Improving SLAM by Exploiting Building Information from Publicly Available Maps and Localization Priors

OLGA VYSOTSKA & CYRILL STACHNISS

Summary:
Maps are needed for a wide range of applications. In the context of mobile robotics, the map learning problem under uncertainty is often referred to as the simultaneous localization and mapping problem. In this paper, we aim at exploiting already available information such as OpenStreetMap data within the SLAM problem. We achieve this by relating the information about buildings with the perceptions of the robot and generate constraints for the pose-graph-based formulation of the SLAM problem. In addition to that, we present a way to select target locations for the robot so that by going there, the robot can expect to reduce it's own pose uncertainty. This localizability information is generated directly from Open Street Map data and supports active localization. We implemented and evaluated our approach using real world data taken in urban environments. Our experiments suggest that we are able to relate the newly built maps with information from Open Street Maps with the laser range finder data from the robot and in this way improve the map quality. The extension to graph-based SLAM provides better aligned maps and adds only a marginal computational overhead. Furthermore, we illustrate that the localizability information is useful to evaluate the ability to localize the robot given a trajectory.

Zusammenfassung: Nahezu alle Navigationsysteme benötigen Karten der Umgebung. Das gleichzeitige Erstellen und nutzen solchen Karten spiele eine zentrale Rolle in der Roboternavigation und wird oft als Simultaneous Localization and Mapping oder SLAM Problem bezeichnet. Nahezu alle gängigen SLAM Systeme ignorieren allerdings Hintergrundwissen oder Resourcen aus dem Netz. In diesem Papier präsentieren wir ein Verfahren, welches Daten von OpenStreetMap während dem Kartenbau nutzen kann. Dazu erweitern wir die klassische Pose-Graph Formulierung des SLAM Problem und integrieren zusätzliche Abhängigkeiten zwischen Aufnahmeposen und existierendem Kartenmaterial. Darüber hinaus können wir schätzen, welchen Regionen der Karte sich zur Positionsbestimmung besonders eignen und somit die Trajektorien des Roboters positiv bzgl. erwarteter Lokalisierbarkeit bewerten. Wir haben unseren Ansatz auf zwei realen Robotersystemen implementiert und evaluiert. Wie unsere Experimente zeigen, verbessert unser Verfahren die resultierenden Karten ohne dabei den Rechenaufwand substanziell zu erhöhen.
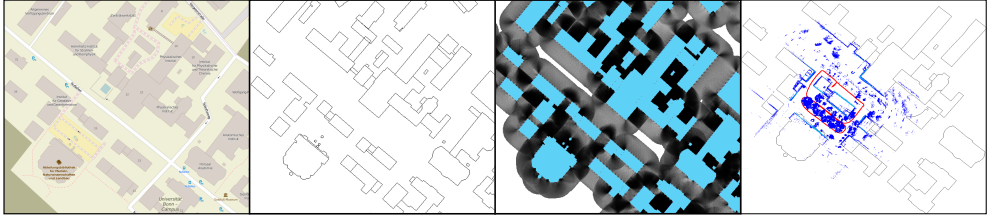
Fig. 1: From left to right: Screenshot from OpenStreetMap; map that we render for alignment; computed localizability map; resulting robot map.

1 Introduction

Most robot navigation systems require a map of the environment as well as the current pose of the vehicle in this map. Thus, having an online building procedure for a consistent map of the robot's surroundings is one of the essential prerequisites for the reliable robot operation. As errors in the robot's pose accumulate over time, building large scale maps from odometry and laser range information often leads to a drift in the trajectory estimate. GPS information can be used to compensate for that drift. This works well if a sufficient number of satellites is visible. In urban environments, however, narrow streets, high buildings and trees can hinder the capabilities of the receiver and disturb the GPS signals. This may result in a poor positioning performance. In addition to that, performing loop closures reduces the drift, but it forces the robot to re-visit places in the environment, e.g. re-entering the same street.

Recently, alternative methods for enhancing the outdoor mapping are gaining attention in the various communities, such as coupling the information from publicly available maps, like aerial photographs KÜMMERLE et al. (2011b) or OpenStreetMap (OSM) data with standard localization HENTSCHEL & WAGNER (2010) or Simultaneous Localization and Mapping (SLAM) approaches. We see exploiting such background information as orthogonal to using GPS information as aligning sensor observations with publicly available maps works typically well in GPS-denied environments. The ideas of incorporating additional prior information about the environment are also explored in context of Unmanned Aerial Vehicles (UAV), see (GERKE, 2011) and (UNGER et al., 2016).

The main contribution of this paper is a novel approach to align the robot's trajectory to OpenStreetMap data and to estimate vantage points that are likely to reduce the pose uncertainty of the robot. We integrate the ability to relate the obtained laser range measurements with the OSM data into our SLAM framework using pose graphs. This approach uses buildings as landmarks for SLAM to align the robots trajectory with existing maps. Not all places in an environment allow for matching the onboard perceptions with the OSM data. In some areas, the building structure is not distinctive enough for the robot to find an alignment. For example the result of the localization is ambiguous when the robot is located in a 'corridor'-like environment, e.g. between two long buildings, see Figure 2(left). In this case, even obtaining ideal measurements, e.g. endpoints located on the walls, will not improve the localization, since the nearby poses explain the environment as well as a query one. To perform the alignment in a better way, we would like to avoid navigating through the ambiguous regions and prefer the regions with more distinctive structure like the one in Figure 2(left-middle), where the ideal measurement originated from the query pose has a low likelihood of being observed from nearby poses. Thus, we also propose a technique that turns the information from publicly available maps into so-called localizability maps, i.e. maps that indicate how well the robot is expected to establish the data associations between its own sensor readings and the OSM information and thus can localize itself.

This papers extends our recent work along two dimensions. First, our previous work, see
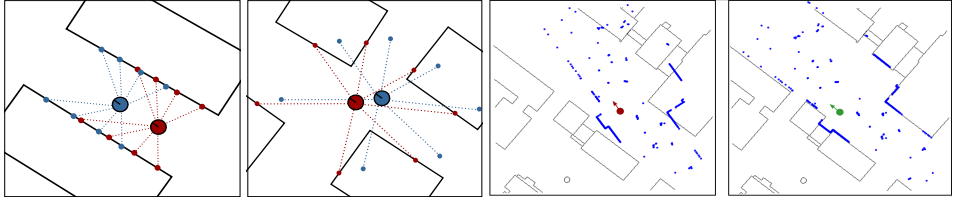
Fig. 2: First and second image: uninformative vs. informative pose. In the left image the pose of the robot (red) is not informative, because by applying the small transformation to the robot's pose (blue) the virtual measurement explains the surrounding as well as from the previous pose, whereas in the second image the pose of the robot (red) is informative, since the transformations of the pose (blue) decreases the likelihood of measurement. Third and forth image: An example correction of the robot's pose based on the aligning of the scan (blue) to the buildings in the map (black); left image shows the robot pose before correction and right afterwards.

