Autonomous Robots manuscript No.
(will be inserted by the editor)

Large Scale Graph-based SLAM Using Aerial Images as Prior Information

Rainer Kümmerle · Bastian Steder · Christian Dornhege · Alexander Kleiner · Giorgio Grisetti · Wolfram Burgard

Received: date / Accepted: date

Abstract The problem of learning a map with a mobile robot has been intensively studied in the past and is usually referred to as the simultaneous localization and mapping (SLAM) problem. However, most existing solutions to the SLAM problem learn the maps from scratch and have no means for incorporating prior information. In this paper, we present a novel SLAM approach that achieves global consistency by utilizing publicly accessible aerial photographs as prior information. It inserts correspondences found between stereo and three-dimensional range data and the aerial images as constraints into a graph-based formulation of the SLAM problem. We evaluate our algorithm based on large real-world datasets acquired even in mixed in- and outdoor environments by comparing the global accuracy with state-of-the-art SLAM approaches and GPS. The experimental results demonstrate that the maps acquired with our method show increased global consistency.

Keywords mapping, localization, aerial images

1 Introduction

The ability to acquire accurate models of the environment is widely regarded as one of the fundamental preconditions for truly autonomous robots. In the context of mobile robots, these models typically are maps of the environment that support different tasks including localization and path planning. The problem of estimating a map with a mobile robot navigating through and perceiving its environment has been studied intensively and is usually referred to as the simultaneous localization and mapping (SLAM) problem.

Originally, the SLAM problem has been formulated independently of any specific prior about the environment and most SLAM approaches seek to determine the most likely map and robot trajectory given a sequence of observations without taking into account special priors. However, priors can greatly improve solutions to the SLAM problem. Consider, for example, a scenario, in which a globally consistent map is required or in which the robot has to navigate to a target location specified in global terms such as given by a GPS coordinate. Corresponding applications include rescue or surveillance missions in which one requires specific areas to be covered. Unfortunately, GPS typically suffers from outages so that a robot only relying on GPS information might encounter substantial positioning errors. At the same time, even sophisticated SLAM algorithms cannot fully compensate for these errors as there still might be lacking constraints between certain observations combined with large odometry errors. However, even in situations with substantial overlap between consecutive observations, the matching processes might result in errors that linearly propagate over time and lead to substantial absolute errors. Consider, for example, a mobile robot mapping a linear structure (such as a corridor of a building or the passage between two parallel buildings). Typically, this corridor will be slightly curved in the resulting map. Whereas this is not critical in many applications as the computed maps are generally locally consistent [Howard, 2004], it might be sub-optimal in application scenarios in which global consistency is required, such as those discussed above.

In this paper, we present an approach that overcomes these problems by utilizing aerial photographs for calculating global constraints within a graph-representation of the SLAM problem. In our approach, these constraints are

All authors are with the
University of Freiburg, Dept. of Computer Science, Georges-Köhler-Allee 079, 79110 Freiburg, Germany
Tel.: +49-761-203-8006, Fax: +49-761-203-8007
E-mail: {kuemmerl,steder,dornhege,kleiner,grisetti,burgard}@informatik.uni-freiburg.de

This is a preprint of an article published in Journal of Autonomous Robots. The final publication is available at www.springerlink.com

(a) (b) (c)

Fig. 1: Motivating example comparing standard SLAM (a), localization using aerial imagery as prior information (b), and our combined approach (c). Note the misalignment relative to the outer wall of the building in (a). Whereas the localization applied in (b), which relies on aerial images, yields proper alignments, it cannot provide accurate estimates inside the building. Combining the information of both algorithms yields the best result (c).

obtained by matching features from the sensor data of the robot to the aerial image. In particular, we consider 3D point clouds obtained by a laser range finder and the data provided by a stereo camera.

Compared to traditional SLAM approaches, the use of a global prior enables our technique to provide more accurate solutions by limiting the error when visiting unknown regions. In contrast to approaches that seek to directly localize a robot in an outdoor environment, our approach is able to operate reliably even when the prior is not available, for example, because of the lack of appropriate matches. Therefore, it is suitable for mixed indoor/outdoor operation. Figure 1 shows a motivating example and compares the outcome of our approach with the ones obtained by applying a state-of-the-art SLAM algorithm and a pure localization method using aerial images.

The approach proposed in this paper relies on the so called graph formulation of the SLAM problem [Lu and Milios, 1997; Olson *et al.*, 2006]. In this variant of the SLAM problem, every node of the graph represents a robot pose and an observation taken at this pose. Edges in the graph represent relative transformations between nodes computed from overlapping observations. Additionally, our system computes its global position for every node employing a variant of Monte-Carlo localization (MCL) that uses 3D laser scans or stereo images as observations and aerial images as reference maps. The use of 3D information allows our system to determine the portions of the image and of the 3D scene that can be reliably matched. It computes these matches by detecting structures that potentially correspond to intensity variations in the aerial image. In the case of the stereo camera our approach extracts visual features that can be matched with the aerial image.

GPS is a popular device for obtaining position estimates. Whereas it has also been used to localize mobile vehicles operating outdoors, we found that the accuracy of this estimate is in general not sufficient to obtain precise maps, especially when the robot moves close to buildings or in narrow streets.

The approach proposed in this paper works as follows: we apply a variant of Monte Carlo localization [Dellaert *et al.*, 1998] to localize a robot by matching sensor data to aerial images of the environment. To achieve this, our approach selects the portions of the sensor information and of the image that can be reliably matched. These correspondences are added as constraints in a graph-based formulation of the SLAM problem. Note that our system preserves the flexibility of traditional SLAM approaches and can also be used in absence of any prior information. However, when the prior is available our system provides highly accurate solutions also in pathological datasets (i.e., when no loop closures take place). We validate the results with a large-scale dataset acquired in a mixed in- and outdoor environment. We furthermore compare our method with state-of-the-art SLAM approaches and with GPS.

This paper is an extended version of an already published previous work [Kümmerle *et al.*, 2009] in which we presented a sensor model for 3D point clouds that reflects three-dimensional structures visible in the aerial images. This paper additionally provides a sensor model for stereo vision systems that allows us to address distinct two-dimensional features as line markings or pathways.

This paper is organized as follows. After discussing related work, we will give an overview over our system followed by a detailed description of the individual components in Section 3. We then will present experiments designed to evaluate the quality of the resulting maps obtained with our algorithm in Section 4. We furthermore will com-

pare our approach with a state-of-the-art SLAM system that does not use any prior information.

2 Related Work

SLAM techniques for mobile robots can be classified according to the underlying estimation technique. The most popular approaches are extended Kalman filters (EKFs) [Leonard and Durrant-Whyte, 1991; Smith *et al.*, 1990], sparse extended information filters [Eustice *et al.*, 2005; Thrun *et al.*, 2004], particle filters [Montemerlo *et al.*, 2003], and least square error minimization approaches [Lu and Milios, 1997; Frese *et al.*, 2005; Gutmann and Konolige, 1999]. The effectiveness of the EKF approaches comes from the fact that they estimate a fully correlated posterior about landmark maps and robot poses [Leonard and Durrant-Whyte, 1991; Smith *et al.*, 1990]. Their weakness lies in the strong assumptions that have to be made upon both, the robot motion model and the sensor noise. If these assumptions are violated, the filter is likely to diverge [Julier *et al.*, 1995; Uhlmann, 1995].

An alternative approach is to find maximum likelihood maps by the application of least square error minimization. The idea underlying these methods is to compute a network of relations given the sequence of sensor readings. These relations represent the spatial constraints between the poses of the robot. In this paper, we also follow this approach. Lu and Milios [1997] first applied this technique in robotics to address the SLAM problem by optimizing the whole network at once. Gutmann and Konolige [1999] propose an effective way for constructing such a network and for detecting loop closures while running an incremental estimation algorithm.

