Robust Visual Self-localization and Navigation in Outdoor Environments Using Slow Feature Analysis

Benjamin Metka

Thesis submitted to the Faculty of Technology
at the Bielefeld University for obtaining the academic degree
Doctor of Engineering (Dr. Ing.)

Supervisors:

Prof. Dr. Ute Bauer-Wersing, Frankfurt University of Applied Sciences
Dr. Mathias Franzius, Honda Research Institute Europe GmbH
Prof. Dr. Helge Ritter, Bielefeld University

Printed on non-aging paper according to DIN-ISO 9706.

Abstract

Self-localization and navigation in outdoor environments are fundamental problems a mobile robot has to solve in order to autonomously execute tasks in a spatial environment. Techniques based on the Global Positioning System (GPS) or laser-range finders have been well established but suffer from the drawbacks of limited satellite availability or high hardware effort and costs. Vision-based methods can provide an interesting alternative, but are still a field of active research due to the challenges of visual perception such as illumination and weather changes or long-term seasonal effects.

This thesis approaches the problem of robust visual self-localization and navigation using a biologically motivated model based on unsupervised Slow Feature Analysis (SFA). It is inspired by the discovery of neurons in a rat's brain that form a neural representation of the animal's spatial attributes. A similar hierarchical SFA network has been shown to learn representations of either the position or the orientation directly from the visual input of a virtual rat depending on the movement statistics during training.

An extension to the hierarchical SFA network is introduced that allows to learn an orientation invariant representation of the position by manipulating the perceived image statistics exploiting the properties of panoramic vision. The model is applied on a mobile robot in real world open field experiments obtaining localization accuracies comparable to state-of-the-art approaches. The self-localization performance can be further improved by incorporating wheel odometry into the purely vision based approach. To achieve this, a method for the unsupervised learning of a mapping from slow feature to metric space is developed. Robustness w.r.t. short- and long-term appearance changes is tackled by re-structuring the temporal order of the training image sequence based on the identification of crossings in the training trajectory. Re-inserting images of the same place in different conditions into the training sequence increases the temporal variation of environmental effects and thereby improves invariance due to the slowness objective of SFA. Finally, a straightforward method for navigation in slow feature space is presented. Navigation can be performed efficiently by following the SFA-gradient, approximated from distance measurements between the slow feature values at the target and the current location. It is shown that the properties of the learned representations enable complex navigation behaviors without explicit trajectory planning.

Acknowledgments

First, I would like to thank my university supervisors Ute Bauer-Wersing and Helge Ritter for giving me the opportunity to work on this thesis. Especially, I want to thank Ute Bauer-Wersing for all the involved organizational work, her encouraging words and some late-night proof reading just before submission deadline. Big thanks also to Mathias Franzius for giving me a lot of freedom in doing my work and sharing his bright ideas with me when I was lost. This thesis would definitely miss some pages without his support. I would also like to thank the Bachelor and Master students who contributed to the work presented in this thesis: Annika Besetzny, Marius Anderie, Muhammad Haris and Benjamin Löffler. I want to thank the Honda Research Institute for funding my PhD position and all the colleagues for creating a friendly and constructive working atmosphere. Special thanks to the people who shared the office with me for many on- and off-topic discussions, the relaxed atmosphere and for making the four years a joyful experience: Amadeus, Christian, Dennis, Marvin and Viktor.

Contents

List of Tables

List of Figures

1 Introduction

Nowadays, there already exist domestic service robots that perform repetitive or unpleasant tasks to support us in our daily lives. Vacuum cleaning and lawn mowing robots have been one of the first autonomous robots available as consumer products. Although they are enjoying a growing popularity, their current capabilities are still rather limited. The employed navigation strategies are often constrained to movements along random line segments combined with reactive collision avoidance and some functionality to return to the charging station. To implement a more intelligent navigation behavior a mobile robot needs to create an internal representation or a map from a previously unknown environment in order to determine its own position and plan efficient and viable trajectories. The problems of building a map, localizing within this map as well as planning and executing a path to a target location are fundamental to many robotic application scenarios. This has raised great research interest in technologies that enable a mobile device to precisely navigate in unconstrained environments. Techniques based on the Global Positioning System (GPS) or laser-range finders have been well established. However, the limited accuracy and availability of GPS and the high cost of laser-range finders prevent their use in domestic service robots produced for the mass market. Cameras on the other hand are cheap, small and passive sensors that offer rich information about the environment and thus provide an interesting alternative. A number of vacuum cleaning robots are already equipped with a camera (e.g. Dyson 360 Eye, Samsung Hauzen) and implement more advanced navigation strategies in the constrained indoor scenario using visual information from the static room ceiling [60]. Research in the field of vision based outdoor navigation is steadily progressing as well and recent work has shown impressive results in mapping large scale environments (e.g. [22, 136, 80, 35, 109]). However, long-term operation in unconstrained outdoor environments is still not robustly solved due to the challenges of visual perception such as changing lighting or weather conditions, different day times or seasons and structural scene changes that strongly influence the visual appearance of a place.

Compared to current technical systems many animals have excellent navigation capabilities and are able to quickly and robustly find their way to a food source or their nest. In the brain of rodents spatial information is encoded by different cell types in the hippocampal formation. Place cells fire whenever the animal is within a specific part of the environment and are mostly insensitive to the orientation of the animal [120]. Head-direction cells, on the other hand, are active when the animal is facing in a certain direction and are invariant w.r.t. its position [154]. Both cell types have been shown to be strongly driven by visual input [59]. The brain is able to extract high level information, like the own position and orientation in the environment, from the raw visual signals received by the retina. While the sensory signals of single receptors may change very rapidly, e.g. even by slight eye movement, the embedded high level information typically changes on a much lower timescale. This observation has led to the concept of slowness learning [41, 143, 162, 73]. It has already been demonstrated in recent work that a hierarchical network consisting of unsupervised Slow Feature Analysis (SFA) [162] nodes can model the firing behavior of either place cells or head-direction cells from the visual input of a virtual rat only [42]. A theoretical analysis of the biomorphic model in [42] has shown that in slowness learning, the resulting representation strongly depends on the movement statistics of the animal. Position encoding with invariance to head direction requires a relatively large amount of head rotation around the yaw axis compared to translational movement during mapping of the environment. While such movement may be realistic for a rodent exploring its environment, it is inefficient for a robot with a fixed camera.

The goal of this thesis is the extension and further investigation of the biologically motivated SFA model in order to derive methods for self-localization, the creation of robust environmental representations and navigation that can be applied in outdoor open field scenarios on a real mobile robot.

1.1 Contributions and Outline

This thesis employs the biologically motivated SFA model for spatial representation learning as a basis to address three fundamental problems a mobile robot has to solve in order to autonomously plan and execute tasks within its environment: the ability to perform self-localization, the creation of robust environment representations and the navigation to a specific target location.

Chapter 2 gives an overview of related work which is concerned with solving these problems using vision as the only sensory input.

Chapter 3 details the biologically motivated SFA model that is used to learn a representation of the environment directly from the visual input of a mobile robot. It is based on unsupervised slowness learning and encodes the position of the robot as slowly varying features. The intuition behind slowness learning as well as the concrete algorithm Slow Feature Analysis (SFA) are presented first. Afterwards, the SFA model for spatial cell learning from [42] is introduced. We present an extension to this model allowing to learn orientation invariant representations of the position without requiring a large amount of physical rotational movement. The last section describes the methods for analyzing the learned slow feature representations.

The procedures for generating and capturing the data for the simulator and real world experiments are described in chapter 4. To perform a quantitative metric evaluation the knowledge of the robot's true position within the environment is required. Since this ground truth information is not directly available in real world settings it has to be acquired by an external system. In the last section of this chapter, we describe a method for ground truth data acquisition based on optical marker detection.

In chapter 5, the spatial accuracy of the learned slow feature representation is analyzed in various simulator and real world self-localization experiments and compared to state-of-the-art vision based methods. Furthermore, we present an unsupervised learning approach to obtain a mapping from slow feature to metric space. The learned mapping enables the integration of odometry information into the self-localization process to further improve performance. In the last section, an alternative approach for learning spatial SFA representations from single and multiple tracked landmark views is presented.

The problem of creating robust environmental representations enabling a mobile robot to reliably localize itself in changing outdoor scenarios using visual input from a camera only is tackled in chapter 6. First, we investigate the long-term robustness of local visual features computed for distinct image patches. These features are commonly used in the context of localization and mapping and could also serve to create alternative image representations for training the SFA model. Based on these findings, we propose a generic approach to improve long-term mapping and localization robustness by learning a selection criterion for long-term stable visual features which can be integrated into the standard feature processing pipeline. As an alternative, we introduce a unified approach towards long-term robustness that is solely based on SFA. It takes advantage of the invariance learning capabilities of SFA by restructuring the temporal order of the training sequence in order to promote robustness w.r.t. short- and long-term environmental effects.

In chapter 7, we propose a straightforward approach for efficient navigation in slow feature space using gradient descent. A navigation direction can be inferred from distance measurements between the slow feature values at the current and the target location. It is experimentally shown that the learned slow feature representations enable a reliable and efficient navigation and implicitly encode information about obstacles which are reflected in the SFA gradients. Thus, complex navigation tasks can be solved without explicit trajectory or obstacle avoidance planning. Furthermore, we present preliminary results on an extension to the proposed navigation method for improving robustness in real world applications scenarios and empirically investigate interesting properties of the slow feature representations leading to surprising navigation behaviors.

Finally, chapter 8 summarizes the main contributions and concludes this thesis.

1.2 Publications in the Context of this Thesis

- M. Franzius, B. Metka, and U. Bauer-Wersing. *Unsupervised Learning of Metric Representations with Slow Features.* Submitted to the International Conference on Intelligent Robots and Systems (IROS), 2018.

- B. Metka, M. Franzius, and U. Bauer-Wersing. *Bio-inspired visual self-localization in real world scenarios using slow feature analysis.* PLOS ONE, 13(9):1-18, 2018.

- B. Metka, M. Franzius, and U. Bauer-Wersing. *Efficient Navigation Using Slow Feature Gradients.* In Proceedings of the 30th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1311-1316, Vancouver, Canada, 2017.

- M. Haris, B. Metka, M. Franzius, and U. Bauer-Wersing. *Condition Invariant Visual Localization Using Slow Feature Analysis.* In Machine Learning Reports 03/2017, pages 7-8, 2017.

- B. Metka, M. Franzius, and U. Bauer-Wersing. *Improving Robustness of Slow Feature Analysis Based Localization Using Loop Closure Events.* In Proceedings of the 25th International Conference on Artificial Neural Networks (ICANN), pages 489-496, Barcelona, Spain, 2016.

- B. Metka, A. Besetzny, U. Bauer-Wersing, and M. Franzius. *Predicting the Long-Term Robustness of Visual Features.* In Proceedings of the 17th International Conference on Advanced Robotics (ICAR), pages 465-470, Istanbul, Turkey, 2015.

- B. Metka, M. Franzius, and U. Bauer-Wersing. *Outdoor Self-Localization of a Mobile Robot Using Slow Feature Analysis.* In Proceedings of the 20th International Conference on Neural Information Processing (ICONIP), pages 249-256, Daegu, South Korea, 2013.

2 Localization, Mapping and Navigation

This chapter serves as an overview of the main methods enabling autonomously navigating mobile robots using vision as the only sensory input. In order to determine its own location within the environment a robot needs an internal representation of the environment. However, the construction of such a map from sensor measurements in turn requires knowledge of the precise position. Therefore, the problem of localization and mapping is usually solved simultaneously in an incremental fashion. The next section briefly introduces established approaches for simultaneous localization and mapping (SLAM) but also reviews methods trying to mimic biological models and the recently emerging methods based on deep learning. Changes in the environment caused by different lighting conditions, seasons or structural scene changes induce high variability into the appearance of a place and thus pose a severe challenge for vision based localization and mapping methods. Section 2.2 gives an overview of a variety of approaches aiming at robust long-term operation. The ability to create a map and to determine the own position are the prerequisites enabling a mobile robot to perform the high-level task of navigation. Navigation methods based on different environment representations are reviewed in section 2.3.

2.1 Localization and Mapping

The ability to build a map of the environment and to determine the own location within the acquired map is a prerequisite for autonomously acting mobile robots. While localization and mapping can be performed with different kinds of sensors, vision based approaches are especially appealing because of the low cost, weight and the high availability of cameras. Research in the field is steadily progressing and recent work has shown impressive results in mapping large scale environments (e.g. [22, 136, 80, 35, 109]). Most vision based approaches extract local visual features from the captured images to estimate the motion of the camera and create a sparse 3D representation of the environment. The first step in feature extraction is the identification of accurately localizable and distinguishable interest points in the image like corners [53, 140, 130] or blobs [92, 86, 10]. Afterwards, a descriptor is created from the surrounding image patch using gradient information [86, 97, 10] or pixel-wise intensity comparisons [17, 131, 82, 2]. Correspondences between features from the current image and stored map features can be established by a nearest neighbor search in descriptor space. This process is called feature matching.

Determining the own position within the environment requires some kind of internal representation or map. In its simplest form, such a representation consists of a database of images collected for a distinctive set of places. Localization can then be solved by a search for the database image which is closest to the image of the current location. To perform the matching efficiently the images are usually transformed to a lower dimensional representation, e.g. by extracting local visual features and storing them in a tree structure [135]. In topological maps the place representations are stored in nodes that are linked to neighboring places, which adds knowledge about the connectivity between places [23, 22, 102, 95]. The current estimate of the own position is a strong prior which allows to reduce the search space and consequently improves accuracy. Adding spatial information from ego motion estimates to the links between places allows to reconstruct the spatial layout of the environment and enhances navigation capabilities [3, 100]. The ability to recognize previously seen places, known as loop closure detection, is also required in other mapping systems as a means to re-localize after tracking failures or in the absence of sensor measurements. Loop closures allow to correct the current pose estimate and to reduce the uncertainty. An extensive overview of place recognition and topological mapping is given in [88].

Estimating the ego motion of a camera from a sequence of images is known as visual odometry [117]. Initially, the camera motion between two frames is recovered from the essential matrix which can be estimated from five feature correspondences [116]. Given the relative camera motion and the two image projections of a point, the 3D position of the point can be reconstructed by triangulation [55]. Subsequently, the camera motion is obtained from 3D-2D correspondences and the application of nonlinear optimization techniques which minimize the re-projection error. The quality of the estimated trajectory can be improved by jointly optimizing the pose of the camera as well as the sparse 3D scene structure applying bundle adjustment [157] over a local window of past frames. An extensive tutorial on feature based visual odometry is presented in [133, 45]. Another approach for camera motion estimation and 3D scene reconstruction is based on direct image alignment using dense information from all [115] or semi-dense information from high gradient pixels [36]. Based on the recent image and its corresponding inverse depth map the pose of the camera is estimated by finding the motion parameters generating a synthetic view that minimizes the photometric error w.r.t. the current image. Using monocular vision only, the scale of the estimated camera motion and scene depth is an arbitrary factor. The absolute scale can be recovered using additional sensors [118], knowledge about the size of a reference object [26] or the height of the camera when moving on the ground plane [141].

Although visual odometry is very precise for limited trajectory lengths, the estimate of the own position will inevitably diverge from the real one since small errors accumulate over time. This drift can only be corrected by relating current sensor measurements to a previously constructed map. By detecting a loop closure the deviation of the current estimate from the past one can be corrected and back-propagated along the trajectory. In addition to pose drift, monocular approaches also need to account for a drift in scale which is tackled in [147] by using similarity transformations to represent camera motion. The problem of incrementally building a map of the environment and at the same time determining the own position within this map is known as simultaneous localization and mapping (SLAM). To solve the SLAM problem there exist mainly three paradigms that will be briefly discussed in the following.

Extended Kalman Filter The work by Smith et al. [139] introduced the Extended Kalman Filter (EKF) formulation of the SLAM problem. The core principle is to represent the pose of the camera and the positions of map features as a joint probability distribution with a single state vector and a corresponding covariance matrix reflecting the uncertainties. Based on the current estimate the next pose is predicted using a motion model and the expected position of the map features is computed. Associating the measured features to the map features enables a correction of the estimate. The Kalman equations require a linear motion and measurement model in order to maintain a Gaussian distribution. This is achieved by linearizing the involved functions around the current mean. The first real-time capable monocular EKF-SLAM system was presented by Davison et al. [25, 26]. They estimated the full 3D pose of a hand-waved camera and 3D feature locations in an indoor environment assuming a constant velocity model. Since the complexity of updating the covariance matrix is quadratic in the number of features, the map size is limited to a few hundred features in practice. The authors of [20] employed a sub-mapping strategy to enable the application in larger scale outdoor environments. Estimating the depth of a feature requires at least two measurements from different viewpoints. In [26] feature initialization is delayed until the depth uncertainty is small enough. The authors of [107] instead used an inverse depth parametrization which allows to directly integrate new features so that they immediately contribute to improving the estimate. Despite its successful application in real-time visual SLAM there remain some issues with the EKF approach. Besides the computational scaling it can not represent a multi-modal distribution of the current state caused by ambiguous measurements. Falsely established data associations lead to a divergence of the estimate that can not be corrected afterwards. Furthermore, the required linearization introduces errors in the estimate.

Particle Filter A Rao-Blackwellized particle filter solution to the SLAM problem was first introduced in [105] and later improved in a follow up work [106]. The approach maintains a set of particles where each particle represents an estimate of the trajectory together with its own feature map. The map features of a single particle are represented by low dimensional EKFs, exploiting the fact that the positions of map features are conditionally independent given the trajectory. The complexity is logarithmic in the number of features, enabling the creation of maps containing thousands of features. In contrast to the EKF approach it is possible to accurately represent the state estimate as a multi-modal distribution. The process starts with the generation of random particles. A motion model is applied in order to predict the next position of the robot and the expected position of map features. After a data association step the map is updated and the agreement of predicted and measured feature positions is used to assign an importance weight to each particle. In the subsequent re-sampling step the importance weights are used to remove unlikely samples and to replace them by new ones. While the original work used range sensors, the particle filter approach was also successfully applied using monocular [32] and stereo cameras [137]. One problem is the determination of the particle set size, that is needed to accurately map a certain environment and to maintain a sufficiently diverse set over long trajectories.

Graph Optimization Most modern approaches formulate SLAM as a problem of pose-graph optimization [16]. The nodes in the graph correspond to camera poses or feature locations that are connected via edges representing spatial measurements from odometry and feature observations. The constructed graph is processed using nonlinear optimization (bundle adjustment) to find the spatial configuration of nodes that minimizes the measurement error. Although the graph formulation was first introduced in 1997 [89], it has only become popular in recent years with the introduction of efficient and robust techniques (e.g. [28, 52, 121]) and the publication of generic graph optimization frameworks (e.g. [1, 76, 63]). Klein and Murray [70] presented their Parallel Tracking and Mapping (PTAM) approach, a real-time capable Monocular SLAM system. They perform feature and pose tracking in one thread while the map optimization is performed on a subset of carefully selected keyframes in the background. Strasdat et al. [147] presented a keyframe-based method using similarity instead of rigid body transformations to deal with the inherent problem of scale drift in monocular SLAM. In [146, 148] they concluded that the performance of graph optimization methods is superior to probabilistic filtering approaches (EKF, particle filter) when the number of features is increased. Recently, the feature-based ORB-SLAM [109] and the semi-dense LSD-SLAM [35] have been demonstrated to enable precise localization and mapping in large scale environments using a single camera and running in real-time on the CPU.

A detailed introduction into probabilistic filtering for SLAM can be found in the tutorials from Durrant-Whyte and Bailey [31, 7] and the book from Thrun et al. [155]. A tutorial on graph-based SLAM can be found in [51]. A survey of visual SLAM methods is presented in [46]. The current state of the art and open challenges are discussed in [16].

Biologically Inspired Models Many animals have excellent localization and navigation capabilities and seem to be able to easily find their way to a food source or nest location even in difficult environmental conditions. Ants are assumed to combine path integration and image matching where the current scene view is compared to stored snapshots from specific locations in order to navigate in their natural habitat [160, 21]. In [77] the authors implemented a model of ant navigation on a real robot in a desert environment with artificial landmarks. Path integration is based on wheel odometry and global heading direction obtained from a polarized-light compass system. The compass direction was used to align the perceived panoramic view from the current location to the stored snapshot at a target location. Navigation was then performed by computing a homing vector based on the image matching. Due to their dichromatic vision with peak sensitivities in the ultraviolet and green range the authors of [124] suggest that ants might extract and store skyline information, i.e. the border between sky and none-sky regions to determine a homing direction. In [144] the authors present results from topological localization using binary images encoding sky/non-sky pixels as a representation for places along a 2km route.

In 1971 O'Keefe and Dostrovsky discovered the existence of place cells in the hippocampus of rats whose activity is highly correlated with the animal's location in the environment [120]. Several years later, neurons encoding the orientation of the rat, so called head-direction cells, have also been identified [154]. A computational model of place and head-direction cells was presented in [5]. Visual cues and path integration were combined in a Hebbian learning framework to create a population of place cells enabling a small robot to navigate within a 60 × 60 cm area with bar-coded walls. A similar approach was presented in [9] where individual places and their spatial relations were encoded in a topological map. The model was also able to learn and unlearn navigation actions towards specific goal locations. Experiments have been performed in an eight-arm radial maze and a single and double T-Maze with artificial visual cues on the walls.

The focus of the aforementioned models is rather on producing plausible animal navigation behavior than performance in robotic scenarios. However, another approach inspired from rat navigation, called RatSLAM [101], is also concerned with real world application scenarios. The pose is encoded by an activity packet in a 3D continuous attractor network with axes representing (x, y, φ), i.e. the pose of the robot. Self-motion cues and visual template matching inject energy into the network shifting the peak of activity. To enable the mapping of larger environments the model was extended by organizing unique combinations of local views and pose codes in a topological experience map. The map is optimized after loop closures using graph relaxation and enables the model to maintain a consistent spatial representation over extended periods of time. In [99] a 66 kilometer urban road network was successfully mapped with a single webcam.

A comparison of mapping and navigation principles from biology and robotics is given in [98].

Deep Learning The technological and methodical progress in recent years enabled the training of deep convolutional neural networks (CNNs) and led to major advancements in many fields of computer vision e.g. image classification [74, 56], object detection [128, 129] and image segmentation [50, 85]. The well established SLAM methods are focused on multiple view geometry as well as on probabilistic methods and optimization techniques. However, since SLAM systems are highly modular researchers tried to solve different parts of the SLAM pipeline using CNNs. In [29] the authors used a small CNN to extract image patch descriptors that are superior to handcrafted ones like SIFT [86] and SURF [10] in image classification tasks. The training data was generated by randomly sampling 32×32 patches and applying a family of transformations like translations, rotations and color adjustments. The set of transformed patch variants was declared as one class and the network was trained to discriminate between classes. The problem of feature matching, i.e. identifying the same patch across images, was approached in [163] by learning a similarity function with a CNN. Multiple architectures were trained with tuples of image patches representing either the same patch extracted from different images or dissimilar ones. In several feature matching experiments the best results were achieved using a two-channel architecture where the two patches are processed as a single image made of two channels. A model for joint end-to-end learning of dense scene depth and ego-motion from monocular images was presented in [168]. The synthesis of new views based on the scene depth and ego-motion is the basis of jointly training two CNNs for each task. The depth prediction network processes a single image and assigns a depth value to each pixel. The ego-motion network takes as input a sequence of images and outputs the Euler angles and translation vectors from each source view to a reference view. The depth and ego-motion estimates are then used to synthesize the subsequent view. The loss is defined as the sum of absolute differences between the pixel intensities of the real and the synthesized view. Evaluations on depth and ego-motion benchmarks demonstrated a performance comparable to state of the art methods. The authors of [153] integrated a CNN for pixel-wise depth prediction into a dense SLAM system. The predicted depth was fused with the depth values estimated by the SLAM system to improve accuracy in low texture/gradient image regions and under pure rotational movement which prevents geometric depth estimation due to the lack of a stereo baseline. A complete model for end-to-end regression from monocular images to camera poses coined Pose-Net was proposed in [68]. A CNN network pre-trained on a large scale image classification task is used to regress the 3D position and orientation of camera in previously explored scenes. The ground truth is generated using a feature based structure from motion (SfM) approach which is similar to SLAM approaches relying on pose-graph graph optimization previously introduced in this chapter. The network output is a 7-dimensional vector representing the 3D position and the orientation encoded as quaternion. The loss is defined as the Euclidean norm between the predicted and the ground truth pose with an additional scaling term to balance the influence of position and orientation errors. The localization error in the presented experiments is higher compared to feature based localization w.r.t. to the point cloud created from SfM. However, due to the large data set used for pre-training, the obtained convolutional features enabled localization under a range of varying appearances, e.g. daytime or weather, where the feature based approaches failed. In a follow up work [67], the authors extended their model by a fine-tuning step using a geometric loss function defined by the re-projection error of 3D scene points given the estimated pose. Although the localization accuracy improved over the base model it is still worse than a feature based approach. Currently end-to-end learning for camera localization does not achieve state-of-the-art performance. However, it is superior in terms of robustness w.r.t. appearance changes of the environment. A further advantage of the CNN pose regression is the fixed model size and interference time which are both independent from the size of the mapped environment. Considering the decades of research invested into SLAM algorithms and the recent emergence of end-to-end deep learning approaches, we will probably see further advancements in the future. In the short term, some of the stages in the classic SLAM pipeline might be replaced by learning methods.

2.2 Long-term Robustness

Appearance changes of the environment induce high visual diversity into images of the same place visited at different times. This poses a severe challenge for vision based localization and mapping methods. Therefore, different approaches towards long-term autonomy have been proposed recently.

Dynamic Maps Over time the appearance of the environment might undergo substantial changes in appearance so that a previously constructed map becomes obsolete. If the current sensor measurements are no longer coherent with the stored map data, localization will inevitably fail. In order to reflect changes in the environment the map can be updated by removing data which does no longer conform to the current environmental condition and adding new measurements. Instead of updating the sensor representation of a place, a map might also include multiple representations of the same place in different conditions. In [27] the authors create a topological map of the environment where each node represents a specific place together with a descriptor obtained from the corresponding sensor measurements. The descriptors are SURF-features [10] extracted from the images. A short-term and long-term memory structure is employed to deal with temporarily and structural changes in an indoor environment. Stable features are gradually moved from short-term to long-term memory to adapt the map to a changing environment. The capacity of the long-term memory is constrained by a forgetting mechanism which removes unused features. In a nine week indoor experiment an improvement was shown compared to using a static map representation. Selecting the right parameters for updating the long-term memory depends on the dynamics of the environment in order to find the right balance between stability and plasticity. Dynamic changes in indoor environments are addressed in [72]. The authors present a system based on stereo visual odometry and visual feature based place recognition to create multiple representations of the environment over time. The map is represented as a pose graph of keyframes where the nodes contain a feature representation which is used by the place recognition module. In case of odometry failures and for global localization the current sub-map is linked with a high uncertainty to the existing map. If the place recognition system detects a loop closure, the sub-map is linked to the existing map and the initial 'weak link' is removed. The update and deletion of nodes is designed to preserve diversity while at the same time limiting the maximum number of nodes. Since the approach relies on visual features for place recognition, the maintenance of a consistent map is only possible under slight appearance changes. A similar approach of Churchill et al. [19] is to build and maintain dynamic maps of the environment where the diversity in the appearance of the environment is captured by different visual experiences. A visual experience is a sequence of estimated poses and the corresponding visual features obtained with a stereo visual odometry system. Multiple localizer running in parallel try to match the current frame to existing experiences. In case the system fails to localize a new experience is created. The authors demonstrate localization and mapping in an outdoor environment at different day times and changing weather conditions over the course of three month. Since the approach requires the successful localization in previous experiences in order to link the current one to the existing map, it can only deal with gradual changes. Milford et al. presented an extension to their RatSLAM model to enable long-term navigation in a dynamic indoor environment over the course of two weeks [100]. The unique combinations of local views and pose codes from the continuous attractor network are defined as experiences which are organized in a graph like map that enables the model to maintain a consistent spatial representation over extended periods of time. Graph relaxation is used to correct the map after loop closure detections. If the robot visits a new place or the appearance a known place has changed a new experience is created. To prevent the map from growing indefinitely nodes from regions with a high density of experiences are deleted randomly.

Robust Representations Instead of adapting the map to changes in the environment another approach towards long-term autonomy is to transform the sensor measurements to robust or invariant representation which are less affected by appearance variations. Considering short timescales, changes in illumination are one of the main causes for the failure of a vision based localization system. Lighting invariance is tackled by several authors at different levels of the image processing pipeline. In [165] the exposure time of a camera is optimized using a gradient-based image quality metric which exploits the cameras' photometric response function. The authors demonstrate a superior performance in visual odometry tasks compared to the camera's built in auto-exposure control. In [93] the effects of shadows are mitigated by a transformation of the images to a shadow invariant representation where the pixel values are a function of the underlying material property. Mapping and localization is then performed in parallel with standard gray-scale and illumination invariant images.

Local visual features are broadly used in the context of visual SLAM. To some extent they are robust w.r.t. lighting, viewpoint and scale changes. However, due to illumination effects, cast shadows and dynamic objects visual features extracted from a reference frame can usually only be matched within a limited period of time and the number of true positive matches might decrease drastically even after a few hours [125]. The authors of [158] investigated the suitability of SIFT and SURF features for coarse topological image based localization in a long-term outdoor scenario. Their results from a nine month experiment have shown that a reliable localization is not possible using descriptor matching alone. Through the application of the epipolar constraint, which takes the geometric relation between matched features into account, they could reduce the number of false positives and achieved a successful localization in 85%-90% of the trials [159]. The authors of [65] improve the robustness of topological localization using visual word occurrences by only considering features that can be persistently tracked over several frames and storing their average. In [62] a certain track is traversed several times under different conditions while keeping track of feature occurrences per place. The statistics collected during the training runs allow to model the probability of feature visibility per place.

Some authors proposed learning approaches to obtain illumination invariant feature descriptors. In [18] features are tracked over a sequence of images from a time-lapse video featuring dynamic lighting conditions. Matching and non-matching pairs of image patches are discriminated by a contrastive cost function. Genetic optimization was used in [78, 79] to obtain an illumination invariant descriptor from a pool of elementary descriptor building blocks. Although the authors demonstrate superior performance with respect to standard feature matching, illumination invariance addresses only a part of possible appearance changes.

Instead of focusing on small image structures like corners, blobs or edges the authors of [94] propose to learn place specific detectors for broader image regions which likely correspond to physical objects like windows, trees or traffic signs. Provided with several images of the same place in different conditions they train a number of linear Support Vecor Machines (SVMs) per place to robustly detect distinctive elements in the scene. Odometry information between nodes in a topological map is used as a selection prior in order to choose the place specific SVMs. The authors demonstrate successful coarse metric localization under challenging appearance variations. However, their approach requires to select images of the same place from different runs for training the SVMs which might be hard to accomplish in the first place.

Approaches using features from a pre-trained deep Convolutional Neural Network (CNN) for robust place recognition have been proposed by several authors. Sünderhauf et al. [149] investigated the effectiveness of CNN features extracted from different layers of AlexNet [74]. They concluded that features from the third convolutional layer are highly robust w.r.t. appearance changes while features from higher layers are less dependent on the viewpoint. Using the CNN features as holistic image descriptor improved the place recognition performance over existing methods based on conventional visual features and sequence matching. Depending on the specific data set, either a network trained especially for semantic scene recognition [167] or a network trained for generic object recognition [74] performed best. In [150] they extended the approach to achieve condition and viewpoint invariance using CNN descriptors computed for distinctive image regions obtained by an object proposal method [169]. In [4] the authors propose a method which integrates a trainable Vector of Locally Aggregated Descriptors (VLAD) layer into a CNN. The VLAD vector aggregates the distances of quantized features to their nearest visual word from a code book. The network is trained with a ranking loss function on Google Street View Time Machine where images of the same place in different conditions can be obtained. The output of the VLAD layer is used as image descriptor and the place recognition is performed by a nearest neighbor search. The methods based on CNN features were proven to enable place recognition under challenging conditions providing coarse metric localization. However, the proposed methods have high demands for computational and memory resources which renders them unsuitable for the application on small mobile platforms.

Image Sequence Matching Milford et al. [102] demonstrated localization along one dimensional routes across difficult conditions with severe changes in appearance. The approach, named SeqSLAM, matches sequences of images rather than finding a single global best match. Matching is performed directly on the down-sampled, patch-normalized images. The holistic image matching over sequences restricts this approach to one dimensional traversals along a defined route without deviations in lateral position and assumes a constant velocity. Improvements to this approach were presented in [123]. The robustness is increased by blackening out the sky regions before matching the images. Instead of sampling at a fixed rate, the sampling of images along the trajectory is driven by distance measurements from odometry to deal with variable velocities. The tolerance w.r.t. lateral deviations is increased by matching images over a predefined range of offsets. Naseer et al. [112] use a dense grid of Histogram of Oriented Gradients (HOG) [24] as image descriptors. They build a data association graph that relates image sequences retrieved in different seasons and solve the visual place recognition problem by computing network flows in the association graph. In a follow up work they have demonstrated that the performance improves further when using features from pre-trained CNN as global image descriptors [111]. While the approaches demonstrate robust place recognition under severe appearance changes, the sequence matching and the assumption of similar viewpoints renders them impractical for localization in open field scenarios.

Appearance Change Prediction A place might look very different when it is observed in different conditions, e.g. when comparing its appearance in the morning and the afternoon or in summer and winter. Hence, when using global image descriptors for image comparison in different conditions the distance in descriptor space might become prohibitively large. Instead of directly matching images from different conditions some authors proposed to learn a mapping that allows to translate the appearance of a place from one condition to another. In [113, 114] the authors create a common vocabulary of corresponding visual words from aligned image streams captured in different seasons along the same route. The images from the current condition are segmented into visual words which are then translated to the target condition using the learned vocabulary. The authors demonstrated that sequence based place recognition (SeqSLAM) benefits from the appearance change prediction. Global illumination changes occurring over the course of a day are tackled in [87]. A linear regression model is trained with image pairs of the same place at different times of the day in order to learn the corresponding transformation. Results from their experiments show that the appearance change prediction yields a substantial performance improvement compared to direct image matching between different daytimes. In [84] the authors train coupled Generative Adversarial Networks to translate between images from different seasons. Although the methods have been shown to improve the localization performance, the identification and management of conditions has not been investigated so far.

2.3 Navigation

In order to execute tasks in a spatial environment a mobile robot needs to plan a viable path to a given target location and then execute this plan using appropriate motion commands and avoiding collisions with objects. These navigaton strategies have different levels of complexity ranging from reactive motion execution to path planning in metrical maps [96, 13].

Reactive techniques for collision avoidance can be carried out without having an environmental representation using only the currently available sensor measurements. The authors of [142] demonstrated a method based on optical flow [58], which is defined as the 2D displacement of every pixel between consecutive frames captured with a moving camera, to circumnavigate obstacles. Objects in the field of view create optical flow vectors occupying increasingly larger areas of the image when they are approached by the robot. In order to avoid collisions the magnitude of the optical flow was kept in balance between the left and right half of the image.

Navigation to a target location which is in the direct line of sight is known as visual homing. Since the difference between the image from a given target location and images from nearby locations increases smoothly over space, navigation can be performed by successively estimating the movement direction that minimizes the distance in image space [164, 104]. Navigation in larger environments with a restricted viewing area requires a representation containing several snapshots organized in a topological map. In [44] images from distinct places have been stored as nodes in a topological map where the links between nodes represent their adjacency relationships. The planning of a global path was implemented using a graph search algorithm. A visual homing method based on feature correspondences was used to navigate between nodes. A similar approach using omnidirectional vision was presented in [14].

