Aus dem Institut für Technische Informatik
der Universität zu Lübeck

Direktor: Prof. Dr.-Ing. Mladen Berekovic

Robust Localization and Mapping in Changing Environments using Semantic Perception

Inauguraldissertation
zur
Erlangung der Doktorwürde
der Universität zu Lübeck
Aus der Sektion Informatik / Technik

Vorgelegt von
Dipl.-Inf. (FH) Marian Himstedt
aus Frankfurt (Oder)

Lübeck, 2018

1. Berichterstatter: Prof. Dr.-Ing. Erik Maehle, Universität zu Lübeck

2. Berichterstatter: Prof. Dr.-Ing. Alexander Schlaefer, TU Hamburg

Tag der mündlichen Prüfung: 29.08.2018

Zum Druck genehmigt: Lübeck, den 30.08.2018

Zusammenfassung

Die Fähigkeit, die eigene Position und Orientierung im Raum zu bestimmen, ist eine Grundvoraussetzung für mobile Roboter. Es bietet die Basis für die Planung von Pfaden, um von der aktuellen Position zu definierten Zielpunkten zu gelangen. Weiterhin ermöglicht es, ortsabhängige Dienste an der aktuellen Position anzubieten. Von besonderer Bedeutung ist dabei, dass die Positionsbestimmung in einem festen, globalen Koordinatensystem erfolgen muss, das unabhängig von der Bewegung des Roboters ist, um die Prozesssicherheit gewährleisten zu können.

In Außenumgebungen bietet GPS die Möglichkeit, eine absolute Position zu bestimmen. Die Verfügbarkeit und Genauigkeit dieses Systems ist allerdings stark von der Umgebung abhängig. So steht dieser Dienst nur in Bereichen, die ein hinreichend freies Sichtfeld zu GPS-Satelliten gewährleisten, zur Verfügung. Dies schließt beispielsweise den Einsatz in dicht bebauten urbanen Umgebungen (sog. Häuserschluchten), Wäldern, Tunneln und auch innerhalb von Gebäuden aus. Ziel dieser Arbeit ist daher die Entwicklung von Lokalisationsmechanismen, die unabhängig von der technischen Infrastruktur der Umgebung sind.

Die infrastrukturunabhängige Positionsbestimmung benötigt Umgebungskarten, die als Referenz dienen und zum Abgleich mit dem aktuellen Sensoreindruck genutzt werden können. Es gibt zwar für viele Gebäude Grundrisse, jedoch kann nicht grunsätzlich davon ausgegangen werden, dass diese immer zur Verfügung stehen. Ferner ist auch unklar, ob sie genügend Umgebungselemente enthalten, die mit einem Sensoreindruck korreliert werden können. Teil dieser Arbeit ist die Entwicklung eines Algorithmus, der den Aufbau von Umgebungskarten ermöglicht und dabei lediglich die am Roboter zur Verfügung stehende Sensorik verwendet. Insbesondere ist in diesem Rahmen ein neuartiges Verfahren entwickelt worden, welches einen effizienten Vergleich von Sensoreindrücken erlaubt und daher auch die Kartierung sehr weitläufiger Umgebungen möglich macht.

Die vorliegende Arbeit widmet sich weiterhin der Erkennung und Klassifikation von Objekten im Kontext mobiler Robotik. Die dabei entwickelten Verfahren sind insbesondere hinsichtlich ihrer Performance optimiert. Es wird gezeigt, wie Objekterkennung für die semantische Annotation von Umgebungskarten genutzt werden kann.

Die Herausforderung bei der Positionsbestimmung unter Nutzung von lediglich der am Roboter zur Verfügung stehenden Sensorik im Vergleich zu Referenzsystemen wie GPS ist die Gewährleistung der Robustheit in dynamischen Umgebungen. Der Zustand der Umgebung kann sich über die Zeit wesentlich zu dem zur initialen Kartierung erfassten verändern. Diese können beispielsweise aufgrund geparkter Fahrzeuge, Personen oder auch palettierter Waren in der Intralogistik entstehen. Im Rahmen dieser Arbeit wird ein Algorithmus zur Lokalisierung speziell für den Einsatz in veränderlichen Umgebungen vorgestellt. Im Vergleich zum aktuellen Stand der Forschung wird diese Robustheit durch

die Einbeziehung von Objekterkennung, semantisch annotierten Umgebungskarten und a-priori Wissen über die Einsatzumgebung erreicht.

Die entwickelten Verfahren setzen entfernungsmessende Sensoren, die an einem mobilen Roboter angebracht sind, voraus. Experimentelle Untersuchungen zeigen Ergebnisse, welche auf Basis von Messdaten eines 2D Laserscanners oder einer RGBD-Kamera beruhen. Letztere arbeitet nach dem Prinzip des stukturierten Lichts und bietet die Basis für die Objekterkennung, welche im Rahmen der semantischen Kartenannotation und der Lokalisierung benutzt wird.

Abstract

The ability to estimate the position and orientation is a fundamental requirement for mobile robots. For example it serves as input for planning paths from the current position to specific destinations and enables location-dependent services.

The position estimation has to be carried out with respect to a fixed global coordinate frame being independent of the robot's motion in order to ensure safe operation.

GPS is a common technology being utilized for absolute positioning in outdoor environments. However, the availability and accuracy of this system is highly reliant on the environment settings which entails that the service can only be used in areas with sufficient line-of-sight to GPS satellites. This limitation hampers the application of GPS inside buildings, dense urban environments (referred to as urban canyons) and tunnels. This thesis therefore aims at investigating localization algorithms being independent of any infrastructure setup in the operating environment.

The infrastructure-less position estimation requires maps of the environment serving as prior for matching sensor observations during operation. Even though there exist floor plans for buildings, it cannot be assumed that these contain all relevant structures of the environment in order to enable a correlation with sensors observations. One part of this work addresses the design of algorithms enabling to build maps of the environment using solely on-board sensors of a mobile robot. In particular we present a novel method for efficiently matching sensor data which enables the mapping of large-scale environments.

This thesis further investigates object recognition with application to mobile robotics. The presented methods are explicitly optimized in regards of performance in order to meet runtime requirements. It is shown how object recognition can be utilized for semantic map annotation. Semantic maps contribute to numerous tasks for mobile service robots which is demonstrated for an intralogistics application scenario.

The position estimation using only on-board sensors faces a major challenge compared to systems such as GPS, which is the robustness in dynamic environments. The state of the environment can drastically change over time compared to the one being initially captured. This might occur due to parked vehicles, humans or palleted goods in intralogistics. Therefore an algorithm particularly addressing changing environments is presented within this work. In contrast to the state of the art, the robustness is achieved by incorporating object recognition, semantically annotated maps and prior knowledge about the operating environment.

The implemented methods assume the availability of range measuring sensors being mounted to a mobile robot. The results presented in our experiments are obtained by using 2D range data of laser range finders or RGBD cameras. The latter make use of structured light to measure depth and pose the fundamental for the object recognition being utilized by our semantic map annotation and localization.

Acknowledgements

It is a couple of years ago when I first got in touch with mobile robots in a student project. They are cool, smart and can indeed revolutionize our every day life. They clean your carpet, serve coffee and save your life on the road. The combination of edge-cutting AI research and exciting moving machines has drawn my particular interest. It has been an indispensable companion in my PhD studies and has finally motivated this thesis.

At first I would like to thank my advisor Prof. Erik Maehle for employing me in his lab and giving me the chance to prepare this thesis.

Special thanks also to Prof. Hans-Joachim Böhme for providing the mentioned student project and introducing me to the field of mobile robotics. Hans, I thank you for your endless support in my studies and the first years as a PhD student in your lab. The exceptional freedom in the research allowed me to deepen my knowledge. I also thank Sven Hellbach, Frank Bahrmann, Peter Poschmann, Marc Donner, Richard Schmidt, Mathias Rudolph and Johannes Fonfara for sharing a great time in the lab.

I'm also very grateful that I got the chance to spend an amazing time with a great group at the University of Technology Sydney under excellent supervision of Alen Alempijevic. Alen, thank you a lot for the collaboration and your support before, during and after my stay in Sydney. Thanks again for the warm welcome and exchange to all of the lab.

Also I would like to thank my former colleagues of the ITI for the beautiful time. For the scientific investigations on coffee with Patrick Weiss, for sharing recent experiences of our little kids with Christopher Blochwitz, the helicopter flights in our testing warehouse with Uli Behrje and all those after-work sessions at the *Pool*. Thanks guys!

My very special thanks are due to my wife. Tine, thank you very much for your ongoing understanding, support and incentive to getting this thesis done. Thanks to my little one Nele for your long midday sleeps for writing and the memorable moments for switching off.

Thanks to my parents for their continuous support of myself and the encouragement for my scientific work.

Contents

Chapter 1.

Introduction

1.1. Motivation

Localization and mapping are one of the most important requirements for the navigation of mobile robots. The availability of maps allows robots to plan efficient paths to given goals avoiding static obstacles on the track. They further enable to assign relevant targets in the environment a global position with respect to the map which can be reapproached at any point in time, as for example, charging stations. Providing the robot can localize itself with respect to that map, we are able to estimate paths from our current location to any point on the map that can be physically approached. When following the path, the robot's location is continously tracked which in turn is utilized to estimate control parameters.

Given a map of the environment and the capibility of localizing with respect to it, a mobile robot is enabled to provide high-level services. In this way a service robot in a museum, for instance, can show around visitors guiding them from one exhibit to another. Thanks to the map and the knowledge about its location, it is able to replan routes to exhibits on-the-fly bypassing visitor crowds or blocked paths. Once a tour is finished, the robot can seamlessly return to a visitor meeting point.

Service robots have also become of interest for applications in intralogistics. While automated guided vehicles (AGVs) have been in use in warehouses for a long time, the launch of autonomous service robots has drawn particular interest in the recent years. Typically AGVs follow static routes being taught prior to the automatic operation or predetermined by magnetic tracks on the floor. The localization is carried out either using fiducials or magnetic point guidance. The effort for setting up either of these systems is enormous which makes switching to AGVs an expensive investment for companies. Thanks to the increasing performance of autonomous service robots, the operators of warehouses can look forward to significant cost reductions while simultaneously being provided machines with superior intelligence. They enable to avoid blocked rack corridors and bypass parked vehicles and dropped ware goods on the track. In addition to that, autonomous robots can be easily adapted to modifications in the environment.

The advances in mobile robotics have also pushed the development of autonomous robotic cars. The DARPA grand challenges in 2005 and 2007 have had a significant impact on the future developments of automated cars. Driver assistance systems for keeping the lane, observing blind spots and emergency braking are de-facto part of almost all upper-class vehicles and in increasing number of mid-range segment. It can be anticipated that the amount of features enabling automated driving in production vehicles will further increase in the next years. This is not solely an advancement of the driver's convenience but especially a benefical contribution to making driving safer and reducing evitable car accidents. The localization of on-road vehicles used to be done using GPS and in the case of automated driving rather using differential GPS (DGPS) due to its increased precision. One of the key lessons learned at the DARPA Challenges concerns the localization in urban environments based on GPS [24]. Since cities consists of a significant amount of narrow streets these sensors tend to fail making the positioning highly unreliable. GPS is also unavailable in tunnels, parking garages and partly fails in rural areas when pass-

ing forests and gorges. Even if these sensors are fused with other local cues, the pose estimates substantially deviates from the actual location which is crucial for automated driving. An alternative solution to GPS is essential for the introduction of autonomous robotic cars for on-road driving.

For all mentioned application scenarios of autonomous robots, there can be noticed an increasing interest in the recent years. Substantial progresses in robotics and machine learning research have rendered this possible. The transition from automated to autonomous systems is demanding but very important and significant step. It is one major part of the fourth industrial revolution and will also fundamentally change our traffic infrastructure. Autonomous robotic systems will have a deep impact on our society and everyday life.

Loalization and mapping are one of the key topics for mobile robotic systems. The autonomous behaviour requires a relatively high degree of independence of the environment. All applications share one common problem: the quest for infrastructure-free navigation. This means that the operation of a robot should not be reliant on the support of any external sources for navigation such as artificial landmarks, GPS or ultra-wideband positioning systems. The goal is to carry out the entire positioning solely based on on-board sensors. Thanks to existing technologies such as driver assistance systems and AGVs, a multitude of target platforms is already equipped with range measuring sensors such as stereo cameras, laser and radar sensors. Thus the infrastructure-free navigation actually does not require additional sensor setups for autonomous robots. Range measuring sensors are particularly well-suited as they enable robust solutions also in the presence of changing illumination conditions and direct sun exposure.

Omitting infrastructure-based navigation solutions allows a number of key benefits. First, the initial setup is extensively simplified since the environment does not have to be changed. This minimizes disruptions in the preparation of warehouses and the procurement of transmitters or reflectors resulting in a significant saving of expenses. For mobile robots aimed to provide services in a museum, shopping mall or for an individual fair, the costs for the initial setup are a crucial point. Expensive installation costs might entail decisions against the acquisition of mobile robots and thus also hamper the introduction of robots for novel applications. Also, it is rather unlikely that these systems are set up at larger scales when, for instance, considering autonomous driving. The costs for preparing and maintaining all tunnels, narrow streets and parking garages at country scale would be enormous making these systems rather less suitable for this scenario. Infrastructure-free navigation is indispensable for the mentioned and also many other applications of robotic systems. It does not only allow to substantially reduce costs but also provides robots with a higher level of autonomy.

The novel achievements in robotics are not obtained for free. Infrastructure-free navigation raises new challenges that have not been existing to this extent before. Positioning used to be a closed-form solution being derived from the relative constellation of observed satellites or artificial landmarks. Using solely on-board sensors for navigation, in contrast, requires the robot to autonomously recognize places in the environment using naturally occurring landmarks. A robot constantly incorporates new measurements,

recognizes previously visited places and keeps track of its position in order to generate maps which is well known as the simultaneous localization and mapping (SLAM) problem. The maps achieved in this way serve as input for the navigation in a designated environment. While SLAM can generally be applied straightforward to limited lab sizes, the application in large-scale environments and in particular using cost-effective sensors such as cameras still raises a number challenges for research.

Based on the mentioned application scenarios and requirements, we can summarize the following key questions for localization and mapping using autonomous robots:

Q 1: How can places be efficiently recognized using range data?

Q 2: How do we build prior maps of large-scale environments?

Q 3: How do we handle uncertainties due to repetitive structures?

Q 4: How can object recognition contribute to robotic navigation?

Q 5: How can we benefit from platform and environmental constraints?

Q 6: How can map-based localization be enabled in the presence of high dynamic changes?

This thesis provides answers to the above mentioned questions presenting insights into novel solutions for mobile robotics.

1.2. Contributions

Our work presents methods for localization and mapping of mobile robots in changing, large-scale environments without using infrastructural conditions.

In particular we present:

- GLARE and GRAPE which are two novel algorithms for place recognition using range data
- A mapping framework enabling both high performance and precision by post processing sensor data
- An efficient solution to object recognition achieved by exploiting environmental and platform constraints
- An algorithm for generating semantic maps
- A novel approach for map-based localization in changing environments

1.3. Publications

Parts of this thesis were published in international journals and conference proceedings. The remainder of this section provides a chronological overview of publications which are incorporated by this thesis. In addition to that, we list those articles which were written within the PhD studies but are not reported on in the context of this thesis.

Journal Articles:

- **Marian Himstedt**, Erik Maehle: *Online Semantic Mapping of Logistic Environments using RGB-D Cameras.* International Journal of Advanced Robotic Systems, 2017

Conference and Workshop Contributions:

- **Marian Himstedt**, Erik Maehle: *Semantic Monte-Carlo Localization in Changing Environments using RGB-D Cameras.* European Conference on Mobile Robots (ECMR), Paris, France, 2017

- **Marian Himstedt**, Erik Maehle: *Camera-based Obstacle Classification for Automated Reach Trucks using Deep Learning.* International Symposium on Robotics (ISR), Munich, Germany, 2016

- **Marian Himstedt**, Erik Maehle: *Geometry matters: Place Recognition in 2D Range Scans using Geometrical Surface Relations.* European Conference on Mobile Robots (ECMR), Lincoln, UK, 2015

- **Marian Himstedt**, Jan Frost, Sven Hellbach, Hans-Joachim Böhme, Erik Maehle: *Large Scale Place Recognition in 2D LIDAR Scans using Geometrical Landmark Relations.* Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, United States, 2014

- **Marian Himstedt**, Sabrina Keil, Sven Hellbach, Hans-Joachim Böhme*: A robust graph-based framework for building precise maps from laser range scans.* Proceedings of the Workshop on Robust and Multimodal Inference in Factor Graphs, IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 2013

This thesis does not report on:

- Sven Hellbach, **Marian Himstedt**, Frank Bahrmann, Martin Riedel, Thomas Villmann, Hans-Joachim Böhme: *Find rooms for improvement: Towards semi-automatic labeling of occupancy grid maps.* 21st International Conference on Neural Information Processing (ICONIP), Sarawak, Malaysia, 2014

- Sven Hellbach, **Marian Himstedt**, Frank Bahrmann, Martin Riedel, Thomas Villmann, Hans-Joachim Boehme: *Some room for GLVQ: Semantic Labeling of occupancy grid maps.* Proceedings of the Workshop on Self-Organizing Maps, Mittweida, Germany, 2014

- Sven Hellbach, Frank Bahrmann, Marc Donner, **Marian Himstedt**, Mathias Klingner, Johannes Fonfara, Peter Poschmann, Richard Schmidt, Hans-Joachim Boehme: *Learning as an essential ingredient for a Tour Guide Robot.* Proceedings of the Workshop - New Challenges in Neural Computation 2013 (NC2 2013), pp. 53-60, Machine Learning Reports, Saarbrücken, 2013

- Marc Donner, **Marian Himstedt**, Sven Hellbach, Hans-Joachim Böhme: *Awakening history: Preparing a museum tour guide robot for augmenting exhibits*. Proceedings of the 6th European Conference on Mobile Robots (ECMR), Barcelona, Spain, 2013

- Sven Hellbach, **Marian Himstedt**, Hans-Joachim Böhme: *What's around me: Towards Non-negative Matrix Factorization based Localization.* Proceedings of the 6th European Conference on Mobile Robots (ECMR), Barcelona, Spain, 2013

- **Marian Himstedt**, Alen Alempijevic, Liang Zhao, Shoudong Huang, Hans-Joachim Böhme*: Towards robust vision-based self-localization of vehicles in dense urban environments.* Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 2012

- Hans-Joachim Böhme, Sven Hellbach, Frank Bahrmann, Marc Donner, Johannes Fonfara, **Marian Himstedt**, Mathias Klingner, Peter Poschmann, Mathias Rudolf, Richard Schmidt: *Assistance Robotics: A survival guide for real world scenarios.* Poster and Demo Track of the 35th German Conference on Artificial Intelligence (KI), Saarbrücken, Germany, 2012

- **Marian Himstedt**, Sven Hellbach, Hans-Joachim Böhme*: Feature extraction from Occupancy Grid Maps using Non-negative Matrix Factorization.* Proceedings of the Workshop - New Challenges in Neural Computation 2012 (NC2 2012), Machine Learning Reports, Graz, Austria, 2012

1.4. Collaborations

Parts of this thesis are results of collaborations with other researchers. A detailed overview for the corresponding chapters is provided in the following:

- Chapter 3: The novel place recognition algorithms being introduced is a joint work with Jan Frost, Sven Hellbach, Hans-Joachim Böhme and Erik Maehle. Jan Frost contributed to parts of the C++ implementation of the algorithms and the experimental evaluations. The remaining authors acted as advisors and supported by

means of fruitful discussions. The general concept, algorithm design, the majority of the software implementation and experimental investigations were carried out by the author of this thesis.

- Chapter 4: The SLAM framework which is described in this chapter was mainly designed and implemented by the author of this thesis. An initial solution was developed at the Cognitive Robotics group at the HTW Dresden with advisory support of Sven Hellbach and Hans-Joachim Böhme. Sabrina Keil also contributed in the post-processing of the experimental data obtained in the museum *Exhibitions of Technology Dresden*. The framework was substantially extended and refactored during the first author's PhD studies at the University of Lübeck. However, the framework does not share algorithms or software with *UzL-ITI-SLAM* [56, 70] which has been established by Jan Helge Klüssendorff and Jan Frost in parallel.
- Chapters 5-7: The algorithms and implemented software of Part II are results of investigations which were advised by Erik Maehle and solely carried out by the author of this thesis. The robot platform which is used in the experiments was established in close collaboration with partners of the research project *FTF out-of-the-box* A.1. The platform-specific software on the mobile robot was developed in cooperation with Ulrich Behrje who also implemented the odometric motion estimation being utilized within the experimental evaluation of Chapters 5-7.

1.5. Structure

The thesis is organized as follows.

Chapter 1 motivates the topic of this thesis and summarizes the aims and key contributions.

Chapter 2 outlines the methodological background of this thesis. It can easily be skipped if the reader is certain with the general concepts of place recognition, SLAM and map-based localization.

Part I of our thesis addresses the use of generic approaches to localization and mapping.

Chapter 3 introduces a novel concept utilizing geometrical relations for matching places and landmarks. Next to the algorithmic details of our algorithms, we present experimental results for publicly available datasets.

Chapter 4 shows how the presented place recognition algorithms can be integrated into a graph-based SLAM framework.

Part II of our thesis addresses the use of semantic perception and representations for localization and mapping.

Chapter 5 describes a system for object recognition. We combine concepts of deep learning and geometric object descriptors to establish an efficient algorithm for robotic navigation.

Chapter 6 unites our SLAM framework and object recognition and efficient inference methods to generate semantic maps.

Chapter 7 demonstrates the benefits of semantic perception for Monte Carlo localization in changing environments.

Chapter 8 summarizes our novel achievements. Before concluding the thesis, we discuss the contribution of the presented solutions and motivate future work.

Chapter 2.

Background

2.1. Map-based Localization

The goal of map-based localization is to estimate the pose of the robot with respect to an a-priori given representation of the environment. Therefore we have to match the current sensor observation with the prior map. As a result we obtain a robot pose in the coordinate frame of the map. The Monte-Carlo localization (MCL) is the most common algorithm in this field [160]. MCL utilizes a particle filter and is the default algorithm in the middleware ROS. The key idea of MCL is to approximate the robot pose by a set of particles. Each particle expresses a state of the robot with the particle state $\mathbf{x}_t^k$ being a putative pose in the prior map, $\mathbf{x}_t = (x, y, \psi)$ with $\mathbf{x} \in SE(2)$. The particles are further assigned individual weights w^k. The major stages of MCL are the prediction, the update and the resampling which are further detailed in the following.

Prediction. The state $\mathbf{x}_t'$ at time t is predicted given the previous state $\mathbf{x}_{t-1}$, odometry readings and a motion model. The motion carried out by the robot along with some random noise is incorporated by the particle set. As a result we obtain the predicted states $\mathbf{x}_t'$ for each particle k.

Update. Each sensor observation z_t is incorporated and triggers an update of the particle filter. We evaluate the likelihood for $p(z_t|x_t^k, m)$ given the state of particle k. Literally speaking, we are estimating the likelihood of making the observation z_t given the state x_t^k and the map m. A particle's weight w^k is set according to the observation likelihood, thus we can find the following relation $w^k \propto p(z_t|x_t^k, m)$. In the literature there can be found different observation models [160]. For proximity sensors the likelihood field model is commonly used as it provides significant advantages in regards of performance [160]. We can summarize this as follows:

1. The range readings of observation z_t are projected into the map coordinate frame given the particle's pose $\mathbf{x}_t^k$ and the fixed angular orientation of the individual measurement beams resulting in a set of endpoints in the map coordinate frame.

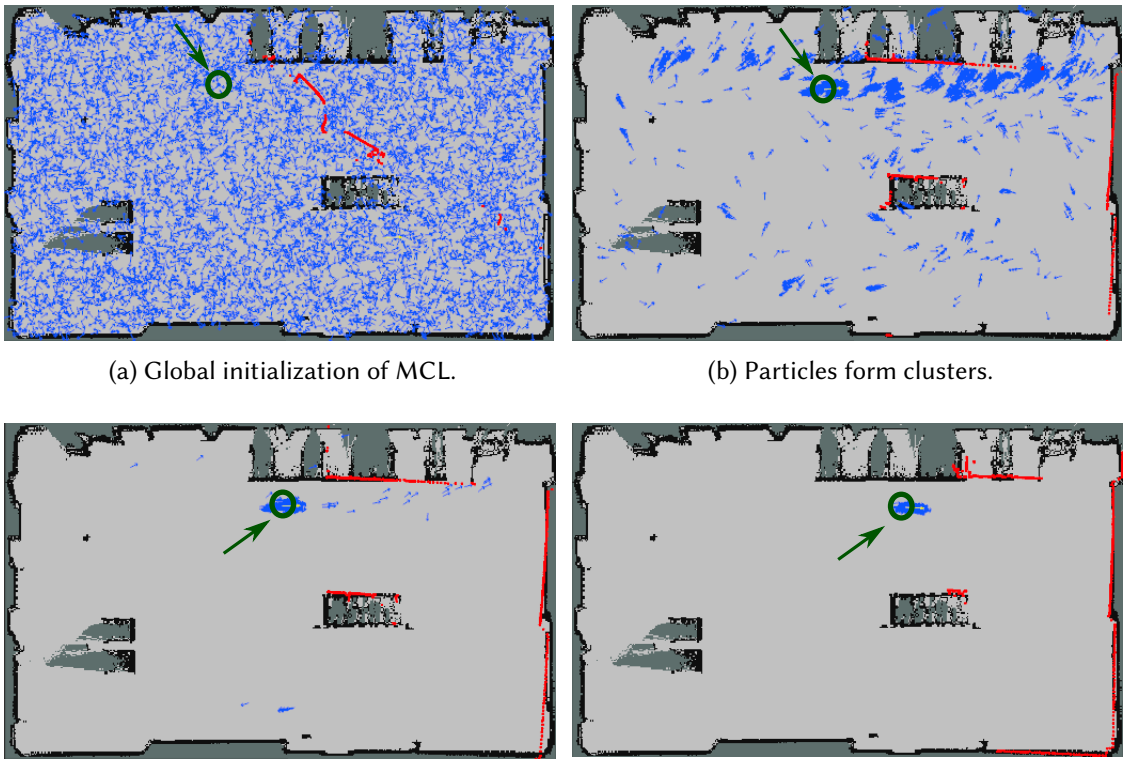
2. The nearest grid cells in the map are looked up for all endpoints.

3. The distance values for all map cells being hit are obtained from the likelihood field.

4. The estimated probabilities $p_{hit}(z_t|x_t^k, m)$ about z_t hitting a map cell m_i are mixed with additional probability distributions accounting for measurement uncertainty and random noise.

The distance values (3) can be calculated offline once and supplied by means of lookup tables at runtime. This is the main advantage of the endpoint-based likelihood field model compared to other such as the raycasting model [160].

Resampling. After a number of update iterations the probability mass accumulates for a limited number of particles. The key idea is to increase the particle density in areas that are more likely to represent the high-confident hypotheses about the robot pose. Likewise the density can be reduced for less likely states. Thus we draw new samples around particles with higher weights which consequently speeds up the convergence allowing to represent high-confident robot pose hypotheses more accurately thanks to the higher sample density.



(a) Global initialization of MCL.

(b) Particles form clusters.

(c) A large particle cluster remains.

(d) Particle filter has converged.

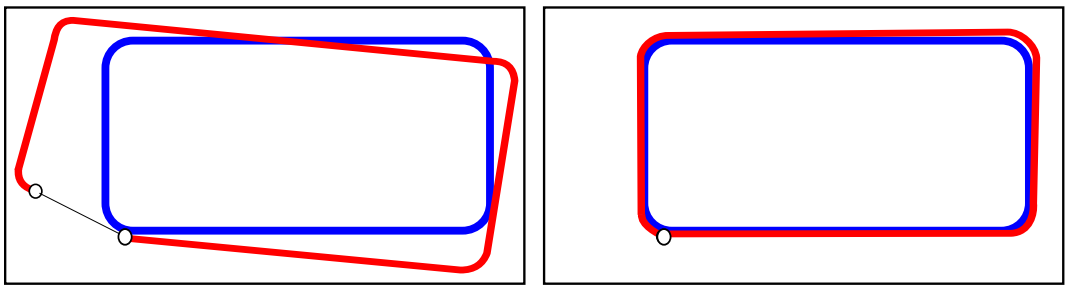
Figure 2.1.: **MCL**. This figures illustrates the absolute pose estimation for a mobile robot using Monte-Carlo localization (MCL). The particles (blue), the true pose (green circle) and the laser endpoints (red) projected into the map coordinate frame given the current pose estimate are shown. The particles are sampled around all free cells of the map to initialize the particle filter (a). Having incorporated several observations, the particles start forming clusters (b). In (c) one large cluster with a high density and a few separate particles remain. In (d) the filter has converged and tracks one hypothesis. The projected laser points fit well onto the map cells.

The entire MCL algorithm is illustrated by Fig. 2.1. For a more detailed derivation of the Monte-Carlo localization and the utilized models, the reader is referred to [160].

2.2. Simultaneous Localization and Mapping

Map-based localization requires a representation of the environment to estimate a robot's pose. Since these prior maps are not always available, it might be necessary to generate them prior to any autonomous operation.

If an accurate range sensor and an external reference system providing absolute poses of the robot are available, a map can be built straightforward by simply projecting the measurements into a common global coordinate system. However, in the absence of such a system, a mobile robot has to concurrently estimate its pose and a map of the traversed environment. This is known as the Simultaneous Localization and Mapping (SLAM) problem. It explicitly deals with the uncertainty of the robot localization in an a-priori unknown environment. The uncertainty increases with the distance traveled by the robot. It can be reduced if the robot re-traverses parts of the environment it has already seen before which is referred to as loop closures. The detection and incorporation of loop closures in order to minimize pose errors and generate consistent maps is a key topic in SLAM (see also Fig. 2.2).



(a) SLAM trajectory before loop closure. (b) SLAM trajectory after loop closure.

Figure 2.2.: **SLAM**. This figures shows an example for simultaneous localization and mapping. Fig. (a) illustrates a robot traversing an a-priori unknown environment and estimates its pose (red). The pose error increases with the distance traveled compared to the ground truth trajectory (blue). The uncertainties can only be reduced by loop closure detections (circles at begin and end of path) which is demonstrated in Fig. (b). This is why loop closures are fundamental for SLAM.

SLAM can be used for numerous applications, the most important ones for mobile robots can be summarized as follows:

1. Map building in the absence of a position reference system
2. The localization of a mobile robot in an a-priori unknown environment (without initial map)
3. Live-long mapping of environments

4. Fusion and smoothing of absolute reference and onboard sensor measurements to increase the accuracy of the estimated trajectory and map

The majority of autonomous mobile robots operating in indoor environments ranging from airports [165] and warehouses [137] to shopping malls [65] and museums [25] make use of category (1) and subsequently apply map-based localization. Robotic systems for search and rescue in disaster environments [16], markerless augmented reality systems ([92]) are the main representatives for category (2). Systems based on (3) are often motivated for the use in highly dynamic environments as described in-depth in [46, 102]. Category (4) particularly addresses autonomous on-road driving [107, 161] and unmanned aerial vehicle applications [84]. This thesis focuses on the application of SLAM for category (1). There exists a number of methods solving the SLAM problem. An overview of the fundamental algorithms is provided by [62, 160]. We will highlight a graph-based representation of SLAM in Chapter 4.

2.3. Place Recognition

Place recognition describes the problem of associating the current sensor observation with all previous ones in order to decide whether these are originated from the same place in the environment.

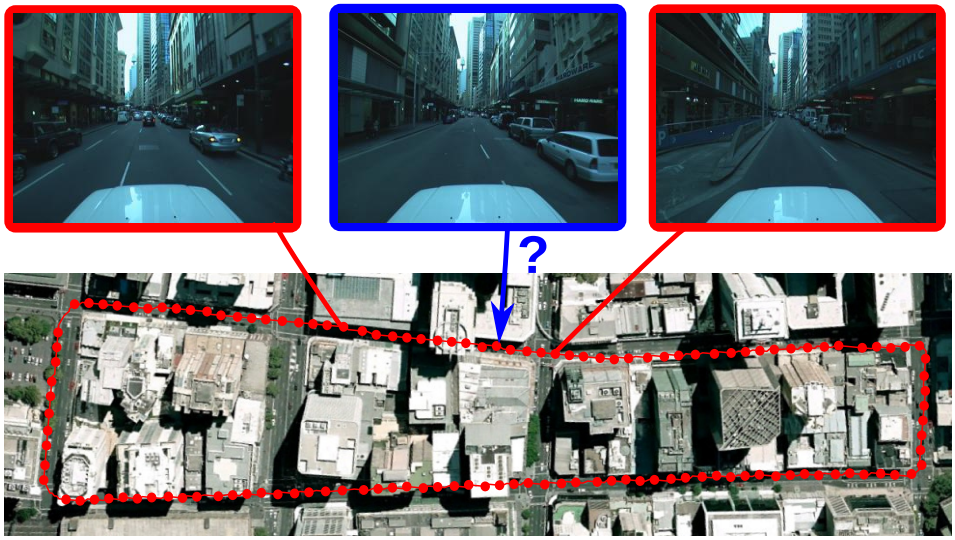


Figure 2.3.: **Place recognition**. This figures shows a topological graph (red) with each node being assigned a camera keyframe. The graph is overlayed on a satellite image. A mobile robot captures a further image (blue) which is utilized by the place recognition algorithm to find corresponding images. A pose can be estimated with respect to these and subsequently with respect to the map.

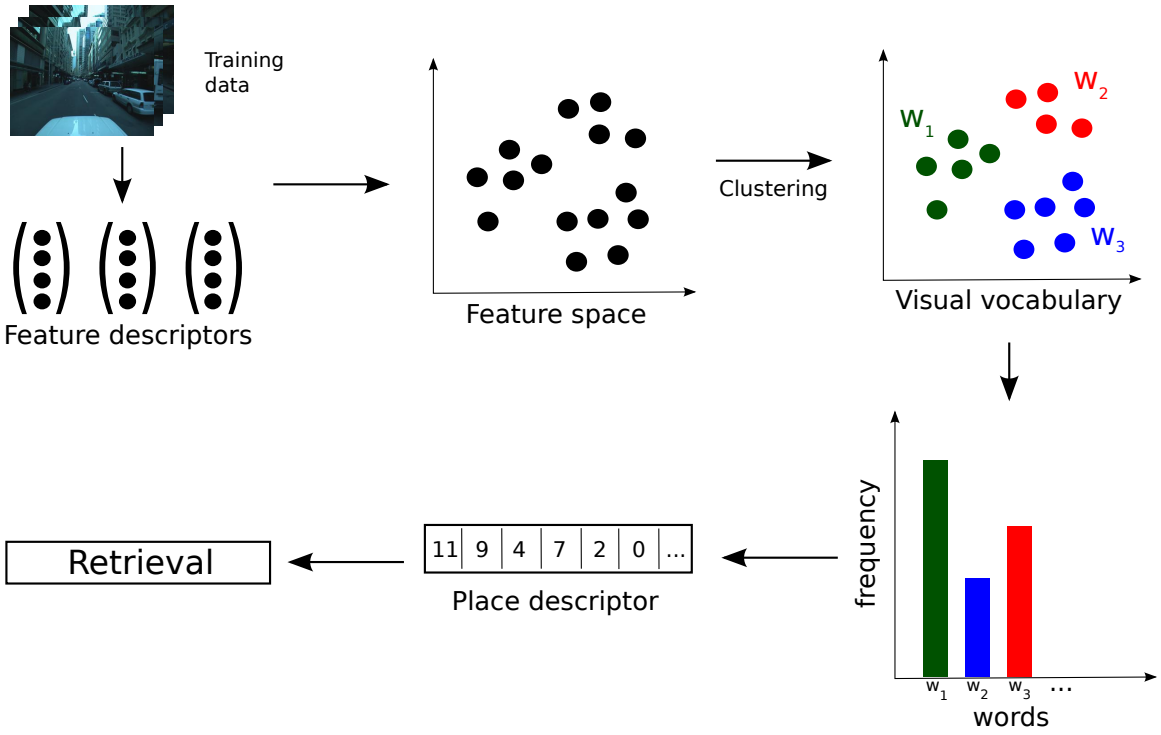


Figure 2.4.: **BOW model**. This figures illustrates the bag-of-words model which is a common method to find compact representations of sensor data. It enables efficient retrieval algorithms and hence a significant performance speed-up for place recognition.

Literally speaking, it actually implements a retrieval matching the appearance of input sensor data to a database of collected frames. The successful recognition of a place does not actually imply that the algorithm returns an absolute position on a metric map. It rather provides indices to stored images or other sensor data that can be associated with it. Subsequently, a relative pose can be estimated for the input data with respect to the matching sensor data (e.g. an image or range scan). An absolute pose with respect to the map can be calculated based on this relative pose and the pose of the matching frame.

The number of places and associated sensor impressions increase rapidly with the distance traveled by a mobile robot. Thus it is important to find compact representations and efficient retrieval algorithms in order to enable a mobile robot to incorporate this information while executing tasks, at a high frequency. This runtime requirement typically makes feature extraction and description or compression methods indispensable as a direct matching of sensor data is too complex. The bag-of-words model is commonly used to generate compact representation of sensor data (see also Fig. 2.4) [37, 164].

Place recognition is useful for a number of applications. The most common is the detection of loop closures in SLAM as already demonstrated by Fig. 2.2 [37, 164]. It allows an efficient retrieval of potential loop closure candidates while traversing the environment. In addition to that, place recognition can speed up the initialization of other map-based

localization systems such as MCL and can be run in parallel to recover from failures. A rather less common but still useful benefit of place recognition is its ability to align datasets captured at multiple times [22].

The main difference of place recognition compared to map-based localization addresses the expected efficiency. Place recognition algorithms commonly make use of feature extraction, efficient data structures and fast retrieval methods resulting in a significant performance boost ([73, 164]). The tracking of a pose is a more suitable task for map-based localization algorithms such as MCL though.

Part I.

Generic Representations

Chapter 3.

Place Recognition using Geometrical Relations[1]

[1] Parts of this chapter have already been published in [73, 77]

3.1. Motivation

Visual solutions for place recognition have been exhaustively investigated in the recent years. The utilization of range data for this task is rather less common, but provides a number of key benefits: First it is, depending on the physical model of the range sensor, usually invariant to lightning changes. Second, 2D laser range finders can be found on the majority of mobile robotic systems since they are often demanded by law for safety reasons. On the other hand, they provide less information than visual sensors due the limited vertical field of view which has to be taken into account. Exhaustive one-by-one matching of individual laser scans is impractical in terms of time complexity when looking at large scale environments making descriptive place signatures and efficient retrieval algorithms indispensable.

The state of the art in place recognition using range data is inspired by algorithms utilized in computer vision. Usually a number of interest points serving as landmarks is extracted from the range scan. A descriptor for each landmark is generated by incorporating the appearance of the surrounding neighborhood. As usual in bag-of-words (BOW) techniques these descriptors can be quantized to words enabling laser scans to be represented by histograms of word occurrences. While achieving promising results in visual applications [37], BOW performs rather poor in conjunction with 2D range data [164], particularly when looking at outdoor environments. The retrieval is conducted by matching sets of landmark descriptors associated with individual range scans. The algorithm which will be presented in this chapter differs from the state of the art since we incorporate the geometrical relations of landmarks rather than describing local characteristics around landmarks. The most obvious illustration is given if one considers a mobile robot operating in a forest. A typical place recognition algorithm would extract points of interest which here most probably refer to trees. Since there is a large number of visually similar trees, one can imagine that a retrieval based on tree descriptors poses a challenge. However, it can be observed that the spatial configurations of trees in forests provides a contributive feature for associating places thanks to the rather random positions of trees in the nature. For indoor environments this phenomenon is not as obvious, however, these properties can also be observed in a slightly mitigated manner.

In this chapter we will show that the transition from appearance-based to geometrical features outperforms the state of the art in place recognition using 2D range data for both, outdoor and indoor environments.

3.2. Related work

Place recognition has been investigated over longer periods of time with the origins going back to the work of Engelson [47] and Aycard et al.[11] in the early 1990s. Initially place recognition has mainly been utilized for loop closure detection in SLAM. Several sensor types have been integrated for this task, with laser range finders being of particular interest in the beginning.

This section gives an overview of state of the art dealing with place recognition. At first we survey vision-based algorithms for this application. Subsequently the relevant work in this field utilizing geometric constraints is further investigated. This commences our transition to range-based methods being discussed in detail. Our approach is justified against prior work in this field with two highly related algorithms being outlined in detail.

Vision-based place recognition

In [11, 47] the first camera-based approaches were presented which was motivated by the first algorithms for visual navigation of mobile robots. A survey of place recognition for loop closure detection in SLAM is given by [169]. While efficient retrieval methods and data structures have already been available, the computer hardware and the runtime requirements of robots limited the amount of suitable feature detectors. The processing of camera images demanded high computational loads hampering their use for applications with increased runtime requirements.

Initially visual place recognition was conducted by generating global descriptors of images. Ulrich et al. utilize color histograms extracted from omni-directional camera images [166]. Also, Lamon et al. uses this kind of image source and concatenates local features such as edges and corners into one place descriptor [103].

Lowe et al. presented scale invariant feature transform (SIFT) which has become of high popularity in the computer vision and robotics communities. SIFT comprises a local feature detector and a high-dimensional descriptor which was initially motivated for object recognition tasks. Se et al. for the first time utilize SIFT for mobile robot localization and mapping reporting promising results. Many other researchers also adapted it for this tasks in the following years [8, 76, 95]. Even though SIFT improved the performance in object and place recognition, the runtime again rendered applications with high performance requirements difficult. Yonglong et al. proposed an implementation for GPUs which substantially increased the processing time and thus contributed to a rapid dissemination of SIFT [171]. In the following years there have been published several alternatives to SIFT achieving better results for specific conditions or better performance. Speeded-up robust features (SURF) became a famous competitor enabling results comparable to SIFT but with better performance also on CPU architectures [13]. Svab et al. presented a hardware accelerated version based on FPGAs allowing SURF to be used on low-power robotic systems [158]. Calonder et al. made their algorithm BRIEF publicly available [26]. In contrast to other approaches, it allows high performance thanks to the use of binary descriptors which can be efficiently matched. Rosten et al. presented FAST which consists of a detector being trained on corner-like features using machine learning [138]. Similar to BRIEF, FAST enables images to be processed at a relatively high frame rate. For place recognition, researchers have begun mixing feature detectors and descriptors of different methods [70]. Depending on the application's goal, one can, for example, generate a high number of features with worse repeatability at high frame rates or fewer, more stable ones at a lower frame rate. Also, the utilized feature descriptors perform differently in regards of invariances to illumination, perspective and scale. The run time,

stability and matching performance can be adjusted to environment and robot conditions. The majority of the mentioned image features were shown to be capable for place recognition at larger scales ([38, 76]).

Cummins and Newman demonstrated the use of SURF and the probabilistic framework FABMAP for a distance of about $1000km$ at a scale of $100km\,x\,100km$ in a UK road network [38]. The place recognition algorithms based on local image features such as SIFT, SURF and BRIEF commonly assume that illumination conditions do not globally change to much. This means that the place recognition and hence also the mapping of large scale environments using these features is limited to either short periods of time or to multi-session mapping being continued at similar lightning conditions. Researchers have demonstrated that the scale of the environment is no longer the main issue ([38]). However, the recognition of places in the presence of significant changes still remains an open challenge in robotics. A notable amount of research being dedicated to the long-term application of place recognition has been established. Johns and Yang investigate the generation of vocabularies consisting of quantized feature descriptors collected at different times of a day [88]. The variances in appearance of individual features are captured in order to obtain descriptors being maximally stable. Similarly Carlevaris-Bianco and Eustice propose to explicitly normalize descriptors generated at varying times [28]. Churchill and Newman also capture multiple appearance settings of the environment but rather than normalizing descriptors they store all descriptor sets for each illumination and seasonal condition [32]. As a result the authors obtain a concatenated map with each place being assigned experiences in a range of $5 - 30$ views for the reported experiments. The goal of the mentioned approaches is to get local features such as SIFT to work in changing environments. Since these features are off-the-shelve not sufficiently invariant to this extent of change, authors either generate multiple vocabularies or apply different strategies of normalization.

In recent years authors have also investigated alternatives to local image features for place recognition. The quest for solutions in changing environments has revived global image descriptors which had also been a common method before the availability of SIFT and SURF. In this line Sünderhauf et al. propose BRIEF-GIST which generates a binary descriptor of the entire image. Liu and Zhang propose an image descriptor for place recognition based on Gabor filters with varying orientations and frequencies [109]. Milford and Wyeth introduce SeqSLAM matching image sequences rather then single images [117]. Their approach has significantly pushed the state of the art in visual place recognition and introduces a new key reference work for other researchers. The authors report on successful correspondences being found across extensive changes in daylight, weather and season. The correlation is carried out based on relatively long series of low-resolution image patch representations. Sequential information has also been processed prior to SeqSLAM, as for instance by particle filters [53, 54]. However, these methods constantly incorporate information by means of re-sampling hypotheses with respect to previous filtering steps. The key idea of utilizing image sequences has been adapted in a number of successive methods [9, 68, 122, 167].

Neubert et al. introduce an algorithm being able to model seasonal changes [125]. Having completed a prior training stage, it allows to predict the appearance of a place for example in the summer time given an image captured in the winter time and vice versa. The authors utilize superpixel-based segmentation with each extracted segment being used to learn a cross-season vocabulary. Sünderhauf et al. further motivate the use of convolutional neural networks (CNN) for place recognition [156]. Similarly to [125], the authors extract small image segments with each being subsequently passed to a CNN. The outputs obtained by the high-level convolutional layers are utilized as feature descriptor which in-turn serve as base for associating places [156].

The recurrence of global image or region descriptors has been shown to be superior to local image features in the presence of changing environment conditions [68, 115, 117, 122]. However, they have also entailed a new challenge if the algorithms are meant to be integrated into metric localization and mapping frameworks: That is, the estimation of a relative pose of two corresponding places. Metric positions can be obtained straightforward for local features given calibrated cameras. However, if features cannot be associated to edge-like regions in an image, it is usually more difficult to exactly localize them in other images. Milford et al. demonstrate an increased invariance in pose which, however, is rather limited to lateral shifts of the camera origins [116, 133]. Larger perspective changes in the viewpoint can cause significant variances in appearance for larger image regions. The majority of the state-of-the-art approaches based on sequence processing report results obtained from on-road scenes since this is of particular interest in industry and science. While environmental conditions have a significant impact which makes image-based retrieval difficult, the differences in the pose are rather present in terms of lateral shifts which relaxes the requirements for relative pose estimation [115, 133]. Indoor environments, in contrast, usually provide larger variances in the viewpoints due to increased open space and fewer limitations in terms of drivable areas.

The research results for visual place recognition that have been published in the recent years are substantive in terms of robustness and scalability. Further significant advances can be expected through the recent use of deep learning in conjunction with place recognition as already motivated by [116, 156].

Spatial relations in vision-based place recognition

The use of geometric information for recognition applications has recently drawn attention again within computer vision research. For example, Johns and Yang model discretized bearings and distances of visible landmarks within the image coordinate system [88]. Paul and Newman also include spatial constraints of co-occuring features within a graphical model [132]. The implemented system FABMAP-3D is also utilized for place recognition. FABMAP-3D differs from other approaches such as [88] in that it does not stick to the image coordinate system but instead uses metrically scaled depth values obtained from a stereo rig. The graph's nodes are quantized into visual words and the edges are distributions over distances to adjacent visual features. The approach enables an efficient geometric verification by introducing distances into the retrieval. However, due to

the limitation of incorporating only adjacent features, FABMAP-3D does not exploit the benefit of pairwise distances for *all* co-occurring visual features which is justified by the high complexity of $O(N^2)$ constraints.

Another approach which was presented by Clemente et al. utilizes relative distances of co-occurring landmarks in order to evaluate the matching quality of submaps of the environment [36]. However, the complexity of the matching procedure limits its use for rather small numbers of submaps. Finman et al. proposed physical words to encode spatial relations of objects segmented from RGB-D images [50]. Rather than using appearance descriptors for place recognition, the authors build dictionaries of physical words describing discrete spatial configurations of objects in a scene.

Place recognition based on laser range scans

There can be found less prior work on place recognition using 2D laser scans than for camera images. For example Bosse and Zlot investigate different combinations of algorithms for the detection and description of local features being extracted from multiple scans and fused into submaps [21]. The authors propose a recognition system which matches landmark descriptors based on nearest neighbour search. Each corresponding set votes for the associated submap. The top matching candidates are subsequently checked for geometric consistency using projection histograms and (dense) scan matching based on ICP.

In the work of Granström et al. multiple features such as curvature and average range are evaluated by an Adaboost classifier [59]. Their approach enables to detect corresponding scans but at the cost of significant computational expenses as demonstrated in [164].

The aforementioned papers [21, 59, 164] detect features in the origin laser range data which are subsequently used as keypoints. As an alternative to this, there exists approaches that make use of occupancy grid maps for place recognition and localization [72]. Sequences of range scans are incorporated into local maps being built incrementally. A fusion of multiple sensor types is also enabled thanks to the use of generic occupancy grids. In [72], the authors specifically utilize non-negative matrix factorization (NMF) to extract geometric primitives by combining adjacent grid cells. In this way, the algorithm is able to detect features (e.g. corners, t-junctions) for specific environment types. Here, the memory consumption of occupancy grid maps for large-scale environments can be significantly reduced since local maps can be represented as a distribution of geometric primitives. A similar method which is used for appearance-based place recognition in large databases.

Kosnar et al. propose to use shape matching methods from computer vision such as FFT and Ring Projection Histograms for place recognition showing promising results for indoor environments [96]. Similar to the NMF-based approach, the authors rely on the entire range scans rather than landmarks enabling improved recognition rates for indoor environments with a high amount of structureless walls.

Next to the presented work, there also exist approaches utilizing 3D range data in the literature. For example, Steder et al. propose to detect local points of interest in

range images [149]. These are subsequently used to generate descriptors incorporating the region around keypoints. The descriptor vectors are quantized and matched by the use of bag-of-words (BOW) models which is also common in appearance-based place recognition [37, 57].

In [112], Magnusson fits normal surface primitives based on range data. The elliptic shapes of these primitives are classified with the results being concatenated into a feature vector. The recognition of places is carried out based on the distribution of primitive types being present. The spatial configuration of surface primitives is not considered.