VYSOTSKA & STACHNISS (2016), only addresses the registration, is passive and does not evaluate the localizability of a place. The previous paper focuses on alignment procedure and presents results on multiple different trajectories. In contrast to that, this paper additionally describes in detail the process of creating the localizability map as well as the extended experimental results.

2 Related Work

The first work in robotics that addressed SLAM through least squares was the work of LU & MIL-IOS (1997). Subsequently, GUTMANN & KONOLIGE (2000) focused on means for constructing pose-graphs and for detecting loop closures. Over the last 15 years, a large number of different approaches to graph-based SLAM have been proposed, for an overview see: (BAILEY & DURRANT-WHYTE, 2006a,b; GRISETTI et al., 2010a; SALAS-MORENO et al., 2013; STACHNISS, In Press).

For example, the work by KONOLIGE et al. (2010) describes a pose-graph implementation for building the linearized system in an efficient way. Solving the linear system of equations leads to optimizing the robot poses and thus map estimates. OLSON et al. (2006) investigates the use of stochastic gradient descent and GRISETTI et al. (2009) proposed an extension of Olson's approach that uses a tree parametrization of the nodes in 2D and 3D. Thrun's GraphSLAM approach (THRUN & MONTEMERLO, 2006) applies variable elimination techniques to reduce the dimensionality of the optimization problem as well as hierarchical techniques. Most SLAM approaches assume Gaussian errors in the constraints. This renders them sensitive to data association outliers. A number of approaches has been proposed to overcome this problem. For example, the approach of SÜNDERHAUF & PROTZEL (2012) scales the effect of potential outlier constraints while AGARWAL et al. (2013) proposed dynamic covariance scaling as an alternative scaling approach that does not increase the number of variables that need to be optimized. RRR (Realizing, Reversing, Recovering) by LATIF et al. (2012) tries to identify outliers by searching for the set of edges that are consistent with each other. It then rejects potentially wrong constraints.

Recently, several localization approaches were proposed that use the information from Open-StreetMap (HAKLAY & WEBER, 2008) to improve robot localization. Most of them incorporate this information into the observation model of the Monte Carlo localization (MCL). For example, HENTSCHEL et al. (2008) represent buildings as 2D line features. This line map is then used to calculate the expected range measurement at a certain robot's locations and combines MCL with a form of Kalman filtering. In our work, we also use the information about buildings, but integrate the correspondences through an ICP (Iterative Closest Point)-based matching procedure into a graph-based SLAM framework. Another approach, which is proposed by FLOROS et al. (2013), uses chamfer matching to align robot's trajectory with the road network extracted from publicly

available maps. Each particle in MCL is weighted according to the reported chamfer matching cost. In contrast to that, we select building information as this enables our system to deviate from the exact structure of the road network. Also, typically the metrical information about the road size is missing. Moreover, buildings are easier to detect in 2D range scans then the road surface. RUCHTI et al. (2015) also use the information about the road network. Instead of relying on visual odometry as FLOROS et al. (2013), they classify the 3D laser scans into road/non-road surfaces and the classification result is incorporated into the weight of the particles in the Monte Carlo localization. In contrast to that, our work incorporates building information obtained from OpenStreetMap into graph-based SLAM as additional edge constraints. An approach by PINK et al. (2009) generates features like markings of the street lanes from aerial images, matches them to the features extracted from the camera mounted in the car, and uses this information in visual navigation framework. In our work, we use a laser scanner and thus are not bounded to follow the road network. Similar to the other approaches, BRUBAKER et al. (2016) also consider the road network from the OpenStreetMap and a camera pair to perform localization.

Another approach to maintain the global consistency of the robot's maps was proposed by KÜMMERLE et al. (2011b). The consistency is achieved by augmenting the pose-graph formulation with additional constraints that come from the matching the robot's perceptions to the information from the aerial images. The aerial images are transformed into a line map using the Canny edge detector, whereas we render the map directly from the OSM information. Additionally, the authors use a variant of Monte-Carlo localization for localizing the robot within the line-map and then optimize the robot poses using a pose-graph SLAM formulation. In contrast to that, we use the information about the building locations directly from the publicly available data and incorporate this information directly into the pose-graph formulation without deploying Monte-Carlo sampling techniques.

Computing likelihood maps for localization is a well studied problem, for example in the field of active vision. The main focus here is to find the suitable vantage points for the camera for better object detection or to enhance the visual SLAM algorithms (BAJCSY, 1988; ALOIMONOS et al., 1988; CHEN et al., 2011; KIM & EUSTICE, 2014). In our work, we are interested in the similar goal of finding regions where good vantage points are located, but our primary sensor setup is a laser scanner. In the robotics community the problem of estimating the localization likelihood maps or localizability maps has also been studied in context of Teach and Repeat paradigm. For example, FURGALE & BARFOOT (2010) compute a teach corridor within which the robot can localize well in a repeat phase. DEQUAIRE et al. (2016) deploy Gaussian Processes to predict the localization envelope, the region around the taught trajectory, where the robot obtains reliable visual features for localization in a repeat phase. The authors use robust visual features as well as local path curvature to make the predictions. VELEZ et al. (2011) propose an approach to improve the object detections by planning the navigation in that way that allows the detector to be certain about the object. In the vicinity of each detectable object, they compute a likelihood field that indicates locations where reliable measurements can be taken. In our work we do not rely on any pre-trained detectors. The work with which our approach shares most similarity is the work of ROY et al. (1998). In this work they present an approach for navigating a robot, called coastal navigation, which generates the trajectories for the mobile robot that reduce the likelihood of localization errors. They estimate the likelihood of a point in the map as the amount of information content. It is computed as the difference between the expected entropy of the robot's pose given a sensor measurement and the entropy of the prior belief about the pose. The more information the cell in the map contains, the higher the likelihood. In our case, we compute the eigenvalues of the covariance estimate of the robot's pose and consequently, the smaller the selected eigenvalue, the smaller the uncertainty and the higher the likelihood will be. For the computation, ROY et al. (1998) assume to have a map of the environment constructed by a robot and the prior probability distribution about the robot's position. In our approach, we also consider to have a map of the environment, but in the form of a coarse map, rendered from the OpenStreetMap data.