Graph search techniques like A* [54] are also used to plan trajectories in occupancy grid maps where the environment is discretized into equally sized cells with an assigned probability of being occupied by an obstacle. They are usually generated using range sensors like stereo vision [34, 110]. For navigation in grid or topological maps A* is guaranteed to find the optimal path given an admissible distance heuristic. However, it is memory and computationally intensive for large environments with many obstacles. During the path execution deviations caused by sensor measurements have to be detected and corrected. If the deviations become too large a re-planning step has to be initiated.

Instead of finding a path from the current to a target location one can create a universal plan, which assigns a motion command to every position in the environment leading the robot to a specified target. The authors of [5] created such a universal plan to implement navigation in their biomimetic model of place cells. They assigned a reward to the target location and used reinforcement learning to obtain a policy which selects the motion command with the highest expected reward in response to an input from the place cell network. However, the required additional learning phase with random explorations of the environment might not be feasible in real world application scenarios.

Another approach for navigation in metrical space is the potential field method that is based on gradient descent in a vector force field defined by an attractor at the target position and repulsive forces from obstacles [69, 8]. It is an elegant formulation of the navigation problem, however, a known limitation of the approach are local minima caused by certain types of obstacles or their spatial configuration [156]. By designing an optimal navigation function having a global minimum this problem can be avoided [30]. However, determining such a function is only feasible for small environments with a low complexity [96].

The feature-based maps introduced in a previous section allow to precisely localize a mobile robot and accurately model the sparse scene structure while being memory efficient. However, since the absence of a feature does not necessarily imply free space, e.g. a low-textured wall might not be represented in the map, these maps are not optimal in terms of path planning and navigation [46]. A general review of mapping and navigation strategies can be found in [96] and with a focus on vision based techniques in [13].

3 Unsupervised Learning of Spatial Representations

This chapter introduces a model based on unsupervised slowness learning that enables a mobile robot to extract a spatial representation of the environment directly from the visual input captured during an exploration phase. The resulting representation encodes the position of the robot as a set of slowly varying features that are invariant w.r.t. its specific orientation. The intuition behind the principle of slowness learning is given in section 3.1. Slow Feature Analysis (SFA), the concrete algorithm that is used in this work, is discussed in section 3.2. It has been shown in previous work that a hierarchical, converging SFA network can model the activity of cells in a rat's brain that form a neural representation of its spatial attributes by directly processing the views from a virtual rat [42]. The model learns either representations of the position or the orientation depending on the movement statistics during the unsupervised learning process. This hierarchical SFA network for spatial cell learning is the basis for this work and is presented in section 3.3. The specific network architecture and a training scheme for learning orientation invariant representations of the position is described in section 3.4. The methods for analyzing the learned slow feature representations are detailed in section 3.5.

3.1 Principle of Slowness Learning

Extracting relevant information from received sensory signals is an important prerequisite to interact with the environment. When we visually perceive a scene our brain is able to extract a high level representation from the raw visual sensory signals it receives. If an object passes our field of view the stimuli of a single receptor in the retina may change very rapidly, while the high level information (what objects are present, and where are they located) usually changes on a much slower timescale. Since the reconstruction of relevant information from the received signal is not directly coupled to a feedback or supervision signal it is assumed to be guided by statistical regularities in the input data. One of these regularities is the difference in the timescales of the quickly varying stimuli and the slowly varying high level representation. This leads to the assumption that slowness is a general learning objective in the brain. If the relevant information is expected to change slowly it should be possible to recover it by extracting slowly varying features that are embedded in the raw visual stimuli. The resulting learning principle does not rely on external supervision signals, i.e. it is unsupervised, and thus only depends on the statistics of the training data. Although slowness learning is concerned with identifying slowly varying signals the extraction of these signals needs to be instantaneous in order to adequately react to relevant events.

A well known approach for unsupervised learning is Principal Component Analysis (PCA). It finds a rotated coordinate system such that the dimensions of the data in the new coordinate system are de-correlated. Furthermore, it sorts the eigenvectors, which form the new basis vectors, in descending order according to the corresponding eigenvalues. Hence, PCA is often used for dimensionality reduction by discarding dimension with low variance. In contrast to unsupervised slowness learning, the temporal order of the data samples is irrelevant to PCA. Therefore, PCA yields the same result for different permutations of the data. However, the temporal structure of the data often contains useful information and one might want to obtain similar outputs for temporally close input samples. Measures of similarity or temporal stability constitute the basis for slowness learning methods [41, 143, 73].

3.2 Slow Feature Analysis

Slow Feature Analysis (SFA) as introduced in [161, 162] is the slowness learning method used in this thesis. SFA solves the learning problem of finding instantaneous scalar input-output functions $g_j(\boldsymbol{x})$ that transform a multidimensional time series $\boldsymbol{x}(t)$, in our case images along a trajectory, to slowly varying output signals such that the signals

$$s_j(t) := g_j(\boldsymbol{x}(t))$$

minimize

$$\Delta(s_j) := \langle \dot{s}_j^2 \rangle_t$$

under the constraints

$$\begin{aligned} \langle s_j \rangle_t &= 0 \text{ (zero mean)}, \\ \langle s_j^2 \rangle_t &= 1 \text{ (unit variance)}, \\ \forall i < j : \langle s_i s_j \rangle_t &= 0 \text{ (decorrelation and order)} \end{aligned}$$

with $\langle \cdot \rangle_t$ and $\dot{s}$ indicating temporal averaging and the derivative of s, respectively. The Δ-value is a measure of the temporal slowness of the signal $s_j(t)$. It is given by the mean

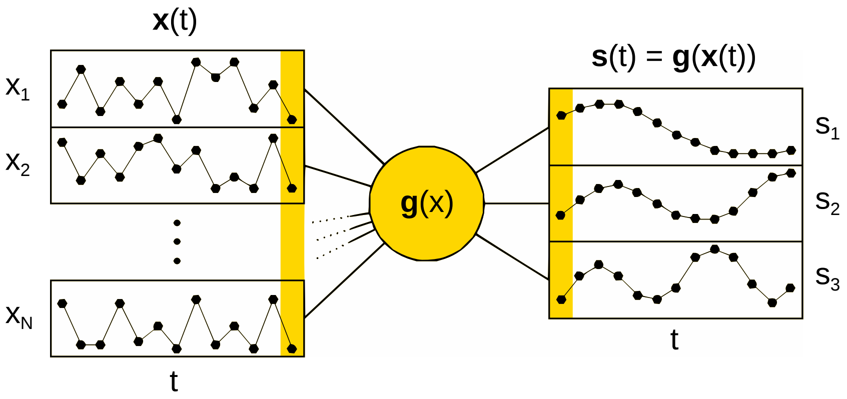


Figure 3.1: Illustration of the optimization problem solved by SFA. SFA finds functions $g(x)$ that transform a time varying multidimensional input signal $x(t)$ to output signals $s(t) = g(x(t))$ that vary as slow as possible. Once the training is finished slow features are computed instantaneously from a single snapshot of the input signal. Adapted from Figure 1 in `http://www.scholarpedia.org/article/Slow_feature_analysis`.

of the signal's squared temporal derivative, so small Δ-values indicate slowly varying signals. The constraints avoid the trivial constant solution that is maximally slow but carries no information and ensure that different functions g code for different aspects of the input. Furthermore, slow features s are required to be instantaneous outputs of functions g so that slowly varying signals can not be obtained by temporal filtering. The optimization problem solved by SFA is illustrated in Fig. 3.1. If one considers a finite function space, e.g. all polynomials of a degree two, SFA can be implemented by performing the following sequence of steps:

- First, the data is expanded into the non-linear space that is considered for the given problem, e.g. all polynomials of degree two.

- Subtracting the sample mean centers the expanded data points and satisfies the zero mean constraint.

- Applying PCA to the covariance matrix of the expanded and centered data points yields a set of eigenvectors which are the basis of a new coordinate system where the dimensions are de-correlated. The data points are normalized by projecting them on the set of eigenvectors and dividing by the square root of the corresponding eigenvalues.

- The temporal variation is measured on the normalized data points by approximating the temporal derivatives with the differences between consecutive data points. Applying another PCA to the covariance matrix of the temporal derivatives and projecting the data on the axes with the smallest variance yields the slow features.

The function $g(x)$ is represented by the sequence of all steps. A closed form solution of SFA based on solving a generalized eigenvalue problem was presented in [11]. The implementation of SFA that is used in this work is part of the Modular toolkit for Data Processing (MDP) [170].

SFA originates from the field of computational neuroscience and has been used to model complex cells in the primate visual system [11]. However, it was also applied in a number of technical applications, like human action recognition [166], monocular road segmentation [75] or object recognition and pose estimation [43] to extract invariant features or to obtain low dimensional and meaningful representations from the raw input data.

Since most problems of interest are non-linear the data is usually expanded into the considered function space (e.g. all polynomials of degree $2 - 3$). Due to the non-linear expansion SFA becomes impractical for high dimensional data as the complexity is cubic in the number of dimensions. In order to efficiently process high dimensional data SFA can be applied iteratively in a hierarchical converging network. The input data is partitioned into small blocks which serve as input to distinct SFA nodes in the input layer. Blocks of locally learned SFA-outputs from these nodes are then fed as inputs to the next layer of SFA nodes. A limitation of the number of SFA-outputs that are passed to the next layer and the block-processing reduce the overall dimensionality with every layer. At some point, global SFA becomes feasible with a single node that effectively perceives the whole input data. Although the hierarchical processing does not guarantee to find the globally optimal solution it has been proven to yield feasible results in many practical applications [37].

3.3 Model for the Formation of Place and Head-Direction Cells

Cells in the hippocampus of rodents have been discovered that form a neural representation of the animal's spatial attributes like its position in space or its head-direction. Place cells fire whenever the animal is in a particular location and are independent from its orientation [120]. Head-direction cells on the other hand are invariant with respect to the spatial position and are only sensitive to the orientation of the animal [154]. Franzius et al. [42] introduced a model consisting of multiple, converging layers of SFA-nodes that is capable of extracting spatial information directly from the raw visual stimuli of a virtual rat. The last node in the network performs sparse coding and produces responses similar to those of place and head-direction cells. Experiments were performed in a rectangular simulator environment with textured walls. The model was trained with the 320° views of the rat that were captured during a random exploration of the environment following Brownian motion with different ratios of translational and rotational velocities. It has been shown that the type of spatial cells that develop only depends on the movement statistics of the virtual rat during the training phase. For low a translational speed and quick head movements the resulting SFA-outputs are invariant with respect to the orientation and only code for the position of the rat. Slow head movement and fast translational speed results in functions that are position invariant and code for the head-direction. They also introduced an analytical method to determine the theoretically optimal solutions under the constraints that the environment is kept unchanged for the duration of the experiment. Having knowledge about the spatial configuration of the rat, defined by its position and head-direction (x, y, φ), the corresponding view can be determined. The same applies for the views that determine the exact configuration of the rat if the environment is diverse enough. This leads to the simplified problem of performing SFA on the low dimensional configuration space instead of the high dimensional views. In this case it becomes feasible to compute the optimal solution for SFA analytically. For a rectangular shaped training area the derived optimal output functions encode the position on the coordinate axes and the orientation of the robot as standing cosine-/sine waves.

3.4 Model Architecture and Training

3.4.1 Orientation Invariance

For the scenario of a robustly self-localizing and navigating mobile robot, we want to find functions that encode the robot's position on the x- and y-axis as slowly varying features and are invariant with respect to its orientation. As stated in the previous section, learned slow features strongly depend on the movement statistics of the mobile robot during the training phase. In order to achieve orientation invariance, the orientation of the robot has to change on a faster timescale than its position. A constantly rotating robot with a fixed camera is inconvenient to drive, and a robot with a rotating camera is undesirable for mechanical stability and simplicity. As an alternative, we use an omnidirectional imaging system which allows to easily add simulated rotational movement of the robot to manipulate movement statistics. Thus, the model is able to find orientation invariant representations of its own position without having to rotate the camera or the robot physically. During the training phase we simulate a full rotation for every captured image. Since for panoramic images a lateral shift is equivalent to a rotation around the yaw axis we can simulate a full rotation by shifting a sliding window over the periodic panoramic views (see Fig. 3.2 for an illustration). Throughout the experiments we use a window equal to 100% of the image size so that each rotated view contains the whole image, incrementally shifted along the lateral direction. Please note that achieving orientation invariance is a non-trivial task even when using a 100% window. An analysis of using windows of various sizes will be given in section 5.1.3.

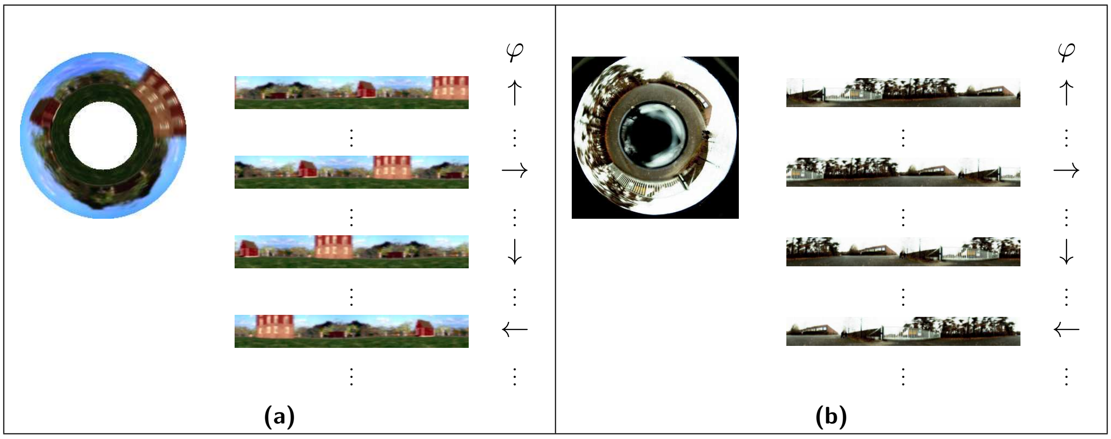


Figure 3.2: Simulated rotation for (a) simulator and (b) real world experiments. The circular image of the surrounding is transformed to a panoramic view with periodic boundaries. Rotation is simulated for every view from one location by laterally sliding a window over the panoramic image with increments of 5 pixels. Thus the variable φ denotes the relative orientation w.r.t. the robot's global orientation. Arrows indicate a relative orientation of 0°, 90°, 180° and 270°.

3.4.2 Network Architecture and Training

As input image dimensionality is too high to learn slow features in a single step, we employ a hierarchical, converging network similar to [42]. Instead of applying a final sparse coding step the SFA-outputs of the final node will be used directly as spatial representation. The network is made of several layers, each consisting of multiple SFA-nodes arranged on a regular grid. Each node performs a sequence of steps: linear SFA for dimensionality reduction, quadratic expansion of the reduced signals, and another SFA step for slow feature extraction. The nodes in the lowest layer process overlapping patches of 10×10 image pixels and are positioned every 5 pixels. In the lower layers the number of nodes and their dimensionality depends on the concrete setting, but dimensionality is chosen to be a maximum of 300 for numerical stability. The region of the input data visible to a node increases with every subsequent layer. The highest layer contains a single node, whose first (i.e. slowest) n outputs $s_{1...n}$ we use as environmental representation and which we call SFA-outputs.

The layers are trained subsequently with all temporally ordered training images. A full rotation is simulated for every panoramic image by incrementally shifting it laterally by 5 pixels. For panoramic images a rotation on the spot around the yaw axis is equivalent to a lateral shift of the image. Instead of training each node individually, a single node per layer is trained with stimuli from all node locations in its layer and replicated throughout the layer after training. This technique is similar to weight sharing in Neural Networks. Note that this design is chosen only for its computational efficiency and that network performance increases for individually learned nodes [42]. After the training the

n slowest SFA-outputs $s_{1...n}$ are the orientation invariant encoding of the robot's location and are computed instantaneously from a single image. The stated model parameters are in accordance with the originally proposed model. The concrete values have been slightly adapted in the experiments to account for different image resolutions. However, the model has been shown to be robust under a range of parameter settings for image resolution, number of layers, receptive field size and overlap [42]. An illustration of the model is given in Fig. 3.3.

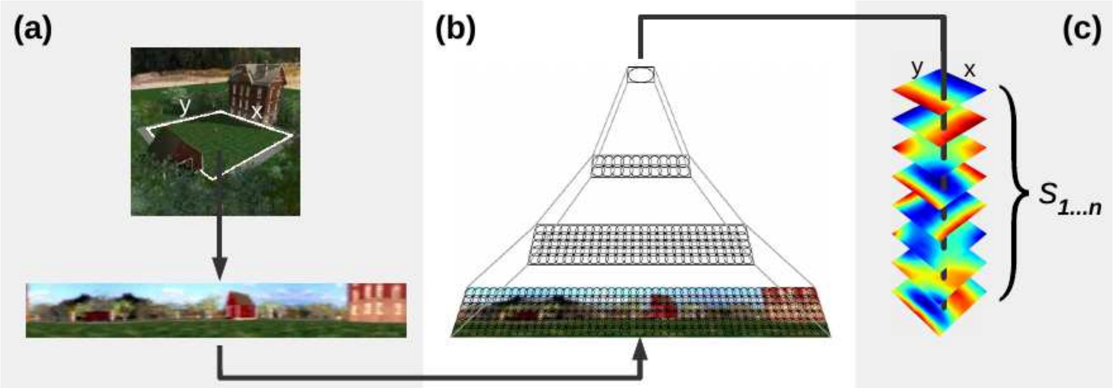


Figure 3.3: Model architecture. (a) The robot's view associated with a certain position $\boldsymbol{p} := (x, y)$ is steadily captured and transformed to a panoramic view. (b) The view is processed by the four layer network where each node in the network performs linear SFA for dimensionality reduction followed by a quadratic SFA for slow feature extraction. (c) The n slowest SFA-outputs $s_{1...n}$ over all positions $\boldsymbol{p}$. The color coded outputs, so-called spatial firing maps, ideally show characteristic gradients along the coordinate axes and look the same independent of the specific orientation. Thus, SFA-outputs $s_{1...n}$ at position $\boldsymbol{p}$ are the orientation invariant encoding of location.

3.5 Analysis of the Learned Representations

For the task of self-localization and navigation the learned SFA representations ideally code for the position of the robot and are orientation invariant. According to [42], the sensitivity of an SFA-output function $s_j, j = 1...n$ to the spatial position $\boldsymbol{p} := (x, y)$ is characterized by its mean positional variance $\eta_{\boldsymbol{p}}$ over all orientations φ: $\eta_{\boldsymbol{p}} = \langle \text{var}_{\boldsymbol{p}}(s(\boldsymbol{p}, \varphi))\rangle_{\varphi}$. Similarly, the sensitivity to the orientation φ is characterized by its mean orientation variance η_{φ} over all positions $\boldsymbol{p}$: $\eta_{\varphi} = \langle \text{var}_{\varphi}(s(\boldsymbol{p}, \varphi))\rangle_{\boldsymbol{p}}$. In the ideal case $\eta_{\boldsymbol{p}} = 1$ and $\eta_{\varphi} = 0$, if a function only codes for the robot's position on the x-and y-axis and is completely orientation invariant. The spatial information encoded by an SFA-output will be visualized by two dimensional spatial firing maps (see Fig. 3.3c). They illustrate the color-coded SFA-output value for every position $\boldsymbol{p} := (x, y)$. An output which codes for the position on a certain axis ideally produces a map that shows a color gradient along this axis. If the SFA-outputs are perfectly orientation invariant the gradients should be clearly visible regardless of the specific orientation.

To perform a quantitative metric evaluation of the learned SFA-representation we compute a regression function from the quadratically expanded slow feature outputs to the metric ground truth positions from a training run. The obtained mapping from slow feature to metric space will then be used to evaluate the localization accuracy on a separate test run and to determine the distance to a given target location in the navigation experiments. Please note that the ground truth coordinates are only used for evaluation purposes and that the slow feature representations are learned using visual input only.

4 Data Recording and Ground Truth Acquisition

This chapter describes the procedures for generating the data that was used to evaluate the introduced methods in simulator and real world experiments. For reasons of simplicity and the benefit of a static environment and full control over the configuration space, a first validation of the approaches was conducted in simulated environments. Section 4.1 presents the simulator environments and the process for data generation and recording. A quantitative metric evaluation of the learned slow feature representation requires knowledge of the robot's true position within the environment. In contrast to the simulator this ground truth information is not directly available and therefore has to be monitored by an external system. A method for ground truth data acquisition based on optical marker detection is detailed in section 4.2.1. The experimental platforms and the data generation procedures for the real world experiments are described in section 4.2.2.

4.1 Data Generation in the Simulator

Artificial data generated with a simulator is used in various experiments presented in this thesis to validate the introduced methods in a fully controllable setting. The simulator used in the experiments presented in section 5.1.1 was based on existing software available at the Honda Research Institute. The virtual environment is made of green area, trees and some houses and resembles a park or a garden. Images have been rendered once at discretized positions forming a regular grid of 30×30 units. From every position the view of the virtual camera was mapped to a conic mirror to construct an omnidirectional image. The movement trajectory of the training and test runs was constructed afterwards by arranging the images and the corresponding coordinates to a continuous walk.

For reasons of greater flexibility and to achieve a higher quality of the rendered images we used the 3D software Blender[1] and its Python API to generate data for the further

[1] `https://www.blender.org/`

simulator experiments[2]. The garden-like environment was created by randomly drawing from a pre-defined set of suitable 3D objects. The objects were placed on a textured ground plane at non-overlapping positions defined by randomly chosen polar coordinates with radii from a certain range. The area within the minimum radius defines the space where the virtual robot can freely move. The ground plane and the objects are enclosed by a spherical textured object to mimic a horizon and sky. The omnidirectional camera was created from a virtual camera pointing at an ellipsoid with a reflecting texture. Illustrations of the simulator environments and rendered images are shown in the corresponding experiment sections.

4.2 Data Generation in the Real World

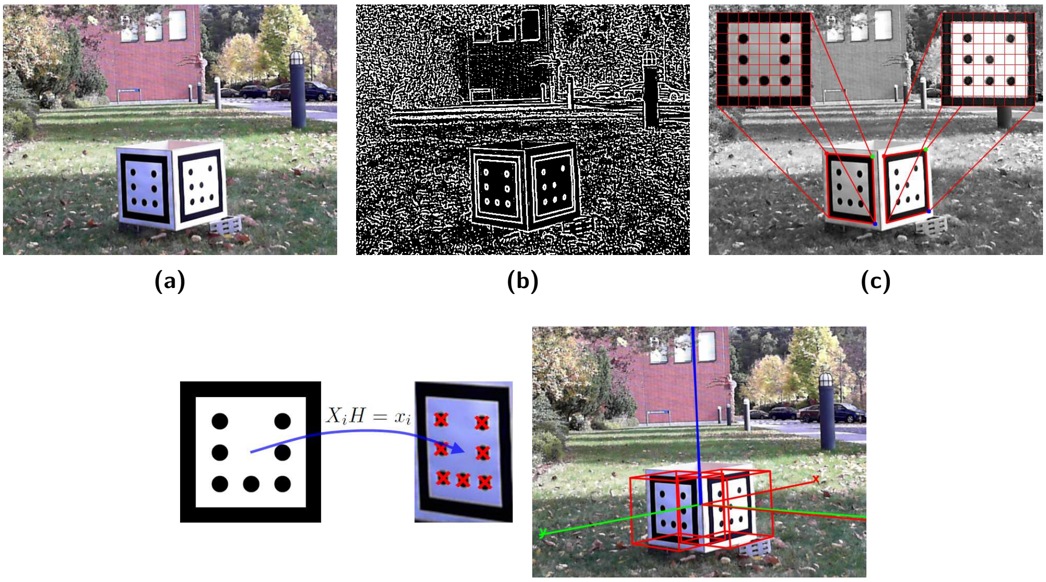
4.2.1 Ground Truth Acquisition

To asses the quality of the learned slow feature representations in a metric way the true position of the robot needs to be assigned to the corresponding images. While this ground truth data is directly available in the simulator environments it has to be acquired using an appropriate method in the experiments. Using the odometry, i.e. the integrated ego-motion estimates computed from the readings of the internal wheel encoders, is not feasible since small errors in the estimates accumulate over time. For indoor applications several approaches based on sensors mounted on the room ceiling have been proposed (e.g. [138]), but these approaches turned out to be unfeasible for outdoor applications. To keep ground truth acquisition flexible and robust we mounted a 30 cm cube on the robot with optical, binary markers attached to its facets (Fig. 4.2) and used an external monitoring system for optical marker detection and pose estimation. The basis for the software is the Aruco-library [47]. We adapted the marker design as well as the marker detection and pose estimation procedures in order to meet our requirements regarding robustness and accuracy. An initial pose estimate for a detected marker is obtained by estimating and decomposing the homography matrix [90] that projects a marker's 3D features from the $z = 0$ plane to the corresponding 2D image features identified in the current frame. The pose is then further refined using non-linear optimization (Levenberg-Marquardt [91]) which minimizes the re-projection error. The detection and pose estimation process is illustrated in Fig. 4.1. Since the estimated 3D poses of the detected markers are defined in the coordinate system of the camera, they might lie on a plane that is rotated w.r.t. the ground plane the robot is moving in. Therefore, we applied Principal Component Analysis (PCA) to the estimated marker coordinates and projected them on the two axes with largest variance in advance of further processing steps.

In an experimental setup with a high resolution camera the method provided a detection

[2] Thanks to Marius Anderie for contributing to the implementation during his Bachelor Thesis.

up to a distance of 18 meters with a mean Euclidean deviation of 3.4 cm, as manually verified by laser distance meter.



(a) (b) (c)

(d) (e)

Figure 4.1: Illustration of the marker detection and pose estimation process. (a) The original view of the input image. (b) Result of an adaptive threshold operation applied to the grayscale image. (c) Contours of possible marker candidates are warped to a frontal view to verify the binary encoded marker ID. (d) Computation of the homography that maps the 3D marker points to the corresponding 2D image points. (e) The marker poses are extracted from the homography and then refined using non-linear optimization.

4.2.2 Data Recording

The mobile robot used in the real world experiments is a Pioneer 3AT equipped with an omnidirectional imaging system on top (see Fig. 4.2). In the experiments different imaging system have been used. In the first real world experiment in section 5.1.2 the omnidirectional imaging system is made of a camera pointing at a chrome-colored plastic ellipsoid. For the experiments in section 5.2 where the performance of the SFA-model is compared to other methods a high quality omnidirectional vision system[3] has been used. The trajectories were driven manually using a wireless joypad. Two notebooks with synchronized clocks were used to collect the data during the experiments. One notebook, which was placed on the robot, saved the camera images together with the current timestamp and converted the signals received from the joypad to the correspond-

[3] https://www.0-360.com/

ing motion commands. The second notebook was used to run the software for ground truth data acquisition based on the optical marker detection method described in the previous section. The pose of the robot was measured throughout the experiments and saved together with the current time stamp. In a post-processing step each image was assigned with a ground truth position using linear interpolation based on the timestamps. For the long-term experiments in section 6.2.2 we used a different robot platform equipped with a spherical lens camera[4]. The robot is able to autonomously follow a certain closed loop trajectory defined by a border wire. Since the robot begins and terminates operation in a base station the exact position and orientation in the beginning and the end of the closed loop trajectory are known. This allows to detect accumulated errors and to correct the estimated trajectory by distributing the weighted error backwards along the trajectory. Therefore, the robot's trajectory can be precisely reconstructed using wheel odometry and a gyroscope [33]. The resulting estimated trajectory is considered as ground truth information which is saved together with the current image to an attached storage device.
Example images from the different camera systems and specific details regarding the data recording will be given in the sections covering the respective experiments.

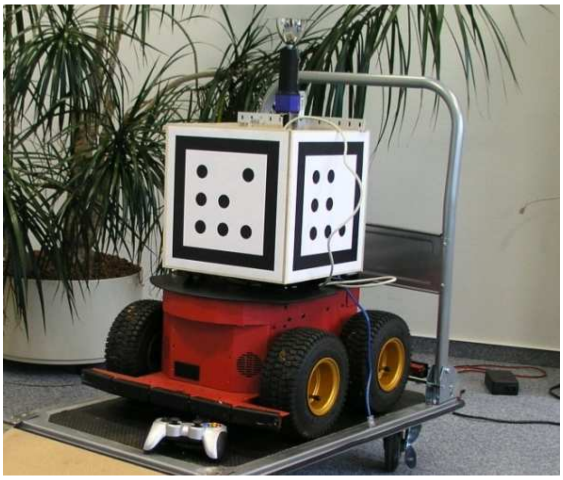


Figure 4.2: Pioneer 3AT equipped with an omnidirectional vision system and the marker-box for ground truth data acquisition.

[4] https://kodakpixpro.com/Americas/cameras/actioncam/sp360/

5 Self-localization

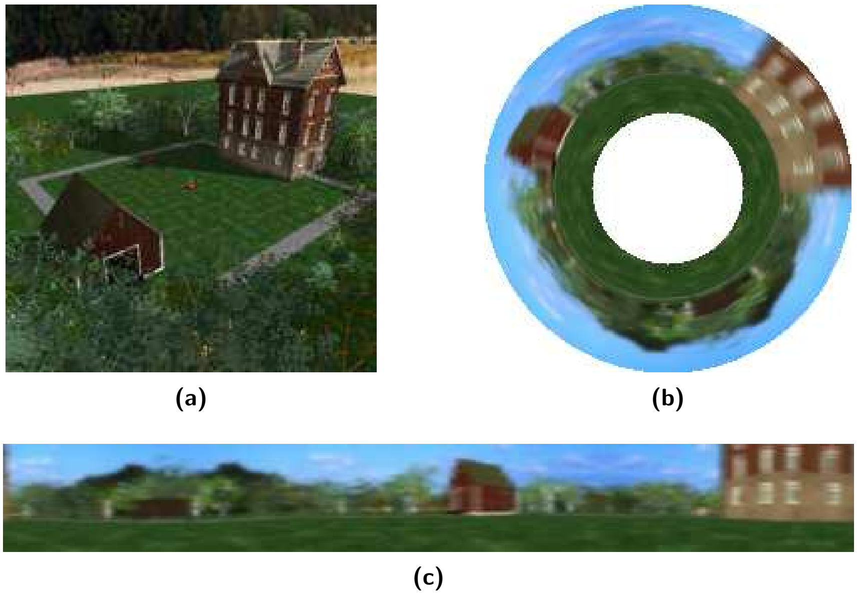
Slow Feature Analysis applied to the temporal sequence of visual input from a mobile robot during exploration of a certain environment yields representations of the robot's position or orientation depending on the movement statistics. Using an omnidirectional camera allows to manipulate the perceived image statistics by simulating a full rotation for every captured image. After the unsupervised learning phase the resulting SFA functions ideally code for the position on the x- and y-axis and are invariant with respect to the orientation of the robot. In order to be useful for higher level tasks such as navigating to a certain location in the environment the quality of the learned representation has to be adequate. To quantify and visualize the encoded spatial information of the SFA-outputs in a metric way we compute a regression function from the SFA-outputs from a training run to the metric ground truth positions and subsequently apply it to SFA-outputs from a separate test run. The quality of the learned representations can then be assessed quantitatively by performing a self-localization task and measuring the metric accuracy w.r.t. the ground truth. Additionally, it allows to compute the sensitivity of the SFA functions to the spatial position $\boldsymbol{p} := (x, y)$ given by the mean positional variance $\eta_{\boldsymbol{p}}$ and the sensitivity to the orientation φ, characterized by its mean orientation variance η_φ. Qualitative information about the learned slow feature representations can be obtained by plotting the individual color coded SFA-outputs over every position in the training area. Ideally, these so called spatial firing maps show orthogonal gradients along the coordinate axis for the first two outputs.

In the first section the SFA-model is validated by applying it in a simulator and a real world experiment. The resulting SFA representations are analyzed w.r.t. the quality of the spatial coding and their orientation invariance. Section 5.2 compares the localization accuracy of the SFA-model to state of the art visual simultaneous localization and mapping (SLAM) methods in further indoor and outdoor environments. While the SFA-model estimates an absolute position from a single image other approaches usually incorporate ego motion information and incrementally build up their belief of the own position. In section 5.3 we present a method to combine odometry information with the SFA estimates in probabilistic filter. Therefore, we propose an unsupervised learning approach to obtain the mapping from slow feature outputs to metric coordinates by imposing constraints on the trajectory and using odometry measurements. In the last section an alternative model for SFA-localization is presented which learns spatial representations from single or multiple tracked landmark views [1].

5.1 Validation of the Approach

5.1.1 Localization in a Simulated Environment

The model for SFA localization was first applied in a virtual reality simulator to validate the model under entirely controllable settings and to present an analysis of the spatial encoding resulting from optimal conditions. The virtual robot was placed on discrete positions forming a regular 30×30 grid. We recorded 624 omnidirectional RGB images for the training set and 196 for the test set and transformed them to panoramic views with a resolution of 350×40 pixel. Figure 5.1 shows the simulated environment from a top view and an example of an omnidirectional as well as a panoramic image. The network architecture, defined by the number of layers, the arrangement of the receptive fields (RF) and their dimensionality, is given in Table 5.1.



(a)

(b)

(c)

Figure 5.1: Simulator environment. (a) A top view of the simulator environment. (b) A rendered omnidirectional image. (c) Corresponding panoramic image.

Results

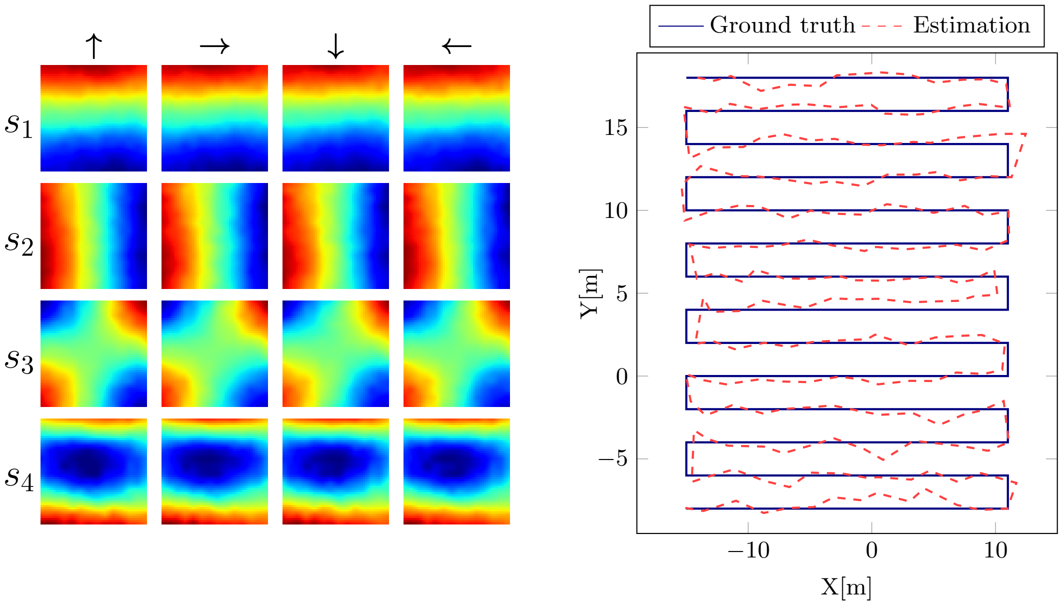
All resulting SFA-outputs have a high spatial structure and are almost completely orientation invariant as their outputs for the training views have a mean positional variance

[1] Thanks to Benjamin Löffler for the contributions made during his Master Thesis.