Geometrical FLIRT Phrases

Tipaldi et al. presented Geometrical FLIRT Phrases (GFP) which is highly related to our work [164]. For the sake of completeness it is specifically detailed in the following. The origin of GFP can be ascribed the initial contribution FLIRT published in [162]. It consists of a library providing several methods for feature detection and description in 2D range scans. A core contribution for feature-based scan matching is given by a beta grid descriptor capturing the local appearance around points of interest [162]. The authors demonstrate that this kind of descriptor outperforms the shape context descriptors being commonly used in computer vision [96]. Inspired by the visual recognition systems such as [38], the feature descriptors are quantized into words to obtain compact BOW models. Similar to [38], this assumes learning a prior vocabulary [164]. The experimental analyses of Tipaldi et al. show that common BOW-based recognition using FLIRT descriptors achieve an insufficient performance, particularly in outdoor environments [164]. Based on prior investigations of Zhang et al. [173], the authors present Geometrical FLIRT phrases (GFP) [164] extending the existing FLIRT library by sparse geometrical verification. To be more precisely, GFPs allow to preserve spatial constraints by matching feature sets in a cyclic order. It can be shown that GFP achieves better results than BOW [164]. However, the performance of GFP in larger environments with repetitive structures is still limited due to numerous potential matches that have to be considered [164]. Although the feature sets are matched in a cyclic order, the distinctive features of spatial characteristics such as distances and bearings are not incorporated within the retrieval.

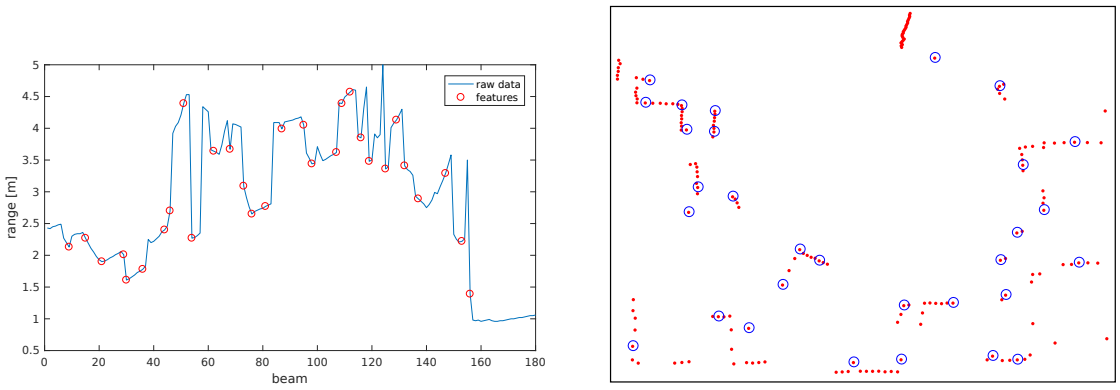
Summary

The majority of the presented work for place recognition using range data uses local interest points with quantized descriptors and requires vocabularies or prior training stages. None of the aforementioned methods working on range scans explicitly utilizes geometrical properties of co-occurring features. Particularly the limited distinctiveness of 2D range measurements motivate further investigation of the spatial configurations of observed landmarks by incorporating bearings and displacements of all co-occurring landmarks.

3.3. Geometrical Landmark Relations

In the following sections we will present a novel algorithm bridging the aforementioned gap in place recognition by introducing **G**eometrical **La**ndmark **Re**lations (GLARE).

3.3.1. Feature Detection



(a) Features extracted from range scan.

(b) Features and scan in 2D

Figure 3.1.: Illustration of the extraction of features from 2D range scans. Figure (a) shows the range measurements as a curve (blue) and local extrema being detected (red points). Figure (b) shows a 2D projection of range measurements and the extracted features.

Given an input 2D range scan, we extract features which serve as landmarks. Inspired by Tipaldi et al. we utilize points of high curvature since these are demonstrated to provide promising results when working with 2D range measurements [162]. The input range scan is assumed to be a one dimensional curve $g(b)$ which is projected into a multi-scale representation $G(b;t)$:

$$G(b;t) = (K_t * g)(b) \tag{3.1}$$

where t denotes the scale and K_t the utilized kernel to smooth the range signal. In our implementation we use a Gaussian kernel with standard deviation t :

$$K_t(x) = \frac{1}{\sqrt{2\pi}t} e^{-\frac{x^2}{2t^2}} \tag{3.2}$$

The kernels being used to smooth the input data are normalized in order to be ensure invariance with respect to the sampling density. Interest points in terms of peaks are detected for each scale t. The extracted points are local minima and maxima of the second derivative constructed from the smoothed curve $G(b;t)$:

$$\nabla^2 G(b;t) = (D_2 * G)(b;t) \tag{3.3}$$

$$(D_2{*}G)(b;t) = G(b-1;t) - 2G(b;t) + G(b+1;t) \tag{3.4}$$

The scale space theory is inspired from visual features such as SIFT [110] and SURF [13]. It allows to detect peaks at different resolutions which is beneficial for reducing the viewpoint variance of the extracted landmarks and enable more robustness in the presence of noisy range data.

The feature detection is illustrated by Figure 3.1. Based on this feature extraction we are given a set of N landmarks for the k-th range scan.

3.3.2. Encoding Spatial Relations of Landmarks

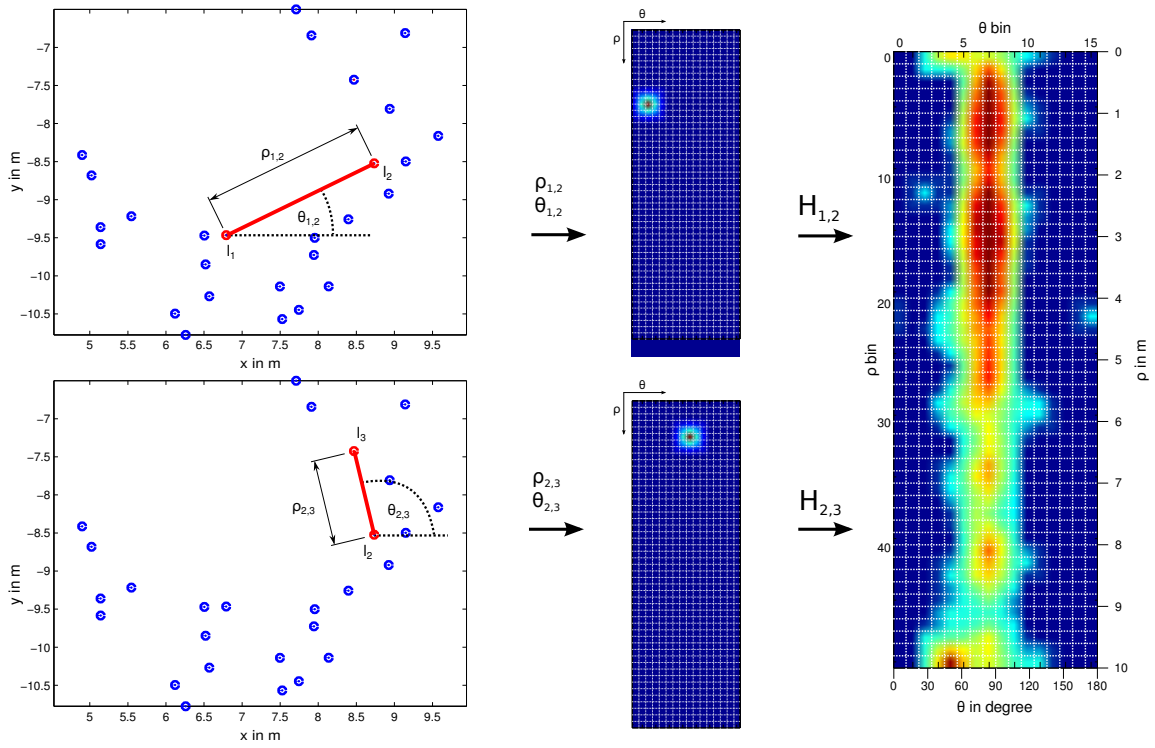


Figure 3.2.: GLARE Signatures: The orientation $\theta_{i,j}$ and distance and $\rho_{i,j}$ for each landmark relation (left) is modeled as a Gaussian distribution (center). Incorporating all landmark descriptions $H_{i,j}$ results in the scan signature $S^{(k)}$ (right).

Based on the k-th range scan and the extracted set of N landmarks we calculate the Euclidean distances $\rho_{i,j}$ of each landmark $l_i = \{x_i, y_i\}$ to all others $l_j = \{x_j, y_j\}$ with $i \neq j$ in the input scan's coordinate frame. We further estimate the bearings $\theta_{i,j}$ and $\theta_{j,i}$ for all landmarks as follows:

$$\theta_{i,j} = atan2(y_i - y_j, x_i - x_j) \tag{3.5}$$

Since $\theta_{i,j}$ and $\theta_{j,i}$ are redundant, we stick to the bearing $\theta^+_{i,j}$ with $\theta^+_{i,j} = \max(\theta_{i,j}, \theta_{j,i})$ for the following processing. With all $\rho_{i,j}$ and $\theta^+_{i,j}$ being estimated, a distribution of geometrical relations is obtained. The bearings θ^+ and distances $\rho_{i,j}$ are discretized and associated to bins of uniform size:

$$\left(\theta^+_{i,j}, \rho_{i,j}\right) \in bin\,(n_\theta, n_\rho) \tag{3.6}$$

Each landmark relation is incorporated as a multivariate Gaussian with its mean being set according to the associated bin $n = (n_\rho, n_\theta)$ and a measurement covariance matrix Σ_H. A 2D histogram $H_{i,j}$ is generated with each position $m = (m_\rho, m_\theta)$ being estimated as:

$$H_{i,j}(m) = \mathcal{N}(m - n,\, \Sigma_H) \tag{3.7}$$

The usage of a discretized Gaussian instead of the original values $\left(\theta^+_{i,j}, \rho_{i,j}\right)$ enables the pre-computation of histogram elements limited by Σ_H and thus a significant performance speed-up. The signature $S^{(k_i)}$ for the landmark l_i is concatenated according to:

$$S^{(k_i)} = \sum_j H_{i,j} \tag{3.8}$$

Having obtained signatures for individual landmarks $S^{(k_i)}$, we are also able to generate a signature for the k-th range scan. This is done by means of a normalized sum given the entire set of landmark signatures:

$$S^{(k)} = \eta \sum_i S^{(k_i)} \tag{3.9}$$

Here, η denotes a normalization factor. An example for the estimation of GLARE signatures is shown in Figure 3.2.

The aforementioned solution differs from the state of the art in place recognition which typically make use of landmark descriptors being assigned to quantized words of an a-priori learned vocabulary [88, 162]. These approaches use hard-voting which means that each descriptor is assigned exactly one word of the vocabulary. GLARE instead applies soft-voting since also adjacent cells of discretized histogram positions are incorporated. The amount of smoothness can be adjusted by the matrix Σ_H which models the uncertainties of the range measurements. This can be set according to accuracy of the range sensor and localization quality of the utilized feature detector. The soft-voting is especially well-suited for the processing of noisy range measurements.

3.3.3. Efficient Scan Retrieval

Once the scan signatures are built, we aim at preserving these in a global repository S and at the same time maintain an index structure for efficient retrieval. Due to the high-dimensional signature vectors being stored in S, it is necessary to use approximate nearest neighbour search (ANN) to allow for fast retrieval [10]. The kd-trees being commonly

used select the dimension with the largest variance to bisect the input data. According to [120] it can be observed that this structure performs unsatisfactorily in conjunction with high-dimensional data. It is therefore advantageous to utilize a multitude of randomized kd-trees. Here, the splitting hyper-planes are randomly selected from the most variant dimensions which results in more suitable representation for high-dimensional data.

The ANN search being used differs from the exact nearest neighbour search in that the amount of points being considered is limited to ϵ_{max}. Given the bound is set properly ANN provides a suitable approximation. Since this is a trade-off of precision and runtime, the value has to be determined carefully. Also the behavior might change during runtime when data points are added to the trees which entails changes in the balance quality. Thus the number of points being recently added and the amount being considered for search (ϵ_{max}) have to be adjusted for the target application. If GLARE is used for localization with respect to a given map, it can be assumed that the kd-trees are well-balanced since an optimal ordering can be found once within an offline preparation. However, when using the algorithm in SLAM for loop closure detection, the repository is constantly expanded as new parts of the environment are traversed. This implies that the kd-trees have to be rebalanced online. In our implementation, individual kd-trees are reorganized once the amount of appended elements exceeds one third of the total number of the tree. As this procedure can be complex for larger environments scales, it is necessary to run this in parallel to the remaining SLAM modules in order to ensure reliability in the continuous pose estimation while avoiding to miss-out on observations.

Assuming a scan signature has been successfully generated, it can be utilized as a query g_i. The repository S is subsequently searched for potential scan correspondences L given g_i and the distance function $dist_S$:

$$L = \min_{k}[dist_S(g_i, S)],\ k \leq K \tag{3.10}$$

Different distance functions $dist_S$ can be integrated at this stage. In our case we make use of the L1 norm. In order to reduce the search space we restrict the retrieval to the K nearest neighbours. This value has to be set properly in order to achieve an optimum in regards of precision/recall and runtime. The evaluation of a very large amount of putative place correspondences entails that a significant number of scans have to be checked for geometric consistency. This process is computationally expensive as it involves a number of trigonometric operations on the feature sets. On the other hand, the risk of missing correct correspondences increases with fewer candidates being considered. Typically this value should be set depending on whether additional information such as a pose prior is available (e.g. in SLAM), the repository size, the impact of missing out correspondences and runtime limitations. Also, the amount of self-similarity being expected for the operation environment should be considered. Our experimental section will further investigate this parameter.

The use of approximate nearest neighbour search for retrieval enables optimal performance in conjunction with GLARE signatures. Other methods such as the one of inverted files are common in visual recognition systems [37, 88] and GFP [164]. This enables quantized features to directly point to those images or scans they were observed which literally

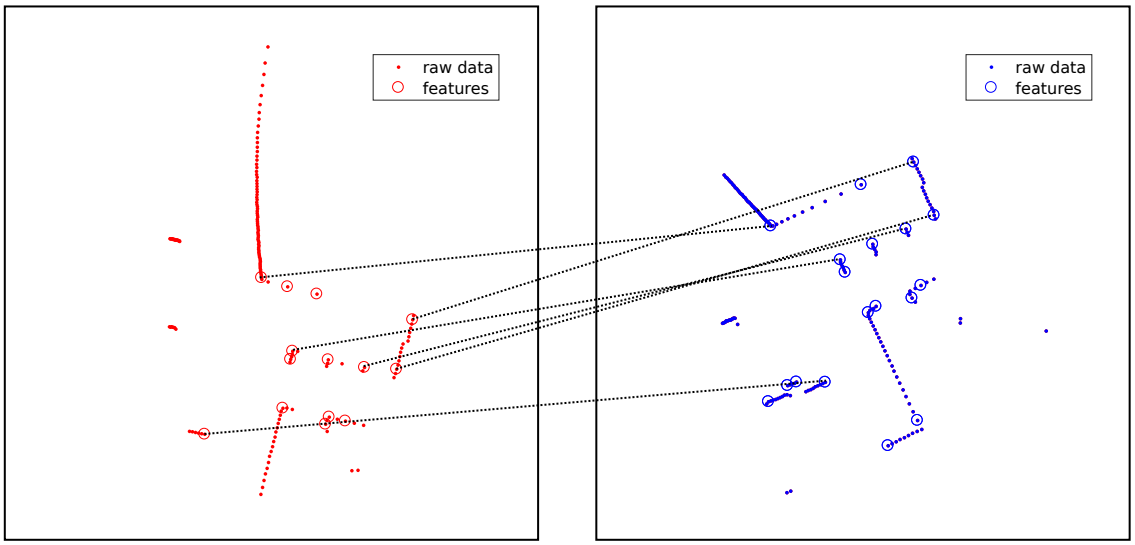


Figure 3.3.: Geometric verification. This figure visualizes a reference (red) and an observation (blue) range scan being matched. First, our algorithm extracts local features from either scans. Each feature is assigned a GLARE descriptor which is subsequently utilized to find correspondences Samong the two scans (shown as lines in the plot). A RANSAC-based feature set matching is used to estimate a rigid transformation between observation and reference scans.

inverts the search. Subsequently the retrieval only considers places that have been picked by this. However, due its dense histograms inverted files are not recommendable in combination with GLARE. Inverted files are commonly combined with BOW-based retrieval with the latter generating rather sparse histograms of visual word occurrences [37, 88]. This is due to the fact that only a small number of visual words are actually observed for individual places. GLARE, in contrast, incorporates all landmark co-occurrences resulting in $N^2/2$ compared to N (BOW) histogram contributions. In addition to that, we utilize soft-voting which implies that observations contribute to the closest bin as well as adjacent bins in the histogram. This allows more robustness thanks to an expended retrieval which is well-suited for noisy, potentially inaccurate, range measurements.

3.3.4. Geometric Verification

In the preceding section we exhaustively described that range scans can be transfered into a representation which allows to efficiently search for similar scans in order to determine whether these refer to the same physical location. However, repetitive structures in the environment might entail ambiguities in that spatially distant places generate similar GLARE signatures. In order to reduce the number of false positive matches, we further introduce a geometric verification step which is applied to the top K matches returned by the retrieval system.

For this purpose we utilize **Ran**dom **Sa**mple **C**onsensus (RANSAC), a common method for fitting a model to data [51]. The key idea is to select a minimal subset which is required to fit a specific model and subsequently apply the estimated parameters to the entire input data. In the case of scan matching, our model is a rigid-body transformation in 2D with translation and rotation assuming a constant scale [162]. The number of inlier and the fitting quality of the estimate are used to evaluate the geometric consistency of the putative match.

We denote r as a set of N_r landmarks associated with the place $S^{(k)}$ and o as a set of N_o landmarks associated with the observation scan. The position of a landmark o_i is referred to as $o_{i,pos}$ and the descriptor as $o_{i,desc}$ respectively. The consistency check aims at finding corresponding landmarks in the reference and observations scans serving as precondition for estimating a rigid transformation. The correspondence search is again accomplished by incorporating the geometrical relations of observed landmarks. Therefore each landmark l_i detected in the k-th range scan is assigned a signature $S^{(k_i)}$ according to Eq. 3.8. Corresponding landmarks are identified by matching the landmark signature $S^{(k_i)}$ of a place $S^{(k)}$ against those observed in the top K candidates of the retrieval.

If all true correspondences of the scans were given, a transformation of these could be estimated in closed form (see also Eq. 3.19). However, since there are likely to be outliers in the correspondence set being estimated based on the landmark descriptors, we have to generate and evaluate multiple hypotheses. Those landmark correspondences having a distance below a threshold τ_{desc} are incorporated in our RANSAC-based matching which generates a candidate parameter $a = (x_{tr}, y_{tr}, \theta_{tr})$ for each hypothesis given the two scans.

Thanks to the limitation to 2D, we require only two landmark correspondences $\hat{C} = \{o_i; r_j\}$ for estimating $\mathbf{T}$. At first we estimate the relative orientations θ_{ref} and θ_{obs} based on the given landmark correspondences $(o_{i,1}, r_{j,1})$ and $(o_{i,2}, r_{j,2})$ for the reference and observation scan respectively:

$$\theta_{obs} = \arctan(y_{i,1} - y_{i,2}, x_{i,1} - x_{i,2}) \tag{3.11}$$

$$\theta_{ref} = \arctan(y_{j,1} - y_{j,2}, x_{j,1} - x_{j,2}) \tag{3.12}$$

The rotational component θ can be calculated as:

$$\theta = \theta_{obs} - \theta_{ref} \tag{3.13}$$

Given the twist angle θ we are able to estimate the rigid transformation expressed by the parameter vector $\mathbf{a}$:

$$\mathbf{a} = hypot(\hat{C}) := \begin{pmatrix} x_{j,2} - \cos\theta\, x_{i,2} + \sin\theta\, y_{i,2} \\ y_{j,2} + \sin\theta\, x_{i,2} + \cos\theta\, y_{i,2} \\ \theta \end{pmatrix} \tag{3.14}$$

$$T = t(\mathbf{a}, \mathbf{x}) := \begin{pmatrix} cos\theta_{tr} & sin\theta_{tr} \\ -sin\theta_{tr} & cos\theta_{tr} \end{pmatrix} \mathbf{x} + \begin{pmatrix} x_{tr} \\ y_{tr} \end{pmatrix} \tag{3.15}$$

The transformation T is applied to the observation scan in order to directly compare the positions of the corresponding landmarks and the overlap of the two scans. The algorithm for the verification of the hypothesis is shown in Algorithm 1. This procedure is repeated multiple times with the goal of finding the best hypothesis a.

Algorithm 1 verify_hypot (o, r, a) :

for all o_i **do**
  $score = 0$
  $min = \{\}$
  $d_{min} = 0$
  Apply transform to landmark
  $\hat{o}_i = t(a, o_i)$
  for all r_j **do**
    $d_{pos} = dist(o_{i,pos}, r_{j,pos})$
    if $d_{pos} < d_{min}$ **then**
      $min = j$
      $d_{min} = d_{pos}$
    end if
  end for
  if $d_{min} < \tau_{pos}$ **then**
    Add pair $\{o_i, r_{min}\}$ to inlier set C_{inlier}
  end if
  $score = score + d_{min}$
end for
return $score, C_{inlier}$

The residual error ϵ as well as the number of inlier correspondences indicate whether the two given scans satisfy our geometric consistency check and hence are considered as matching places. As usual in RANSAC, the best model is selected after a number of iterations N. This number denotes a lower bound and is obtained from the target probability p of drawing at least one sample without outliers and the ratio r_{out} of expected outliers in the data and the number of samples s required to estimate the model:

$$r_{out} = \frac{\#\, outliers}{\#\, all\ samples} \tag{3.16}$$

$$N \geq \frac{\log(1-p)}{\log(1-(1-r_{out})^s)} \tag{3.17}$$

The transformation T_{best} of the best model returned after N iterations is refined in a post optimization step. Based on all inlier correspondences C_{inlier} we again estimate a

rigid transformation $\mathbf{a}$ by minimizing the squared residual error function $\epsilon^2(|\,\mathbf{x}\,|)$ which can be expressed by the following energy function:

$$E(\mathbf{a}) = \sum_{i=1}^{N_o} \min_j \epsilon^2(|\mathbf{r_j} - t(\mathbf{a}, \mathbf{o_i})|) \tag{3.18}$$

We use the Euclidean distance as error function such that $\epsilon^2(|\,\mathbf{x}\,|) = \|\mathbf{x}\|^2$. The optimal registration of the reference and observation scan is obtained by minimizing over $\mathbf{a}$:

$$\hat{\mathbf{a}} = \underset{a}{\operatorname{argmin}} \sum_{i=1}^{N_o} \min_j \epsilon^2(|\mathbf{r_j} - t(\mathbf{a}, \mathbf{o_i})|) \tag{3.19}$$

The entire RANSAC based consistency check is explained in Algorithm 2. Since we require a set of solely two landmark correspondences, the check for 3-DOF is significantly more efficient than those used for image matching. For comparison, the requirements of these algorithms are shown in Table 3.1. The underlying mathematical models for these sensors such as the essential and fundamental matrices being used to fit transformations are more complex due to the higher degree of freedom. The reduced runtime requirements of the presented model allows to incorporate more putative place candidates K in the retrieval which typically results in increased recall rates as it will be shown in the experimental section. However, it is still expensive for $K > 100$ which is why it is important to achieve place and landmark descriptors at the best possible distinctiveness in order to reduce the number of required checks.

Sensor	Model	# Parameters s	# Iterations N
2D range sensors	Rigid transform	2	16
Stereo camera	Essential matrix	3	34
Calibrated camera	Essential matrix	5	145
Uncalibrated camera	Fundamental matrix	8	1177

Table 3.1.: Runtime requirements for different models and sensors for $p = 0.99$ and $r_{out} = 0.5$.

Algorithm 2 RANSAC_matching_model (r, o, M) :

for all o_i **do**
  for all r_j **do**
    $d_{desc} = dist(o_{i,desc}, r_{j,desc})$
    if $d_{desc} < \tau_{desc}$ **then**
      Add pair $\{o_i, r_j\}$ to correspondence set C
    end if
  end for
end for
Check for minimum number of correspondences
if $size(C) \geq 2$ **then**
  loop M times
    Draw random consensus $\hat{C} \in C$ of 2 point pairs
    $i = rand(1, size(C))$
    $j = rand(1, size(C))\ with\ j \neq i$
    $a = hypot(\hat{C})$
    $\epsilon = verify\_hypot(a, C)$
    if $\epsilon < \epsilon_{best}$ **then**
      $T_{best} = a$, $\epsilon_{best} = \epsilon$, $C_{inlier} \in C$
    end if
  end loop
  Optimize pose based on all inlier (see Eq. 3.19)
  $T_{opt} = optimize\_pose(C_{inlier})$
else
  $C_{inlier} = \emptyset$
end if
return $T_{opt}, \epsilon_{best}, C_{inlier}$

3.3.5. Experiments

This section is dedicated to experimental evaluation of our novel algorithm GLARE. We therefore made use of four publicly available datasets which are also part of state-of-the-art investigations [162, 164]. In particular, we include three datasets of outdoor environments and one dataset being captured indoors (see Table 3.2).

Setup

The goal of our experiments was an in-depth comparison of GLARE to the most related approach GFP [164]. Specifically, we investigated the two variants GLARE-1 and GLARE-2. GLARE-1 solely incorporates distances of co-occurring landmarks omitting relative bearings. GLARE-2, in contrast, makes use of both of both, distances and bearings. For either of the variants we utilized 100 equally spaced bins with each having a width of

$0.5m$ for the outdoor datasets and 50 bins having a width of $0.2m$ for the indoor datasets. GLARE-2 further expends the signature space by using 8 angular bins. Our opponent algorithm GFP is parametrized with those settings being recommended in [164].

Our experimental investigations were carried out for all three algorithms, GLARE-1, GLARE-2 and GFP using different numbers of nearest neighbours, in particular with $K = 50$, $K = 100$ and $K = 500$. Our RANSAC-based verification filters putative matches not fulfilling the rigidity constraints by evaluating the residual error of the transformation. The thresholds are set to $0.5m$ linear, $0.2rad$ angular error respectively. These values were kept fixed while the minimal amount of N_{corr} inlier correspondences C_{inlier} are varied (here: $N_{corr} = 1, 2, ..., 32$). This was used in order to generate precision/recall curves.

Name	Type	# Scans	# Landmarks	Path length [m]
Intel-lab	indoor	2672	39392	360.7
FR-Clinic	outdoor	6917	190760	1437.6
Victoria Park	outdoor	5751	81795	4206.14
Kenmore	outdoor	13063	499237	6588.34

Table 3.2.: Datasets used in our experiments. More details can be found in Appendix A.2.

Evaluation Methodology

For the evaluation of the presented algorithm we make use of the precision/recall curve. The origins of these curves go back to the development of information retrieval algorithms and have become the state-of-the-art evaluation method in the field of place recognition [56, 57, 164]. Precision/recall curves describe the ratio of correctly recognized places (true positives) compared to

1. all estimated place correspondences (precision)
2. the total number of possible place correspondences including those being correctly recognized and those being missed (recall)

In particular we can retain precision and recall as follows:

$$precision = tp \,/\, (tp + fp) \tag{3.20}$$
$$recall = tp \,/\, (tp + fn) \tag{3.21}$$

with tp, fp, tn, fn denoting true positive, false positive, true negative and false negative respectively. In the literature authors use precision/recall curves with different preconditions. In particular, they either assume that putative place correspondences:

1. are solely obtained from the retrieval algorithms [56, 57] or
2. have undergone additional geometric verification or filter steps [162, 164]

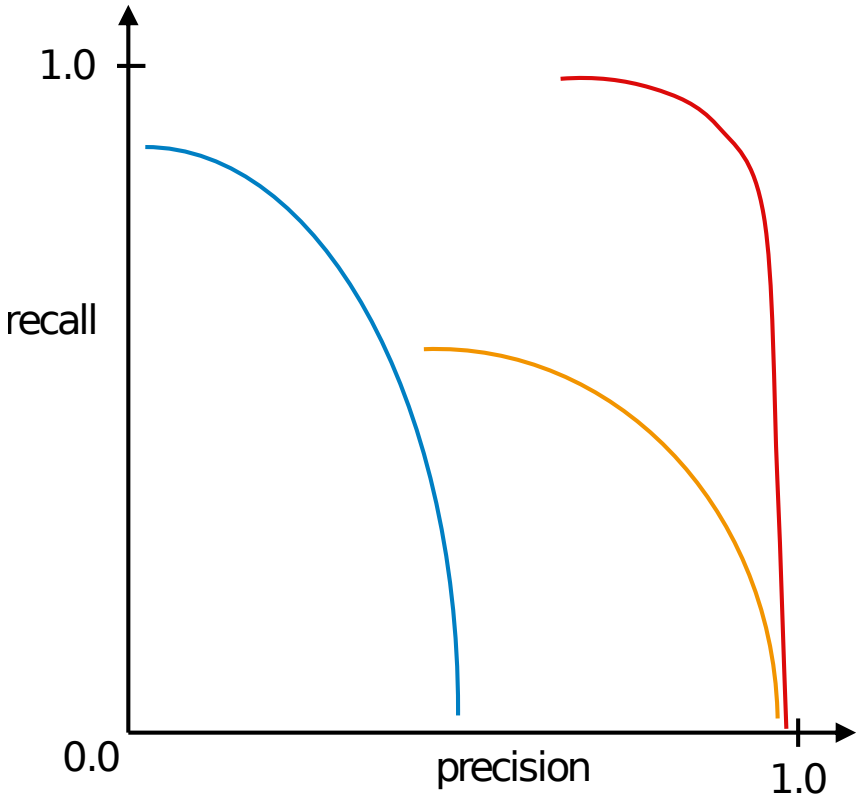


Figure 3.4.: **Precision/recall curves**. This figure illustrates precision/recall curves which are used for the evaluation of our place recognition algorithm. In particular, it can be seen three curves showing different characteristics. The red curve (right) refers to the best performance showing a recall close to 1.0 at 100% precision. The orange curve (center) achieves a recall rate much below the red curve. For increasing precision values the recall drops close to zero. The blue curve (left) performs better in terms of recall compared to the orange one. However, the precision remains below 0.5 even for a very low recall values. Indeed, the red one is best. Whether the blue or the orange curve is better depends on the target application. If additional filters or robust methods are available, then a lower precision can be acceptable. In this case, the system benefits from more putative place correspondences (blue one is better). If this is not available fatal consequences such as divergences are risked.

In our evaluation we make use of option (2) since this is common for place recognition in 2D range data and enables a better comparison to the state of the art in this field [162, 164].

In particular, this means that we use GLARE to recognize observed places with our database and the RANSAC-based geometric verification (see Section 3.3.4) to evaluate our system. Thus the precision/recall values obtained for our system are originated from geometrically verified correspondence estimates. A place correspondence is considered

correct if the distance of the pose estimated using RANSAC and the ground truth pose is below $0.5m$ (translation) and $0.2rad$ (rotation).

How can precision/recall curves being interpreted? An illustrative example for this is shown in Fig. 3.4. It visualizes three curves based on different precision/recall values. Generally speaking, the closer this curve to the upper right corner $(1.0, 1.0)$ the better the performance of the algorithm. However, it depends on the application whether which of the parts precision and recall are more important. For the global localization with a method being capable of presenting multi-modal distributions (e.g. a particle filter) the precision does not have to be 100% since these estimates are further verified by the filter based on multiple observations. Here, a high recall is advantageous in order to allow a fast convergence of the filter to the true pose, otherwise this takes longer. For application as loop closure detection in SLAM, a slightly lower recall rate can be acceptable since the system does not have to detect each possible loop closure. Of course, the more loop closures are incorporated the more accurately we can estimate the poses and the map. The requirements for the precision depend on the fact whether the SLAM algorithm is robust against loop closure outliers [4, 154] or can incorporate additional information, e.g. by matching sequences [117]. We will detail this specific problem in Chapter 4 of this thesis.

Results

The figures 3.5-3.6 show the results obtained in our experiments. It can be clearly seen that GLARE-1 as well as GLARE-2 outperform GFP on all datasets being captured outdoors. Only if a significant larger number of nearest neighbours K is incorporated, GFP achieves comparable results. Increasing this value for GLARE does not entail notable differences in precision/recall which demonstrates that this is not necessary. This number K determines the number of full geometric verifications being required. Since this step is computationally expensive, it has an essential impact on the run time of the place recognition. GFP is less efficient than GLARE as it requires a lot more geometric verification steps for achieving comparable results for the investigated outdoor datasets. However, it can also be seen that GLARE performs slightly worse on the indoor dataset intel-lab (see Fig. 3.5-3.6). We suppose that this can be ascribed the common repetitive elements in man-made building structures causing more self-similarity and thus less variance in the landmark displacements.

Our experiments also reveal that GLARE-2 outperforms GLARE-1 which is not surprising as the former incorporates more information (relative bearings of landmarks). However, the differences are rather minor which demonstrates that the distance information is a more distinctive feature than the bearing. It should be noted that GLARE, in contrast to GFP, does not require any prior training stage as signatures can be learned online.

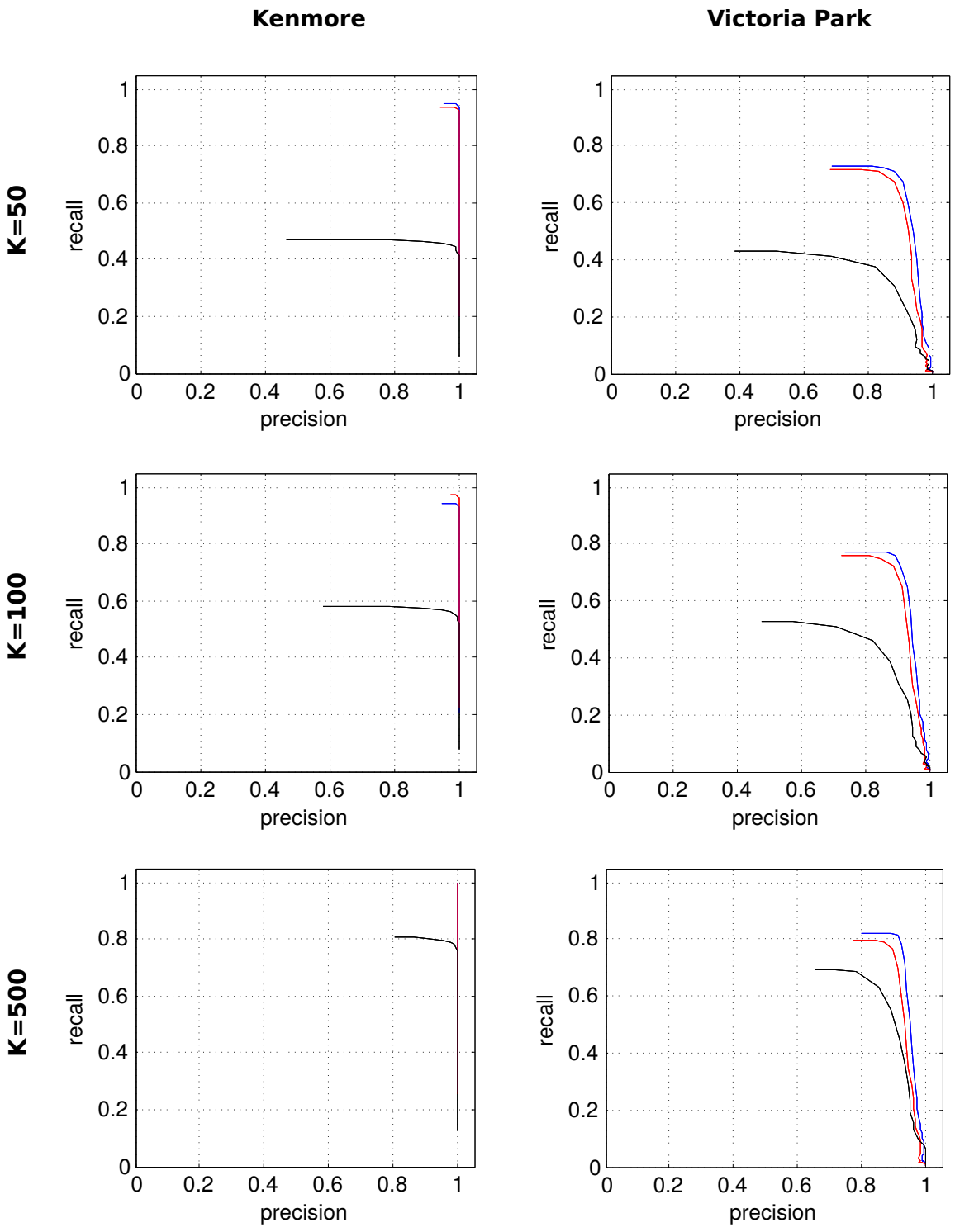


Figure 3.5.: Experimental results obtained using GLARE-1 (red), GLARE-2 (blue) and GFP (black) are shown for different number of nearest neighbours K taken into consideration $(50, 100, 500)$. The results for GLARE-1/2 on the Kenmore dataset ($K = 500$) are so close that only one is visible in the plot.

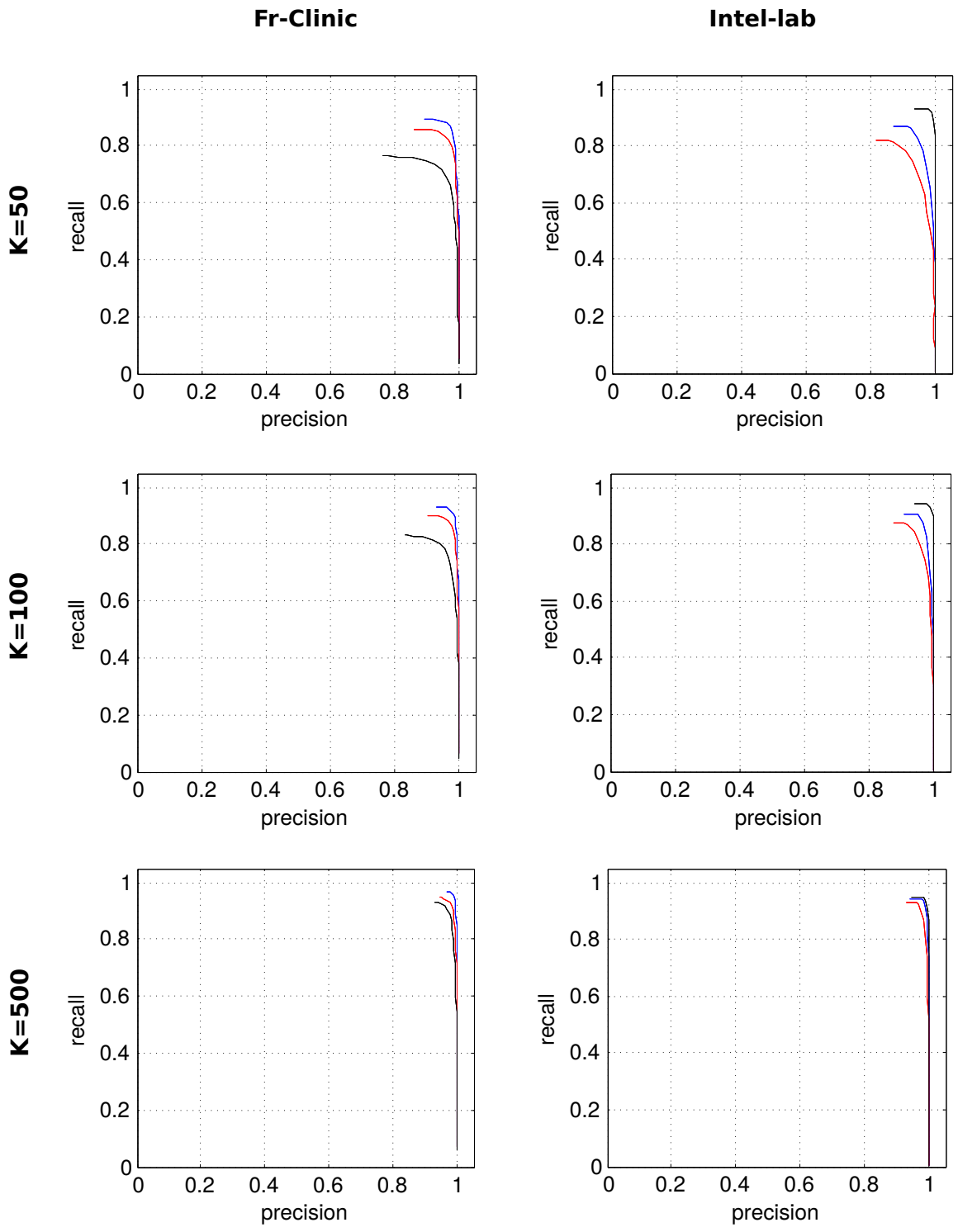


Figure 3.6.: Experimental results obtained using GLARE-1 (red), GLARE-2 (blue) and GFP (black) are shown for different number of nearest neighbours K taken into consideration $(50, 100, 500)$.

Run time

Table 3.3 summarizes the mean run times for GLARE and GFP at the example of the Kenmore dataset. As already mentioned, GFP has to consider a large set of nearest neighbours in order to be comparable to GLARE (see also Fig. 3.5-3.6). This is why we set $K = 500$ for the runtime evaluation. A direct comparison of GLARE-1 and GLARE-2 shows that the former performs slightly better. This is of course due to the larger signature space of GLARE-2 which causes increased runtime for the scan retrieval. The geometric verification is not as much affected by this since the generation and matching of small sets of GLARE-2 landmark signatures can be neglected. This also motivates a combination of GLARE-1 for scan retrieval and GLARE-2 for geometric verification in order to obtain an optimal balance of memory consumption, runtime and precision/recall. Given the results shown in Fig. 3.5-3.6, GFP would have to incorporate even more than $K = 500$ nearest neighbours in order to get close to the precision/recall of GLARE. Increasing K, however, entails a significant runtime expense making the place recognition algorithm impracticable for applications such as loop closure detection. As all candidates utilize the same methods for feature detection, there is not any notable difference recognizable (see also Section 3.4.1). The description phase is slightly faster for GLARE, particularly GLARE-1.

Table 3.3 also reveals that the geometric verification of GFP is remarkably slower that for GLARE. We suppose that this can be referred to two main reasons. First, the GLARE landmark descriptors are more distinctive for this type of environment which means that a smaller set of putative landmark correspondences can be passed to RANSAC which significantly reduces the runtime. Second, the GLARE place signatures can be better distinguished which entails a reduced set of nearest neighbours being considered whereas GFP uses the maximum allowed number of $K = 500$. As shown in Fig. 3.5, a number of $K = 50$ neighbours is actually sufficient for GLARE. This allows a substantial reduction of the runtime as the geometric verification is the most expensive part of our place recognition system.

Table 3.3.: Mean run time on Kenmore dataset for $K = 500$

	GLARE-1 [ms]	GLARE-2 [ms]	GFP [ms]
feature detection	7.5402	7.298	7.276
feature description	0.768	1.291	1.797
scan retrieval	18.578	25.312	81.462
geometric verification	89.596	88.928	330.586
total	**116.482**	**122.829**	**421.121**

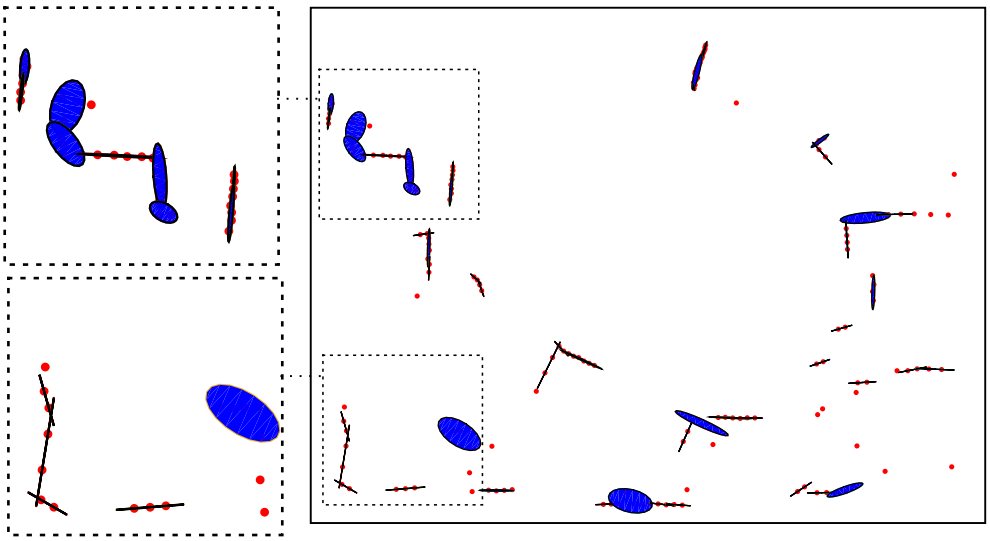
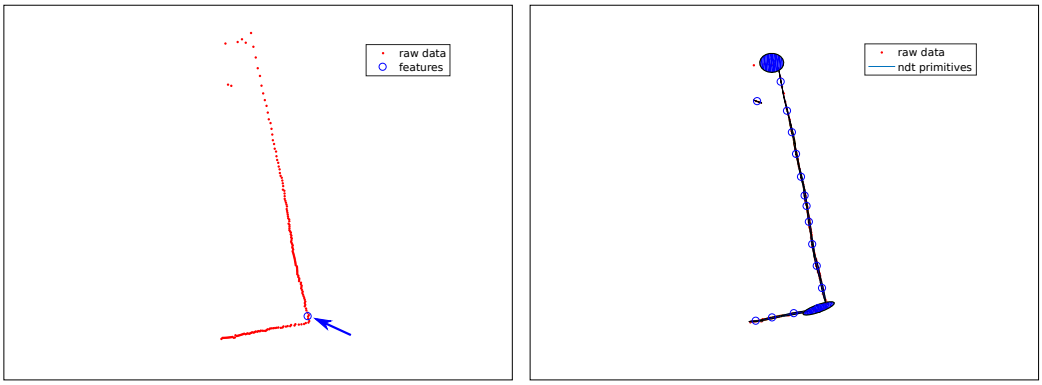


Figure 3.7.: Normal distribution transform. This figure shows a normal distribution transform (blue) plotted on top of the raw 2D range scan (red). The measurement end points are projected onto a regular grid. A normal distribution (blue ellipse) is estimated for each non-empty grid cell. This representation allows to extract features while simultaneously preserving a dense description of a subsampled range scan. It is beneficial for place recognition and pose estimation since the features are not limited to points at curvature extrema.

3.4. From Landmarks to Surface Primitives

GLARE outperforms the state of the art in place recognition using range data. The difference becomes apparent in outdoor environments since geometrical relations are particularly well-suited due to the presence of more landmarks and their more natural spatial distributions. Indoor environments are more challenging since man-made buildings often consist of long repetitive structures such as walls which generate less landmarks in terms of curvature discontinuities (see also Fig. 3.8). This is one reason why one can observe that, in conjunction with range scans, landmark maps are rather used outdoors whereas grid maps are often utilized indoors for localization and mapping purposes. We introduce an algorithm that bridges the gap between these map types allowing the description of places with few extrema in curvature and at the same time enabling efficient feature-based matching. This is rendered possible by describing occupied subspaces in terms of their surface characteristics. We introduce **G**eometrical **R**elations of Surf**a**ce **P**rimitiv**e**s (GRAPE) which, similar to GLARE, utilize the geometrical relations of co-occurring surface primitives to generate place and landmark signatures. The algorithm is exhaustively described in the following section.



(a) Features extracted from curvature extrema (GLARE).

(b) Normal surface primitives (GRAPE).

Figure 3.8.: Comparison GRAPE/GLARE. This figure shows the feature extraction from 2D range scans at a structureless indoor place (end of a corridor). The raw scan points are red in either plot (a)-(b). As it can be seen in (a), the feature detection based on curvature extrema as it used in GLARE and GFP extracts only one landmark. Since our pose estimation requires two point correspondences, we are unable to estimate a rigid transformation at this place based on these algorithms. Figure (b), in contrast, illustrates features being extracted as normal surface primitives based on our algorithm GRAPE which is able provide a dense description of the place independently of existing extrema in the range measurements.

3.4.1. Extraction of Surface Primitives

At first, all measurements of a 2D range scan are projected onto a regular grid. Similar to Magnusson et al. [112] we estimate a normal distribution with mean μ_i and a covariance matrix Σ_i for the measurements of each non-empty cell i. The surface orientations θ_i of each primitive is required in order to model the spatial relations. For this purpose we utilize an eigenvalue decomposition of the surface primitive's covariance matrix. The eigenvector e_{min} with the smallest eigenvalue is selected for estimating the orientation $\hat{\theta}_i$:

$$\hat{\theta}_i = atan2(e_{min}^{\{y\}}, e_{min}^{\{x\}}) \tag{3.22}$$

Since the eigenvector e_{min} is not necessarily pointing towards the sensor's origin, we explicitely account for this by estimating the primitives' orientation ψ_{orig} towards the sensor's origin:

$$\psi_{orig} = atan2(\mu_i^{\{y\}}, \mu_i^{\{x\}}) + \pi \tag{3.23}$$

If the displacement of ψ_{orig} and $\hat{\theta}_i$ exceeds a threshold τ_{max}, we map the surface primitive's orientation as follows:

$$\theta_i = \begin{cases} \hat{\theta}_i & if\ (\psi_{orig} - \hat{\theta}_i) < \tau_{max} \\ \hat{\theta}_i + \pi & otherwise \end{cases} \tag{3.24}$$

In this way it is ensured that θ_i is assigned the expected direction. We empirically set $\tau_{max} = \pi/3$, which however is not too crucial. The size and resolution of the grid have to be justified according to the type of environment and the sensor used. As a result of this step we obtain a set of surface primitives $l_i = \{\mu, \Sigma, \theta\}_i$ for each range scan.

3.4.2. Encoding Spatial Relations of Surface Primitives

The encoding of spatial relations for co-occurring surface primitives is, except for minor changes, equivalent to GLARE (see Section 3.3). For a set of N surface primitives $l_1, \ldots, l_N$ detected in the k-th range scan we estimate the distances $\rho_{i,j}$ of each primitive $l_i = \{\mu, \Sigma, \theta\}_i$ to all others $l_j = \{\mu, \Sigma, \theta\}_j$ of the set with $i \neq j$ within the local coordinate frame of the range scan. In contrast to GLARE, we utilize the Mahalanobis rather than the Euclidean distance:

$$dist(l_i, l_j) = ((\mu_i - \mu_j)\mathbf{\Sigma_j}^{-1}(\mu_i - \mu_j))^{\frac{1}{2}} \tag{3.25}$$

A notable property of this metric is that the distances $dist(l_i, l_j)$ and $dist(l_j, l_i)$ are not necessarily equivalent. This is due to the fact that distances are estimated given the mean of the first and second primitive and incorporating the covariance of the latter. If the covariances Σ_i and Σ_j are different, we will also obtain different distance results.