All the SLAM methods discussed above do not take into account any prior knowledge about the environment. On the other hand, several authors addressed the problem of utilizing prior knowledge to localize a robot outdoors. For example, Korah and Rasmussen [2004] use image processing techniques to extract roads on aerial images. This information is then applied to improve the quality of GPS paths using a particle filter by calculating the particle weight according to its position relative to the streets. Leung *et al.* [2008] present a particle filter system performing localization on aerial photographs by matching images taken from the ground with a monocular vision system. The approach detects line features to find correspondences between the aerial and ground images. Whereas it applies a Canny edge detector and progressive probabilistic Hough transform to find lines in aerial images, it performs a vanishing point analysis for estimating building wall orientations in the monocular vision data. The approach achieves an average positioning accuracy of several meters. Ding *et al.* [2008] use a vanishing point analysis to extract 2D corners from aerial images and inertial tracking data. They also extract 2D corners from LiDAR generated depth maps and apply a multi-stage process to match these corners with those from the aerial image. The corresponding matches finally yield a fine estimation of the camera pose that is used to texture the LiDAR models with the aerial images. Chen and Wang [2007] use an energy minimization technique to merge prior information from aerial images and mapping. They perform mapping by constructing sub-maps consisting of 3D point clouds, that are constrained by relations. Using a Canny edge detector, they compute a vector field from the image that models force towards the detected edges. The sum of the forces applied to each point corresponds to the energy measure in the minimization process, when placing a sub-map into the vector field of the image. Parsley and Julier [2009] demonstrate how to incorporate a heterogeneous prior map into an extended Kalman filter for SLAM. They show that such a prior bounds the error while the robot travels in open-loop. Lee *et al.* [2007] use the road graph from a given prior map for SLAM. Under the assumption that the vehicle follows only roads they can constrain the probabilistic model to the roads and thus achieve higher accuracy than traditional FastSLAM. Dogruer *et al.* [2007] utilized soft computing techniques for segmenting aerial images into different regions, such as buildings, roads, and forests. They applied MCL on the segmented maps. However, compared to the approach presented in this paper, their technique strongly depends on the color distribution of the aerial images since different objects on these images might share similar color characteristics.

Früh and Zakhor [2004] described the generation of edge images from aerial photographs for 2D laser-based localization. As they state in their paper, localization errors might occur if rooftops seen on the aerial image significantly differ from the building footprint observed by the 2D scanner. The method proposed in this paper computes a 2D structure from a 3D observation, which is more likely to match with the features extracted from the aerial image. This leads to an improved robustness in finding location correspondences. Additionally, our system is not limited to operate in areas where the prior is available. When no prior is available, our algorithm operates without relevant performance loss compared to standard SLAM approaches which do not utilize any prior. Our system furthermore allows a robot to operate in mixed indoor/outdoor scenarios.

Sofman *et al.* [2006] introduced an online learning system predicting terrain travel costs for unmanned ground vehicles (UGVs) on a large scale. They extract features from locally observed 3D point clouds and generalize them on overhead data such as aerial photographs, allowing the UGVs to navigate on less obstructed paths. Montemerlo and Thrun [2004] presented an approach similar to the one presented in this paper. The major difference to our technique is that they use GPS to obtain the prior. Due to the increased

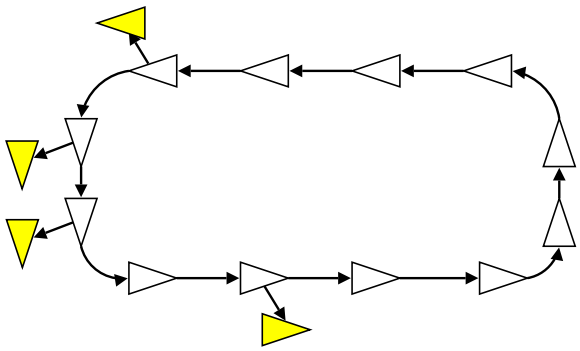


Fig. 2: The graph representation of our method. In contrast to the standard approach, we additionally integrate global constraints (shown in yellow / light gray) given by the prior information.

noise which affects the GPS measurements this prior can lead to larger estimation errors in the resulting maps

3 Graph-SLAM with Prior Information from Aerial Images

Our system relies on a graph-based formulation of the SLAM problem. It operates on a sequence of 3D scans and odometry measurements. Every node of the graph represents a position of the robot at which a sensor measurement was acquired. Every edge stands for a constraint between the two poses of the robot. In addition to direct links between consecutive poses, it can integrate prior information (when available) which in our case is given in form of an aerial image.

This prior information is introduced to the graph-based SLAM framework as global constraints on the nodes of the graph, as shown in Figure 2. These global constraints are absolute locations obtained by MCL [Dellaert *et al.*, 1998] on a map computed from the aerial images. As these images are captured from a viewpoint significantly different from the one of the robot, we extract corresponding 2D features from the 3D measurements obtained from a laser scanner or a stereo camera which is more likely to be consistent with the one visible in the image. In this way, we can prevent the system from introducing inconsistent prior information. To integrate the observations over time, we apply a probabilistic localization approach realized by a particle filter.

In the following we explain how we adapted MCL to operate on aerial images and how to select the points in the 3D measurements to be considered in the observation model. After describing how to utilize the data of a 3D range finder, we present a sensor model which uses a stereo camera to localize the vehicle. Subsequently we describe our graph-based SLAM framework.

3.1 Monte Carlo Localization

To estimate the pose $\mathbf{x}$ 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 location $\mathbf{x}_t$ of the robot at time t given all observations $\mathbf{z}_{1:t}$ and all control inputs $\mathbf{u}_{0:t-1}$. This posterior is updated as follows:

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

Here, α is a normalization constant which ensures 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* $p(\mathbf{z}_t \mid \mathbf{x}_t)$.

For the implementation of the described filtering scheme, we use a sample-based approach which is commonly known as *Monte Carlo localization* (MCL) [Dellaert *et al.*, 1998]. MCL 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 according to 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 re-sampling step can remove good samples from the filter which can lead to particle impoverishment. To decide when to perform the re-sampling step, we calculate the number N_{eff} of effective particles according to the formula proposed in [Doucet *et al.*, 2001]

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

where $\widetilde{w^{[i]}}$ refers to the normalized weight of sample i and we only re-sample 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 SLAM [Grisetti *et al.*, 2005]. To initialize the particle filter we draw the particle positions according to a Gaussian distribution, whose mean corresponds to the current GPS estimate. In our current implementation, we use 1,000 particles.

Thus far, we described the general framework of MCL. One contribution of this paper are two different sensor models for determining the likelihood $p(\mathbf{z} \mid \mathbf{x})$ of a measurement $\mathbf{z}$ given a position $\mathbf{x}$ within an aerial image. Whereas the first one, described in the following section, operates on 3D data obtained with a sweeping laser scanner, the second one, described in Section 3.3, is designed for 3D data obtained from a stereo camera system.

3.2 Sensor Model for 3D Range Scans in Aerial Images

The task of the sensor model is to determine the likelihood $p(\mathbf{z} \mid \mathbf{x})$ of a 3D range scan $\mathbf{z}$ given the robot is at pose $\mathbf{x}$. In our current system, we apply the so called endpoint model or likelihood fields [Thrun *et al.*, 2005]. Let z_k be the k-th measurement of a 3D scan $\mathbf{z}$. The endpoint model computes the likelihood of z_k based on the distance between the scan point z'_k corresponding to z_k re-projected onto the map according to the pose $\mathbf{x}$ of the robot and the point in the map d'_k which is closest to z'_k as:

$$p(\mathbf{z} \mid \mathbf{x}) = f(\|z'_1 - d'_1\|, \ldots, \|z'_k - d'_k\|). \tag{3}$$

Under the assumption that the beams are independent and the sensor noise is normally distributed we can rewrite (3) as

$$f(\|z'_1 - d'_1\|, \ldots, \|z'_k - d'_k\|) \propto \prod_j e^{\frac{(z'_j - d'_j)^2}{\sigma^2}}. \tag{4}$$

Since the aerial image only contains 2D information about the scene, we need to select a set of beams from the 3D scan, which are likely to result in structures, that can be identified and matched in the image. In other words, we need to transform both, the scan and the image to a set of 2D points which can be compared via the function $f(\cdot)$.