Layer	Number of RFs (w×h)	RF size (w×h)	Stride (w×h)	Input dim	Output dim
1	69 × 7	10 × 10	5 × 5	300	14
2	22 × 3	6 × 3	3 × 2	252	16
3	10 × 1	4 × 3	2 × 1	192	16
4	1 × 1	1 × 1	1 × 1	160	8

Table 5.1: Network parameters for the simulator experiment. Number of Receptive Fields (RF) per layer, RF size and stride are given for every layer of the SFA network.

$\eta_p \approx 1$ and the mean orientation variance η_φ ranges from 0.00 (s_1) to 0.17 (s_8). This is also reflected in the spatial firing maps in Fig. 5.2a which show an obvious encoding for the position on the coordinate axes and look nearly identical under different orientations. These results are very similar to the theoretically predicted optimal SFA solutions given in [42]. Since here in the simulator, the views of the training- and test-run are identical for the same location we only use the test data for the regression analysis. Random



(a) (b)

Figure 5.2: Results for the simulated environment. (a) *Spatial firing maps* of the four slowest SFA-outputs $s_{1...4}$ for relative orientations $0°, 90°, 180°$ and $270°$. Obviously, the first and second outputs are spatially orthogonal, coding for y- and x-position, respectively. Output values are monotonically increasing from north to south and east to west. The third function is a mixture of the first two functions and function four is a higher oscillating representation of the first one. (b) Ground truth and estimated coordinates computed by the regression. Estimations are averaged over the windows of the simulated rotation for one location.

50/50 splits are used to train the regression and evaluate the coordinate prediction. Repeating it 100 times results in an overall mean absolute error (MAE) for the x- and y-coordinate estimation of 1.83% and 1.68%, relative to the coordinate range of the test run (Fig. 5.2b). The number of slow feature outputs used for the experiments has been chosen based on the analysis of the training and test error by varying the number from 2 to 12. The minimum test error is obtained for 8 slow feature outputs. The behavior of the training and test error curves is shown in Fig. 5.3.

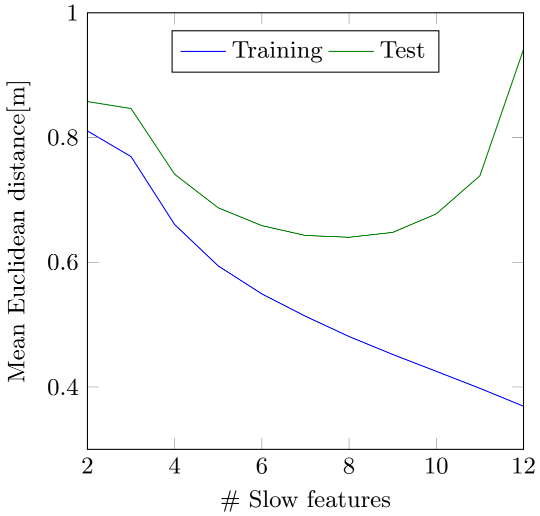
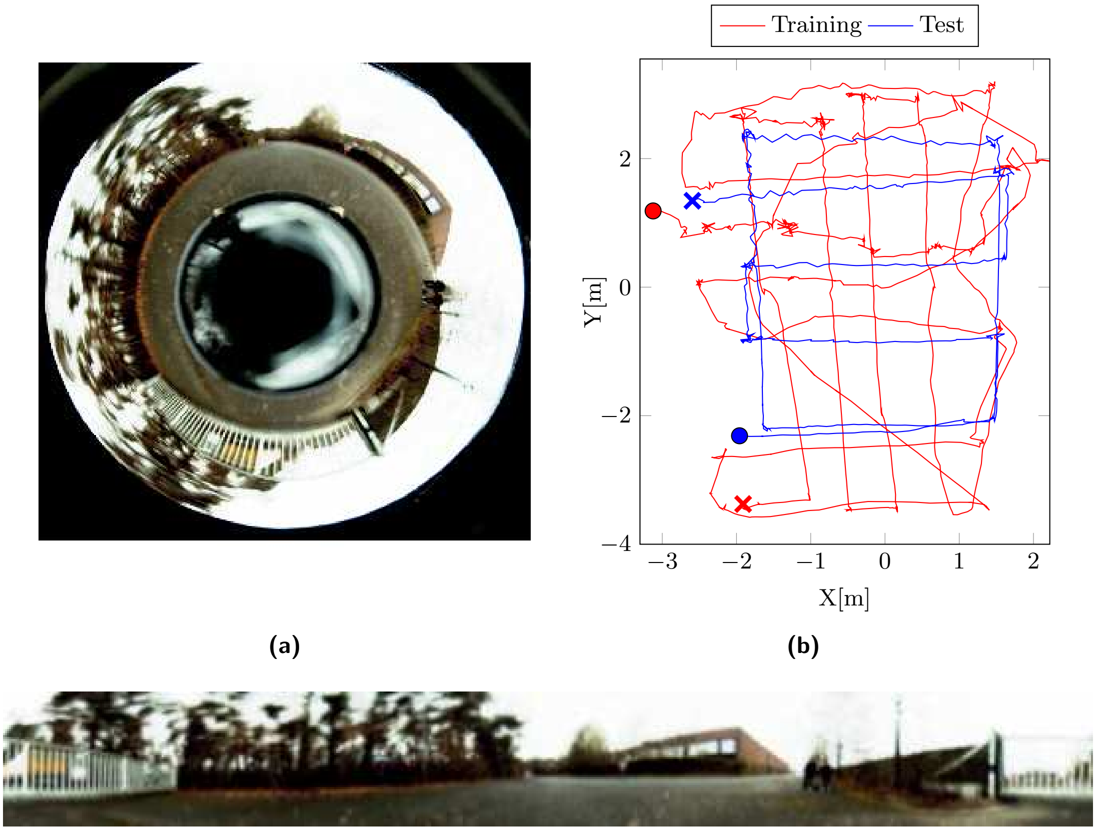


Figure 5.3: Training and test error for a varying number of slow feature outputs. The error is given as the mean Euclidean distance from ground truth. Up to 8 slow feature outputs the training as well as the test error are decreasing. Using more slow feature outputs results in overfitting the training data.

5.1.2 Localization in a Real World Environment

The experiment was transferred to an outdoor scenario to examine how the model copes with real-world conditions like a non-static environment, changing light conditions and noisy sensor readings. Outdoor experiments were performed within an area of approximately 5 × 7 meters on asphalted ground. Test data was recorded directly after the training data. The training and test sets consist of 5900 and 2800 RGB panoramic images with a resolution of 600 × 60 pixels. During the training and the test phase the robot was manually moved with a wireless joystick at a maximum velocity of 40 cm/s in a grid like trajectory so that the translations along the x- and y-axis were fairly equal distributed with respect to the traveled distance (Fig. 5.4b). The parameters of the SFA-network are given in Table 5.2.



(a)

(b)

(c)

Figure 5.4: Example images and trajectories of the experiment (a) Image from the omnidirectional camera mounted on top of the robot. (b) Trajectory of the training- and test-run. Start and end points are marked by a cross and a circle, respectively. (c) Panoramic image captured by the omnidirectional camera.

Layer	**Number of RFs (w×h)**	**RF size (w×h)**	**Stride (w×h)**	**Input dim**	**Output dim**
1	99 × 9	12 × 12	6 × 6	432	12
2	48 × 3	5 × 5	2 × 2	300	14
3	15 × 1	6 × 3	3 × 1	252	16
4	1 × 1	1 × 1	1 × 1	240	8

Table 5.2: Network parameters for the outdoor experiment. The number of Receptive Fields (RF) per layer, RF size and stride are given for every layer of the SFA network.

Results

All SFA-outputs of the network have a mean positional variance $\eta_p \approx 1$ and their mean orientation variance η_φ ranges from 0.00 (s_1) to 0.05 (s_8) and thus are almost only coding for spatial position while being orientation invariant. Note that the lower magnitude of η_φ, compared to the simulation results, is caused by the faster changing orientation due to the robot's additional real rotation.

As expected, the spatial firing maps in Fig. 5.5a) do not encode position as clearly as in the simulator environment due to the non-static environment, the inhomogeneous sampling and variations in velocity. Spatial firing maps of the first function encode the position on the y-axis, while x-position is encoded less obviously in the maps of outputs three and four. In contrast to the simulation we compute the regression from

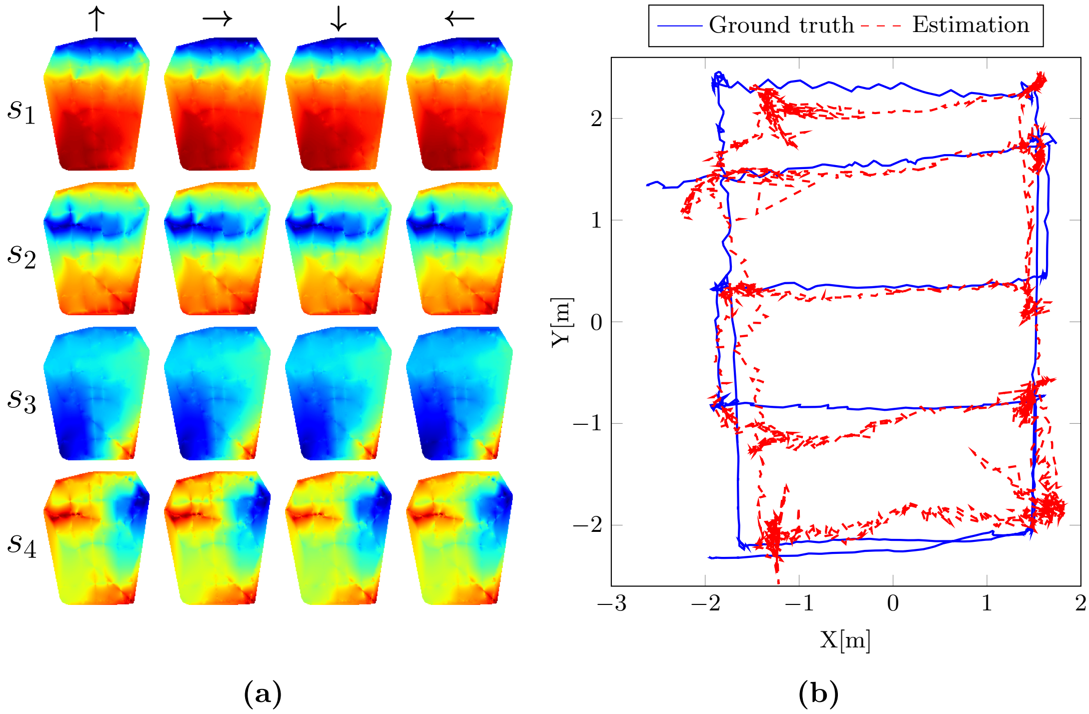


Figure 5.5: Results for the real world environment. (a) Spatial firing maps of the four slowest SFA-outputs $s_{1...4}$ for relative orientations $0°, 90°, 180°$ and $270°$. First SFA-output encodes the position on the y-axis with low values in the north and high values in the south. Notice the area in the south-west with highest values. This region has been passed multiple times, so that environmental changes led to variations. Second function is a higher oscillating representation of the first one, which indicates that other varying components of the configuration space changed at least twice as fast as the y-position. Functions three and four suggest weak encoding of the x- and y-position. (b) Ground truth and estimated positions for the test run. Estimations are averaged over the simulated rotation for one location.

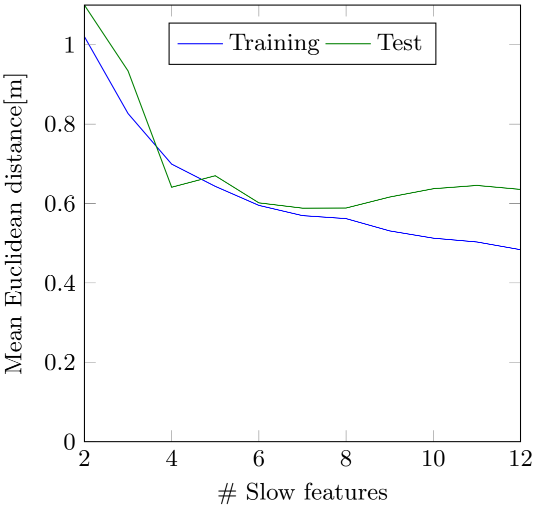


Figure 5.6: Training and test error for a varying number of slow feature outputs. The error is given as the mean Euclidean distance from ground truth. As in the simulator experiments the test error is minimal when using 8 slow feature outputs. For a further increasing number of SFA-outputs the regression model starts to overfit the training data.

the SFA-outputs to the metric ground truth positions for the training data and apply it to SFA-outputs on the test set. The resulting MAE is 0.23 meter (5.3%) for the x-coordinate and 0.175 meter (3.7%) for the y-coordinate. The standard deviation amounts to 0.20 and 0.13 meter respectively. Higher errors can be noticed in a small area in the west that was not passed in the training-run (see Fig. 5.4b) and an area in the south west, which could also be noticed in the spatial firing map with the highest SFA-outputs. Another prominent area with higher errors is located in the north west, where the maps of outputs two and three show discontinuities.

Minor deviations can be observed at turning points in the trajectory, where vibrations of the vision system caused distortions in the unwarped panoramic images. Even though the coding for the x-position is less obvious compared to the simulation, it is apparently sufficient for self-localization.

As in the simulator experiment the optimal number of slow feature outputs has been evaluated by analyzing the training and test error for an increasing number of slow feature outputs in the range from 2 to 12. Using the 8 slowest feature outputs resulted in a minimal test error. The development of the training and test error curves is shown in Fig. 5.6.

5.1.3 The Impact of the Window Size

Learning location specific and orientation invariant functions with the SFA-model requires that the orientation of the robot changes on a faster timescale than its translation,

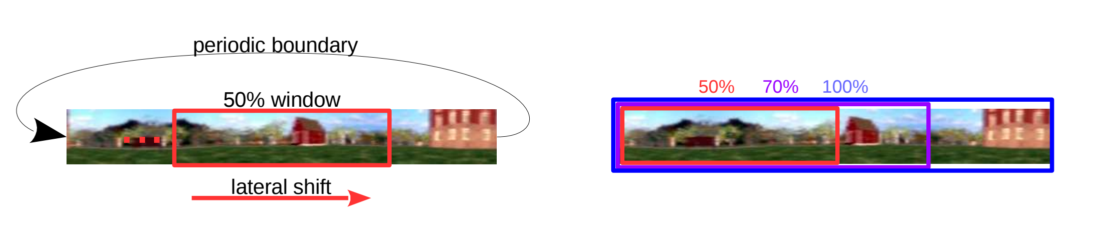


Figure 5.7: Simulated rotation with varying window sizes. During the simulated rotation the sliding window is laterally shifted along the images' x-axis. The periodic image boundaries allow to simulate a full rotation. The part of the image covered by the window represents the data that is processed at one time step. The size of the sliding window is given as the percentage of the original panoramic view.

since the spatial encoding of the SFA-model depends on the movement statistics during training. To change the perceived image statistics a complete rotation is simulated for every image by laterally sliding a window over the periodic panoramic views (see Fig. 5.7). For panoramic images a lateral shift is equivalent to a rotation of the image sensor on the spot around its yaw axis. In the experiments we used a window size of 100% which means that the whole image is processed by the model but incrementally shifted in every step of the simulated rotation. Learning with smaller windows would decrease the computational complexity. However, experiments with different window sizes show that the orientation variance, and hence the localization error, is increasing with smaller windows. This effect is illustrated in Fig. 5.8a and Fig. 5.8b. Please note that the models have been trained on a smaller resolution than in the previous experiments in order to accelerate processing time. Thus, the results for the 100% window are not equal to the ones stated in the previous simulator experiment.

We conjecture that the complexity of the learning problem is getting too high if the window size is reduced. For an optimal performance the output of the functions should be nearly constant for the input perceived during a simulated rotation and vary after a change in position. During a simulated rotation with a 100% window the output SFA-node receives statistics from the whole image with a lateral shift for every step. With decreasing window sizes, however, the input statistics perceived by the output node vary increasingly which requires learning more complex functions. To increase the function space the expansion of the input data was changed from quadratic to cubic. Furthermore, the learning problem was simplified by training individual SFA-nodes per receptive field instead of sharing the weights across one layer. Experiments with these modifications and a window size of 50% resulted in a considerable improvement for the simulated environment (see Fig. 5.8a and Fig. 5.8b. For the non-static and noisy outdoor data-sets the encoding of the location did not improve with the complex model and 50% windows and it is hard to determine which information is extracted from the expanded high dimensional input. Depending on the requirements a trade-off between mapping quality and computation time has to be made. Thus, we use a window size of 100% throughout the experiments.

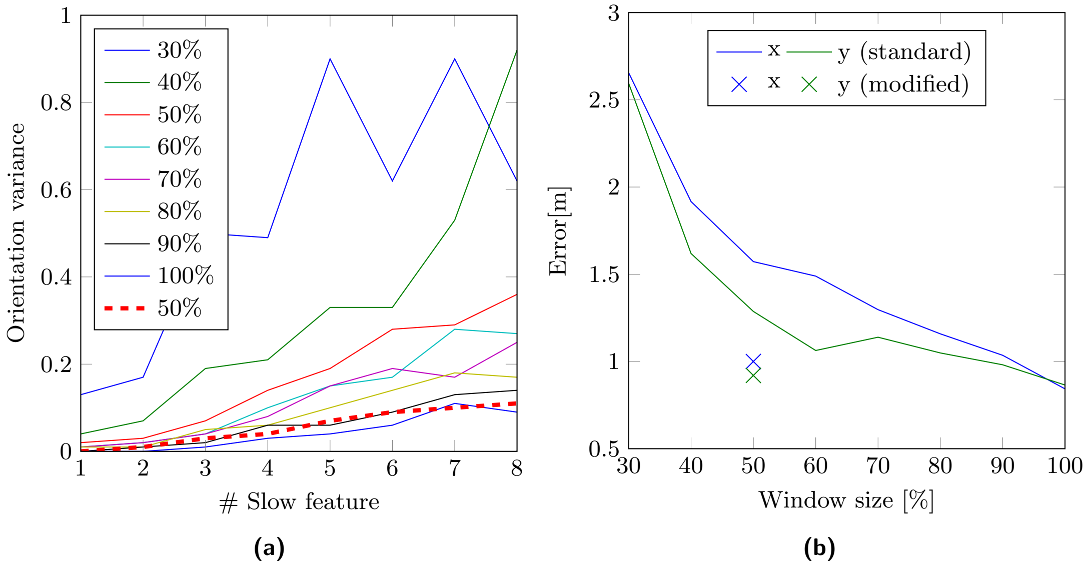


Figure 5.8: Effect of different window sizes. (a) The encoded orientation variance of the eight slowest features for different window sizes. Variance of the modified network with a 50% window is shown by the red dotted line. (b) Localization errors obtained with networks trained with the eight slowest features and increasing window sizes. Error of the modified network for a 50% window is indicated by crosses. Performance of the modified network ranges between the original network trained with 90% and 100% windows.

5.1.4 Discussion

The biologically motivated concept of SFA-localization was systematically transferred step by step into a self-localization task of a mobile robot and successfully applied in a simulated and a real world outdoor environment. Despite its simplicity the system demonstrates a reasonable localization performance. Explorations in the simulated environment have shown that SFA combined with simulated rotation of an omnidirectional view allows for self-localization with errors of under 2% relative to the coordinate range. Experiments in the outdoor environment resulted in an average self-localization accuracy of 0.23 meter (5.3%) for the x coordinate and 0.175 meter (3.7%) for the y-coordinate, which is significantly smaller than the robot's own size (approx. 50 × 50 cm). Using the first 8 SFA-outputs resulted in the highest localization accuracy in the simulator as well as in the real world experiment. Therefore, we conducted the following localization experiments in this chapter with the same number of outputs. The results from the investigation of different window sizes suggest that orientation invariance can be learned from smaller windows by increasing the function space of the network. However, for the more noisy real world data increasing the complexity of the network deteriorated the localization performance. Thus, we used a window size of 100% for the following experiments.

5.2 Comparison to Visual Simultaneous Localization and Mapping Methods

The problem of visual self-localization in unknown environments has been investigated in great detail as an inherent part of the simultaneous localization and mapping (SLAM) problem. Geometric SLAM approaches typically require highly calibrated optics and extract sparse visual features to estimate the ego-motion of the camera and the features' 3D position from correspondences between successive frames. Methods fusing ego-motion estimates and sensor readings in a probabilistic framework (e.g. Extended Kalman Filter, Particle Filter) have been proposed [26, 32]. Recent approaches [70, 145, 109] represent the map as a pose-graph of keyframes which are connected by spatial constraints like ego motion estimates and feature observations. Loop closure detections enable the correction of accumulated drift by a global optimization of the pose-graph. The 3D position of features and camera poses is jointly optimized by local bundle adjustment minimizing the re-projection error. Direct methods, on the other hand, do not rely on sparse image features but instead estimate the camera motion and scene depth performing direct image alignment by minimizing the difference in pixel intensities. They make use of the whole image [115], which yields a dense 3D reconstruction, or only image regions with high gradients [35], which requires less computational resources and results in a semi-dense reconstruction of the environment.

SLAM systems based on pose graph-optimization generally consist of a front end that establishes image correspondences and performs ego-motion estimation and loop closure detection. The backend uses the information provided by the front end to build and update the map which involves methods from graph theory, optimization and probabilistic estimation. The underlying methods evolved over the last 20 years, therefore modern SLAM approaches have grown to highly complex technical systems. Their successful application furthermore requires sensor calibration and a careful parameter selection. In comparison, the presented SFA-network is a rather straightforward model for self-localization in the sense that it applies the same unsupervised learning rule in a hierarchical network directly to the images from an uncalibrated image sensor. Furthermore, it has also been shown in [42] that the hierarchical model is robust under a range of parameter settings for image resolution, number of layers, receptive field size and overlap. An advantage of the SLAM methods is that they incrementally build a map of the environment and are able to simultaneously localize within this map. The SFA-model requires an initial offline learning phase where the environment is evenly sampled as it is based on a closed form solution for solving a generalized eigenvalue problem. But once trained, localization is absolute and instantaneous since slow features can be computed from a single snapshot of the environment. Thus, localization is not affected by drift over time and there is no need to deal with re-localization. Besides the even sampling the model has no further restrictions on the movement pattern and is able to deal with pure rotational movement which poses a problem to the aforementioned geometric methods. These properties render the model suitable for service robot scenarios.

To investigate the localization capabilities in a realistic setting we applied the biologically motivated model of SFA localization in small scale open field scenarios in indoor and outdoor environments and compare its performance for the first time with the feature based ORB-SLAM [109] and the semi-dense LSD-SLAM [35]. The methods have been chosen because they allow a metric evaluation, represent the state of the art in monocular visual SLAM and are made available by the authors[23].

The robot was moved with a wireless joystick during the training- and test-runs at a maximum velocity of 20 cm/s. Localization accuracy was evaluated for the test run only. The SFA-model requires an offline training phase to learn the spatial representation of the environment. SLAM methods, on the other hand, perform mapping and pose estimation incrementally and online, why the localization accuracy can be evaluated on the test run directly. To make a fair comparison we also provided the SLAM-methods with image data from the training and test run and measured the performance on the test run. We used the default configuration given by the authors and executed the SLAM methods in mapping mode for all experiments to allow for map updates and pose correction in the subsequent test run. Their localization accuracy is evaluated over five runs since the results are non-deterministic due to the parallel execution of the mapping and tracking threads. The parameters of the SFA-network used in the real world experiments are given in Table 5.3. To evaluate the localization accuracy the estimated trajectories are aligned to the ground truth trajectories by finding the rotation and translation between the two 3D-point sets which minimizes the mean squared error as described in [6]. As the absolute scale can not be recovered from a single camera we perform the fitting over a predefined scale range.

5.2.1 Image Acquisition and Preprocessing

LSD- and ORB-SLAM require a calibrated camera operating at a high framerate while the SFA-model processes omnidirectional images in order to facilitate learning of orientation invariant representations. Therefore two different camera types were used for image

[2] https://github.com/raulmur/ORB_SLAM

[3] https://github.com/tum-vision/lsd_slam

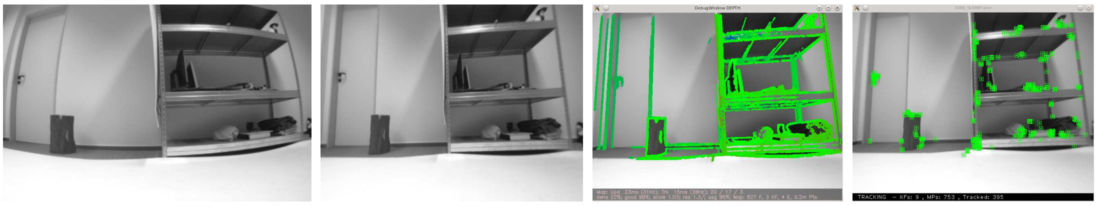
Layer	Number of RFs (w×h)	RF size (w×h)	Stride (w×h)	Input dim	Output dim
1	101×7	9×10	4×5	90	12
2	49×2	5×5	2×2	300	12
3	23×1	5×2	2×1	120	12
4	1×1	1×1	1×1	276	8

Table 5.3: Network parameters for the real world experiments. Number of Receptive Fields (RF) per layer, RF size and stride are given for every layer of the SFA network.

acquisition. The omnidirectional camera captures images with a framerate of 8 frames per second (fps) and is mounted above the marker box. For the SLAM methods we used a global shutter camera, equipped with a fisheye lens and operating at a framerate of 40 fps. Camera and lens are equal to the ones used by the authors of [35]. The camera was mounted on the front side of the robot heading orthogonal to the driving direction. This setup was chosen to enable wider baseline stereo correspondences and to enhance the robustness of the tracking during rotational movement. In case of a limited field of view, forward and rotational movement leads to small baseline stereo correspondences between successive keyframes. This increases the depth ambiguity and might cause a complete failure of the tracking system. Early results with a forward facing camera were systematically worse. Images and ground truth coordinates are saved together with the current timestamp to enable a synchronization of image data and ground truth measurements. The offset from the cameras to the center of the marker box is measured manually and integrated into the ground truth computation. Exposure of both cameras was set to automatic mode to account for changing lighting conditions during the recordings. Images of the perspective camera are captured in grayscale with a resolution of 752 × 480 pixels. The undistorted images are cropped to 640 × 480 pixels (see Fig. 5.9). The omnidirectional images are unwarped to panoramic views with a resolution of 409 × 40 pixels and converted to grayscale. The image data is then normalized to zero mean and unit variance to gain robustness against global illumination changes. The rough terrain in the outdoor environment causes changes in the tilt angle of the robot. Thus image statistics from the same place with different physical orientations are not the same and our orientation invariance learning does not work anymore. Therefore we randomly shifted the center of every omnidirectional image by an offset from -5 to 5 pixels for the computation of the panoramic views. This way the resulting representations become invariant with respect to the tilt angle of the robot.

5.2.2 Experiments in an Indoor Environment

The datasets for the experiments were recorded in an indoor environment covering an area of about 4×4 meters. Two experiments with different movement characteristics have



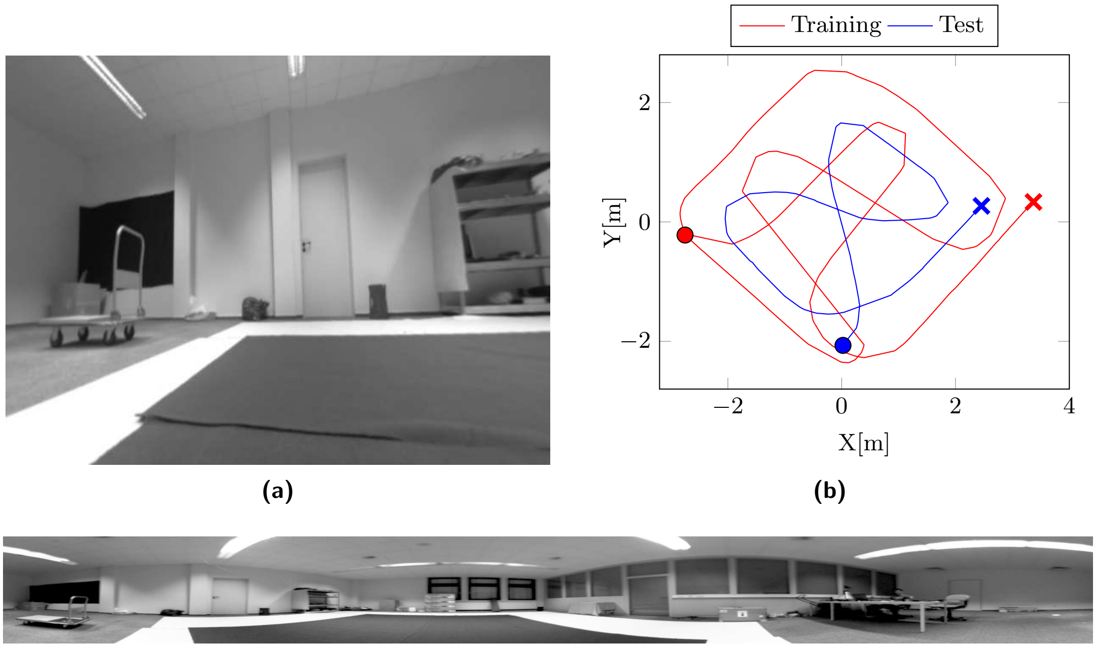
(a) (b) (c) (d)

Figure 5.9: (a) Original Image taken with a fisheye lens. (b) Images are undistorted and cropped before being processed by the SLAM-methods. (c) Image regions with high gradients from the current keyframe processed by LSD-SLAM. (d) Extracted ORB-features of the current frame.

been performed since it influences the mapping results of the methods in different ways. The training trajectory for both experiments evenly samples the area with crossings along the coordinate axis resulting in a grid-like pattern. In the first experiment turn maneuvers were executed with a large curve radius while the robot was turned at the spot in the second experiment. Turning on the spot promotes the spatial encoding of the SFA-model because it naturally leads to a larger amount of overlap between different parts of the trajectory for a similar track length. Crossing points in the trajectory ensure that image data from the same place at different points in time are presented to the SFA-model which improves spatial encoding. Pure rotational movement during the mapping phase is problematic for the SLAM-methods since the camera motion and depth estimation requires a certain amount of translation between successive frames. Larger curve radii are thus necessary to achieve a good ratio of rotational and translational movement. In principle this turn characteristic does not pose a problem to the SFA-model but might decrease the quality of the spatial representation since the overlap of the trajectory is quite low compared to trajectories of the same length where the robot turns on the spot (cf. Fig. 5.10b and Fig. 5.13).

Experiment I

The trajectory follows a grid-like structure that evenly covers the training area. Turn maneuvers were performed with a large curve radius. As stated above this ensures a proper ratio of rotational and translational movement required by SLAM-methods during the mapping phase while this is not optimal for the SFA-model. The trajectory of the training- and test-runs are given in Fig. 5.10a. Example images from both cameras are illustrated in Fig. 5.10b, 5.10c.



(a)

(b)

(c)

Figure 5.10: Experiment in the indoor environment. (a) Undistorted image from the perspective camera mounted on the side of the robot. (b) Trajectory of the training- and test-run. (c) Panoramic image captured by the omnidirectional camera.

Results

The mean positional variance of the resulting SFA-outputs η_p is ≈ 1 and the mean orientation variance η_φ is ≈ 0. SFA-outputs thus have a high spatial structure and are almost completely orientation invariant. The spatial firing maps of the four slowest SFA-outputs shown in Fig. 5.11 do not show an obvious encoding of the position with clear gradients along the coordinate axis as in the simulator experiment. The first function seems to be coding for the distance to the borders while outputs two and three suggest coding for the x- and y-coordinate. The mean Euclidean distance is 0.21m. The best localization performance is achieved with LSD-Slam with a median localization error of 0.19m when the train- and test-images are used and an error of 0.12m when using the test-images alone. The accuracy is quite constant over the runs except for the fifth run on the test data.

The accuracy of ORB-Slam amounts to a median error of 0.45m on the training- and test-images while the interquartile range of the five runs is quite high with 0.27m. On the test-images alone the variance of the errors is lower and the median error amounts to 0.23m. The performance of ORB-Slam probably suffers from the low textured indoor environment which is disadvantageous for the amount and distribution of robust visual

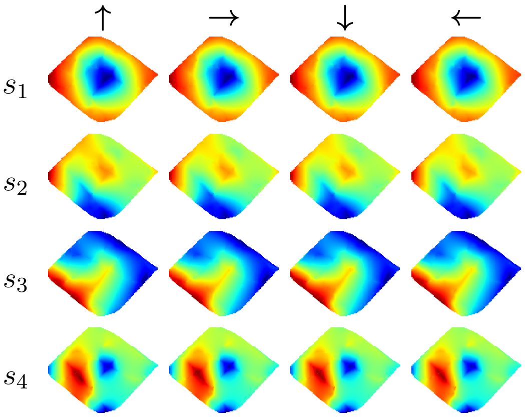
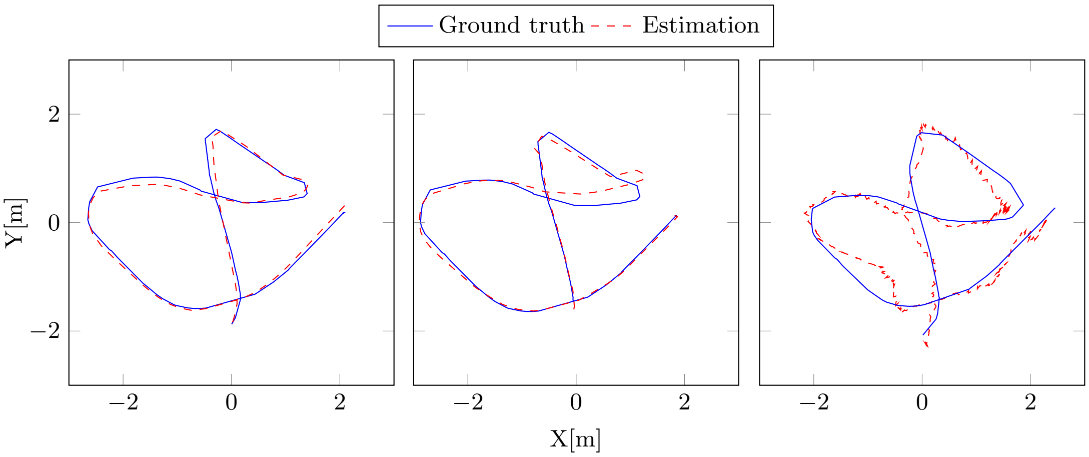


Figure 5.11: Spatial firing maps of the first four SFA-outputs. The first function seems to be encoding the distance to the borders of the area. Functions two and three suggest encoding of the x- and y-position while the gradients along the coordinate axis are not as clear as in the simulator experiments. All outputs are highly orientation invariant.

features. Surprisingly the performance of the SLAM-methods is worse when images from the training run are used for the experiment. We expected that mapping quality would improve through the additional information from the training run. Instead the constructed pose-graphs often got corrupted due to tracking failures. The results are presented in detail in Table 5.4. The resulting trajectories of the best runs of the different methods are illustrated in Fig. 5.12.