Similar to GLARE, the bearings $\Delta\theta_{i,j}$ and $\Delta\theta_{j,i}$ of co-occurring primitives are estimated according to Eq. 3.5. The remaining procedure of generating primitive and place signatures is the same as for GLARE (see Section 3.3). As a result of this step we again obtain individual descriptors $S^{(k_i)}$ for each surface primitive and a composite place signature $S^{(k)}$ for each scan k.

3.4.3. Experiments

In order to evaluate GRAPE, a number of experiments were carried out. The first experiment again shows the recognition performance on publicly available datasets. Here the goal is to compare the results of Geometrical Relations of Surface Primitives (GRAPE) to the state-of-the-art approach GFP [164] and our previously presented algorithm GLARE. The second experiment analyzes the recognition performance of GRAPE over longer periods of time. While the first experiment is rather focusing on place recognition for detecting loop closures in SLAM, the second demonstrates GRAPEs performance for place recognition over longer periods of time.

Experiment 1 - GRAPE vs. State of the art

For this experiment we selected four different publicly available datasets of indoor and outdoor environments which were also used in [73] and [164] to evaluate GLARE and GFP respectively (see Table 3.2).

GFPs are generated using the optimal settings, as shown in [164]. For GLARE and GRAPE we use 8 angular bins, 100 linearly sized bins with a size of 0.5 m for the outdoor datasets and 50 bins with a size of 0.2 m for the indoor datasets (see [73]). GRAPE is initialized with grid cell sizes of 0.25 m for indoor and 1.0 m for outdoor datasets. We tested GRAPE, GLARE and GFP on all datasets with a number of $K = 50$ nearest neighbours. The geometric verification rejects putative matches of places by thresholding based on the residual error (linear: $0.5m$, angular: $0.2rad$). Again, the thresholds for estimating the precision/recall curves are obtained using different numbers of inlier correspondences (see Section 3.3). The datasets are matched one-by-one, but ignoring trivial self-matches. This procedure is similar to the one presented in [164]. The results are shown in Figure 3.9.

Experiment 2 - Long-term Recognition using GRAPE

This experiment evaluates the recognition of GRAPE over longer periods of time. For this purpose we make use of a subset of five datasets of the MIT Stata Center collection [48] covering a total period of time of more than 2 months (see Table 3.4). The ground truth supplemented with these datasets provides an accuracy of about 2cm. The GRAPEs generated for the first dataset (2012-01-18) serve as prior map for the subsequent datasets. This demonstrates the recognition performance of GRAPE in terms of global localization. The results are shown in Figure 3.10. Only the results of the first dataset (2012-01-18) are obtained similar to the first experiment by matching the dataset to itself but again excluding trivial self-matches. The ground truth poses of datasets 2-5 are used to find overlapping areas of the environment being traversed. Hence we exclude those scans of the datasets 2-5 that are mapped outside the area of dataset 1. The settings for GRAPE are similar to the ones for the indoor environments of the first experiment. The results are shown in Figure 3.10.

#	Date	# Scans	# Primitives	Path length [m]
1	2012-01-18	2562	237559	683.0
2	2012-01-25	1355	135246	348.0
3	2012-01-28	2279	212455	635.0
4	2012-02-02	2806	275128	1003.0
5	2012-04-02	1726	154336	606.0

Table 3.4.: Logs of Stata Center dataset collection being usedin our experiments. More details about the environment can be found in Appendix A.2.

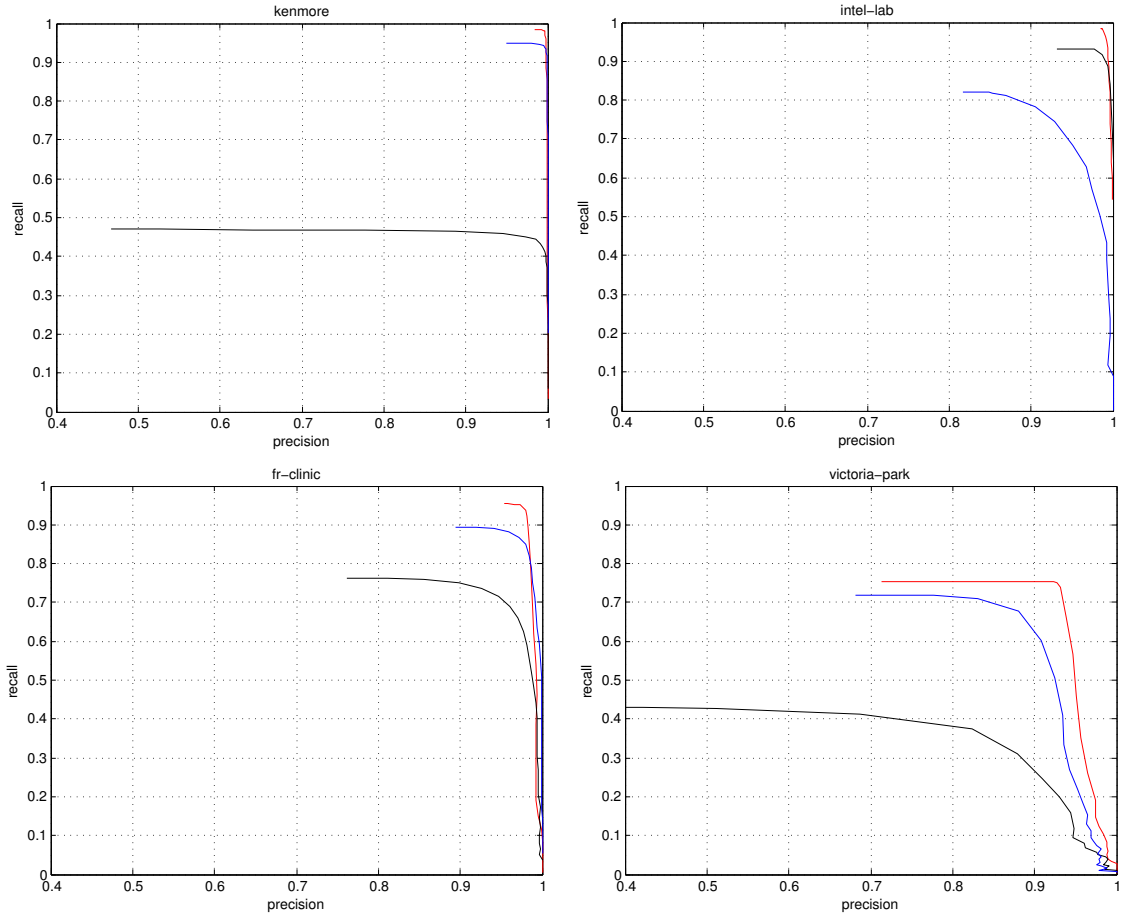


Figure 3.9.: Experimental results obtained using GRAPE (red), GLARE (blue) and GFP (black) are shown for $K = 50$ nearest neighbours taken into consideration.

Run time

The run times for GRAPE, GLARE and GFP are shown in Table 3.3. GFP only gets close to GLARE's and GRAPE's recall rates for a large number of nearest neigbours which is why $K = 500$ is used for this experiment. Actually GFP requires even more putative neighbours to be considered, however, the run time for this becomes highly impracticable.

Table 3.5.: Mean run time on Kenmore dataset for $K = 500$

	GRAPE [ms]	GLARE [ms]	GFP [ms]
feature detection	11.121	7.298	7.276
feature description	6.322	1.291	1.797
scan retrieval	26.198	25.312	81.462
verification	102.596	88.928	330.586
total	**146.237**	**122.829**	**421.121**

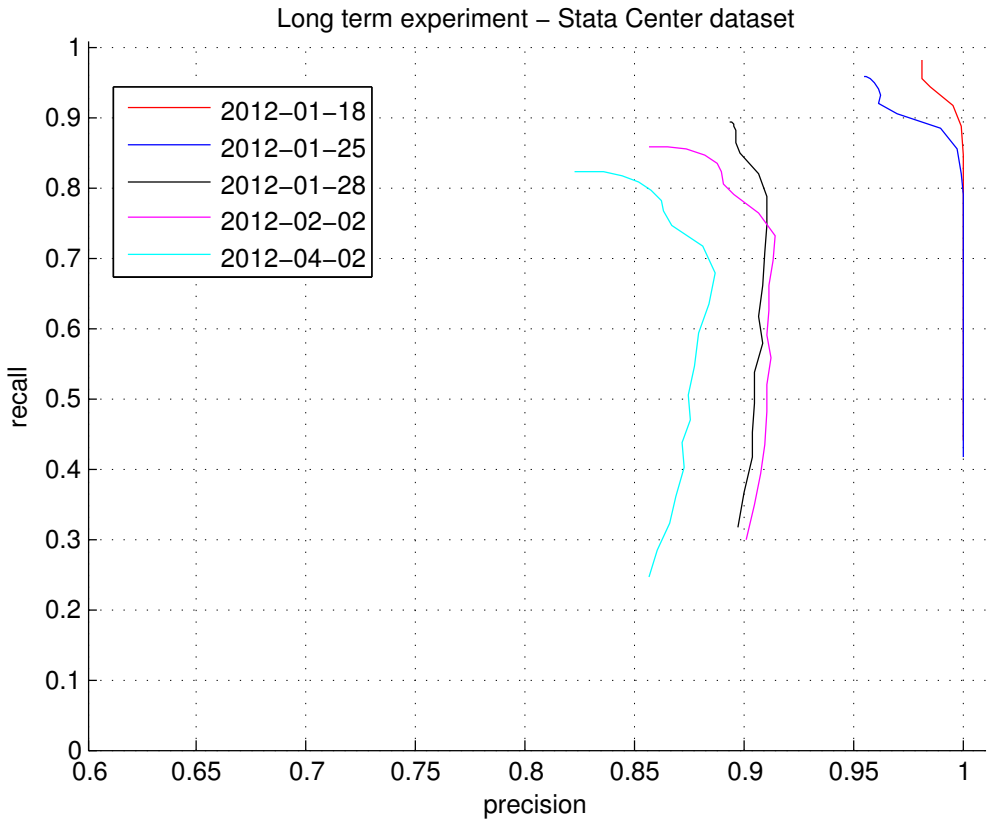


Figure 3.10.: Place recognition results using GRAPE based on a subset of the Stata Center dataset collection. Note that the axis scale and boundaries differ from the previous plots.

The run times for feature detection for GLARE/GFP are very similar since they share the same algorithms (see Section 3.4.1). The feature detection and description phase of GRAPE takes slightly longer than GLARE/GFP since generating the local normal surface primitive map is more time consuming. It is apparent that the geometric verification for GRAPE/GLARE is significantly faster than GFP. This is due to the fact that the relative distances of landmarks allow to reject more false-positive feature correspondences than the appearance based descriptors of GFP. We observed that the majority of features on the Kenmore dataset refer to point-like features resulting in very similar GFP descriptors. Thus the number of putative correspondences passed to RANSAC is smaller for GRAPE/GLARE compared to GFP. Note that GRAPE already achieved very accurate results for $K = 50$ (see Figure 3.9), which would substantially reduce the run time needed for geometric verification.

Discussion

It is obvious that GRAPE outperforms both GFP and GLARE on all datasets in the first experiment. We observed that GFP gets closer to GRAPE with a high number of K nearest

neighbours taken into consideration which again confirms the results we achieved in previous experiments in Section 3.3. The differences in precision/recall for GRAPE are rather small for increasing number of K. This number essentially determines the run time of the place recognition since the geometric verifications that need to be carried out for K scans are computationally expensive. Even though GLARE already achieves promising results, the surface primitives of GRAPE further help distinguishing similar places. Particularly in large corridors or rather empty rooms GRAPE outperforms GLARE since the latter lacks the availability of sufficient landmarks. GRAPE considers all occupied spaces regardless of their curvature characteristics. Even in outdoor environments such as parks or forests GRAPE remains close to GLARE since trees are detected as surface primitives and their beneficial geometrical relations can be taken into account as well. The second experiment quantitatively shows GRAPEs performance for long-term place recognition. The recognition is slightly worse than for the first experiment which is mainly due to the fact that the robot is likely to move further from the reference paths given by the first dataset and viewpoints have likely changed. Secondly, it is obvious that the recognition performance slightly drops with increasing time difference to the reference dataset which is due to structural changes in the environment, e.g. office interiors being moved and doors being closed. This problem has to be tackled differently by modeling changes. However, this is not explicitly taken into account by neither GRAPE nor GLARE.

3.5. Chapter Conclusions

This chapter introduced the algorithms Geometrical Landmark Relations (GLARE) and Geometrical Relations of Surface Primitives (GRAPE) designed for place recognition in 2D range data. Both model relative landmark or primitive relations captured from single range scans being transferred to scan signatures. Thereby we are able to implement the scan retrieval by means of an approximate nearest neighbour search avoiding expensive one-by-one matchings.

The state of the art uses appearance-based methods generating high-dimensional descriptors around landmarks which refer to points of high curvature. The presented algorithms fundamentally differ from this since they focus on generating descriptors of the spatial configurations of co-occurring landmarks or surface primitives. Rather than just being utilized for consistency checks, the geometric settings themselves become a feature.

GRAPE was shown to perform better than the state-of-the-art approach GFP for both, indoor and outdoor environments. However, for GLARE this applies only to the latter. GLARE and GRAPE omit prior training stages such as generating a vocabulary which becomes an indispensable feature for applications in a-priori unknown environments.

Our novel place recognition algorithms are supplemented with a geometric verification which allows to estimate a relative transformation of two corresponding landmark sets. This provides valuable input for a global localization of a mobile robot and SLAM algorithms. The latter benefits from the fundamental that transformations are provided in the

coordinate frame of the respective range scans rendering a straightforward integration into graph-based SLAM frameworks possible.

Thanks to the more generic descriptive power of surface primitives, GRAPE provides an outstanding recognition performance for a multitude of environment types ranging from suburbs, parks, hallways, offices which was demonstrated on the datasets kenmore, victoria park, stata center and intel respectively. Thanks to the elliptic shapes of the surface primitives, GRAPE is able to automatically adapt to dominant features in these environments, as for instance trees, walls or doors. We expect this to be a valuable contribution towards more environment specific descriptions of places serving as input for SLAM and map-based localization.

Geometrical relations were shown to be superior in outdoor environments which confirms the intuition of rather random distribution of landmarks. Based on our experimental results we can infer that these also perform better in indoor environments when using surface primitives which was not expected since man-made buildings typically consist of numerous symmetric structures. We assume that the increased recognition also results from the lack of landmarks which cause the worse performance of common appearance-based methods such as GFP.

Chapter 4.

Generic Simultaneous Localization and Mapping using 2D Range Data[1]

[1] Parts of this chapter have already been published in [74]

4.1. Motivation

A mobile robot requires precise estimates about its position and orientation with respect to a fixed reference frame in order to ensure a robust autonomous operation. This can easily be enabled by the use of artificial landmarks which, however, entails a significant setup expenses and provides limited flexibility. The state of the art utilizes prior maps of the environment being generated based on-board sensors which allows to omit the setup of artificial landmarks. If we assume that the trajectory driven by a robot is known, a map be generated straightforward. However, prior maps are typically used for localization if there is not any global reference system available. Particularly in indoor environments they are usually unavailable. If a robot misses both, a map of the environment and a position reference, it has to concurrently estimate its position and a map of the environment which is commonly referred to as Simultaneous Localization and Mapping (SLAM) problem [160].

An important requirement for SLAM algorithms in mobile robotics is its ability to estimate the traveled path and a map at high frequency. When using SLAM for autonomous exploration, the navigation algorithms utilize the provided map for generating new goals. In order to estimate paths to these, planning algorithms need the knowledge about free space and present obstacles in close proximity. The runtime requirement also addresses the initial mapping in conjunction with manual steering. Here, the human operator can be supported by providing information about subspaces of the environment which have already been visited. Also, it enables to guide the driver with the goal of preventing avoidable uncertainties. If, for example, the driver stops the robot immediately after closing a large loop, he can be advised to continue driving in order to further minimize the robot pose uncertainty by incorporating additional loop closures after entering a known part of the environment. It can further be assumed that the usability of the system's human machine interface is substantially improved if the user is constantly shown the latest map state and the overall map quality. A second requirement for our SLAM algorithm arises from the types of environment being considered. The application of service robots for logistics, museums or outdoors commonly implies working spaces of increased size which makes scalable algorithms indispensable. This particularly entails that all SLAM components have to work in sub-linear time with respect to the path length. Loosely speaking, the algorithms are supposed to process new information regardless of whether the robot moves in a small lab or a large warehouse. The larger the environment the more likely we will be faced with recognition challenges such as perceptual aliasing. In warehouses, for example, there can be found a large number of hallways surrounded by racks which might be hardly distinguished in neither their appearance nor their geometric settings. Also, office-like environments often contain many long corridors. The visual similarity and the symmetry in man-made buildings result in inevitable uncertainties when detecting loop closures. Thus, the SLAM algorithm has to be robust in terms of outliers while keeping a moderate balance of incorporating loop closures and avoiding diverged estimates. In the context of the initial setup of a mobile robot within its target working space, there are typically more tasks to be undertaken than just the generation

of a map. Often times the robot is also taught specific goals such as exhibits in museums or reloading points in warehouses being either automatically detected or manually entered. In the application phase the robot is expected to approach these goals in order to present specific information to visitors or to pick up ware goods. These use cases typically put high requirements in terms of the precision of the robot pose. Even if additional information such as the estimation of pallet poses or exhibit signs are utilized for reactive driving approaches, it is beneficial to firstly achieve a reasonable initial pose to minimize the amount of time bound for this process. Thus it can be summarized that we need maps with a high resolution and maximal local precision.

This chapter introduces a SLAM framework that is able to fulfill the above mentioned requirements. This includes a front-end providing spatial relations of robot poses by means of motions and loop closures. The latter are efficiently obtained using the place recognition algorithm GLARE (see Chapter 3). This information is passed to a back-end that maintains pose relations and estimates the path traveled by the robot. A robust optimization method is utilized in order to account for errors in the data association. In addition to that, we constantly generate an occupancy grid map of the perceived environment. The estimation of the path and the map is carried out close at a high frequency. Having completed the teach-in drive, we use a joint pose and map optimization method which is conducted offline based on the collected data. Thanks to this post-processing, we obtain a highly-accurate map for subsequent navigation tasks. We present experimental results carried out with two different robotic platforms: a museum tour guide and an automated guided vehicle. The data is obtained from different types of proximity sensors: laser range finders and one or multiple RGB-D cameras.

4.2. Related Work

One of the key questions that comes up when reading about robotic localization and mapping is the following: *Isn't SLAM already solved ?* This simple question entails a relatively complex answer. A simple *yes* or *no* is impossible since also many famous researchers controversially discuss about it [55]. The majority agrees that the answer depends on a number of conditions with the following being the key ones:

- Size of the environment
- Layout of the environment, e.g. cyclic, long corridors
- Amount and potential behaviour of dynamic objects
- Extent of systematic and non-systematic structural changes
- Kind of sensor, e.g. range-bearing or bearing-only
- Operating range and accuracy of range measuring sensors

It can be observed that these conditions are of high importance for SLAM while state-of-the-art algorithms often perform differently.

There has been established a fundamental amount of research in the field of SLAM in the last decades. The origins can be ascribed to the early work of Durrant-Whyte [45] and Leonard et al. [106] back in the late 1980s. Over many years researchers have investigated the understanding and the mathematical principles of the SLAM problem. In the last decade, the research has focused on the robustness and scalability of the algorithms. We will provide an overview of the research field highlighting fundamental algorithms in the following.

Feature-based SLAM

The first solutions to the SLAM problem are realized based on extended Kalman filters (EKF) [106]. All poses and landmark positions are stored in one state vector which is fully updated with each observation. Due to the complexity of $\mathcal{O}(N^2)$ with N being the number of landmarks, the computational burden of EKF-SLAM is high, particularly for increasing environment sizes [106]. This drawback of EKF-SLAM has been opposed by different strategies. For example Huang et al. propose to divide the landmarks into submaps with little overlap and run EKF-SLAM on these [83]. Having estimated the submaps, a subsequent submap joining is applied. Paz et al. presents a divide and conquer extension enabling to run EKF-SLAM with a complexity of $\mathcal{O}(N)$. Civera et al. utilizes an inverse depth parametrization to make EKF available for monocular SLAM.[34] Contrary to these approaches, Montemerlo et al. presents FastSLAM which factorizes the SLAM posterior into a product of conditional landmark distributions and trajectories [118]. Their key idea is to use a Rao-Blackwellized particle filter with each particle maintaining its own map.

Grid-based SLAM

In addition to feature-based SLAM, there has established a notable amount of work using occupancy grids as map representation. Due to their importance for navigation tasks, researchers have investigated approaches working directly with this representation including the data association. Fox et al. demonstrate the use of FastSLAM for grid maps [54]. Also Grisetti et al. utilizes Rao-Blackwellized particle filter in conjunction with adaptive proposals and selective resampling [64]. Their implementation *GMapping* has become one of the most famous mapping algorithms in the robotics community remaining state of the art for many years and also the default algorithm in middlewares such as ROS. An alternative system, *HectorSLAM* is provided by Kohlbrecher et al. [93]. In contrast to *GMapping*, their approach also enables 3D mapping and copes without odometry. *HectorSLAM* uses robust scan matching applied to grid maps of multiple resolutions. It poses an alternative for robotic platforms lacking wheel odometry.

Graph-based SLAM

A further class of algorithms is given by methods utilizing graph optimization for solving SLAM. The origins of these can be ascribed the early work of Thrun et al. [161] which are able to build maps based on data collections of urban environments within an offline optimization. An efficient solution using gradient descent is presented by Olson et al. [130]. A tree based algorithm which the authors refer to as TORO is introduced by Grisetti et al. [61]. Their approach is able to limit the graph optimization to the boundaries of the map being traversed rather then the length of the covered trajectory [61]. Kretzschmar and Stachniss propose to further improve graph-based SLAM by reducing the optimization complexity using methods of information theory [98]. Dellaert and Kaess introduce $\sqrt{SAM}$ which enables efficient online SLAM by smoothing the square root of the information matrix.

Blanco et al. investigate hybrid metric-topological SLAM which 8provides a generic solution for building hierarchical models of submaps designated for the operation in large-scale environments [18]. This renders the fusion of metric SLAM systems, as for example *RBPF* [64], and visual place recognition such as *FABMAP* [37] possible. The association and optimization is then carried out based on a graph of submaps [18]. Konolige et al. establish *Karto* which provides a grid-based front-end with efficient loop closure detection and graph-based optimization back-end.

In recent years, there have been established a number of generic graph optimization libraries which also provide a basis structure for SLAM algorithms. A famous representative is the library *g2o* [100] which is utilized by a lot of SLAM frameworks. Also HOG-MAN provides such a structure while enabling online operation with a limited update mechanism. Dellaert et al. make the library *gtsam* with interfaces for 2D and 3D SLAM available to the community [40]. An incremental solution enabling optimization based SLAM for online applications is given by iSAM [91]. The key idea of this approach is that only a small subset of the variables in a factor graph need to be updated for incremental operation. Kaess et al. further extended their algorithm enabling incremental variable reordering and re-linearization using a Bayes tree [90]. In [90] it is demonstrated that iSAM outperforms other state-of-the-art approaches as HOG-MAN [63] and Karto on real-world datasets. Several SLAM frameworks have been proposed that use *g2o* or TORO as back-end. In this line the work of Hartmann et al. results in a robust system that based on odometry and RGB-D data achieves promising results in indoor environments [56, 70]. The authors use an efficient appearance-based loop closure detection. Also Labbe et al. use appearance-based place recognition on top of TORO [102]. Both, [70] and [94], enable fast operation, an integration into the middleware ROS and the generation of occupancy grid maps being frequently updated. Alternatively to these approaches that optimize a sparse pose graph, Whelan et al. implement dense SLAM omitting the extraction of features. The high computational load is accounted for by using a GPU [168]. Mur et al. use the bundle adjustment interface of *g2o* to enable SLAM using solely monocular cameras [121]. Contrary to these approaches based on generic graph optimization libraries, Dubbelmann et al. make use of the algebraic structure of Lie groups to optimize chains of

pose online [43]. Their approach, coined *COP-SLAM*, achieves results being comparable to those for *g2o* but entails orders of magnitude less computational load.

Robust Optimization for Graph-based SLAM

The graph-based solutions to SLAM have significantly pushed the state of the art since it provides high efficiency also for large-scale environments while simultaneously achieving high accuracy. The application of SLAM for increasing environment sizes has also revealed the need for robust optimization in the presence of failures in the data association. The state of the art graph optimization irreversibly diverges in the presence of false loop closures, as demonstrated in [155]. Olson and Agarwal suggest the use of a mixture of Gaussians to model uncertainty in data association [129]. Sünderhauf and Protzel propose a robust optimization method enabling the optimization back-end to modify the pose graph by switching loop closure constraints. Agarwal et al. omit the use of additional variables [4] as it is done in [155]. Instead of switching constraints, the authors mitigate the contribution of outliers constantly by scaling the covariances of loop closure constraints [4].

Summary

We presented the most relevant work in SLAM and have shown the development path taken in the last two decades. Getting back to our question whether SLAM is already solved, we can point out a number of achievements. It can be observed that the majority of available algorithms is able to solve SLAM. Thus it is generally possible to localize in an unknown environment while concurrently building a map. Hierarchical feature-based and graph-based approaches have significantly pushed the algorithmic development in regards of scale enabling the operation in whole buildings, suburb or even cities. However, there are a number of open challenges remaining which can be derived from our survey. Many algorithms assume specific sensors or sensor classes in order to work properly. GMapping, for instance, requires laser range finders with high operating ranges. In future research, it is desirable to design SLAM algorithms in such a way that they become less sensor specific. To our knowledge, there is currently no SLAM framework available enabling the utilization of, for instance, RGB-D cameras and laser range finders. The state of the art focuses on a single sensor class even though the architecture of a subset is kept generic. The majority of those frameworks working with RGB-D cameras currently utilize the RGB data for loop closure detection rather than the depth data.

4.3. Framework Overview

Similar to a multitude of other SLAM frameworks, we utilize a number of state-of-the-art methods which are combined with our own algorithms. Our system is divided into a front-end and a back-end. This division is advantageous since it allows to easily replace

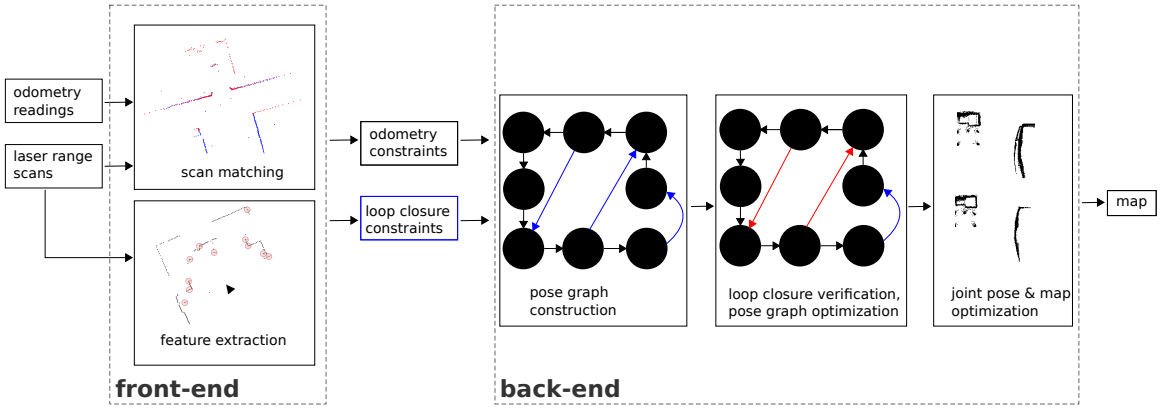


Figure 4.1.: This figures shows an overview of our SLAM framework.

modules, e.g. using different place recognition algorithms or sensors without having to adapt the optimization which is part of the back-end. The front-end is provided all relevant sensor data, which in our case is the odometry obtained from wheel encoders and 2D range scans. It forwards odometry and loop closure constraints to the back-end which in turn generates a pose graph from this information. The graph is optimized and along with the range scans used to render an occupancy grid map. At the end of the mapping process, an additional map optimization is executed in order to maximize the precision of the final map. Our framework is illustrated by Fig. 4.1. All components are exhaustively described in the following sections.

4.4. The Mapping Front-end

This section introduces the front-end of our SLAM framework. It describes the generation of initial pose estimates of sequential robot poses. Moreover the front-end contains a place recognition system providing potential loop closures to the back-end. Each of the components will be detailed in this section.

4.4.1. Initial Estimate

The relative transformation of the consecutive robot poses x_i and x_{i+1} of the trajectory is estimated based on wheel encoder readings. Assuming a mobile robot moves in a 2D space, its state vector can be expressed by $\mathbf{x} = (x, y, \phi)^T$. The motion estimation based on odometric measurements is subject to several error sources. First, the wheel properties, particularly the tire pressure and hence the wheel diameter can change over longer periods of time. In addition to that the wheels might slip, for example on a wet or slippery floor. These errors commonly entail a notable drift in the odometric motion estimation which can accumulate significantly over time.

4.4.2. Extracting Range Scans from RGB-D Data

If we are already given range data $s(b)$ of a 2D laser range finder, this section can be skipped. For RGB-D data a projection of a 3D input point cloud D onto a 2D plane has to be accomplished. A top-down projection of D is generated by converting each point $p^{(k)}$ of D from Cartesian to polar coordinates as follows:

$$\begin{pmatrix} \theta \\ \rho \end{pmatrix}^{(k)} = \begin{pmatrix} atan2(p_y^{(k)}, p_x^{(k)}) \\ \sqrt{(p_x^{(k)})^2 + (p_y^{(k)})^2} \end{pmatrix} \tag{4.1}$$

where $\theta^{(k)}$ refers to the bearing and $\rho^{(k)}$ to the range of the point k relative to the camera origin. We further introduce θ_{hfov} as the entire camera's horizontal field of view and θ_{cone} as the width of one measurement cone. Either of the parameters can be estimated straight-forward based on camera calibration parameters. Given the horizontal field of view of the camera θ_{hfov} and the image sensor's resolution α_w, we calculate the width of one measurement cone as follows:

$$\theta_{cone} = \frac{\theta_{hfov}}{\alpha_w} \tag{4.2}$$

Depending on the depth measuring sensor used and the resulting density of the obtained point cloud, it is recommendable to either exclude cones without depth values or interpolate the depth data with a Gaussian kernel around these locations.

Let us assume that $\theta^-_{hfov,b}$ and $\theta^+_{hfov,b}$ refer to the boundaries of the measurement cone b. Then a vector of point indices s with the minimal values $s(b)$ can be calculated:

$$s(b) = \min_{\theta^-_{hfov,b} \leq \theta^{(l)} < \theta^+_{hfov,b}} (\rho^{(l)}) \,,\, l \in D \tag{4.3}$$

In this way we obtain the contour $s(b)$ consisting of points with each being closest to the camera origin inside a cone b. The range scan $s(b)$ is henceforth

4.4.3. Scan Matching

The aforementioned drift in the pose estimate can be reduced by matching range scans of successive robot poses. We utilize PL-ICP which is a variant of the algorithm iterative closest points (ICP) as presented in [30] (see also Fig. 4.2). First, the two input scans are searched for corresponding points. Censi exhaustively describes several methods for this search in [30]. They differ from our methods presented in Chapter 3 in that they retrieve correspondences within a certain search area around candidate points bounded by distance and angular constraints. The correspondences o_j and r_k originated from the observation scan o and the reference scan r respectively are stored as $c^{\{i\}}_{j;k}$ in the set C. Given the correspondence set, the goal of ICP is to find a rigid transformation $T = (\mathbf{R}, \mathbf{t})$ which is used to project the observation range scan o into the coordinate frame of the reference scan r. Ideally, the reference and the projected observation scan

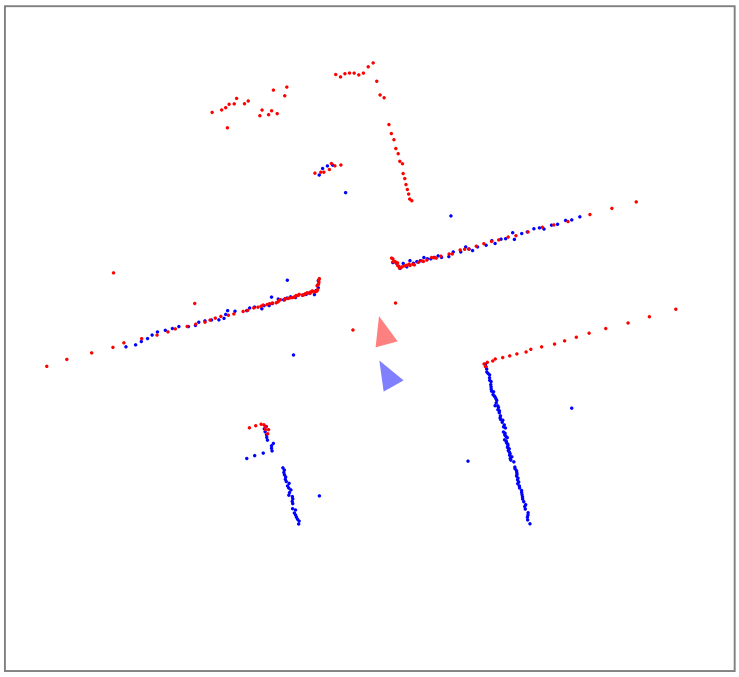


Figure 4.2.: **ICP**. This figure shows an observation scan (blue) which is matched to a reference scan (red) using iterative closest points (ICP) to find a rigid transformation. The origins of the scans are shown as blue and red triangles respectively.

$\hat{o}$ have a maximal overlap after applying T. This process is iteratively executed while applying the following objective function:

$$E\left(\mathbf{R}, \mathbf{t}\right) = \frac{1}{N_r} \sum_i \left(\mathbf{n_i^T} \left[r_i - \mathbf{R}\, o_i - \mathbf{t}\right]\right)^2 \tag{4.4}$$

Here n_i^T denotes the normal vector to the surface defined at r_j at the projected point of o_k. The variable R refers to the rotation matrix which is defined in 2D as:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{4.5}$$

Compared to other ICP algorithms, PL-ICP makes use of a point-to-line metric which enables fast convergence. This results in a runtime speed-up of up to 40 times compared to common ICP variants [30].

In contrast to feature-based scan matching as used in GLARE (see Chapter 3), ICP is more sensitive to local minima. Thus is does not provide robust estimates in the presence of large displacements which is mainly due to the correspondence search. However, thanks to the use of all measurement points rather than solely features, ICP can achieve a higher precision which is its main objective within our framework. Thanks to the odometry we are given a reasonable prior for scan matching which significantly reduces the risk of local minima.

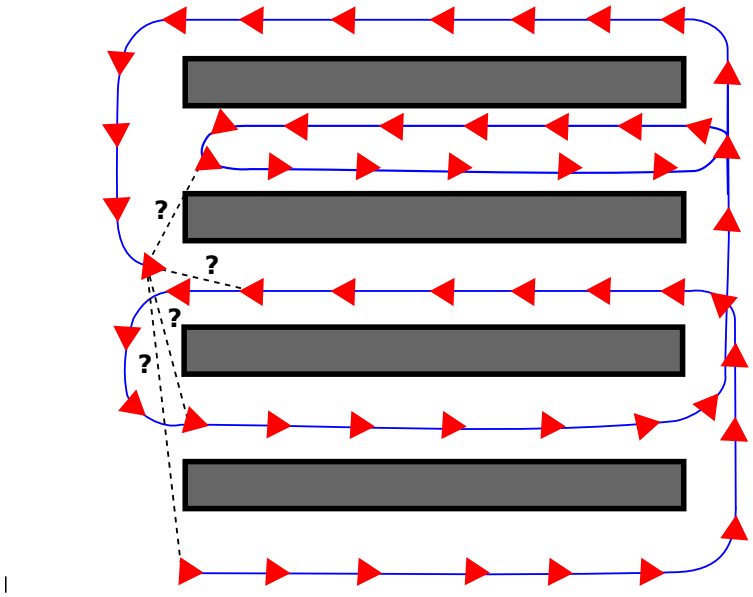
The crucial step of ICP is the correspondence search. As we have learned in our previous chapters, this generally poses a difficult association problem. Our place recognition algorithms GLARE and GRAPE extract distinctive features from the range scan which are matched using RANSAC in order to solve the correspondence problem. ICP uses different strategies for this. For each point r_i of the reference scan we search for a corresponding point in the projected observation scan $\hat{o}$. Thanks to the use of 2D range data, it can be exploited that points have a radial ordering. Different heuristics are used to obtain an efficient search, in particular to find suitable starting points and stop criteria. Censi proposes to initialize the search at the scan index of the previous correspondence. The stop criteria, that is the size of search window, is set according to the odometric prior. The corresponding scan index j refers to that point of $\hat{o}$ having the minimum distance to one point r_k:

$$j = \underset{j}{argmin} \left(||\hat{o}_j - r_k|| \right) \tag{4.6}$$

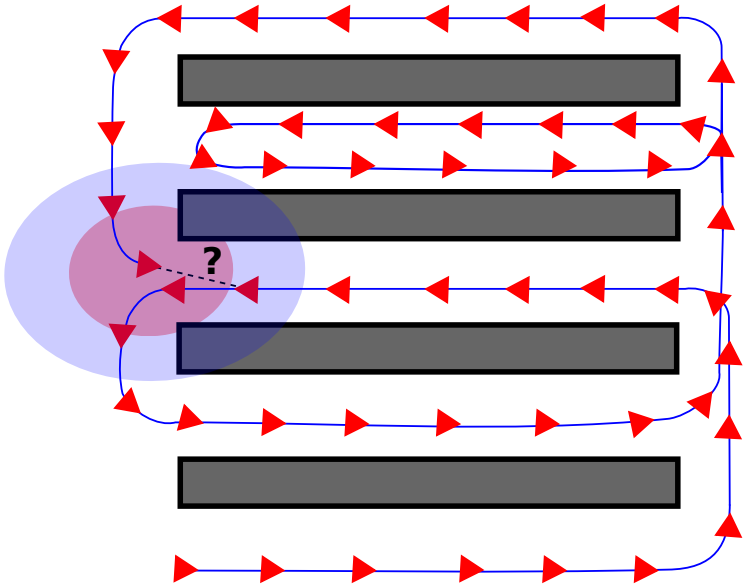
Note that only correspondences $c_{j;k}$ with a distance below a threshold τ_{icp} are considered for further processing. Algorithm 3 provides an overview of the core processing steps of ICP. A more exhaustive derivation of the utilized scan matching algorithm and correspondence search can be found in [30].

Algorithm 3 icp_matching $(r, o, \mathbf{T_{init}})$:

$\mathbf{T} = \mathbf{T_{init}}$
loop until convergence
    Project observation points into coordinate frame of reference
    $\hat{o} = apply\_transform(o, \mathbf{T})$
    Find corresponding scan points
    $C = find\_correspondences(r, \hat{o})$
    Filter false correspondences
    $\hat{C} = remove\_outlier(C)$
    if $size(\hat{C}) \geq N_{min}$ **then**
        Minimize error function according to Eq. (4.4)
        $(\mathbf{T}, \epsilon) = estimate\_transform(\hat{C})$
        Check residual error of transform
        if $\epsilon > \epsilon_{max}$ **then**
            break
        end if
    else
        break
    end if
end loop
return $\mathbf{T}, \epsilon$



(a) Loop closure detection incorporating all visited places.



(b) Loop closure detection with restricted search.

Figure 4.3.: This figure illustrates two different possibilities for determining which loop closures are incorporated in a simulated scenario. A SLAM graph is built while traversing the environment (Poses are red, odometry constraints blue and loop closure constraints are dotted black). Either of the methods search all previously observed places by matching their signatures. Variant (a) generates loop closure constraints for all matching places ignoring their spatial positions in the graph. Variant (b) constantly estimates the pose uncertainty which is utilized to define a search range for refining the loop closure candidates being spatially relevant. The search range is proportionally scaled with the pose covariance. Given an unlimited pose uncertainty, variants (a) and (b) provide equal results.

4.4.4. Loop Closure Detection

A key component of any SLAM system addresses the ability to recognize previously observed places in the environment which are referred to as loop closures. Thanks to these a SLAM algorithm is able to account for pose uncertainties accumulated over parts of a trajectory. In our SLAM framework we make use of the algorithm Geometrical Relations of Surface Primitives (GRAPE; see Chapter 3). Due to its increased robustness in indoor environments, GRAPE is the preferred choice, rather than GLARE. In Section 3 we explained the search for the K scans being most similar with the place signatures S. This retrieval includes places of the entire database ignoring their spatial positions to enable the place recognition to be used for initial localization. When utilizing this algorithm for loop closure detection within SLAM, we can benefit from the pose prior being constantly estimated in order to restrict the search to relevant regions that have been visited. Therefore we slightly modify the candidate retrieval as follows:

$$L = \min_{k}[dist_S(g_i, S)],\ k \leq K \tag{4.7}$$

with L describing the set of loop closure candidates being the k nearest neighbours of the query g_i with respect to all generated place signatures S. Our restricted search matrix $\hat{\mathbf{C}}_\mathbf{i}$ is obtained from the marginal covariance matrix $\mathbf{C}_i$ associated with the current pose x_i and a constant scale factor d_{sc}:

$$\hat{\mathbf{C}}_i = d_{sc} \cdot \mathbf{C}_i \tag{4.8}$$

The loop closure search is then restricted based on the Mahalanobis distance from the scaled covariance matrix $\hat{\mathbf{C}}_i$:

$$dist(x_i, D, x_j) = [(\mathbf{x_j} - \mathbf{x_i})\, \mathbf{D^{-1}}\, (\mathbf{x_j} - \mathbf{x_i})]^{\frac{1}{2}},\ x_j \in L \tag{4.9}$$

The set of loop closure candidates L being obtained based on matching the place signatures is refined based on the pose prior which results in the set $\hat{L}$ with $\hat{L} \subset L$:

$$\hat{L} = \left\{ dist(\mathbf{x_i}, \hat{C}_i, \mathbf{x_j}) < \tau_{lc}, \mathbf{x_j} \in L \right\} \tag{4.10}$$

A loop closure constraint is kept in the remaining set $\hat{L}$ if the pose difference is below a threshold τ_{lc}.

Our loop closure retrieval incorporates two criteria: First, the similarity measure of the place signatures and second, the spatial distances to loop closure candidates based on the uncertainty of the current pose estimate. It is possible to frequently run the place recognition considering corresponding places of the entire database. However, it potentially entails an increased number of false positive associations, particularly in environments with many repetitive structures. Even though, our framework is able to handle false loop closure constraints, the risk of divergence and errors is increased while simultaneously making the optimization more complex. Other approaches such as Karto [94], use an a-priori fixed radius to restrict the loop retrieval. This is crucial when closing large loops

as pose priors might deviate significantly from the true pose due to accumulated odometric errors. The radius is hard to set and loop closures are likely to be missed. Setting a large radius, however, substantially increases the run time of Karto and potentially entails the incorporation of false loop closure candidates results. We found that a search with dynamic radius enables an optimal balance of limiting the number of false positive loop closure detections while simultaneously reducing the search and optimization run times. Providing a suitable statistical model for predicting the pose uncertainty C_i can be found, the risk of missing loop closure candidates is rather minor. The radius can be set quite optimistically. A certain amount of wrong associations can be compensated which is necessary since these are likely to occur at self-similar places. The restricted search minimizes these uncertainties but cannot generally avoid them. In the following sections we will learn how remaining false loop closure constraints can be identified and circumvented in the optimization.

4.5. The Optimization Back-end

Based on the information provided by the front-end we can build an initial graph of robot poses and edges with each describing a spatial constraint for two poses. The task of the back-end is to optimize this graph. This section gives a brief introduction to pose graph SLAM. Subsequently it is shown how data association errors are accounted for and a map of the environment is generated.

4.5.1. Pose Graph SLAM

Pose graph SLAM optimizes robot poses $\mathbf{x_i}$ of a given trajectory. The graph consists of vertices expressing the robot poses $\mathbf{x_i}$ and edges describing the relative spatial configuration of the poses $\mathbf{x_i}$ and $\mathbf{x_j}$. The motion of a robot $\mathbf{u_i} = \Delta(x, y, \phi)$ is accomplished which enables a transition from the state $\mathbf{x_i}$ to $\mathbf{x_{i+1}}$. This action is incorporated by a motion model as:

$$\mathbf{x_{i+1}} \sim \mathcal{N}(f(\mathbf{x_i}, \mathbf{u_i}), \Sigma_i) \tag{4.11}$$

Actions can also be extended from consecutive poses to loop closures detected for $\mathbf{x_i}$ and $\mathbf{x_j}$ by $\mathbf{u_{ij}}$. This results in the following condition for state $\mathbf{x_j}$:

$$\mathbf{x_j} \sim \mathcal{N}(f(\mathbf{x_i}, \mathbf{u_{ij}}), \Lambda_{ij}) \tag{4.12}$$

Providing the entire set of states X and actions U we estimate the maximum a posteriori (MAP) of robot poses X^* for the joint probability distribution

$$X^* = \underset{X}{\operatorname{argmax}}\, P(X|U) \tag{4.13}$$

We therefore make use of the a factorization, such that:

$$P(X|U) \propto \prod P(\mathbf{x_{i+1}}|\mathbf{x_i},\mathbf{u_i}) \cdot \prod_{ij} P(\mathbf{x_j}|\mathbf{x_i},\mathbf{u_{ij}}) \tag{4.14}$$

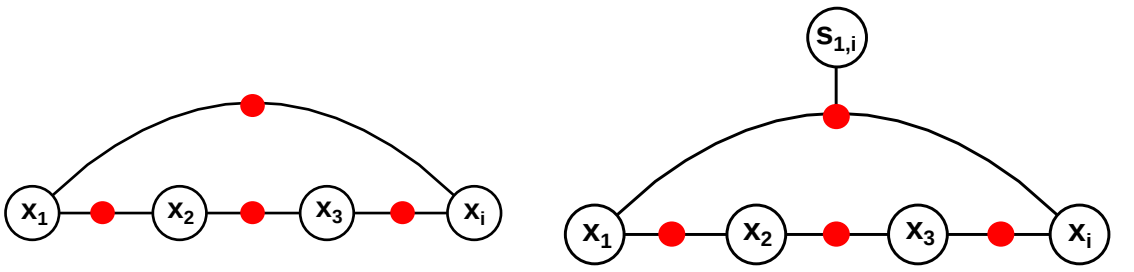
Here, $P(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_i)$ describes odometry constraints and $P(\mathbf{x}_j|\mathbf{x}_i, \mathbf{u}_{ij})$ loop closure constraints respectively. Based on our objective function of Eq. 4.13 we can describe our pose graph optimization in terms of a nonlinear least squares problem following the initial ideas of [42]:

$$X^* = \underset{X}{\operatorname{argmin}} \sum_i \left\| e_i^{odo} \right\|^2_{\sum_i} + \sum_{ij} \left\| e_{ij}^{lc} \right\|^2_{\Lambda_{ij}} \tag{4.15}$$

with $e_i^{odo} = f(\mathbf{x_i}, \mathbf{u_i}) - \mathbf{x_{i+1}}$ and $e_{ij}^{lc} = f(\mathbf{x_i}, \mathbf{u_{ij}}) - \mathbf{x_j}$. This least-squares problem can be solved using Gauss-Newton. A more extensive derivation is given by [155].

4.5.2. Robust Optimization

Common algorithms such as [42] using least squares based optimization for SLAM as stated by Eq. 4.15 assume that the data association maintains a pose graph solely consisting of valid constraints. This entails that the structure of the graph is fixed during optimization supposing all loop closure constraints are correct. This assumption sets up high demands for the place recognition which likewise would be parametrized rather conservative to avoid mistakes. Thanks to geometric verification, e.g. using RANSAC, as described in Chapter 3, a large number of failures in place recognition can be correctly removed. However, the presence of repetitive structures, particularly in large-scale environments, commonly entails an increased amount of places being similar in both, appearance and geometry. Thus the occurrence of false loop closure detections is likely and should be expected.



(a) Conventional graph-based SLAM (b) Graph-based SLAM with switchable loop closures

Figure 4.4.: This figure illustrates conventional graph-based SLAM with poses being represented by variables (a). Both, odometry and loop closure constraints are threated equally in the optimization. (b) uses additional variables for loop closures which can be switched within the optimization.

A solution for the mentioned association errors is given by robust optimization that tolerates incorrect loop closures. Rather than just avoiding wrong loop closures, the goal is to mitigate their contribution within the optimization process. For this purpose we make use of *switchable constraints* as presented by Sünderhauf and Protzel [155]. Their key idea is a modification of the objective function and the utilization of factor graphs with the factors expressing switchable loop closure constraints. Thanks to this, constraints of a graph are not fixed and can be adjusted during optimization. Loop closures are described as switch variables s_{ij} and can be switched off. The expected confidence of a loop closure to the optimization is taken into consideration by its initial value γ_{ij} and the corresponding covariance matrix Ξ_{ij}. This allows to set the uncertainties of individual loop closures which can, for example, be provided by a place recognition algorithm. The switch priors are required in order to avoid all loop closures being switched off by the optimizer as exhaustively shown in [155]. Our objective function of Eq. 4.15 is extended by the switchable constraints as follows:

$$X^*, S^* = \underset{X,S}{\operatorname{argmin}} \sum_i \left\| e_i^{odo} \right\|^2_{\sum_i} + \sum_{ij} \left\| e_{ij}^{slc} \right\|^2_{\Lambda_{ij}} + \sum_{ij} \left\| e_{ij}^{sp} \right\|^2_{\Xi_{ij}} \tag{4.16}$$

with $e_{ij}^{slc} = \Psi(s_{ij}) \cdot (f(\mathbf{x_i}, \mathbf{u_{ij}}) - \mathbf{x_j})$ describing the switchable loop closure constraints and $e_{ij}^{sp} = \gamma_{ij} - s_{ij}$ the switch priors. The function Ψ allows to map continuous input numbers s_{ij} to the range of $[0, 1]$. A function value $\Psi(s_{ij}) \approx 0$ entails that the loop closure constraint s_{ij} is disabled. As recommended in [155] we utilize a simple linear switching function Ψ which can be expressed as follows:

$$\Psi(s_{ij}) = \begin{cases} 0 & s_{ij} < 0 \\ \frac{1}{a} s_{ij} & 0 \leq s_{ij} \leq a \\ 1 & s_{ij} > a \end{cases} \tag{4.17}$$

with parameter a being set as $a = 1$. The behaviour of the optimization can be guided by the selection of the switch function Ψ, the switch priors γ and their associated covariance matrices Ξ. An exhaustive mathematical derivation of *switchable constrains* is provided in [155].

4.5.3. Generating Occupancy Grid Maps

During online operation a map of the environment is built with new observations being constantly incorporated. For this purpose we make use of two different representations, the *occupancy grid maps* and the *binary grid maps*. Either of them are described in the following.