3 Graph-Based SLAM Exploiting Existing Maps as Background Knowledge

The optimization step in graph-based SLAM systems aims at finding the configuration of the nodes that minimizes the error induced by observations. We consider pose-graphs, i.e., SLAM graphs in which the nodes correspond to robot poses. In general the pose of the robot consists of the location of the robot and its orientation. In this paper, the pose representation of the robot's pose is a 3 dimensional vector, consisting of two translational and one rotational components. This yields a state vector $X = (x_1, \ldots, x_n)^\top$ where x_i is the pose of node i. The error function $e_{ij}(X)$ for a single constraint between the nodes i and j is often the difference between an expected measurement $f(x_i, x_j)$ (relative pose between nodes i and j) and the obtained measurement z_{ij}:

$$e_{ij}(X) \quad = \quad e_{ij}(x_i, x_j) \quad = \quad f(x_i, x_j) - z_{ij}. \tag{1}$$

Note that alternative representations can be used to avoid problems resulting from singularities in the angular components, see (GRISETTI et al., 2010b) for details. As the error functions are typically non-linear, we linearize $e_{ij}(X)$ around the current best estimate

$$e_{ij}(X + \Delta X) \quad \simeq \quad e_{ij}(X) + J_{ij} \Delta X. \tag{2}$$

Here, J_{ij} is the Jacobian of the non-linear error function computed in the current state. Thus, the resulting minimization problem turns into

$$X^* \quad = \quad \operatorname*{argmin}_X \sum_{ij} e_{ij}(X)^\top \Lambda_{ij} e_{ij}(X), \tag{3}$$

where Λ_{ij} is the information matrix, also referred to as weight matrix associated to a constraint. Up to this point, this is the standard formulation of pose-graph SLAM and Eq. (3) can be solved as a least squares problem. To reduce the impact of the outliers, we use a robust kernel function, namely dynamic covariance scaling as proposed by AGARWAL et al. (2013). This approach rescales the error function $e_{ij}(X)$ depending on its magnitude to reweight the impact of potential outliers. This is, up to a parameter, equivalent to using a Geman-McClure kernel. This scheme for down weighting the impact of outliers is also often referred to as robust estimation, see for an overview (FÖRSTNER & WROBEL, 2016).

3.1 *Error Function Exploiting Existing Maps*

In order to incorporate additional knowledge into the optimization process and relate the pose-graph to existing data, we analogously extend the error function to

$$X^* = \operatorname*{argmin}_X \sum_{ij} e_{ij}(X)^\top \Lambda_{ij} e_{ij}(X) + F^{map}(X), \tag{4}$$

where $F^{map}(X)$ is the error introduced by the mismatch between the robot's observation and the map information. Analogous to pose-pose constraints, we split up the component $F^{map}(X)$ into individual constraints between robot poses and the OpenStreetMap information:

$$F^{map}(X) \quad = \quad \sum_i e_i^{map}(X)^\top \Lambda_i e_i^{map}(X). \tag{5}$$

The key elements in Eq. (5) are the error function $e_i^{map}(X)$ and corresponding information or weight matrix Λ_i. The remainder of this section describes how to define such an error function $e_i^{map}(X)$ and respective information matrix Λ_i. Intuitively, the error function adds an additional constraint to the graph that anchors a pose of the robot to a specific location in the map. The key
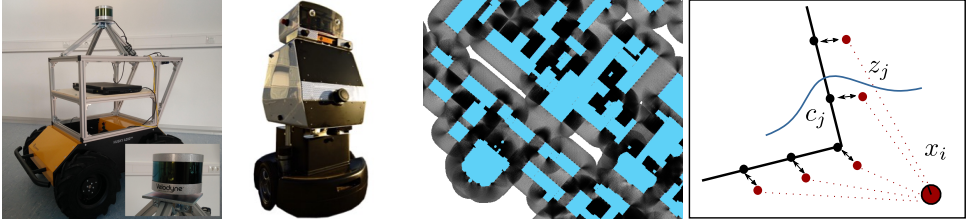
Fig. 3: Left and second image: Robots used in our experiments: robot equipped with Velodyne VLP-16 laser scanner mounted parallel to the ground and a Velodyne HDL-32E mounted on the head of the Obelix robot. Third image: An example of a localizability map. The darker the regions the bigger the likelihood to obtain in informative measurement. The buildings are marked in blue. Right: The total error of transforming virtual scan w.r.t the pose x_i depends on the distances from the measurement endpoints z_j to corresponding closest points in the buildings c_j.

challenge here is to make the correct data association between the map and the robot's own sensor readings, obtained from the pose stored in the node of the pose-graph. Once this data association is solved and the correct coordinate transformations between the robot's poses and the map are computed, least squares error minimization will provide us with the global alignment.

To make the data association between the map and the robot's poses, we use the building information in the map and the data from a 2D or 3D laser range finder installed on the robot. When aligning laser range data with the building information from OSM, a central challenge is that the laser scanner observes a large number of objects in the scene that are not stored in the map. Examples for such objects, which are not present in the publicly available map, are trees, cars or pedestrians. In contrast to most other approaches that perform localization on OSM data, we choose buildings as our features to make the data association and to compute the alignment. The majority of approaches rely on the road network to localize the robot. This is perfectly fine for cars or robots moving on the roads, but often limits the application to robots that operate on sidewalks, foot paths, or in pedestrian zones and do not follow the road network, as no good data association between the trajectory and the road network can be found in such cases.

3.2 *Error Function Exploiting Building Information for Robots Equipped with Laser Range Scanners*

In our work, we use the information about the buildings' geo-locations rather than a road network as for example done by RUCHTI et al. (2015) to enable the robot to take paths independently from the road network. We obtain the building information directly from OpenStreetMap, which can be downloaded in form of an XML-file. Inside this file, the individual buildings are stored as separate nodes. Each node is a closed polygon describing the geo-referenced walls of the building, which directly yields a map of lines that shows the walls of the buildings in the environment; see the black polygons in the two rightmost images of Figure 2 for an example. A laser scanner typically provides the scan of the environment covering a large number of objects that are not buildings and does so at a comparably high level of detail. This may hinder the matching procedure to make the correct data association between map and laser scan. Therefore, we filter the range scans so that most of the non-building objects are removed. We investigated several techniques and in the end opted for an unsupervised approach that performs filtering based on line extraction. It does not require manually labeled training data and can be executed efficiently. We employ the Douglas-Peuker algorithm for converting the raw 2D range scan into a polyline. We convert the polyline into a set of potentially disconnected lines based on two parameters: the length of a line and the number of laser end points assigned to each line. The problem of detecting building structures has

also been investigated in the context of reconstructing the 3D structure. For example FISCHER et al. (1998) use generic models to extract 3D buildings from the aerial images. HUBER et al. (2003) fuse the LIDAR data with aerial imagery and apply polyhedral models to reconstruct the buildings. We, on the other hand, detect buildings in the single 2D laser scan, not taking into account the information from the maps.