	Train- and Test-Run						**Test-Run**					
	1	2	3	4	5	Median	1	2	3	4	5	Median
ORB	0.22	0.45	0.57	0.49	0.12	**0.45**	0.25	0.25	0.16	0.18	0.23	**0.23**
LSD	0.17	0.30	0.18	0.15	0.38	**0.18**	0.11	0.11	0.13	0.12	0.40	**0.12**
SFA	**0.21**											

Table 5.4: Localization accuracies for indoor experiment I. Accuracies are given in meters as the mean Euclidean distance from all ground truth measurements. The performance of LSD- and ORB-Slam is measured over five runs since the results are not deterministic due to the parallel execution of the tracking and mapping threads. The SFA-localization requires an offline training phase and thus is deterministic so that only one value is given for the mean Euclidean distance.



(a) LSD (0.11m) **(b)** ORB (0.12m) **(c)** SFA (0.21m)

Figure 5.12: Estimated trajectories of the best runs. (a) The trajectory estimated by LSD-Slam clearly follows the ground truth with small deviations. (b) Deviations in the trajectory produced by ORB-Slam start to get greater after the left turn where the curve radius of the camera is quite small. (c) Since the SFA-localization is absolute and no pose filtering is performed the trajectory is in general more noisy. The accuracy decreases near the borders.

Experiment II

The second experiment was conducted with a different movement strategy (see Fig. 5.13). Turning maneuvers were performed on the spot resulting in a denser sampling of the area and larger overlaps in the trajectory which is beneficial for the SFA-localization. Monocular SLAM-methods on the other hand have problems with pure rotational movement since it is not possible to triangulate features without a sufficiently large baseline so that they easily lose tracking. The Movement strategy for SFA is only relevant in the training phase while it works for every trajectory during testing.

Results

As in the first experiment SFA-outputs have a high spatial structure while being invariant with respect to the orientation of the robot with a mean positional variance η_p of ≈ 1 and a mean orientation variance η_φ of ≈ 0. The spatial firing maps presented in Fig. 5.14 again seem to be encoding the distance to the center mixed with positional encoding which can be seen by a gradient along the coordinate axis. The best localization performance is achieved by the SFA-model with a mean localization error of 0.13m. Both SLAM-methods fail completely on the test trajectory alone. When using the training and test data tracking failures during the test run are retained by a re-localization. The median localization errors of ORB- and LSD-Slam amount to 0.78m and 0.44m

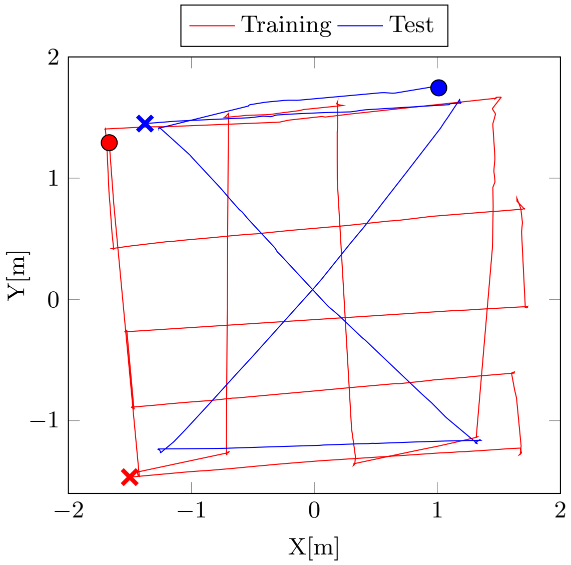


Figure 5.13: Training- and test-trajectory of the second experiment which is more favorable for the SFA-localization because of more crossing points and a denser sampling of the area. Trajectories are challenging for the SLAM-methods because of the high amount of rotational movement. Start and end points are marked by a cross and a circle, respectively.

respectively. The results are presented in detail in Table 5.5. The resulting trajectories of the best runs of the different methods are illustrated in Fig. 5.15.

	Train- and Test-Run						**Test-Run**					
	1	2	3	4	5	Median	1	2	3	4	5	Median
ORB	0.41	1.23	1.01	0.78	0.46	**0.78**	1.34	1.34	1.34	1.34	1.34	**1.34**
LSD	0.27	0.58	0.75	0.22	0.44	**0.44**	1.10	1.33	1.08	1.08	1.32	**1.10**
SFA	**0.13**											

Table 5.5: Localization accuracies for indoor experiment II. Accuracies are given in meters as the mean Euclidean distance from all ground truth measurements. The performance of LSD- and ORB-Slam is measured over five runs since the results are not deterministic due to the parallel execution of the tracking and mapping threads. The SFA-localization requires an offline training phase and thus is deterministic so that only one value is given for the mean Euclidean distance.

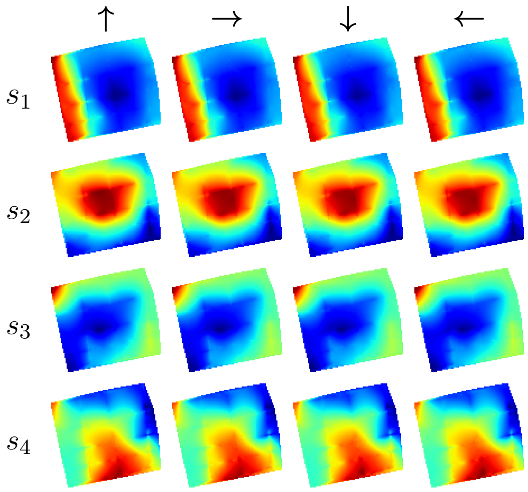
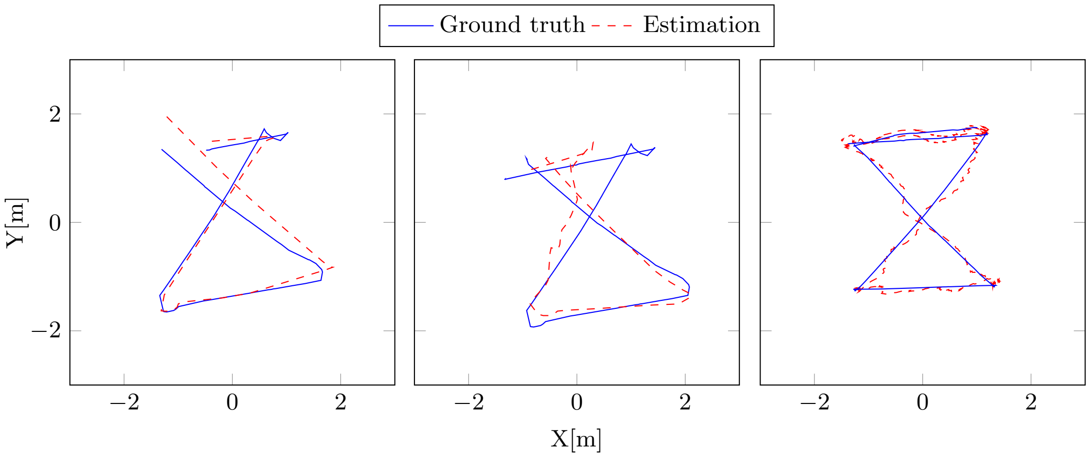


Figure 5.14: Spatial firing maps of the four slowest SFA-outputs. First two functions seem to be coding for the distance to the borders but also show a gradient along the coordinate axis.



(a) LSD (0.27m)

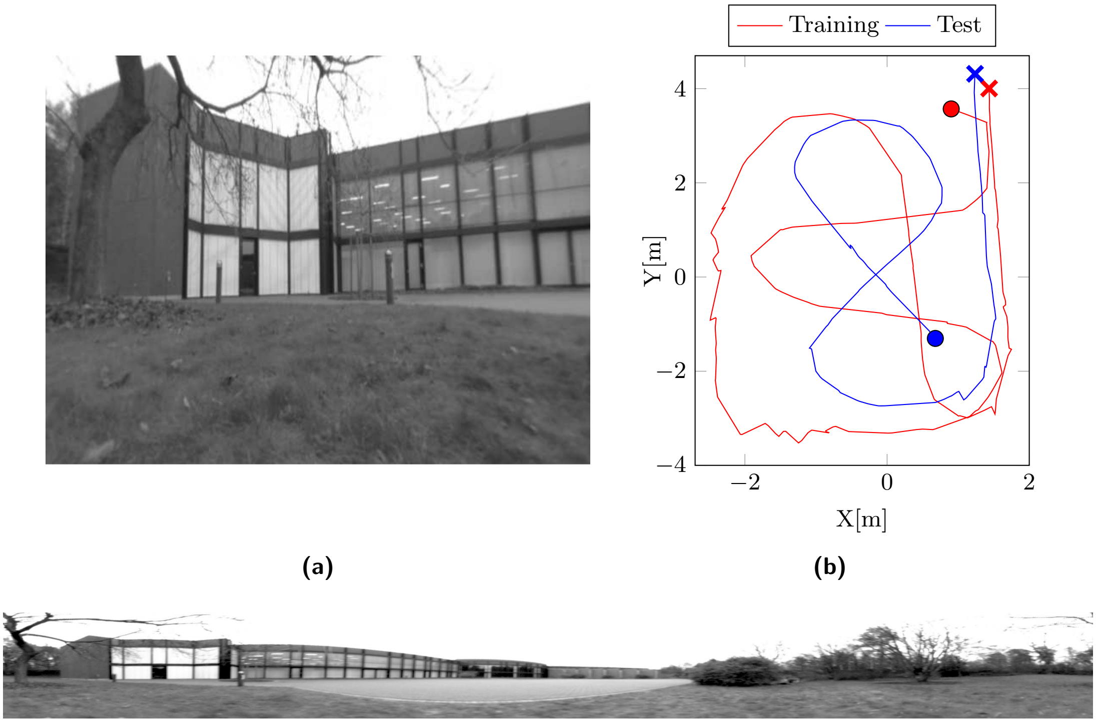
(b) ORB (0.41m)

(c) SFA (0.13m)

Figure 5.15: Estimated trajectories of the best runs. (a) LSD-Slam is not able to re-localize during the first seconds so that no pose estimates are available. The trajectory closely follows ground truth until localization quality decreases after the third turn. (b) ORB-Slam is instantaneously able to re-localize in the map. However, the following pose estimates clearly deviate from ground truth. (c) The estimated trajectory of the SFA-model clearly follows the ground truth.

5.2.3 Experiments in an Outdoor Environment

Outdoor experiments were performed within an area of approximately 5 × 7 meters on rather uneven ground covered by grass. Recordings were done in the late afternoon with modest changes in lighting conditions. The trajectory of the training- and test-run are given in Fig. 5.16b. Example images from both cameras are illustrated in Fig. 5.16a, 5.16c.



(a)

(b)

(c)

Figure 5.16: Experiment in the outdoor Environment. (a) Undistorted image from the perspective camera mounted on the side of the robot. (b) Trajectory of the training- and test-run. Start and end points are marked by a cross and a circle, respectively. (c) Panoramic image captured by the omnidirectional camera.

Results

The resulting SFA-outputs show a clear spatial coding and are orientation invariant. Spatial firing maps illustrated in Fig. 5.17 show a slightly rotated gradient along the coordinate axis. Due to the uneven ground and the difficult lighting conditions the dataset is challenging for all methods. Both, LSD- and ORB-SLAM, have problems with scale estimation in the first part of the trajectory leading to larger errors in the localization. Even though the trajectory of the SFA-model exhibits larger variance in local estimates the performance is best on this data set with a mean localization error of 0.33m. Due to the instantaneous and absolute localization the model is not affected by drift over time. ORB-Slam achieves a median accuracy of 0.35m on the test data alone

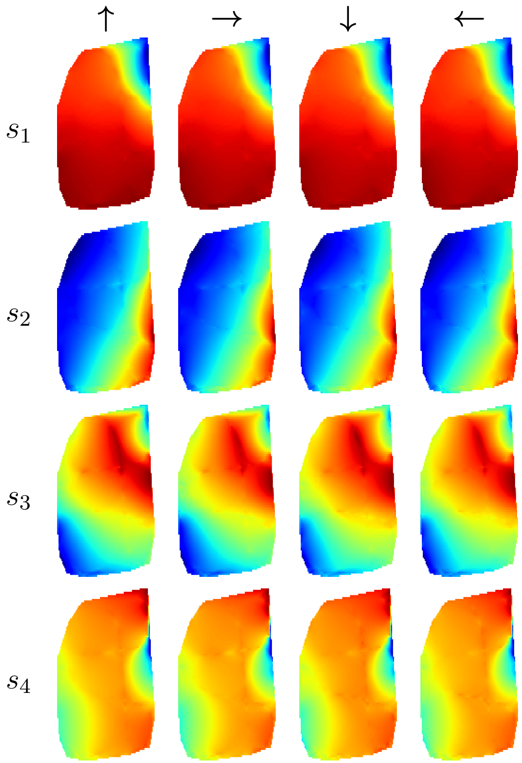


Figure 5.17: Spatial firing maps of the four slowest SFA-outputs. First two SFA functions show spatial encoding while directions in the the data with least temporal variation are slightly rotated with respect to the coordinate axis. This is the case if the temporal variation of the x- and y-coordinate is nearly equal. Functions three and four are higher modes of the first two outputs.

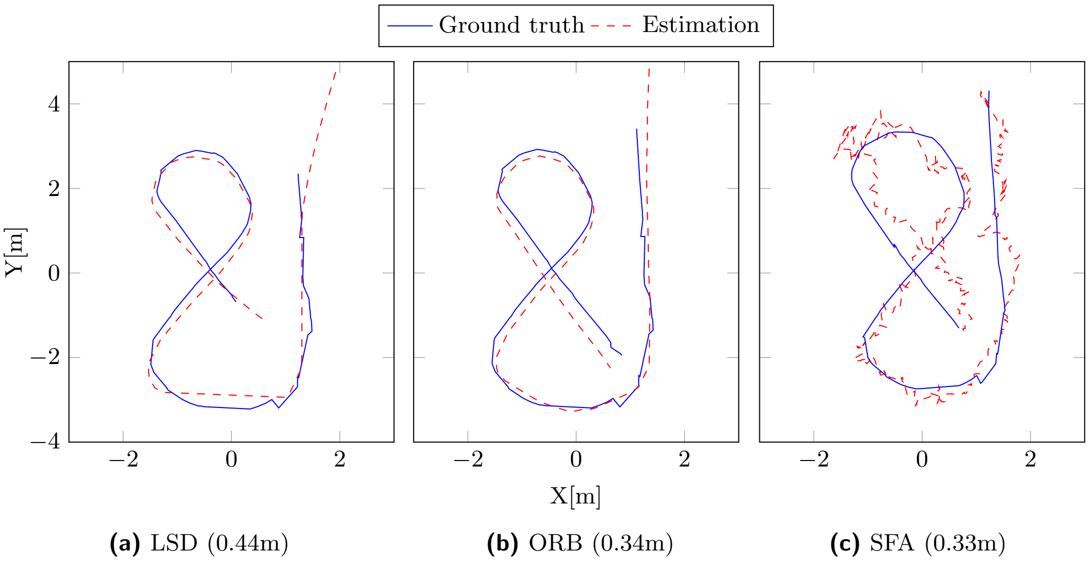
followed by LSD-Slam with a median accuracy of 0.44m when the training and test data is used. The results are presented in detail in Table 5.6. The resulting trajectories of the best runs of the different methods are illustrated in Fig. 5.18.

5.2.4 Discussion

Results of the experiment show that the localization performance of the straight forward model is competitive to state of the art geometric methods and can even surpass them for certain trajectories. In contrast to the SLAM methods the SFA-model requires an offline learning phase with an even sampling of the area. After the training phase localization is instantaneous and absolute which obviates dealing with drift over time and re-localization. The training trajectory has to include a certain amount of crossings to support spatial coding which renders SFA inappropriate for localization along one dimensional routes like road tracks. Potential application domains could be service robotics

	Train- and Test-Run						**Test-Run**					
	1	2	3	4	5	Median	1	2	3	4	5	Median
ORB	0.99	0.52	1.86	0.61	1.16	**0.99**	0.35	0.35	0.71	0.63	0.34	**0.35**
LSD	0.47	0.60	0.69	0.53	0.44	**0.53**	1.50	1.52	1.49	1.51	1.55	**1.51**
SFA	**0.33**											

Table 5.6: Localization accuracies for the outdoor experiment. Accuracies are given in meters as the mean Euclidean distance from all ground truth measurements. The performance of LSD- and ORB-Slam is measured over five runs since the results are not deterministic due to the parallel execution of the tracking and mapping threads. The SFA-localization requires an offline training phase and thus is deterministic so that only one value is given for the mean Euclidean distance. In this experiment the instantaneous and absolute position estimates from the SFA-model result in the best performance.



(a) LSD (0.44m) **(b)** ORB (0.34m) **(c)** SFA (0.33m)

Figure 5.18: Estimated trajectories of the best runs. (a) Estimated trajectory of LSD-Slam clearly follows the ground truth while the scale is estimated incorrectly in the beginning of the trajectory. (b) ORB-Slam also has problems with scale estimation in the beginning. The best performance is achieved on the test data alone where only one loop closure occurs so that the estimation starts to drift over time. (c) SFA estimates have a higher variance which can be explained by the uneven ground. However, the ability of instantaneous and absolute position estimation result in the best performance for this experiment.

which require localization in open field scenarios. Since the spatial representations are directly learned from image data they are more susceptible to appearance changes in the environment than a feature based method. In chapter 6 we will investigate learning strategies and feature representations that improve robustness of the representations. In contrast to the SLAM methods the SFA-model only uses visual input to perform absolute localization on a single image basis. A substantial gain in performance can be expected when odometry measurements are incorporated in the estimation.

5.3 Odometry Integration

Place and head direction cells in rats are strongly driven by visual input [59]. However, information from the vestibular system of the animal also contributes to the activity of these cells. Experiments conducted in darkness, i.e. in the absence of external visual cues, have shown that the firing patterns of the spatial cells remain stable for some minutes [132, 126]. This indicates that the animal performs path integration by incrementally updating the belief of its own pose based on self-motion cues. Over longer periods of time errors accumulate and the belief of the rat's own pose starts to diverge from the true one. The resulting drift can only be corrected by receiving feedback from visual input or some other kind of external sensor measurement.

The SFA model presented in this work does not integrate self-motion cues over time but instantaneously estimates the position in slow feature space from a single image. Since it relies on visual input only, it can not fully explain firing behavior of spatial cells and is thus classified as a local view model [126]. As the hierarchical SFA-model learns a complex function of the spatial position directly from high dimensional visual input the estimates for consecutive measurements exhibit some variation but are absolute with respect to the global coordinate system. Hence, our model is complementary to a path integration model which is locally consistent but drifts over time. The combination of both models constitutes a more complete representation of spatial cell firing behavior. Provided that the uncertainties of the individual sensor modalities are known the weighted combination of internal and external measurements leads to an improved localization accuracy compared to estimations based on the respective ones alone. It is also a common approach in SLAM methods where the trajectory is estimated incrementally based on ego-motion estimates and accumulated errors are corrected using information from loop closure detections, i.e. the robot identifies a place it has seen before [31].

To combine the slow feature outputs with self-motion cues they have to be in a common coordinate system. In the previous localization experiments a supervised regression model was trained to learn the mapping from slow features to metric coordinates. The ground truth label information was obtained from pose estimations of a visual marker box attached to the robot. Such a metric mapping function enables the combination of the absolute slow feature estimates with the self-motion cues from the robot's odometry to increase accuracy and to obtain smoother trajectories. Additionally, it allows to visualize the learned SFA-representations and the driven trajectories and to communicate them to a potential user of the system. For realistic application scenarios, however, the use of additional external infrastructure is not a feasible solution. Therefore, we propose a method to learn the mapping function from slow feature space to metric space in an unsupervised fashion.

5.3.1 Unsupervised Metric Learning

Given the slow feature vector $\mathbf{s} \in \mathbb{R}^J$ computed for the image from a given position $\mathbf{p} := (x, y)^\top$ we want to find the weight matrix $\mathbf{W} = (\mathbf{w}^\mathbf{x}\ \mathbf{w}^\mathbf{y}) \in \mathbb{R}^{J\times 2}$ such that the error $\boldsymbol{\varepsilon} \in \mathbb{R}^2$ for the estimation $\mathbf{p} = \mathbf{W}^\top\mathbf{s} + \boldsymbol{\varepsilon}$ is minimal. Without external measurements of the robot's position $\mathbf{p}$ the only source of metric information available is from ego-motion estimates. As already stated, pose estimation solely based on odometry accumulates errors over time and thus does not provide suitable label information to learn the weight matrix $\mathbf{W}$ directly. Especially errors in the orientation measurements cause large deviations. The distance measurements along a certain direction, on the other hand, are very precise. Learning the weight matrix $\mathbf{W}$, using these distance measurements, requires to impose constraints on the trajectory of the robot. The robot needs to drive along straight lines such that the training area is evenly covered and there exists a certain number of intersections between the lines. The prerequisite of such a movement strategy is a valid assumption considering the movement pattern of current household robots.

A line l consists of M points $\mathbf{P} = (\mathbf{p}_1, \dots, \mathbf{p}_M)$. At every point $\mathbf{p}_m$ we record the slow feature vector $\mathbf{s}_m$ computed for the corresponding image and the current distance measurement d_m to the origin $\mathbf{o}$ of line l where $d_m = ||\mathbf{p}_m - \mathbf{o}||_2$ and $\mathbf{o} := (x_0, y_0)^\top$. Based on the orientation α, the origin $\mathbf{o}$ and the distance measurement d_m the reconstruction of a point is given by the equation $\mathbf{p}_m = \mathbf{o} + d_m(\cos(\alpha)\ \sin(\alpha))^\top$. Given a proper weight matrix $\mathbf{W}$ the reconstruction of the same point using slow feature vector $\mathbf{s}_m$ is defined by $\mathbf{p}_m = \mathbf{W}^\top\mathbf{s}_m$. However, the line parameters $\mathbf{o}$ and α as well as the weight matrix $\mathbf{W}$ are unknown and need to be estimated. Given optimal parameters the difference between the point-wise estimations based on the line parameters and the weight matrix should be zero. Thus, the parameters can be learned simultaneously by minimizing the difference in the point-wise reconstruction. The distance measurements from odometry induce the correct metric scale while the intersections of the line segments and the weights ensure a globally consistent mapping. For N line segments the cost function for parameters $\theta = (\alpha_n, \mathbf{o}_n, \mathbf{W})$ is the following:

$$C(\theta) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}||\mathbf{o}_n + d_{n,m}\left(cos(\alpha_n) \quad sin(\alpha_n)\right)^\top - \mathbf{W}^\top\mathbf{s}_{n,m}||_2^2 \tag{5.1}$$

Where the number of points M depends on the specific line l_n. The corresponding partial derivatives w.r.t. the parameters θ are given by:

$$\frac{\partial C}{\partial \alpha_n} = \sum_{m=1}^{M} d_{n,m} \begin{pmatrix} sin(\alpha_n) & -cos(\alpha_n) \end{pmatrix} (\mathbf{W}^\top \mathbf{s}_{n,m} - \mathbf{o}_n) \tag{5.2}$$

$$\frac{\partial C}{\partial \mathbf{o}_n} = \sum_{m=1}^{M} \mathbf{o}_n + d_{n,m} \begin{pmatrix} cos(\alpha_n) & sin(\alpha_n) \end{pmatrix}^\top - \mathbf{W}^\top \mathbf{s}_{n,m} \tag{5.3}$$

$$\frac{\partial C}{\partial \mathbf{W}} = \sum_{n=1}^{N} \sum_{m=1}^{M} -\mathbf{s}_{n,m}(\mathbf{o}_n + d_{n,m} \begin{pmatrix} cos(\alpha_n) & sin(\alpha_n) \end{pmatrix}^\top - \mathbf{W}^\top \mathbf{s}_{n,m})^\top \tag{5.4}$$

If the robot explores the environment more efficiently, driving along straight lines in a grid like trajectory with a few crossings along each coordinate axis, the resulting angles between line segments will all be nearly 90°. Hence, the learned linear mapping of slow feature vectors to metric positions defined by the weight matrix might contain a shearing of the coordinate axis. In the most extreme case, the learned parameters will lead to a solution where all points are mapped onto a single line. To encounter this problem the orientation α_n of a line can be constrained such that the relative angle between consecutive line segments corresponds to the measured change in orientation from odometry. Therefore, we add a term to the cost function which punishes the deviation of the relative angle defined by the current estimate of α_n and α_{n+1} from the measured change in orientation obtained from odometry $\measuredangle l_n l_{n+1}$. We express α_n and α_{n+1} as unit vectors to obtain the cosine of the relative angle given by their dot product. The deviation of the relative angle from the measured angle is then defined as the difference between the angles' cosine values. The cost function from 5.1 is extended accordingly resulting in:

$$\begin{aligned} C(\theta) &= \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} ||\mathbf{o}_n + d_{n,m} \begin{pmatrix} cos(\alpha_n) & sin(\alpha_n) \end{pmatrix}^\top - \mathbf{W}^\top \mathbf{s}_{n,m}||_2^2 \\ &+ \frac{1}{2} \sum_{n=1}^{N-1} (\cos(\alpha_n)\cos(\alpha_{n+1}) + \sin(\alpha_n)\sin(\alpha_{n+1}) - \cos(\measuredangle l_n l_{n+1}))^2 \end{aligned} \tag{5.5}$$

The partial derivatives of cost function 5.5 are equal to those of the cost function 5.1 except for the partial derivative of α_n. It is given by the following equation:

$$\begin{aligned} \frac{\partial C}{\partial \alpha_n} &= \sum_{m=1}^{M} d_{n,m} \begin{pmatrix} sin(\alpha_n) & -cos(\alpha_n) \end{pmatrix} (\mathbf{W}^\top \mathbf{s}_{n,m} - \mathbf{o}_n) \\ &+ \begin{cases} \sin(\alpha_n - \alpha_{n+1})(\cos(\measuredangle l_n l_{n+1}) - \cos(\alpha_n - \alpha_{n+1})), & \text{if } n = 1 \\ \sin(\alpha_n - \alpha_{n+1})(\cos(\measuredangle l_n l_{n+1}) - \cos(\alpha_n - \alpha_{n+1})) & \\ + \sin(\alpha_{n-1} - \alpha_n)(\cos(\alpha_{n-1} - \alpha_n) - \cos(\measuredangle l_{n-1} l_n)), & \text{otherwise} \end{cases} \end{aligned} \tag{5.6}$$

A solution for the parameters $\theta = (\alpha_n, \mathbf{o}_n, \mathbf{W})$ can be obtained performing gradient descent on the cost function C with respect to θ. The update in an iteration step t is given by:

$$v_t = \gamma v_{t-1} + \beta \frac{\partial C}{\partial \theta} \tag{5.7}$$

$$\theta_t = \theta_{t-1} - v_t \tag{5.8}$$

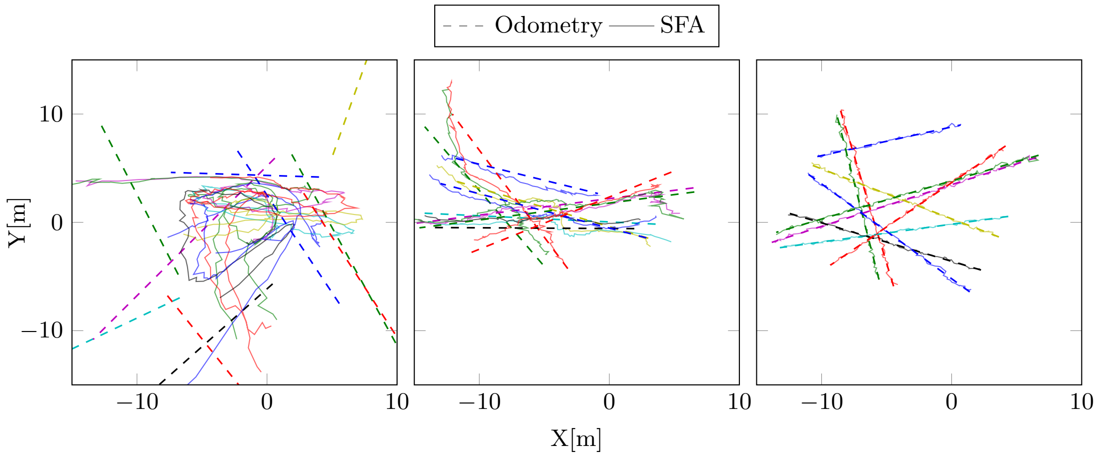
Where β is the learning rate and $\gamma \in (0, 1]$ is a momentum term to increase speed of convergence and regulate the amount of information from previous gradients which is incorporated into the current update.

Note that the found solutions may be translated and rotated against the odometry's coordinate systems and for cost function C it may also be mirrored. If required, some parameters $\mathbf{o}$ and α can be fixed during the optimization to be compatible with the desired coordinate system or by rotating, shifting, and mirroring the solution as a post-processing step.

Experiments

To perform the unsupervised metric learning it is assumed that the slow feature representation of the environment has been learned in advance. The optimal parameters are obtained by minimizing the cost functions given in equations (5.1) and (5.5) using the distances and slow feature vectors sampled along the line segments. The choice of a specific cost function is dependent on the actual trajectory of the robot. The optimization terminates if either the number of maximum iterations is reached or the change in the value of the cost function falls below a threshold. To assess the quality of the learned metric mapping the localization accuracy was measured on a separate test set as the mean Euclidean distance (MED) from the ground truth coordinates. As a reference, the results of training a regression model directly on the ground truth coordinates have been computed as well. We used the eight slowest features as input to the optimization. A nonlinear expansion of the slow features to all monomials of degree 2 yields a 45-dimensional representation, which slightly increases localization accuracy. The number of unknown parameters per line is $1 + 2$ (scalar α, 2D line origin $\mathbf{o}$). Additionaly, the weights $\mathbf{W}$ defining the mapping from slow feature to metric space for two dimensions x, y with 245 dimensions are unknown parameters. Since there is no point of reference between both coordinate systems the estimated coordinates might be rotated and translated. Therefore, the ground truth and estimated metric coordinates have to be aligned before calculating the accuracy. We used the same method as in the experiments in sections 5.2 which is described in section 4.2.1 to obtain the rigid transformation which rotates and translates the estimated coordinates to align them with the ground truth coordinates. The obtained transformation was then again applied to the estimations from our separate test set.

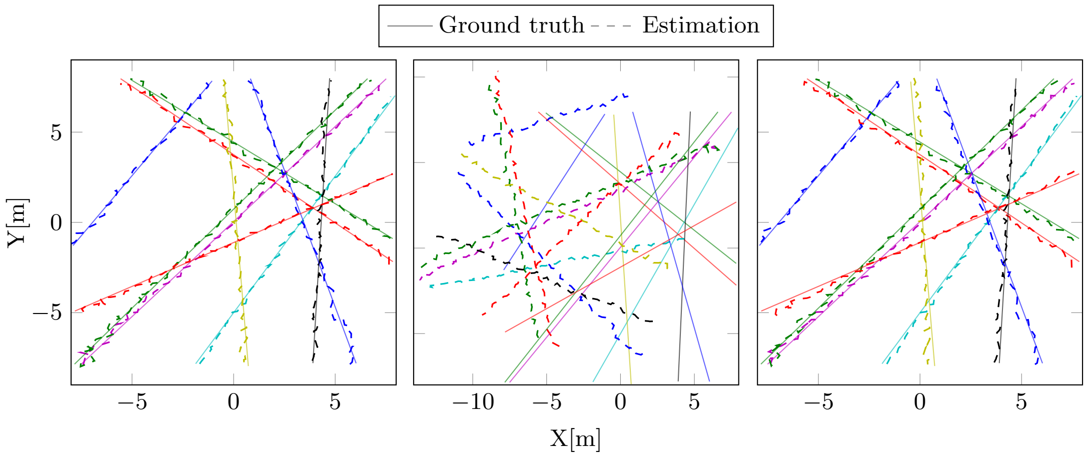
Simulator Experiment The approach was first validated in a simulated garden-like environment created with Blender according to the description in 4.1. The spatial representation was learned by training the SFA-model with 1773 panoramic RGB-images with a resolution of 600 × 60 pixels from a trajectory that covers an area of 16 × 18 meters. Training data for the unsupervised metric learning was gathered by driving along 10 straight line segments with a random orientation and sampling the slow feature vector $\mathbf{s_{n,m}}$, the distance measurement $d_{n,m}$ and the corresponding ground truth coordinates $(x_{n,m}, y_{n,m})^\top$ in 0.2 meter steps. In total 862 points were collected. Start and end points of the line segment were restricted to be within the training area and to have a minimum distance of 8 meters. The parameters for the optimization were initialized with values from a random uniform distribution such that $\alpha \in [0, 2\pi)$, $\mathbf{o}_n^x \in [-8, 8]$, $\mathbf{o}_n^y \in [-9, 9]$ and $\mathbf{w}_j^x$, $\mathbf{w}_j^y \in [-1, 1]$. The learning rate was set to $\beta = 1 \times 10^{-5}$ and the momentum term to $\gamma = 0.95$. The partial derivatives of the cost function are computed according to equations (5.3)-(5.4).



(a) Initial random parameters **(b)** Optimization step 500 **(c)** Optimization step 1000

Figure 5.19: Illustration of the optimization process. (a) Point estimations of the individual line segments resulting from randomly initialized parameters α_n, $\mathbf{o}_n$, $\mathbf{W}$. Each line segment is indicated by a specific color where the point estimations based on the line parameters and odometry are depicted as dotted and the point estimations based on the weight vectors are depicted as solid lines. (b) After 500 iterations the lines resulting from both estimations have converged to similar positions. (c) At iteration 1000 the estimated lines have adjusted their spatial layout.

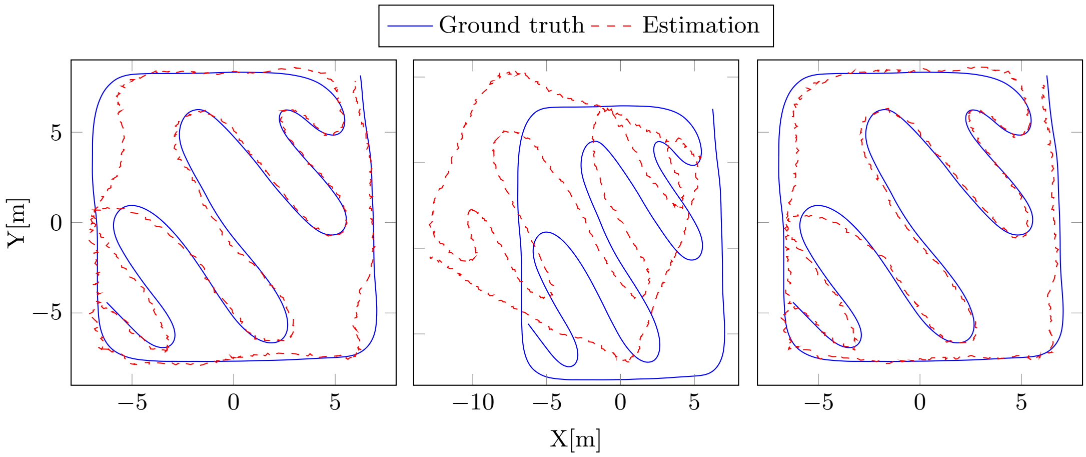
Results The optimization terminated at about 1000 iterations where the change in the value of the cost function fell below the predefined threshold. During the optimization process the points from line pairs of the odometry and slow feature estimations started from random associations in the beginning to converge to similar positions while later on in the process the line estimations adjusted their spatial layout. An illustration of the optimization is shown in Fig. 5.19. For the line segments from the training data the MED from ground truth to the estimated coordinates from the supervised regression model amounts to 0.13 meters. Applying the learned weight vectors from the unsupervised metric learning to the slow feature vectors results in an error of 0.17 on the training set. The supervised and unsupervised estimations for the line segments from the training set are illustrated in Fig. 5.20. Using the weights learned with the supervised regression



(a) Supervised **(b)** Unsupervised **(c)** Unsupervised aligned

Figure 5.20: Comparison of supervised and unsupervised regression for the training data. (a) Estimated coordinates from the supervised regression model are close to the ground truth with a MED of 0.13 meters. (b) Estimations resulting from the unsupervised learned regression weights are rotated and translated with respect to the ground truth coordinates since there is no fixed point of reference. (c) After aligning the estimated coordinates to the ground truth the MED amounts to 0.17 meters.

model to predict coordinates on the separate test trajectory results in a MED of 0.39 meters from the ground truth coordinates. With the unsupervised model the predictions are closer to the ground truth with a MED of 0.36 meters. Predicted trajectories from both models closely follow the ground truth coordinates with noticeable deviations in the south-eastern and north-western part. Considering the line segments from the training data it is apparent that those regions are only sparsely sampled while the density is much higher in the middle-eastern part. The lower accuracy of the supervised regression model might thus be due to slightly overfitting the training data. The estimated trajectories of the supervised and unsupervised regression models are shown in Fig. 5.21.