To extract candidate points from the aerial image we employ the standard Canny edge extraction procedure [Canny, 1986]. The idea behind this is that a height gap in the world corresponds to a change in intensity in the aerial image that can be detected by the edge extraction procedure. In an urban environment, such edges are typically generated by borders of roofs, trees, or fences. Of course, the edge extraction procedure returns a lot of false positives that do not represent any actual 3D structure, like street markings, grass borders, shadows, and other flat markings. All these aspects have to be considered by the sensor model. Figure 4 shows an aerial image and the extracted Canny image along with the likelihood-field.

To transform the 3D scan into a set of 2D points that can be compared to the Canny image, we select a subset of points from the 3D scan and consider their 2D projection in the ground plane. This subset should contain all the points which may be visible in the reference map. To perform this operation we compute the z-buffer [Foley *et al.*, 1993] of a scan from a bird's eye perspective. In this way we discard those points which are occluded in the bird's eye view from the 3D scan. By simulating this view, we handle situations like overhanging roofs, where the house wall is occluded and therefore is not visible in the aerial image in a more sophisticated way.

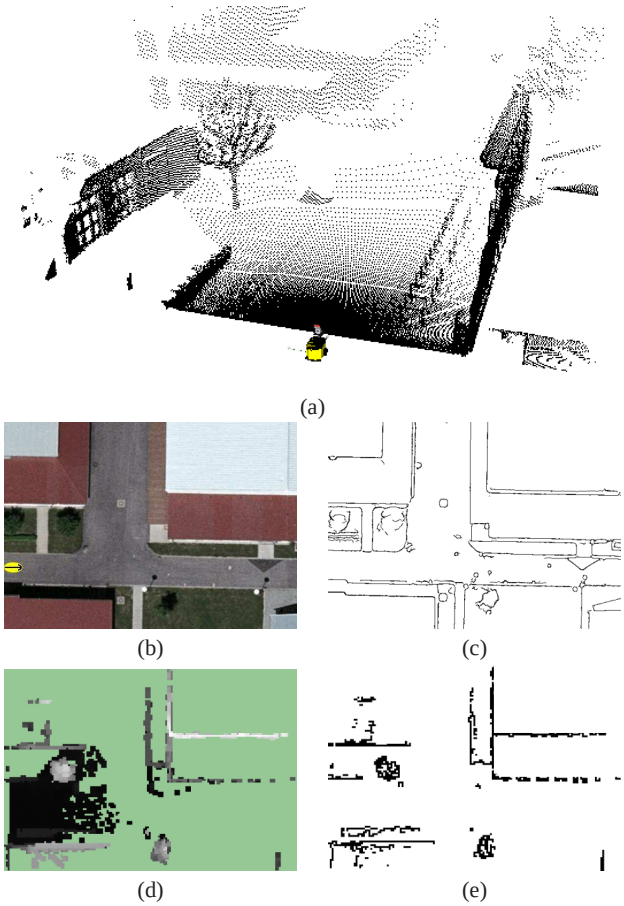
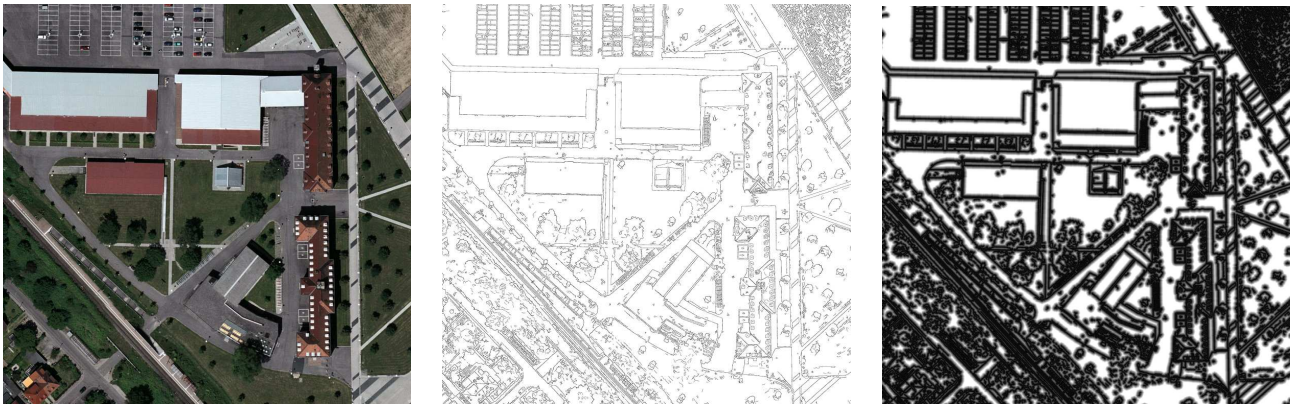


Fig. 3: A 3D scan represented as a point cloud **(a)**, the aerial image of the corresponding area **(b)**, the Canny edges extracted from the aerial image **(c)**, the 3D scene from (a) seen from the top **(d)** (gray values represent the maximal height per cell, the darker a pixel, the lower the height, and the green/bright gray area was not visible in the 3D scan), and positions extracted from (d), where a high variation in height occurred **(e)**.

The regions of the z-buffer that are likely to be visible in the Canny image are the ones that correspond to relevant depth changes. We construct a 2D scan by considering the 2D projection of the points in these regions. This procedure is illustrated by the sequence of images in Figure 3.

An implementation purely based on a 2D scanner (like the approach proposed by Früh and Zakhor [2004]) would not account for occlusions due to overhanging objects. An additional situation, in which our approach is more robust,



(a) (b) (c)

Fig. 4: Google Earth image of the Freiburg campus (a), the corresponding Canny image (b), and the corresponding likelihood field computed from the Canny image (c). Note that the structure of the buildings and the vertical elements is clearly visible despite of the considerable clutter.

is in the presence of trees. In this case a 2D view would only sense the trunk, whereas the whole crown is visible in the aerial image.

In our experiments, we considered variations in height Δ_h of 0.5 m and above as possible positions of edges that could also be visible in the aerial image. We then match the positions of these variations relative to the robot against the Canny edges of the aerial image in a point-by-point fashion and in a similar way like matching of 2D-laser scans against an occupancy grid map.

This sensor model has some limitations. It is susceptible to visually cluttered areas, since it then can find random correspondences in the Canny edges. There is also the possibility of systematic errors, when a wrong line is used for the localization, e.g., in the case of shadows. In our practical experiments we could not find evidence that this leads to substantial errors when one applies position tracking and as long as the robot does not move through such areas for a longer period of time. The main advantages of the endpoint model in this context are that it ignores possible correspondences outside of a certain range and implicitly deals with edge points that do not correspond to any 3D structure.

Our method, of course, also depends on the quality of the aerial images. Perspective distortions in the images can easily introduce errors. However, for the data sets used to carry out our experiments we could not find evidence that this is a major complicating factor.

Finally, we employ a heuristic to detect when the prior is not available, i.e., when the robot is inside of a building or under overhanging structures. This heuristic is based on the 3D perception. If there are range measurements whose endpoints are above the robot, we do not integrate any global constraints from the position estimate, since we assume that the area the robot is sensing is not visible in the aerial image. While a more profound solution regarding place recognition is clearly possible, this conservative heuristic turned out to yield sufficiently accurate results.