4.5.3.1. Probabilistic Grid Maps

Probabilistic occupancy grid maps are the most common representation for mobile robot navigation. They provide a generic structure which can be used for various range sen-

sors. Each grid cell m_j comprises an occupancy probability p_{m_j} with $p_{m_j} \in \mathbb{R}$ and $0 \geq p_{m_j} \leq 1$. As exhaustively described in [160], solving the full posterior over maps $p(m \mid z_{1:t}, x_{1:t})$ at once is impracticable, thus we use an approximation based on the product of the marginals as $p(m_j \mid z_{1:t}, x_{1:t})$. In order to avoid numerical instabilities we convert probabilities to log-odds $l_{t,i}$ and continue working with these:

$$l_{t,i} = \log \frac{p\left(m_j \mid z_{1:t}, x_{1:t}\right)}{1 - p\left(m_j \mid z_{1:t}, x_{1:t}\right)} \tag{4.18}$$

Based on the spatial boundaries of our current graph configuration, we generate an occupancy grid map with each grid cell being initialized as $p(m_j) = 0.5$. In this we threat the space covered by each unvisited cell as unknown. The end points of each observation beam are transformed into the map coordinate frame. Therefore we estimate the corresponding grid cell m_j for each endpoint $z_t^{(l)}$ of the observation z_t:

$$\begin{pmatrix} m_{j,x} \\ m_{j,y} \end{pmatrix} = \lfloor \left[\begin{pmatrix} z_{t,x}^{(l)} \\ z_{t,y}^{(l)} \end{pmatrix} - \begin{pmatrix} c_x \\ c_y \end{pmatrix} \right] \tau_{res}^{-1} + \frac{1}{2} \begin{pmatrix} s_x \\ s_y \end{pmatrix} \rfloor \tag{4.19}$$

Here c_x and c_y refer to the origin of the map which is given by the graph's first pose $\mathbf{x}_1$. The parameters s_x and s_y describe the size of the environment and τ_{res} the resolution of the grid. Each measurement beam is projected in the grid map. If a beam hits an endpoint, then the probabilities of all grid cells being traversed by the beam are decremented. The cell's probability being actually hit by the beam is incremented. Likewise these values are decremented if a measurement beam does not hit an obstacle. In this case we update all cells along the beam within the sensor's operating range. The ray casting algorithm is exemplarily visualized by Fig. 4.5. A more exhaustive explanation of this is given by [160].

As already mentioned we update cell probabilities in the space of log-odds (see Eq. 4.18). Once the mapping is completed, we transfer these back to probabilities as shown in [160].

Since numerous algorithms require a discrete belief about the state of a cell, we further introduce the function $\varsigma\left(p\right)$ mapping continuous-valued probabilities to the discrete states *occupied*, *free* and *unknown*:

$$\varsigma\left(p_{m_j}\right) = \begin{cases} free & p_{m_j} \leq \tau_{free} \\ unknown & \tau_{free} > p_{m_j} > \tau_{occ} \\ occupied & m_j \geq \tau_{occ} \end{cases} \tag{4.20}$$

While new measurements are simply added, we have to rebuild the entire map given the corrected trajectory once a loop closure is conducted. This means that the ray casting algorithm is again applied to all observations.

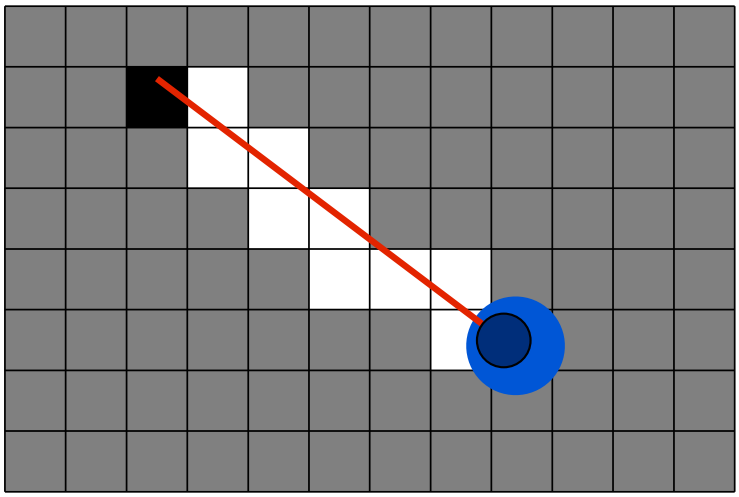


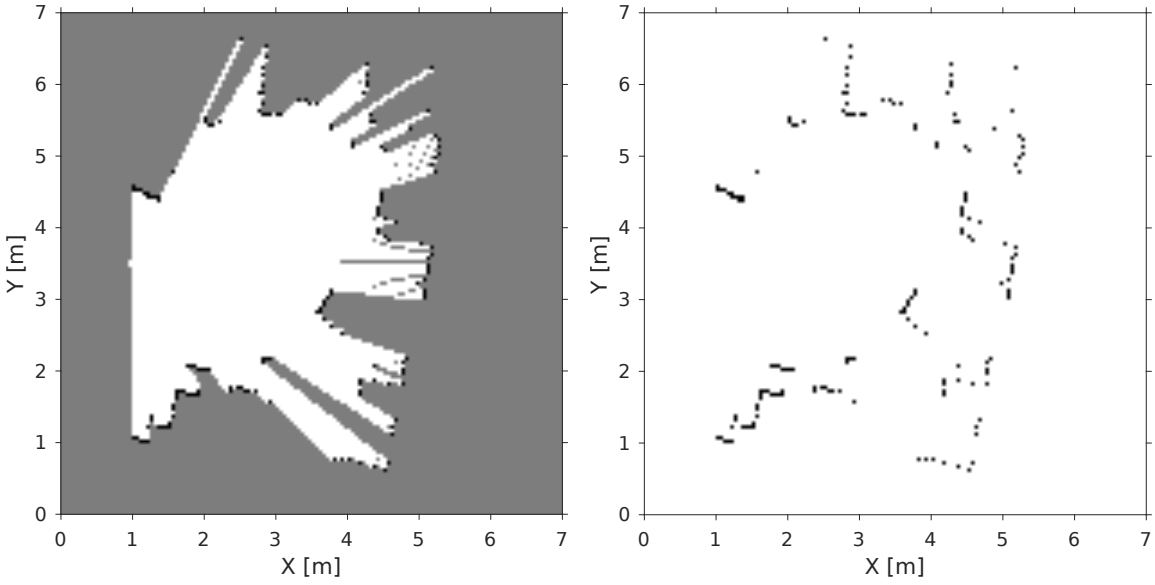
Figure 4.5.: This figure demonstrates the grid line traversal algorithm (ray casting) being used for occupancy grid mapping. A scan is captured using an onboard range sensor on a mobile robot (blue). A measurement beam is projected through the grid map (red). The algorithm traces the entire beam until it approaches the endpoint. This model assumes that those grid cells being traversed by the beam are free. The endpoint falls into a grid cell whose occupancy likelihood is incremented (black). The line traversal is applied to each measurement of the range scan.

4.5.3.2. Binary Grid Maps

Probabilistic grid maps explicitly model unknown space rather than just capturing obstacles. This, however, entails an extensive work load due to the ray casting which has to be carried out for each sensor beam. Thus it is recommendable to consider other map representations for live mapping. We therefore make use of binary occupancy grid maps with each grid cell m_j holding a binary state $p(m_j) \in \{free, occupied\}$. The state *unknown* is not included in this representation. The omitted differentiation of *free* and *unknown* space allows to significantly reduce the runtime. Endpoints of range measurements can be directly projected into the map without traversing individual sensor beams. This model can be referred to as an endpoint model similarly to the likelihood field for AMCL (see Section 2.1).

4.5.4. Post Map Optimization

The components of the framework presented in the preceding sections enable online SLAM with high performance. This enables to frequently provide the latest map state which supports human operators and serves as indispensable precondition when using the framework in the context of autonomous exploration.



(a) Probablistic grid map.

(b) Binary grid map.

Figure 4.6.: This figure shows a probabilistic and a binary occupancy grid map respectively generated using the same input laser range scan. The initial state of a probabilistic map is *unknown* with probability $p(m_i) = 0.5$ which refers to the color *gray* in the visualization. The binary map is initially set to $p(m_i) = 0$ and updated to $p(m_i) = 1$ for all cells being observed as occupied. The binary map does not distinguish between the cell states *unknown* and *free* which is the major difference to the probabilistic representation.

Pose graph SLAM allows to generate a globally consistent map with the local accuracy being reliant on the transformation estimation of the place recognition. Our RANSAC-based method estimates the 2D pose based on the detected feature correspondences not considering the uncertainty of range measurements. Incorporating this is computationally expensive and thus rather unsuitable. Thus we run the optimization once the online mapping process is finished. The graph is built while mapping the environment but the optimization is carried out subsequently using sparse surface adjustment (SSA) [139]. It was established by Ruhnke et al. and first published in [139]. SSA enables the concurrent refinement of sensor poses and raw range measurements. The algorithm is closely related to sparse bundle adjustment (SBA) which is commonly utilized in computer vision [69]. Given an image sequence, SBA aims at optimizing camera poses and 3D points. Specifically, the optimization is carried out through minimization of the reprojection errors for individual 3D points. SSA shares the idea of concurrent pose and scan point optimization, however, it requires point-to-surface correspondences while SBA incorporates point-point correspondences. This can mainly be ascribed the different sensor characteristics of camera sensors and LIDARs. SSA requires smooth surfaces in order to work

properly which is not the case in SBA. These properties can typically be found for indoor rather than outdoor environments.

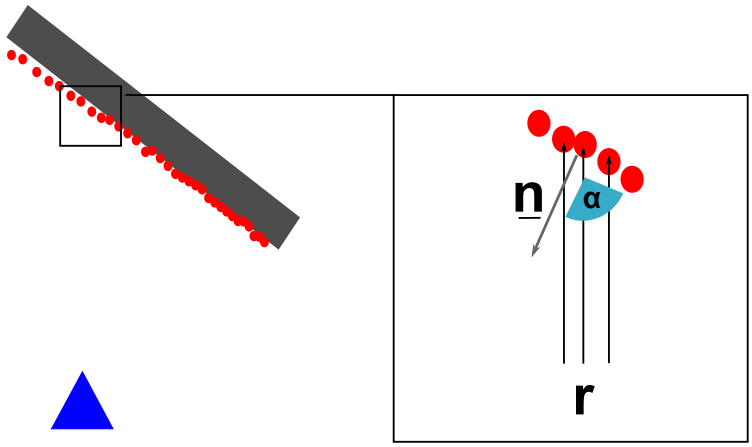
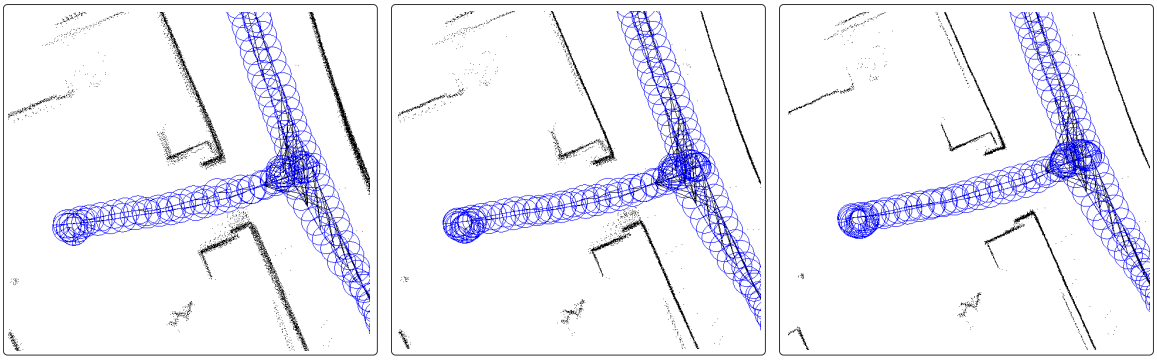


Figure 4.7.: This figure demonstrates sparse surface adjustment (SSA). The environment is assumed to consist of smooth surfaces. A measurement beam r hits a surface based on the time-of-flight (TOF) principle. The incident angle α depends on the sensor pose and the orientation of the target surface. The normal vector $\mathbf{n}$ of the surface is estimated and utilized for further processing. Given the smoothness assumption about the environment and the range sensor model, SSA aims at optimizing beam endpoints improving the accuracy of the raw range measurement. The origin pose is shown as blue triangle.

SSA differs from the previously described scan matching algorithm PL-ICP. ICP aims at minimizing the projection error of an observation with respect to a reference frame. The alignment is established by solely estimating a transformation in 2D. SSA, in contrast, optimizes range measurements individually which enables to also minimize projection errors with fewer rigidity constraints. A custom model incorporates sensor-specific uncertainties of range measurements of LIDAR and RGB-D sensors. Each single measurement beam describes a conic shape and is considered in the optimization. Therefore SSA estimates the normal of the surface being hit. The incident angle of the beam with respect to the surface significantly contributes to the range uncertainty. The larger this angle the more diffuse the reflection of the emitted light and thus the larger the error. Also the material and color of the surface might entail less reflection. The underlying physical model describing this phenomena is the Lambertian law. Incorporating surface properties such as material or color is complex and rather uncertain which is why it is omitted in SSA. Since the algorithm is expected to be generic in terms of the operating environment, it sticks to the estimation and explicit modeling of the incident angles. Therefore the range scan is transferred to a surface primitives with each beam being modeled by its tangent w.r.t. to the surface. These are described by Gaussians having a mean μ_{ik} and covariance $\sum_{ik}$. The mean is set according to beam k of pose i, the covariance based on the local

neighborhood. Surface primitives are slightly adjusted w.r.t. to the tangential direction and rigidly w.r.t. to the normal direction towards the sensor pose.



(a) Initial map (b) Optimization after 2 iterations (c) Optimization converged after 10 iterations.

Figure 4.8.: **SSA**. This figures illustrates the optimization of a submap using SSA. The endpoints of the measurements (black) as well as the robot poses (blue) are shown at different optimization stages.

The concurrent optimization of sensor poses and surface primitives M is described as [139]:

$$X^*, M^* = \underset{X,M}{\operatorname{argmin}} \sum_i \left\| e_i^{odo} \right\|_{\sum_i}^2 + \sum_{m,n} e_{lm}^{surf} + \sum_{i,k} e_{ik}^{meas} \tag{4.21}$$

where e_{mn}^{surf} refers to the tangential and normal error for all surface primitives m and n. The optimization term e_{ik}^{meas} binds a beam k to a sensor pose i. The objective function Eq. 4.21 is utilized for optimization based on Gauss-Newton. The goal is to minimize the distances between corresponding surface primitives. After each optimization step, the correspondence search is repeated and a new configuration based on this is prepared. An extensive derivation of the algorithm SSA is given by [139]. As a result of this post-optimization step we obtain a reconfigured set of poses and associated range measurements. The latter can be expected to be more accurate than the raw measurements, particularly for those having large incident angles with respect to the surface.

4.6. Experiments

This section provides an experimental evaluation of the presented SLAM framework. In particular, it consists of two parts.

The first one investigates the accuracy of the SLAM algorithm in terms of the estimated poses by measuring both, the global error and the drift becoming apparent within the continuous motion estimation. These experiments provide insights into the collaborative functioning of several SLAM components such as loop closure detection, graph

optimization and scan matching. The second part analyses the local accuracy of the final map. Event though this is biased by the pose accuracy, it aims at specifically measuring the contribution of the post map optimization for increasing the local map accuracy.

4.6.1. SLAM - Pose Accuracy

The following sections briefly introduce the evaluation metrics, the datasets being used for the experiments and discuss the individual results obtained for the evaluation of the pose accuracy.

Setup

We evaluated the presented framework based on a number of experiments with varying robotic platforms, proximity sensors, environment types and scales. Three out of four of the investigated datasets are publicly available and actively used for benchmarks within the SLAM research community. The datasets Stata and ITI are originated from a PR2 and PeopleBot robotic platform respectively. The Kenmore dataset was collected with a car, the FTF-Lab dataset with a reach truck. Table 4.1 provides an overview of the investigated datasets. For a more detailed description the reader is referred to Appendix A.1.

Dataset	Kind	Size $[m^2]$	Length $[m]$	# Poses	Range sensor	Public
Stata	Indoor	3542	716	2562	Laser	X
Kenmore	Outdoor	1,331,775	6577	13043	Laser	X
ITI	Indoor	5934	538	3280	Single RGB-D	X
FTF-Lab	Indoor	236	149	1529	Multi RGB-D	-

Table 4.1.: Overview of the investigated datasets.

Evaluation metrics

Our evaluation reveals the results obtained for estimating the traveled path of the robot using the presented SLAM algorithm. The error is quantified in terms of commonly used metrics which are recommended by the Rawseeds benchmark suite [52].

Absolute Trajectory Error (ATE). The ATE measures absolute distances of time-synchronized ground truth $\mathbf{G}$ and SLAM poses $\mathbf{X}$. For each ground truth pose we search for the closest poses in regards of the timestamps and estimate the pose differences $\mathbf{F}$ according to:

$$\mathbf{F}_i = \mathbf{G}^{-1}\mathbf{S}\mathbf{X}_i \tag{4.22}$$

with $\mathbf{S} \in SE(2)$ describing a rigid transform aligning $\mathbf{G}$ and $\mathbf{X}$ given the first 20 poses using ICP [30]. This metric is important for SLAM as it is a measure of global consistency

which is largely affected by loop closures. This measure is not only reliant on the sensor accuracy but also on the scale of the environment and the presence of loop closures. The ATE can be expressed as follows:

$$ATE\left(\mathbf{F}_{1:n}\right) = Median\left[trans\left(\mathbf{F}_i\right)\right] \tag{4.23}$$

Relative Pose Error (RPE). This metric is another method to evaluate the accuracy of SLAM algorithms or to be more general for any motion estimation algorithm (e.g. visual odometry). Similarly to the ATE, we require associations of ground truth to SLAM poses based on timestamps in order to estimate the relative poses $\mathbf{E}$:

$$\mathbf{E}_i = \left(\mathbf{G}_i^{-1}\mathbf{G}_{i+\Delta_{rpe}}\right)^{-1}\left(\mathbf{X}_i^{-1}\mathbf{X}_{i+\Delta_{rpe}}\right) \tag{4.24}$$

In contrast to the ATE, however, we evaluate the distances within a specified window of Δ_{rpe} subsequent poses as follows:

$$RPE(\mathbf{E}_{1:n}, \Delta_{rpe}) = Median\left[trans\left(\mathbf{E}_i\right)\right] \tag{4.25}$$

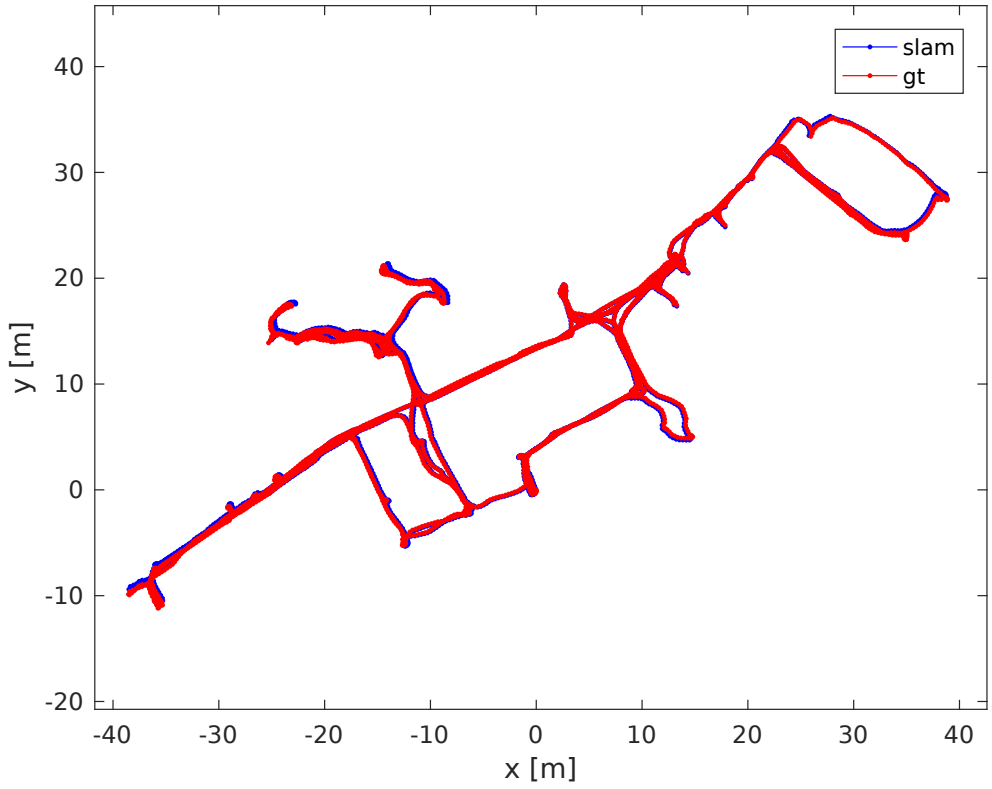
The RPE is a measure of drift that occurs in the continuous motion estimation. It is not reliant on the detection of loop closures. This metric can be parametrized by the windows size Δ_{rpe}. We follow the recommendation of Rawseeds [52] and average over varying window sizes which also mitigates the influence of outliers. In particular, the window size is varied in between $\Delta_{rpe} \in [10; 100]$. The results are calculated using the **m**ulti **r**elative **p**ose **e**rror function:

$$MRPE(\mathbf{E}_{1:n}) = \underset{\Delta_{rpe}}{Median}\left[RPE\left(\mathbf{E}_{1:n}\right)\right], \Delta_{rpe} \in [10; 100] \tag{4.26}$$

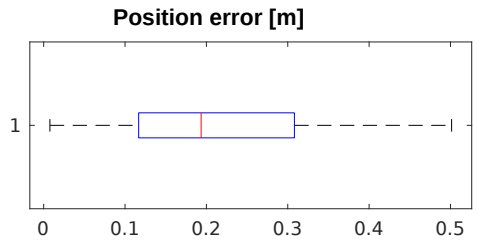
In our experimental evaluation we further calculate min, max, $mean$ and std. This simply replaces the $median$-function in the above mentioned equations for RPE, MRPE and ATE.

Results

We exhaustively present the results obtained for the individual datasets on the following pages.

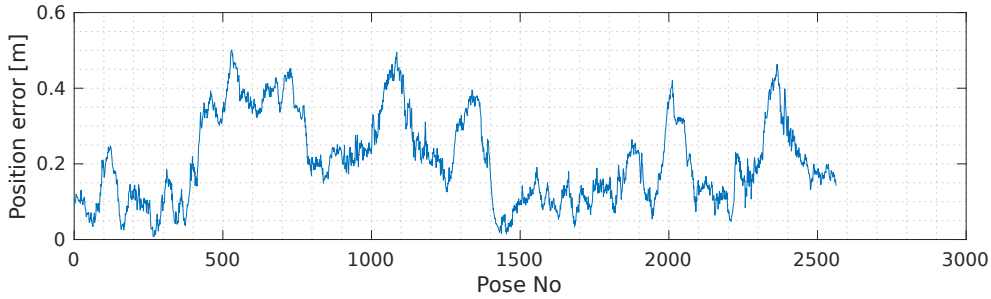


(a) Trajectory.

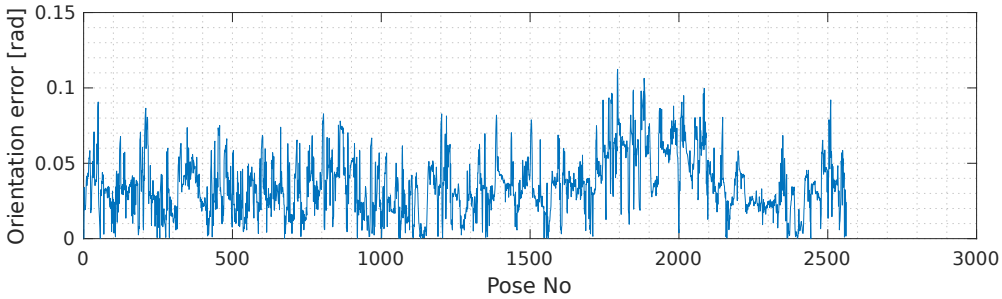


(b) Results.

Figure 4.9.: **Stata**. This Figure shows the experimental results obtained for the Stata dataset. Fig. (a) visualizes the estimated SLAM trajectory compared to the ground truth. The error is summarized in terms of a boxplot in Fig. (b). It can be clearly seen that the estimated path matches to the ground truth for the majority of the trajectory.



(a) Position error (detail).

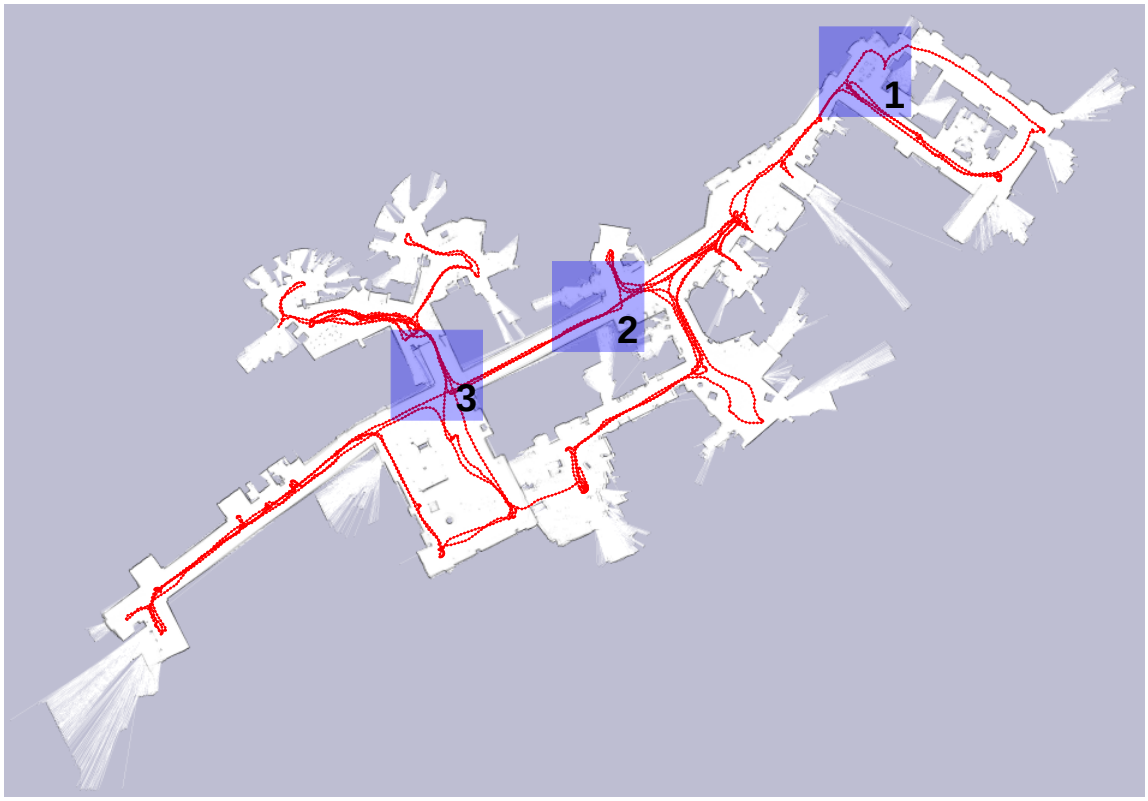


(b) Orientation error (detail).

		Mean	Median	Std	Max	Min
ATE	Position	0.21113	0.19352	0.11533	0.50141	0.0076624
	Orientation	0.035559	0.033562	0.019612	0.11231	0
MRPE	Position	0.0058611	0.0045181	0.0020432	0.10222	2.3909e-06

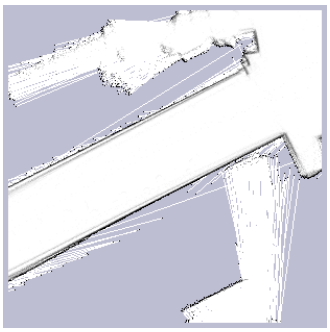
(c) Results.

Figure 4.10.: **Stata**. This figure provides an in-depth presentation of the position and orientation error. The position error is typically below $0.2m$ and never exceeds $0.5m$. The error increases the further robot moves from the center towards the building boundaries due to extended distances to loop closures and long corridor segments. The latter entail a increased uncertainty for scan matching.

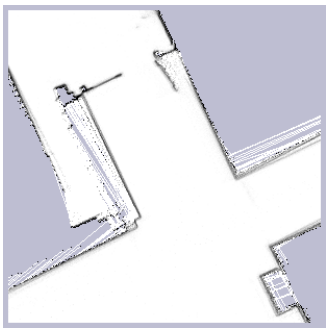


(a) Occupancy grid map and overlayed trajectory.

(b) Detail 1.

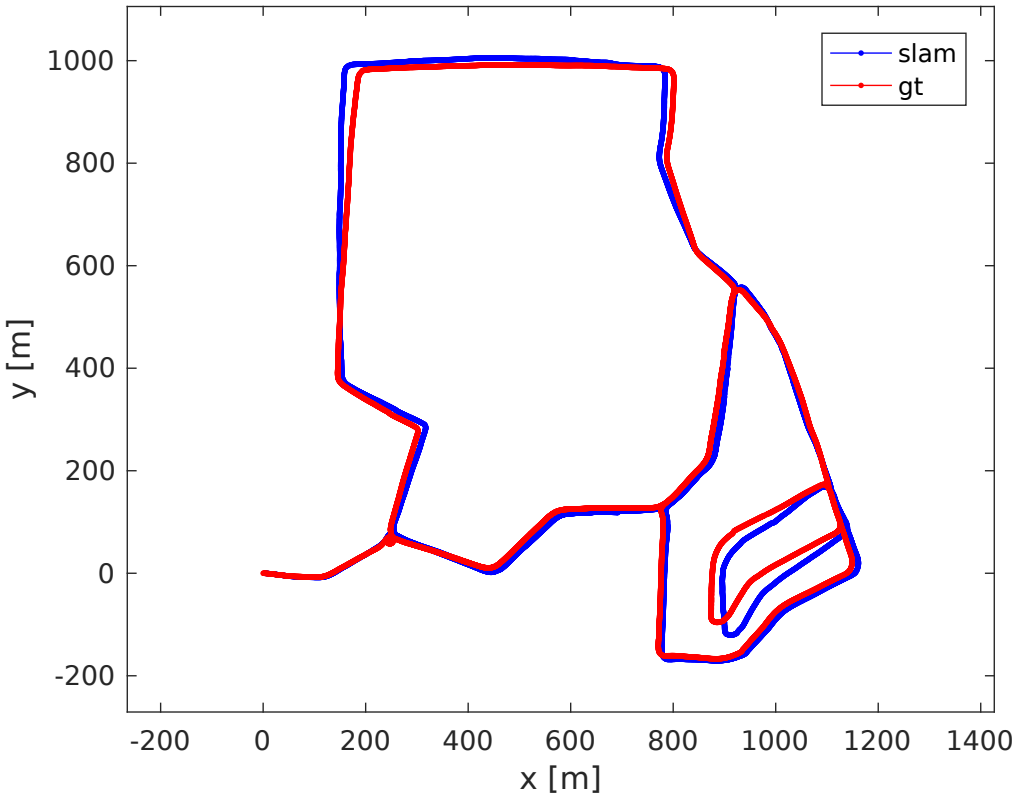


(c) Detail 2.

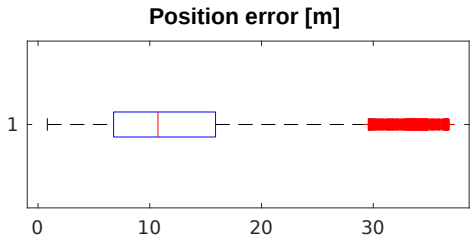


(d) Detail 3.

Figure 4.11.: **Stata**. Fig. (a) shows an occupancy grid map generated from the estimated SLAM trajectory. The map is free of ambiguities and with a correct global alignment. Fig. (b) - (d) show local details.

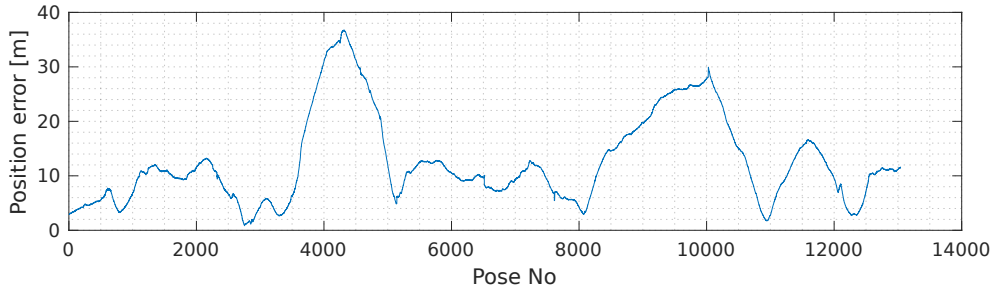


(a) Trajectory.

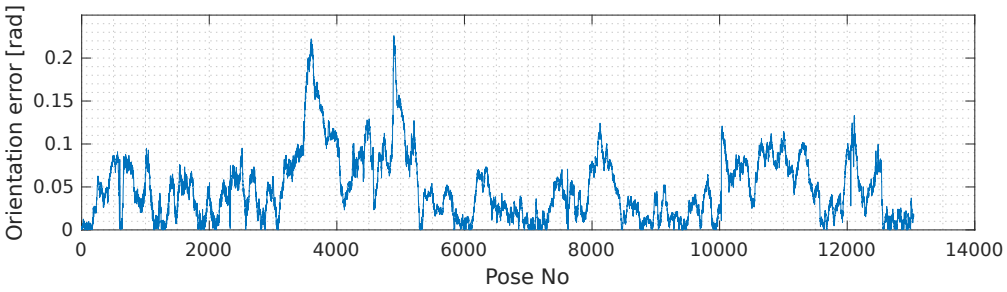


(b) Results.

Figure 4.12.: **Kenmore**. This Figure shows the experimental results obtained for the Kenmore dataset. Fig. (a) visualizes the estimated SLAM trajectory compared to the ground truth. The error is summarized in terms of a boxplot in Fig. (b). It can be clearly seen that the estimated path matches to the ground truth for majority of the trajectory. In the upper left part there can be observed an increased drift which is due to the fact that this area is traversed at the end of the trajectory. The distance to the last loop closure is about $1000m$ which results in an increased accumulated pose drift. A further drift can be noticed for a small loop in the lower right part. A few erroneous scan matchings cause an orientation error entailing a position error which cannot be suffciently accounted for by subsequent loop closures. However, it should be noted that this dataset does not contain any wheel odometry. The vehicle movement is solely estimated based on scan matching.



(a) Position error.

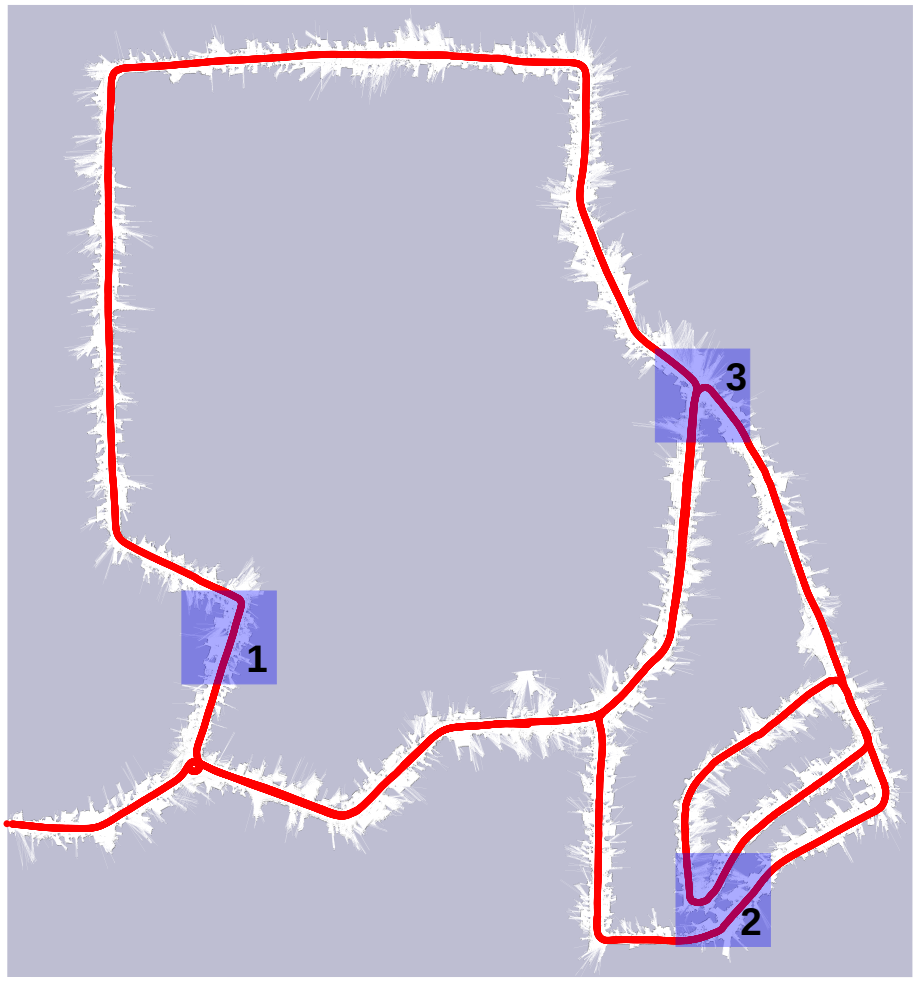


(b) Orientation error.

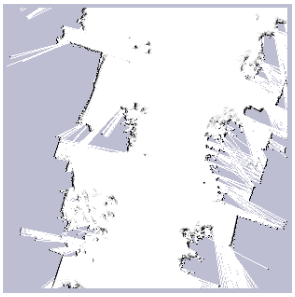
		Mean	Median	Std	Max	Min
ATE	Position	12.625	10.731	8.1134	36.782	0.83311
	Orientation	0.049021	0.041631	0.038422	0.22602	0
MRPE	Position	0.014056	0.012734	0.00296	0.36893	1.1318e-06

(c) Results.

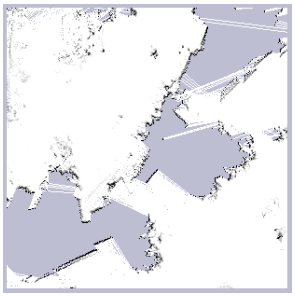
Figure 4.13.: **Kenmore**. A more exhaustive presentation of the error for the estimated SLAM trajectory is shown in Fig. (a)-(c). The peaks in the orientation error around the poses 3800 and 5000 refer to the small loop at the lower right part of the trajectory. This naturally entails a position error which can be noticed around pose 4200 in Fig. (b). The second peak in the position error reflects the pose drift at the upper left part of the trajectory. The median error is at about $10m$ which is larger compared to the results obtained for the other datasets. However, the Kenmore dataset is of a significant larger scale.



(a) Occupancy grid map and overlayed trajectory.



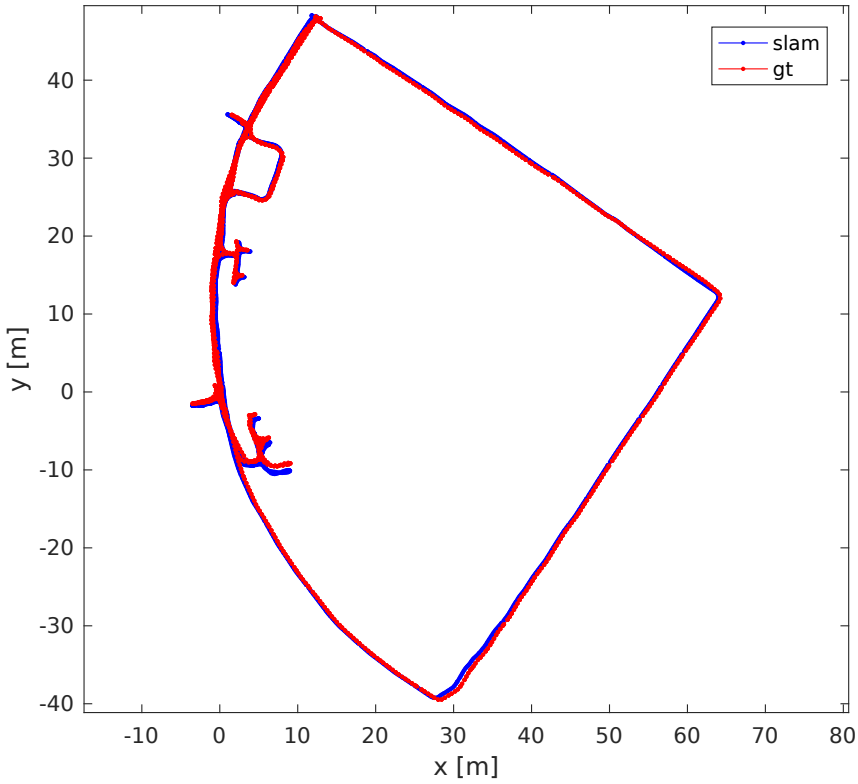
(b) Detail 1.



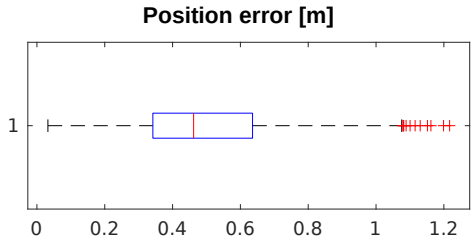
(c) Detail 2.

(d) Detail 3.

Figure 4.14.: **Kenmore**. This figures shows an occupancy grid map of the traversed part of the Kenmore suburb. Thanks to our place recognition and graph optimization we are able to generate a globally consistent map with high local accuracy. The mentioned pose drift for the small loop at lower right section does not entail an unnavigable map. The details (b) - (d) provide insight into the local appearance of the grid map.

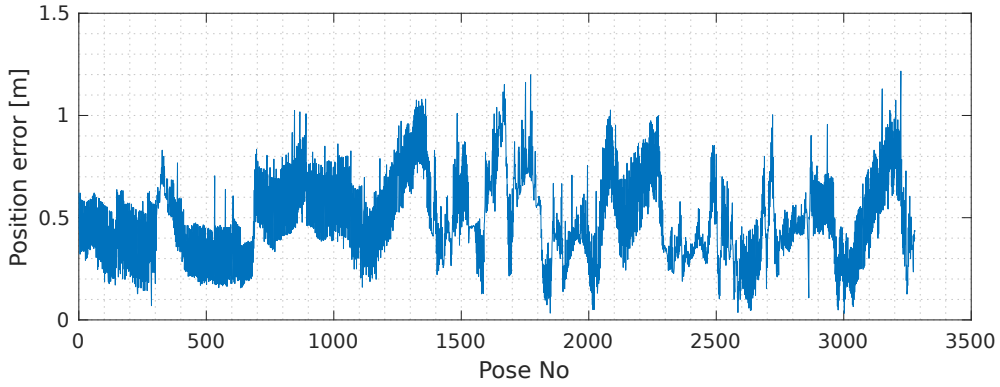


(a) Trajectory.

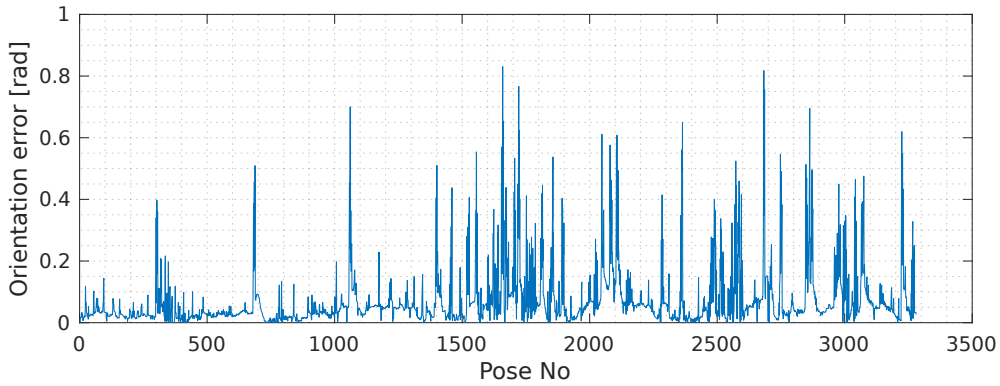


(b) Results.

Figure 4.15.: **ITI**. This Figure shows the experimental results obtained using wheel odometry RGB-D sensor data of the ITI dataset. Fig. (a) visualizes the estimated SLAM trajectory compared to the ground truth. The error is summarized in terms of a boxplot in Fig. (b). It can be clearly seen that the estimated path matches to the ground truth for the majority of the trajectory. The median error is at about $0.46m$.



(a) Position error.

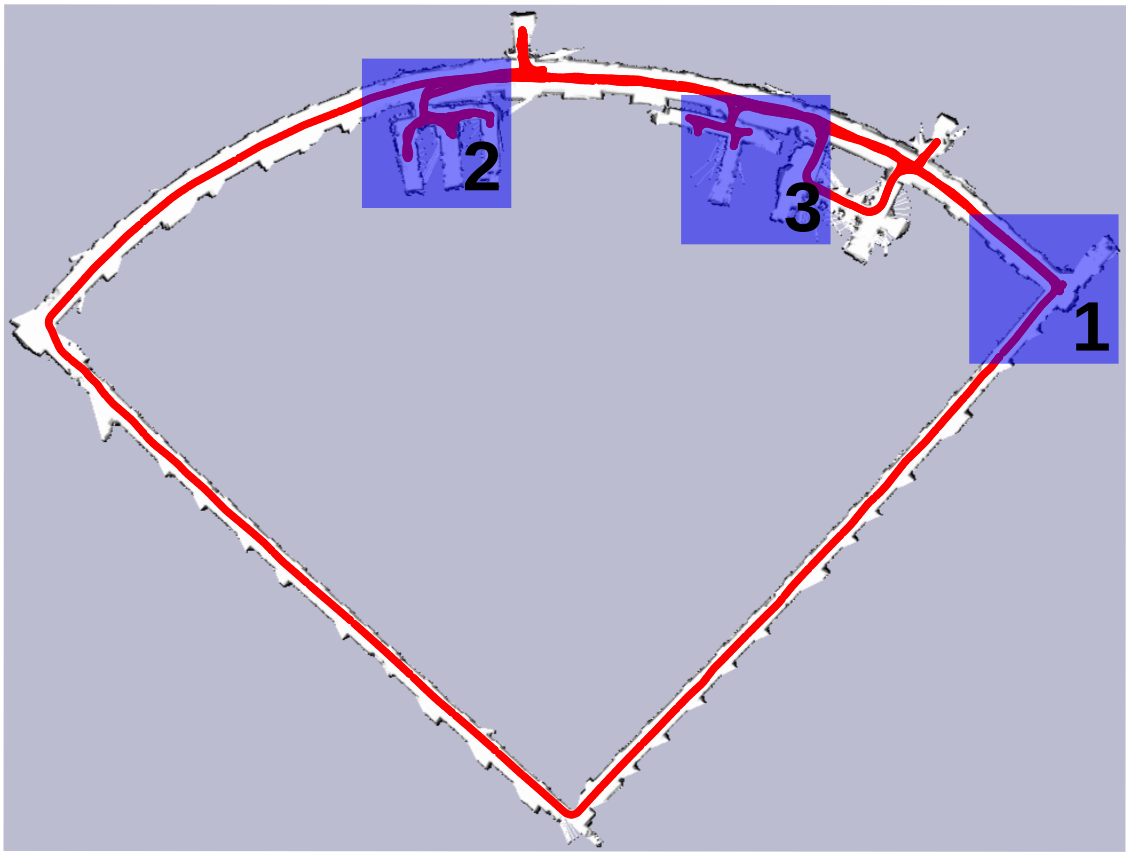


(b) Orientation error.

		Mean	Median	Std	Max	Min
ATE	Position	0.49293	0.46209	0.20848	1.217	0.0324
	Orientation	0.071518	0.040935	0.094678	0.83054	1.1102e-16
MRPE	Position	0.033211	0.018823	0.020611	1.09	3.0379e-06

(c) Results.

Figure 4.16.: **ITI**. This figure shows a more detailed analysis of the pose error for the ITI dataset. The deviation of the SLAM trajectory from the ground truth occurs due to the following reasons. First, it can be noticed that the position error increases the longer the robot moves along the corridor and thus increases the distance to the starting point. A certain pose uncertainty still remains for far-away poses even after a loop closure. Second, it can be observed that a number of larger orientation errors entail pose deviations. We observed that this happens during fast pure-rotational movements of the robot. The error induced by the wheel odometry cannot be sufficiently accounted for by scan matching in some cases which enforces SLAM to rely solely on the uncertain odometric pose. Thanks to the loop closure detection, we are still able to consistently limit the global error.



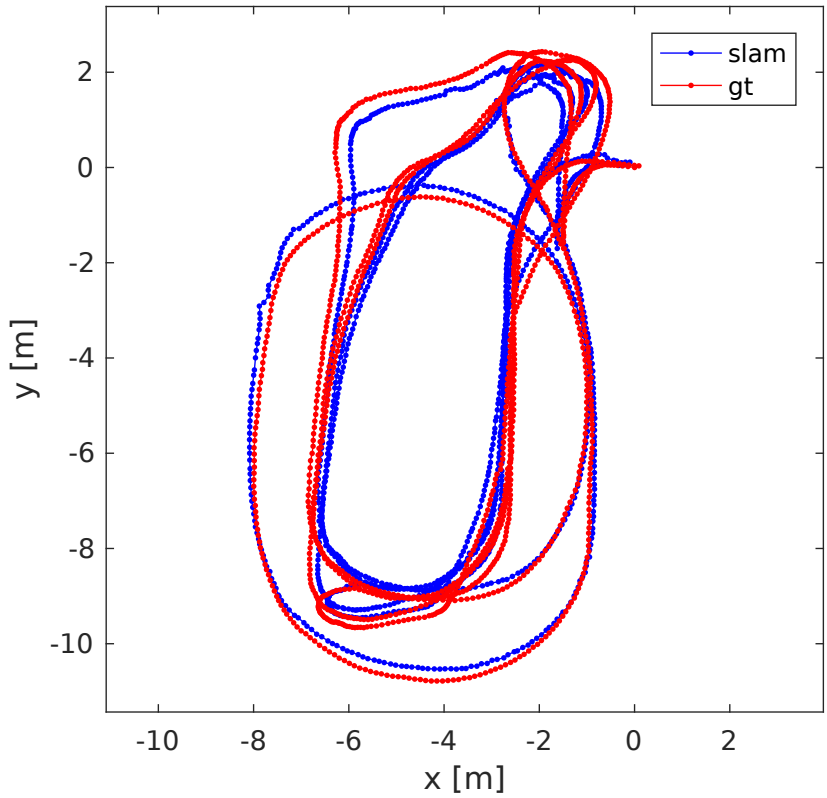
(a) Occupancy grid map and overlayed trajectory.