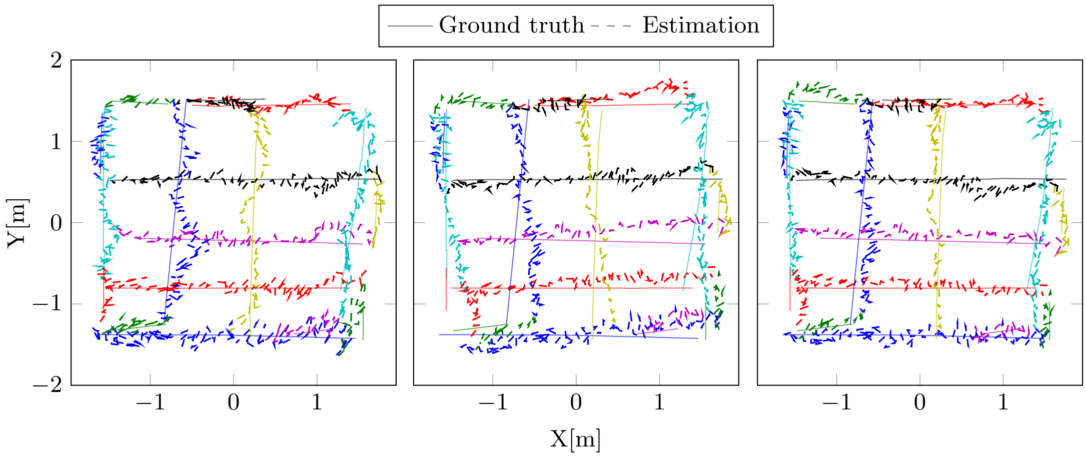


(a) Supervised **(b)** Unsupervised **(c)** Unsupervised aligned

Figure 5.21: Comparison of supervised and unsupervised regression for the test data. (a) Estimated trajectory for the test data using the supervised regression model yields an error of 0.39 meters. (b) The estimations of the unsupervised regression model are not located within the same coordinate system as the ground truth coordinates. (c) Applying the aligning transformation obtained for the training data to the test set results in an error of 0.36 meters. Both estimations closely follow the true trajectory while the supervised model seems to be overfitted on the training data.

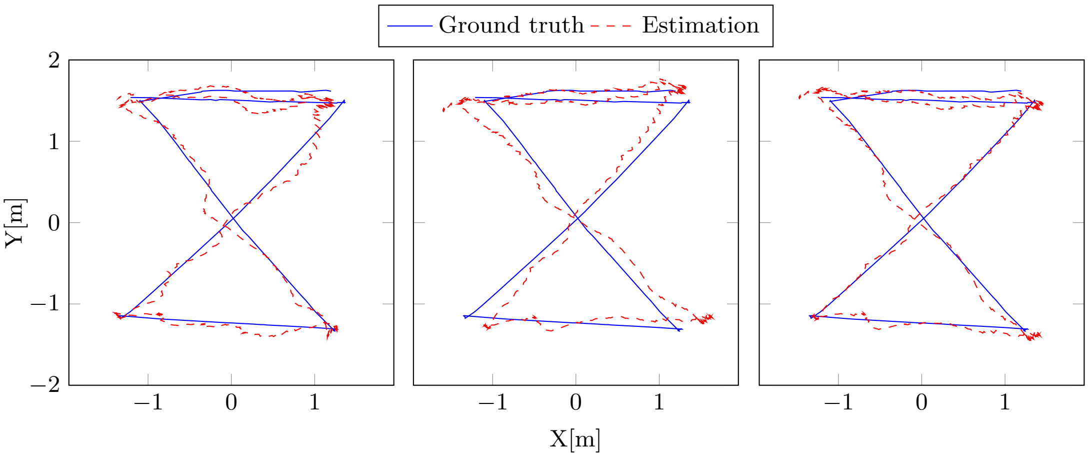
Real World Experiment For the real world experiments the same recordings as in the indoor experiment from section 5.2.2 have been used where the odometry measurements from the robot have been logged together with the image data. To obtain the training data for the unsupervised metric learning we used the same SFA-model, which was already learned on the training sequence, to compute the slow feature outputs for every point on the trajectory. The movement pattern of the robot was to drive along straight lines, turning on the spot and driving along the next straight line. Thus, the points on the trajectory could easily be split into line segments based on the translational and angular velocity measurements from odometry. There were 18 line segments created consisting of a total of 1346 points. In order to speed up the process and support convergence to an optimal solution the line parameters α_n, $\mathbf{o}_n^x$, and $\mathbf{o}_n^y$ have been set to the corresponding odometry position estimates which are freely available. The weights $\mathbf{w}^x$ and $\mathbf{w}^y$ were initialized with the weights from regression models fitted to the odometry estimations. As in the simulator experiment the learning rate is set to $\beta = 1\times10^{-5}$ and the momentum term to $\gamma = 0.95$. Due to the grid like trajectory with intersection angles all being around 90° the cost function from equation (5.5) was used for the optimization.

Results The optimization ran for about 900 iterations until it converged to a stable solution. The prediction accuracy of the unsupervised model on the training data amounts to 0.17 meters after estimations have been aligned to the ground truth coordinates. The supervised model which was trained directly with the ground truth coordinates achieved an accuracy of 0.12 meters. An illustration of the estimations for the training data from both models is shown in Fig. 5.22. Estimations from both models for the test data are equal to a MED of 0.14 meters. The resulting predicted trajectories along with the ground truth trajectories are shown in Fig. 5.23.



(a) Supervised **(b)** Unsupervised **(c)** Unsupervised aligned

Figure 5.22: Comparison of supervised and unsupervised regression for the training data. Each straight line segment is illustrated in a separate color where the solid lines represent the ground truth and dotted lines indicate the estimations of the respective models. (a) The predictions of the supervised model have an error of 0.12 meters closely following the ground truth. (b) The raw predictions of the unsupervised model are slightly rotated and shifted w.r.t. the ground truth coordinates. (c) Aligning the estimations from the unsupervised model to the ground truth coordinates results in a MED of 0.17 meters.



(a) Supervised **(b)** Unsupervised **(c)** Unsupervised aligned

Figure 5.23: Comparison of supervised and unsupervised regression for the test data. (a) The trajectory predicted from the supervised model deviates by 0.14 meters on average from the ground truth trajectory. (b) The raw estimations from the unsupervised model are rotated and translated w.r.t. the ground truth trajectory. (c) Transforming the estimations from the supervised model by the rotation and translation estimated for the training data results in a MED of 0.14 meters from the ground truth.

5.3.2 Fusion of SFA Estimates and Odometry in a Probabilistic Filter

In our scenario the robot has access to relative motion measurements from odometry and absolute measurements from the SFA-model in order to localize itself. Even though the odometry measurements are locally very precise small errors accumulate over time and the belief of the own position starts to diverge from the true position. The estimations from the SFA-model on the other hand have a higher variability but are absolute measurements and thus allow to correct for occurring drift. A mapping function from slow feature to metric space enables the combination of both measurements in a common coordinate system. To achieve the highest possible accuracy the combination needs to be optimal considering the uncertainties of both measurements. For linear systems the Kalman Filter [64] is the optimal estimator that combines the information of different uncertain sources to obtain the values of interest together with their uncertainties. A state transition model is used to predict the value for the next time step and a measurement model is used to correct the value based on observations. The state of the system is represented as a Gaussian distribution. However, for our scenario of mobile robot localization the state transition model involves trigonometric functions which lead to a nonlinear system. The Extended Kalman Filter (EKF) linearizes the state transition and measurement model around the current estimate to ensure the distributions remain Gaussian. Then, the equations of the linear Kalman Filter can be applied. It is the standard method for the problem of vehicle state estimation [66] and is also applied in Visual SLAM [26]. Here, we use the Constant Turn Rate Velocity (CTRV) model [83] as the state transition model for the mobile robot. The measurement model incorporates the absolute estimations resulting from the mapping of the current slow feature outputs to metric coordinates.

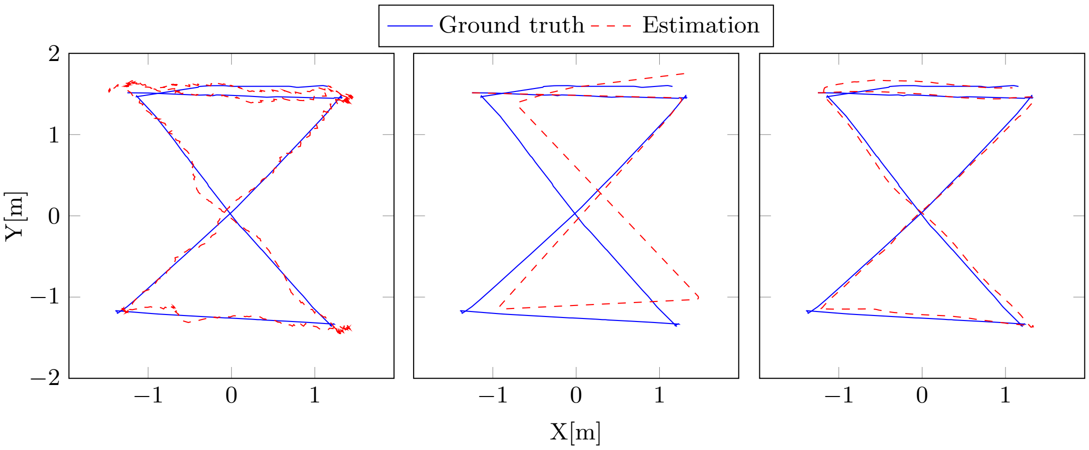
Real World Experiment

To test the localization performance with the EKF we used the test set from the experiments in section 5.3.1. The absolute coordinate predictions from slow feature outputs were computed using the unsupervised learned regression weights from the corresponding training data. They are used as input for the measurement model while the odometry readings are used as input for the state transition model of the EKF. The values for the process and measurement noise covariance matrices were chosen based on a grid search. For the experiment we assumed that the robot starts from a known location and with known heading which is a valid assumption considering that many service robots begin operating from a base station.

Results As expected, the estimated trajectory of the EKF shows an improvement over the individual estimations since it combines their advantages of global consistency and local smoothness. The accuracy of the predicted trajectory from the SFA-model is 0.14 meters while progression of consecutive points is rather erratic. The trajectory resulting from odometry measurements is locally smooth but especially the errors in the orientation estimation lead to a large divergence over time resulting in a MED deviation of 0.31 meters from the ground truth coordinates. The accuracy of the trajectory estimated from the EKF amounts to 0.11 meters which is an improvement of 3 centimeters or 21% on average compared to the accuracy obtained from the SFA-model. Resulting trajectories from all methods are illustrated in Fig. 5.24.

5.3.3 Discussion

The presented method for unsupervised learning of a mapping from slow feature outputs to metric coordinates was successfully applied in simulator and real world experiments and achieved accuracies in the same order of magnitude as a supervised regression model trained directly on ground truth coordinates. Since it only requires odometry measurements and imposes reasonable constraints on the trajectory, it can be applied in real application scenarios where no external ground truth information is available. The learned metric mapping function enables the visualization of the extracted slow feature representations, the trajectories of the mobile robot and the fusion of SFA estimates and odometry measurements using an Extended Kalman Filter. Thereby, the already



(a) SFA **(b)** Odometry **(c)** EKF

Figure 5.24: Fusion of SFA estimates and odometry using an Extended Kalman Filter. (a) The accuracy of the localization achieved with the SFA-model is 0.14 meters. Due to the absolute coordinate predictions the progression of the trajectory is rather erratic. (b) Measurements from odometry are locally accurate but increasingly diverge over time. The MED from ground truth amounts to 0.31 meters. (c) The EKF filter combines the strength of both estimations resulting in an accuracy of 0.11 meters.

competitive localization accuracy of the SFA-model improved further by 21%. The precision of the resulting metric mapping, and hence also the localization accuracy, might benefit from using visual odometry [117, 133, 45] instead of wheel odometry since it is not affected by wheel slippage.
Although localization and navigation can be performed directly in slow feature space the learned mapping from SFA-outputs to metric space improves performance, transparency and allows better integration with other methods and services based on metric representations.

5.4 Landmark Based SFA-localization

Place cell firing behavior in a rat's brain is strongly driven by visual input. More specifically, they seem to respond to distinctive visual cues or landmarks. It has been shown in experiments that rotation of a distinctive visual cue correspondingly leads to a rotation of place cell activity [108]. A single distinctive landmark with a sufficiently rich texture and 3D structure would be enough to determine the own position and orientation in the environment relative to this landmark. It can be assumed that the presented SFA-model, trained on a sequence of whole images, learns to extract such distinctive landmarks as well in the higher layers of the network. Here we propose to identify a specific landmark in a sequence of images and train an SFA-model on the extracted landmark views to learn spatial codes. Compared to the whole image approach employed so far the complexity of the training process is greatly reduced since invariance w.r.t. the orientation of the robot does not need to be learned. Instead the orientation can be explicitly removed during the preprocessing by aligning all marker views to a common orientation. Additionally, using several landmarks would allow to deal with local occlusions which generally is a problem for whole image based approaches.

In [43] the authors have demonstrated that a hierarchical SFA-network learns representations of an object's position and rotation if it is trained on image sequences featuring object views under varying transformations. If one considers the projection of 3D points to the image plane, the result is equivalent if either the 3D points are transformed by matrix $\mathbf{M}$ or the inverse transformation $\mathbf{M}^{-1}$ is applied to the camera. Hence, representations learned with an hierarchical SFA-model on landmark views can be used to reconstruct the object's as well as the camera pose.

Obtaining the landmark views requires to detect and localize an object in the images. Recent deep learning based approaches for object localization, e.g. [129, 127, 128], are able to detect and localize up to 1000 different object classes. For a first proof of concept, however, we adapted the marker detection described in chapter 4.2.1 to omnidirectional images and used the marker views to learn representations of a mobile robot's position.

5.4.1 Experiments

For the experiments we used the simulated environment created with Blender which is described in section 4.1. Two binary visual markers, with ids 136 and 144, were placed in the simulator environment to serve as easily detectable landmarks. Once a marker is detected its angle φ within the omnidirectional image in polar coordinates can be computed by $\varphi = \arctan2(v, u)$, where (u, v) are the image coordinates of the marker's origin w.r.t. the image center. All marker views were then aligned to a common orientation based on their current angle φ. The size of the marker view is defined by the bounding box with an additional space of 50 pixels in each dimension. The preprocessing and extraction is illustrated in Fig. 5.25. Enlarging the size of the image region allows the SFA-model to incorporate background information into the learning process. Training the SFA-model on the raw marker views led to degenerated solutions were only the distance to the marker was encoded properly but the orientation information, i.e. from which side the marker was observed, was mirrored along the markers upward pointing coordinate axis. This pose ambiguity was also observed for pose estimation of visual markers based on projective geometry [151, 152], especially in the case of near frontal views. Thus, including background information helps the SFA-model to resolve this ambiguity.

After the extraction of the marker views they have been resized to a resolution of $120 \times$

120 pixels. The SFA-model used in the experiments is a four layer model similar to those in the previous localization experiments but with a smaller dimension of the input layer. The evaluation of the learned spatial representations was done by training a regression model on the SFA-outputs and the corresponding ground truth coordinates and measuring the mean Euclidean distance (MED) on a separate test set. The training set consist of 1773 and the test set of 1190 images.

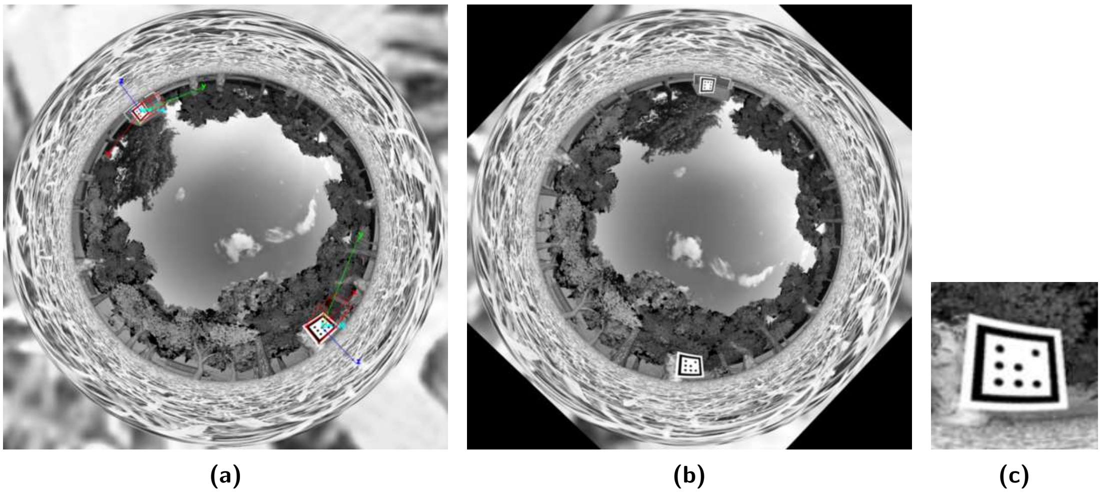


Figure 5.25: Extraction of the marker views. (a) Illustration of the result for the detection process of marker with id 136. (b) The orientation of the marker in polar coordinates is used to align all marker views to common orientation. Thus, orientation invariance does not need to be learned. (c) The image region assigned to a marker view is based on the size of the bounding box which is extended by an amount of 50 pixels in each dimension. All marker views are resized to a resolution of 120 × 120 pixels.

Localization With a Single Landmark

First we investigated the quality of the learned SFA-representations for the individual markers. For the training and evaluation of the SFA- and the regression-model, only samples with valid marker detections have been used. The detection rate for marker 136 was 97% in the training and 99% in the test run. For marker 144 the detection rate was 95% and 96% for the training and test run, respectively. The trajectories and the missed detections of the individual markers are illustrated in Fig. 5.26.

Results The spatial firing maps of the two slowest SFA-outputs for both markers show clear gradients along the coordinate axes while higher oscillating modes and mixtures can be seen for SFA-outputs three and four. The localization accuracy amounts to 0.36

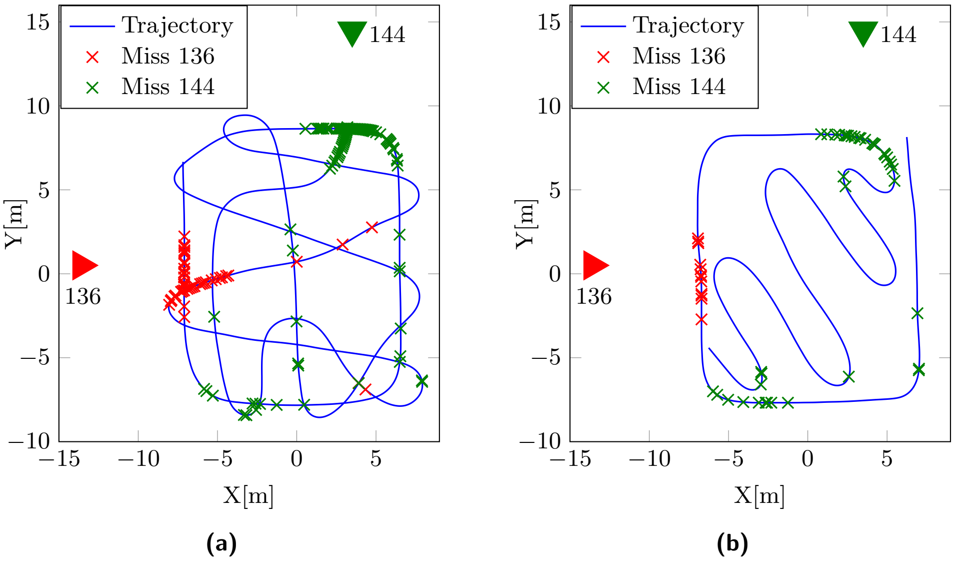


Figure 5.26: Marker visibility for the train and test run. Trajectories for the training and test run are indicated by the blue line. Coordinates on the trajectory where no marker was detected are indicated by crosses. (a) The detection rate in the training run is equal to 97% for marker 136 and 95% for marker 136. Many of the missed detections are in close distance to the markers where the radial distortions had a negative impact on the detection performance. (b) The same effect can be observed on the test trajectory. Nevertheless, marker 136 is still detected in 99% of the images and marker 144 in 97%.

meters for marker 136 and to 0.43 meters for marker 144 on the test trajectory. The spatial firing maps and estimated trajectories are shown in Fig. 5.27.

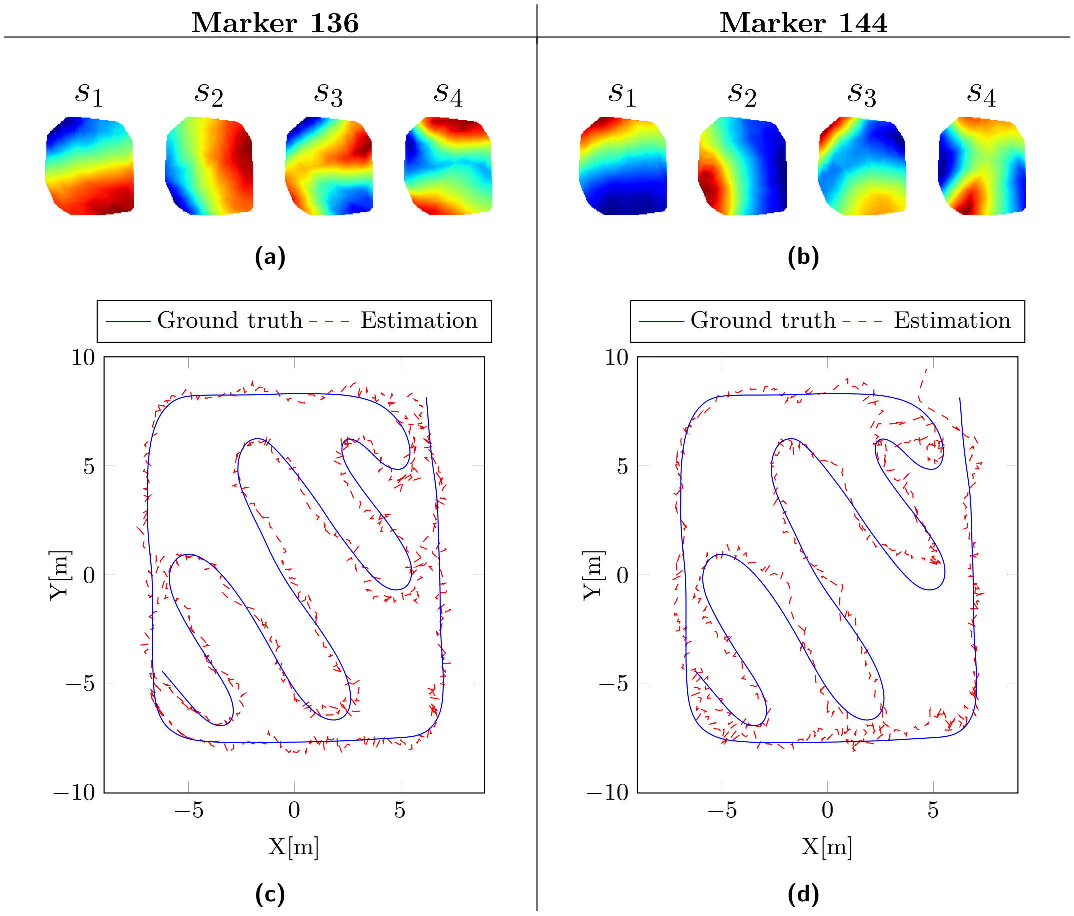


Figure 5.27: Localization results for single markers. The spatial firing maps of marker 136 (a) and marker 144 (b) show clear gradients along the coordinate axes for the first two SFA-outputs and thus suggest strong spatial coding. Spatial firing maps of SFA-outputs three and four show higher modes and mixtures of first two outputs. The MED between the estimations for the test trajectory and ground truth amounts to 0.36 meters for marker 136 (c) and to 0.43 meters for marker 144 (d).

Localization With Two Landmarks

Although the previous experiment has shown that accurate localization is possible using a single landmark a gain in performance can be expected when combining both SFA-models. Landmarks can be combined by simply averaging the outputs of the individual regression models or training another SFA on the combination of slow feature outputs from the individual models. However, in our experiments best results were achieved by stacking the first eight slow feature outputs of the individual models and training a regression model with quadratically expanded vectors stacked together. Since the marker detection is not perfect there were frames where only a single marker was detected. In these cases the corresponding values in the feature vector were set to zero. To facilitate learning of the regression model we also added a binary flag for every marker to the feature vector which indicates valid detections.

Results The localization accuracy for the prediction of the test trajectory is 0.21 meters which is an improvement of 42% compared to the predictions based on the individual markers alone. For comparison: an SFA-model trained with the full images and additional simulated rotation yields an accuracy of 0.23 meters. The resulting estimated trajectory is shown in Fig. 5.28.

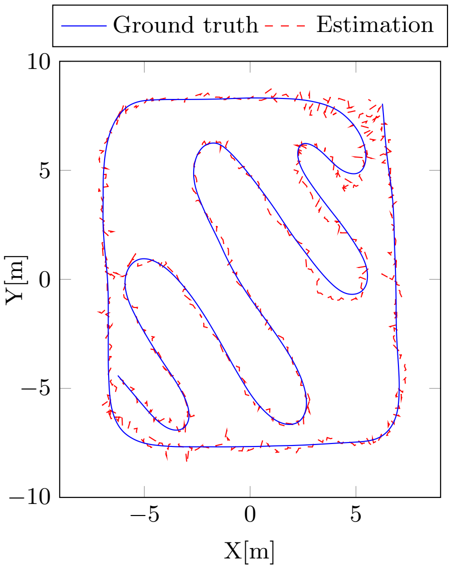


Figure 5.28: Localization result for two markers. Using the combination of the slow feature outputs from the individual models for training a regression model yields an localization accuracy of 0.21 meters.

Localization With Two Landmarks and Occlusions

In the ideal case a landmark is visible from every position in the environment. In real world application scenarios, however, a landmark might be occluded by other objects for longer periods of time. The area within the environment where a landmark can not be observed is thus not contained in the SFA-representation learned from the corresponding landmark views. However, the area might be encoded in the SFA-representation of another model trained with views of a different landmark. Depending on the size of the area where a landmark is not visible the spatial gap between the training sample from before entering and the one after leaving the area might become large. This leads to deviations from the theoretical optimal solutions, where a constant velocity and evenly distributed samples are assumed [42], which further increases the complexity of the learning problem. To investigate this effect occlusions were simulated by defining a coordinate range where the extracted marker views were excluded from training. For marker 136 positions with x- and y-coordinate smaller than zero and for marker 144 positions with x- and y-coordinates greater than zero were discarded. Thereby, an occlusion was simulated within 25% of the training area for each of the two markers. The visibility of the markers is shown in Fig. 5.29. We first investigated the quality of the individual models and afterwards their combination by stacking their slow feature outputs to a common feature vector.

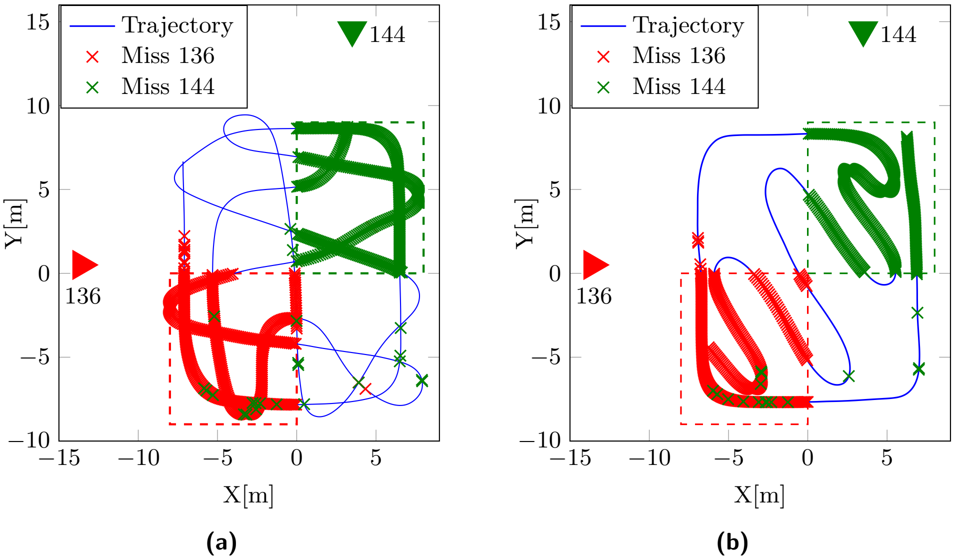


Figure 5.29: Marker visibility for the training and test run with occlusions. The blue line indicates the trajectory. The crosses indicate positions on the trajectory where no marker was detected. The dotted lines illustrate the areas where an occlusion of the specific marker was simulated. (a) The detection rates for the training run are 77% for marker 136 and 71% for marker 144. (b) For the test run the Marker 136 is detected in 73% and marker 144 is detected in 71% of the images.

Results The spatial firing maps from both markers show spatial coding but are slightly disturbed compared to the ideal solutions which is a result from the spatial gaps in the training sequences. Therefore, the complexity of the learning problem is increased which is also reflected in a decreasing localization accuracy. The MED for the predicted test trajectory amounts to 0.67 meters for marker 136 and 0.50 meters for marker 144. The spatial firing maps and the estimated trajectories are illustrated in Fig. 5.30. Note that only points were considered where a valid detection was available. Using the combined slow feature outputs of the individual models results in a an accuracy of 0.32 meters

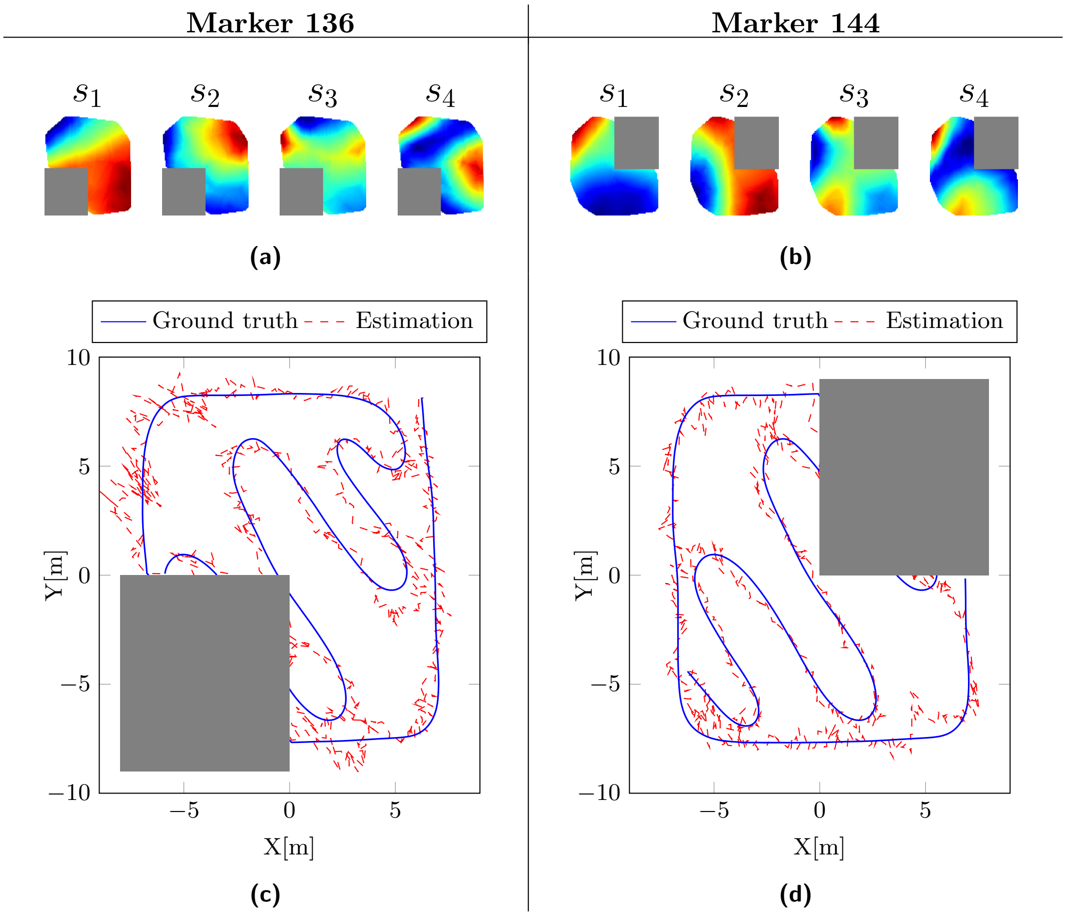


Figure 5.30: Localization results for single markers with occlusions. The light gray rectangles illustrate the area where an occlusion of the specific marker was simulated. The spatial firing maps of marker 136 (a) and marker 144 (b) suggest spatial coding but the gaps in the training trajectory lead to larger temporal derivatives and thus to deviations from the optimal solutions. The accuracy of the estimated trajectory amounts to 0.67 meters for marker 136 (c) and 0.50 meters for marker 144 (d).

which is an improvement of 36% compared to the individual predictions. The estimated trajectory from the combined regression model is shown in Fig. 5.31.

5.4.2 Discussion

The SFA-model successfully extracted spatial representations from a training sequence of single landmark views in a simulated environment enabling a precise localization. Additional landmarks were integrated by stacking the SFA-outputs into a common feature vector to train a regression model for coordinate prediction. For two landmarks and moderate detection miss-rates the combination of both SFA-models even surpasses the

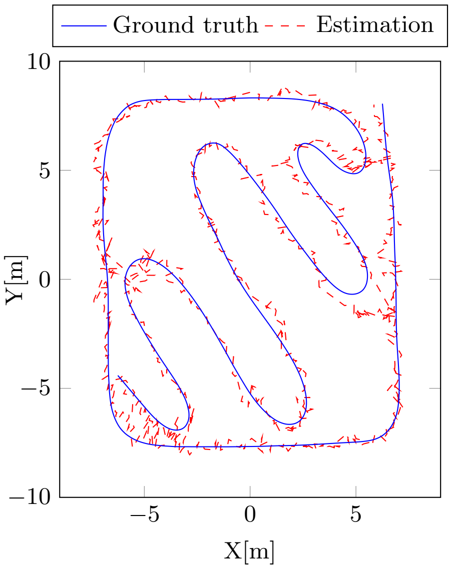


Figure 5.31: Localization results for two markers with occlusions. The estimation based on the combination of the slow feature outputs from the individual models improves the prediction accuracy by 32%. The MED from ground truth is 0.32 meters.

performance of a model trained with whole images. At the same time the training complexity is largely reduced due to the orientation invariant representation of the marker views. To account for miss detections of a single marker the corresponding slow feature values have been set to zero and a binary indicator flag was added to the feature vector to facilitate training of the regression model. The combination of landmark views furthermore enables localization in cases where individual landmarks are not observable from extended spatial regions within the area. Transferring the approach to real world outdoor scenarios would require a robust object detector. Current deep learning approaches for object detection and localization [127, 128, 129] could be used for this purpose by applying them to a sliding window over the unwarped panoramic images. However, for the target scenario of a garden environment the set of appropriate landmark objects is restricted. Many objects that are usually present in a garden are not stationary, e.g. people, animals or furniture, and other objects are not specific enough or might have high variability in their appearance, e.g. trees and plants. Therefore, the careful selection and reliable identification of landmarks would be crucial for a successful application.