3.3 Sensor Model for Stereo Images in Aerial Images

After having described a sensor model for a robot equipped with a 3D laser scanner, we will now focus on using a stereo camera to extract the relevant sensor information for localizing the robot. In addition to the 3D data extracted from the stereo images using the procedure described in the previous section, we utilize the color information to enable the robot to take advantage of flat structures, such as street markings or borders of different ground surfaces that cannot be detected by a range-only device at all or without further post-processing (e.g. curb detection).

To extract the visual information, we proceed as follows. First, we process the stereo image to obtain 3D information. Second, we apply the Canny edge detector to the camera image. This is motivated by the fact that the same edges that are visible in the aerial image might also be visible in the robot's camera image. Since the aerial image is an orthogonal view, we discard features that are not obtained from the ground plane. In the last step we project the 3D points on the ground plane to obtain a 2D set of points which is finally applied in Eqn. (3) and processed in a similar way as the 3D laser range measurements.

The aerial image and the camera images of the robot differ substantially regarding viewpoint and resolution. This can lead to situations, in which the robot detects structures on the ground that are not visible in the aerial image. Consider, for example, Figure 5a which shows a stone pattern in the on-board camera image that typically leads to fine lines

in the Canny image (Fig. 5d). Typically, such fine structures are not visible in the aerial image due to the much lower resolution and might disturb the matching process. To reject these fine structures, one can increase the acceptance threshold of the edge extraction. Figure 5e shows the outcome of the edge extraction with a more selective threshold that is increased to the smallest value that does not result in false positives from the fine structures any more. However, increasing the threshold removes also true positives from the detected edges, i.e., lines that are also visible in the aerial image.

Ideally, one would like to remove the false positives resulting from fine structures near the robot, while keeping true positives that are farther away. This is not possible by adjusting the threshold alone. By exploiting the 3D information provided by our sensors, we relate the distance between the image pixels and the camera with the structure size. The idea is that fine edges that are far away result from large structures visible in the aerial image, but fine edges near the robot represent small structures. Given the distance, we can adapt the level of blur in different regions of the image. Regions closer to the robot will become more blurred than regions farther away. In particular, we process each camera image by adding a distance-dependent blur to each pixel. The size of the kernel k applied to the pixel corresponding to a 3D point having local coordinates (x,y,z) is inversely proportional to its distance in the x-y plane:

$$k = \frac{\alpha}{||(x,y)||}, \tag{5}$$

where α is a scaling factor that depends on the tilt angle of the camera. This dynamic blur can be implemented efficiently using box filters in combination with integral images [Bay *et al.*, 2006]. In this way, we take the low resolution of the aerial image into account. As a consequence, the robot will reduce fine structures that are with high probability not visible in the aerial image. As an example for the application of this distance-dependent blur, consider Figure 5. Note that a 3D scanner together with a calibrated mono camera would lead to similar results.

In our system we employ a *Point Grey Bumblebee2* stereo camera. We use the software library provided by Point Grey to extract the 3D points from the stereo pair as a black box. This library provides us with the 3D position of each pixel in the image. Since the scaling factor α depends on the known geometry of the robot, there are no additional parameters.

3.4 Graph-based Maximum Likelihood SLAM

This section describes the basic algorithm for obtaining the maximum likelihood trajectory of the robot. We apply a graph-based SLAM technique to estimate the most-likely trajectory, i.e., we seek for the maximum-likelihood (ML) configuration like the majority of approaches to graph-based SLAM. The goal of such mapping algorithms is to find the configuration of the nodes that maximizes the likelihood of the observations. Let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ be a vector of parameters, where $\mathbf{x}_i$ describes the pose of node *i*. Let $\mathbf{z}_{ij}$ and Ω_{ij} be respectively the mean and the information matrix of an observation of node j seen from node i, perturbed by Gaussian noise. Let $\mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})$ be a function that computes a difference between the expected observation of the node $\mathbf{x}_j$ seen from the node $\mathbf{x}_i$ and the observation $\mathbf{z}_{ij}$ gathered by the robot. For simplicity of notation, in the rest of the paper we will encode the measurement in the indices of the error function:

$$\mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}) \stackrel{\text{def.}}{=} \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \stackrel{\text{def.}}{=} \mathbf{e}_{ij}(\mathbf{x}). \tag{6}$$

Let $\mathscr{C}$ be the set of pairs of indices for which a constraint (observation) $\mathbf{z}$ exists. The goal of a maximum likelihood approach is to find the configuration of the nodes $\mathbf{x}^*$ that minimizes the negative log likelihood $\mathbf{F}(\mathbf{x})$ of all the observations

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle \in \mathscr{C}} \underbrace{\mathbf{e}_{ij}(\mathbf{x})^T \Omega_{ij} \mathbf{e}_{ij}(\mathbf{x})}_{\mathbf{F}_{ij}} \tag{7}$$

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}} \mathbf{F}(\mathbf{x}). \tag{8}$$

To account for the residual error in each constraint, we can additionally consider the prior information by incorporating the position estimates of our localization approach. To this end, we extend Eqn. (7) as follows:

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i,j \rangle} \mathbf{e}_{ij}(\mathbf{x})^T \Omega_{ij} \mathbf{e}_{ij}(\mathbf{x}) + \sum_i \underbrace{\mathbf{e}(\mathbf{x}_i, \hat{\mathbf{x}}_i)^T \Omega_i \mathbf{e}(\mathbf{x}_i, \hat{\mathbf{x}}_i)}_{\mathbf{F}_i(\mathbf{x})}, \tag{9}$$

where $\hat{\mathbf{x}}_i$ denotes the position as it is estimated by the localization using the bird's eye image and Ω_i is the information matrix of this constraint. In our approach, we compute Ω_i based on the distribution of the samples in MCL. For simplicity of notation in the remainder of this section we will define

$$\mathbf{e}(\mathbf{x}_i, \hat{\mathbf{x}}_i) \stackrel{\text{def.}}{=} \mathbf{e}_i(\mathbf{x}_i) \stackrel{\text{def.}}{=} \mathbf{e}_i(\mathbf{x}) \tag{10}$$

since $\hat{\mathbf{x}}_i$ can be seen as a measurement to the extent of the optimization process, and thus embedded in the indices of the error function.

If a good initial guess $\check{\mathbf{x}}$ of the robot's poses is known, the numerical solution of (8) can be obtained by using the popular Gauss-Newton or Levenberg-Marquardt algorithms [Press *et al.*, 1992, §15.5]. In our case the initial guess is obtained by the odometry of the robot. The idea is to approximate the error function by its first order Taylor expansion around the current initial guess $\check{\mathbf{x}}$

$$\mathbf{e}_{ij}(\check{\mathbf{x}}_i + \Delta\mathbf{x}_i, \check{\mathbf{x}}_j + \Delta\mathbf{x}_j) = \mathbf{e}_{ij}(\check{\mathbf{x}} + \Delta\mathbf{x}) \simeq \mathbf{e}_{ij} + \mathbf{J}_{ij}\Delta\mathbf{x}. \tag{11}$$