Since our aim is to incorporate the knowledge about the environment from the map into the graph optimization procedure to refine the robot's trajectory, the error function for this constraint should reflect the misalignment between the current robot's pose and the map. Intuitively, the bigger is the misalignment between the scan and the buildings in the map, the bigger the error should be. To estimate the (mis-)alignment, we use the Iterative Closest Point (ICP) (BESL & MCKAY, 1992) algorithm to match the current laser scan and the map of building. For finding the correspondences in ICP, standard nearest neighbor data association is applied.

More precisely, the error term $e_i^{map}(x_i)$ is defined based on the difference between the current robot's pose x_i and the pose $\hat{x}_i$, computed by aligning the scan in the building map. The 2D state of the robot x_i consists of translational t_i and rotational θ_i components, i.e., forms an element in $SE(2)$, special euclidean group for two dimensions. The same holds for the state $\hat{x}_i$. Thus, the error function and its Jacobian turns into:

$$e_i^{map}(x_i) = \begin{pmatrix} \hat{R}_i^\top (t_i - \hat{t}_i) \\ \theta_i - \hat{\theta}_i \end{pmatrix} \quad \text{and} \quad J_i = \frac{\partial e^{map}(x_i)}{\partial x_i} = \begin{pmatrix} \hat{R}_i^\top & 0 \\ 0 & 1 \end{pmatrix}. \tag{6}$$

with $\hat{R}_i$ being the standard 2D rotation matrix corresponding to the angle $\hat{\theta}_i$. For the ICP algorithm to operate reliably, we need a good initial guess. In our setup, the initial guess is achieved by either manually specifying the first pose of the robot on the map or by using an initial guess from a consumer GPS with an accuracy of a few meters. The initial guess of all successive poses is then automatically obtained from the odometry constraints of the graph or by incremental scan to scan alignment typically used in graph-based SLAM with laser range finders.

Finally, we have to compute the weight matrix Λ_i of a map constraint, which is the inverse of the covariance matrix of the ICP alignment, i.e., $\Lambda_i = (\Sigma_i^{ICP})^{-1}$. We compute the covariance matrix Σ_i^{ICP} from the ICP result by using the Hessian as described by BENGTSSON & BAERVELDT (2003). This assumes the error function e^{ICP} used in the ICP algorithm to be quadratic near the optimal solution, i.e.,

$$e^{ICP} = \sum_k \|Tp_k - q_k\|^2, \tag{7}$$

where p_k is a point from a laser scanner that belongs to the detected buildings and q_k is a corresponding closest point in the buildings taken from the publicly available map. The optimal transformation T that the ICP algorithm reports is found by minimizing the function e^{ICP} with the covariance matrix of T as

$$\Sigma_i^{ICP} = \text{cov}(T) = 2\sigma^2 \left(\frac{\partial^2}{\partial T^2} e^{ICP} \right)^{-1} = 2\sigma^2 H_{\text{icp}}^{-1}, \tag{8}$$

where H_{icp} is the Hessian matrix of e^{ICP} and σ^2 is the variance factor.

So far, we described how to obtain the error function for 2D range data such as a horizontally mounted 3D range scanner, but it works analoguously on data from a 3D laser scanner. In this paper, we use 2D and 3D range data. In case of 3D data such as the one coming from a Velodyne scanner, for every individual scan we construct a 3D point cloud and generate new virtual 2D laser scans given the planar surfaces in the cloud. We basically follow the approach proposed by WULF et al. (2004), which is also used by BOGOSLAVSKYI et al. (2016) and HENTSCHEL et

al. (2008).

3.3 *Error Minimization*

Given the error function e_i^{map} with corresponding information matrix Λ_i^{map} and Jacobian J_i, we use Levenberg-Marquardt optimization to solve the problem given in Eq. (3). This leads to iteratively solving a linear system of the form

$$(H + \lambda I)\Delta X^* = -b, \tag{9}$$

with

$$H = \sum_{ij} J_{ij}^\top \Lambda_{ij} J_{ij} + \sum_i J_i^\top \Lambda_i J_i \quad \text{and} \quad b = \sum_{ij} J_{ij}^\top \Lambda_{ij} e_{ij} + \sum_i J_i^\top \Lambda_i e_i^{map}. \tag{10}$$

H and b are the key elements and are computed from the linearized error functions and λ is the damping factor used in the Levenberg-Marquardt. The term ΔX^* refers to the increments that are added to the graph configuration in order to minimize our error function in the current iteration. In our implementation, we use the g2o framework by KÜMMERLE et al. (2011a) to conduct the minimization with dynamic covariance scaling by AGARWAL et al. (2013). This yields an update of the graph configuration in every iteration of the form:

$$X \leftarrow X + \Delta X^*. \tag{11}$$

We do not execute this procedure in a batch fashion but selected the incremental option of the g2o optimizer, which allows to optimize the trajectory in chunks. In our current implementation, we trigger an update whenever the robot drove for 25 m. This has two advantages. First, the data are available already online during mapping. Second, the correction of the trajectory up to a point in time t_1 will simplify the data association for the ICP step for subsequent matching with $t > t_1$ and, thus, has the potential to provide a better alignment. As a result of that, we obtain an optimized pose-graph that is aligned with the provided map.

4 Estimating Localizability for Actively Reducing Pose Uncertainty

By using publicly available maps such as OpenStreetMap we are able to better align the robot's trajectory and, hence, the robot's own map to the surrounding environment. This approach, however, relies on the observations that the robot obtains, which in turn depend on the local surroundings of the robot. The ability to match the local perceptions to the OSM data depends on the visibility of buildings and the local geometry or arrangement of the buildings. The goal of this section is to describe an approach to estimate the ability of the robot to align is perception with the OSM data *before* moving there. Thus, we aim at estimating a visibility map from the OSM data and selecting the regions that are expected to provide good vantage points that support the alignment. A measurement will be informative for the robot if (i) the buildings can be detected in the individual scans and (ii) the observed structure allows for reducing robot's pose uncertainty. Thus, we present a method that estimates the regions in the environment, where the informative laser scans are likely to be obtained, given the information from publicly available maps.

