1):59–80.
- Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363.
- Triebel, R., Pfaff, P., and Burgard, W. (2006). 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)*.