(b) Detail 1.

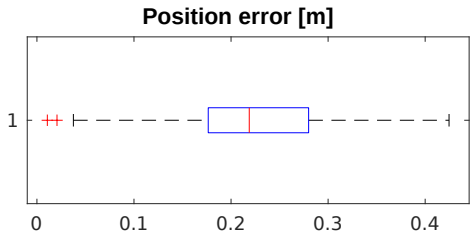
(c) Detail 2.

(d) Detail 3.

Figure 4.17.: **ITI**. This figures shows a globally consistent occupancy grid map for the ITI dataset. Fig. (b) - (d) show details of selected map areas.

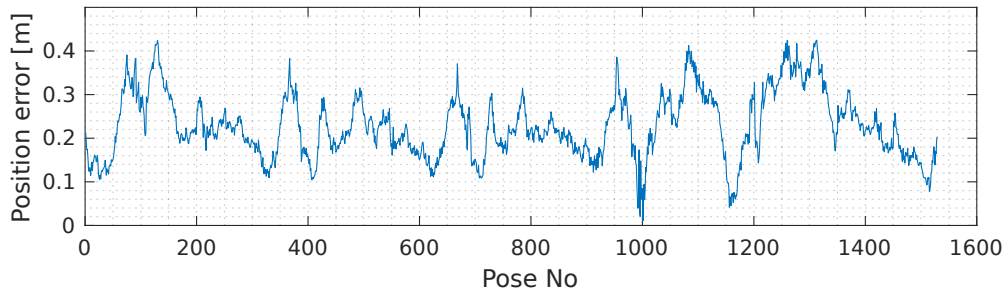


(a) Trajectory.

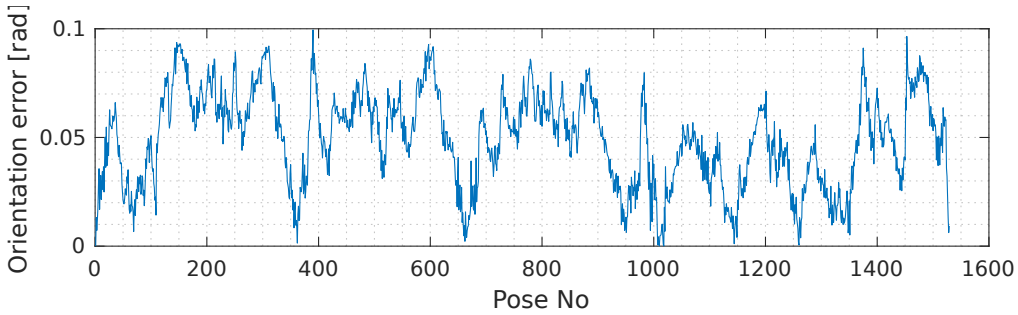


(b) Results.

Figure 4.18.: **FTF-Lab**. This Figure shows the experimental results obtained for the FTF-Lab dataset. Fig. (a) visualizes the estimated SLAM trajectory compared to the ground truth. The error is summarized in terms of a boxplot in Fig. (b). It can be clearly seen that the estimated path matches to the ground truth for the majority of the trajectory.



(a) Position error.

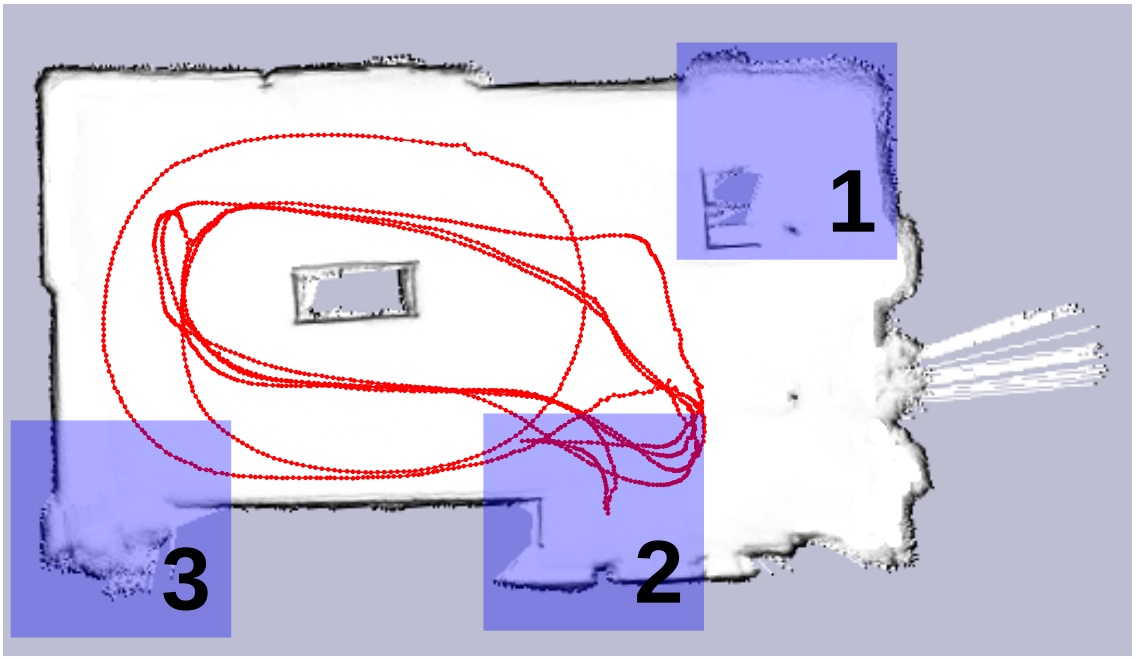


(b) Orientation error.

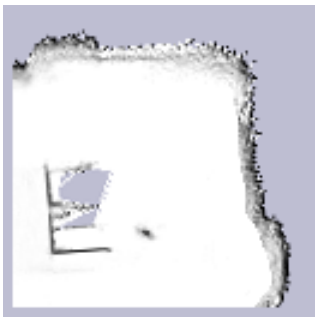
		Mean	Median	Std	Max	Min
ATE	Position	0.22762	0.21877	0.074285	0.42461	0.010737
	Orientation	0.049296	0.051294	0.021289	0.099464	0
MRPE	Position	0.038848	0.034452	0.0106	0.48668	1.2876e-05

(c) Results.

Figure 4.19.: **FTF-Lab**. Fig. (a) - (b) provide a more detailed overview of the position and orientation error. The orientation error remain low and do not cause position deviations in the course of the trajectory. It can be noticed that the position accuracy slightly drops for larger open spaces which result in less reliable distance measurements of the RGB-D sensor. The median position error is at about $0.21m$ and never exceeds $0.42m$.



(a) Occupancy grid map and overlayed trajectory.



(b) Detail 1.

(c) Detail 2.

(d) Detail 3.

Figure 4.20.: **FTF-Lab**. This figure shows an occupancy grid map generated based on the SLAM trajectory for the FTF-Lab dataset. We obtain a globally consistent map which, thanks to SSA provides a high local accuracy. The resolution of the grid map is $\tau_{res} = 0.05$. This map provides a valuable input for map-based localization and path planning. Detail 1 and 2 show areas which have solely been observed from further away. The uncertainty remains larger for this area due to the range-dependent measurement accuracy of the utilized RGB-D sensor which can also be noticed in the generated map.

4.6.2. SLAM - Map accuracy

Setup

The second part of our experimental evaluation addresses the accuracy obtained in the final map estimate of our SLAM framework. For this purpose, we manually steered a mobile robot within the warehouse of the Dresden Exhibitions of Technology. For the experiment we use wheel odometry and laser range data obtained from a SICK S300.

Having estimated the SLAM trajectory, we make use of SSA to optimize the entire map representation as detailed in Section 4.5.4.

Evaluation metrics

The errors of the initial and the final map obtained after optimization is measured based on ground truth measurements. These are achieved through metering distances $L_1 ... L_6$ of distinctive landmarks GT_i in the investigated environment (see also Figure 4.23c). The distances Δ_{L_i} between these salient points were manually estimated in the final map obtained before and after optimization. The metric for measuring the map accuracy is also part of the Rawseeds evaluation methods [52], however it is not as common in the community compared to ATE and RPE. Often researchers analyze errors of the estimated trajectories and overlay generated maps onto floor plans providing these are available. Since our framework aims at generating precise maps, our evaluation explicitly investigates the map error. We found that in the absence of highly-accurate floor plans, the utilization of distinctive landmarks provide an accurate and yet suitable basis. These are determined individually for each experiment and manually extracted from the grid maps. The map error is estimated as follows:

$$\Delta_{L_i} = |GT_i - L_i| \tag{4.27}$$

with GT_i describing the ground truth distance for L_i.

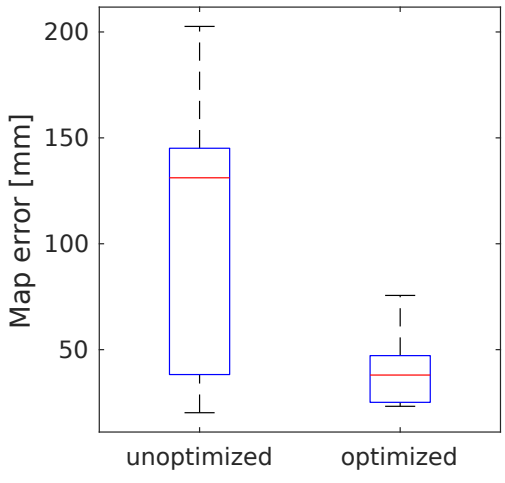
Results

The results of the mapping process are shown as points in a global coordinate frame without the use of post optimization and using SSA respectively in Fig. 4.22. It can be clearly seen that pure pose graph optimization is not sufficient to get accurate and consistent maps. The map consistency is significantly improved by the joint optimization of robot poses and range measurements. The uncertainty in the raw range measurements becomes visible by small alignment errors and a smaller point density. The latter highlights range measurement errors that occur, for instance, due to varying viewpoints and diffuse reflections at surfaces which are well taken into account by SSA. The contribution of the optimization in regards of the point density can be measured by means of entropy. The concatenated entropy values for the global map are presented in Table 4.2. The optimized range measurements enable more tight distributions of beam end points which in turn entails lower entropy values and uncertainties in such regions.

The actual map accuracy is estimated for a subregion of the map. The ground truth, the optimized and unoptimized maps are visualized by Fig. 4.23. The results for the map accuracy are presented by Table 4.23d. Despite the post map optimization the remaining errors appear larger than expected. We expect that this due to the global uncertainty which is also propagated into local map accuracy. Second, we observed increased range measurement uncertainties for the utilized safety laser scanner SICK S300 compared to other laser range sensors which are not certified for person safety.

	ICP	**PG**	**SSA**
Entropy	0.186	0.132	0.102

Table 4.2.: Entropy on maps of experiment 2 built using ICP, pose graph SLAM (PG) and SSA. The values themselves cannot be interpreted straight-forward, however the differences provide a scaled measure that the point density increases when using PG and SSA.

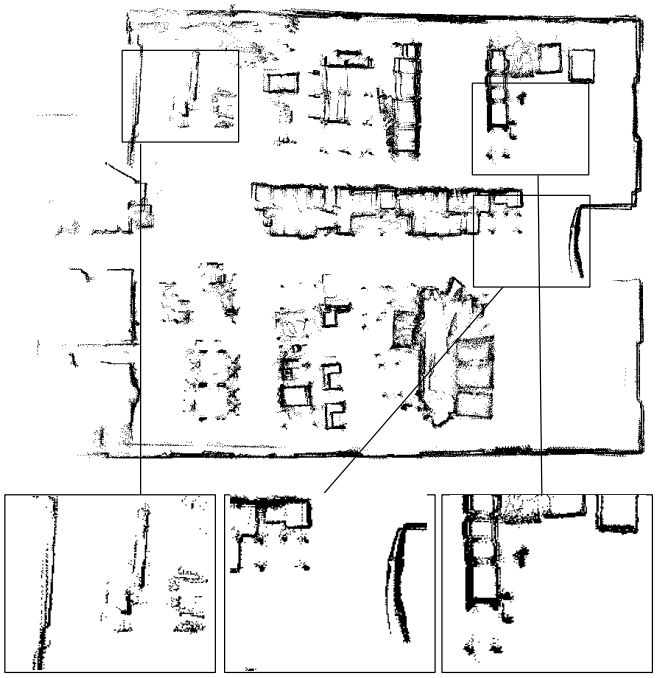


(a) Boxplot of map error.

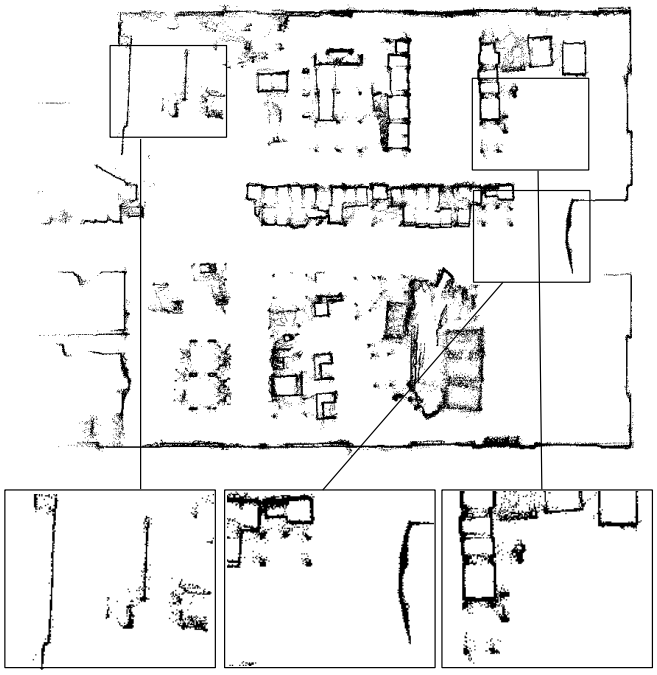
	$\mathbf{GT_i}$	$\mathbf{\Delta^{raw}_{L_i}}$	$\mathbf{\Delta^{opt}_{L_i}}$
L_1	1490.0	202.62	31.81
L_2	762.0	20.22	23.25
L_3	791.0	38.24	25.13
L_4	650.0	145.10	75.60
L_5	892.0	135.13	47.18
L_6	1206.0	127.15	44.21
$median(L)$	-	**131.14**	**38.01**
$mean(L)$	-	111.41	41.20
$std(L)$	-	69.19	19.48
$min(L)$	-	20.22	23.25
$max(L)$	-	202.62	75.60

(b) Map error in detail.

Figure 4.21.: Map error in $[mm]$. The results of the reconstruction are compared to manually obtained ground truth values of reference measurements GT_i. The map error is shown individually for each landmark L_i in the unoptimized map as Δ^{raw}_{map} and in the optimized map respectively as Δ^{opt}_{map}. In addition to that, we provide the median, mean, std, min, max calculated based on all values. Note that GT determine the actual ground truth distances measured and Δ the differences of the distances measured on the map to those for each L_i.

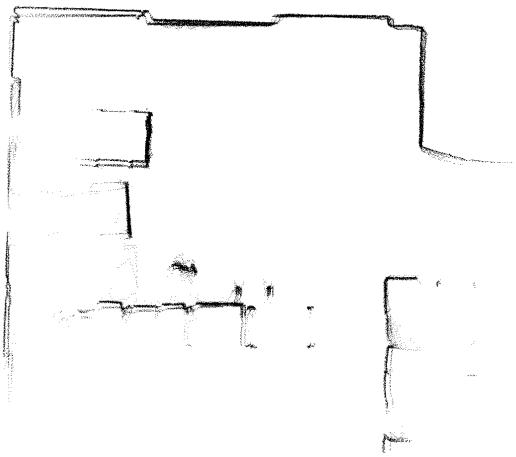


(a) Initial result obtained based on pose graph SLAM. Some structures appear multiple times on the map. Fine contoures are not correctly mapped and seem to blur around objects.

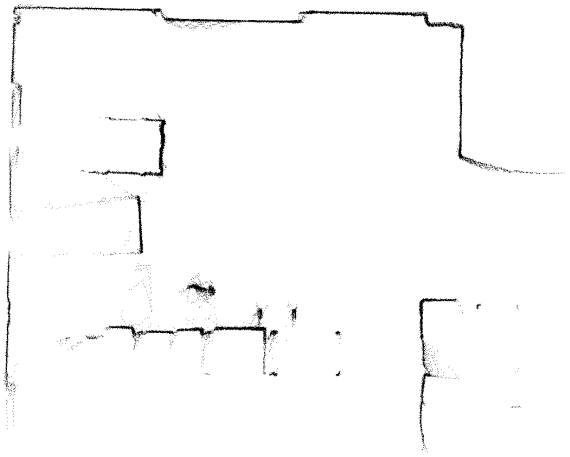


(b) Results obtained based on concurrent optimization of sensor poses and range measurements using SSA. Beam end points are distributed more tightly around surfaces. The occurrences of structures being captured multiple times is notably reduced.

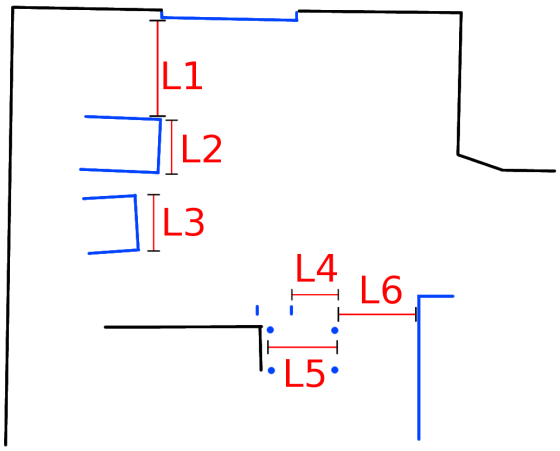
Figure 4.22.: Mapping using graph-based SLAM.



(a) Map built using the initial pose graph



(b) Final map obtained after joint pose and laser measurement optimization.



(c) Ground truth map

(d) Image of investigated subregion

Figure 4.23.: This figure shows the reconstruction of a subregion inside the warehouse of the Exhibitions of Technology, Dresden. The local accuracy of the subregion shown in Fig. (a) is improved using SSA. The optimized map is visualized by Fig. (b). The ground truth map is provided by Fig. (c) showing all incorporated reference measurements L_i. Fig. (d) illustrates the investigated subregion with the utilized mobile robot in the foreground, a rack and paleted goods in the background.

4.6.3. Discussion

Our SLAM framework was evaluated on four datasets captured indoor as well as outdoor environments with laser range finders and RGB-D cameras. The results demonstrate that our algorithms are able to cope with varying sensors, conditions and scales well increasing

environment sizes. The evaluation of the relative pose errors reveals that the local drift between consecutive laser and RGB-D frames most of the time remains below $1.2cm$ and $3.4cm$ respectively. We show that we can achieve a global pose accuracy below $0.2m$ and $0.5m$ using laser range data and RGB-D data respectively for a medium-scale office-like environment. The latter can be improved using multiple sensors which is revealed by the results for the FTF-Lab dataset. Also, our SLAM framework was shown to be capable of mapping large-scale environments covering paths of more than $6km$. It is recommended to revisit parts of the environment multiple times from different directions. particularly for larger loops, if one aims at achieving the maximally accurate estimation of the path and the map. The error increases with the distance to the last loop closure and also remains larger after optimization. This effect can be noticed for the ITI and the Kenmore dataset. Nevertheless we are able to generate globally consistent maps with high local accuracy in all experiments which is a strict requirement of our SLAM framework as it ensures safe navigation and robust localization for long-term operation of mobile robots utilizing the generated maps.

4.7. Chapter Conclusions

This chapter presented a SLAM framework that for online operation enabling the integration into common navigation stacks including those for autonomous exploration tasks. In addition to that, it consists of a post optimization component which allows to increase the accuracy of the final map. This process is launched subsequent to the initial mapping. We found that this combination best meets the balance of runtime requirements and getting the best possible map for long-term autonomous operation.

Our framework is divided into a front-end and a back-end which is also common in other approaches, such as [70, 102]. This guarantees a high degree of flexibility since single components can be easily removed without having to change numerous dependencies. In this way the algorithms for feature extraction or place recognition, for instance, can be replaced without having to modify the graph optimization back-end. The front-end provides estimates about robot motions and loop closures which are processed by the back-end.

Our system expects 2D range scans as input, thus we are able to support a large number of sensors, ranging from laser scanners to different types of RGB-D cameras. Other publicly available frameworks commonly support only RGB-D cameras while RGB images are necessary to detect loop closures [70, 102]. Other approaches being designed for the use of laser range finders can generally be modified for RGB-D cameras [94]. However, the limited power of the utilized methods for loop closure detection hamper their use for environments of increased sizes, particularly for online operation.

In order to allow highly efficient online SLAM using solely range data in large-scale spaces, we make use of our place recognition algorithm GRAPE and efficient pose graph optimization. The latter benefits from the scan matching being applied to subsequent as well as slightly displaced scans of pose chains.

Thanks to the use of iSAM, we are constantly given the current pose uncertainty based on all constraints incorporated by the graph. This information is beneficial for our restricted loop closure retrieval with the pose uncertainty defining a dynamic search radius. This allows us to consider only relevant graph nodes with the search space being automatically expanded when closing large loops. Particularly in this case there exists an increased risk to detect false-positive loop closures making robust optimization methods inevitable. We therefore utilized the concept of switchable constraints which was demonstrated to mitigate the contribution of wrong loop closures and avoids optimization divergence. This renders online SLAM possible even in the presence of a multitude of repetitive structures.

The restricted search in combination with switchable constraints was shown to provide an optimal balance of setting rather optimistic parameters for loop closure detection and on the other side avoiding to overcharge the optimization by passing all loop closures being detected without analyzing their spatial configurations with respect to the current pose estimate and its uncertainty. The restricted search further allows to reduce the runtime requirements since the optimization has to handle fewer variables which contributes to a better online performance.

Based on a post-optimization of robot poses and range measurements, the framework is able to build precise maps which is one of its major objectives. Maps at high accuracy are very contributive for the autonomous operation of mobile robots, particularly in industrial environments. This specifically addresses the localization in such environments allowing to omit the setup of artificial markers. It further enables precise positioning in order to get close to surrounding objects, reloading points or charging stations.

We carried out experiments in different environment types in order to evaluate our framework. First, we investigated the accuracy of the estimated SLAM paths by means of the relative pose error and the absolute trajectory error based on a publicly available dataset of the Stata center collection which is supplemented with high-precision ground truth. We demonstrated that our approach is able to generate maps with an error below $0.04m$ which was shown in-depth in a warehouse environment. Based on that, we are able to provide high-resolution priors for precise map-based localization.

We expect that the integration of new algorithms and the combination of existing optimization methods provide an important fundamental for the navigation of mobile robots. In the following chapters we will learn about novel applications that benefit from the presented framework.

Part II.

Semantic Representations

Chapter 5.

Object Recognition for Robotic Navigation[1]

[1] The content of this chapter has already been published in [75, 78]

5.1. Motivation

The detection of objects which might potentially be an obstacle is a fundamental requirement for robotic navigation. A number of high-level tasks assume domain knowledge about perceived objects. This affects for example robot manipulation tasks that are either limited to a specific object class or require the system to recognize the object class prior to grasping. While almost all mobile robots have at least one sensor system for detecting obstacles in close proximity, only a limited number actually attempts to recognize the class of the obstacle being present. This is mainly due to the fact that sonar and laser range finders have been utilized for mobile robot navigation and particularly obstacle detection over the past decades. The lack of reliable and dense depth information has suspended a wide establishment of cameras as the main sensor for navigation tasks. The rather limited vertical field of view of common range measuring sensors, in contrast, has hampered an in-depth object classification for a multitude of applications. A popular solution has been the fusion of sonar or laser range sensors with monocular cameras. However, these systems require time synchronization and an appropriate extrinsic sensor calibration.

The complex diversity of object classes poses a major challenge for large-scale recognition. This becomes particularly obvious if a robot is expected to recognize any object. However, many mobile robots are prepared to work in one specific environment type, as for example warehouses, homes, offices, or museums. This benefit can be exploited by limiting the object recognition to classes being commonly present in the target environment. Enforcing this constraint does not only allow to reduce the system's complexity, but also enables a better performance and a reduced recognition uncertainty.

The availability of low-cost depth-sensors has essentially contributed to the development of object recognition for robotic applications. In contrast to systems solely working with RGB data of monocular cameras, they enable an efficient segmentation of objects whose appearances can vary extensively due to perspective transformations. Thanks to the depth data, the subsequent pose estimation of objects is simplified since the extraction of keypoint correspondences can be neglected. An object pose can be estimated more stable and accurate due to the density of depth measurements.

For robotic applications it is important that algorithms provide a high performance in order to make crucial information available as fast as possible. Our object detection utilizes an efficient segmentation over range as well as height information. This data is acquired by assuming a 2.5D world which can be defined as follows:

Definition 1. The world is separated into non-overlapping grid cells m_i of equal size with points of 3D objects being projected onto a 2D plane. Each grid cell m_i is further supplemented with a height value which refers to the maximal z-coordinate of a 3D object that falls into m_i with respect to the world coordinate frame. Each grid cell m_i possesses a center of mass which is described by the triplet $\{x, y, h\}$ with h being the height.

In the literature this model is often referred to as height over ground or 2.5D model [12]. In contrast to 3D models, the third dimension describes a scalar height value. We

observed that this model is well-suited for a multitude of man-made structures such as on-road, office or industrial environments consisting of dominant vertical elements. In our work we focus on the application of logistic environments.

Our object recognition exploits the fact of a comparable limited object diversity being expected in warehouses. We therefore utilize geometric features covering physical dimensions of objects such as width and height. In addition to that, we investigate the visual appearance of object observations using a pre-trained Deep Convolutional Neural Network whose features are subsequently evaluated by a multi-class SVM. Our system is trained using solely publicly available image data and is evaluated in an environment it has never seen before which demonstrates its overall ability to generalize from training data. We expect the outcome to be highly beneficial since it allows to predict the behavior and future movements of obstacles given prior object class knowledge.

This chapter presents an approach to object recognition for mobile robots which focuses on runtime performance exploiting environmental constrains and prior knowledge. All components of our object recognition framework will be explained. We will further provide an overview of the state of the art in this field and present experimental results carried out in a warehouse environment.

5.2. Related Work

2D Object Detection

The detection of objects in 2D images has been extensively investigated in computer vision. There exists a number of methods focusing on applications such as, for example, the detection of pedestrians [39], cars [27] and license plates [7]. Typically, the key idea is to utilize a sliding window which is moved over the input image and constantly changed in size in order to detect objects at different depth levels. The content of this window is analyzed by either extracting features first or using the raw data which is rather uncommon due to large variances in the input data. Typical features being used are Local Binary Patterns (LBP) [128] and Histogram of Oriented Gradients (HOG) [39, 174]. The raw data or feature vectors are subsequently classified using, for instance, Support Vector Machines, boosted classifiers or neural networks.

Providing large and appropriate training datasets, the mentioned algorithms are able to achieve promising results for numerous applications [39]. Alternatives to the mentioned sliding window based approaches are given by the keypoint feature extractors such as SIFT [110], SURF [13] and BRIEF [26]. These methods aim at finding points of interest providing a higher intrinsic dimension which typically refer to corners in an image. The surrounding region of the keypoints are used to generate descriptors which are subsequently utilized to find correspondences in other images. These methods enable a certain amount of invariance in scale, lightning and perspective and further allow to estimate a 3D object pose based on correspondences of the 2D keypoints and points of a reconstructed 3D model [144].

Keypoint features and their associated descriptor vectors are commonly used for large-scale object recognition [110, 111, 134]. The features are therefore quantized into visual vocabularies. These allow more compact representations of a set of descriptors by describing objects in terms of a distribution of visual words which is referred to as bag of words (BOW) [134]. The BOW representations provide an efficient source for object retrieval and learning classifiers. Keypoint features allow an efficient matching at larger scales for a reasonable amount of image variances. However, it can be observed that in the presence of significant changes in light and perspective, these methods tend to fail [87, 117, 127, 172]. Some variants address the sensitivity to perspective variance by sampling affine projections of the input image or estimating more high-level manifolds [119]. Here, the higher recall is achieved through significantly increased computational costs [119]. Object detection under significant illumination variances has been investigated by multiple authors, e.g. [17, 172]. Promising results have been demonstrated for specific applications as, for example, the detection and segmentation of road surfaces [6]. There is still a lack of generic methods for extracting and matching keypoints in the presence of major illumination changes which occur, for example, due to different daytimes or low-light conditions [32, 117].

3D Object Detection

The research in object detection using 2D images has achieved precious progress in the last years. However, the availability of low-cost RGB-D sensors has again significantly pushed object detection for a multitude of applications. The above mentioned variances in scale, perspective and illumination can be accounted for more conveniently thanks to the depth images. It enables more computationally efficient algorithms based on depth segmentation. Instead of using scale spaces to incorporate objects at different depth scales, the depth data can be directly utilized to evaluate metrically scaled object proposals. It further allows the segmentation of texture-less objects such as plain walls. While the detection of one object class is essentially simplified, the detection of a larger set of objects also poses a challenge using RGB-D cameras. This is mainly due to the fact that large point clouds (e.g. $640 \times 480 = 307200$ points) have to be processed. Badino et al. presented the algorithm *Stixel* which implements an efficient segmentation in stereo images by exploiting 2.5D world constraints (see Def.5.1) [12]. Chen et al. proposed to use 3D voxel patterns for object detection [170]. Based on images of the publicly available *KITTI* dataset, the authors generate voxel patterns of cars being subsequently vectorized and classified using boosted trees. Their approach enables 3D object detection also in the presence of occlusions.

Deep Learning

The presence of deep convolutional neural networks (CNNs) significantly pushed the development in object and scene recognition. Based on early work of LeCun et al. in the late 1980s [105], there has been established a large number of algorithms such as [99],

[159], [147] in the recent years utilizing CNNs. The baseline recognition performance has tremendously increased as, for example, compared to approaches based on SIFT and bag-of-words (BOW) classification ([148], [108]). In contrast to BOW-based approaches utilizing local features such as SIFT, CNNs use holistic images for feature extraction and classification. Out of the box they are expected to be applied to images with the target object being centered and occupying the majority of the image. Prior image saliency estimation is required in order to make use of CNNs for analyzing images of complex scenes. In this line a number of substantive work has been established, with Selective Search [143] being a well-known representative. More recently, the novel methods Edge Boxes [104] and Bing [31] have proven to be capable for applications with increased runtime requirements. All of these methods ([143], [104], [31]) search the input image for regions of interest being occupied by objects using different heuristics such as the responses of edge detectors.

Semantic Scene Understanding

The research field of semantic scene understanding has been exhaustively investigated by the computer vision community in recent years. The availability of consumer-grade RGB-D cameras has essentially supported this. Silberman et al. proposed to segment RGB-D images into the classes floor, walls and supportive elements and presents an inference model describing physical interactions of these classes [123]. Geiger et al. presented a method for joint inference of 3D objects and layout of indoor scenes [58]. Zheng et al. suggest a Conditional Random Field (CRF) for dense segmentation and semantic labeling. Either of the methods provide powerful tools for semantic labeling achieving outstanding results. However, these implementations require a long computation time. The fast segmentation approach *Stixel*, proposed by Badino et al.[12] has recently been extended by semantic labeling (Stixmantics, [146]). The results reported for outdoor traffic scenes are surprising, moreover, with the short runtime being achieved through the use of dense SIFT descriptors.

Summary

Object detection in 2D can be performed at high frequency on a CPU. However, it suffers from limited invariance in illumination and perspective. RGB-D sensors help achieving more invariance. A lot of approaches for 3D object detection and scene understanding are available with only a few enabling high performance on CPU architectures, e.g. Stixel [12]. Those approaches conducting the classification of geometric models (e.g. [170]) require depth images for training which are significantly harder to obtain at larger scales for numerous object classes than RGB images. There exists a number of datasets for automotive applications, but, for instance, none for logistic environments. The majority of these uses full 3D models making segmentation a computationally expensive task. Only Stixel uses a 2.5D model focusing on typical traffic scenes making it most related to our approach since a top-down range scan is generated and object boundaries are obtained

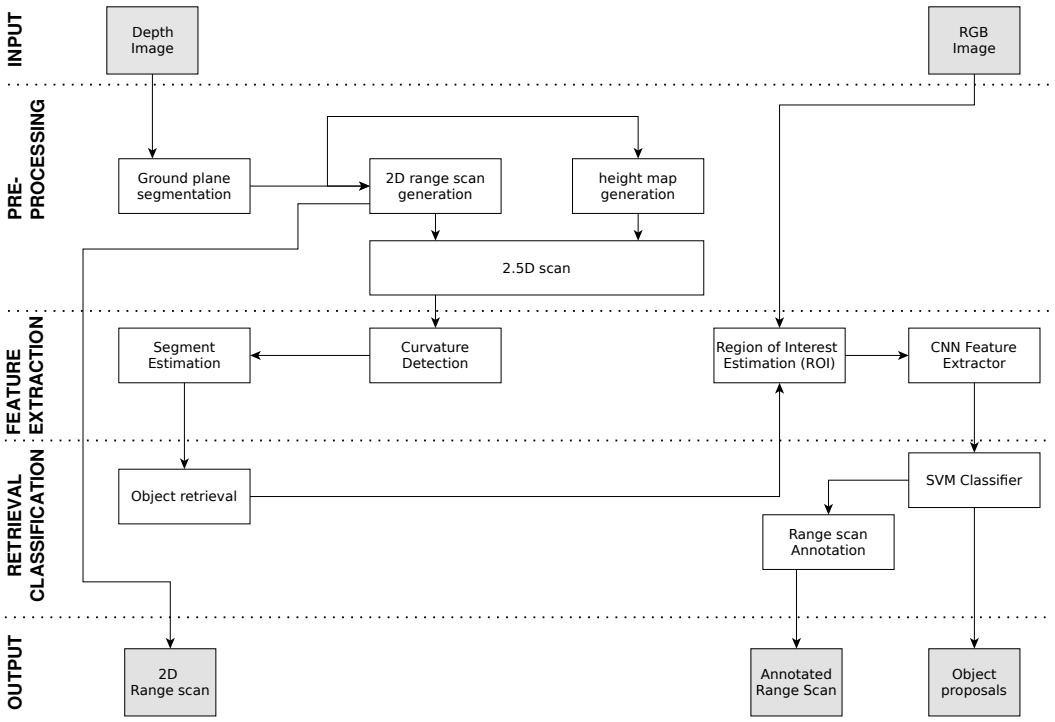


Figure 5.1.: Block diagram showing the architecture of our object recognition framework. The left part of the graph demonstrates the processing of the depth images and the right part the processing of the RGB images.

from range and height differences. However to our knowledge, there is no prior work utilizing this in combination with an object retrieval based on properties such as height and width rather than volumetric models such as voxel sets.

5.3. System Overview

The pipeline of our approach is illustrated by Figure 5.3 and is divided into the following steps:

1. The input depth image is segmented by means of discontinuities in the range and height data
2. For each detected object a region of interest in the RGB image is determined
3. The sub image defined by the region of interest is passed to a CNN and searched for features in the RGB image
4. The CNN's output is evaluated by a multi-class SVM being trained for all expected object classes in the environment

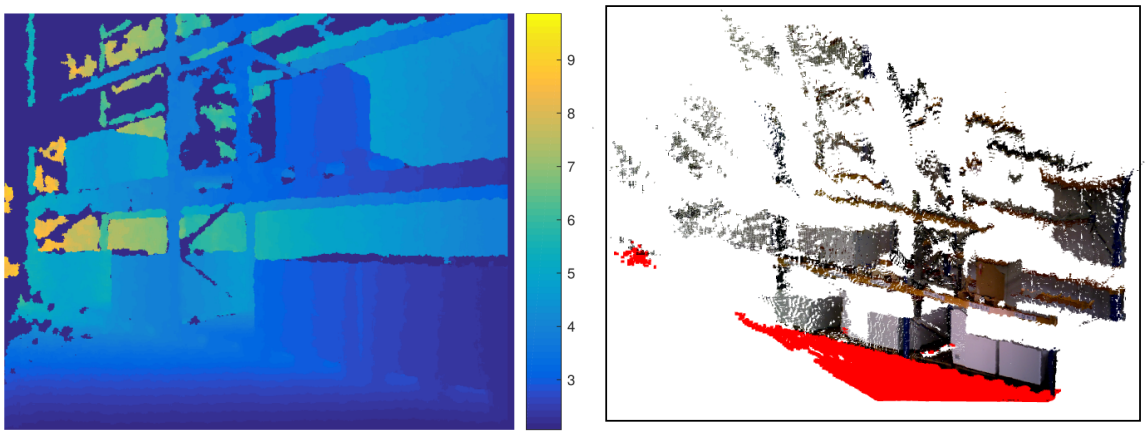
Each component of our object recognition framework is described in depth in the following sections.

5.4. Object Recognition Framework

The input RGB-D data of a camera is searched for objects. Thanks to the range data, the detection of occupied space in close proximity of the vehicle is simplified. A point cloud D is generated based on the input depth image.

5.4.1. Ground Plane Segmentation

At first we filter those points of D reflecting the ground. The ground plane is estimated within the system calibration during a prior teach-in procedure. We therefore fit planes inside the point cloud computed based on the depth image and the calibration parameters. A RANSAC-based implementation for plane estimation of the library PCL [140] is used for this step. The ground plane computed within the teach-in phase is kept fixed. Within an initialization phase we check the validity of the ground plane to avoid miscalibrations due to sensor relocations given sufficient point on the ground can be detected. For performance reasons the continuous re-estimation of the ground plane is omitted during remainung runtime which thanks to fixed camera pose (w.r.t. the vehicle) and 2.5D world constraints provides accurate results. We define the remaining point cloud with the ground plane being removed as $\hat{D}$ (see Fig. 5.2).



(a) Input depth image.

(b) Ground plane in point cloud.

Figure 5.2.: Ground plane segmentation. Figure (a) shows an input depth image which is used to generate a point cloud. Figure (b) illustrates the segmented ground plane (red) within a point cloud.

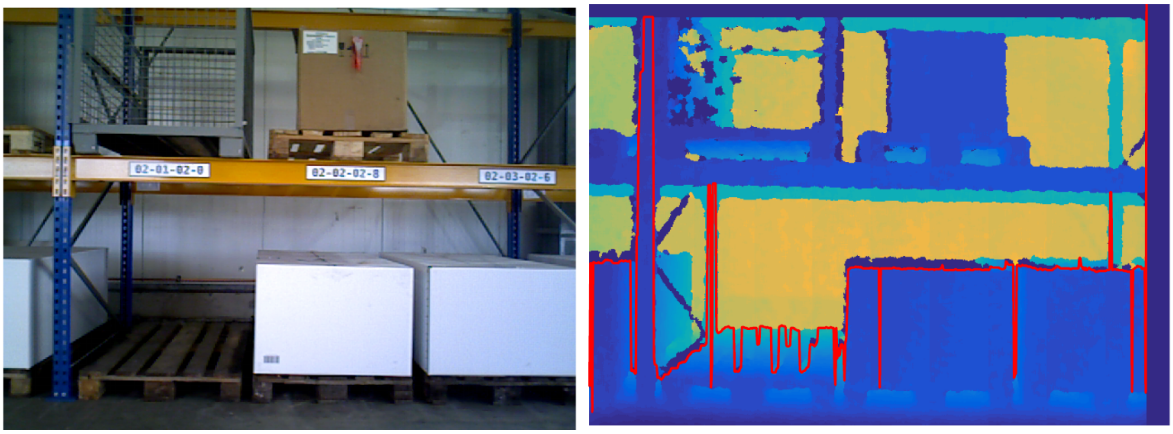
5.4.2. Range and Height Scan

Recalling Eq. 4.1 of Chapter 4 we generate a top down projection of $\hat{D}$ by converting each point $p^{(k)}$ of $\hat{D}$ from Cartesian to polar coordinates. Contrary to Eq. 4.1 we additionally incorporate height values in order to obtain $2.5D$ representations. Thus the estimation is slightly changed as follows:

$$\begin{pmatrix} \theta \\ \rho \\ h \end{pmatrix}^{(k)} = \begin{pmatrix} atan2(p_y^{(k)}, p_x^{(k)}) \\ \sqrt{(p_x^{(k)})^2 + (p_y^{(k)})^2} \\ p_z^{(k)} \end{pmatrix} \tag{5.1}$$

where $\theta^{(k)}$ refers to the bearing, $h^{(k)}$ to the height over ground and $\rho^{(k)}$ to the range of the point k relative to the camera origin.

More details on this, particularly on the definitions of θ_{hfov}, θ_{cone} and the actual estimation of the scan contour $s(b)$ being used in the following can be found in Section 4.4.2 of Chapter 4.



(a) Input RGB image.

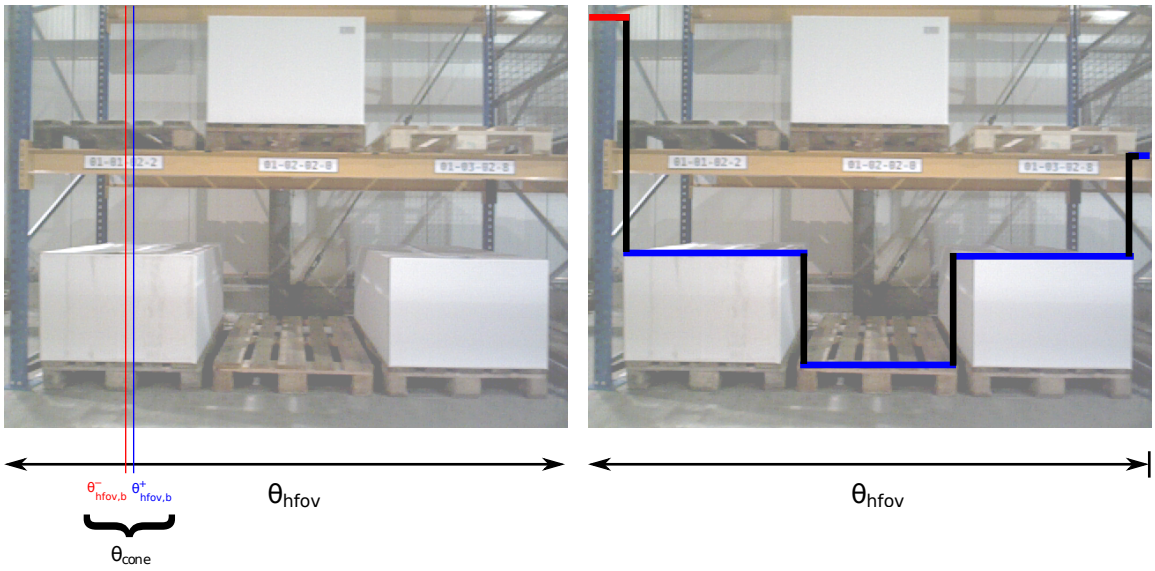
(b) Height data (red) overlayed on depth image.

Figure 5.3.: This figure shows an RGB-D frame with (a) being the RGB image and (b) the depth image. The depth image is overlayed with the estimated height data.

5.4.3. Curvature Detection

In order to detect objects, $s(b)$ is further analyzed. The physical boundaries of objects in the world typically entail notable changes in curvature. We exploit this property to enable an efficient object detection based on features extracted from a 1D contour. We shift the signal $s(b)$ by a constant offset w resulting in $\hat{s}(b)$:

$$\hat{s}(b) = s(b) - s(b - w) \tag{5.2}$$



(a) Measurement cone (b) Height estimation

Figure 5.4.: Figure (a) illustrates the horizontal field of view (θ_{hfov}) of an RGB-D sensor being constructed based on the depth image. A measurement cone θ_{cone} refers to one column of the depth image. The width of θ_{cone} depends on the sensor's size α_w and field of view θ_{hfov}. All subsequent range and height data processing work on the data found within these measurement cones. Figure (b) demonstrates the extraction of height data from the input depth image. For each measurement cone θ_{cone} we estimate the height as the first discontinuity exceeding a certain threshold. Red lines indicate *max-height* values, that is no discontinuity can be found within these cones. Blue lines indicate that height values below *max-height* are estimated. Black lines signal potential transitions between object segments based on height differences.

for $(b - w) > 0$. The difference signal $\hat{s}(b)$ is used to identify peaks in the curvature given a minimum threshold λ_{ds}:

$$e^+ = \{\hat{s}(b-1) > \lambda_{ds}\} \tag{5.3}$$
$$e^- = \{\hat{s}(b) < -\lambda_{ds}\} \tag{5.4}$$

with e^+ and e^- denoting peaks originated from positive and negatives slopes respectively. It is necessary to distinguish this case instead of incorporating the absolute values in order to locate the peaks correctly.

5.4.4. Segment Estimation

Having obtained the putative object boundaries $E = \{e^+ \cup e^-\}$, we utilize the consecutive peaks E_i and E_{i+1} to identify subsets of the contour corresponding to objects in the world. The width $w(i)$ and height $\hat{h}(i)$ for each subset i is obtained as follows:

$$\hat{h}(i) = median\,[\,h(E_i, ..., E_{i+1})] \tag{5.5}$$

$$w(i) = \sqrt{(p_x^{(E_i)} - p_x^{(E_{i+1})})^2 + (p_y^{(E_i)} - p_y^{(E_{i+1})})^2} \tag{5.6}$$

The object proposal o_i consisting of the segment data $\{w, h\}_i$ is forwarded and evaluated by the object retrieval.

5.4.5. Object retrieval

The object observations are defined by the segments $O = \{o_1, o_2, ..., o_n\}$ with $o_i = (w, h)_i$ being its associated property vector. The prior object database is searched for corresponding items based on o_i. We use a KD-tree with Approximate Nearest Neighbour search constructed based on dimensions being commonly observed for particular object classes $P = \{p_1, p_2, ..., p_n\}$. The database consists of n entries with $p_i = (w, h, c)_i$ describing object properties of the class c. The object retrieval implements the nearest neighbour search $NN(o_i, P) \in P$ given the query $o_i \in O$ and a distance metric $dist$:

$$NN(o_i, P) = argmin_{p \in P}\,[dist(o_i, p)] \tag{5.7}$$

The estimated height and width of an object are subject to uncertainties due to the range-dependent measurement accuracy of RGB-D sensors. We account for this by introducing the following measurement matrix $\mathbf{M}$:

$$\mathbf{M} = \begin{pmatrix} 0.5\,m_w m_r & 0 \\ 0 & 0.1\,m_h m_r \end{pmatrix} \tag{5.8}$$

with m_w, m_h and m_r being individual weights for the width, height and range uncertainties. Both, width and height are equally affected by the range uncertainty being modeled by m_r. The factor m_w is set according to the position of the object in the image coordinate frame. This addresses the problem of detections close to the image boundaries resulting in partial object observations. A similar effect can be observed for large objects being close to the image sensor. The constant factors 0.1 (for height) and 0.5 (for width) are used for the covariance estimation in order to account for object occlusions which can occur anywhere in the image and hence are hard to determine during online

operation. Since occlusions rather affect the width estimation of an object, the height provides a more certain contribution to the object class retrieval.

$$m_w = 1 + \begin{cases} p_x(E_i) < \gamma_{x,min} : & e^{-\frac{1}{2}\frac{(p_x(E_i)-\mu_{w,min})^2}{\sigma_w^2}} \\ p_x(E_{i+1}) > \gamma_{x,max} : & e^{-\frac{1}{2}\frac{(p_x(E_{i+1})-\mu_{w,max})^2}{\sigma_w^2}} \\ otherwise : & 0 \end{cases} \tag{5.9}$$

$$m_h = 1 + \begin{cases} p_y(h_i) < \gamma_{y,min} : & e^{-\frac{1}{2}\frac{(p_y(h_i)-\mu_h)^2}{\sigma_h^2}} \\ otherwise : & 0 \end{cases} \tag{5.10}$$

$$m_r = e^{-\frac{1}{2}\frac{(r_i-\mu_r)^2}{\sigma_r^2}} \tag{5.11}$$

The weighting factors m_w and m_h can be interpreted as a Gaussian filter applied to the boundaries of an image. The closer an object approaches the image boundaries the more uncertain is the estimation of its width and height. This weighting filter can be efficiently implemented using a pre-calculated lookup table.

The parameter σ_r is set according to the properties of the range sensor. The Mahalanobis distance is utilized to incorporate the presented observation uncertainty within the nearest neighbour search of Eq. 5.7 which is defined as:

$$dist(o_i, p, C_i) = ((o_i - p)\mathbf{C_i}^{-1}(o_i - p))^{\frac{1}{2}}, p \in P \tag{5.12}$$

As a result of this step we obtain a set of object class proposals $\hat{O}$ matching our database descriptions P based on the object observations O. These proposals are further investigated by means of RGB features which is described in the following.

5.4.6. ROI Estimation

The object proposal $\hat{o}_i$ with the associated properties $\{w, h\}_i$ and contour boundaries (E_i, E_{i+1}) is prepared for further appearance-based analysis. A bounding box around the object is generated as region of interest (ROI) inside the RGB image. We therefore transform the object proposal $\hat{O}$ into the coordinate frame of the RGB camera based on the properties $\{w, h, p_x(E_i), p_x(E_{i+1}), p_y(E_i), p_y(E_{i+1})\}$.

Subsequently the image content of the ROI is passed to the CNN-based feature extraction.

5.4.7. CNN Features

This processing unit extracts features from the RGB image data. This process is restricted to the areas defined by the ROIs of the prior detection step. For this purpose we use a Convolutional Neural Network (CNN). The principle of CNNs can be summarized as follows. A set of convolutional filters is repeatedly applied to the 2D image data. The filter

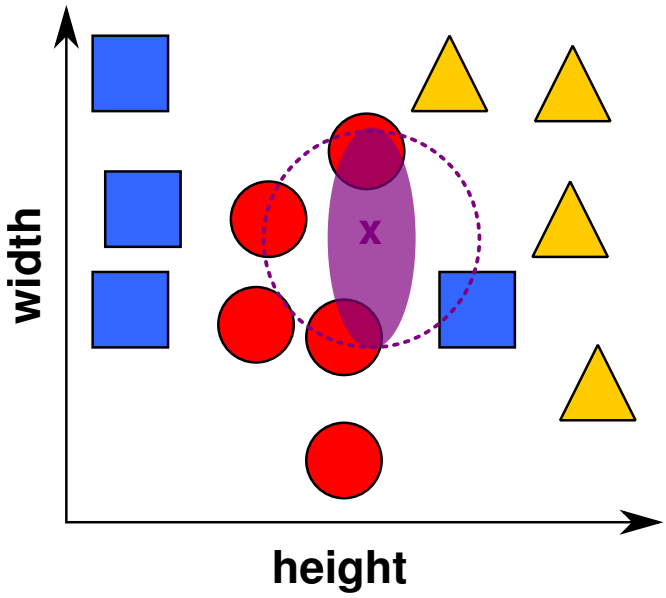


Figure 5.5.: This figure demonstrates the object retrieval based on k-nearest neighbour search. The blue squares, red circle and yellow triangles represent different object categories. In our applications we observed that objects of a specific class have certain height levels but vary in the width depending on the object's orientation and potential occlusions. A nearest neighbour search for the input data (purple cross) is conducted. The larger uncertainty in the width is incorporated by means of the Mahalanobis distance. An Euclidean distance metric (dotted purple circle) would entail more object class candidates since it weights the dimensions height and width equally.