To reason about the informativeness of a particular pose on the map, we need to specify the function, which measures the likelihood of obtaining a certain laser scan given a pose. For our sensor model, we assume that individual beams z_j in the laser scan are independent and that the measurement noise for each beam is Gaussian, i.e., $p(z_j) \sim \mathcal{N}(c_j, \sigma_l)$, where c_j are the closest points in the buildings that correspond to the individual laser beam endpoints z_j, for visualization see Figure 3 (right).
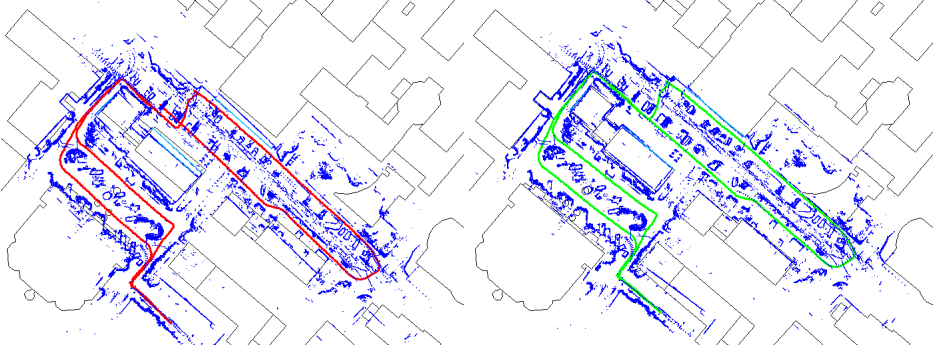
Fig. 4: Example of aligning the robot's trajectory with the buildings on the map and as a result of it improved loop closure, which also leads to more consistent robot map. The OSM map is rendered with 0.3 meters per pixel resolution.

Our goal is to estimate for every potential robot pose the associated uncertainty of the pose estimate based on the visibility of the building structures in the scene. We start with simulating an ideal (virtual) laser scan at every potential pose x_i. This measurement assumes that only the building from OSM data exist in the environment. This simulated scan is generated by performing a ray-cast operation in the maps from OSM. We then estimate how well this ideal measurement matches to the OSM maps under pose uncertainty. We estimate this by applying small perturbations to the robot's pose in x, y and θ. By doing so, we form a set S of potentially similar pose configurations x_j and estimate the corresponding errors which arise by comparing the a virtual scan to the map in the new robot configuration. We estimate an error for a pose configuration as a sum of squared errors $e(z_j)$ of individual beams of the scan. We then perform an approximation taking into account our assumption about the probability distribution of an individual beam. We approximate the probability of taking a virtual measurement in the configuration x_j as $p(x_j) \sim \exp{(-\frac{\sum e(z_j)}{2N\sigma_l^2})}$, where N is size of the scan and $e(z_j)$ is a squared distance between measurement endpoint z_j and closest building in the map. In other words, by applying these actions, we approximate the unknown probability distribution about the robot's pose. Having the samples from a probability distribution, we obtain the covariance matrix, i.e. the uncertainty of the pose, as follows:

$$\mathrm{cov}(x_i) = \sum_{x_j \in S} p(x_j)\,(x_j - q_{x_i})(x_j - q_{x_i})^\top \tag{12}$$

where q_{x_i} corresponds to the coordinates of the query pose.

However, to be able to reason about the informativeness of the different regions in a more quantitative way, we compute the eigenvalues/eigenvectors of the respective covariance matrices and therefore obtain the information in which direction we are most uncertain about the pose. Afterwards, we update our localizability maps with the biggest eigenvalue for every pose. This ensures that we take into account the value of the biggest uncertainty across all dimensions. As a result, the regions in the map with smaller values correspond to the places where the largest uncertainty over individual dimensions is smaller in comparison to other regions or, in other words, the regions where the informative measurements are more likely, see Figure 3 for visualization.
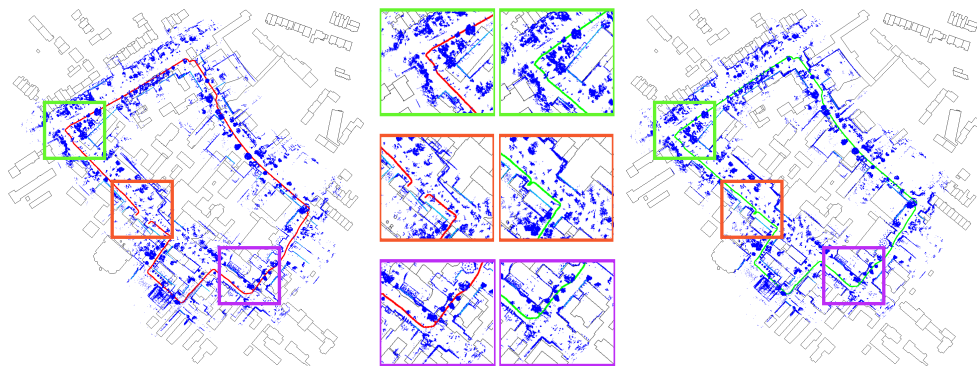
Fig. 5: Left: overlayed trajectory before the optimization. Right: trajectory after optimization. Middle: Zoom in parts of the trajectory; upper row: case of corrected misalignment at the beginning of the trajectory; bottom: fixed misalignment at the end of the trajectory; center: successfully closed loop.

5 Experiments

The evaluation is designed to illustrate the performance of our approach and to support the following claims made in this paper. These key claims are that (i) we improve the map alignment with our approach, (ii) can handle situations in which the map data are partially outdated, for example if buildings have been demolished or new buildings have been built, (iii) all operations yield only a small computational overhead compared to a standard graph-based SLAM system, and (iv) the localizability maps provide information about the ability of the platform to localize along a given trajectory. We performed our experiments in outdoor urban environments using the odometry from two robots, Husky A200 and Obelix, both are depicted in Figure 3 and are equipped with Velodyne VLP-16 and Velodyne HDL-32E laser scanners respectively. For detecting lines in laser scans, we used Douglas-Peuker algorithm and maintain only lines with a length of at least $5\,m$ containing at least 100 laser end points (for a scanner with a $0.25°$ resolution). This clearly eliminates also end-points belonging to walls, but overall, it keeps the number of false-positives small — which is more important for us in order to obtain a robust alignment between laser scan and map.