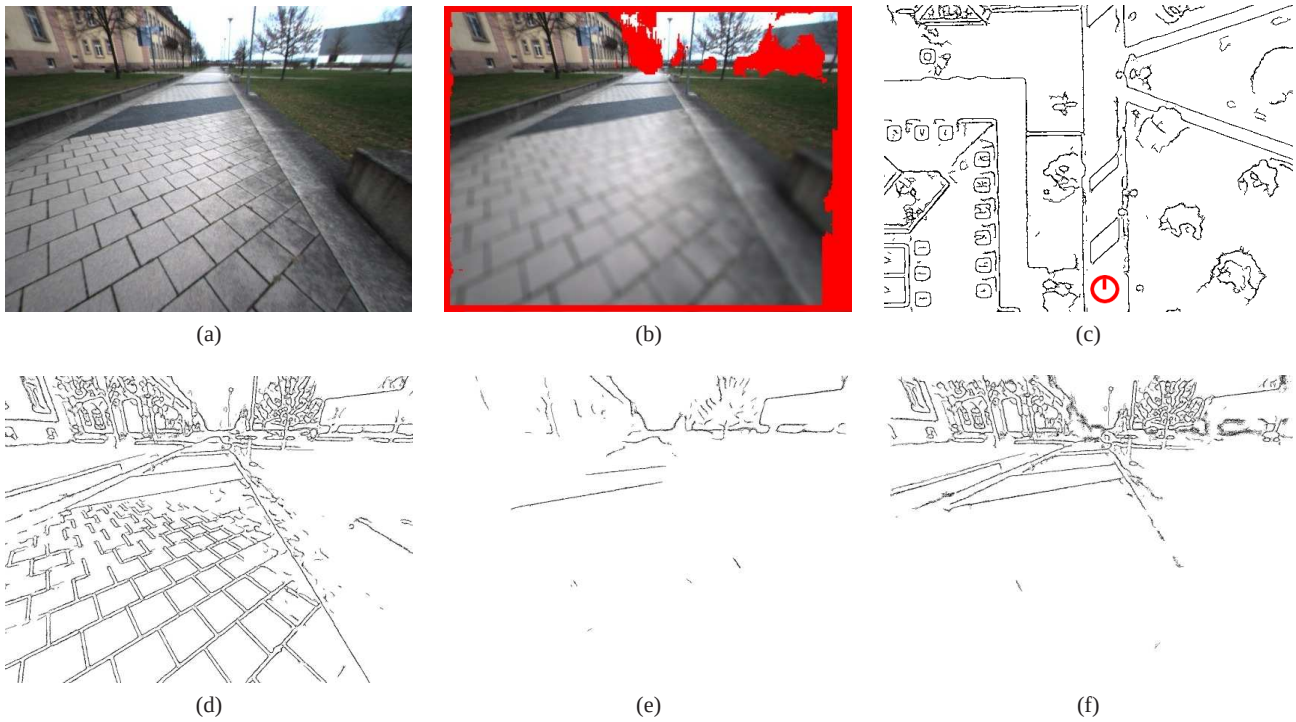


Fig. 5: This figure shows an example for the distance-dependent blurring of the images as a preprocessing for the Canny edge detection. **(a)**: The original camera image. **(b)**: The blurred image, where the strength of the blur is inversely proportional to the 3D distance of the pixel. Red (dark gray) marks areas of the image, where no 3D information is available. **(c)**: The corresponding part in the Canny aerial image with the robot position marked in red (dark gray). **(d)**: Standard Canny on the original camera image using low thresholds. **(e)**: Standard Canny on the original camera image using thresholds that are high enough so that the pattern of the ground directly in front of the robot is not recognized as edges. **(f)**: Standard Canny on the dynamically blurred image using the same thresholds as in (d). Here, most of the important edges, i.e., those on the ground that are also visible in the aerial image were extracted correctly. Yet, the ground pattern in front of the robot was not extracted.

Here $\mathbf{J}_{ij}$ is the Jacobian of $\mathbf{e}_{ij}(\mathbf{x})$ computed for $\check{\mathbf{x}}$ and $\mathbf{e}_{ij} \stackrel{\text{def.}}{=} \mathbf{e}_{ij}(\check{\mathbf{x}})$. Substituting (11) in the error terms $\mathbf{F}_{ij}$ of (7), we obtain

$$
\begin{aligned}
&\mathbf{F}_{ij}(\check{\mathbf{x}}+\Delta\mathbf{x}) \\
&= \mathbf{e}_{ij}(\check{\mathbf{x}}+\Delta\mathbf{x})^T \Omega_{ij} \mathbf{e}_{ij}(\check{\mathbf{x}}+\Delta\mathbf{x}) && (12)\\
&\simeq (\mathbf{e}_{ij}+\mathbf{J}_{ij}\Delta\mathbf{x})^T \Omega_{ij} (\mathbf{e}_{ij}+\mathbf{J}_{ij}\Delta\mathbf{x}) && (13)\\
&= \underbrace{\mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}}_{\mathrm{c}_{ij}} + 2\underbrace{\mathbf{e}_{ij}^T \Omega_{ij} \mathbf{J}_{ij}}_{\mathbf{b}_{ij}^T}\Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\mathbf{J}_{ij}^T \Omega_{ij} \mathbf{J}_{ij}}_{\mathbf{H}_{ij}}\Delta\mathbf{x} && (14)\\
&= \mathrm{c}_{ij} + 2\mathbf{b}_{ij}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_{ij}\Delta\mathbf{x} && (15)
\end{aligned}
$$

In a similar way, we can approximate the functions $\mathbf{F}_i(\mathbf{x})$ as follows

$$
\begin{aligned}
\mathbf{F}_i(\check{\mathbf{x}}+\Delta\mathbf{x}) &= \mathbf{e}_i(\check{\mathbf{x}}+\Delta\mathbf{x})^T \Omega_i \mathbf{e}_i(\check{\mathbf{x}}+\Delta\mathbf{x}) && (16)\\
&\simeq (\mathbf{e}_i+\mathbf{J}_i\Delta\mathbf{x})^T \Omega_i (\mathbf{e}_i+\mathbf{J}_i\Delta\mathbf{x}) && (17)\\
&= \underbrace{\mathbf{e}_i^T \Omega_i \mathbf{e}_i}_{\mathrm{c}_i} + 2\underbrace{\mathbf{e}_i^T \Omega_i \mathbf{J}_i}_{\mathbf{b}_i^T}\Delta\mathbf{x} + \Delta\mathbf{x}^T \underbrace{\mathbf{J}_i^T \Omega_i \mathbf{J}_i}_{\mathbf{H}_i}\Delta\mathbf{x} && (18)\\
&= \mathrm{c}_i + 2\mathbf{b}_i^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_i\Delta\mathbf{x}, && (19)
\end{aligned}
$$

where $\mathbf{J}_i$ is the Jacobian of $\mathbf{e}_i(\mathbf{x})$ computed for $\check{\mathbf{x}}$ and $\mathbf{e}_i \stackrel{\text{def.}}{=} \mathbf{e}_i(\check{\mathbf{x}})$. With this local approximation, we can rewrite the function $\mathbf{F}(\mathbf{x})$ given in (9) as

$$
\begin{aligned}
\mathbf{F}(\check{\mathbf{x}}+\Delta\mathbf{x}) &= \sum_{\langle i,j\rangle\in\mathscr{C}} \mathbf{F}_{ij}(\check{\mathbf{x}}+\Delta\mathbf{x}) + \sum_{i\in\mathscr{G}} \mathbf{F}_i(\check{\mathbf{x}}+\Delta\mathbf{x}) && (20)\\
&\simeq \sum_{\langle i,j\rangle\in\mathscr{C}} \mathrm{c}_{ij} + 2\mathbf{b}_{ij}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_{ij}\Delta\mathbf{x} \\
&\qquad + \sum_{i\in\mathscr{G}} \mathrm{c}_i + 2\mathbf{b}_i^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}_i\Delta\mathbf{x} && (21)\\
&= \mathrm{c} + 2\mathbf{b}^T\Delta\mathbf{x} + \Delta\mathbf{x}^T\mathbf{H}\Delta\mathbf{x}. && (22)
\end{aligned}
$$

The quadratic form in (22) is obtained from (21) by setting

$$\mathrm{c} = \sum \mathrm{c}_{ij} + \sum \mathrm{c}_i \tag{23}$$

$$\mathbf{b} = \sum \mathbf{b}_{ij} + \sum \mathbf{b}_i \tag{24}$$

$$\mathbf{H} = \sum \mathbf{H}_{ij} + \sum \mathbf{H}_i. \tag{25}$$

It can be minimized in $\Delta\mathbf{x}$ by solving the linear system

$$\mathbf{H}\Delta\mathbf{x}^* = -\mathbf{b}. \tag{26}$$

The matrix $\mathbf{H}$ is the information matrix of the system and is sparse by construction, due to the sparsity of the Jacobians. Its number of non-zero blocks is twice the number of unique pairwise constrains plus the number of nodes. This allows for solving (26) by sparse Cholesky factorization. An efficient implementation of sparse Cholesky factorization can be found in the library CSparse [Davis, 2006].