5.5 Conclusion

We presented a biologically motivated model for visual self-localization based on the principle of slowness learning. The model extracts spatial representations of the environment by directly processing raw high-dimensional image data in a hierarchical SFA-network employing a single unsupervised learning rule. The use of an omnidirectional vision system allows to learn orientation invariant representations of the location by modifying the perceived image statistics through additional simulated rotational movement. The resulting SFA-outputs encode the position of the camera as slowly varying features while at the same time being invariant to its orientation. We demonstrated the feasibility of the approach in a simulated environment and compared its performance to state of the art visual SLAM-methods in real world indoor and outdoor experiments. Despite its simplicity, the presented experiments have proven that the learned SFA representations enable a precise localization obtaining accuracies that are on par or even superior compared to state of the art SLAM methods. Integrating odometry and SFA estimates in a probabilistic framework has shown further improvements in localization accuracy and smoothness of the resulting trajectories. The presented method for the unsupervised learning of a mapping from slow feature to the metric space enables the odometry integration in real world application scenarios. An alternative approach for learning spatial representations based on tracked landmark views was proposed. The achieved localization performance is comparable to the one obtained with a model trained on whole images. Additionally, the complexity of the learning problem is reduced since the marker views can be made invariant w.r.t. in plane rotations of the camera by a simple preprocessing step. A further benefit is the capability to deal with local occlusions. However, the transfer of the approach to real world outdoor scenarios would require a method for the identification of suitable landmarks and robust object detectors and thus is beyond the scope of the thesis.

6 Robust Environmental Representations

This chapter deals with the problem of robust long-term localization in open field outdoor environments. The presented model based on hierarchical Slow Feature Analysis (SFA) enables a mobile robot to learn orientation invariant representations of its position directly from images captured during an initial exploration of the environment. The underlying assumption is that the information about the robot's position is embedded in the high dimensional visual input and that it changes slowly compared to the raw sensor signals. Learning an optimal encoding of the robot's position as well as performing precise localization based on the raw visual input requires a static environment. In this scenario only the spatial configuration of the robot (x, y, φ) changes over time, constituting the complete latent space of the perceived visual input.

In real world application scenarios, however, the environment can not be assumed to be static. If there exist environmental variables that change on a slower or equal timescale than the position of the robot during training these variables will be encoded by the learned functions since the SFA algorithm seeks to minimize the temporal variation of the output signals. Hence, the first learned functions might encode rarely occurring events or gradual changes, e.g. doors or curtains that are opened/closed or illumination changes resulting from the transition from sunny to cloudy sky. Depending on the concrete timescale, these slowly changing environmental variables will interfere with the spatial coding to different degrees.

Furthermore, as the proposed SFA-model directly processes the raw pixel values, reliable localization requires that the statistics of the sensory input data are similar to the training phase. However, in real world outdoor scenarios the appearance of a place will inevitably change over time. Considering short timescales, the appearance of the environment might change due to dynamic objects or a change in lighting or weather conditions. Over longer periods of time the appearance of a place might vary due to structural scene changes and seasonal effects on vegetation.

These appearance changes of the environment induce high visual diversity into images of the same place visited at different times. This poses a severe challenge for any vision based localization and mapping method and different approaches towards long-term autonomy have been proposed recently. Invariance w.r.t. lighting changes, as a part of the overall problem, has been tackled by optimizing the exposure time of the camera for visual odometry [165], shadow invariant imaging [93] and methods for learning illumination invariant visual feature descriptors [18, 79]. Instead of constructing a single map of the environment several authors proposed methods for constructing and maintaining multiple representations to capture the diversity of appearances in different conditions [27, 72, 19, 100]. Although the author demonstrated improved long-term robustness the memory demands and the complexity is greatly increased. Additionally, the parameters needed to control map maintenance might need adaption for specific environments and the method might fail in case of drastic appearance changes that prevent linking the current sensor measurements to the existing representation. Temporal integration and occurrence statistics of visual features over multiple recordings along the same trajectory have been used in [61, 62, 65]. The feature based approaches are viewpoint invariant to some degree but struggle with severe appearance changes [158, 102]. However, the modeled feature statistics are environment specific and require several runs before reliable localization can be achieved. Methods based on image sequence matching have been shown to enable visual localization even under severe appearance changes [102, 123, 111]. First, the images are transformed to a more robust representation by a down-sampling step and patch normalization or using the features computed with a pre-trained convolutional neural network. Then, instead of trying to find a single global best match the sequence with the minimal cost is identified. Despite the impressive results the proposed approaches are restricted to localization along a given trajectory and thus not suitable for open field scenarios. Another direction of research is the translation between images captured in different conditions. In [114] the authors learn a visual dictionary using super-pixels from aligned images showing the same place in different distinct conditions. Linear regression is used in [87] to transform images from morning to afternoon targeting at illumination variance over the course of a day. In [84] the authors train coupled Generative Adversarial Networks to translate between images from different seasons. Although the methods produce reasonable results the identification, management and the learning of a translation for new conditions has not been investigated so far. Finally, several approaches have been proposed that use features from pre-trained deep convolutional neural networks for place recognition [149, 150, 4]. Features extracted from different layers have been shown to be invariant w.r.t. to viewpoint and condition in varying degrees. However, the computation and matching of the high dimensional features is computationally demanding and thus not well suited for the application on a mobile robot platform.

In this chapter we tackle the problem of learning robust representations of the environment that enable a mobile robot to robustly localize itself in open field scenarios using visual input from a camera only. In the next section we first investigate the long-term robustness of local visual features which are commonly used in SLAM methods but could also serve to create alternative image representations that can be used with the presented SFA-model. Based on these findings we then propose a generic approach to improve long-term mapping and localization robustness by learning a selection criterion for long-term stable visual features that can be integrated into the standard feature processing pipeline [1]. In section 6.2 we introduce a unified approach towards long-term robustness that is solely based on SFA [2]. It takes advantage of the invariance learning capabilities of SFA by restructuring the temporal order of the training sequence in order to promote robustness w.r.t. short- and long-term environmental effects.

6.1 Robustness of Local Visual Features

Local visual features are commonly used in the context of visual odometry [117, 134, 71] and SLAM [147, 22, 109] to estimate the motion of a camera from feature correspondences between consecutive frames and to create a sparse feature map of the environment. The standard feature processing pipeline consist of feature detection, description and matching. Feature detection is the process of identifying distinct image regions, usually corners [53, 140, 130] or blobs [92, 86, 10], which can be accurately localized and robustly re-detected under slight changes of illumination and viewpoint. After the feature detection step a descriptor is created from the surrounding image patch. Gradient based descriptors [86, 97, 10] accumulate gradient information over a quantized range of orientations in a histogram. Several histograms are computed over a predefined grid and are subsequently concatenated to obtain the descriptor. Recently, several methods have been proposed for creating binary descriptors from pixel-wise intensity comparisons within the features' image patch [17, 131, 82, 2], mainly differing in the selection pattern of pixel pairs. The binary descriptors are faster to compute and require less memory while at the same time achieving a similar performance compared to the gradient based descriptors [57]. Correspondences between the same features in different images are established by a nearest neighbor search in descriptor space using either the Euclidean distance for gradient based or the hamming distance for binary descriptors.

Visual features are designed to be invariant to slight changes in viewpoint and illumination. However, due to dynamic objects, structural scene changes, lighting, weather and seasonal effects the appearance of the environment can change drastically. In long-term outdoor scenarios most of the initially detected visual features can usually only be matched for limited periods of time and the number of true positive matches might decrease drastically even after a few hours [125]. Therefore, most information in the initial feature map is likely to be valid only for short time-frames resulting in an increased probability of false positive matches. Wrongly established feature correspondences can lead to errors in the ego-motion estimation and map creation and thus prevent reliable localization. To reduce the probability of false positives the distance ratio test [86] can

[1] Thanks to Annika Besetzny for the contributions made during her Master Thesis.

[2] Thanks to Muhammad Haris for the contributions made during his Master Thesis

be applied to the first and second nearest neighbor candidates to filter out ambiguous matches. An alternative is to apply a mutual consistency check [117] where the correspondence search between two images is performed in both directions and only features which have each other as mutual match are accepted. Epipolar geometry, which describes the geometric relations of 3D points observed from two or more camera views and their image projections, can be used to perform a guided search and to reject false positive matches within a RANSAC [40] based scheme. However, the number of iterations needed to find a hypothesis grows exponentially with the number of outliers and the outlier ratio must not exceed 50% [45].

6.1.1 Evaluation of the Long-term Robustness

The performance of interest point detectors and descriptors has been evaluated in the context of visual tracking [48] and SLAM [49] but not with a focus on long-term robustness. In [158, 159] the authors investigated the feasibility of feature based topological localization in long term outdoor experiments. They found that SIFT and SURF features enable a localization rate of $80 - 95\%$ when using high resolution images and applying the epipolar constraint. However, the data set used in the experiments has been recorded on a campus area and thus contains many static scene elements like buildings and other man made objects. To investigate the performance of local visual features on image data

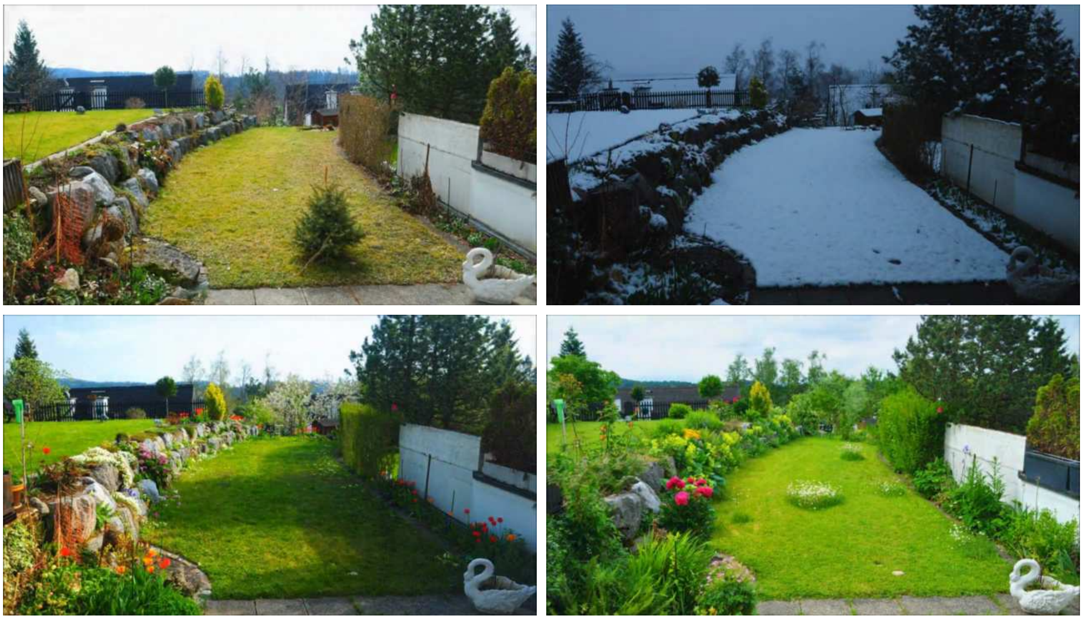


Figure 6.1: Garden time-lapse. Images taken at a regular interval with a fixed camera capture the natural variation in appearance over the seasons. *Source:* `http://www.youtube.com/watch?v=7dhT-IJmqcg&hd=1`.

according to our scenario of long-term localization in garden like environments we performed an evaluation using a time-lapse recording with a fixed camera of a garden over the course of a year (see Fig. 6.1 for example images). For the evaluation we considered SIFT [86], SURF [10] and its upright version and the FAST-detector [130] combined with the BRIEF-descriptor [17]. The upright version of SURF as well as the combination of FAST and BRIEF do not assign an orientation to detected interest points and thus are not invariant with respect to a rotation of the camera around its optical axis. However, for the scenario of a moving ground vehicle changes in the roll angle should be negligible. To evaluate the performance of the methods 1000 features have been extracted from the first frame with each of the aforementioned detector/descriptor combinations and stored as reference. Then we extracted features from every image of a proceeding sequence of 100 frames and matched them against the respective reference features. Matches have been validated using a distance threshold of 10 pixels and applying the mutual consistency check in order to prevent that multiple feature from a query image are matched to the same feature from the reference image.

The performance of the evaluated features is measured in terms of precision and recall. In addition, we also kept track of the count a certain feature from the reference frame was matched over the sequence. The measured precision and recall values of the individual features are shown in Fig. 6.2a. SIFT and the orientation invariant version of SURF performed worst. Since the image set does not feature rotational movement of the camera and the orientation assignment slightly reduces the distinctiveness of the descriptor the upright version of SURF yields a better performance. The best result was obtained with the combination of FAST and BRIEF.

The number of true positive matches over time for the best performing feature FAST/BRIEF is shown in Fig. 6.2b. While there have been about 580 valid matches found in the first frame the number of true positives steadily decreases to about 50 in the last frame where the appearance has changed drastically due to variations in the vegetation. The lowest number of matches was obtained for a frame in winter time where large parts of the scene were covered by snow. In this case only 10 features could be matched successfully. Other negative fluctuations were mainly caused by changes in lighting conditions like cast shadows or overexposed image regions.

The 50 most and least stable features are illustrated in Fig. 6.3. Stable features are mainly located at image patches that correspond to man-made objects like fences and walls, but there are also stable features located at the top of conifers as their foliage is not affected by the seasons and the strong contrast against the sky results in high responses of the interest point detectors. Features from image regions corresponding to lawn and other kinds of vegetation, which strongly varied with the seasons, are the least stable ones.

Although a number of features are persistent over a whole year the overall performance in the examined natural outdoor environment is rather insufficient for the purpose of

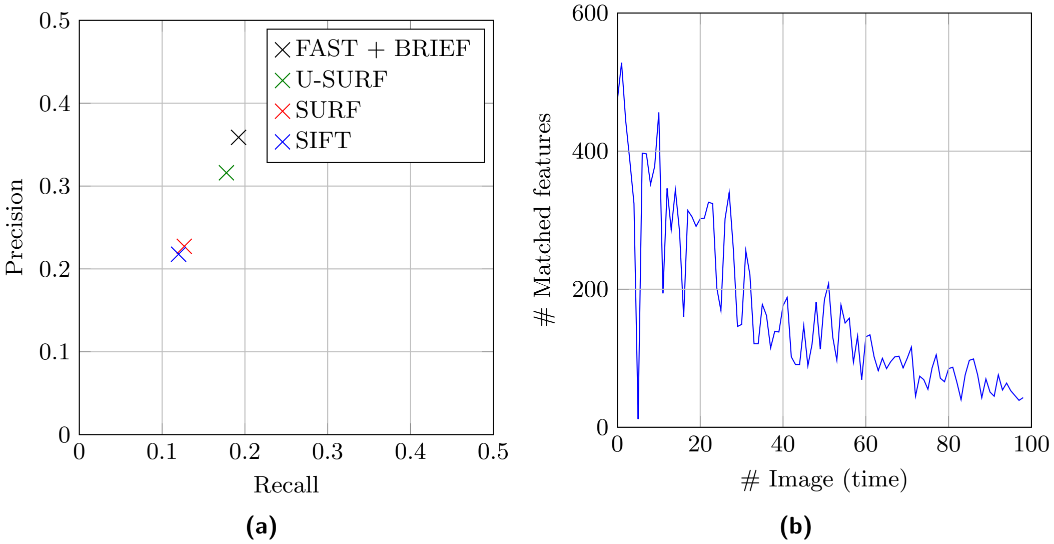


Figure 6.2: Results of the feature evaluation. (a) Precision and Recall of the evaluated features. Orientation invariant features perform worse than the ones that do not account for orientation. The combination of the FAST detector and binary BRIEF descriptor performs best while the overall performance is still rather low. (b) The number of feature matches using FAST and BRIEF steadily decreases with negative fluctuations that are mainly caused by changes in global lighting.

long-term localization. The steady decrease in the ratio of true positive matches w.r.t. the reference set would pose a severe problem for a feature based localization system as it increases the probability of false data associations.

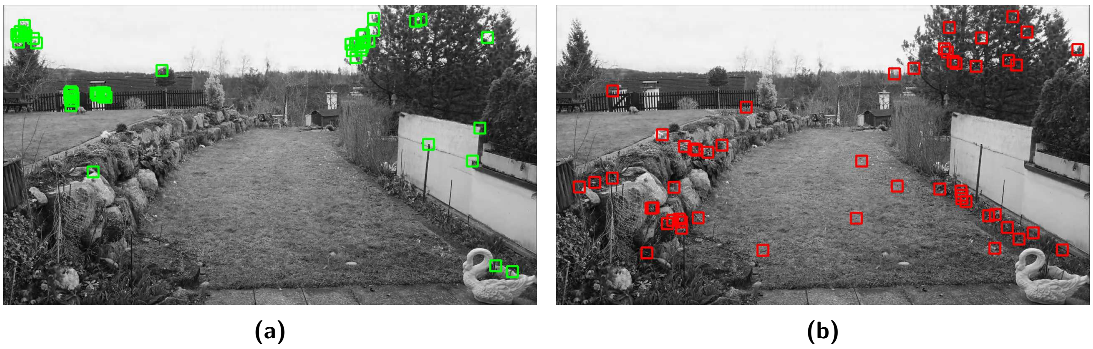


Figure 6.3: Most stable and unstable features. (a) The 50 most stable features are mainly located in image regions that correspond to man made objects. The high contrast between the top of conifers and the sky results in high responses of the feature detectors. Since their foliage does not change with the seasons some stable features can be found at their crowns. (b) The 50 least stable features correspond to image region that are heavily affected by seasonal changes like lawn and other kinds of vegetation.

6.1.2 Long-term Robustness Prediction

The long-term robustness evaluation of visual features in a natural outdoor scenario from the previous section has shown that many of the initially extracted features can only be detected for short time frames. Incorporating these unstable features into the map consequently reduces long-term robustness due to an increased probability of establishing false feature correspondences during localization and loop closure detection. Thus, relying only on the response of the detector function for feature selection is not sufficient for creating long-term robust maps.

In general, a scene contains objects which are likely to be persistent over time like tree chunks, rocks, fences or buildings whereas other objects can be expected to be non-stationary or change their appearance, e.g. cars, people or vegetation. Visual features extracted from image patches correspond to some part of a physical object. Having knowledge about the kind of object would allow to make a prediction about whether a feature is robust and thus contributes useful information to the constructed map or if it should be discarded. However, robust object recognition under varying conditions is a challenging task by itself and requires a large amount of computational resources. Instead, we make the assumption that the image statistics around extracted visual features allows to train a classifier in order to discriminate stable and unstable features. The proposed generic approach for long-term robustness prediction of local visual features can be incorporated as an intermediate filtering stage in the standard feature processing pipeline to reject potentially unstable features during the mapping phase.

The robustness characteristics of visual features are learned from texture and color statistics around their corresponding interest points. The data for training and evaluation is obtained from recordings of a train journey on the same track over the seasons which allows the easy identification of stable and unstable features. A support vector machine is trained with these statistics with the aim of predicting the long-term robustness of features in unknown scenes of the same domain. Application of the proposed approach results in a smaller set of features with a higher percentage of robust ones. Hence, it reduces processing time, memory consumption and improves long-term robustness by decreasing the probability of false positive matches.

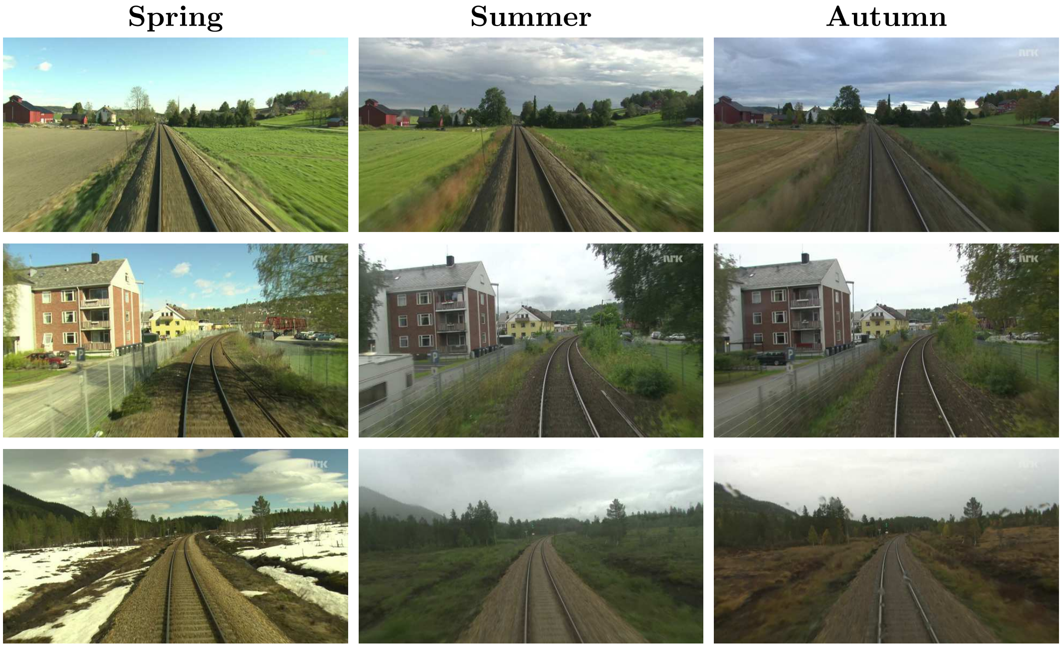


Figure 6.4: Varying appearance of the same location over time. Images are from the Nordlandsbanen Railway data set which consists of recordings of the same track in every season featuring seasonal, weather and lighting effects in man-made and natural environments. Note that season winter was not considered in this work since the drastic changes in appearance do not allow a reasonable amount of feature matches. Content licensed under Creative Commons, Source: NRKbeta.no `https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/`.

Robustness Prediction of Visual Features

In this section we describe the process of training data generation and the learning approach for robustness prediction of local visual features. The proposed method is based on the assumption that texture and color statistics around visual features can be used to train a discriminative model that enables the prediction of a features' long-term robustness. Stable and unstable features are identified in images of the same location at different points in time and under varying environmental conditions. A support vector machine is then trained with labeled feature vectors computed from the visual features' surrounding image patches.

Data Set The data set we use in this work consists of high definition video material from a TV documentary recorded on a train journey on the same 729 kilometer railway track once in every season (see Fig. 6.4 for example images). It features seasonal effects like snow covered ground and color changing foliage as well as different weather and lighting conditions in natural and man-made environments. GPS-readings were recorded in conjunction with the video and used to time-synchronize the recordings of the different seasons. After synchronization the position of the train in one frame from one video corresponds to the same frame in the other videos. Thus frame-accurate ground truth information is available within the accuracy of the GPS-localization.

We extracted frames at a fixed interval throughout the whole videos of every season to generate a manageable amount of image data. The resulting frames contain a broad variety of locations each with a season specific appearance. To further refine the GPS-based alignment the frames from one season are defined as reference and the best matching frame from the other seasons is found within a sequence around the extracted keyframes. Matching is performed by computing the normalized cross-correlation between the down-sampled and patch-normalized images from the reference frame and the ones from the sequence. Frames extracted from parts of the video where the train passed a tunnel, stopped at signal lights or had the windshield wiper turned on have been sorted out in order to not distort the results.

The final training data set contains images of 164 locations in seasons spring, summer and autumn. Recordings from the winter season were not considered for the training process nor the experiments since the extreme appearance did not allow a reasonable amount of feature matches. The TV logo in the top right corner of the extracted frames is masked out for interest point detection and feature description.

Training Data Generation To generate labeled training data we identify stable and unstable features in the data set by tracking them over the seasonal images of every location. For every of the 164 locations we detect interest points in the corresponding seasonal images and compute their descriptors. Descriptors are then matched between all image combinations by performing a k-nearest neighbor search in descriptor space with $k = 2$ and the application of the distance ratio test with a ratio of $r = 0.75$. We consider two features as equal if their descriptors match and the Euclidean distance in image space is smaller than a threshold. We set this threshold to 40 pixel since the viewpoint of the camera is not exactly the same over the different recordings. Features which were successfully matched over all seasonal images of one location are labeled as stable and added to the training set. Features which could not be matched to any of the other seasonal images are labeled as unstable. Since the amount of unstable features is usually many times larger than the stable ones only a subset is chosen randomly so that the classes are balanced. The result of the selection process is illustrated in Fig. 6.5a, 6.5b.

We use the implementations from the OpenCV-library [15] of the FAST interest point detector [130] and the binary BRIEF descriptor [17] as well as SURF [10] for interest point detection and description. The orientation of the features is not encoded since the target application is localization of vehicles moving on the ground plane. The threshold for the detector response has been set to a value so that roughly 1000 interest points were detected per image. From the total number of features found in the 164 seasonal images the selection process yields a balanced set of 10.000 and 14.000 labeled features when using FAST/BRIEF and SURF, respectively. The higher number of selected SURF features can be explained by the fact that interest point detection is performed at multiple scales while FAST interest points are extracted at a fixed scale. Furthermore parts of the image corresponding to physically nearby regions are frequently affected by motion blur which is disadvantageous for the FAST detector since it responds to corner like image structures.

Training Process Feature vectors for the support vector machine training are constructed from the pixel values around the interest points of the stable and unstable features which have been selected for the training set. Features are a combination of low level texture and color information. Texture information is obtained by computing a histogram of the uniform Local Binary Patterns (LBP) [119]. Color information is encoded in an 18 bin hue histogram. The sample points within the patch region around the interest point are weighted by a two dimensional Gaussian window for histogram computation. Concatenation of the histograms yields the 77-dimensional feature vector. The size of the patch used for the computation of the feature vector is chosen to be equal to the size of the descriptor window which is determined by the scale of the interest point. Since the FAST interest points are not localized in scale the image patch is fixed to 48×48 pixels. When using SURF the descriptor window varies with scale. Instead of scaling the LBP-operator we chose to resize the patch to 48×48 pixels with bilinear interpolation for feature computation.

A support vector machine with a Radial Basis Function kernel is trained with feature vectors from the balanced stable and unstable visual features resulting from the selection process. Parameters of the support vector machine are determined by a grid search with five-fold cross validation. The individual steps of the whole process are illustrated in Fig. 6.5.

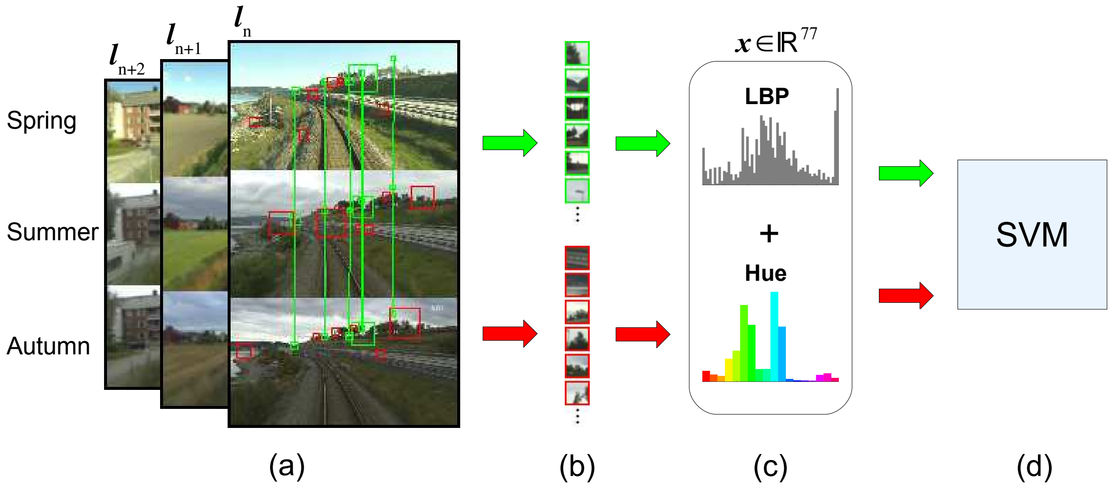


Figure 6.5: Illustration of the training process. (a) Stable and unstable features are identified in the seasonal image sequence of all locations $l_{1...164}$ and selected for training. Features which can be tracked across the seasonal images of the current location l_n, depicted with green squares, are selected as stable samples. Features which were extracted in one season but could not be matched in any of the other seasons are defined as unstable. Since the number of occurrences of unstable features is many times higher only a subset is randomly chosen to obtain balanced classes. The selected unstable samples are represented by red squares. The size of the squares is determined by the scale of the corresponding interest point. (b) Patches of the selected stable and unstable features are resized to 48 × 48 pixel for feature vector computation. (c) A histogram of uniform Local Binary Patterns and a hue histogram are computed on the image patch. Concatenation of the histograms yields the 77-dimensional feature vector. (d) Finally a support vector machine is trained with the labeled feature vectors.

Experiments

The aim of the robustness prediction is to filter out potentially unstable visual features during the mapping phase. This results in more compact maps and a reduced probability of false positive matches compared to conventional feature processing.

We evaluate our filtering approach and the conventional feature processing on a separate test data set which is created in the same way as described in section 6.1.2 but with an offset in time. This ensures that the training and test set do not contain images of the same location. We perform cross-season feature matching on sequences of 60 extracted frames and store the number of true positive and false positive matches. Features from summer and autumn are matched against features from spring which is defined as the reference season. Two features from the reference and query season are defined as equal, and counted as true positive, if their descriptors match and the Euclidean distance in image space is smaller than 40 pixels. If the descriptor of a feature was matched to the descriptor of a feature in the other season but the Euclidean distance is greater than 40 pixels it is considered a false positive match.

In general, a certain number of features is required to obtain distinctive descriptions for individual places as well as accurate pose estimations. However, an increasing amount of features also increases the probability of false positive matches and the demand for computational and storage resources. Therefore, the cross-season matching over the sequence is performed several times using different memory sizes within a reasonable range of $5 - 2000$ features from the reference and query frames, respectively. With the conventional method the appropriate number of features according to the memory limit is obtained by adjusting the threshold of the interest point detector until the limit is reached. In case there were more features extracted, the ones with lower response are sorted out. The robustness filter is only applied for feature extraction from the reference frames. In this case the selection process is different. In the first step a number of interest points is detected that is equal to 1.5 times of the final memory sizes. For every feature we compute the confidence with the trained support vector machine and sort them by their confidence value. Then the top features are selected according to the memory size.

True positive and false positive rates are averaged over one run through the image sequence and memory step. Performance of the conventional feature matching and our robustness filtering approach is compared in terms of the F1-score as the harmonic mean of precision and recall.

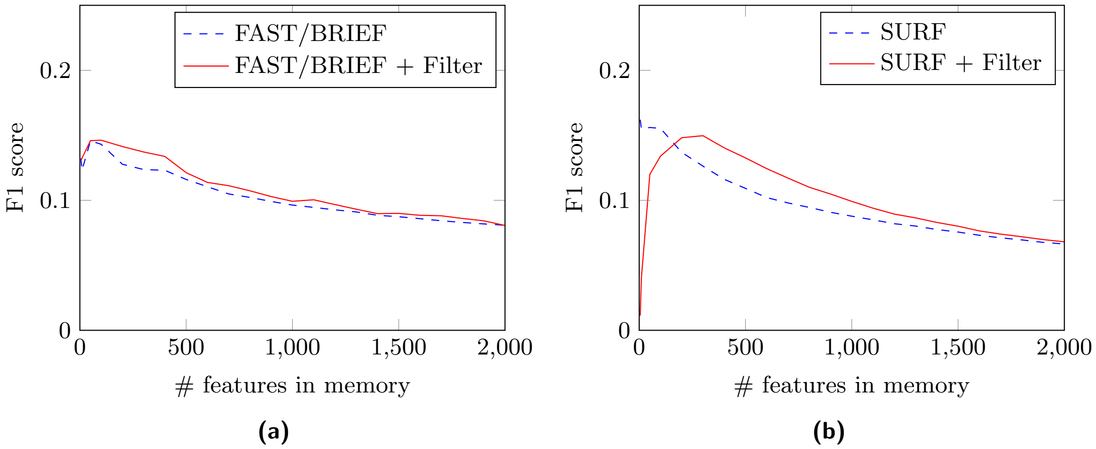


Figure 6.6: Matching from summer to spring. (a) Results from the experiment with FAST/BRIEF. Application of the filter based on robustness prediction yields an average improvement of 4.74%. (b) Results from the experiment with SURF. In the low memory region where false classifications have a stronger impact the F1-score is worse when the robustness filter is applied. In the subsequent memory regions performance with the robust filter leads to a substantial performance gain so that the overall improvement results to 7.72%.

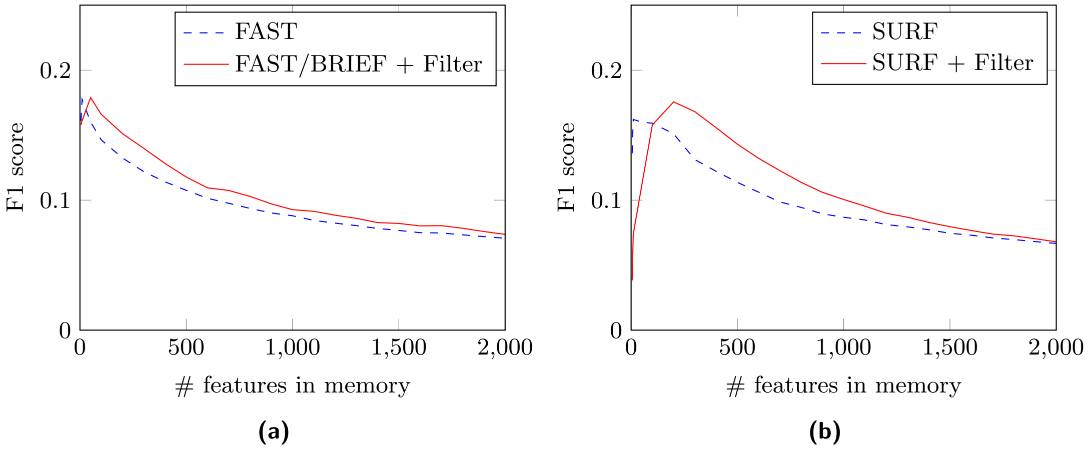


Figure 6.7: Matching from autumn to spring. (a) Results from the experiment with FAST/BRIEF. Application of the robustness filter results in an average performance gain of 8.88%. (b) Results from the experiment with SURF. As in the previous experiment the F1-score obtained with the robustness filter is worse when using up to 150 features for memory. For larger memory sizes feature matching benefits from the application of the filter so that the average performance is increased by 11.79%.

Results Results of the comparison between conventional feature matching and our robustness filtering approach are illustrated in Fig. 6.6 and Fig. 6.7 showing the F1-scores from experiments matching from summer to spring and from autumn to spring, respectively. As expected curves from all experiments decrease with growing memory size since the probability of false positive matches is getting larger. It can be observed that the application of the robustness filter during feature extraction from images of the reference season generally increases feature matching performance compared to conventional feature processing. Applying the proposed additional robustness filter results in an average performance gain between 4.74%-11.79% in terms of the F1-score.