5.1 *SLAM Exploiting OpenStreetMap Data*

The first set of experiments is designed to illustrate that we use the information from publicly available maps to locate the robot within these maps. By considering the individual laser scans obtained by the robot within the alignment procedure, we even have the possibility to find loop closures that are partially missed by the pose-graph SLAM otherwise. Figure 4 depicts a trajectory of the robot overlayed on the map when using traditional 2D graph-based optimization (red) without considering map information and when incorporating the map structure into the optimization process (green). The map in this case is rendered with $0.3\,m$ per pixel resolution and covers the area of $250 \times 300\,m$. As can be seen, not only the robot's own map is better aligned with the structure of the environment, but also the loop closure was correctly detected due to the aligning laser scans to the buildings. To provide a more quantitative evaluation, we compute the error for the final pose of the robot, as a distance between the optimized pose and manually specified ground truth position. The error of the final pose without map information is about $5\,m$, whereas using the map priors lead to an error for the final pose of about $1\,m$. Figure 5 represents another example of the robot's trajectory, here using a 3D Velodyne data, which spans over a significantly larger
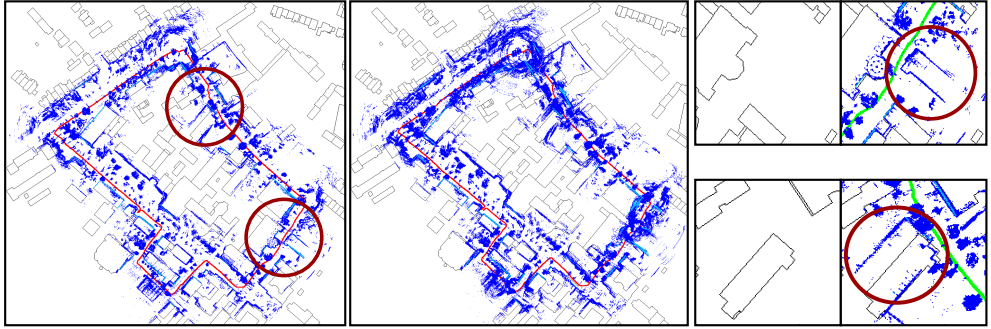
Fig. 6: Enabling / Disabling robust kernel function (DCS). Left: optimization using DCS. Middle: optimization without robust kernel functions. Right: Zoomed views of the map inconsistencies our system can deal with. Light blue: detected buildings
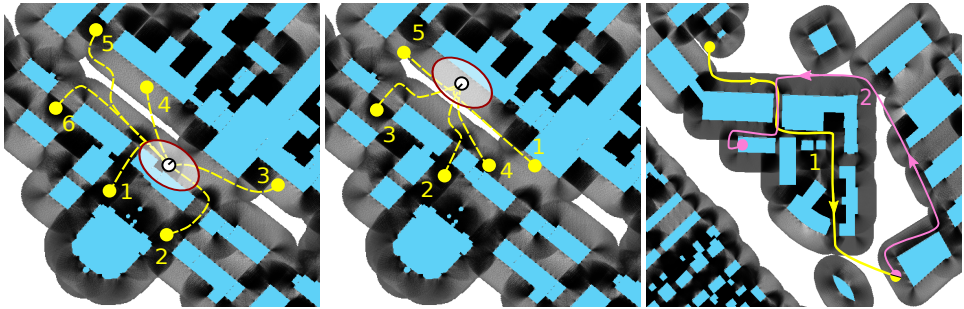

Fig. 7: Planned trajectories. Left: Robot starts in a location which supports an alignment well (setup 1). Middle: Robot starts in a location with low likelihood of being able to compute the right data association between OSM information and its sensor readings (setup 2). Trajectories recorded in Freiburg. Trajectory 1 goes through the likely regions more often than trajectory 2.

area than the previous example. For this experiment the map represents the area of size 500×500 meters and thus is rendered with 0.5 meters per pixel resolution. In Figure 5, only the laser end points that do not belong to the ground plane are plotted. As can be seen, the map produced by the robot is filled with substantial clutter in the environment, which makes the aligning procedure more challenging. Nevertheless, our approach is able to fix the misalignments that come from the inaccuracy of the initial position, see Figure 5 middle column upper and bottom image, as well as to find loop closures missed by the pure pose-graph approach, see Figure 5 middle center. Using the same definition of the localization error as for the previous experiment, the error of the last pose using pure pose-graph formulation is 22 meters, whereas with OSM information is $0.5\,m$.

5.2 *Map Inconsistencies*

The second experiment is designed to show that our approach is able to deal with a certain amount of map inconsistencies. Such inconsistencies result from different sources, for example wrongly mapped buildings, a demolished building that is still present in the map or a building that was built after the time of the map creation. The two examples for inconsistencies shown in Figure 6 are real inconsistencies in OSM data and not artificially simulated. As we take the matching-dependent uncertainty into account, the information about the inconsistencies is incorporated into

Tab. 1: Timing results for processing the whole dataset (full) and processing a chunk (per chunk) of the dataset after driving for $25\,m$; dist - length of the trajectory; pose-graph - processing using standard pose-graph formulation only, OSM - processing time needed to optimize additional edges introduced by OSM constraint.

		full		per chunk	
	dist	pose-graph	osm	pose-graph	osm
dataset 1	168 m	9.89 s	0.9 s	1.75 s	0.16 s
dataset 2	336.6 m	62 s	0.83 s	5.52 s	0.074 s
dataset 3	579.6 m	41.5 s	4.93 s	2.14 s	0.25 s
dataset 4	1040 m	86 s	4.1 s	2.48 s	0.11 s

the optimization process as well. For this experiment we used the same trajectory as for the previous experiment. Figure 6 (third column) depicts two examples of the map inconsistencies that are successfully handled by our approach. The upper image depicts a situation in which the building is visible in the scan and not present in the map and the image in the bottom shows a case where the building is wrongly mapped (building in the map is too small). Our system deals with inconsistencies through the use of a robust kernel function. Figure 6 (middle) shows the effect of disabling the robust kernel function. As can be seen, the robot map gets distorted near the wrongly mapped buildings, corresponding places are marked with circles.

5.3 *Execution Time*

In this experiment, we show that our approach adds only a small computational overhead to the simultaneous localization and mapping process. We ran our algorithm on different datasets with various size and complexity and summarize the runtime results in the Table 1. The datasets are obtained with the robot setup specified before, namely using Husky A200 with Velodyne VLP-16 or Hokuyo laser scanners. As can be seen, the time needed to process additional map knowledge (OSM, 4th and 6th column) is almost negligible in comparison to the time needed for the pose graph SLAM (pose-graph, 2nd and 5th column). This means that we integrated our extension into the optimization procedure without adding a significant computational overhead.

5.4 *Active Localization*

The last set of experiments is designed to show that the localizability maps provide information about the ability of the robot to reduce its pose uncertainty if obtaining scans at given locations in the map. To show this, we initialized the robot's believe with a pose uncertainty of up to $5\,m$ and $20°$. Then, the robot had to select a target location within a $100\,m$ range and to update the believe about its own pose based on the measurements acquired on the way. Figure 7 shows this experiment for two initial locations. In one place, the robot localizes already well given its initial pose (left image) while in the other case, the initial pose does not offer good features for localization (right). For each location, we sample possible target locations and evaluated the ability of the robot to improve its pose estimate while approaching the sampled locations. Several of the selected trajectories end in the likely regions (setup 1 the trajectories 1, 2, 5, 6; setup 2 the trajectories 2, 3), whereas others are located in unlikely regions (setup 1: 3, 4 and setup 2: 1, 4, 5).

We have recorded the individual trajectories separately with our robot in Bonn and measured the ground truth locations at the end of each trajectory. Thus, we are able to compute the localization error as the absolute distance between mean estimate and true location. For more quantitative results, we repeated this experiment for randomly sampled starting locations in an area of 5 m and

Tab. 2: The distribution of the localization errors for the planned trajectories after the execution.