The linearized solution is then obtained by adding to the initial guess the computed increments

$$\mathbf{x}^* = \breve{\mathbf{x}} + \Delta\mathbf{x}^*. \tag{27}$$

The popular Gauss-Newton algorithm iterates the linearization in (22), the solution in (26) and the update step in (27). In every iteration, the previous solution is used as the linearization point and the initial guess. The procedure described above is a general approach to multivariate function minimization, here derived for the special case of the SLAM problem.

The result of the optimization is a set of poses that maximizes the likelihood of all the individual observations. Furthermore, the optimization also accommodates the prior information about the environment to be mapped whenever such information is available. In particular, the objective function encodes the available pose estimates as given by our MCL algorithm described in the previous section. Intuitively the optimization deforms the solution obtained by the relative constraints path to maximize the overall likelihood of all the observations, including the priors. The optimization results in a consistent estimate, as long as the MCL gives the correct position of the vehicle. Note that including the prior information about the environment yields a globally consistent estimate of the trajectory even in situations where no loop closures occur.

4 Experiments

The approach described above has been implemented and evaluated on real data acquired with a *MobileRobots Powerbot* with a *SICK LMS* laser range finder mounted on an *Amtec* wrist unit. The 3D data used for the localization algorithm has been acquired by continuously tilting the laser up and down while the robot moves. The maximum translational velocity of the robot during data acquisition was 0.35 m/s. This relatively low speed allows our robot to obtain 3D data that is sufficiently dense to perform scan matching without the need to acquire the scans in a stop-and-go fashion. During each 3D scan the robot moved up to 2 m. We used the odometry to account for the distortion caused by the movement of the platform. Additionally, we utilize a *Point Grey Bumblebee2* stereo camera to acquire the vision data. Figure 6 depicts the setup of our robot. Although the robot is equipped with an array of sensors, in the experiments we only used the devices mentioned above. Table 1 summarizes the parameters applied in all our experiments.

Fig. 6: The robot used for carrying out the experiments is equipped with a laser range finder mounted on a pan/tilt unit. We obtain 3D data by continuously tilting the laser while the robot moves.

# of particles	1,000
minimum distance between the update steps of the particle filter	2.0 m
grid resolution	0.15 m
standard deviation σ of (4)	2.0 m
height variation threshold Δ_h in §3.2	0.5 m

Table 1: Summary of the parameters applied in our experiments.

4.1 Comparison to GPS

This first experiment aims to show the effectiveness of the localization on aerial images compared with the one achievable with GPS. We manually steered our robot along a 890 m long trajectory through our campus, entering and leaving buildings. The robot captured 445 3D scans that were utilized for localization. We also recorded the GPS data for comparison purposes. The data acquisition took approximately one hour.

Figure 8 compares the GPS estimate with the one obtained by MCL on the aerial view. The higher error of the GPS-based approach is clearly visible. Note that GPS, in contrast to our approach, does not explicitly provide the orientation of the robot.

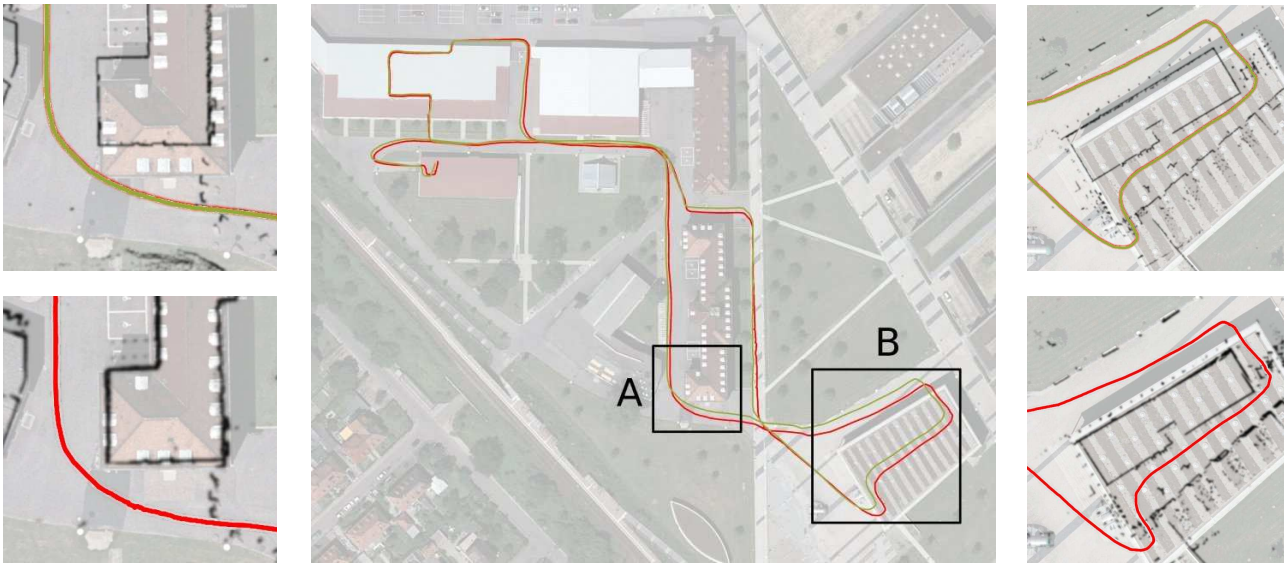


Fig. 7: Comparison of our system to a standard SLAM approach in a complex indoor/outdoor scenario. The center image shows the trajectory estimated by the SLAM approach (bright/yellow) and the trajectory generated by our approach (dark/red) overlaid on the Google Earth image used as prior information. On the left and right side, detailed views of the areas marked in the center image are shown, each including the trajectory and map. The upper images show the results of the standard SLAM approach; detail A on the left and B on the right. The lower images show the results of our system (A on the left side and B on the right). It is clearly visible, that, in contrast to the SLAM algorithm without prior information, the map generated by our approach is accurately aligned with the aerial image.

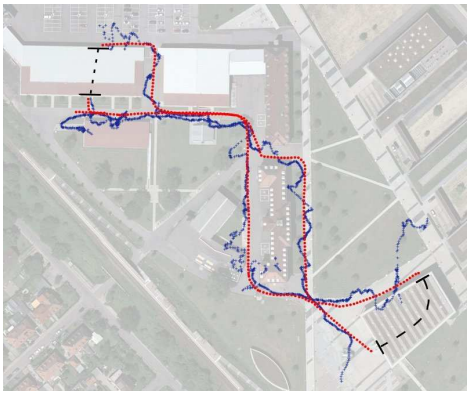


Fig. 8: Comparison between GPS measurements (blue crosses) and global poses from the localization in the aerial image (red circles). Dashed lines indicate transitions through buildings, where GPS and aerial images are unavailable.

4.2 Comparison of 3D Laser and Stereo Camera

The proposed localization based on 3D laser data relies on the extraction of height variations that are matched with the aerial image. In contrast, we can match the visual data provided by the stereo camera to visual features obtained from the ground plane with the aerial image. Features as, e.g., curbs cannot be detected by a 3D range sensor using height variations and flat features as road markings are not seen at all. This experiment is designed to evaluate the performance of using just the data provided by a stereo camera for localizing the robot.