Figure 5.6.: Illustration of a region of interest (ROI) being detected using the depth image data and projected into the RGB camera's coordinate frame. The image content inside the ROI is passed to the feature extraction module.

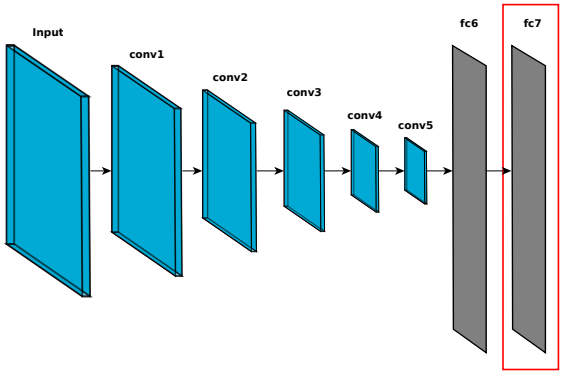


Figure 5.7.: **CNN**. This figure illustrates the architecture of the utilized convolutional neural network (CNN) CaffeNet. The feature vectors generated by the fc7 layer are used for classification (highlighted red).

outputs are collected into non-overlapping grids. The next layer subsamples the input data by applying pooling methods such as taking the maximum or average of the grid. The combination of convolving and sub-sampling the input data is repeatedly carried out at successive network layers. This method allows to learn features at different scales and spatial positions in the image. The complex fully-connected layers of neural networks are typically found at the end of a CNN. The outputs of different CNN layers can be combined for the final output. The CNNs differ significantly from other feature extraction methods used in computer vision since they learn features and their distributions at different levels (e.g. parts, objects, local characteristics) given the training data. Depending on the depth, the layers respond to different scales of an object. The further a layer is located from the input layer the more local will be the response and the smaller the affected area of a firing neuron. As a feature extractor we make use of the pre-trained CNN *CaffeNet* [86] which consists of 7 layers with our system utilizing the fc7-layer. Since this layer is located at the end of the network we obtain a 4096-dimensional feature vector capturing local image characteristics.

5.4.8. Image/ROI Classification

The content of each detected object is classified based on the CNN features extracted within its ROI. We train a two multi-class Support Vector Machine (SVM) following a one-versus-all schema [81]. We therefore train one binary SVM for each class k:

$$b_i \cdot (\mathbf{w_k^T} \cdot \mathbf{x_i} + w_{k,0}) \geq 1 \tag{5.13}$$

with $\mathbf{w_k}$ being the weight vector for class k, $w_{i,0}$ the offset and $b_i \in \{-1; 1\}$ a class-specific value for the sample x_i. We use a linear kernel which can be defined as the following cost function:

$$\Psi_k(\mathbf{w_k}) = \frac{1}{2}\mathbf{w_k^T}\mathbf{w_k} \tag{5.14}$$

An observation sample x is then classified by all K SVMs:

$$\hat{y} = \underset{k \in \{1..K\}}{argmax}\, \Psi_k(x) \tag{5.15}$$

with $\hat{y}$ being the class label with the highest score.

For a more exhaustive derivation of SVMs, the reader is referred to the introductory literature of Duda et al. [44]. The utilized multi-class support is adapted from Hsu and Lin [81].

We observed that using a linear kernel provides promising classification results while keeping the computational costs at a minimum. This is necessary since a more complex system has to evaluate a large number of classifiers for each obstacle being detected at a high frequency.

Each SVM is trained with positive samples of one target class and negative samples being randomly drawn from the other classes as well as images describing various objects not being recognized by our system such as walls, ladders and windows (see Fig. 5.8).

The ROIs and images being evaluated are labeled with the class having the minimum distance to the input feature vector. Those object proposals exceeding a distance threshold τ_{svm}, are labeled as *unknown*.

The combination of SVMs and CNNs provides a number of advantages compared to soft-max layers being typically placed at the end of a CNN for classification. First, the performance of this layer is reliant on preceding network layers which is why they are typically reconfigured as well. Training a CNN is computationally expensive and requires a large appropriate set of training images in order to enable reasonable generalization performance. Solely continuing the training of the soft-max layer is indeed possible but to our experience performs worse than SVMs which is also shown in [82]. Second, we also found that combining SVMs with CNNs enables a straightforward adaption to other application scenarios since training SVMs with input features of a static CNN can be accomplished with reasonable effort, both in terms of system requirements and training time.

5.4.9. Range Scan Annotation

The range scan s being generated in the previous steps is fused with the detected object proposals. Each cone $s(b)$ is assigned the corresponding class label. Those not being described by a known object class, are assigned the label *unknown*. If an object was detected for the entire image, we assign each measurement having a range ρ below ρ_{max} the estimated class label. This annotated range scan provides the fundamental for our semantic mapping algorithm.

5.5. Experiments

In this project we exemplarily trained an SVM using CNN features for the following object classes: forklift trucks (Forklift), humans (Human), palleted goods (Pallet). These classes

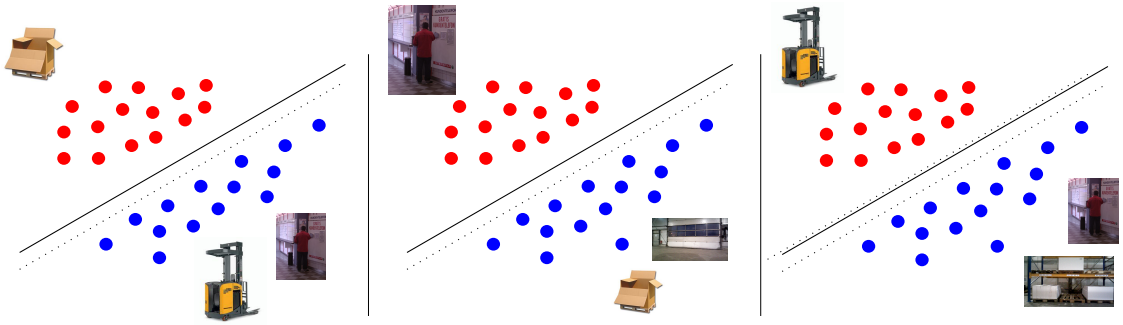


Figure 5.8.: This figure illustrates our multi-class SVM. A binary SVM is trained for each class. Positive samples are originated from the target class and negative samples are randomly drawn from other object classes.

are expected to be most common in warehouse environments. In our first experiments we evaluated the contribution of training an additional class explicitly accounting for walls, large racks and clutter being expected in warehouses. However, we observed that adding this class rather introduced unintended classification uncertainty since it covers widely spread clusters in feature space due to their large variance in visual appearance. By not explicitly considering the background we are able to mitigate deteriorations of classification. Thanks to our prior segmentation we already get rid of the majority of the clutter.

5.5.1. Evaluation Methodologies

This section describes the underlying evaluation methodologies being used in this chapter.

The performance of classifiers is evaluated based on receiver-operating curves (ROC) and properties that can be derived from these (see also Fig. 5.9). This method has become the state of the art in computer vision and machine learning [2]. Similarly to the precision-recall curves we introduced in Section 3.3.5 (Chapter 3), we again differentiate true positives/negatives (tp/tn) and false positives/negatives (fp/fn) respectively. Having trained a classifier, a testing or validation dataset with class labels is passed to our evaluation. By varying the distance threshold τ_{svm} we get different decisions for the same classifier. This entails varying true and false positive rates which in turn are utilized to describe a ROC curve. For each point on this curve we can generate exactly one confusion matrix describing all correct and incorrect decisions. Generally speaking, the closer the ROC curve gets to the point (1.0 [tp]; 0.0 [fp]), the better the classifier. Mathematically this can be expressed by the area under the curve (AUC) which is the integral of the ROC curve. This value is within the range of $[0; 1]$ with $AUC = 0.5$ expressing a random decision. A more illustrative explanation of ROC and AUC can be found in Fig. 5.9.

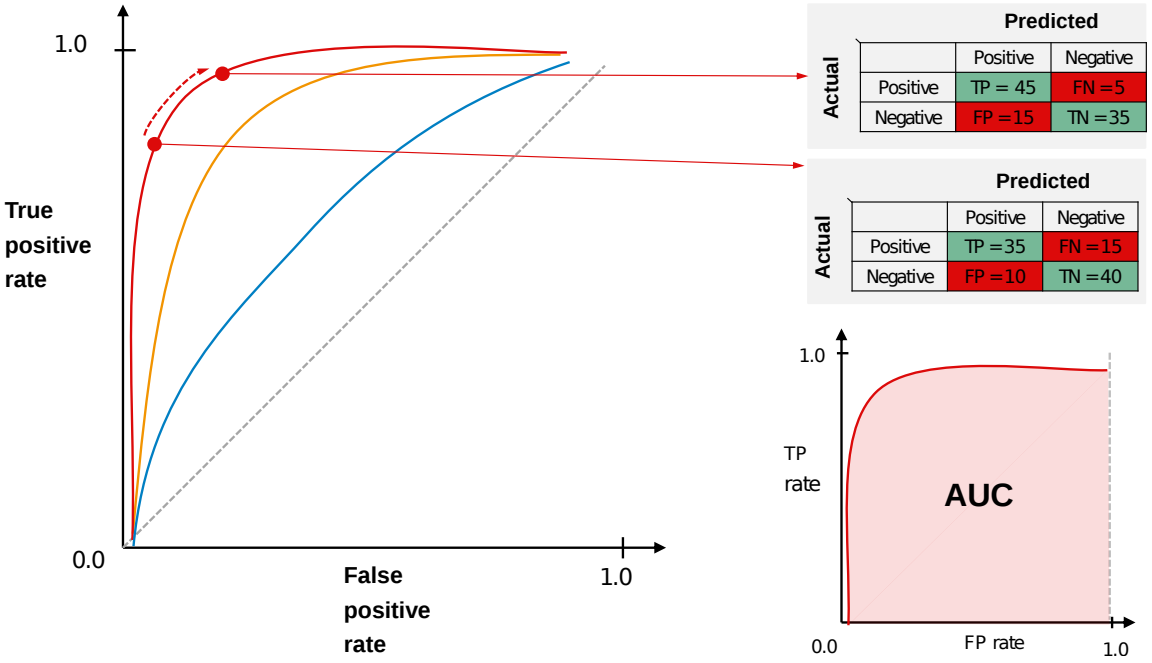


Figure 5.9.: **ROC**. This figure illustrates receiver-operating (ROC) curves which are used in our evaluation. The plot on the left shows three curves which refer to different classifiers being trained. The red curve (left) performs best. It achieves a high true positive rate with only a few false alarms (false positives). The orange one is slightly worse than the red curve. The blue curve performs the worst of all three with a result close to a classifier returning random decisions (dotted line). The behaviour of a trained classifier can be adjusted by picking varying thresholds which is visualized by the red points. These can literally be slided along the curve which entails different confusion matrices (tables on the right). The integral over the ROC curve which is referred to as area under the curve (AUC) is used in order to evaluate the trained classifier (lower right plot).

5.5.2. Datasets

Our recognition system is trained based on publicly available image data for the mentioned object classes. Specifically we use the Image-net database [41] for annotated images of forklifts and pallets. The training data for the class *human* is obtained from the INRIA person dataset [39]. Since the number of training samples obtained from these sources is limited, we added further training images from the internet. This process was automated using the Microsoft Bing API [1]. All training images obtained in this way are manually inspected and partly cropped. Note that this data is solely originated from publicly available image sources. Our system was trained and tested given this data. An additional dataset captured in a typical warehouse environment was recorded in order to evaluate the generalization ability of our system. This validation dataset was captured

by manually steering an AGV equipped with an RGB-D camera (see A.1). We applied depth segmentation and feature extraction to the depth and RGB images as explained in the preceding sections. The extracted ROIs for obstacle priors are subsequently passed to our set of classifiers.

Component	Description	Variable	Value	Ref
2.5D Scan Generation	Sensor resolution (width)	α_w	$640.0\,px$	Sec. 5.4.2
	Sensor field of view (horizontal)	θ_{hfov}	$1.012\,rad$	Sec. 5.4.2
Scan Annotation	Maximum incorporated ranges	ρ_{max}	$5.0\,m$	Sec. 5.4.9
Curvature Detection	Window size for difference signal	w	4	Sec. 5.4.3
	Threshold for peak detection	λ_{ds}	$0.09\,m$	Sec. 5.4.3
Object Retrieval	Mean for distance uncertainty of object	μ_r	$0.0\,m$	Eq. 5.11
	Mean for height uncertainty of object	μ_h	$3.36\,px$	Eq. 5.10
	Mean for uncertainty estimation (left)	$\mu_{w,min}$	$3.36\,px$	Eq. 5.9
	Mean for uncertainty estimation (right)	$\mu_{w,max}$	$636.69\,px$	Eq. 5.9
	Threshold for uncertainty estimation (right)	$\gamma_{x,max}$	$560.0\,px$	Eq. 5.9
	Threshold for uncertainty estimation (left)	$\gamma_{x,min}$	$80.0\,px$	Eq. 5.9
	Threshold for uncertainty estimation (bottom)	$\gamma_{y,min}$	$80.0\,px$	Eq. 5.10
	Std. dev. for width uncertainty	σ_w	$31.05\,px$	Eq. 5.9
	Std. dev. for height uncertainty	σ_h	$31.05\,px$	Eq. 5.10
	Std. dev. for range uncertainty	σ_r	$2.0\,m$	Eq. 5.11
Image/ROI Classification	Threshold for incorporating object class	τ_{svm}	0.04	Sec. 5.4.8

Table 5.1.: Parameter selection for object recognition experiments. The tables provides references to more exhaustive descriptions of the parameters (right column).

5.5.3. Segmentation

This experiment analyzes the performance of the object detection system before the retrieval and classification components. We therefore run the detection and store the input RGB images along with the overlaid bounding boxes. The amount of correct, incorrect

and missing detections is determined irrespective of the actual class labeling. Thresholds and parameters of the segmentation are kept fixed. As a result we obtain an overall true-positive detection rate of about $\mathbf{87.21}\%$.

5.5.4. Classification

The Table 5.3 shows the confusion matrix obtained for the test dataset. It is obvious that the classes can be confidently separated from each other. The results obtained for the class *forklift* are slightly worse than those for the other classes. This is probably due to the fact that this class captures a large variety of different wheeled vehicles typical for warehouses ranging from small automated lifting carts to large forklift trucks. The other classes rather vary in pose variance than actual visual appearance.

Class	# Images			# Iterations	# Support Vectors
	Train	Test	Val		
Forklift	288	288	241	152	76
Human	182	182	107	171	70
Pallet	147	147	652	89	67

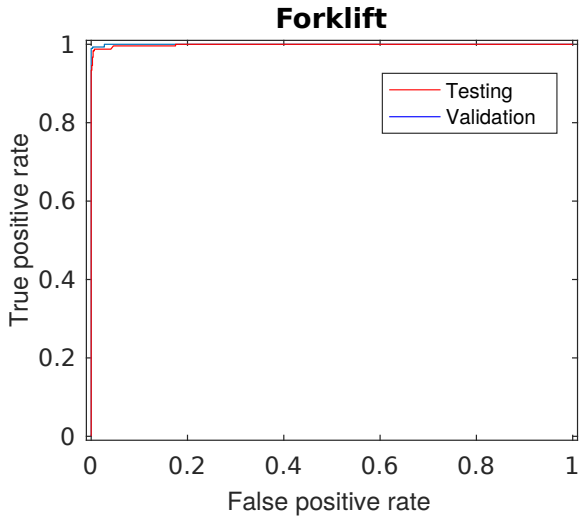
Table 5.2.: Details of the training phase: Number of training, testing and validation images, training iterations, number of support vectors are shown for each class.

Forklift	Pallet	Human	...	Acc
286	0	2	Forklift	0.993
1	146	0	Pallet	0.993
1	0	181	Human	0.994

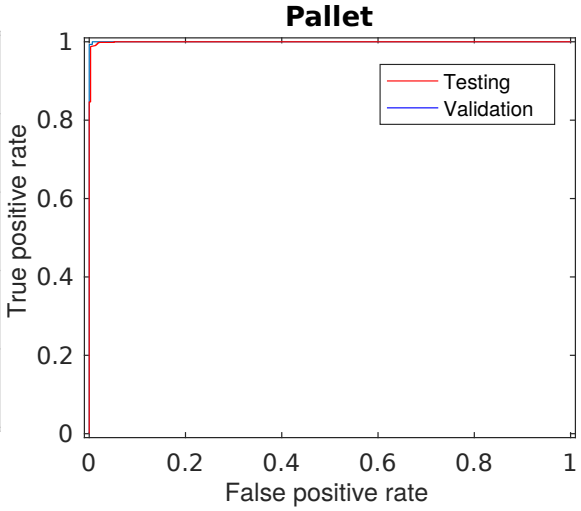
Table 5.3.: Confusion matrix obtained for the testing dataset. *Acc* denotes the overall classification accuracy for the given class.

Table 5.4 shows the confusion matrix obtained for the validation dataset. The classification results are outstanding particularly if one considers the notable differences of the image data recorded with an Asus Xtion camera inside the testing environment and the image data found on the internet. Photos obtained from this source are typically recorded with cameras having large sensors and hence provide images of higher quality and information density. All classifiers achieve accuracies better than 97%. This is a notable progress compared to common methods using HOG features ([39]).

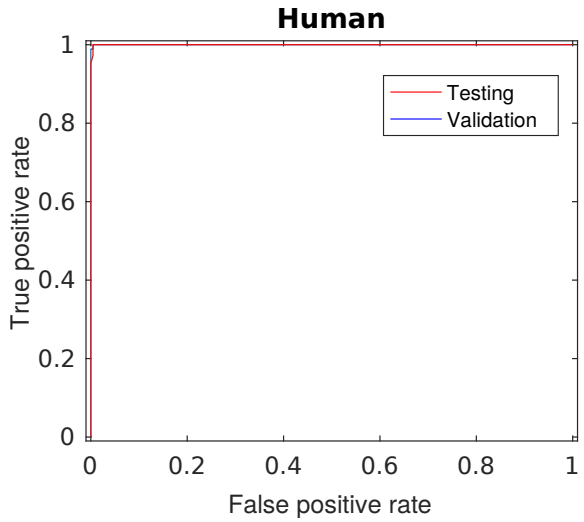
We observed that the CNN features obtained from the fc-7 layer provide a substantial benefit for distinguishing different classes. Experiments with adjacent layers showed comparable results whereas those extracted at lower ones performed worse. Our system relies on linear kernels for the SVMs which did not show any disadvantages in our exper-



(a) ROC curve for object class *Forklift*.



(b) ROC curves for object class *Pallet*.



(c) ROC curves for object class *Human*.

Object Class	AUC Test	AUC Val
Forklift	0.9998	0.9987
Pallet	0.9999	0.9994
Human	0.9999	0.9998

(d) AUC values for all classes.

Figure 5.10.: ROC curves for all object classes. The figures show the results of the prior training (red) and the ones of the validation (blue). Table (d) presents the AUC values obtained for test and validation dataset respectively. It can be clearly seen that the trained classifiers are optimal in regards of the ROC curves being obtained. All curves remain close to a zero FP-rate while achieving almost a 100% TP-rate. Consequently the resulting areas under the curve (AUC) are almost 1.0 which is best from a classifier's point of view. The evaluation of the validation dataset will provide insights whether the learned models are able to generalize from the training data.

Forklift	Pallet	Human	...	Acc
238	1	2	Forklift	0.988
10	636	6	Pallet	0.976
2	0	105	Human	0.981

Table 5.4.: Confusion matrix obtained for the validation dataset. *Acc* denotes the overall classification accuracy for the given class.

Figure 5.11.: Examples of training dataset. The data contains images of various kinds of forklifts, palleted goods and humans. It is solely originated from publicly available sources.

iments. Samples of our training/testing as well as validation datasets are illustrated by the Figures 5.11 and 5.12 respectively.

5.5.5. System requirements

The experiments were carried out on a Dell E6320 laptop equipped with an Intel Core i7 (dual core CPU) and 8GB of RAM. We did not use any GPU or other hardware acceleration. Table 5.5 summarizes the mean run time of the entire system and the individual components.

Figure 5.12.: Results obtained on the validation dataset captured in a warehouse environment. Our system is able to detect and recognize objects of the classes: *forklifts*, *humans* and *palleted goods* under varying poses and illumination conditions.

	Component	Time [ms]	
		Mean	Max
	Object recognition		
	Ground plane estimation	16.91	17.19
	Range scan generation	9.53	9.88
	Height scan generation	5.12	6.31
	Curvature detection	0.31	0.4
	Segment estimation	10.86	19.2
	Object retrieval	0.91	1.74
	ROI estimation	23.09	24.11
	CNN feature extraction	73.49	156.09
	Image/ROI classification	18.81	29.67
	Non-max suppression	2.11	2.55
	Range scan annotation	1.71	1.93
	Total	**162.85**	**269.07**

Table 5.5.: Performance of our system and individual components.

5.5.6. Discussion

We observed that the utilized one-versus-all schema for multi-class classification is well-suited for our application. Even though, multiple authors reported slightly worse results for this compared to the one-versus-one schema [5, 81], we could not observe notable differences. Thanks to this, the runtime can be reduced since we have to train solely one SVM for each class. The one-versus-one schema requires $K\,(K-1)/2$ classifiers which scales quadratically in the number of classes K. This becomes notable in both, training and online classification. The largest performance speed up can be ascribed the linear SVM kernel being utilized. We demonstrated that the presented approach can be integrated into robotic systems allowing obstacle detection and classification at a frame rate of about $4-6\,Hz$. A large number of applications using CNNs require GPUs for performance reasons. Our experimental platform, an automated guided vehicle, is equipped with an electric engine and would, in fact, have sufficient power to serve a GPU. However, we omitted using this architecture for generalization reasons since it is not always available on mobile robots. The detection does not have to run at full frame rate for our application. Once an obstacle is detected and classified, a tracker can be initialized to follow its movements. This is why it is generally not necessary to apply the CNN feature extraction to each detected ROI. It might be asked if the increased computational requirements entailed by the use of CNNs can actually be justified given our reported results (see Table 5.4) or if these are too powerful for the presented application. Considering the application of detecting and classifying humans in images [39], we can observe substantive improvements. It should be noted that our system is able to detect and classify other objects as well. We expect that the use of CNNs allows more scalability turning it into an

indispensable solution for classification with a larger number of classes (e.g. more than 100).

5.6. Chapter Conclusions

This chapter introduced an approach to obstacle detection and classification using methods of deep learning. We motivated the benefit of incorporating environment-specific knowledge. The presented algorithms were applied on an AGV inside a warehouse while specifically learning common obstacle classes expected for this type of environment.

Our prior segmentation enables to efficiently generate object proposals by exploiting geometric properties of objects being observed in a scene. These geometric features are fused with textural properties of objects which are extracted based on deep learned features from a CNN and subsequently evaluated by a multi-class SVM. The entire training stage uses data solely obtained from publicly available image data and a-priori known object dimensions. Our system is evaluated on sensor data originated from an environment which neither the CNN nor the SVM have ever seen before which emphasizes our system's strengths of generalization and potential application in a-priori unknown logistic environments. We expect this to be important for automated forklifts in warehouses providing a valuable fundamental for intelligent robotic navigation. In the following chapters we will present potential applications which highly benefit from our object recognition system.

Chapter 6.

From Objects to Semantic Maps[1]

[1] The content of this chapter has already been published in [78]

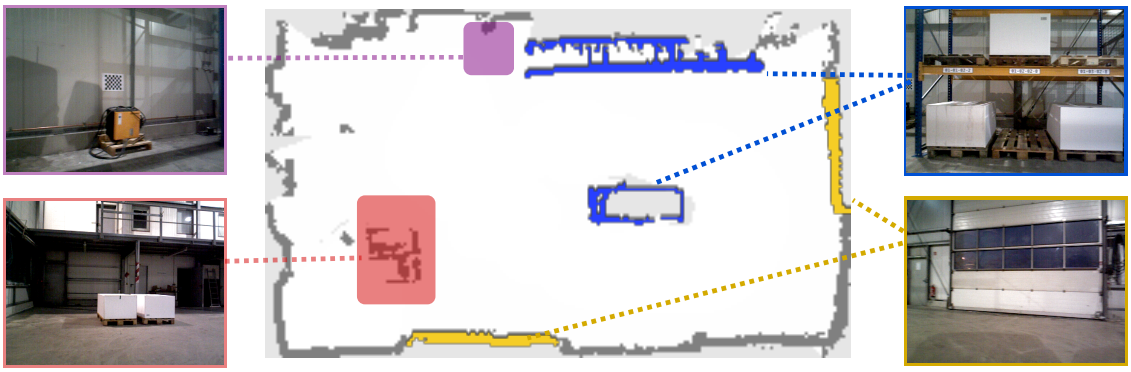


Figure 6.1.: Illustration of an occupancy grid map and overlay of semantic annotations. Our algorithm automatically detects *charging stations*, *reloading areas* (left), *racks* and *gates* (right). These object classes are of particular interest for logistic environments.

6.1. Motivation

Mobile robots require spatial representations of their environment in order to enable autonomous navigation. Geometric maps are commonly used with occupancy grid maps being the state of the art. These enable basic robotic navigation, but off-the-shelf lack the availability of environment-specific information. This chapter presents an approach to semantic mapping which augments common geometric representations by objects being relevant for logistic environments.

Navigation algorithms such as path planning, obstacle avoidance and localization benefit from the additional knowledge about obstacles in the surrounding environment. For example the presence of *gates* in warehouses might pose a problem for AGVs due to suddenly appearing humans, manually driven vehicles or even other AGVs if there are no further communication systems available. Being aware of the presence of *gates*, algorithms are able to incorporate this which potentially results in safer and more robust navigation.

Semantic maps provide a fundamental resource for AGVs operating as mobile service robots in logistic environments. In this way, they enable a vehicle to go e.g. to the «*storage bin - 12-02-01*», «*reloading point - 5*» or «*charging station - 7*» with all positions being stored in the map. Close-to-market systems solve this with the help of a human supervisor manually annotating maps or in rather exceptional cases using voice and gesture commands [131]. The work load induced by this process is substantive and expensive, particularly for the initial setup of such a system. Thus the automatic annotation of maps is appreciated for the introduction of AGVs in warehouses and is highly beneficial in the context of Industry 4.0.

Our approach aims at building semantic maps online while steering an AGV inside a warehouse. This makes highly-efficient object detection, SLAM and map inference indispensable.

The key contributions of this chapter can be summarized as follows:

- An online semantic mapping framework with a high performance
- Integration of a SLAM framework for estimating globally consistent positions of semantic annotations
- An efficient solution for handling uncertainties of object recognition
- Experimental evaluation in an a-priori unknown logistic environment

This chapter is organized as follows. First we will give an overview of the state of the art. Having motivated the objects being of interest for semantic mapping, we will show how these can be integrated in our graph-based SLAM system. Subsequently it will be shown how uncertainty arising from object recognition can be incorporated. Eventually it will be explained how semantic grid maps can be rendered from this information. We present our experimental results and discuss the key contributions of this chapter.

6.2. Related Work

The existing work focusing on the topic of this chapter can be categorized into different research fields among the computer vision and robotics communities, specifically these are: *object-based SLAM*, *semantic mapping* and *place categorization*. We will provide an overview for each research field and will summarize how our work is related to it.

Object-based SLAM

Thanks to the availability of highly efficient graph-based optimization libraries such as *g2o* [100] and *iSAM* [89], the application of Visual SLAM for online operation is rendered possible. The state of the art in SLAM focuses on the optimization of poses and landmarks. Loop closures are commonly identified by means of local image features such as *SURF* [35]), geometric features such as *GLARE* [73], CNN landmarks [157] or holistic image matching [117]. There is only a limited number of existing work utilizing semantic information in SLAM. Strasdat et al. presented *SLAM++* which explicitly uses objects instead of landmarks [151]. Their approach is able to continuously track the camera pose while mapping objects in the surrounding environment. The authors mention that a relocalization based on object matching takes places once the tracking is lost. It is further emphasized that thanks to object representations the memory consumption is significantly reduced compared to dense geometric representations. The generic object pose estimation used in their work requires a GPU in order to enable close high performance. Civera et al. presented a fast mapping and loop-closure detection system with high performance omitting the use of GPUs [35]. The authors use SURF features quantized inside a Bag-Of-Words approach for recognizing a-priori learned feature representations of objects which are incorporated in the map estimate. Due to the use of a monocular camera

the object poses are estimated by 2D-3D feature correspondences and a perspective n-point algorithm.

Semantic Mapping

Nüchter et al. proposed a semantic mapping approach that applies plane segmentation to 3D LIDAR data [126]. Though enabling high performance, it is restricted to planar regions. Stückler et al. presented an object-based extension for SLAM being able to generate dense 3D object maps at a relatively high frame rate [152]. The authors utilize simple region features extracted from RGB and depth data to segment object regions from point clouds. The objects are classified using random decision forests and subsequently used to concurrently track the camera's pose and estimate the 3D poses of the objects. Even though, the approach is highly related to ours since the authors also make use of the geometric properties of objects, we expect that the classification accuracy obtained with simple RGB region features can be substantially increased by CNNs.

The work in the field of semantic mapping described above aims at fitting particular geometric primitives to the input data [126] or recognizing specific objects with the result of dense 3D maps. A further category of semantic mapping focuses on the detection of particular objects or regions being of interest in a robot's working space. Grimmett et al., for example, investigates the automatic mapping of parking spaces in garages based on lane marking detection in camera images [60]. The authors motivate that the identified parking spaces can subsequently be used for automated valet parking of driverless vehicles in car parks. Beinschob et al. presents an approach for mapping logistic environments including the detection of storage places and the automated generation of road maps for AGVs [14]. Their work presents an comprehensive object recognition framework segmenting horizontal and vertical pillars of high-level racks and fitting storage bins of defined sizes. The 3D input data is obtained from a tilting 2D laser range finder.

Semantic Place Categorization

Semantic mapping in the context of labeling spatial sub-spaces with object categories has been investigated by several authors of the mobile robotics field. The early work of Mozos et al. extracts features from 2D laser range data and trains Adaboost classifiers for distinguishing places of different categories. Pronobis et al. extends this by fusing data from 2D laser range finders and cameras in a SVM-based place classification [136] which is extensively evaluated on a publicly available dataset [135]. Hellbach et al. suggest the use of Non-Negative Matrix Factorization (NMF) to automatically extract relevant features from occupancy grid maps which are subsequently utilized for identifying place categories [71]. More recently, Sünderhauf et al. proposed an approach to semantic categorization using visual sensors [153]. The authors make use of deep-learned features extracted with a CNN which are passed to a random forest classifier. A continuous factor graph model is used to infer from the object observations. The area being covered by the

camera's field of view is labeled according to the output of the graphical model. The authors integrate their approach with grid mapping algorithms, such as GMapping [64] and Octomap [80] in order to build geometric maps with the free space being supplemented by place category labels.

Summary

The existing methods for SLAM and semantic mapping either focus on the use of objects as landmarks [150] or provide fewer accuracy in large-scale object classification [35, 152] due to the image features being used. The majority of related work in semantic mapping considers simple geometric features such as planes or focuses on the extraction of one object class being of interest for the investigated environment type (e.g. parking [60] or storage spaces [14]). The state of the art in semantic place categorization is extensive but differs from our approach since maps are annotated on the scale of rooms or buildings rather than objects.

To our knowledge there exists no prior work on online semantic mapping with the result of light-weight geometric maps with occupied space being assigned object labels at the accuracy of deep-learned models. Also we found the application of semantic mapping with recognition of multiple object classes in logistic environments to be novel in the robotics research.

6.3. System Overview

6.3.1. Architecture

We utilize the object recognition system of Chapter 5 and the SLAM framework of Chapter 4. The object recognition module generates a plain range scan as well as an annotated range scan. The former is passed to the SLAM module which in combination with the vehicle's odometry is used to constantly estimate the vehicle's path and updates the occupancy grid map of the traversed environment. The semantic mapping module maintains a graph of object points based on the annotated range scans and SLAM poses. An overview of the interfaces of the semantic mapping module is given by Fig. 6.2.

6.3.2. Object Recognition

The structure of the object recognition system described in Chapter 5 is slightly modified in order to account for large, static objects which might not be completely captured by the image sensor and hence does not necessarily satisfy our constraints for segmentation, which are discontinuities in the range and height data. Also, these might not be detected if a large object remains close to the image sensor.

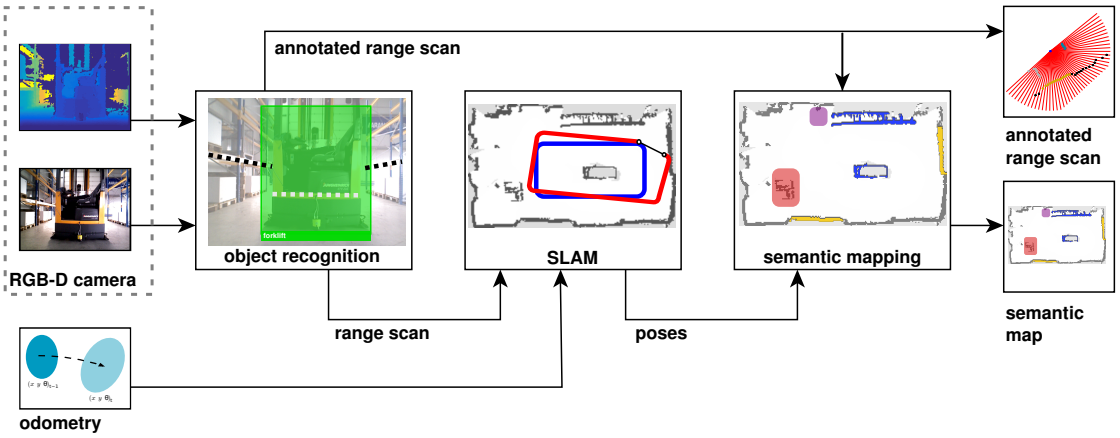


Figure 6.2.: The input RGB-D is utilized for generating a 2D range scan and searched for objects of interest. This data is fused into an annotated range scan serving as input for the semantic mapping and other navigation algorithms. The plain 2D range scan as well as the odometry is passed to our SLAM framework to estimate the vehicle's trajectory and an occupancy grid map. The global coordinates of the object detections are constantly maintained by the semantic mapping algorithm given the SLAM path and the annotated range scan.

Class	**Recognition methods**		**Bounding box**		**Obstacle type**	
	Detection	Classifier	ROI	Image	Dynamic	Static
Pallet	CNN+GEOM	SVM	X		X	
Rack	CNN	SVM		X		X
Gate	CNN	SVM		X		X
Human	CNN+GEOM	SVM	X		X	
Forklift	CNN+GEOM	SVM	X		X	
Charging St	-	-		X		X
Reloading A	Group Merging	-				X

Table 6.1.: Overview of all considered object classes. The table provides the required recognition methods, the evaluation window (ROI or entire image) and the obstacle type (dynamic or static) for each class. The method CNN+GEOM indicates the combined method of deep learned appearance-based features with geometrical ones. Reloading areas are inferred from the observation of multiple, close-by pallets. All dynamic objects are detected by the system but not incorporated in the final map.

The object recognition for our semantic mapping system considers the following object classes: *forklift*, *human*, *pallet*, *rack* and *gate*. The dynamic object classes *forklift* and *human* are detected but not incorporated in the semantic map since their positions are

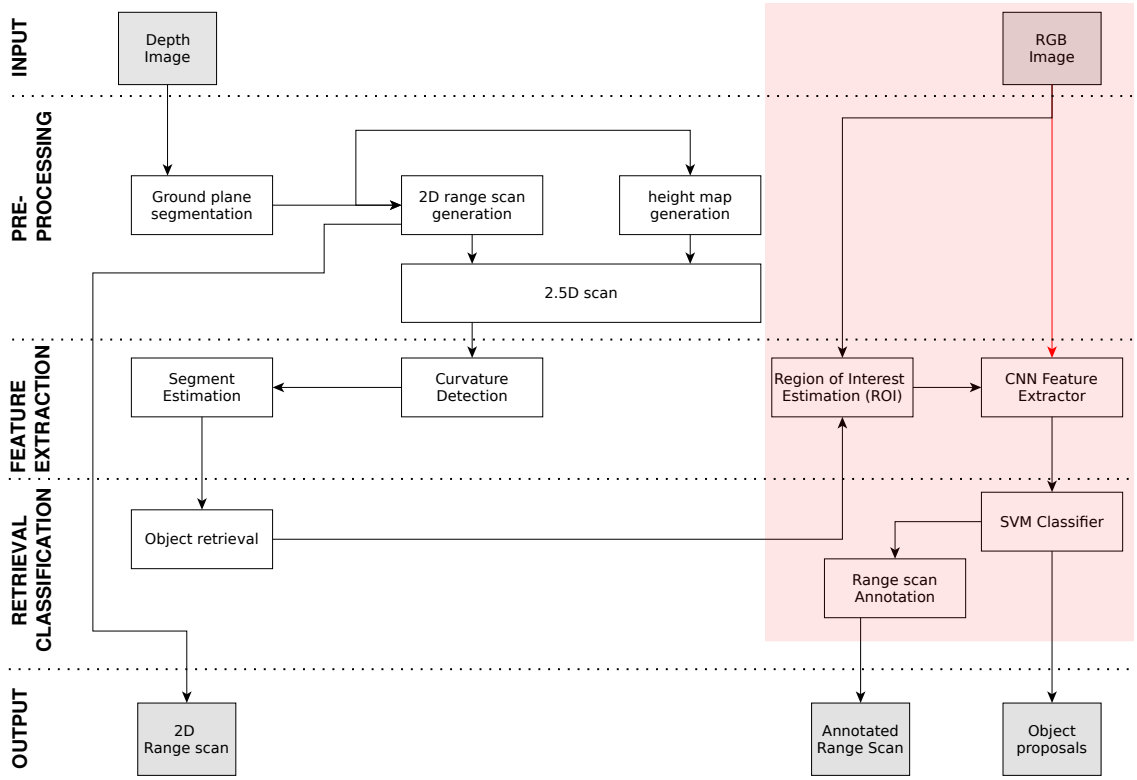


Figure 6.3.: The existing object recognition system of Chapter 5 is slightly modified which is highlighted red. The additional red edge accounts for large objects being recognized based on analyzing the entire RGB image.

subject to change and hence do not contribute to navigation algorithms such as localization and path planning.

Our object recognition system, described in Chapter 5, is extended as follows in order to account for the requirements of semantic annotation. First, we train a set of two multi-class Support Vector Machine (SVM) for dynamic objects (*human*,*forklift*,*pallet*) and large static objects (*gate*, *rack*) respectively. Therefore we make use of one set of binary SVMs for ROI classification and another set for classifying entire images. It is possible to recognize multiple dynamic and one static object in a single image, e.g. pallets placed inside a rack. In this case, the corresponding measurements are assigned a mixture label (e.g. *pallet-rack*) in order to allow other applications to decide for the relevant class. Observations of multiple pallets are used to infer reloading areas. Table 6.1 provides an overview of all object classes being incorporated by our semantic mapping framework. In addition to the mentioned objects, we automatically detect charging stations based on checkerboards being placed on top of them. This is done based on a checkerboard detection system of the Robot Operating System (ROS) in order to ensure a high accuracy in the relative pose estimation for initialization the position of a robot and calibrating the cameras.

6.4. Integration of Objects in SLAM

6.4.1. Pose Graph SLAM

In Chapter 4 we defined the state vector $\mathbf{x} = (x, y, \phi)^T$ describing a particular pose $\mathbf{x_i}$ of a pose graph with $\mathbf{u_i}$ and Σ_i being an odometric motion and associated measurement covariance respectively. We further introduced the odometry constraints $P(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{u}_i)$ and the loop closure constraints $P(\mathbf{x}_j|\mathbf{x}_i, \mathbf{u}_{ij})$ representing the graph's edges e_i^{odo} and e_{ij}^{lc} respectively. Recalling the fundamentals of Chapter 4, we constantly incorporate the vehicle's motions and loop closure detections. The pose graph is subsequently optimized according to the following equation:

$$X^* = \underset{X}{\operatorname{argmin}} \sum_i \left\| e_i^{odo} \right\|^2_{\sum_i} + \sum_{ij} \left\| e_{ij}^{lc} \right\|^2_{\Lambda_{ij}} \tag{6.1}$$

6.4.2. Object Proposals

Given the RGB-D input data we continuously detect objects which are incorporated during the mapping process. Each object detection O_k* is linked to a reference pose $\mathbf{x_i}$ and associated with a set of measurement points representing the object. The reference poses are continuously updated during the graph optimization. The resulting differences of the poses typically occur as a result of loop closures and when for example re-traversing longer trajectory paths from an opposite direction. These changes have to be taken into account for referencing the detected objects with respect to the pose graph and eventually to the global map. The object points O_k* are kept in the coordinate frame of the corresponding pose $\mathbf{x_i}$. For higher-level layers of our semantic mapping framework, we frequently transform the object points into the global coordinate frame of the map with the origin being centered at the first SLAM pose $\mathbf{x_1}$.

It is generally possible to include the object points in the graph optimization, similar to joint map and pose refinement as detailed in Chapter 4. However, this additional optimization is quite expensive and hence might entail a significant performance loss for online mapping which is why we omit this step here.

6.5. Probabilistic Grid Mapping with Semantic Annotation

This Section explains our method for transferring a graph-based map structure into a global semantic grid map. First we will introduce how a regular spatial structure is generated and the object observations are transfered. The subsequent inference on the initial estimate of the map will be discussed.

6.5.1. From a Pose Graph to a Semantic Grid Map

Given an optimized graph with poses $X*$ constructed as described in Section 6.4, all points $\mathbf{O_k}$ associated with the pose $\mathbf{x_i}$ are transformed into a global map reference which results in a transformed set of points $\mathbf{O_k}*$. We define m as our map with m_j representing a grid cell. The (x, y)-coordinates of the center-of-mass of m_j are expressed by $m_{j,x}$ and $m_{j,y}$ respectively. The transformation of each point $o_k^{(l)}$ into the map coordinate frame is carried out as follows:

$$\begin{pmatrix} m_{j,x} \\ m_{j,y} \end{pmatrix} = \lfloor \left[\begin{pmatrix} o_{k,x}^{(l)} \\ o_{k,y}^{(l)} \end{pmatrix} - \begin{pmatrix} c_x \\ c_y \end{pmatrix} \right] \tau_{res}^{-1} + \frac{1}{2} \begin{pmatrix} s_x \\ s_y \end{pmatrix} \rfloor \tag{6.2}$$

where s_x and s_y refer to the size of the entire grid map, c_x and c_y to the origin of the map and τ_{res} to the grid resolution. The origin c of the map is defined by the first pose of the SLAM graph, thus $c = \mathbf{x_1}$. The value τ_{res} has to be set appropriately and suit the underlying RGB-D sensor characteristics.

Each grid cell m_j consists of a probability distribution $p(m_j)$ with each element describing one object class. Having estimated the corresponding grid cell m_j for each point $o_k^{(l)}$, we are able to update $p(m_j)$ for m_j given our observations $p(o_k^{(l)})$. Note that for occupancy grid maps m_j describes *one* occupancy probability, whereas m_j in our approach describes a discrete probability distribution of object affiliations. The states *unknown*, *occupied,* and *free* can be inferred from our $p(m_j)$ though. Each distribution $p(m_j)$ is estimated by those points of $O*$ that end inside m_j according to Eq. 6.2. We define this subset of $O*$ as O_j*.

$$p(m_j) = \eta \sum_k p(o^{(l)}),\ k \in O_j \tag{6.3}$$

where η denotes a normalization factor. The initial state of each m_j is a uniform distribution.

Our approach implements an end point model with each measurement point being directly assigned to the corresponding grid cell omitting cells inside the ray's cone. This is necessary in order to ensure an efficient map update. A number of robotic navigation tasks such as path planning often require more comprehensive environment descriptions which explicitly consider free space. For performance reasons we avoid expensive ray-casting models accompanied with this and suggest building those subsequently or in parallel but slightly delayed.

Since the map size and the number of object likelihoods stored in the cells easily increase in large scale environments, it is recommended to use sparse instead of dense matrices. Particularly high-resolution maps usually encode a lot of free space. The point transformations being applied frequently are highly optimized thanks to efficient matrix operations.

6.5.2. Inference

The initial estimate of the semantic map m typically consists of uncertainties which can occur due to the following reasons. Accumulated errors of the pose estimation might remain after the pose graph optimization if the distance traveled by the robot from the last loop closure is too long which is why errors cannot be adequately corrected. This effect can potentially be noticed in path segments at the end of a trajectory which results in object detections being incorrectly assigned to spatial grid cells m_j. In addition to that, the map estimate is induced by uncertainties in the object recognition due to the detections of multiple object classes. This effect can be observed more frequently, particularly in the presence of object classes of similar visual appearance and geometric properties as, for instance, *wall* and *gate*. It affects the distributions of object affiliations $p(m)$, rather than the spatial positions of m. Uncertainties due to this reason are not just a drawback: they also allow to identify false-positive object detections based on multiple observations, potentially originated from varying perspectives or different path segments.

Due to the above mentioned reasons, it is necessary to explicitly account for the uncertainties in the semantic map estimation. For this purpose, we make use of the Shannon entropy which is separately estimated for each grid cell as follows:

$$H(m_j) = \sum_k p(m_j^{(k)}) \log(\frac{1}{p(m_j^{(k)})}) \tag{6.4}$$

The Shannon entropy provides a beneficial measure of uncertainty as revealed by the probability distributions of the semantic grid cells. The measures of $H(m_j)$ are utilized to infer the final class label from $p(m_j)$. Providing a value of $H(m_j) < \lambda_{high}$, we set the final label of m_j* to the label with the highest probability as follows:

$$\begin{aligned} p(m_j)_{orig} &= max(p(m_j^{(1)}), ..., p(m_j^{(k)})) \\ p(m_j)_{surf} &= max(p(m_{j,surf}^{(1)}), ..., p(m_{j,surf}^{(k)})) \end{aligned}$$

old eq.

$$p(m_j*) = \begin{cases} H(m_j) > \lambda_{high} : & p(m_j)_{orig} \\ \lambda_{low} \leq H(m_j) \leq \lambda_{high} : & p(m_j)_{surf} \\ H(m_j) < \lambda_{low} : & p(cl_{unknown}) \end{cases} \tag{6.5}$$

$$p(m_j*) = \begin{cases} H(m_j) < \lambda_{low} : & p(m_j)_{orig} \\ \lambda_{low} \leq H(m_j) \leq \lambda_{high} : & p(m_j)_{surf} \\ H(m_j) > \lambda_{high} : & p(cl_{unknown}) \end{cases} \tag{6.6}$$

$$p(m_j)_{surf} = \eta \sum p(m_k),\ k \in m_{j,surf} \tag{6.7}$$

The differentation of high-, low- and unconfident estimates is required to ensure robust class label assignments. A large entropy results from a larger diversity and hence

more uncertain class label estimation. Those cells possessing an entropy measure $H(m_j)$ above a threshold λ_{high} are directly assigned the class label *unknown*. For low entropy measures we assume a high-confident estimate and set the class label to the best vote of the cell which refers to $p(m_j)_{orig}$. Estimates with values in the range of λ_{low} and λ_{high} are considered for further investigation. We therefore analyze adjacent cells of m_j sharing the same surface entity. The crux here is that we do not incorporate all adjacent grid cells of m_j, but those that are likely originated from the same physical object in the world based on height and surface constraints.

Semantic grid cells providing a high confidence $H(m_j)$ are directly adopted omitting the consideration of adjacent cells. This is done in order to avoid confusions in distribution $p(m_j)$ which can occur, for example, when small objects such as rack poles contribute to only a single or a low number of grid cells. The object labels of those are risked to be filtered out if it is merged with a potentially diverse neighborhood.

The presented method enables efficient inference of final class labels m_j* for each entity of the semantic map. Technically it is possible to store the probability distribution m_j. However, this requires a large amount of memory for large-scale environments and increasing numbers of object classes. We further expect that the majority of applications utilizing semantic maps need direct knowledge about the underlying object classes, rather than a distribution of object likelihoods. In order to incorporate the uncertainty in the class estimate, it is recommended to store the entropy values $H(m)$ along with m_j*.

6.6. Experiments

This section presents the experimental results obtained using the object recognition and semantic mapping algorithms as explained in this and the preceding chapter.

6.6.1. Setup

We evaluate the presented approach by experiments carried out in a warehouse which consists of a multitude of common objects expected for this type of environment. The data is collected with a reach truck which has been fully automated for a research project. For the purpose of generating a semantic map, we manually steer the vehicle inside the warehouse. The RGB-D image data utilized by our system is recorded using an Asus Xtion camera which is aligned sidewards with respect to the vehicle's direction of travel. The parameter selection for our experiments is summarized in Table 6.2.

6.6.2. Classification

We further evaluated the classification accuracy achieved based on the object retrieval priors and the appearance-based classifier. Table 6.3 provides details about the training phases of the SVMs including iterations, number of support vectors and amount of training images. As already mentioned, we make use of a set of 500 negative training sam-

Component	Description	Variable	Value	Ref
SLAM, Semantic Mapping	Grid map resolution	τ_{res}	$0.05\,m$	Eq. 4.19, Eq. 6.6
Object recognition	Parameter selection: see experimental section for object recognition (Sec. 5.5)	–	–	Tab. 5.1
Semantic mapping	Upper threshold for label inference	λ_{high}	1.2	Eq. 6.6
	Lower threshold for label inference	λ_{low}	0.3	Eq. 6.6

Table 6.2.: Parameter selection for experiments. The tables provides references to more exhaustive descriptions of the parameters (right column).

ples capturing objects not being recognized by our system. For each object class being trained we randomly sample about 150 images from this set and another 150 samples are selected from the training images of the other object classes considered by our system. The dataset is split into 50% training and 50% testing images. The entire training and testing dataset is obtained from publicly available image sources being supplemented by our own image collection. These images are edited in the way such that only the object of interest is visible. We captured an additional validation dataset consisting of RGB and depth images in the mentioned warehouse.