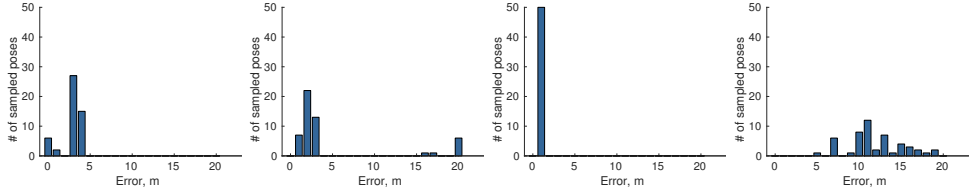|	Endpoint region	Trajectory	Localization error, %		
			decreased	increased	diverged
setup 1	likely	1	94	0	6
		2	100	0	0
		5	86	0	14
		6	98	2	0
	¬likely	3	84	4	12
		4	0	94	6
setup 2	likely	3	42	4	54
		2	100	0	0
	¬likely	1	0	100	0
		5	0	84	16
		4	0	100	0

Fig. 8: Localization error distribution for the setup 1 for the trajectory 2 with the endpoint in likely region (left) and for the trajectory 3 with the endpoint in unlikely region (2nd). Localization error distribution for the setup 2 for the trajectory 2 with the endpoint in likely region (3rd) and for the trajectory 1 with the endpoint in unlikely region (right).

$20°$ in orientation. The results are summarized in the Table 2. The table depicts the number of runs in which the the localization errors decreased, increased or even diverged (error larger than $20\,\mathrm{m}$). As can be seen, trajectories that lead through or end in likely regions result in a better localization on average, independent from the starting pose. For the trajectories that lead through the likely regions (setup 1: 1, 2, 5, 6 and setup 2: 2, 3) the localization error is reduced in $86 - 100\%$ of the cases, whereas for the trajectories that prefer unlikely regions (setup 1: 3, 4 and setup 2: 1, 5, 4) the localization error mostly increases. Additionally, the need of navigating through the likely regions becomes more important if the starting robot pose lies in the unlikely region as in setup 2. If the robot starts in a region that supports localization (and thus it is well localized), the gain of the localizability maps is obviously limited (setup 1: 3). Figure 8 shows the distribution of the localization error for setups 1 and 2 respectively. As can be seen for the trajectories in the likely regions (left images) the error is less than $5\,\mathrm{m}$, i.e. less than the original pose uncertainty. However, the trajectories from the unlikely regions (right images) may result in the decrease of the localization error as in Figure 8 (left) but also in an increase as for example in Figure 8 (right).

The last experiment is designed to show that trajectories that lead through likely regions in the localizability map lead more often to a decreased localization error than the trajectories that that lead through less likely regions. To illustrate this, we recorded the trajectories depicted in the Figure 7 (right). We recorded the data using the robot Obelix equipped with a Velodyne HDL-32E in Freiburg. The area depicted in Figure 7 (right) and Figure 9 is approximately $350 \times 320\,m$. The first trajectory passes through the likely regions and leads to a decrease in the localization error more often than the second trajectory on the right, which leads through regions that do not support localization that well, see Figure 9. We would like to point out that the start and end points in this experiment were selected manually so that they either lead trough likely regions in the localizability map or not. Thus, we investigate the relevance of the proposed localizability
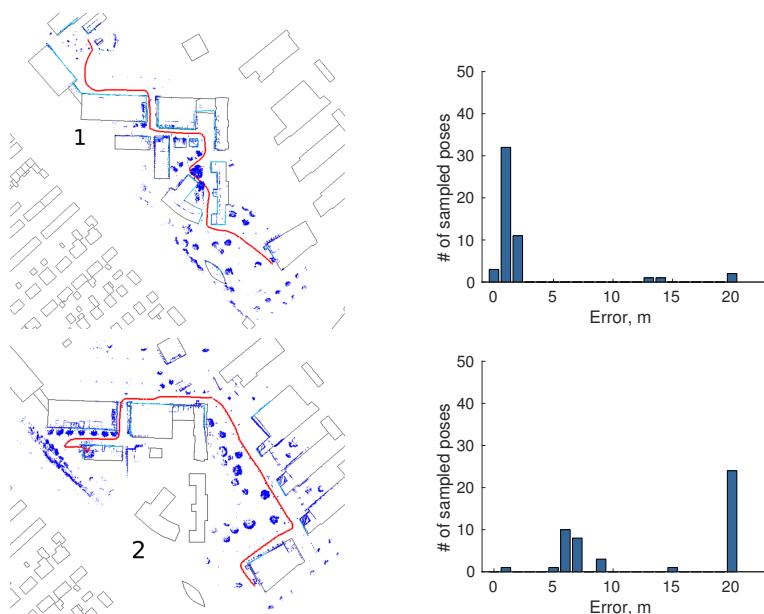
Fig. 9: First column: Obtained robot maps after traveling along the paths depicted in Figure 7(right). Right column: Corresponding distribution of the localization errors, when having uncertainty in the initial pose.

information on the resulting localization performance of the robot traveling along a path and not the performance of a specific planning or target point selection algorithm. Incorporating a planning and navigation system that automatically use the localizability maps is a more involved problem and beyond the scope of this paper.

6 Conclusion

In this paper, we presented a novel approach to improve the quality of maps built with mobile robots by exploiting information from publicly available maps such as Open Street Map data. Our approach seeks to find an alignment between the laser scanner data recorded in the mobile platform and the building information from OpenStreetMap data. In addition to that, we estimate the ability of the robot to localize itself in a given region of the map by computing a so-called localizability map. As we have illustrated through a large set of real world experiments, the exploitation of OSM data improves the map alignment process and provides relevant information about the ability of the robot to localize itself in certain locations.

References

AGARWAL, P., TIPALDI, G., SPINELLO, L., STACHNISS, C. & BURGARD, W., 2013: Robust map optimization using dynamic covariance scaling. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 62–69.

ALOIMONOS, J., WEISS, I. & BANDYOPADHYAY, A., 1988: Active vision. – International journal of computer vision **1** (4): 333–356.

BAILEY, T. & DURRANT-WHYTE, H., 2006a: Simultaneous localisation and mapping (SLAM): Part I. – Robotics & Automation Magazine **13** (2): 99–110.

BAILEY, T. & DURRANT-WHYTE, H., 2006b: Simultaneous localisation and mapping (SLAM): Part II. – Robotics & Automation Magazine **13** (3): 108 –117.

BAJCSY, R., 1988: Active perception. – Proceedings of the IEEE **76** (8): 966–1005.