To compare the two proposed sensor models we steered the robot along a 680 m long trajectory on our campus. While driving, the robot again collected 3D scans like in the experiment describe above. Additionally, the robot recorded stereo vision data. The stereo camera is mounted approximately 1.2 m above the ground and tilted downwards by 30 degrees. This setup allows the robot to observe the ground surface. Using this data we analyzed the position estimate of MCL using the two different sensor models described in Sections 3.2 and 3.3. Note that we set the update rate of the two approaches to the same frequency. Therefore, both approaches integrate the same number of sensor readings, i.e., we discard stereo images which are available at higher rates than the 3D laser scans generated by our platform. Figure 9 shows the trajectory estimate of the two approaches. As can be seen from the image, the estimate using vision is more accurate in this case. Here, the robot localizes itself on the foot path going through the vegetated area whereas the estimate using only 3D laser data is off the foot path due to the lack of a sufficiently dense 3D structure in this area. In the other parts of the trajectory, the estimate of the two ap-

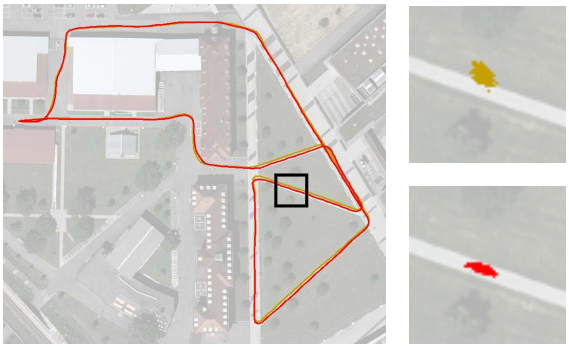


Fig. 9: Comparison between MCL using 3D laser scans and stereo vision data. The trajectory as it is estimated based on the 3D laser and vision data is shown in yellow / light gray and red / dark gray, respectively. The trajectory estimate using vision localizes the robot on the foot path whereas the laser based localization is slightly off. The right column shows a magnified view of the black rectangle and shows the particle cloud for MCL using 3D laser scans (top) and stereo data (bottom).

proaches overlay with each other, i.e., we could not observe a substantial difference in the position estimate.

4.3 Global Map Consistency

The goal of this set of experiments is to evaluate the ability of our system to create a consistent map of a large mixed in- and outdoor environment and to compare it against a state-of-the-art SLAM approach similar to the one proposed by Olson [2008]. Whereas the constraints between the nodes are generated as suggested by Olson, Eqn. (7) is optimized using TORO [Grisetti *et al.*, 2009]. For evaluating the global map consistency we recorded data in two different environments, our campus and a residential area. These two areas differ substantially. Whereas the campus area contains only a few large buildings, the residential area consists of several rather small houses along with front gardens surrounded by fences and hedges. Additionally, cars are parked on the narrow streets. Figure 4(a) and Figure 12 show aerial images of the two test sites. First we describe the experiment carried out in the campus environment, followed by a description of the experiment in the residential area.

We evaluate the global consistency of the generated maps obtained with both approaches. To this end, we recorded five data sets by steering the robot through our campus area. In each run the robot followed approximately the same trajectory. The trajectory of one of these data sets as it is estimated by our approach and a standard graph-based SLAM method is shown in Figure 7.

For each of the two approaches (our method using the aerial image and the graph-based SLAM technique that uses

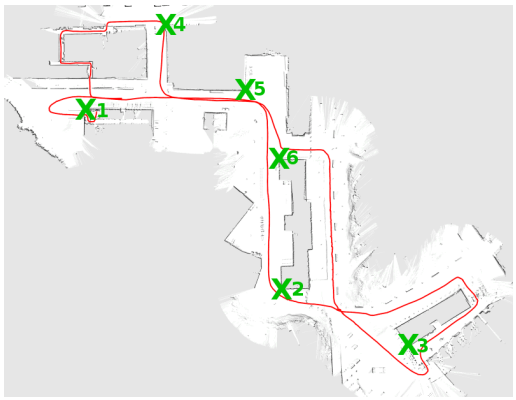


Fig. 10: The six points (corners on the buildings) we used for evaluation are marked as crosses on the map.

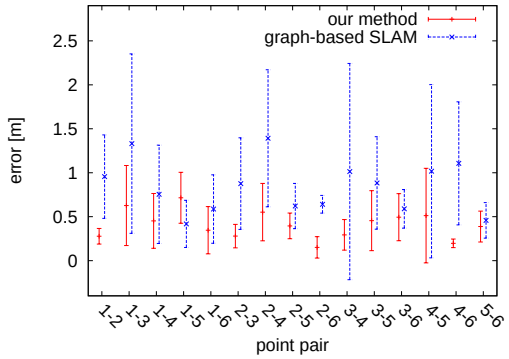


Fig. 11: Error bars ($\alpha = 0.05$) for the estimated distances between the six points used for evaluating the map consistency.

no prior information) we calculated the maximum likelihood map by processing the acquired data of each run.

For each of the five data sets we evaluated the global consistency of the maps by manually measuring the distances between six easily distinguishable points on the campus. We compared these distances to the corresponding distances in the maps (see Figure 10). We computed the average error in the distance between these points. The result of this comparison is summarized in Figure 11. As ground-truth data we considered the so-called *Automatisierte Liegenschaftskarte* which we obtained from the German land registry office. It contains the outer walls of all buildings where the coordinates are stored in a global reference frame.

An additional experiment was carried out in a residential area. An aerial image of this area is visible in Figure 12. We steered our robot five times on the streets along an approximately 710 m long trajectory. The data was recorded at different times and on several days, i.e., parts of the environment were subject to change. For example, the position of shadows changed and cars were parked in different lo-

Fig. 12: Aerial image of a residential area. The two areas marked with rectangles impose challenges to the localization algorithm. In the region marked on the left the localization using stereo vision fails. Within the area marked on the right the localization using 3D laser data is inaccurate. Using both sensors as input results in a accurate localization for the whole environment. The seven points we used for evaluation are marked as crosses on the aerial image.

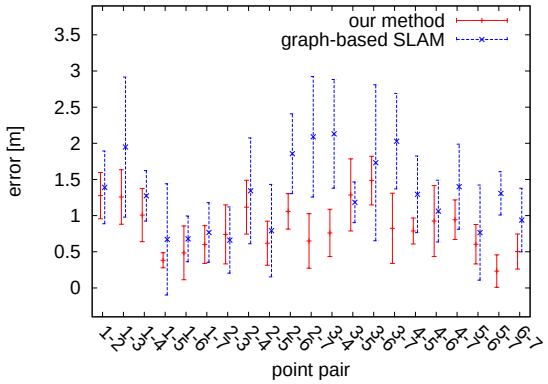


Fig. 13: Error bars ($\alpha = 0.05$) for the estimated distances between the seven points used for evaluating the map consistency.

cations. This environment is less structured than our campus environment. In particular, the parts of the environment which are marked in Figure 12 impose challenges for the MCL. The area marked on the right is dominated by vegetation along a railway embankment resulting in cluttered 3D range measurements. In this area, our approach using only 3D laser data is unable to accurately localize the robot and the MCL is likely to diverge. In the area marked on the left, the street is partially occluded due to overhanging trees. Here, the localization using stereo vision data is unable to robustly localize the vehicle. However, using both sensors, the 3D laser data and the stereo images, our approach is able to localize the robot also in these two areas. The vision data provides useful information about the road borders that are not observed by the 3D laser close to the railway whereas the 3D laser measures the trees and building structures in the other problematic region. Fusing the information of both sensors allows the robot to reliably track its position in the whole environment. For each run we computed the maximum likelihood estimates of the map for our approach and standard graph-based SLAM without prior information.