In the experiments with SURF the performance of our robustness filtering approach is worse than the conventional method in low memory regions using up to about 150 features. The model apparently rejected robust features as unstable which has a negative impact on performance especially when only few features are memorized. The same effect can be observed in both experiments since the robustness filter is applied for the reference season spring. In the subsequent memory regions performance with the robust filter leads to a substantial performance gain so that the overall average improvement results to 7.72% and 11.79%.

In the experiments with FAST/BRIEF the performance gain is smaller but more steady regarding the low memory regions. The average performance improvement is equal to 4.74% and 8.88% when matching from summer to spring and matching from autumn to

spring respectively.
Surprisingly the performance of both variants is better when matching features from autumn to spring than matching features from summer to spring even if the temporal distance is shorter. We assume that this effect results from the fact that the vegetation might be more similar.

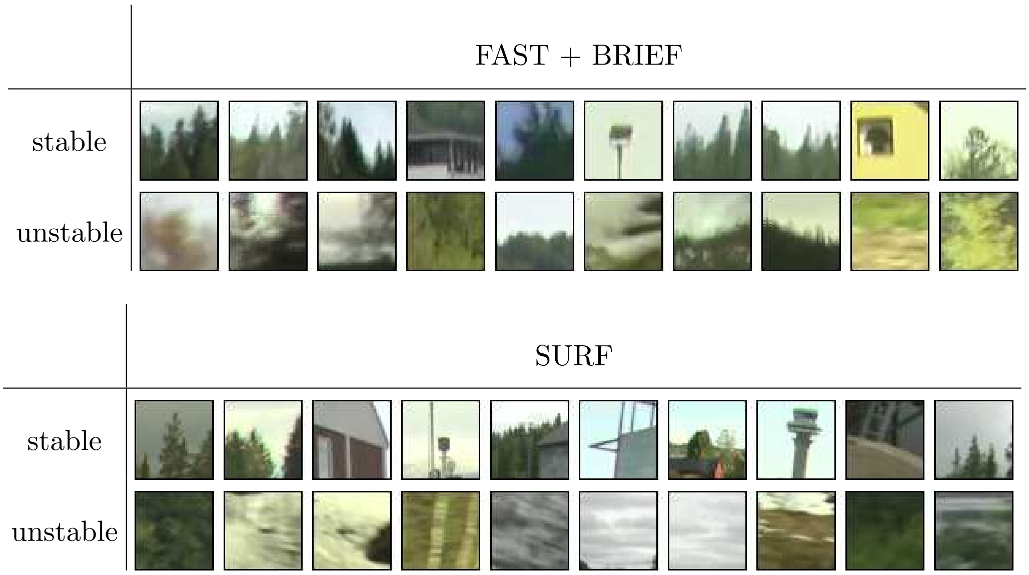


Figure 6.8: Most stable and unstable features. Features from all seasonal images of the test data set were ranked by the confidence value of the robustness prediction to determine the 10 most stable and unstable features. Features with high confidence are found on man-made objects like lanterns or buildings which is what we expected. Surprisingly features on top of conifers against the skyline give the highest confidence values. Low confidence features are mainly found in regions with little texture and contrast.

Discussion

We have presented a method to train a model for long-term robustness prediction of visual features using images of the same location across seasons to obtain representative training statistics of stable and unstable visual features. The model can be incorporated into the standard feature processing pipeline as an intermediate filtering stage to predict the long-term robustness of the extracted features and reject the potentially unstable ones.
Experiments on a separate test set have shown the capability of the model to generalize to unseen features of the same domain. Integration of the model for robustness prediction during feature extraction from the reference images resulted in an increased F1-score between 4.74%-11.79% compared to conventional feature processing. While we expected

a higher gain in performance it has to be noted that learning a model to distinguish between stable and unstable features is a hard problem because of the high diversity of visual features and an overlap between the classes. In real world scenarios overlapping classes can not be avoided since even the most stable features might be occluded, overgrown by vegetation or disappear due to cast shadows. Therefore a misclassification can always occur although a features' characteristics conform to the learned model.

In general the results were quite surprising since we expected to find robust features on man-made objects. Instead, the most stable features are found at tree tops of conifers with high contrast against the skyline (see Fig. 6.8). This can be explained by the nature of the data set which contains a lot more natural than man-made urban scenes and the selection bias of the interest point detectors.

Since the approach is simple and generally applicable it would be interesting to evaluate it in different kinds of environments. An interesting data source would be time-lapse recordings which capture the appearance changes of a location over time with a fixed camera because they allow the easy identification of stable and unstable features. However, recordings from several different places would be required to capture general feature characteristics and to learn a useful model.

An extension to further improve long-term robustness would be the integration of lighting invariant descriptors e.g. [18, 79] which could be easily incorporated into the proposed approach.

6.2 Learning Robust Representations with SFA

The model for SFA-localization introduced in this work learns a spatial representation of the environment by the extraction of slowly varying features from the high dimensional visual input during an initial exploration phase. In a static environment the slowest resulting SFA-outputs code for the position of the robot and enable a precise localization. However, if there are environmental effects during the learning phase, like illumination changes, varying on an equal or slower timescale than the position of the robot they will be encoded in the resulting SFA representation and interfere with the spatial coding. Furthermore, long-term appearance changes of the environment occurring between the learning and localization phase, like seasonal effects on vegetation, drastically affect overall image statistics and thus will prevent successful localization.

In the following section we approach the problem of dealing with short-term appearance changes, affecting the quality of the spatial representation, and long-term appearance changes, preventing successful localization. Trough the use of image preprocessing techniques and alternative image representations the effect of appearance changes on the image statistics could be reduced. However, the chosen preprocessing and representations would need to provide perfect invariance w.r.t. slowly changing environmental variables. Otherwise these variables would still be encoded in the slowest SFA-outputs affecting the quality of the learned representation. Therefore, we propose to use the invariance learning capabilities of the SFA method to tackle the problems of short- and long-term robustness. We extend the model using loop closures in the trajectory to restructure the training data for improved robustness. Images from loop closures, representing the same place under different environmental conditions, are re-inserted in the temporally ordered image sequence. This increases temporal variation of environmental effects and is a feedback signal for the SFA-model that has to find functions producing a similar output due to its slowness objective.

6.2.1 Learning Short-term Invariant Representations

If one assumes a static environment only the spatial configuration of the robot, given by (x, y, φ), changes over time. In this ideal scenario the slowest resulting SFA-functions will be representations of the robot's position or orientation depending on the movement statistics during training. In a real world world scenario, however, the environment can not be assumed to be static and other slowly changing environmental variables, e.g. global illumination, will be embedded in the image data. Since the SFA-model directly processes the raw pixel values the learned representations are susceptible to such appearance changes of the environment varying on an equal or slower timescale than the position of the robot. To deal with this problem we propose to use invariance learning, which is the basis of SFA, in order to learn representations that are not affected by environmental changes during the training phase. We use a method to recognize a previously visited place, i.e. loop closure detection, which allows us to re-insert images of the same place, with a possibly different appearance, in the temporal sequence of training images. Thereby, the variation of environmental effects is increased. Thus, it is a feedback signal for the SFA-model, since the slowness objective enforces the learning of functions that produce similar outputs for temporally close training samples. By restructuring the temporal order of the training sequence we can provide the unsupervised SFA learning with an external supervisory signal.

Loop Closure Detection

To validate the feasibility of the approach we first used ground truth information about the robot's position to identify loop closures in the training trajectory. A positive match, i.e. the result of a nearest neighbor search, requires that the spatial distance between the match candidates is smaller than a predefined threshold and that there is a minimum temporal gap between them. In real world application scenarios, where no external ground truth information is available, loop closures can be identified using image information. A common approach for loop closure detection is the visual Bag of Words (BoW) model where each image is represented by the occurrences of visual words from a dictionary (e.g. [23, 22]). Here we created a vocabulary of 1500 visual words by the application of k-means clustering to SURF-Features [10] extracted from every training image on dense grid. Since the target scenario is localization in small to medium scale open field environments the features are extracted with a fixed scale to enhance the spatial specificity of the resulting visual word histograms. Loop closure matches can then be determined by a comparison of the distances between histograms.

Training Using Feedback

Like in the standard approach the training sequence for the SFA-model is initially created from the temporally ordered images. If a loop closure match is identified in the image sequence, the past image is aligned to the orientation of the current one by finding the lateral offset which minimizes the image distance of the two panoramic views. The aligned image is then re-inserted into the training sequence. However, simply re-inserting the aligned image would only marginally increase the perceived variation of environmental effects. Therefore, the re-insertion is incorporated into the simulated rotation, which is performed to learn orientation invariant representations, by creating an interleaved sequence of rotated views from the former and the current image. This way, we artificially create additional variation of any environmental variable with every step of the simulated rotation.

Experiments

Experimental Setup Experiments are conducted in a simulated garden like environment covering an area of 16 × 18 meters which was created with Blender according to section 4.1. Images from the simulated omnidirectional camera are captured with a resolution of 500 × 500 pixels and transformed to panoramic views with a size of 600 × 55 pixels. The training and test trajectory consist of 1773 and 1090 poses that evenly cover the area. Crossings in the training trajectory improve spatial coding of the SFA-model and enable the extended model to get feedback from loop closures. The trajectories and the 62 loop closures determined from ground truth information are illustrated in Fig. 6.9.

Localization in a Static Environment Initially, we compared the standard and the extended model in a static environment to obtain a reference for the performance under optimal conditions and to investigate the effect of using feedback from loop closures.

Results Since the feedback only slightly changes the distribution of visited places, the resulting representations of both models are nearly identical, leading to the conclusion that using feedback does not deteriorate performance. Spatial firing maps of the first two SFA-outputs, shown in Fig. 6.10a and 6.10b, show clear gradients along the coordinate

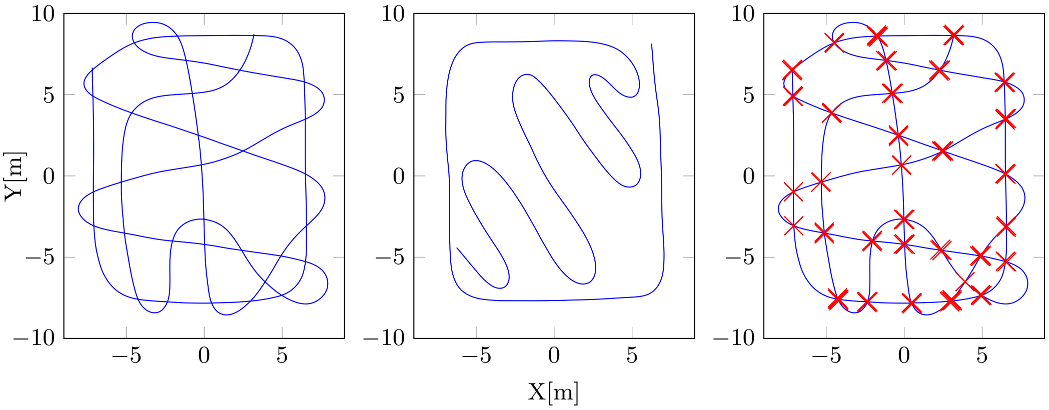


Figure 6.9: Left: Training trajectory, Middle: Test trajectory, Right: Loop closures

axes. SFA-outputs three and four are mixtures of the first two outputs. Estimated trajectories illustrated in Fig. 6.10c and 6.10d are very close to the ground truth with mean Euclidean deviations of 0.24m and 0.23m, respectively.

Localization with Changing Light In this experiment we investigated the effect of changing light intensity on the localization performance of the standard model and validated the feasibility of the feedback mechanism for improved robustness w.r.t. environmental effects. Intensity of the artificial light source was increased over the duration of the training run and thus was the slowest varying latent variable embedded in the image statistics. Training images illustrating the effect are shown in Fig. 6.11.

Results The quality of the spatial representations learned by the standard model is clearly deteriorated by the changing light intensity. Spatial coding is not observable in the spatial firing maps shown in Fig. 6.12a, while at least some position information is contained in the SFA-outputs since the estimated trajectory is not random (see Fig. 6.12c). The mean Euclidean deviation from the ground truth is 2.4m. Using the feedback from loop closures enables the SFA-model to learn representations that are more invariant against changing light intensity. Spatial firing maps illustrated in Fig. 6.12b show a clear gradient along the coordinate axis for SFA-outputs one and two, while outputs three and four are mixtures of the first two outputs. The mean Euclidean deviation from ground truth amounts to 0.46m. The estimated trajectory can be seen in Fig. 6.12d.

Localization with a Dynamic Object In this experiment we investigate the effect of a dynamic object. A textured cylinder is moved along a circle around the training area performing one circumnavigation during the training phase so that its location is the

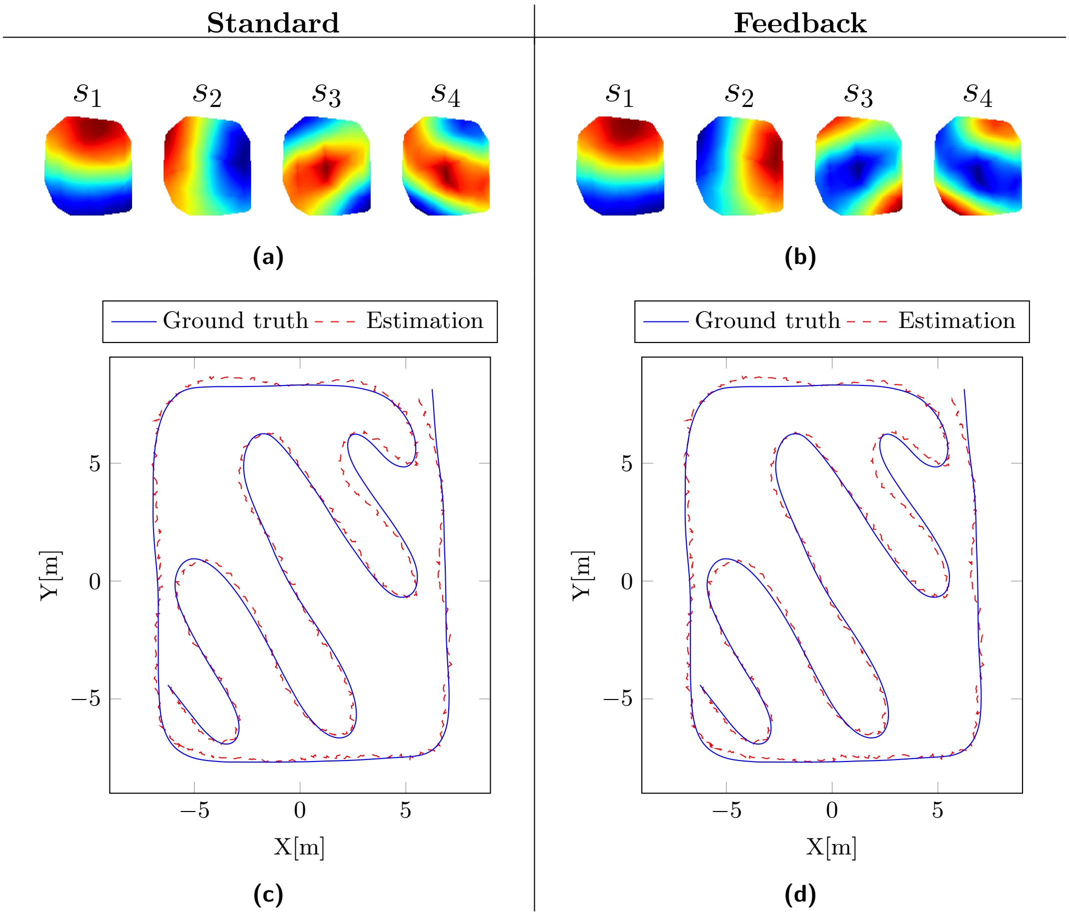


Figure 6.10: Results in the static environment. Spatial firing maps of the standard (a) and extended model (b) show strong spatial coding. Estimated trajectories of the respective models in (c) and (d) are close to the ground truth.

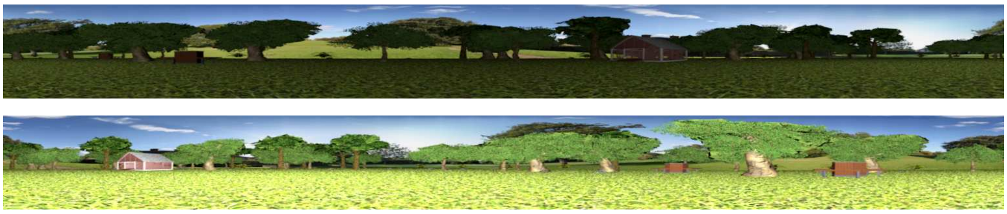
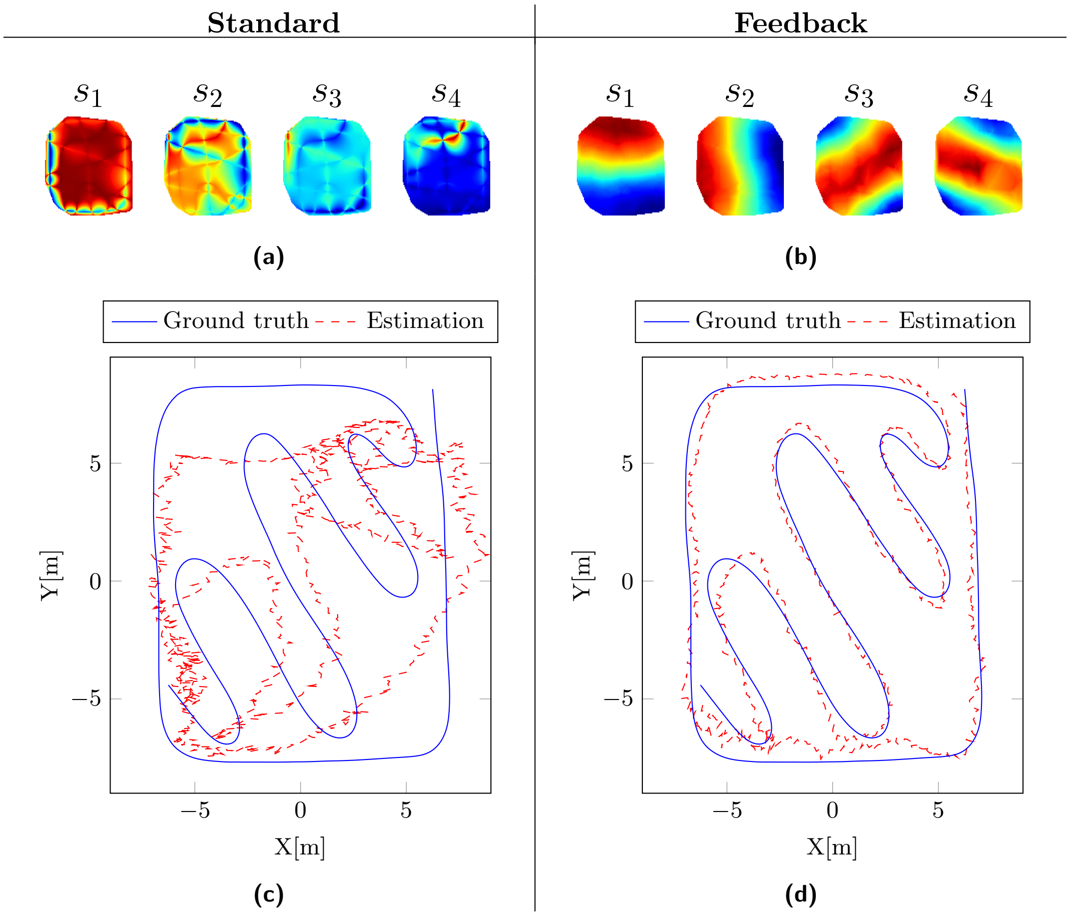


Figure 6.11: Changing light. First and last image of the training sequences. The effect on the appearance of increasing light intensity over the run.

slowest changing variable. Fig. 6.13 shows the first and the last image containing the dynamic object.



(a)

(b)

(c)

(d)

Figure 6.12: Results with changing light. (a) Spatial firing maps from the standard SFA-model clearly show that the learned representations are affected by the slowly changing environmental variable since no spatial coding is observable. (b) Localization performance is deteriorated but not random which indicates at least weak position coding. (c) Characteristic gradients along the coordinate axis in the spatial firing maps of first two SFA-outputs from the extended model suggest strong spatial coding. Restructuring the training sequence enabled the model to learn an invariance w.r.t. the slowly changing light.(d) Localization accuracy clearly improves with the extended model.

Results The effect of the dynamic object on the resulting representations is not as big as expected. Spatial firing maps of the first two SFA-outputs from both models, shown in Fig. 6.14a and 6.14b, show gradients along the coordinate axis. Accuracy of the estimated trajectories is only slightly worse than in the static environment as both models achieve a mean Euclidean deviation of 0.29m. Estimated trajectories of both models are shown in Fig. 6.14c and 6.14d. The dynamic object seems to produce local noise only but no high level information about its position is encoded in the SFA-outputs.

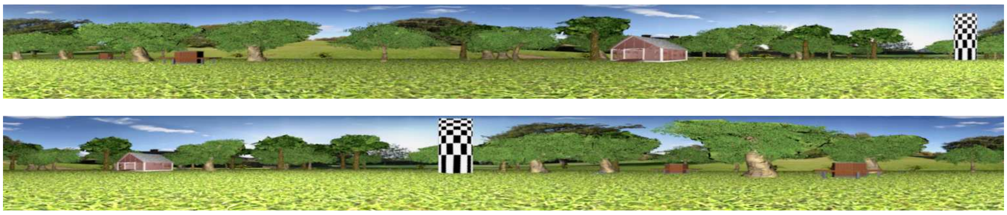


Figure 6.13: Dynamic object. First and last image of the training sequence. A textured cylinder is moved along a circle around the training area.

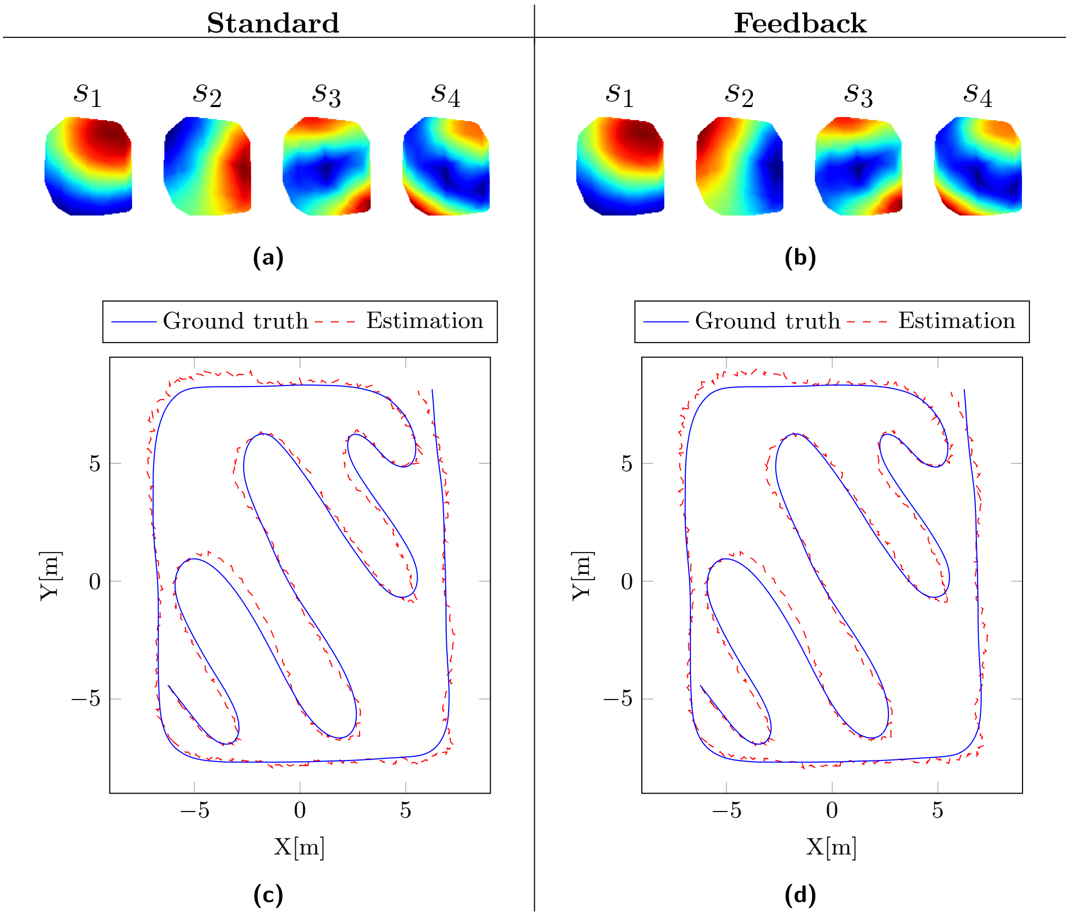


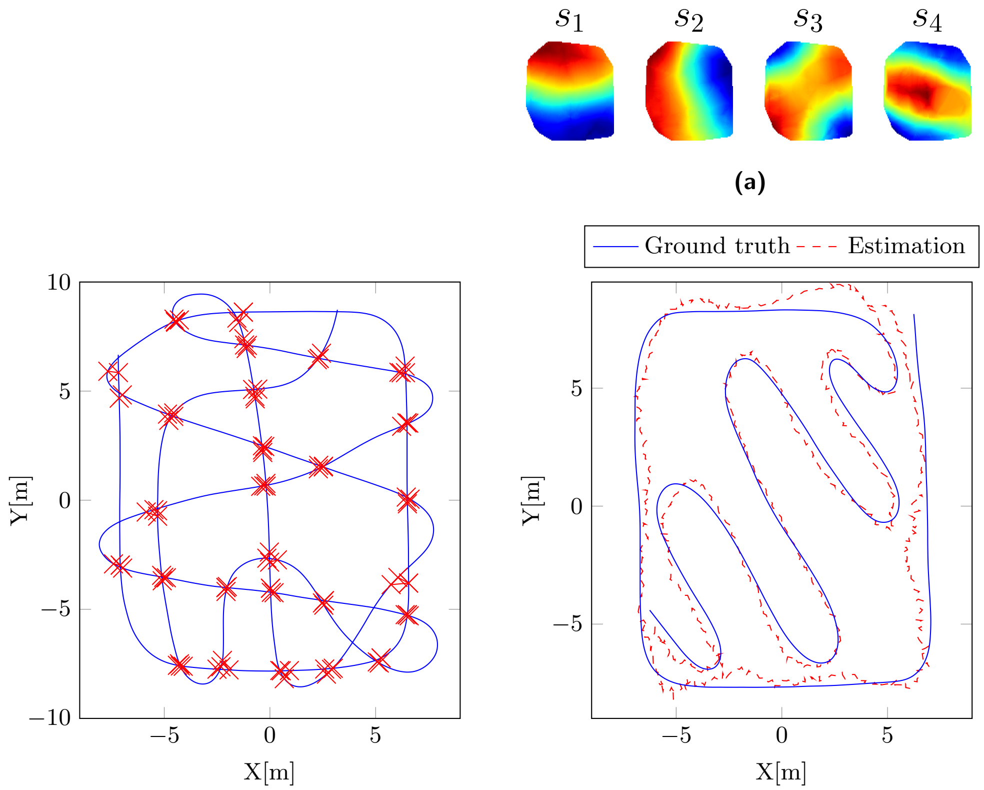
Figure 6.14: Results with a dynamic object. Spatial firing maps from the standard model (a) and maps of the model using feedback (b) are nearly identical showing clear gradients along the coordinate axis. Estimated trajectory of the standard model (c) and the extended model (d) are close to the ground truth.

Localization Using Feedback from BoW Loop Closures Ground truth loop closures used in the previous experiments had a mean Euclidean distance of 0.06m between match candidates. However, ground truth is obviously not available in realistic settings. In this experiment we used a bag of visual words model for loop closure detection. Defining 0.1m as the maximum Euclidean distance for a positive match resulted in a mean average precision of 0.52. The 54 accepted matches with a mean Euclidean distance of 0.27m are depicted in Fig. 6.15b. The experiment was performed on the data set featuring changing light intensity since the effect of using the feedback was clearly visible.

Results The resulting first two SFA-outputs show strong spatial coding indicated by the characteristic gradients observable in the spatial firing maps shown in while outputs three and four are mixtures and higher modes (see Fig. 6.15a). The resulting localization accuracy of 0.49m greatly improved over the standard model with a mean Euclidean deviation of 2.4m. As expected, in comparison to the model using feedback from ground truth loop closures with an accuracy of 0.46m, performance is slightly reduced. The estimated trajectory is shown in Fig. 6.15c.

Discussion

In this section we presented an extension to the biologically motivated model for SFA-localization using feedback from loop closures in order to improve robustness of the learned representation w.r.t. slowly varying environmental effects during training. Re-inserting images of the same place from the past in the temporally ordered image stream increases variation of environmental effects and thus is a feedback signal for the SFA-learning algorithm since it has to produce similar outputs for temporarily close inputs in order to optimize the slowness objective. We have shown that feedback from loop closures improves robustness especially for changing lighting conditions. Experiments with loop closure matches from a BoW-approach suggest the applicability of the model in real world scenarios. An elaborate solution to further improve the performance given imprecise loop closures from visual word histograms could be the use of a weighted SFA-formulation, as described in [38, 39]. Here, training samples are organized in a graph where the connecting edges represent their similarity regarding the labels.



(a)

(b)

(c)

Figure 6.15: Results with changing light using feedback from BoW loop closures. (a) Loop closures determined by matching visual word histograms. (b) Spatial firing maps suggest position coding in the first two SFA-outputs while outputs three and four show the influence of changing light intensity. (c) Localization performance clearly surpasses the standard model while deviations are larger compared to the model using ground truth loop closures.

6.2.2 Learning Long-term Invariant Representations

In real world outdoor scenarios varying environmental conditions like lighting, weather or seasonal effects have a strong impact on the appearance of a scene. If the image statistics at execution time are very different from those during mapping localization w.r.t. a previously learned representation will fail since the complex functions learned by the SFA-model will not generalize well to the input data.

In the previous section, loop closures in the trajectory have been used to re-insert images in the training sequence in order to increase the temporal variation of environmental effects. Changing the training statistics in such a way enables the SFA-model to learn an invariance w.r.t. slowly varying environmental effects during the training phase. Here, we extend this approach to long-term recordings along the same closed loop trajectory

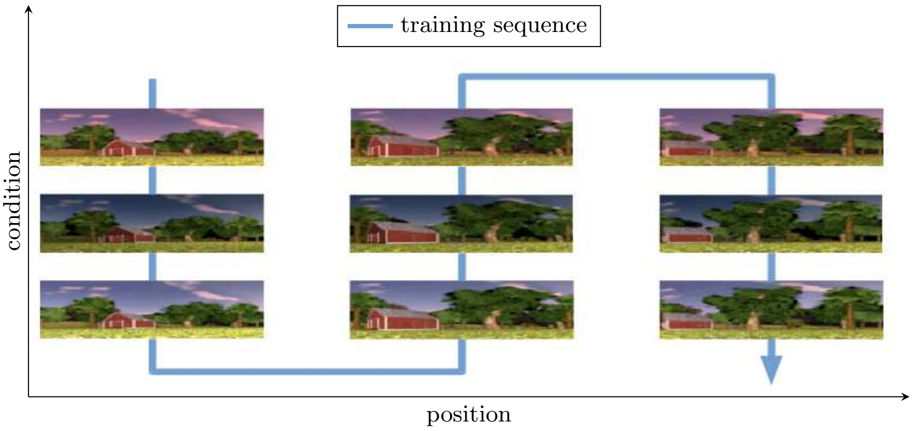


Figure 6.16: Illustration of the training sequence generation. The training data consists of images along the same trajectory in different environmental conditions. Establishing position correspondences between the recordings allows us to create a training sequence where environmental conditions change faster than the position of the robot. Images from the same place in different conditions are successively added before proceeding to the next position.

which allows the easy identification of dense position correspondences between recordings in different environmental conditions. Using the position correspondences enables the creation of a training sequence where images of the same place in all conditions are successively added before proceeding to the next place on the trajectory. The organization of the training data is illustrated in Fig. 6.16. In order to extract slowly varying features the SFA-model has to learn functions that are invariant w.r.t. environmental changes and only code for the position.

The proposed approach is first validated in a simulator where the position correspondences are known and varying environmental conditions can be easily generated. In a further experiment the approach is then validated in real world outdoor recordings from a period of three month.

Simulator Experiment

The proposed approach was first validated in a simulated environment created using Blender described in section 4.1. The purpose of conducting the experiments in a simulated environment was to prove the concepts described in the previous chapter. A virtual robot traversed a trajectory covering an area of 15×15 meters. We captured 10 image sets along the same trajectory, each consisting of 279 panoramic images with a resolution of 600×60 pixels. For every set, a change of the environmental condition is simulated by a random variation of the lighting parameters (see Fig. 6.17) resulting in non trivial illumination changes. The parameters include energy $\in [3, 8]$, the y-coordinate $\in [-10, 10]$ and the intensity of the red channel $\in [0.5, 1]$. Based on position correspondences, we reorder the training sequence in such a way that the environmental condition varies faster than the position of the robot. The model is trained with an increasing number of data sets $[1, 9]$ and the performance is tested on the successive set by computing a regression function from the SFA-outputs to ground truth positions (x, y). We repeated the same procedure with 10 random permutations of the image sets.

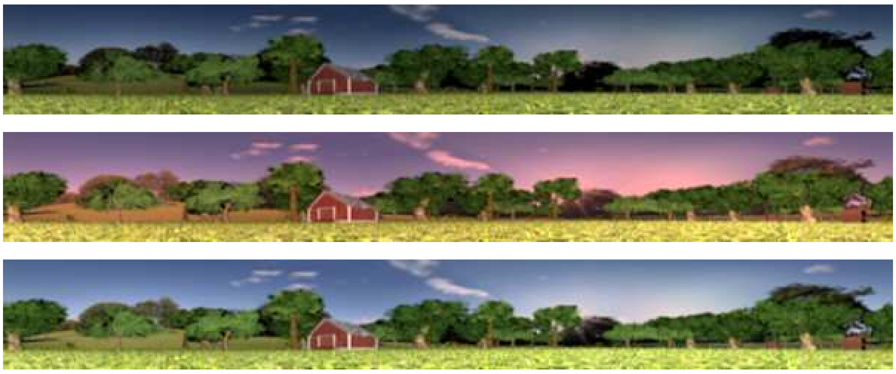


Figure 6.17: Simulated change in lighting condition. Lighting changes are simulated by randomly varying the parameters of an artificial light source for every data set.

Results Using only one data set to learn an environmental representation does not even enable a coarse localization since the localization error is too high. Adding additional data sets from different conditions increasingly improves the localization performance on unseen test data. For nine training sets the test accuracy amounts to an average of 0.35m over 10 random permutations. The localization performance for an increasing number of training sets is shown in Fig. 6.18a. One of the estimated trajectories obtained with an SFA-model trained on nine data sets and the corresponding ground truth are illustrated in Fig. 6.18b.