Class	# Images			# Iterations	# Support Vectors
	Train	Test	Val		
Forklift	288	288	—	152	76
Human	182	182	—	171	70
Pallet	147	147	652	89	67
Rack	108	108	904	188	75
Gate	122	122	365	192	69

Table 6.3.: Details of the training phase: Number of training, testing and validation images, training iterations, number of support vectors are shown for each class. The dataset for validating our semantic mapping algorithm does not contain humans or forklifts which is why the validation dataset does not contain images of these classes. Note that the indicated sample numbers refer to positive images containing the referring object class. In addition to that we randomly select 300 negative samples for each class obtained from the other object classes and additional publicly available images.

The ground truth for the validation dataset is obtained by manually labeling the input RGB images. We assume that at least 50% of an object of interest has to be visible in

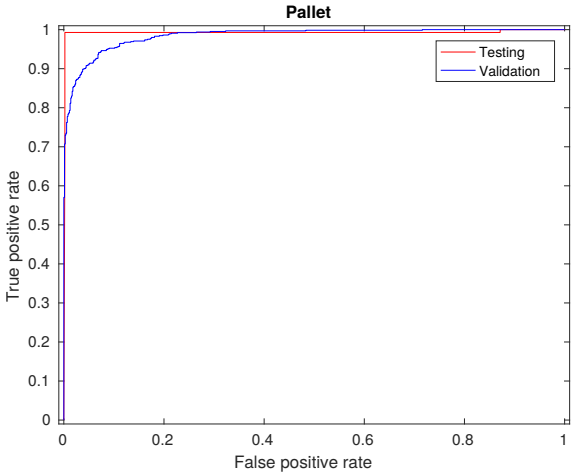
order to assign an image the corresponding class label. This addresses the classification of entire images (for gates, racks) as well as the ROI-based classification of objects such as forklifts, pallets and humans. The object segmentation expects this amount in order to minimize false-positive detections while simultaneously enabling as many as possible recognitions of objects in the presence of clutter and and when remaining close to the image boundaries. The results are shown in Figure 6.4. The values of the ROC curves are obtained by varying distance thresholds τ_{svm} of the SVM classification. Note that none of the training and testing images captured inside the warehouse are used for the validation. It is shown that the AUCs for the training datasets achieve high values which confirms our results of Chapter 5. The AUCs for the classes gate and rack again achieve promising values above 0.96 on the validation data. The class rack, however, performs slightly worse with an AUC value about 0.85 which can be referred to the following reasons. First, the drivable paths around racks are typically quite narrow due to the tight setup of high-level racks in warehouses. This in combination with the restricted field of view of our RGB-D camera (less than 60, see also Appendix A.1) entails that only a limited part of a rack becomes actually visible. The partial observation and occlusion of objects are addressed in our training stage by cropping input images. However, the dominant vertical pillars which provide a distinctive feature for racks are not observed in each camera image when passing by a rack. This entails a slightly reduced classification performance compared to the other classes. We suppose that RGB-D cameras with larger field of views could substantially improve the accuracy for rack class and probably also for the gate class. Second, the distinction of pallets and the surrounding rack structure provides a challenge. As the entire image is used as input for the classifier, the pallets increase the uncertainty for correctly recognizing racks, particularly if the camera remains close to the rack which entails that single pallets might take up the majority of the camera's field of view.

6.6.3. SLAM

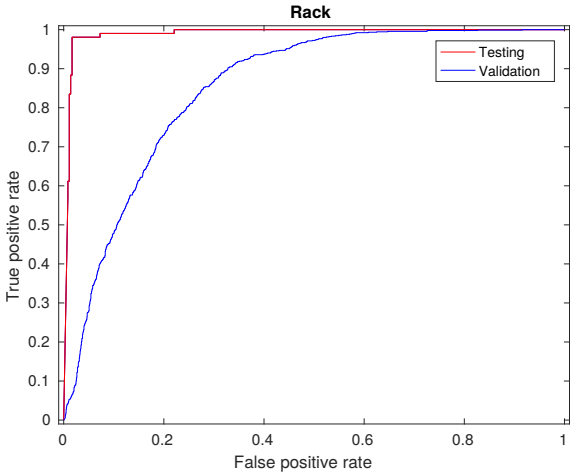
Our graph-based SLAM provides the fundamental for the spatial mapping of objects and the estimation of the vehicle's pose w.r.t. the environment 4. In Section 4.6.1 we described the underlying metrics being used. The ground truth is obtained using a laser range finder and the algorithms described in [74]. The accuracy of the estimated trajectory is expected to be below $0.04m$ thanks to the high measurement accuracy of the laser range finder and a joint map and pose refinement [74]. The results of the presented camera-based SLAM approach is compared to the ground truth based on the absolute trajectory error (ATE) [19]. The mean error is $0.26m$ in the position and about $0.07rad$ in the orientation respectively. A more exhaustive description of the utilized SLAM algorithm and evaluation methods is provided by Chapter 4.

6.6.4. Semantic Labeling

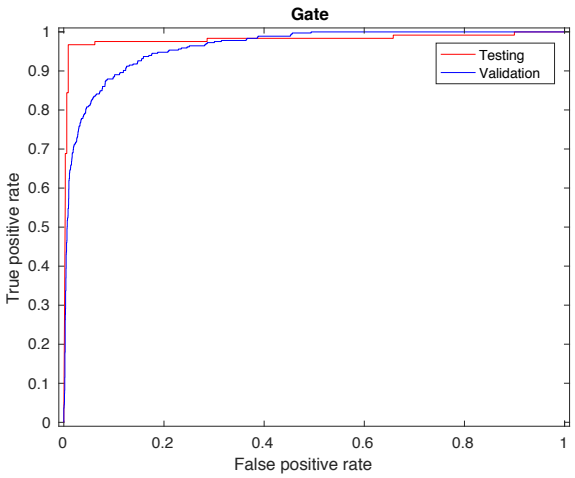
In addition to the evaluation of the accuracy of our SLAM framework, we investigated the key contribution of this chapter which is the semantic annotation of the occupancy grid



(a) ROC curve for object class *Pallet*. $AUC = 0.984$



(b) ROC curve for object class *Rack*. $AUC = 0.852$



(c) ROC curve for object class *Gate*. $AUC = 0.961$

Object Class	AUC Test	AUC Val
Pallet	0.999	0.984
Rack	0.989	0.852
Gate	0.981	0.961

(d) AUC values for all classes.

Figure 6.4.: ROC curves for all object classes captured on the validation dataset. The figures show the results of the prior training (red) and the ones of the validation (blue). The AUC values for all classes are presented in Table (d). For the performance of the classification of forklifts and humans, please refer to Fig. 5.10 in Chapter 5. It can be observed a slightly worse AUC value for the rack class, particularly compared to the result obtained in the previous chapter and for the other classes. We suppose that this is due to significant differences in the viewpoints for the training and validation images.

(a) Semantic Map with ground truth labels.

(b) Initial semantic labels without optimization .

(c) Initial semantic labels with optimization based on uncertainty estimation and incorporating adjacent cells.

Figure 6.5.: The figures demonstrate our experimental results obtained in a warehouse environment. The ground truth labels are obtained by manually annotating all non-empty grid cells.

map. Our ground truth is generated by manually labeling the grid cells of our prior map provided by the SLAM algorithm. The evaluation of the semantic annotation is carried out by comparing the ground truth to those being obtained initially (unoptimized) and those being optimized. The results are visualized in Figure 6.5 and summarized in Table 6.4.

Method	# Matching labels	Accuracy
Unoptimized labeling	10310 / 11278	0.914
Optimized labeling	11021 / 11278	0.977

Table 6.4.: Semantic labeling results.

The results reveal that thanks to our object recognition system we are able to achieve a reasonable performance of about 91% without using any optimization. However, the presented method for inferring class labels enables an important increase of the accuracy to almost 98%. Hence this optimization procedure contributes to a more stable semantic map generation. The uncertainties become particularly apparent around the gates (orange labels, Fig. 6.5). Here, the adjacent cells sharing continuous surfaces are incorporated which results in a notable uncertainty reduction in these areas. The mapping of racks is affected by pallets being placed inside these. Depending on the distance and the amount of dominant rack structures being detected our system only recognizes the pallets inside the racks. The observations consequently vote for the same grid cells which entails in an increased uncertainty. Taking adjacent cells into corporation also improves the labeling for this class.

6.6.5. Runtime

In our experiments we use a Dell Latitude E6320 laptop equipped with an Intel Core i7 dual-core processor and 8GB RAM. The processing time required by each system component is detailed in Table 6.5. Our system runs at about $4\,Hz$ in the mean and never drops below $2\,Hz$. Note that some of the components do not necessarily have to run all the time, e.g. the occupancy grid map can be re-estimated at a much lower rate or can be even be done only once at the end of the trajectory. The semantic map is estimated independently of the occupancy grid map and requires significantly less computation time due to the end-point model being utilized instead of expensive ray-casting. Also the rates of the loop closure detection and graph optimization can be reduced to e.g. $0.5\,Hz$. Based on these simplifications and extensive use of parallel programming, we observed de facto run times of about $5 - 10\,Hz$.

6.6.6. Discussion

Our object segmentation and retrieval provides an efficient fundamental for semantic mapping. The classification results obtained for the validation dataset demonstrate the

	Component	Time [ms]	
		Mean	Max
	Object recognition	**162.85**	**269.07**
	SLAM	**85.07**	**176.66**
	Loop closure detection	23.44	37.89
	Graph optimization	22.35	52.8
	Occupancy grid mapping	39.28	85.97
	Semantic Annotation	**25.32**	**57.07**
	Object point transformation	3.45	3.67
	Label assignment	8.91	9.86
	Inference	12.96	14.22
	Total	**273.24**	**498.8**

Table 6.5.: Performance of our system and individual components.

high generalization performance of the CNN and SVM. This becomes particularly obvious since the appearance of many objects on the training images differ from those in the testing warehouse. The detection of pallets and gates is outstanding, the one for racks is slightly worse. We expect that this is due to the fact that our training data mainly consists of racks being fully equipped with pallets which is not the case for all racks in our testing environment. Thanks to the descriptive power of the CNN features, we are able to achieve a high classification accuracy for SVMs with linear kernels given a relatively small amount of training data, especially compared to approaches based on HOG features [39]. The CNN features in combination with observations of multiple view-points being correctly referenced by our SLAM algorithm enables the correct labeling of more than 97% of the grid cells.

6.7. Chapter Conclusions

We introduced a solution for semantic mapping with application to logistic environments. For this purpose we utilized the object recognition system described in Chapter 5.

We presented how points of object observations can be integrated in a graph-based SLAM framework in order to enable online map updates. It is further demonstrated how this graph can be transfered to a grid map with each cell being assigned an object label. Efficient inference of object proposals is carried out by analyzing observation uncertainties. The overall mapping system is evaluated in a warehouse with common object classes being relevant for this type of environment.

Our system runs at about $4 - 5\,Hz$ which is sufficient for initial mapping with AGVs. Thanks to methods of deep learning, we are able to achieve a high object classification accuracy. The efficient object segmentation and sparse pose graph optimization being incorporated enable high performance which is important for robotic applications. We

expect that the presented system contributes to an emerging interest in AGVs for logistic environments while simultaneously motivating the incorporation of the presented algorithms for other robotic applications. By bridging the gap from existing technologies enabling autonomous navigation and the significant initial expense of manual map annotations, we look forward to the upcoming fourth industrial revolution. In the following chapter we will motivate the benefit of semantic maps for the navigation of mobile robots.

Chapter 7.

Map-based Localization in Dynamic Environments using Semantic Perception[1]

[1] The content of this chapter has already been published in [79]

7.1. Motivation

Mobile robots require a robust estimate about their pose with respect to a prior map to provide services and to ensure safe navigation. Occupancy grid maps are commonly used since they provide valuable input not only for localization, but also for path planning and obstacle avoidance. A number of requirements for all of these modules have to be met in order to obtain a robust system while avoiding to maintain different maps for each of them. Special attention is required in changing environments. For localization, it is necessary to find sufficient correspondences among the prior map and the current observation. For global path planning, it is beneficial to avoid all known obstacles, while local planning layers can account for minor changes or dynamic objects. It appears rather important to ensure that the global topology in terms of traversability is not violated.

Existing systems face the above mentioned requirements as follows, they:

1. assume a static environment or

2. continuously map the environment with the latest state being preserved for future tasks or

3. have to observe their working space over longer times in order to identify and predict systematic changes

It can be shown that implementations based on (1) are not able to provide robust localization results in changing environments [114]. Depending on the kind and extent of change, the pose estimation can slightly drift or completely diverge from the true pose. Systems based on category (2) are likely to risk the above mentioned requirement of global traversability of path planning. If, for instance, a large obstacle is placed inside a hallway, which is observed by a mobile robot, this subspace in the map will be labeled *occupied* based on a single observation. This, in turn, might entail that parts of the map become inaccessible from the path planning's perspective until the update will be reverted based on novel observations. This event can be noticed in numerous robotic applications, e.g. pallets and parked vehicles in warehouse hallways or humans in populated public spaces. Without further evidence about the nature of change, the systems based on (2) might ensure localization performance but potentially generate unnavigatable maps. The observation and incorporation of changes in map subspaces over time is investigated by approaches based on the category (3). These were demonstrated to achieve promising results in localization while still maintaining valid maps for navigation. The underlying models, however, typically require a quantitative number of observations for achieving relevant information about map subspaces that can be utilized for predictions and uncertainty estimation. Due to the missing domain knowledge, the process can become of substantive complexity when analyzing individual grid cells of large maps while raising the crucial question whether robotic navigation can actually benefit from all this information. This becomes particularly apparent when modeling the free space of large corridors

being frequently passed by humans or robots. Visited cells of this space constantly undergo state transitions. How is this distinguished from transitions caused by semi-static objects as, for example, pallets at reloading points or parked vehicles?

Contrary to the systems based on (1) - (3), we introduce Semantic Monte Carlo Localization (SMCL) which augments existing methods by object recognition to enable robust localization while maintaining valid maps for navigation in changing environments. Our approach copes without the need of long-term observations or continuous mapping. The key idea is to recognize objects being commonly present in the target environment. Our approach utilizes semantic maps being generated once based on the mentioned object classes. Our measurements are originated from a RGB-D camera providing depth and color data from which object classes can be inferred. The probabilistic association of places in the map and observations is supported by this information. Measurements are incorporated according to the expected contribution of the associated object class which can be determined a-priori.

Our approach is experimentally evaluated on an automated guided vehicle (AGV) in a warehouse environment which is subject to frequent changes. This application requires reliable localization over periods of time. Full autonomy is required once the AGV has passed a teach-in drive with the support of a human supervisor. We demonstrate that SMCL is able to robustly estimate the AGV's global pose throughout all experiments. We expect that our approach is valuable for service robots in environment types being subject to changes whose origins can be determined a-priori: e.g. humans in populated environments, palleted goods in warehouses or cars on streets.

The key contributions of this chapter can be summarized as follows:

- Introduction of an algorithm enabling robust localization in changing environments using semantic perception
- No continuous observation or mapping of environment is required
- Consumer-grade RGB-D cameras are used instead of expensive laser range finders

The chapter is organized as follows. First we will give an overview of the related work in the area of long-term localization. Section 7.3 demonstrates how the localization algorithm is embedded in our overall system. In Section 7.4 we will explain how semantic information can be integrated into Monte-Carlo localization. We will present our experimental results and discuss the key benefits before concluding this chapter.

7.2. Related Work

This sections provides an overview of work which is related to localization in dynamic environments. This mainly addresses the research field of map-based localization using cameras and range sensors, however, for the sake of completeness, we further include relevant work in live-long SLAM for updating maps of changing environments.

Map-based Localization using Vision Sensors

The estimation of the camera pose with respect to a prior map has been extensively studied in the computer vision literature [85, 144]. The core processing sequence is comparable for most of the approaches. First, a 3D model of the environment is built based on large sets of images using structure from motion [69]. Points being observed in multiple camera frames are back-projected based on triangulations. During localization the algorithms have to establish 2D-3D correspondences which is done using descriptors for local features (e.g. SIFT [110]). These are sensitive to illumination changes as we have already discussed in Chapter 3. The state of the art in this field uses different methods for retrieving 3D map points based on camera images. Sattler et al. implements an efficient voting scheme in a graph of co-visible points [145]. Agarwal et al. propose to estimate a rigid transform of 3D points being tracked over multiple monocular camera frames and 3D map points obtained from triangulated Google Street View images to localize a camera [3].

Similarly Carlevaris-Bianco et al. investigate camera-based vehicle localization in urban environments [29]. Their approach extracts SIFT features from an omni-directional camera while the 3D coordinates of these are estimated based on data of a 3D laser range finder. Similarly to [85, 144, 145], the authors establish 2D-3D correspondences for a subsequent localization. In our prior work on visual self-localization in urban environments we omit the triangulation of feature points [76]. Feature points of camera images being associated with places on a map are matched with the current observation. In order to distinguish close-by places we infer the relative depth values of features by means of optical flow. In this way, we can topologically localize the vehicle with respect to a prior map using particle filters. The granularity of the estimated pose can be adjusted by the chosen distances between reference places of the map.

The aforementioned approaches share the assumption that illumination conditions do not substantially change. As already discussed in Chapter 3, this is crucial for the long-term autonomy of mobile robots in real-world applications. Brubaker et al. present a technique omitting the need of 2D-3D correspondences for visual localization [23]. The authors propose to use visual odometry to estimate the path traveled by the vehicle which is subsequently matched to a road network being provided by means of publicly available OpenStreetMap data. This approach is relatively insensitive to illumination changes as long as the present light is sufficient for visual odometry. Experimental results are reported for on-road driving which best suits the algorithm due to restricted driving paths. It cannot necessarily be transferred to other applications, for example indoors, without predefined lanes.

In [28, 33], the authors propose to generate map databases collecting feature descriptors of varying environmental conditions. During localization it efficiently searches for the best matching descriptor set. The authors report exhaustive experimental results proofing that this concept is suitable for visual self-localization.

While there are numerous SLAM frameworks available using RGB-D cameras (see Chapter 4), the number of approaches implementing map-based localization is rather

limited. Fallon et al. simulate RGB-D frames in a virtual environment model which are used in conjunction with particle filters to localize an RGB-D camera [49]. Also Groß et al. utilize particle filters for the visual self-localization of a service robot in a dynamic, highly symmetric home store environment using images captured with an omni-directional camera [66].

Map-based Localization using Range Sensors

The most common method for map-based localization using range sensors is given by the Monte Carlo localization (*mcl*), often implemented and referred to as Adaptive Monte Carlo Localization (*amcl*) [160]. The majority of work in this research field are extensions of *amcl*. The early work of Thrun et al. suggests to explicitly detect and model short readings being a frequent source of association problems due to dynamic objects such as humans [160]. Thanks to this extension, the authors reported on a more reliable pose estimation for an automated tour guide robot in a museum [25]. Meyer-Delius et al. presented the idea of temporal maps being able to capture dynamic changes over multiple time frames through agglomerated incorporation of temporary local maps and a global map [113]. With the experience gained from this idea, Tipaldi et al. further suggested the use of Hidden Markov Models (HMM) to model state transitions of individual grid cells of a global map [163]. Their system learns state transitions «*free* $\rightarrow$ *occupied*» and vice versa. A Rao-Blackwellized particle filter (RBPF) is used to infer the robot pose in this map representation. Saarinen et al. presented the concept of independent markov chains which, similarly to [163], are learnt for each grid cell [141]. The stochastic process counts events of state transitions and based on that allows to predict probabilities of dynamic changes. In [142] the authors further demonstrated the robustness in dynamic environments based on a MCL variant utilizing a Normal Distribution Transform (NDT). More recently, Krajnik et al. proposed Frequency Map Enhancement (*fremen*) which builds probabilistic functions of time to model environments [97]. A mixture of *amcl* and the SLAM algorithm *gmapping* [64] is used to continuously build spatio-temporal maps and enable global localization with respect to a prior map. Based on sets of observations made at varying times and timescales, this representation learns to predict putative changes and frequently incorporates recent information. In order to ensure robust state estimates over longer periods of time, the authors suggest to regularly re-initialize the global localization (*amcl)* at a known place (e.g. a charging station).

Life-long SLAM

The application of SLAM for long-term robot pose estimation has been investigated by several researchers. Labbe et al. [102] and Hartmann et al. [70] proposed solutions for SLAM with appearance-based loop closure detection which are shown to be capable of multi-session mapping. Einhorn et al. utilize the normal distribution transform (NDT) to infer loop closure candidates and estimate transformations of graph nodes [46]. Either of the authors re-localize a mobile robot in a SLAM graph [46, 70, 102] while constantly

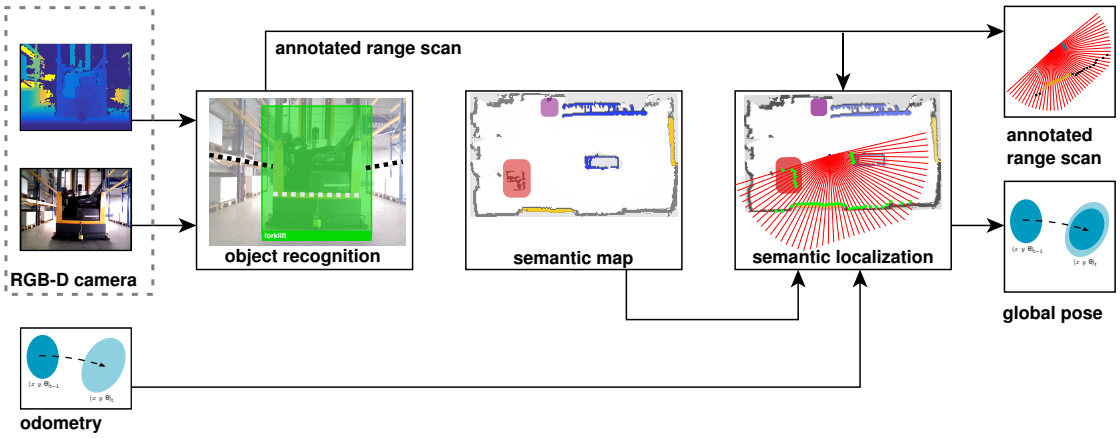


Figure 7.1.: The graph provides an overview of our semantic perception framework and visualizes how the localization is integrated. The input RGB-D sensor data is passed to the object recognition (see Chapter 5). This module generates a 2D projection of the 3D point cloud with measuring points being at either a specified height for unknown objects or at object-specific height references, e.g. the first horizontal pillar of a rack. This range scan is supplemented by object class labels for each beam. SMCL evaluates the annotated range scan and estimates the vehicle's pose with respect to the prior semantic map.

incorporating new observations in the map. Carlevaris-Bianco and Eustice further propose to reduce the amount of nodes in a graph to ensure constant runtime over longer periods of time. Also, Einhorn et al. uses vertex pruning to avoid an unlimited increase of the graph size.

Summary

Algorithms for map-based localization incorporate changes based on continuous observations of the environment. The systems presented in [97] and [124] are able to predict changes once the environment has been sufficiently observed. The robustness in [97] is achieved based on frequent re-initializations and repeated mapping. None of the related approaches incorporates semantic perception to enable robust localization in the presence of significant changes right after building an initial map. Existing SLAM-based systems keep running all the time while constantly overriding subspaces of the map. Without appropriate semantic perception this is undesirable for a number of applications since maps can rapidly become unnavigable.

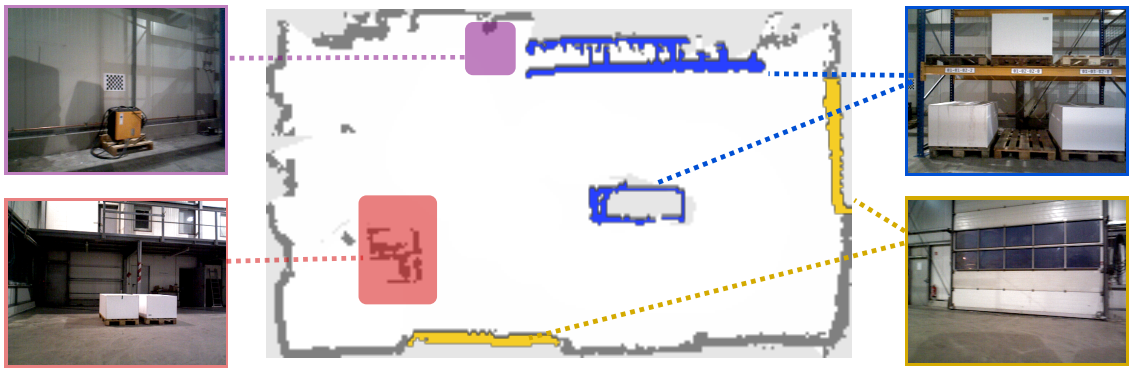


Figure 7.2.: The semantic layer is plotted on top of an occupancy grid map. Grid cells are supplemented by object class labels. *Gates* and *racks* are identified based on our object recognition system. *Reloading areas* (red) are inferred from multiple pallet observations outside of racks. *Charging stations* (purple) are tagged with a checkerboard sign and automatically mapped.

7.3. System Overview

The semantic perception framework is a key element of our system. It consists of the components object recognition (see Chapter 5), semantic mapping (see Chapter 6) and semantic localization. The Fig.7.1 illustrates the structure of the entire system highlighting also the parts of object recognition and semantic mapping. This demonstrates their positions within the overall system and its connections to other components.

7.4. Localization

Given a semantic map m, a particle filter is used in order to perform global localization. The robot's state at a discrete time step t is expressed by its pose $x_t = (x, y, \theta)$ containing the 2D location (x, y) and orientation θ. The state x_t is predicted based on odometric readings which are incorporated by the motion model. The observation model evaluates sensor measurements z_t obtained from the RGB-D sensor with respect to the prior map.

7.4.1. Motion Model

The motion $u_t = (v_t, \omega_t)^T$ carried out by the robot is modeled by its translational velocity v_t and rotational velocity ω_t. The odometry is subject to noise which can occur due to a variety of sources, for instance: wheel slip, varying inflation pressures of tyres or load

balance of the vehicle. In order to incorporate this uncertainty in our motion model, we add zero mean Gaussian noise ϵ_b with standard deviation b as detailed in [160]:

$$\begin{pmatrix} \hat{v}' \\ \hat{\omega}' \end{pmatrix} = \begin{pmatrix} v + \epsilon_{\alpha_1|v|+\alpha_2|\omega|} \\ \omega + \epsilon_{\alpha_3|v|+\alpha_4|\omega|} \end{pmatrix} \tag{7.1}$$

The values $\alpha_1, ..., \alpha_4$ refer to vehicle specific parameters which can be set according to the accuracy of the utilized wheel encoders. The state transition from the previous state x_{t-1} to the predicted state x'_t after Δt units of time is defined by the following motion model:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + \hat{v}\Delta t \cos(\theta + \hat{\omega}\Delta t) \\ y + \hat{v}\Delta t \sin(\theta + \hat{\omega}\Delta t) \\ \theta + \hat{\omega}\Delta t \end{pmatrix} \tag{7.2}$$

7.4.2. Observation Model

The observation model utilizes the latest range sensor measurement to evaluate each particle x_t^k. Given the map m and the observation z_t, we estimate the observation likelihood $p\left(z_t \,|\, x_t^k, m\right)$ for the k-th particle.

The key contribution of our work affects the observation model. Typically it consists of a range model projecting endpoints of sensor beams in the map coordinate frame originated at the particle's state x_t^k. The occupancy value of the hit grid cell m_i is evaluated and the distance r_{m_i} is estimated. Our approach additionally considers object class labels of semantic grid maps to evaluate correspondences of range measurement points and grid cells. We will further refer to the sensor observation of an individual beam l at time t as $z_t^l = \{r, o\}_t^l$ with r being the measured range and o the estimated object class for the l-th beam.

Range model

This model evaluates the particle set according to the differences of the measured range r_t^k and the range projected from the particle's pose to the map grid cell m_i being hit by r_t^k. The closest cell m_i is found using nearest neighbour search as exhaustively described in [160]. Given the ranges r_t^k and r_{m_i}, the likelihood p_r of measuring r_t^l given m and particle state x_t^k can be estimated as:

$$p_r\left(r_t^l \,|\, x_t^k, m\right) = \exp\left(-\frac{|r_t^l - r_{m_i}|}{2\sigma_r^2}\right) \tag{7.3}$$

with σ_r^2 describing the expected measurement uncertainty.

This range model is the core of the likelihood field sensor model which is also used in AMCL providing the basis to compute a particle's weight.

Object model

The semantic perception is integrated inside the object part of the observation model. It incorporates class-specific observation uncertainties which can occur due to objects of similar visual appearance or geometric properties. The object correspondence matrix describes this a-priori knowledge which expresses the probability of measuring a certain object class given the object class of the cell o_{m_i}:

$$p_o\left(o_t^l \equiv o_{m_i}\right) = p\left(o_t^l \,|\, o_{m_i}\right) \tag{7.4}$$

The object correspondence matrix does not have to be symmetric, hence $O_{ij} \neq O_{ji}$ is a valid constraint. This enables to specifically account for single-sided observation uncertainties. The matrix used for our system is given by Table 7.1.

o_{m_i}	**Rack**	**Pallet**	**Gate**	**Forklift**	**Human**	**Unknown**	o_t^l
	0.99	ϵ_{min}	0.5	ϵ_{min}	ϵ_{min}	0.5	**Rack**
	–	–	–	–	–	–	**Pallet**
	0.5	ϵ_{min}	0.99	ϵ_{min}	ϵ_{min}	0.5	**Gate**
	–	–	–	–	–	–	**Forklift**
	–	–	–	–	–	–	**Human**
	0.5	ϵ_{min}	0.5	ϵ_{min}	ϵ_{min}	0.99	**Unknown**

Table 7.1.: The object correspondence matrix describes the a-priori probabilities of measuring certain object classes. As already mentioned, the dynamic classes *pallet, forklift* and *human* are not stored in the map. Hence these observations do not contribute to the pose estimate. The value ϵ_{min} is a small nonzero value utilized in order to avoid numerical issues.

Model fusion

The presented sub-models for range and object perception are probabilistically fused within a common sensor model. The individual components of this mixture are weighted in order to account for sensor and environment characteristics. For instance, the weights of the range model can be adjusted according to the measurement accuracy of the depth sensor. The weight for the object model can be set with respect to the classification accuracy or overall uncertainty being expected.

The mathematical derivation for the mixture model p given the measurement z_t^k can be expressed as follows:

$$p(z_t^l \,|\, x_t^k, m) = z_r \cdot p_r + z_o \cdot p_o \tag{7.5}$$

with z_r and z_o denoting the prior weighting factors for the individual components. All measurements z_t at time t are incorporated according to:

$$p(z_t \mid x_t^k, m) = \prod_l p(z_t^l \mid x_t^k, m) \tag{7.6}$$

Summary

The presented semantic sensor model provides an extension to the generic likelihood field model. More precisely, the association of putative correspondences of projected measurement endpoints and map grid cells is established using the latter. Instead of purely evaluating particles based on the estimated spatial displacements, the semantic model considers additional information that can be observed using RGB-D cameras. Similarly to the generic likelihood field, the pose estimate will be more accurate and robust the more correspondences are found. The key difference, however, is that the semantic sensor model significantly mitigates or even disables the contribution of false correspondences which is not the case for the generic sensor model. This literally matches any observed obstacle to the closest one in the map which may result in significant pose deviations. For estimating the pose w.r.t. to the prior map it is important to use stable correspondences. Points inside a subspace of the map being subject to frequent changes, which are not necessarily predictable, rather confuse the localization algorithm than contributing to robustness or accuracy.

7.5. Experiments

We evaluate the presented approach for semantic localization by experiments carried out in a warehouse which consists of a multitude of common objects expected for this type of environment. The data is collected using a reach truck which has been fully automated for a research project. For the purpose of semantic mapping and localization, we manually steered the vehicle inside the warehouse. The RGB-D data utilized by our system is recorded using two Asus Xtion cameras with one being aligned forwards and one sidewards with respect to the vehicle's direction of travel.

7.5.1. Parameters

Our algorithm requires different parameters being described in the previous sections. For the sack of completeness, the choice of parameter values used in our experiments is provided in Table 7.2.

Component	Description	Variable	Value
particle filter	motion model	α_1, ..., α_4	$[1.0, 1.0, 1.0, 1.0]\ m$
	range model std	σ_r	$0.2m$
	range model weight	z_r	0.5
	object model weight	z_o	0.5
map	occupancy threshold	p_{occ}	0.7
	free threshold	p_{free}	0.3
classification	distance threshold	τ_{svm}	0.4

Table 7.2.: Overview of all relevant parameters and their values used in our experiments.

7.5.2. Datasets

A number of datasets were recorded given the described setup. This set consists of different experiments focusing on particular problems the state of the art algorithm AMCL faces in changing environments. We will shortly describe each of the datasets in the following.

Static

This dataset was captured subsequently to the mapping process with no objects being moved in order to estimate the baseline localization accuracy of the presented approach. It serves as reference for further experiments.

Gate

The vehicle is steered towards an open sectional gate. During the mapping process this gate was closed and consequently marked *occupied* in the occupancy grid map. Within this experiment we place three pallets below the open gate area. The boundaries of the pallets are $0.65m$ apart from the actual gate and hence are outside of the recorded map area. This demonstrates a common highly dynamic area inside a warehouse with the gate being frequently opened and closed and storage goods being moved.

Rack

The reack truck is driven through a hallway surrounded by high-level racks. The pallets of the lower sections are slightly moved about $0.1m$ towards the hallway. This experiment investigates the impact of changing content inside racks on the localization accuracy. This effect can be observed due to accumulated errors occurring during automated reloading processes or more obviously due to pallets being stocked by humans in environments with both automated systems and human operators.

Reloading

Particularly larger free space areas of warehouses are subject to changes due to palleted goods being frequently moved. We recorded a trajectory which passes a typical reloading area of a warehouse in order to analyze the algorithm's robustness in the presence of significant changes.

Mixed

This dataset investigates the algorithm's performance over longer time covering a trajectory length of about $269m$ and a period of time of about $28min$. The environment is constantly reconfigured during this experiment. This mainly addresses pallets that are moved inside racks, around reloading areas and gates.

7.5.3. Ground truth

The ground truth trajectory for each dataset is estimated using a SICK S-300 laser range finder. Thanks to the large field of view ($270°$), long range ($30m$) and sub-centimeter accuracy of the sensor, we are able to provide high-resolution ground truth data. Since also a laser-based localization system (e.g. based on AMCL) would be affected by the environment changes being investigated, a SLAM algorithm is used to generate individual maps and estimate the traversed paths. The underlying graph-based SLAM framework which makes use of a joint map and pose estimation is exhaustively described in Chapter 4. In order to align the trajectories to a common global frame, a checkerboard with known dimensions is placed on top of a charging station. A checkerboard detection system is utilized to estimate the initial pose at the charging station. This system further serves as additional, highly accurate, loop closure detection for aligning the ground truth trajectory. Note that this system is not used in any way by the localization algorithms being evaluated here, but solely for generating the ground truth. The accuracy of the ground truth is below $0.05m$ (see also Chapter 4).

7.5.4. Results

Figure 7.3 illustrates the results obtained for SMCL and AMCL respectively.

Static

Both, AMCL and SMCL provide robust and accurate results. The position accuracy is at about $0.21m$ in the mean. This confirms the common expectation that AMCL performs well. The differences of SMCL and AMCL are minor.

Rack

The results of AMCL are adequate for minor changes inside the racks. However, once a significant amount of storage content is moved, the localization error of AMCL increases. Since AMCL does not distinguish particular objects, it estimates the vehicle's pose w.r.t. the rack's changing content which can be palleted goods or the static pillars. SMCL, in contrast, correctly identifies the high-level racks and relies on the first horizontal pillar for the localization which ensures robust and accurate results within warehouse hallways. The error of SMCL remains constant to its baseline whereas the error of AMCL increases by about $0.1m$ which is related to the amount and degree of the change inside the racks.

Reloading

It can be observed that AMCL fails once the vehicle enters the spacious reloading area in front of the gate. The pose estimate significantly deviates due to a notable amount of change being present in this area. The error of AMCL increases to about $2.38m$. The particle filter of AMCL diverges in this experiment and does not recover without re-initialization. SMCL correctly identifies pallets in this area and robustly tracks the vehicle's pose.

Gate

As expected, AMCL literally pushes the vehicle towards the occupied map area being actually covered by the gate. SMCL correctly recognizes that the perceived objects are pallets instead of parts of the gate and hence its pose estimate does not deviate. AMCL deviates by about $0.83m$ at the gate area and continues with an increased error. AMCL's lack of semantic perception potentially attracts the vehicle to move outside the warehouse. This might be a danger for persons working in front of warehouses if a vehicle is not expected to leave the warehouse. Safety-related facilities are not necessarily available outside. Moreover, the currently available RGB-D sensors do not work in the presence of significant amount of sunlight.

Mixed

AMCL frequently deviates during this experiment up to a maximum error of about $1.47m$. SMCL performs with a better mean error of about $0.25m$. In some situations during this experiment SMCL deviates up to a maximum error of about $0.76m$. We expect that this has happened during continuous loops in the reconfigured reloading area. Due to the limited perception field, SMCL was unable to observe sufficient stable points and hence increasingly relies on odometric measurements. Even though SMCL's pose estimate becomes more inaccurate, it can be observed that the uncertainty increases as well with the covariance still covering the true pose. This provides the ability to reduce the accumulated error once the vehicle enters an area with more stable reference points being present.

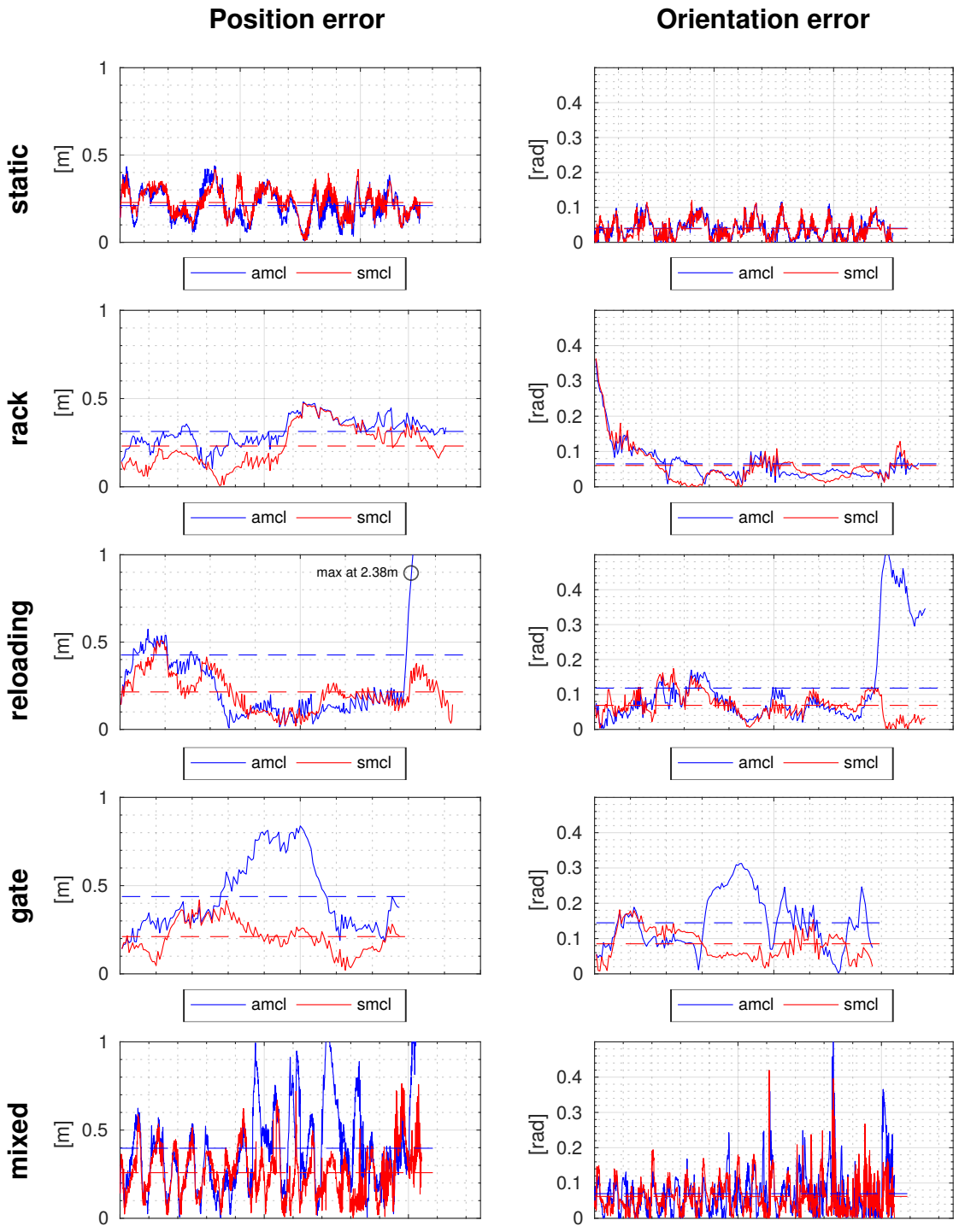


Figure 7.3.: Experimental results obtained for the individual datasets *static*, *rack*, *reloading*, *gate* and *mixed* (one for each row). The left column shows the position error, the right column the orientation error. The dashed lines show the mean error on the individual datasets for either algorithms. The baseline accuracy of SMCL and AMCL is at about $0.21m$. Due to significant changes in the environments, such as the closing state of a gate, the relocations of pallets and parked vehicles, the error of AMCL increases enormously (see *gate*) or AMCL diverges (see *reloading*). SMCL, in constrast, is able to robustly estimate the vehicle's pose even in the presence of high dynamic changes.

If the system is aware of its uncertainty in the global pose estimate, it is able to adjust its behavior, for example by reducing its speed, turning on its warning beacon or requesting a human supervisor. These actions cannot be enabled if the system is not aware of the uncertainty. This can be observed for AMCL: the larger pose error does not necessarily imply an increased uncertainty estimate. This is due to the fact that AMCL matches measurements of dynamic objects to the map not taking into account different object classes and their properties.

7.5.5. Discussion

Either of the algorithms achieve a baseline accuracy of about 0.21m. This differs from those results that can be obtained using laser range finders (LRFs) due to the following reasons. Firstly, the measurement accuracy of an LRF is significantly higher over the entire operating range compared to RGB-D sensors, particularly in the case of Kinect-like cameras as being used in our experiments. Secondly, the operating range and field-of-view of the LRF is substantially larger, e.g. (30m; 270deg) for a SICK S-300 vs. (5m; 58deg) for a ASUS Xtion camera. This difference becomes particularly apparent in areas of larger free space where RGB-D cameras can hardly perceive any obstacle whereas LRFs can typically observe a large area over a long time. The limited sensing properties of RGB-D sensors also cause an increased uncertainty and hence limit the accuracy of the maps even when using customized SLAM methods as detailed in Chapter 4. Nevertheless, the use of these cameras enables a number of benefits, such as reduced costs and a high information density for object recognition. The latter provides a substantial requirement for the semantic perception layer of SMCL.

AMCL performs well as long as sufficient static obstacles are present and observable (e.g. large walls), otherwise it fails or becomes inaccurate. The error of AMCL in the gate and reloading area (up to $2.38m$) is not acceptable for automated systems. Drive requests might not be fulfilled since targets are failed. The consequences of association errors at a gate can be enormous since an automated vehicle can be attracted outside the warehouse into an unmapped area which might be inaccessible for itself and not expected by humans residing in this area. The increased position errors of AMCL close to the racks (about 0.32m) might be acceptable if an additional system can be utilized for actual reloading processes, e.g. by measuring pallet poses as targets and apply reactive approaching strategies. However, the increased uncertainty in combination with the lack of semantic perception can cause more significant errors or filter outages, if more dynamic objects such as other AGVs or unexpected pallets are present. Pose estimates of SMCL also deviate in the presence of significant changes, but orders of magnitude less than AMCL does. The uncertainty of SMCL increases but still covers the true pose which enables SMCL to correct its pose and minimize its uncertainty once sufficient correspondences are found.

7.6. Chapter Conclusions

Semantic maps provide a number of substantial benefits for mobile robots. This chapter presents an approach utilizing these as prior for the map-based localization of an AGV. The most commonly used algorithm AMCL provides accurate results and enables global localization. However, the underlying sensor model hampers its use in dynamic environments being subject to numerous changes over time. The goal here is to incorporate the semantic information of the map inside the core of the localization. The association of map and sensor data is augmented by additional knowledge about known objects classes and their predicted properties in terms of dynamic changes and contribution to the global pose estimate. Our approach requires only limited additional overhead for modeling changes of the environment within the map representation since these are not constantly incorporated into spatio-temporal models as it is the case for the majority of related algorithms. Our approach demonstrates that these expensive models are not necessarily required if prior knowledge about commonly observed objects is available. The semantic perception layer is not exclusively designated for the localization system. A multitude of other components benefit from this as well, e.g. the human-robot-interaction, mapping of storage places or tailored obstacle avoidance.

The accuracy and robustness of SMCL was demonstrated to outperform the state of the art in map-based localization. We expect the outcome of our novel approach to be highly beneficial for the emerging application of AGVs in logistic environments and motivate the utilization of semantic mapping and localization for other robotic applications. The key advantage, that is the limited number of commonly observed object classes, is feasible for many other scenarios. Detecting cars, buses and pedestrians will likely enable higher localization accuracy for on-road driving. Likewise the navigation of mobile robots in shopping malls, stations and airports can benefit from the recognition and consideration of humans, carts and suitcases.

Chapter 8.

Conclusion

Efficient and robust algorithms for localization and mapping are a fundamental requirement for autonomous mobile robots. While substantial contributions to SLAM, map-based localization and place recognition have been made in the recent years, a number of open problems remain. This addresses the scalability of algorithms for increasing environment sizes and dynamic changes. For a multitude of applications it cannot be assumed that the state of the environment being initially captured will remain unchanged during the operation time of a robot. This condition has to be met in order to guarantee robust navigation over longer periods of time.

This thesis presents algorithms and software frameworks investigating the aforementioned challenges. We literally describe the path from the state of the art in localization and mapping towards improved generic methods for large-scale environments and finally towards models incorporating environment-specific knowledge. In the beginning we determined the key questions aimed to be answered by this thesis. We will refer to these in the following.

The first contribution of this thesis is given by the two new algorithms for place recognition, GLARE and GRAPE. In contrast to the majority of the state of the art, we do solely utilize 2D range data rather than camera images for this purpose. Also, we do not adopt the common technique of generating appearance descriptors of interest points. A novel concept modeling the spatial relations of co-occurring landmarks has been introduced. This allows to achieve high precision-recall performance while simultaneously reducing the run time. Thanks to specifically tailored data structures and retrieval algorithms, GLARE is able to detect loop closures at a high frequency even at the scale of a suburb which we demonstrate in our experiments. It was shown that our approach outperforms the state of the art in both, precision-recall and run time. We further introduce GRAPE, an extension of GLARE, which builds dense descriptors of places by the use of surface primitives. It renders place recognition also possible in environments with limited descriptive power in terms of curvature extrema. GRAPE performs similarly to GLARE outdoors and achieves even better results indoors. Our algorithms GLARE and GRAPE provide a substantive base for large-scale place recognition using 2D range data solving the key challenge reported by **Q 1**.

The second contribution is a SLAM framework which can be used with several range measuring sensors. There exist numerous SLAM algorithms being designed for the exclusive use of laser range finders or RGB-D cameras. The former often provide only limited scalability for large-scale environments due the limited performance in the detection of loop closures. Camera-based approaches can scale better thanks to the higher information density for place recognition. However, they are more sensitive to changing illumination conditions and perspective variances. We are bridging the gap by providing a generic framework utilizing 2D range data which can, for example, be originated from either laser range finders or RGB-D cameras. Our approach unites efficient algorithms for online SLAM and performs a subsequent optimization enabling to increase the accuracy of the map which provides answer to question **Q 2**. We implement a restricted search in order to account for loop closure detections in the presence of run time limitations by automatically adapting to the uncertainty of the current pose estimate. The integration

of GRAPE for this purpose allows to identify a large number of loop closures. Repetitive structures in the environment naturally entail uncertainty in the loop closure detection. Our framework minimizes this by using a restricted loop closure search. All remaining association errors are handled by robust factor graph representation being able to mitigate the impact of outliers. This feature of our SLAM framework solves the key challenge of question **Q 3**. Within exhaustive experimental evaluations we demonstrate the robustness, accuracy and scalability of our SLAM framework using range data obtained from laser scanners and depth cameras.

Our third contribution deals with object recognition in the context of mobile robotics. While object recognition in 3D is computationally expensive in general, we are able to achieve high performance by exploiting constraints about the environment. In particular, we assume that the world can be described by objects in 2D space having a specific height over ground. This allows us to efficiently segment the range data by means of contour processing with object boundaries being identified as curvature extrema which answers our key question **Q 5**. Geometric descriptors capturing height and width data of objects are generated based on the detected segments. These are classified given the geometric and textural features. The latter are extracted from the RGB image using a convolutional neural network. The use of RGB-D cameras enables us to achieve better results compared to solely using 2D image data since the additional range data can be used to obtain more invariance in regards of lightning changes and perspective transformations. Thanks to the limited object diversity being present in the investigated environment and the beneficial properties of deep learned features we are able to obtain an outstanding recognition performance which is demonstrated by experimental evaluations.

Our fourth contribution unites the SLAM framework and the object recognition system to generate semantic maps. The range measurements are therefore extended by object labels and constantly updated along with the pose graph optimization. The object observations are potentially uncertain since, for instance, objects are not necessarily visible from different viewpoints. We account for this uncertainty by the use of efficient inference methods optimizing the object label estimates. As a result we obtain a pose graph with associated object points which can subsequently be transferred to a semantically annotated occupancy grid map. We omit extensive manual labeling of maps by human supervisors. The additional information being gained provides beneficial knowledge for robotic navigation tasks.

Our final contribution is dedicated to the challenging problem of long-term localization in changing environments. While commonly used algorithms provide accurate results in static environments, they tend to fail in the presence of dynamic changes. This addresses particularly stationary objects that change their positions over time entailing structural modifications of the environment state. The impact of this is intensified if the utilized sensor possesses a limited field of view and operating range as it is the case for RGB-D cameras. Our approach SMCL uses semantic maps and object recognition to robustly determine the vehicle's pose. Instead of solely projecting scan points into the map and comparing to the nearest obstacle, our observation model is supplemented additional object class knowledge which is taken into consideration. In this way we directly ex-

clude dynamic objects from the absolute pose estimate and mitigate the contribution of observations whose object class labels do not comply. We evaluate the novel model in numerous experiments carried out in a warehouse considering several scenarios that can be commonly expected in this type of environment. As a result it can be clearly shown that SMCL is able to provide robust and accurate results while the commonly used algorithm AMCL systematically fails once the vehicle enters sub-areas of the map being subject to dynamic changes. We have shown that AMCL approaches extents in the pose error which hamper the reliable operation of autonomous robots and the fulfillment of tasks. Thanks to the use of semantic perception and prior knowledge about our environment, SMCL works on-the-fly and omits uncertain map updates caused by frequent environment changes. Our algorithm solves our key question **Q 6** and therefore also provides a valuable application of object recognition (see **Q 4**).