Unfortunately, an evaluation based on the ground truth map is not possible for this environment, since most of the houses are not observed due to the fences and hedges along the street. We therefore have to rely on a highly accurate GPS receiver which achieves a sub-meter accuracy. Accumulating GPS data for a longer time period allows to obtain even more accurate position estimates. To evaluate the output of our approach and the standard graph-based SLAM algorithm, we recorded the GPS information in each run. Additionally, we selected seven positions for which an accurate GPS estimate was available and we steered the robot over the same positions in each run. We measured the distance between the locations as determined by GPS and compared the distances with the maximum likelihood estimates of our approach and standard graph-based SLAM for each run. Figure 13 summarizes the results. While our approach is able to achieve an average error of 0.85 m the graph-based SLAM algorithm without prior information achieved an average error of 1.3 m.

As these two experiments reveal, SLAM without prior information results in a larger error than obtained with our approach in both environments. Additionally, the standard deviation of the estimated distances is substantially smaller than the standard deviation obtained with a graph-based SLAM approach that does not utilize prior information. Our approach is able to estimate a globally consistent map on each data set. Note that similar accuracies with respect to globally consistency might be obtained with a standard SLAM procedure if the data contained more loop closures. This indicates an additional advantage of our method, namely that it in principle does not require loop closures to achieve global consistency, at least when the prior is available.

4.4 Local Alignment Errors

Ideally, the result of a SLAM algorithm should perfectly correspond to the ground truth. For example, the straight wall of a building should lead to a straight structure in the resulting map. However, the residual errors in the scan matching process typically lead to a slightly bended wall. We investigated this in our five data sets for both SLAM algorithms by analyzing an approximately 70 m long building on our campus.

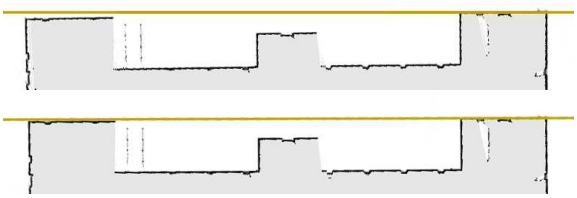


Fig. 14: Close-up view of an outer wall of a building as it is estimated by graph-based SLAM (top) and our method with prior information (bottom). In both images a horizontal line visualizes the true orientation of the wall. As can be seen from the image, graph-based SLAM bends the straight walls of the building more than our approach.

This building corresponds to the longest straight structure in this environment and was therefore chosen for evaluation. To measure the accuracy, we approximated the first part of the wall by a line and extended this line to the other end of the building. In a perfectly estimated map, both corners of the building are located on this line. Figure 14 depicts a typical result. On average the distance between the horizontal line and the corner of the building for standard graph-based SLAM is 0.5 m whereas it is 0.2 m for our approach in the five data sets.

5 Conclusion

In this paper, we presented an approach to solve the SLAM problem in mixed in- and outdoor environments based on 3D range information and using aerial images as prior information. To incorporate the prior given by the aerial images into the graph-based SLAM procedure, we utilize a variant of Monte-Carlo localization with a novel sensor model for matching 3D laser scans to aerial images. Additionally, we suggested a sensor model for using a stereo camera to localize the robot given an aerial image. Given the prior our approach can achieve accurate global consistency without the need to close loops.

Our method has been implemented and tested in a complex indoor/outdoor setting. Practical experiments carried out on data recorded with a real robot demonstrate that our algorithm outperforms state-of-the-art approaches for solving the SLAM problem that have no access to prior information. In situations, in which no global constraints are available, our approach is equivalent to standard graphical SLAM techniques. Thus, our method can be regarded as an extension to existing solutions of the SLAM problem.

Acknowledgments

This paper has partly been supported by the DFG under contract number SFB/TR-8 and by the European Commission under contract number FP7-231888-EUROPA.

References

[Bay *et al.*, 2006] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2006.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[Chen and Wang, 2007] C. Chen and H. Wang. Large-scale loop-closing by fusing range data and aerial image. *Int. Journal of Robotics and Automation*, 22(2):160–169, 2007.

[Davis, 2006] T. A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2006. Part of the SIAM Book Series on the Fundamentals of Algorithms.

[Dellaert *et al.*, 1998] 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)*, 1998.

[Ding *et al.*, 2008] M. Ding, K. Lyngbaek, and A. Zakhor. Automatic registration of aerial imagery with untextured 3d lidar models. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[Dogruer *et al.*, 2007] C. U. Dogruer, K. A. Bugra, and M. Dolen. Global urban localization of outdoor mobile robots using satellite images. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[Doucet *et al.*, 2001] A. Doucet, N. de Freitas, and N. Gordan, editors. *Sequential Monte-Carlo Methods in Practice*. Springer Verlag, 2001.

[Eustice *et al.*, 2005] 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, 2005.

[Foley *et al.*, 1993] J. D. Foley, A. Van Dam, K Feiner, J.F. Hughes, and Phillips R.L. *Introduction to Computer Graphics*. Addison-Wesley, 1993.

[Frese *et al.*, 2005] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):1–12, 2005.

[Früh and Zakhor, 2004] C. Früh and A. Zakhor. An automated method for large-scale, ground-based city model acquisition. *Int. Journal of Computer Vision*, 60:5–24, 2004.

[Grisetti *et al.*, 2005] 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, 2005.

[Grisetti *et al.*, 2009] G. Grisetti, C. Stachniss, and W. Burgard. Nonlinear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 2009. In press.

[Gutmann and Konolige, 1999] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.

[Howard, 2004] A. Howard. Multi-robot mapping using manifold representations. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 4198–4203, 2004.

[Julier *et al.*, 1995] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. of the American Control Conference*, pages 1628–1632, 1995.

[Korah and Rasmussen, 2004] T. Korah and C. Rasmussen. Probabilistic contour extraction with model-switching for vehicle localization. In *IEEE Intelligent Vehicles Symposium*, pages 710–715, 2004.

[Kümmerle *et al.*, 2009] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard. Large scale graph-based SLAM using aerial images as prior information. In *Proc. of Robotics: Science and Systems (RSS)*, Seattle, WA, USA, June 2009.

[Lee *et al.*, 2007] Kwang Wee Lee, Sardha Wijesoma, and Javier Ibañez Guzmán. A constrained slam approach to robust and accurate localisation of autonomous ground vehicles. *Robot. Auton. Syst.*, 55(7):527–540, 2007.

[Leonard and Durrant-Whyte, 1991] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4):376–382, 1991.

[Leung *et al.*, 2008] K. Y. K. Leung, C. M. Clark, and J. P. Huissoon. Localization in urban environments by matching ground level video images with an aerial image. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2008.

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

[Montemerlo and Thrun, 2004] M. Montemerlo and S. Thrun. Large-scale robotic 3-d mapping of urban structures. In *Proc. of the Int. Symposium on Experimental Robotic(ISER)*, pages 141–150, 2004.

[Montemerlo *et al.*, 2003] 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, 2003.

[Olson *et al.*, 2006] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2262–2269, 2006.

[Olson, 2008] E. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[Parsley and Julier, 2009] M. Parsley and S. J. Julier. Slam with a heterogeneous prior map. In *SEAS-DTC Conference*, 2009.

[Press *et al.*, 1992] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes, 2nd Edition*. Cambridge Univ. Press, 1992.

[Smith *et al.*, 1990] 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.

[Sofman *et al.*, 2006] B. Sofman, E. L. Ratliff, J. A. Bagnell, N. Vandapel, and T. Stentz. Improving robot navigation through self-supervised online learning. In *Proc. of Robotics: Science and Systems (RSS)*, August 2006.

[Thrun *et al.*, 2004] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Int. Journal of Robotics Research*, 23(7/8):693–716, 2004.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[Uhlmann, 1995] J. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. PhD thesis, University of Oxford, 1995.