BENGTSSON, O. & BAERVELDT, A.-J., 2003: Robot localization based on scan-matching estimating the covariance matrix for the IDC algorithm. – Robotics and Autonomous Systems **44** (1): 29 – 40.

BESL, P. & MCKAY, N. D., 1992: A method for registration of 3-d shapes. – IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (2): 239–256.

BOGOSLAVSKYI, I., MAZURAN, M. & STACHNISS, C., 2016: Robust homing for autonomous robots. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2550–2556.

BRUBAKER, M., GEIGER, A. & URTASUN, R., 2016: Map-based probabilistic visual self-localization. – IEEE transactions on pattern analysis and machine intelligence **38** (4): 652–665.

CHEN, S., LI, Y. & KWOK, N. M., 2011: Active vision in robotic systems: A survey of recent developments. – Int. Journal of Robotics Research **30** (11): 1343–1377.

DEQUAIRE, J., TONG, C. H., CHURCHILL, W. & POSNER, I., 2016: Off the beaten track: Predicting localisation performance in visual teach and repeat. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 795–800.

FISCHER, A., KOLBE, T. H., LANG, F., CREMERS, A. B., FÖRSTNER, W., PLÜMER, L. & STEINHAGE, V., 1998: Extracting buildings from aerial images using hierarchical aggregation in 2d and 3d. – Computer Vision and Image Understanding **72** (2): 185–203.

FLOROS, G., VAN DER ZANDER, B. & LEIBE, B., 2013: Openstreetslam: Global vehicle localization using openstreetmaps. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 1054–1059.

FÖRSTNER, W. & WROBEL, B., 2016: Photogrammetric Computer Vision. Springer. chapter Robust estimation and outlier detection, 141–159.

FURGALE, P. & BARFOOT, T. D., 2010: Visual teach and repeat for long-range rover autonomy. – J. Field Robot. **27** (5): 534–560.

GERKE, M., 2011: Using horizontal and vertical building structure to constrain indirect sensor orientation. – ISPRS Journal of Photogrammetry and Remote Sensing **66** (3): 307–316.

GRISETTI, G., KÜMMERLE, R., STACHNISS, C. & BURGARD, W., 2010a: A tutorial on graph-based SLAM. – IEEE Transactions on Intelligent Transportation Systems Magazine **2**: 31–43.

GRISETTI, G., KÜMMERLE, R., STACHNISS, C., FRESE, U. & HERTZBERG, C., 2010b: Hierarchical optimization on manifolds for online 2D and 3D mapping. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 273–278.

GRISETTI, G., STACHNISS, C. & BURGARD, W., 2009: Non-linear constraint network optimization for efficient map learning. – IEEE Transactions on Intelligent Transportation Systems **10**: 428 – 439.

GUTMANN, J.-S. & KONOLIGE, K., 2000: Incremental mapping of large cyclic environments. – Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA), 318–325.

HAKLAY, M. & WEBER, P., 2008: Openstreetmap: User-generated street maps. – Pervasive Computing, IEEE **7** (4): 12–18.

HENTSCHEL, M. & WAGNER, B., 2010: Autonomous robot navigation based on openstreetmap geodata. – Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE, 1645–1650.

HENTSCHEL, M., WULF, O. & WAGNER, B., 2008: A gps and laser-based localization for urban and non-urban outdoor environments. – Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 149–154.

HUBER, M., SCHICKLER, W., HINZ, S. & BAUMGARTNER, A., 2003: Fusion of lidar data and aerial imagery for automatic reconstruction of building surfaces. – Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on, 82–86.

KIM, A. & EUSTICE, R. M., 2014: Active visual slam for robotic area coverage: Theory and experiment. – Int. Journal of Robotics Research.

KONOLIGE, K., GRISETTI, G., KÜMMERLE, R., BURGARD, W., LIMKETKAI, B. & VINCENT, R., 2010: Sparse pose adjustment for 2d mapping. – Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 22 – 29.

KÜMMERLE, R., GRISETTI, G., STRASDAT, H., KONOLIGE, K. & BURGARD, W., 2011a: g2o: A general framework for graph optimization. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 3607 – 3613.

KÜMMERLE, R., STEDER, B., DORNHEGE, C., KLEINER, A.AND GRISETTI, G. & BURGARD, W., 2011b: Large scale graph-based slam using aerial images as prior information. – Autonomous

Robots **30** (1): 25–39.

LATIF, Y., CADENA, C. & NEIRA, J., 2012: Robust loop closing over time. – Proc. of Robotics: Science and Systems (RSS).

LU, F. & MILIOS, E., 1997: Globally consistent range scan alignment for environment mapping. – Autonomous Robots **4**: 333–349.

OLSON, E., LEONARD, J. & TELLER, S., 2006: Fast iterative optimization of pose graphs with poor initial estimates. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2262–2269.

PINK, O., MOOSMANN, F. & BACHMANN, A., 2009: Visual features for vehicle localization and ego-motion estimation. – IEEE Intelligent Vehicles Symposium, 254–260.

ROY, N., BURGARD, W., FOX, D. & THRUN, S., 1998: Coastal navigation – robot motion with uncertainty. – Proceedings of the AAAI Fall Symposium: Planning with POMDPs, Stanford, CA, USA.

RUCHTI, P., STEDER, B., RUHNKE, M. & BURGARD, W., 2015: Localization on openstreetmap data using a 3d laser scanner. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 5260–5265.

SALAS-MORENO, R. F., NEWCOMBE, R. A., STRASDAT, H., KELLY, P. H. & DAVISON, A. J., 2013: Slam++: Simultaneous localisation and mapping at the level of objects. – Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1352–1359.

STACHNISS, C., In Press: Springer Handbook of Photogrammetry. Springer. chapter Simultaneous Localization and Mapping.

SÜNDERHAUF, N. & PROTZEL, P., 2012: Switchable constraints for robust pose graph slam. – Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 1879–1884.

THRUN, S. & MONTEMERLO, M., 2006: The graph SLAM algorithm with applications to large-scale mapping of urban structures. – Int. Journal of Robotics Research **25** (5-6): 403.

UNGER, J., ROTTENSTEINER, F. & HEIPKE, C., 2016: Integration of a generalised building model into the pose estimation of uas images. – ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XLI-B1**: 1057–1064.

VELEZ, J., HEMANN, G., HUANG, A. S., POSNER, I. & ROY, N., 2011: Planning to perceive: Exploiting mobility for robust object detection.. – International Conference on Automated Planning and Scheduling (ICAPS), 266–273.

VYSOTSKA, O. & STACHNISS, C., 2016: Exploiting building information from publicly available maps in graph-based slam. – Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 4511–4516.

WULF, O., ARRAS, K. O., CHRISTENSEN, H. I. & WAGNER, B., 2004: 2d mapping of cluttered indoor environments by means of 3d perception. – Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 4204–4209.