The novel algorithms and software frameworks of this thesis are mainly motivated by the scenarios and experiments presented in the respective chapters. However, we expect that also other applications can benefit from them. For example, the place signatures generated with GLARE and GRAPE could be used to label occupancy grid maps with respect to their room category such as corridors, open-space lobby or office. The features necessary for this classification are already encoded by the place signatures. Our object recognition provides numerous potential applications, particularly for human robot interaction. In chapter 5 we motivate the incorporation of object knowledge for defining individual clearances for obstacle avoidance. This enables more socially acceptable bypassing of humans, increased safety when passing other vehicles by tracking their motions and also accelerated avoiding maneuvers in the presence of static obstacles such as palleted goods. We expect this to be of high relevance as it provides a lot of potential to improve the acceptance and efficiency of autonomous robots. The introduction of semantic Monte-Carlo localization utilizes prior knowledge of commonly expected objects which is incorporated when matching observations scans to the initial map. With our modified observation model we hope to motivate a wider use of RGB-D cameras and additional sensor specific information, as for example height and reflectance. This can support the estimation of the absolute pose in terms of both, more distinctiveness and robustness in the presence of structural changes.

We motivate the benefits of semantic maps for long-term localization. However, it is not limited to this application. In conjunction with automated warehousing they also allow to identify storage bins based on detected high-level racks. Gate areas and reloading points pose valuable inputs for path planning and safety-related modules. This meta information is currently either not available on AGVs or supplemented by means of expensive manual annotations of human supervisors.

The first part of our thesis introduces generic methods enabling robotic mapping and localization independently of the environment type. The semantic models being presented in the second part make additional features available but still rely on the previously presented place recognition and SLAM algorithms. The semantic mapping algorithm, for instance, works on top of our SLAM framework. A more tight fusion of the generic and semantic models provides a lot of potential. Currently our place recognition is solely a

fundamental for the semantic mapping. However, the semantic perception can also return beneficial information to improve place recognition. For instance, it can be utilized for filtering dynamic or semi-static objects when generating GLARE signatures which supports a higher robustness for SLAM in dynamic environments. In our thesis we motivate the application of semantic perception for warehouse environments. However, the novel achievements are not restricted to this and can easily be transferred to other applications. We expect that semantic models have an enormous impact for pushing the state of the art in localization and mapping. Robotic systems can benefit from additional environment-specific knowledge which enables more robust algorithms for long-term autonomous navigation within a multitude of applications ranging from on-road driving to flexible automated warehousing and service robots in shopping malls, hospitals or museums.

This thesis presents the methodological background for recognizing objects, spatial mapping and pose estimation. The achievements provide relevant contributions for the mobile robotics and computer vision communities. We hope to motivate further investigation in incorporating semantic perception and prior knowledge about the environment into existing generic algorithms in order to enable more robust and situation-aware robotic systems.

Appendix A.

Datasets

A.1. FTF-Lab - A Testbed for AGVs in Logistics

The application of mobile robotics in intralogistics has become of particular interest again. Autonomous navigation provides a substantial enhancement for automated guided vehicles enabling more flexibility and performance. For this purpose, a testbed was set up inside a warehouse in Lübeck, Germany.

The FTF-Lab is a small warehouse of about $11\,m\,x\,19\,m$ size with a high ceiling. It consists of two gates, four high level racks, several kinds of palleted goods and a number of doors and windows with medium incidence of daylight. For experiments we make use of an automated reach truck of type Jungheinrich ETV-216. It is equipped with wheel odometry, several 3D cameras and a SICK S300 safety laser scanner (see also Fig. A.2). The odometry is obtained from three wheel encoders with one being placed at the drive wheel and two at the back wheels. An overview of the sensors at the vehicle is given by Table A.1.

(a) AGV model, sensors and AGV coordinte system.

(b) AGV in FTF-Lab.

Figure A.1.: **FTF-Lab**. Fig. (a) illustrates a model of the AGV with the point of origin being centered between the back wheels. The RGB-D cameras are mounted at the front and the right side of the vehicle respectively (red). The laser range finder is mounted at the front of the AGV (blue). The other cameras are not used in the experiments of this thesis. Fig. (b) shows the AGV in the testing warehouse FTF-Lab consisting of high-level racks, gates and different palleted goods.

Sensor	Type	Model	Max. Range	Ang. Res.	HFOV/ VFOV
Laser	2D Lidar	SICK S300 Expert	30 m	0.00882466°	270°/-
Front Camera	Structured light	Asus Xtion Pro Live	5.0 m *	0.00158166°	58°/ 45°
Side Camera	Structured light	Asus Xtion Pro Live	5.0 m *	0.00158166°	58°/ 45°

Table A.1.: This table shows an overview about all utilized sensors and their characteristics. * The official operating range indicated by the manufacturer is 0.8 - 3.5m. However, the sensor still provides measurements beyond 3.5m but with increasing uncertainty.

(a) Front RGB-D camera (red) and laser range finder (blue).

(b) Side RGB-D camera (red).

Figure A.2.: **AGV**. This figure shows the AGV with the sensors used in this thesis being highlighted.

A.2. Public Datasets

The Figures A.3-A.8 provide details about the public datasets being utilized in this thesis.

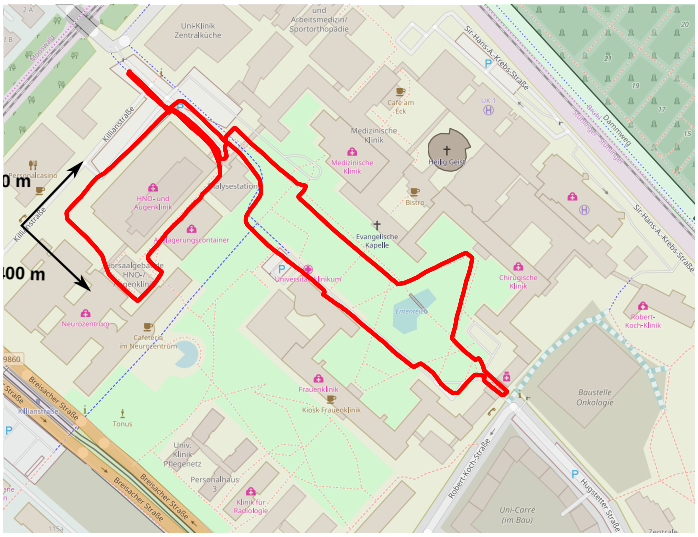


Figure A.3.: **Fr-Clinic**. This dataset was recorded outside the hospital of Freiburg, Germany. The surrounding environment mainly consists of building facades and trees. The utilized range sensor is a SICK LMS laser range finder. More information can be found in [101].

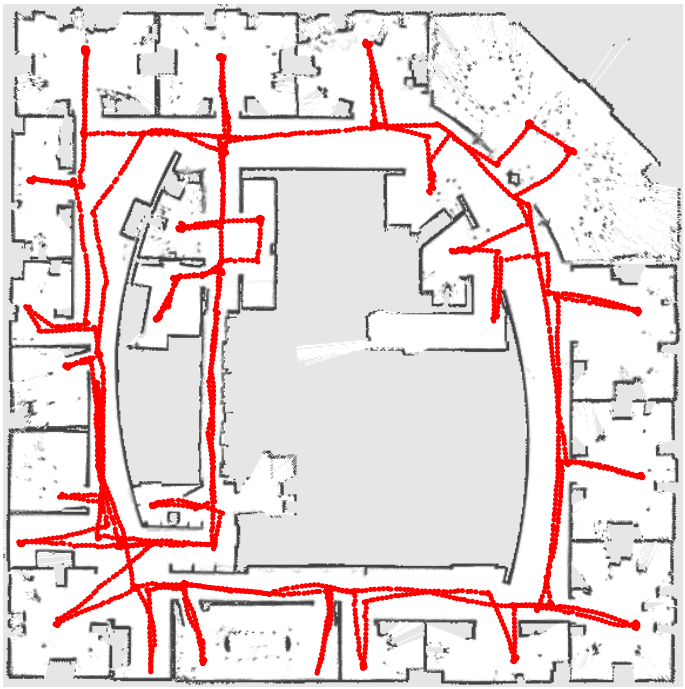
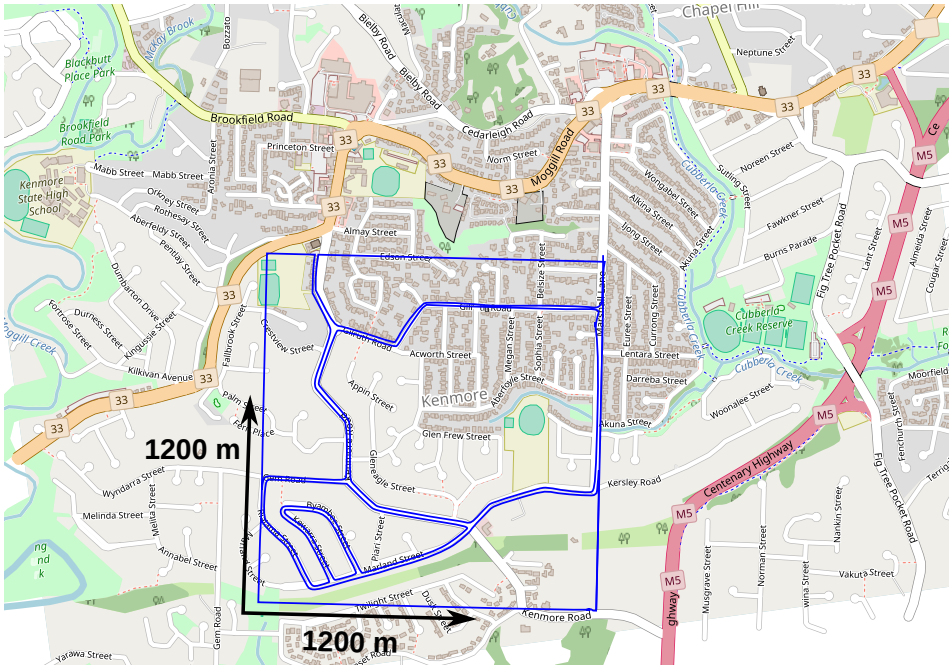


Figure A.4.: **Intel-lab.** This dataset was recorded by Dirk HKähnel inside the Intel research lab in Seattle, WA. It is a typical office environment consisting of small rooms and narrow corridors. The utilized sensor is a SICK laser range finder. The dataset has become part of numerous benchmarks (e.g. [101]).

(a) (b) (c)

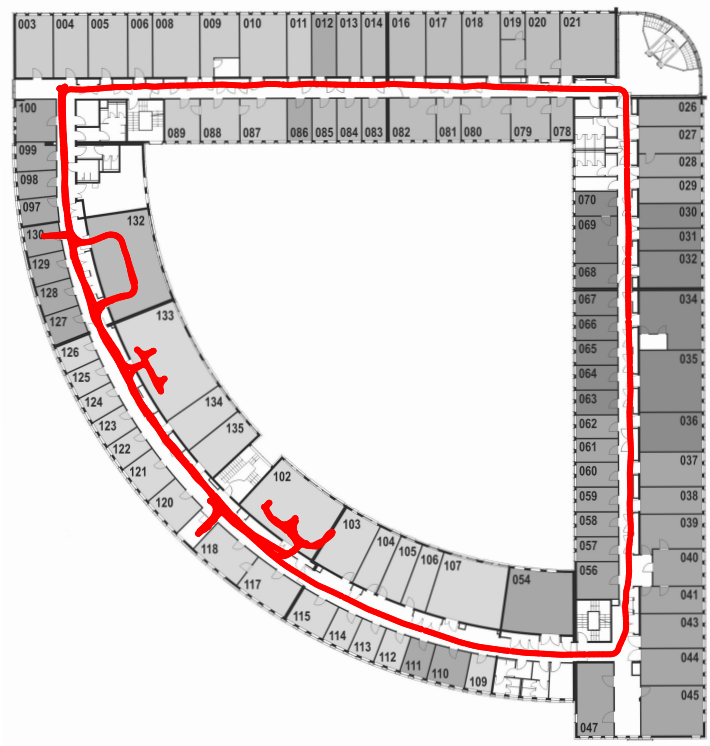


(d)

(e) (f) (g)

Figure A.5.: **Kenmore**. This dataset was recorded with SICK LMS laser range finders from the rooftop of moving a car in Kenmore, a suburb of Brisbane, Australia. Fig. (d) shows a map with the operation environment and traversed streets being highlighted (blue). The dataset was published by Bosse and Zlot [20]. The images shown in Fig. (a)-(c) and (e)-(g) are not part of this dataset, but were also captured in Kenmore which provides an impression of suburb's visual appearance [15].

(a) (b) (c)

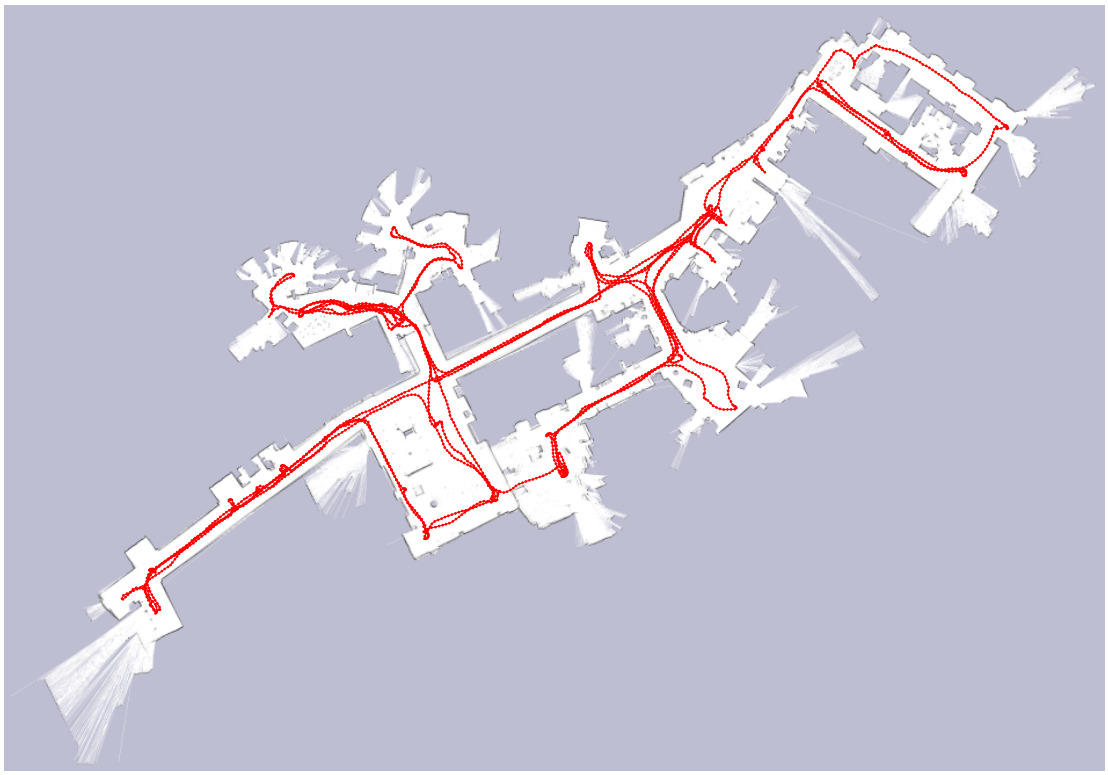


(d)

(e) (f) (g)

Figure A.6.: **ITI**. This dataset was captured on level two of building 64 at the University of Lübeck, Germany. Fig. (d) shows a floor plan with the traversed trajectory being overlaid (red). Fig. (a)-(c) and (e)-(g) illustrate typical scenes of this environment. More details can be found in [56]

(a) (b) (c)

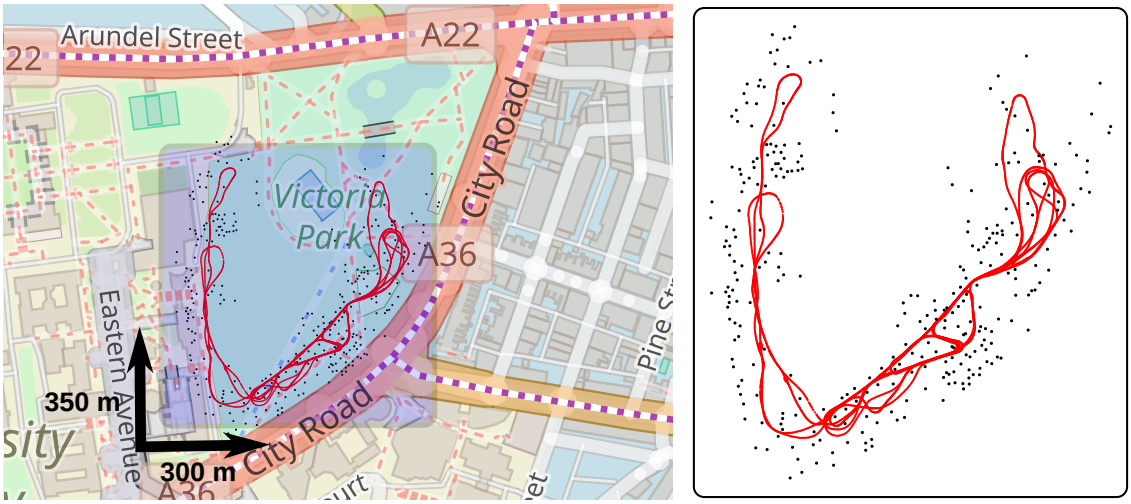


(d)

(e) (f) (g)

Figure A.7.: **Stata**. This dataset was captured inside the MIT Stata Center building. We make use of a subset being collected on level 2 of the building. Fig. (d) shows an occupancy grid map with the traversed trajectory being overlaid (red). Fig. (a)-(c) and (e)-(g) illustrate typical scenes of this office-like environment. More details can be found in [48].

(a) (b)



(c) (d)

(e) (f)

Figure A.8.: **Victoria Park**. This dataset was captured inside the Victoria park next to the campus of the University of Sydney, Australia. Fig. (c) shows an overview map of the surrounding area, Fig. (d) is a detail of this map showing the traversed trajectory (red) and trees serving as landmarks (black circles). Fig. (a)-(b) and (e)-(f) illustrate the visual appearance of the park area. More information can be found in [67].

Bibliography

[1] *Microsoft Bing Search API*. `https://datamarket.azure.com/dataset/bing/search/`. [Online; accessed 8-April-2016].

[2] *Evaluation : from Precision , Recall and F-measure to Roc*. Volume 2, 2011.

[3] Agarwal, Pratik, Wolfram Burgard and Luciano Spinello: *Metric Localization using Google Street View*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2015.

[4] Agarwal, Pratik, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss and Wolfram Burgard: *Robust map optimization using dynamic covariance scaling*. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 62–69. IEEE, 2013.

[5] Allwein, Erin L., Robert E. Schapire and Yoram Singer: *Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers*. Journal on Machine Learning Research, 1:113–141, September 2001.

[6] Alvarez, José M, A Lopez and Ramon Baldrich: *Illuminant-invariant model-based road segmentation*. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1175–1180. IEEE, 2008.

[7] Anagnostopoulos, C. N.E., I. E. Anagnostopoulos, V. Loumos and E. Kayafas: *A License Plate-Recognition Algorithm for Intelligent Transportation System Applications*. Transactions on Intelligent Transportation Systems, 7(3):377–392, September 2006.

[8] Andreasson, H. and T. Duckett: *Topological Localization for Mobile Robots Using Omni-directional Vision and Local Features*. In *Proc. IAV 2004, the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004.

[9] Arroyo, R., P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes and S. Bronte: *Fast and effective visual place recognition using binary codes and disparity information*. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3089–3094, Sept 2014.

[10] Arya, Sunil, David M. Mount, Nathan S. Netanyahu, Ruth Silverman and Angela Y. Wu: *An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions*. Journal on ACM, 45(6):891–923, November 1998.

[11] Aycard, O., F. Charpillet, D. Fohr and J. F. Mari: *Place learning and recognition using hidden Markov models*. In *Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 3, pages 1741–1747, Sep 1997.

[12] Badino, Hernán, Uwe Franke and David Pfeiffer: *The Stixel World - A Compact Medium Level Representation of the 3D-World*. In *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, pages 51–60, 2009.

[13] Bay, Herbert, Andreas Ess, Tinne Tuytelaars and Luc Van Gool: *Speeded-Up Robust Features (SURF).* Comput. Vis. Image Underst., 110(3):346–359, June 2008.

[14] Beinschob, P., M. Meyer, C. Reinke, V. Digani, C. Secchi and L. Sabattini: *Semi-Automated Map Creation for Fast Deployment of AGV Fleets in Modern Logistics.* Robotics and Autonomous Systems, 2016.

[15] Bichsel, Robert and Paulo Borges: *Kenmore Dataset*, Apr 2014.

[16] Birk, Andreas, Burkhard Wiggerich, Heiko Bülow, Max Pfingsthorn and Sören Schwertfeger: *Safety, security, and rescue missions with an unmanned aerial vehicle (uav).* Journal of Intelligent & Robotic Systems, 64(1):57–76, 2011.

[17] Biswas, Soma, Gaurav Aggarwal and Rama Chellappa: *Robust estimation of albedo for illumination-invariant matching and shape recovery.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5):884–899, 2009.

[18] Blanco, J.-L., J.-A. Fernandez-Madrigal and J. Gonzalez: *A New Approach for Large-Scale Localization and Mapping: Hybrid Metric-Topological SLAM.* In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2061 –2067, 2007.

[19] Bonarini, Andrea, Wolfram Burgard, Giulio Fontana, Matteo Matteucci, Domenico Giorgio Sorrenti and Juan Domingo Tardos: *RAWSEEDS: Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets.* In *In proceedings of IROS'06 Workshop on Benchmarks in Robotics Research*, 2006.

[20] Bosse, Michael and Robert Zlot: *Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM.* I. J. Robotic Res., 27(6):667–691, 2008.

[21] Bosse, Michael and Robert Zlot: *Keypoint design and evaluation for place recognition in 2D lidar maps.* Robotics and Autonomous Systems, 57(12):1211–1224, 2009.

[22] Bosse, Michael and Robert Zlot: *Place recognition using keypoint voting in large 3D lidar datasets.* In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2677–2684. IEEE, 2013.

[23] Brubaker, Marcus A., Andreas Geiger and Raquel Urtasun: *Lost! Leveraging the Crowd for Probabilistic Visual Self-Localization.* In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[24] Buehler, Martin, Karl Iagnemma and Sanjiv Singh: *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. springer, 2009.

[25] Burgard, Wolfram, Armin B. Cremers, Dieter Fox, Dirk HÃ¤hnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner and Sebastian Thrun: *Experiences with an interactive museum tour-guide robot.* Artificial Intelligence, 114(1):3 – 55, 1999.

[26] Calonder, Michael, Vincent Lepetit, Christoph Strecha and Pascal Fua: *BRIEF: Binary Robust Independent Elementary Features*, pages 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[27] Caraffi, C., T. VojÃÅ, J. TrefnÃ½, J. Å ochman and J. Matas: *A system for real-time detection and tracking of vehicles from a single car-mounted camera.* In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 975–982, Sept 2012.

[28] Carlevaris-Bianco, Nicholas and Ryan M. Eustice: *Learning visual feature descriptors for dynamic lighting conditions.* In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2769–2776, Chicago, IL, USA, September 2014.

[29] Carlevaris-Bianco, Nicholas, Anush Mohan, James R. McBride and Ryan M. Eustice: *Visual localization in fused image and laser range data.* In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4378–4385, San Francisco, CA, USA, September 2011.

[30] Censi, Andrea: *An ICP variant using a point-to-line metric.* In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2008.

[31] Cheng, Ming-Ming, Ziming Zhang, Wen-Yan Lin and Philip H. S. Torr: *BING: Binarized Normed Gradients for Objectness Estimation at 300fps.* In *IEEE CVPR*, 2014.

[32] Churchill, Winston and Paul Newman: *Practice Makes Perfect? Managing and Leveraging Visual Experiences for Lifelong Navigation.* In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.

[33] Churchill, Winston and Paul Newman: *Experience-based Navigation for Long-term Localisation.* The International Journal of Robotics Research (IJRR), 2013.

[34] Civera, Javier, Andrew J Davison and JMM Montiel: *Inverse depth parametrization for monocular SLAM.* IEEE Transactions on Robotics, 24(5):932–945, 2008.

[35] Civera, Javier, Dorian Galvez-Lopez, L. Riazuelo, Juan D. Tardos and J. M. M. Montiel: *Towards semantic SLAM using a monocular camera.* In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1277 –1284, Sept. 2011.

[36] Clemente, Laura A., Andrew J. Davison, Ian D. Reid, José Neira and Juan D. Tardós: *Mapping Large Loops with a Single Hand-Held Camera.* In *Robotics: Science and Systems*, 2007.

[37] Cummins, Mark and Paul Newman: *FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance.* The International Journal of Robotics Research, 27(6):647–665, 2008.

[38] Cummins, Mark and Paul Newman: *Appearance-only SLAM at large scale with FAB-MAP 2.0.* The International Journal of Robotics Research, 30(9):1100–1123, 2011.

[39] Dalal, N.: *INRIA person dataset.* `http://pascal.inrialpes.fr/data/human/`, 2005. [Online; accessed 14-September-2015].

[40] Dellaert, Frank: *Factor graphs and GTSAM: A hands-on introduction.* Technical Report, Georgia Institute of Technology, 2012.

[41] Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei: *ImageNet: A Large-Scale Hierarchical Image Database.* In *CVPR09*, 2009.

[42] Dissanayake, M. W. M. Gamini, Paul Newman, Steven Clark, Hugh F. Durrant-whyte and M. Csorba: *A solution to the simultaneous localization and map building (SLAM) problem*. IEEE Transactions on Robotics and Automation, 17:229–241, 2001.

[43] Dubbelman, Gijs and Brett Browning: *Closed-form online pose-chain slam*. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5190–5197. IEEE, 2013.

[44] Duda, Richard O., Peter E. Hart and David G. Stork: *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[45] Durrant-Whyte, Hugh F: *Uncertain geometry in robotics*. IEEE Journal on Robotics and Automation, 4(1):23–31, 1988.

[46] Einhorn, E. and H. M. Gross: *Generic 2D/3D SLAM with NDT maps for lifelong application*. In *2013 European Conference on Mobile Robots*, pages 240–247, Sept 2013.

[47] Engelson, Sean P.: *Active place recognition using image signatures*. Volume 1828, pages 1828 – 1828 – 12, 1992.

[48] Fallon, M., H. Johannsson, Michael Kaess and J.J. Leonard: *The MIT Stata Center Dataset*. International Journal of Robotics Research (IJRR), 32(14):1695–1699, December 2013.

[49] Fallon, M. F., H. Johannsson and J. J. Leonard: *Efficient scene simulation for robust monte carlo localization using an RGB-D camera*. In *2012 IEEE International Conference on Robotics and Automation*, pages 1663–1670, May 2012.

[50] Finman, R., T. Whelan, L. Paull and J. J. Leonard: *Physical Words for Place Recognition in Dense RGB-D Maps*. In *ICRA workshop on visual place recognition in changing environments*, June 2014.

[51] Fischler, M. A. and R. C. Bolles: *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Communications of the ACM, 24:381–395, 1981.

[52] Fontana, Giulio, Matteo Matteucci and Domenico G. Sorrenti: *Rawseeds: Building a Benchmarking Toolkit for Autonomous Robotics*, pages 55–68. Springer International Publishing, Cham, 2014.

[53] Fox, D.: *KLD-Sampling: Adaptive Particle Filters*. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[54] Fox, D Hähnel D, W Burgard and S Thrun: *A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements*. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[55] Frese, Udo: *Interview: Is SLAM Solved?* KI - Künstliche Intelligenz, 24(3):255–257, Sep 2010.

[56] Frost, Jan: *Robust and scalable visual simultaneous localization and mapping in indoor environments using RGBD cameras*. PhD thesis, 2017.

[57] Galvez-Lopez, Dorian and J. D. Tardos: *Bags of Binary Words for Fast Place Recognition in Image Sequences*. IEEE Transactions on Robotics, 28(5):1188–1197, October 2012.

[58] Geiger, Andreas and Chaohui Wang: *Joint 3D Object and Layout Inference from a single RGB-D Image*. In *German Conference on Pattern Recognition (GCPR)*, 2015.

[59] Granström, Karl, Thomas B. Schön, Juan I. Nieto and Fabio T. Ramos: *Learning to close loops from range data.* International Journal of Robotic Research, 30(14):1728–1754, 2011.

[60] Grimmett, Hugo, Mathias Buerki, Lina Paz, Pedro Piniés, Paul Furgale, Ingmar Posner and Paul Newman: *Integrating Metric and Semantic Maps for Vision-Only Automated Parking*. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.

[61] Grisetti, G., S. Grzonka, C. Stachniss, P. Pfaff and W. Burgard: *Efficient Estimation of Accurate Maximum Likelihood Maps in 3D*. pages 3472–3478, San Diego, CA (USA), 2007. DOI: 10.1109/IROS.2007.4399030.

[62] Grisetti, G., R. KFmmerle, C. Stachniss and W. Burgard: *A Tutorial on Graph-Based SLAM*. Intelligent Transportation Systems Magazine, IEEE, 2(4):31–43, 2010.

[63] Grisetti, G., R. Kummerle, C. Stachniss, U. Frese and C. Hertzberg: *Hierarchical optimization on manifolds for online 2D and 3D mapping*. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 273–278, 2010.

[64] Grisetti, G., C. Stachniss and W. Burgard: *Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling*. In *Robotics and Automation, 2005. Proc. of the 2005 IEEE International Conference on*, pages 2432 – 2437, Apr. 2005.

[65] Gross, H-M, H Boehme, Ch Schroeter, Steffen Müller, Alexander König, Erik Einhorn, Ch Martin, Matthias Merten and Andreas Bley: *TOOMAS: interactive shopping guide robots in everyday use-final implementation and experiences from long-term field trials*. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2005–2012. IEEE, 2009.

[66] Gross, H. M., A. Koenig, H. J. Boehme and C. Schroeter: *Vision-based Monte Carlo self-localization for a mobile service robot acting as shopping assistant in a home store*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 256–262 vol.1, 2002.

[67] Guivant, J. and E. Nebot. Technical Report, 2001.

[68] Hansen, P. and B. Browning: *Visual place recognition using HMM sequence matching*. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4549–4555, Sept 2014.

[69] Hartley, R. I. and A. Zisserman: *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, Second edition, 2004.

[70] Hartmann, Jan, Jan Helge Klüssendorff and Erik Maehle: *A unified visual graph-based approach to navigation for wheeled mobile robots.* In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1915–1922. IEEE, 2013.

[71] Hellbach, Sven, Marian Himstedt, Frank Bahrmann, Martin Riedel, Thomas Villmann and Hans-Joachim Böhme: *Find Rooms for Improvement: Towards Semi-automatic Labeling of Occupancy Grid Maps.* In *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part III*, 2014.

[72] Hellbach, Sven, Marian Himstedt and Hans-Joachim Boehme: *What's around me: Towards non-negative matrix factorization based localization.* In *Mobile Robots (ECMR), 2013 European Conference on*, pages 228–233, Sept 2013.

[73] Himstedt, M., J. Frost, S. Hellbach, H.-J. Böhme and E. Maehle: *Large scale place recognition in 2D LIDAR scans using Geometrical Landmark Relations.* In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2014.

[74] Himstedt, M., S. Keil, S. Hellbach and H.-J. Böhme: *A robust graph-based framework for building precise maps from laser range scans.* In *Proceedings of the Workshop on Robust and Multimodal Inference in Factor Graphs. IEEE ICRA*, May 2013.

[75] Himstedt, M. and E. Maehle: *Camera-based Obstacle Classification for Automated Reach Trucks Using Deep Learning.* In *Proceedings of ISR 2016: 47st International Symposium on Robotics*, pages 1–6, June 2016.

[76] Himstedt, Marian, Alen Alempijevic, Liang Zhao, Shoudong Huang and Hans-Joachim Böhme: *Towards robust vision-based self-localization of vehicles in dense urban environments.* In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012.

[77] Himstedt, Marian and Erik Maehle: *Geometry matters: Place Recognition in 2D range scans using Geometrical Surface Relations.* In *Mobile Robots (ECMR), 2015 European Conference on*, Sept 2015.

[78] Himstedt, Marian and Erik Maehle: *Online semantic mapping of logistic environments using RGB-D cameras.* International Journal of Advanced Robotic Systems, 14(4), 2017.

[79] Himstedt, Marian and Erik Maehle: *Semantic Monte-Carlo Localization in Changing Environments using RGB-D Cameras.* In *Mobile Robots (ECMR), 2017 European Conference on*, Sept 2017.

[80] Hornung, Armin, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss and Wolfram Burgard: *OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees.* Autonomous Robots, 2013.

[81] Hsu, Chih-Wei and Chih-Jen Lin: *A Comparison of Methods for Multi-class Support Vector Machines.* IEEE Transactions on Neural Networks, 13(2):415–425, March 2002.

[82] Huang, Fu-Jie and Yann LeCun: *Large-Scale Learning with SVM and Convolutional Nets for Generic Object Categorization.* In *Proc. Computer Vision and Pattern Recognition Conference (CVPR'06)*, 2006.

[83] Huang, Shoudong, Zhan Wang and Gamini Dissanayake: *Sparse local submap joining filter for building large-scale maps.* IEEE Transactions on Robotics, 24(5):1121–1130, 2008.

[84] Indelman, Vadim, Stephen Williams, Michael Kaess and Frank Dellaert: *Factor graph based incremental smoothing in inertial navigation systems.* In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 2154–2161. IEEE, 2012.

[85] Irschara, A., C. Zach, J. M. Frahm and H. Bischof: *From structure-from-motion point clouds to fast location recognition.* In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606, June 2009.

[86] Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama and Trevor Darrell: *Caffe: Convolutional Architecture for Fast Feature Embedding.* arXiv preprint arXiv:1408.5093, 2014.

[87] Johns, E and G-Z Yang: *Feature Co-occurrence Maps: Appearance-based Localisation Throughout the Day.* In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3212–3218. IEEE, 2013.

[88] Johns, E. and Guang-Zhong Yang: *Dynamic scene models for incremental, long-term, appearance-based localisation.* In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2731–2736, May 2013.

[89] Kaess, M., H. Johannsson, R. Roberts, V. Ila, J.J. Leonard and F. Dellaert: *iSAM2: Incremental Smoothing and Mapping with Fluid Relinearization and Incremental Variable Reordering.* In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3281–3288, Shanghai, China, May 2011.

[90] Kaess, M., H. Johannsson, R. Roberts, V. Ila, J.J. Leonard and F. Dellaert: *iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree.* Intl. J. of Robotics Research (IJRR), 31:217–236, February 2012.

[91] Kaess, M., A. Ranganathan and F. Dellaert: *iSAM: Incremental Smoothing and Mapping.* IEEE Trans. on Robotics (TRO), 24(6):1365–1378, December 2008.

[92] Klein, Georg and David Murray: *Parallel tracking and mapping for small AR workspaces.* In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[93] Kohlbrecher, Stefan, Oskar Von Stryk, Johannes Meyer and Uwe Klingauf: *A flexible and scalable slam system with full 3d motion estimation.* In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160. IEEE, 2011.

[94] Konolige, K., G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai and R. Vincent: *Efficient Sparse Pose Adjustment for 2D Mapping.* In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[95] Kosecka, Jana, Fayin Li and Xiaolong Yang: *Global localization and relative positioning based on scale-invariant keypoints.* Robotics and Autonomous Systems, 52:27–38, 2005.

[96] Kosnar, Karel, Vojtech Vonasek, Miroslav Kulich and Libor Preucil: *Comparison of shape matching techniques for place recognition.* In *Mobile Robots (ECMR), 2013 European Conference on*, pages 107–112. IEEE, 2013.

[97] Krajník, Tomáš, Jaime Pulido Fentanes, Oscar M. Mozos, Tom Duckett, Johan Ekekrantz and Marc Hanheide: *Long-Term Topological Localization for Service Robots in Dynamic Environments using Spectral Maps.* In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[98] Kretzschmar, Henrik and Cyrill Stachniss: *Information-Theoretic Compression of Pose Graphs for Laser-Based SLAM.* 31:1219–1230, 2012.

[99] Krizhevsky, Alex, Ilya Sutskever and Geoffrey E. Hinton: *ImageNet Classification with Deep Convolutional Neural Networks.* In Pereira, F., C.J.C. Burges, L. Bottou and K.Q. Weinberger (editors): *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[100] Kummerle, Rainer, G. Grisetti, H. Strasdat, Kurt Konolige and Wolfram Burgard: *g2o: A General Framework for Graph Optimization.* In *ICRA*, Shanghai, 2011.

[101] Kümmerle, Rainer, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss and Alexander Kleiner: *On measuring the accuracy of SLAM algorithms.* Autonomous Robots, 27(4):387, Sep 2009.

[102] Labbe, M. and F. Michaud: *Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation.* IEEE Transactions on Robotics, 29(3):734–745, June 2013.

[103] Lamon, P., I. Nourbakhsh, B. Jensen and R. Siegwart: *Deriving and matching image fingerprint sequences for mobile robot localization.* In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 1609–1614 vol.2, 2001.

[104] Larry Zitnick, Piotr Dollar: *Edge Boxes: Locating Object Proposals from Edges.* In *ECCV*. European Conference on Computer Vision, September 2014.

[105] LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: *Backpropagation Applied to Handwritten Zip Code Recognition.* Neural Computation, 1:541–551, 1989.

[106] Leonard, John, Hugh Durrant-Whyte and Ingemar J Cox: *Dynamic map building for autonomous mobile robot.* In *Intelligent Robots and Systems' 90.'Towards a New Frontier of Applications', Proceedings. IROS'90. IEEE International Workshop on*, pages 89–96. IEEE, 1990.

[107] Leutenegger, Stefan, Simon Lynen, Michael Bosse, Roland Siegwart and Paul Furgale: *Keyframe-based visual–inertial odometry using nonlinear optimization.* The International Journal of Robotics Research, 34(3):314–334, 2015.

[108] Li, Fei-Fei and Pietro Perona: *A Bayesian Hierarchical Model for Learning Natural Scene Categories.* In *CVPR*, 2005.

[109] Liu, Yang and Hong Zhang: *Visual loop closure detection with a compact image descriptor.* In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1051–1056, Oct 2012.

[110] Lowe, David G.: *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, 60(2):91–110, Nov 2004.

[111] Luo, Jun, Yong Ma, Erina Takikawa, Shihong Lao, Masato Kawade and Bao-Liang Lu: *Person-specific SIFT features for face recognition.* In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2. IEEE, 2007.

[112] Magnusson, Martin, H. Andreasson, A. Nuchter and A.J. Lilienthal: *Appearance-based loop detection from 3D laser data using the normal distributions transform.* In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2009.

[113] Meyer-Delius, D., J. Hess, G. Grisetti and W. Burgard: *Temporary Maps for Robust Localization in Semi-static Environments.* In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.

[114] Meyer-Delius, Daniel: *Probabilistic Modeling of Dynamic Environments for Mobile Robots.* PhD thesis, 2011.

[115] Milford, Michael: *Vision-based place recognition: how low can you go?* The International Journal of Robotics Research, 32(7):766–789, 2013.

[116] Milford, Michael, Stephanie Lowry, Niko Sünderhauf, Sareh Shirazi, Edward Pepperell, Ben Upcroft, Chunhua Shen, Guosheng Lin, Fayao Liu, Cesar Cadena and Ian Reid: *Sequence searching with deep-learnt depth for condition-and viewpoint-invariant route-based place recognition.* In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2015)*, pages 18–25, Boston, MA, 2015.

[117] Milford, Michael and Gordon Wyeth: *SeqSLAM : visual route-based navigation for sunny summer days and stormy winter nights.* In *IEEE International Conferece on Robotics and Automation (ICRA 2012)*, pages 1643–1649, River Centre, Saint Paul, Minnesota, 2012.

[118] Montemerlo, Michael, Sebastian Thrun, Daphne Koller and Ben Wegbreit: *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem.* pages 593–598. AAAI Press / The MIT Press, 2002.

[119] Morel, Jean-Michel and Guoshen Yu: *ASIFT: A New Framework for Fully Affine Invariant Image Comparison.* SIAM J. Img. Sci., 2(2):438–469, April 2009.

[120] Muja, Marius and David G. Lowe: *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.* In *International Conference on Computer Vision Theory and Application (VISSAPP)*, pages 331–340, 2009.

[121] Mur-Artal, Raul, Jose Maria Martinez Montiel and Juan D Tardos: *ORB-SLAM: a versatile and accurate monocular SLAM system.* IEEE Transactions on Robotics, 31(5):1147–1163, 2015.

[122] Naseer, Tayyab, Luciano Spinello, Wolfram Burgard and Cyrill Stachniss: *Robust Visual Robot Localization Across Seasons Using Network Flows.* In *AAAI*, 2014.

[123] Nathan Silberman, Derek Hoiem, Pushmeet Kohli and Rob Fergus: *Indoor Segmentation and Support Inference from RGBD Images.* In *ECCV*, 2012.

[124] Neubert, Peer, Niko Sunderhauf and Peter Protzel: *Superpixel-based appearance change prediction for long-term navigation across seasons.* Robotics and Autonomous Systems, 69:15–27, August 2014.

[125] Neubert, Peer, Niko Sünderhauf and Peter Protzel: *Superpixel-based appearance change prediction for long-term navigation across seasons.* Robotics and Autonomous Systems, 69:15 – 27, 2015.

[126] Nüchter, Andreas and Joachim Hertzberg: *Towards Semantic Maps for Mobile Robots.* Robot. Auton. Syst., 56(11):915–926, November 2008.

[127] Nuske, Stephen, Jonathan Roberts and Gordon Wyeth: *Robust outdoor visual localization using a three-dimensional-edge map.* Journal of Field Robotics, 26(9):728–756, 2009.

[128] Ojala, T., M. Pietikainen and D. Harwood: *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions.* In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 582–585, Oct 1994.

[129] Olson, Edwin and Pratik Agarwal: *Inference on networks of mixtures for robust robot mapping.* In *Proceedings of Robotics: Science and Systems (RSS)*, Sydney, Australia, July 2012.

[130] Olson, Edwin, John Leonard and Seth Teller: *Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates.* pages 2262–2269, 2006.

[131] Overmeyer, L., F. Podszus and L. Dohrmann: *Multimodal speech and gesture control of AGVs, including EEG-based measurements of cognitive workload.* CIRP Annals - Manufacturing Technology, 65(1):425–428, 2016.

[132] Paul, R. and P. Newman: *FAB-MAP 3D: Topological mapping with spatial and visual appearance.* In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010.

[133] Pepperell, Edward, Peter Corke and Michael Milford: *Towards Vision-Based Pose- and Condition-Invariant Place Recognition along Routes.* In *Australasian Conference on Robotics and Automation (ACRA2014)*, University of Melbourne, Melbourne, Australia, December 2014. Australian Robotics & Automation Association ARAA.

[134] Philbin, James, Ondrej Chum, Michael Isard, Josef Sivic and Andrew Zisserman: *Lost in quantization: Improving particular object retrieval in large scale image databases.* In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[135] Pronobis, Andrzej and Barbara Caputo: *COLD: COsy Localization Database.* The International Journal of Robotics Research (IJRR), 28(5):588–594, May 2009.

[136] Pronobis, Andrzej, Oscar M. Mozos, Barbara Caputo and Patric Jensfelt: *Multi-modal Semantic Place Classification.* The International Journal of Robotics Research (IJRR), Special Issue on Robotic Vision, 29(2-3):298–320, February 2010.

[137] Reinke, Christoph and Patric Beinschob: *Strategies for contour-based self-localization in large-scale modern warehouses.* In *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on*, pages 223–227. IEEE, 2013.

[138] Rosten, Edward, Reid Porter and Tom Drummond: *FASTER and better: A machine learning approach to corner detection.* IEEE Trans. Pattern Analysis and Machine Intelligence, 32:105–119, 2010.

[139] Ruhnke, M., R. Kümmerle, G. Grisetti and W Burgard: *Highly Accurate Maximum Likelihood Laser Mapping by Jointly Optimizing Laser Points and Robot Poses.* In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[140] Rusu, Radu Bogdan and Steve Cousins: *3D is here: Point Cloud Library (PCL).* In *IEEE ICRA*, Shanghai, China, May 9-13 2011.

[141] Saarinen, Jari, Henrik Andreasson and Achim J. Lilienthal: *Independent Markov Chain Occupancy Grid Maps for Representation of Dynamic Environments.* In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.

[142] Saarinen, Jari, Henrik Andreasson, Todor Stoyanov and Achim J. Lilienthal: *Normal distributions transform Monte-Carlo localization (NDT-MCL).* In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[143] Sande, K. E. A. van de, J. R. R. Uijlings, T. Gevers and A. W. M. Smeulders: *Segmentation As Selective Search for Object Recognition.* In *IEEE ICCV*, 2011.

[144] Sattler, T., B. Leibe and L. Kobbelt: *Fast image-based localization using direct 2D-to-3D matching.* In *2011 International Conference on Computer Vision*, pages 667–674, Nov 2011.

[145] Sattler, Torsten, Bastian Leibe and Leif Kobbelt: *Improving image-based localization by active correspondence search.* In *European conference on computer vision*, pages 752–765. Springer, 2012.

[146] Scharwächter, Timo, Markus Enzweiler, Uwe Franke and Stefan Roth: *Stixmantics: A Medium-Level Model for Real-Time Semantic Scene Understanding.* In *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, 2014.

[147] Sharif Razavian, Ali, Hossein Azizpour, Josephine Sullivan and Stefan Carlsson: *CNN Features Off-the-Shelf: An Astounding Baseline for Recognition.* In *IEEE CVPR Workshops*, June 2014.

[148] Sivic, J., B. C. Russell, A. A. Efros, A. Zisserman and W. T. Freeman: *Discovering Object Categories in Image Collections.* In *Proceedings of the International Conference on Computer Vision*, 2005.

[149] Steder, B., G. Grisetti and W. Burgard: *Robust place recognition for 3D range data based on point features.* In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010.

[150] Strasdat, Hauke, Richard A. Newcombe, Renato F. Salas-Moreno, Paul H.J. Kelly and Andrew J. Davison: *SLAM++: Simultaneous Localisation and Mapping at the Level of Objects*. IEEE CVPR, year = 2013, pages = 1352-1359, address = Los Alamitos, CA, USA,.

[151] Strasdat, Hauke, Richard A. Newcombe, Renato F. Salas-Moreno, Paul H.J. Kelly and Andrew J. Davison: *SLAM++: Simultaneous Localisation and Mapping at the Level of Objects*. 2013 IEEE Conference on Computer Vision and Pattern Recognition, pages 1352–1359, 2013.

[152] Stückler, J., N. Biresev and S. Behnke: *Semantic mapping using object-class segmentation of RGB-D images.* In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3005–3010, Oct 2012.

[153] Sünderhauf, N., F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft and M. Milford: *Place categorization and semantic mapping on a mobile robot.* In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016.

[154] Sunderhauf, N. and P. Protzel: *Towards a robust back-end for pose graph SLAM.* In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1254 –1261, may 2012.

[155] Sünderhauf, Niko: *Robust Optimization for Simultaneous Localization and Mapping.* PhD thesis, 2012.

[156] Sünderhauf, Niko, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft and Michael Milford: *Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free.* In *Robotics: Science and Systems*, July 2015.

[157] Sünderhauf, Niko, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft and Michael Milford: *Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free.* In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[158] Svab, J., T. Krajnik, J.Faigl and L. Preucil: *FPGA-based Speeded Up Robust Features.* In *2009 IEEE International Conference on Technologies for Practical Robot Applications 2009*, November 2009.

[159] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich: *Going Deeper with Convolutions.* In *CVPR 2015*, 2015.

[160] Thrun, S., W. Burgard and D. : *Probabilistic robotics.* Intelligent robotics and autonomous agents. MIT Press, 2005.

[161] Thrun, Sebastian and Michael Montemerlo: *The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures.* The International Journal of Robotics Research, 25(5-6):403–429, 2006.

[162] Tipaldi, G.D. and K.O. Arras: *FLIRT - Interest regions for 2D range data.* In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3616 –3622, 2010.

[163] Tipaldi, Gian Diego, Daniel Meyer-Delius and Wolfram Burgard: *Lifelong localization in changing environments.* International Journal of Robotics Research, 32(14), December 2013.

[164] Tipaldi, Gian Diego, Luciano Spinello and Wolfram Burgard: *Geometrical FLIRT Phrases for Large Scale Place Recognition in 2D Range Data.* In *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[165] Triebel, Rudolph, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, Daniel Cremers, Vanessa Evers, Michelangelo Fiore et al.: *Spencer: A socially aware service robot for passenger guidance and help in busy airports.* In *Field and Service Robotics*, pages 607–622. Springer, 2016.

[166] Ulrich, I. and I. Nourbakhsh: *Appearance-based place recognition for topological localization.* In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 1023–1029 vol.2, 2000.

[167] Vysotska, O. and C. Stachniss: *Lazy Sequences Matching Under Substantial Appearance Changes.* In *Workshop on Visual Place Recognition in Changing Environments at the International Conference on Robotics and Automation (ICRA)*, 2015.

[168] Whelan, Thomas, Hordur Johannsson, Michael Kaess, John J Leonard and John McDonald: *Robust real-time visual odometry for dense RGB-D mapping.* In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5724–5731. IEEE, 2013.

[169] Williams, B., M. Cummins, J. Neira, P. Newman, I. Reid and J. Tardós: *A comparison of loop closing techniques in monocular SLAM.* Robotics and Autonomous Systems, 2009.

[170] Xiang, Yu, Wongun Choi, Yuanqing Lin and Silvio Savarese: *Data-Driven 3D Voxel Patterns for Object Category Recognition.* In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1903–1911. 2015.

[171] Yonglong, Z., M. Kuizhi, J. Xiang and D. Peixiang: *Parallelization and Optimization of SIFT on GPU Using CUDA.* In *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pages 1351–1358, Nov 2013.

[172] Yu, Yinan, Kaiqi Huang, Wei Chen and Tieniu Tan: *A novel algorithm for view and illumination invariant image matching.* IEEE transactions on image processing, 21(1):229–240, 2012.

[173] Zhang, Yimeng, Zhaoyin Jia and Tsuhan Chen: *Image retrieval with geometry-preserving visual phrases.* In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 809–816, 2011.

[174] Zhu, Qiang, Mei-Chen Yeh, Kwang-Ting Cheng and S. Avidan: *Fast Human Detection Using a Cascade of Histograms of Oriented Gradients.* In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498, 2006.