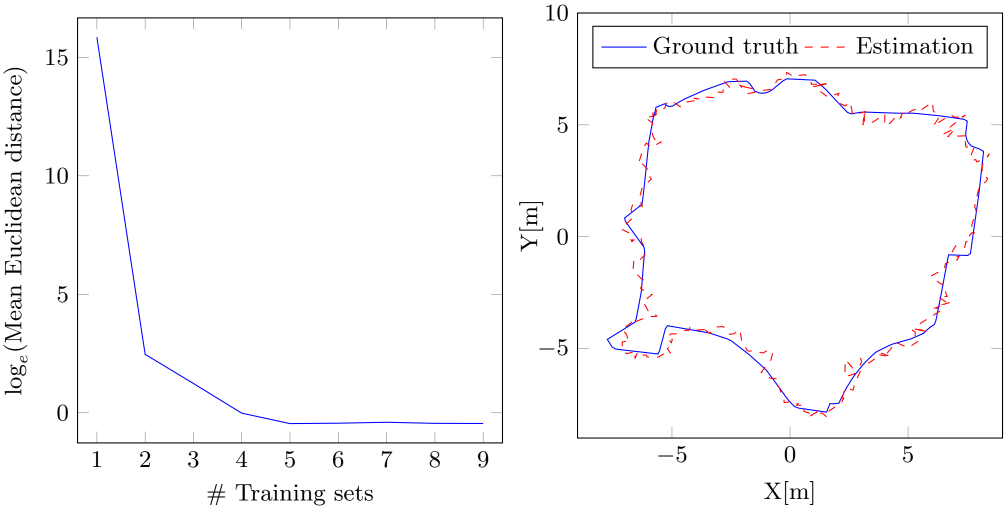


Figure 6.18: Localization performance for an increasing number of training sets (a) The plot shows the localization error as the mean Euclidean deviation from ground truth coordinates over 10 random permutations of the image sets. With only one training set accurate localization is not possible since localization errors are tremendously high. Using further data sets in different environmental conditions for training an increasingly invariant representation of the environment can be learned. (b) The illustrated trajectory is estimated for an unseen test set using nine training sets. The accuracy amounts to 0.35 meters.

Real World Experiment

In order to validate the approach in a real world experiment data sets with images from the same trajectory in different environmental conditions have been recorded over the period of three month from May to July in 2017 in Offenbach. It features different daytimes, lighting conditions and structural changes of the scene. Three example images from the same place in different conditions are shown in Fig. 6.19. As a feasible solution to acquire ground truth annotated recordings in an outdoor scenario we used a mobile robot platform which can precisely follow a given closed loop trajectory that is determined by a border wire. Since the start and end position, as well as the orientation within the base station, are known it is possible to use the odometry estimation from the robot to obtain precise position estimates. Accumulated errors in the position and orientation estimation can then be used to distribute the weighted errors backwards in the trajectory [33]. The area enclosed by the border wire amounts to 15 × 9m.

Results Using only one training set to learn an environmental representation is not sufficient to perform reliable localization in a different condition since the mean error amounts to 6.13m. With an increasing number of training sets the localization error

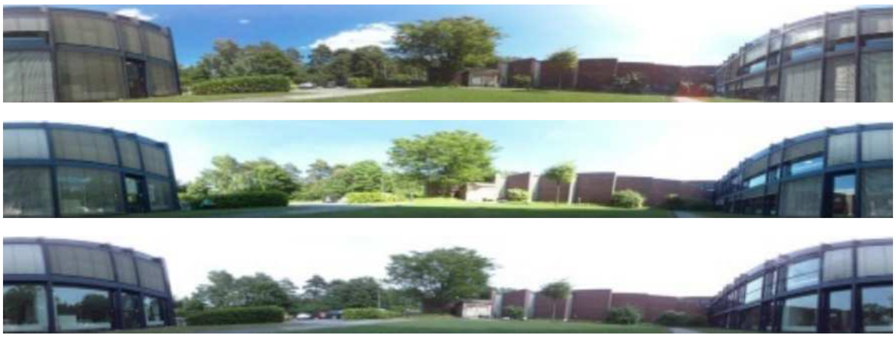


Figure 6.19: Example images for different environmental conditions. The training data was recorded over a period of three month featuring different daytimes, lighting conditions and structural scene changes. Although the recording period is rather short for long-term experiments the images exhibit significant changes in appearance. Different weather and lighting conditions drastically change the appearance of the sky and also the regions covered by shadows and their intensity. Furthermore, there were walking people and driving cars and some structural changes between the recordings with the most obvious variation being the opened/closed blinds of the windows.

quickly decreases for predictions on the next set which has not yet been used for training. The mean Euclidean distance between the predicted coordinates and ground truth for a different number of training sets is shown in Fig. 6.20a. The initial error of 6.13m quickly decreases to an error of 0.66m when using nine data sets for training. The resulting estimated trajectory from the SFA-model trained with nine data sets is illustrated in Fig. 6.20b.

Discussion

The results from the experiments have demonstrated the capability of the SFA-model to learn an increasingly invariant representation of the environment for robust long-term localization. To achieve condition invariance we created a training sequence where environmental effects change on a faster timescale than the location using position correspondences between recordings in different conditions. In both experiments localization is not feasible when using only data from one condition in order to localize in a different condition. The significantly larger error in the simulator experiments is due to the fact that the variations in image appearance are more drastically than in the real world experiments. The performance quickly increased for additional training sets in both experiments. We conclude that the SFA-model is able to generalize to different levels of environmental effects present in the training data.

The robust long-term localization on the boundary of the working area alone is not sufficient to implement complex navigation behavior. However, the absolute position estimates in the vicinity of the border wire could be used in combination with wheel- or

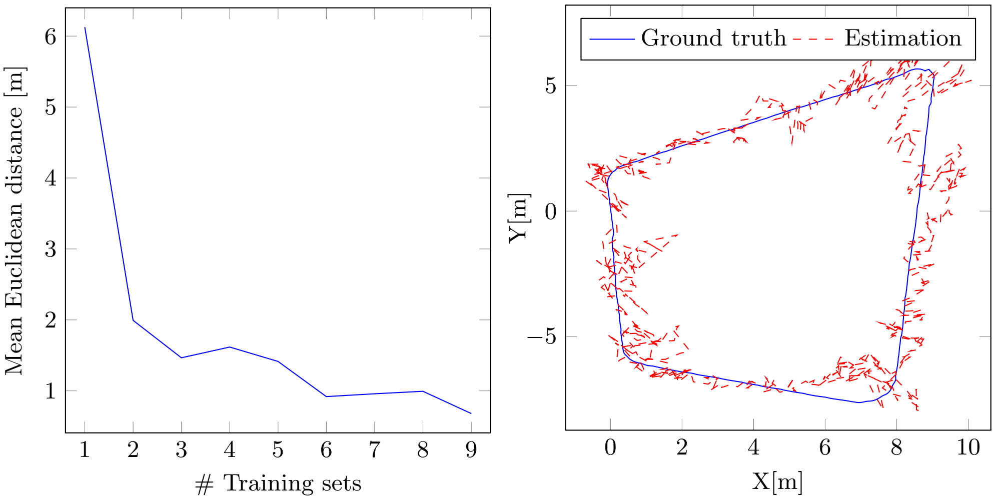


Figure 6.20: Localization performance for an increasing number of training sets (a) The plot shows the development of the localization performance dependent on the number of training sets in different conditions. The quickly decreasing error demonstrates that the model is able to learn an increasingly invariant representation of the environment. Localization accuracy is evaluated for the next set which so far was not included in the training data. (b) The estimated trajectory and corresponding ground truth for the an unseen test set using nine training sets. The mean Euclidean deviation from ground truth is 0.66 meters.

visual-odometry fused in a probabilistic framework. To enable this application the SFA-model would need to learn condition and orientation invariant representations. However, this massively increases the difficulty of the learning problem and there are many possible ways to structure the training sequence that have to be evaluated. Another open question is if both invariances should be learned jointly or in different layers. Investigating the generalization capabilities of the lower SFA-layers might also be of interest in future work.

6.3 Conclusion

In this chapter we approached the problem of robust long-term localization in open field outdoor environments using the visual input from a single camera only. Short- and long-term environmental effects like dynamic objects, different daytimes, weather conditions and seasonal changes drastically impact the appearance of a place and thus pose a major problem for vision based mapping and localization methods. Here, we focused on increasing the robustness of the map by the selection of long-term stable elements of the scene for map representation and furthermore proposed a unified approach solely based on the invariance learning capabilities of SFA.

We proposed a method for robustness prediction of visual features which are commonly used in SLAM methods to create a sparse map of the environment but might also be used to create alternative image representation for SFA-learning. A classification model was trained with cross seasonal images from corresponding places in order to discriminate between stable and unstable features. Since the model can easily be incorporated into the standard feature processing pipeline for stable feature selection it is applicable in any feature based approach. The performance for cross season feature matching increased between 4.74%-11.79% compared to conventional feature processing and could be further improved using lighting invariant descriptors e.g. [18, 79]. However, the performance increase might not be sufficient to achieve long-term robustness unstructured outdoor environments. One problem are the overlapping class distributions which can not be avoided since even the most stable features might be occluded, overgrown by vegetation or disappear due to cast shadows. Furthermore, the approach presumes that the employed interest point detector produces repeatable results under challenging conditions.

As an alternative approach for obtaining robust environmental we presented a unified approach which is solely based on the invariance learning capabilities of the SFA-algorithm. First, we approached the problem of slowly changing environmental variables during training which might interfere with the spatial coding. The identification of loop-closures in the training trajectory allows to change the perceived image statistics by re- inserting images of the same place from the past in the temporally ordered image sequence. Thereby, the variation of environmental effects is increased and we can provide the unsupervised SFA-learning algorithm with a supervisory signal regarding its slowness objective. Results from the experiments have demonstrated that feedback from loop-closures improves robustness especially for changing lighting conditions.

In order to learn invariant representations for long-term robust outdoor localization we extended the approach to recordings along the same trajectory in different conditions. Due to the closed loop trajectory and the exact knowledge of the start and end pose of the robot we could establish dense position correspondences between recordings in different conditions. Based on the position correspondences we created a training sequence where images from the same place are successively added before proceeding with the next place on the trajectory. In this way, the perceived environmental condition changes faster than the position. Therefore, the SFA-model needs to learn functions that are invariant w.r.t. environmental changes in order to encode the slowly varying position. Results from the experiments in the simulator and the real world have shown that the model is able to learn an increasingly invariant representation of the environment using data sets from different conditions. For the practical applications it has to be investigated in future work how the condition invariance learning can be best combined with the orientation invariance learning. It would also be interesting to explore the generalization capabilities of the slow features learned in lower layers to new outdoor environments.

7 Navigation Using Slow Feature Gradients

Navigation is a crucial ability for autonomous mobile robots operating in a spatial environment. To perform complex navigation tasks a robot needs an internal representation of the environment to estimate its own location and to plan a viable and safe path to a target. There exists a variety of methods enabling a mobile robot to create such a representation using vision as the only sensory input. The resulting internal maps represent the environment in different ways, e.g. as a graph structure reflecting the topology, a discretized occupancy grid or a continuous space representation leading to different navigation strategies with varying levels of complexity [96, 13, 46].

Navigation is one of the most challenging tasks for mobile robots. Many animals, on the other hand, have excellent navigation capabilities. The paths they take may be suboptimal, but they are rapidly selected, flexible and result in an adaptive and robust navigation behavior. In this chapter we present a new method for navigation in slow feature space using gradient descent which builds upon the orientation invariant representations of the location which are learned in advance with the biologically motivated SFA-model. After the unsupervised learning of the environmental representation, navigation can be performed efficiently by following the SFA-gradient, approximated from distance measurements between the target and the current value. Since the slowest two SFA-outputs ideally encode the x- and y-coordinate of the robot as half cosine-/sine functions they change monotonically over space and are de-correlated fostering a global minimum at the target location.

A common approach to realize navigation in topological or occupancy grid maps is to use a graph search algorithm like A* [54]. Given an admissible distance heuristic it is guaranteed to find the optimal path but it is memory and computationally intensive for large environments with many obstacles. Moreover, during the execution of a planned trajectory deviations from the path have to be detected and corrected. If the deviation becomes too large a new planning step has to be initiated. The potential field method is a an approach for navigation in continuous metric spaces that is based on gradient ascent in a vector force field defined by an attractor at the target position and repulsive forces from obstacles [69, 8]. Although it is an elegant solution, a known limitation of the approach are local minima caused by certain types of obstacles or their spatial configuration [156]. These local minima can be avoided by designing an optimal navigation function that has a global minimum [30]. However, determining such a function is only feasible for small environments with low complexity [13]. Minimizing an image distance function of panoramic images from the current and a target location is used in [103, 104] to obtain a homing vector. A prerequisite for obtaining a navigation direction using this method is the visibility of the target location from the current position. Path planning in environments with restricted visibility thus requires a representation containing several snapshots organized in a topological graph (e.g. [44]). Navigation in the low dimensional SFA-representations of an environment has been approached using reinforcement learning in order to obtain policies that guide an agent to a goal location in a simplified version of the Morris water maze task [81] and with views from a mobile robot [12]. Although the presented results demonstrate the feasibility of the method an additional massive learning phase is necessary to obtain the policies that determine the executed motion commands in response to a measurement.

In the next section we introduce a straightforward and efficient approach for navigating directly in slow feature space using gradient descent. A navigation direction can be inferred by distance measurements between the value at the current and the target location. We experimentally show that the method enables a reliable navigation and that the learned slow feature representations implicitly encode information about obstacles which are reflected in the gradients. Thus, complex navigation tasks can be solved without requiring explicit trajectory or obstacle avoidance planning. In section 7.2 we present preliminary results on further extensions to the proposed navigation method and empirically investigate further potentials of the slow feature representations for efficient navigation.

7.1 Navigation with Slow Feature Gradients

Due to the simulated rotation during the training phase the learned slow feature representations are invariant with respect to the orientation of the robot and only code for its position. Hence, the slowest position encoding SFA-outputs change monotonically over space. Given two points in 2D space we can take the difference between their slow feature representations to define a cost function and estimate a navigation direction by approximating the gradient of the cost surface. Navigation between a start and a target location can be achieved by performing gradient descent on the cost surface. The slow feature representations of the visual inputs at the target locations can be acquired during the training phase at points of interest e.g. the charging station of the robot. Furthermore, due to the slowness objective of the SFA learning algorithm, obstacles should be implicitly encoded in the resulting representations. Since a mobile wheeled robot can not directly get over obstacles, the mean temporal distance between sensor readings from opposite sides will be large compared to nearby measurements on the same side. In order to generate slowly varying output signals the slow feature representations should "flow" around the obstacles.

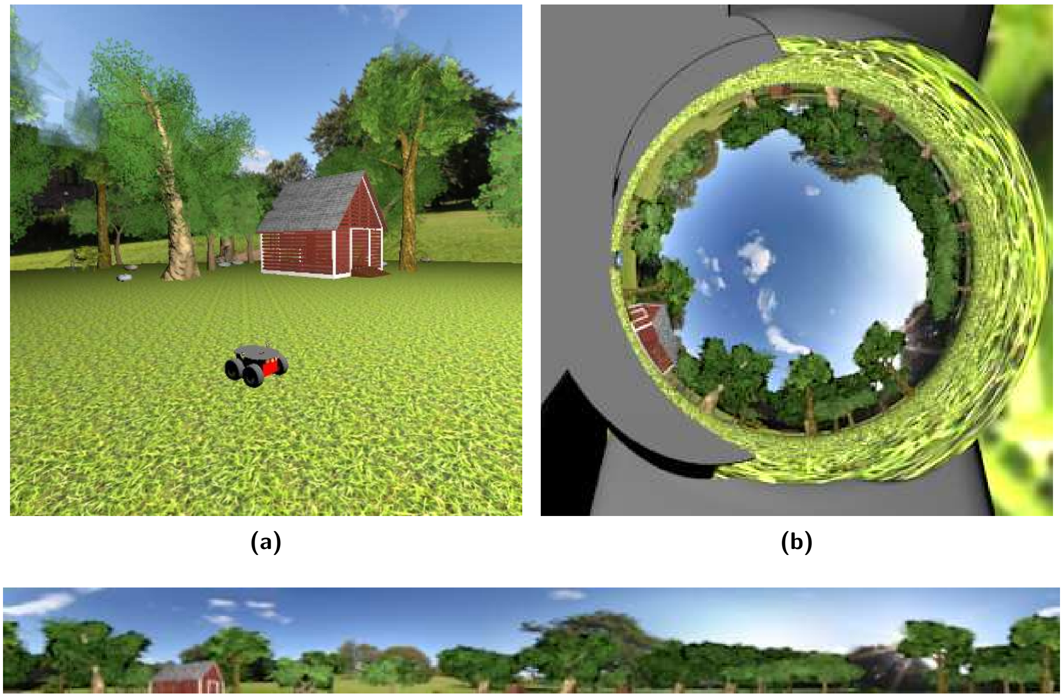
Assume that n slow feature components have been chosen to be used for the representation of the environment. The current position in 2D space is given by $p := (x_p, y_p)$. The mapping function $f : \mathbb{R}^2 \mapsto \mathbb{R}^n$ transforms a position to the slow feature space by processing the corresponding image. Given a target position $t := (x_t, y_t)$, we define the function $C : \mathbb{R}^n \mapsto \mathbb{R}$ which computes the cost from p to t, taking only $f(p)$ as input. C can be any distance function such as the Euclidean distance: $C(f(p)) = \sqrt{\sum_{i=1}^{n}(f(p)_i - f(t)_i)^2}$ Ideally, we require the analytic gradient $\frac{\partial C(f(p))}{\partial p}$ as navigation direction which is, however, infeasible to obtain. Therefore, we compute its local linear approximation. In addition to the cost of the current position $C(f(p))$ we acquire at least two additional cost values $C(f(p_1))$, $C(f(p_2))$ for points "close-by", which have to be non-collinear, and which are used to fit a plane to the surface of C.

7.1.1 Implementation

In order to acquire the cost function measurements from at least three nearby positions in an efficient way, we place one omnidirectional camera on each side of the robot with a fixed offset to the robots coordinate system. Thus, we obtain two measurements for each time step. To estimate the first gradient at time t_0 we take the difference quotient from the cost values computed for the image from the left and the right camera and the corresponding translation vector. After the first step, along the estimated gradient we can measure two additional cost values at time t_1. Next we estimate the plane defined by the four points obtained at t_0 and t_1. Therefore, we take three points to define two vectors and compute the plane normal from their cross product. We repeat this step for all four possible combinations and take the mean of the normal vectors to approximate the gradient by the slope of the corresponding plane. Then we make a step along the gradient direction and replace the points from t_0 with those from t_1 and those from t_1 with the cost values measured at the new location. This process is repeated until a predefined precision is achieved or the maximum number of iterations is reached. The estimated gradient is multiplied with a scaling factor η. A momentum term γ is used to incorporate information from past gradient information to improve convergence and overcome local minima.

7.1.2 Experiments

The proposed navigation method was evaluated in experiments in a simulated garden like environment with a size of 18×22 meters (for reference: the robot has a side length of ≈ 0.7 meters). It has been created using the 3D-software Blender and its python API as described in section 4.1. The virtual robot is equipped with two omnidirectional cameras attached to its left and right side. Images from the omnidirectional cameras are rendered with a resolution of 300×300 pixel and unwarped to panoramic views



(a)

(b)

(c)

Figure 7.1: Simulated environment. (a) Experiments were performed in simulated garden like environment. (b) Example image from the omnidirectional camera. One camera is attached to each side of the robot. (c) The field of view of the unwarped panoramic images is cropped to discard static image regions.

with a resolution of 600 × 60 pixel. The vertical field of view of the panoramic images is reduced to discard the static parts of the image i.e. the other camera and the robot. The simulator environment as well as a rendered image from the omnidirectional camera and the unwarped panoramic view are illustrated in Fig. 7.1. During the training phase the robot starts to move along a line with a random orientation. In case it reaches the border of the training area or the border of an obstacle a new orientation is chosen randomly and the robot follows the new direction. The velocity is kept constant with 0.2 units per time step. We captured 5000 images for the experiments to ensure an even sampling of the environment. Using more images increases the quality of the learned representations while a reasonable representation can be obtained with less images on a directed path with few crossing along each coordinate axis [42]. Ideally, the first two slow feature functions are representations of the robot's x- and y-coordinate. Hence, they are orthogonal and change monotonically over space which guarantees a global minimum of cost function C. Therefore we set $n = 2$ and do not consider higher functions. The gradient scaling factor is set to $\eta = 1.0$ and the momentum to $\gamma = 0.5$

for all experiments shown in the following sections. The navigation task is considered as successfully completed if the robot ends up within a radius of 0.5 units with respect to the specified target location. The maximum number of iterations is set to 400. After the unsupervised learning phase we create the spatial firing maps 1...n by plotting the color-coded SFA-outputs $s_{1...n}$ for the images captured on a fixed grid. We create the plots of the cost-surface accordingly by plotting the color-coded cost from all grid positions to the target position in slow feature space. Please note that metric information is only used for illustration purposes and that the spatial representations are solely learned from the images.

Navigation in an Open Field Scenario

To validate our approach we first tested the navigation method in an open field scenario. The random training trajectory can be seen in Fig. 7.2a. The spatial firing maps, shown in Fig. 7.2b, of the four slowest functions show strong spatial coding illustrated by the characteristic gradients along the coordinate axes of the first two functions. Function three is a mixture of the first two functions and function four is a higher mode of the second one. We performed 50 trials with randomly chosen start and target points from within the training area so that the minimum distance between them amounts at least to 15 meters. We evaluated the success rate and the efficiency of the trajectories. The efficiency was calculated as the ratio of the direct distance and the traveled distance so that the highest efficiency value is one. The efficiency was only considered for successful trials and is then given as the average.

Results The robot successfully navigated to the target location in 49 out of the total 50 trials resulting in a success rate of 0.98. The efficiency of the resulting trajectories compared to the direct distance amounts to 0.94. In the only attempt where the navigation failed, the robot got obviously stuck in a local minimum close to the target location. Example trajectories from successful navigation trials as well as the only failure are shown in Fig. 7.3.

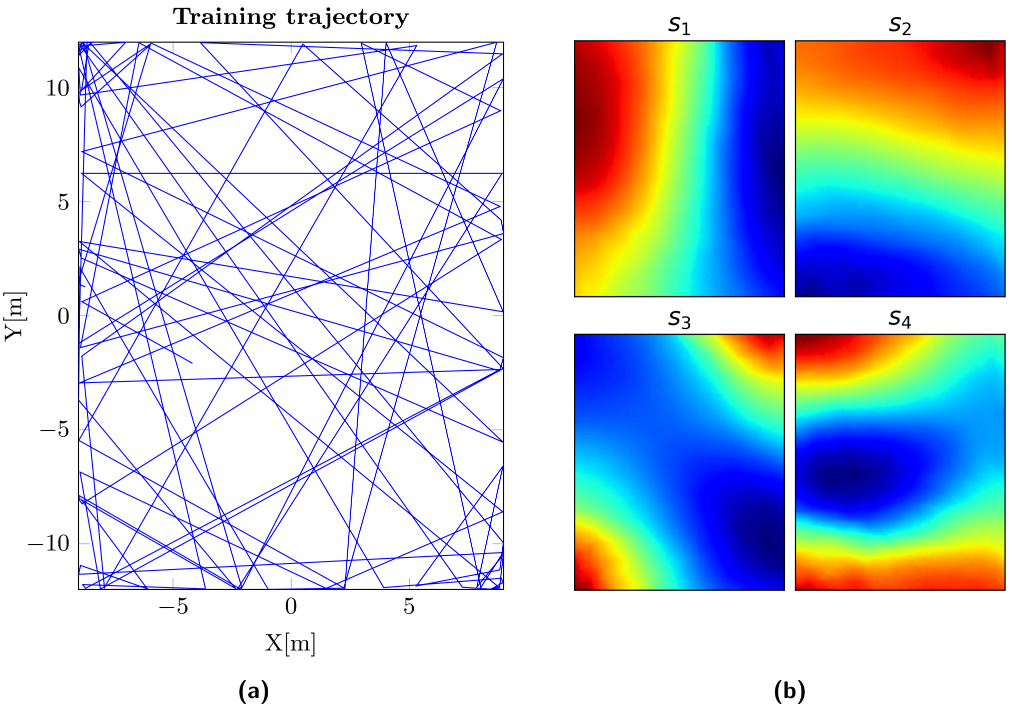


Figure 7.2: (a) The training trajectory consists of 5000 positions along line segments with random orientation. (b) Spatial firing maps of the first two SFA functions clearly encode the position along the coordinate axes illustrated by the characteristic gradients. Function three is a mixture of the first two functions, whereas function four is a higher mode of the second one.

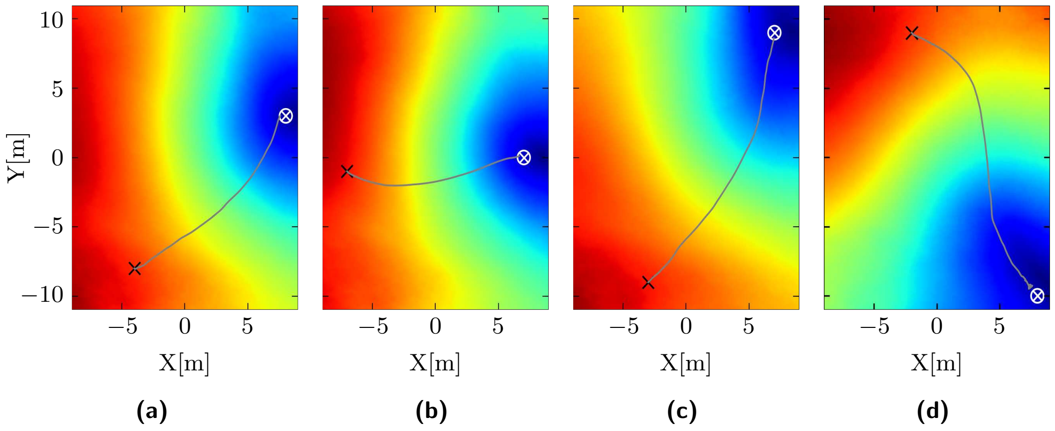
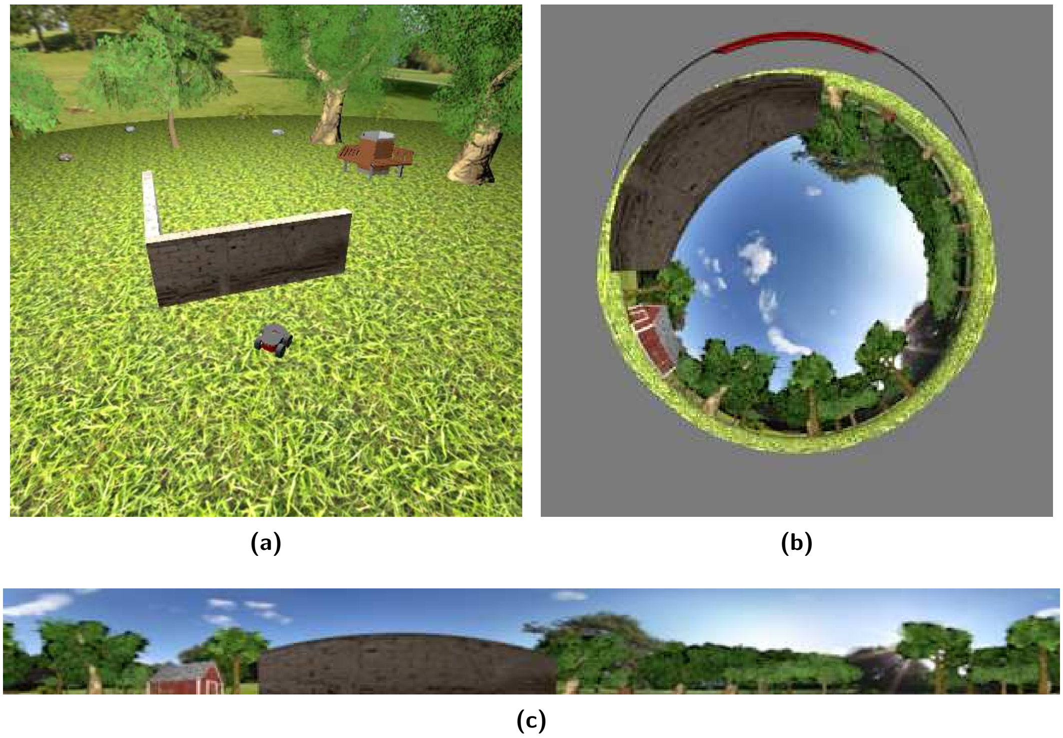


Figure 7.3: Resulting trajectories. The start and target positions are marked by a black cross and a white circle, respectively. (a)-(c) The robot successfully navigated to the target position performing gradient descent on the first two slow feature outputs. (d) In one out of the 50 trials the robot got stuck in a local minimum in close proximity to the target location.

Navigation with an Obstacle



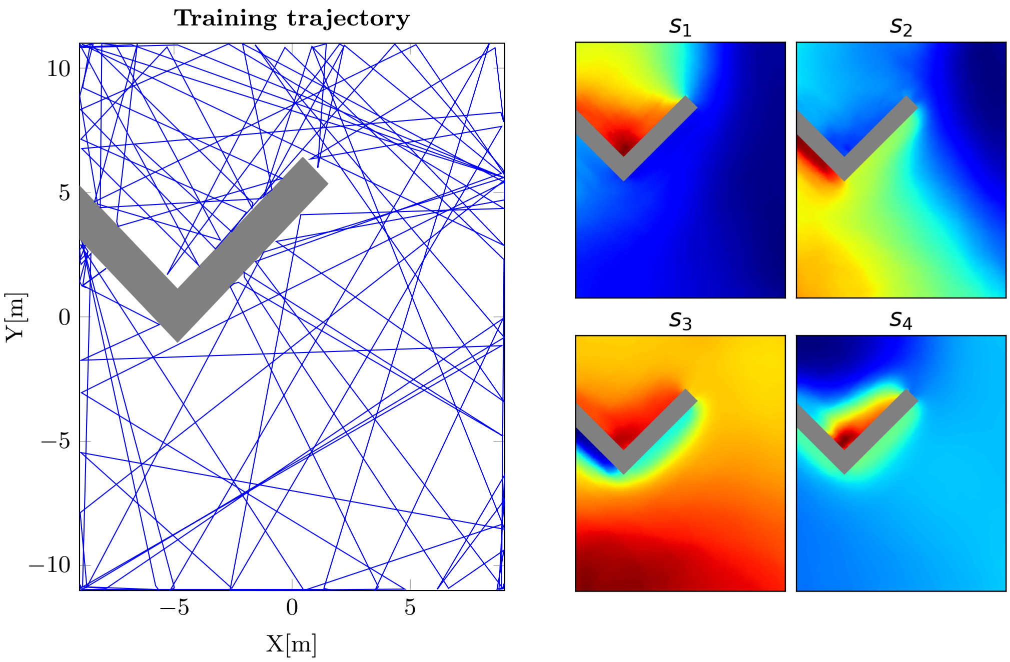
(a)

(b)

(c)

Figure 7.4: Simulated environment with an obstacle. (a) A v-shaped obstacle is placed in the simulated garden like environment. (b) Example image from the omnidirectional camera. (c) The field of view of the unwarped panoramic images is cropped to discard static image regions.

For the next experiment we placed a v-shaped obstacle in the scene to validate the assumption that the slow feature representations implicitly encode information about obstacles in the scene and allow the robot to circumnavigate obstacles by simply following the steepest gradient. The simulator environment with an obstacle placed in the scene, as well as a rendered image from one of the omnidirectional cameras and the unwarped panoramic view are illustrated in Fig. 7.4. The target location is kept fixed on the upper side of the obstacle, since it is the most interesting configuration, while the starting locations are randomly drawn from the lower half of the training area on the opposite side of the obstacle. Again, we performed 50 trials using SFA outputs s_1 and s_2 from the learned environmental representation. The random training trajectory consisting of 5000 samples is shown in Fig. 7.5a. The spatial firing maps of the first four SFA-outputs are illustrated in Fig. 7.5b. The implicit encoding of the obstacle is clearly visible in the maps of the first two functions. The gradients gradually change with position along the axes of rotated coordinate system and flow around the corners of the obstacle where most of the variance is encoded. To compute the efficiency of the resulting trajectories we discretized the training area into an occupancy grid with a cell size of 0.1 units and applied A* to obtain the optimal path.



(a) **(b)**

Figure 7.5: (a) The training trajectory consists of 5000 positions along line segments with a random orientation. (b) The spatial firing map of the first SFA function contains the steepest gradient around the upper right corner of the obstacle where most of the variance is concentrated. The gradients in the spatial firing map of the second function are steepest around the lower corner of the obstacle. The spatial firing maps of the third and fourth function are difficult to interpret.

Results The robot reached the target location in 88% of the trials, successfully circumnavigating the obstacle following a nearly optimal trajectory with a mean efficiency of 76%. Examples of the resulting trajectories from the experiment are illustrated in Fig. 7.6. In case of a failure, the robot got stuck in a local minima. Since most of the variance is concentrated in regions near the obstacle the gradients for large parts of the training area are relatively flat. Using a more sophisticated method for gradient descent, these failure cases could probably be resolved.

7.1.3 Discussion

The results from the experiments have demonstrated that navigation can be performed directly in slow feature space using gradient descent. Since the resulting slowest two

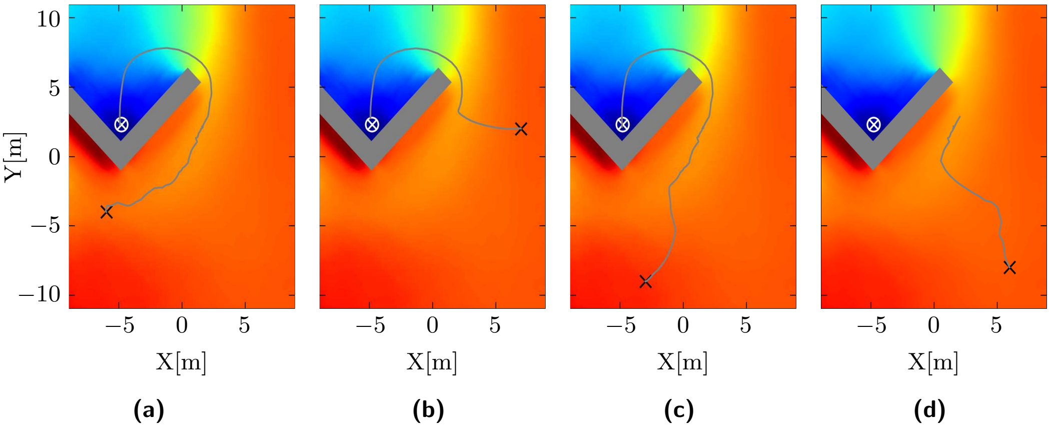


Figure 7.6: Resulting trajectories. The start and target positions are marked by a black cross and a white circle, respectively. (a)-(c) Using the first two slow feature outputs enables the robot to successfully navigate around the obstacle to the target location. (d) Since most of the variance of the slow feature representations is concentrated near the obstacle the gradients in large parts of the training area are rather flat. Therefore the robot got stuck in some of the trials where the SFA-output at the start location was already similar to the value in the flat region.

SFA-outputs ideally encode the x- and y-coordinate as half cosine-/sine-functions the cost surface is very likely going to have a global minimum. A navigation direction can be obtained very efficiently by approximating the gradient from three close-by measurements of the cost function. In the open field simulator experiments the robot reached the target in 98% of the trials with an efficiency of 0.94 compared to the direct distance. The presence of an obstacle determines the mean temporal distance between points on opposite sides which leads to an implicit encoding in the slow feature representations. Hence, circumnavigating obstacles requires no explicit planning of the trajectory but is accomplished by simply following the steepest gradient. In the experiments where the target position was behind an obstacle the navigation was successful in 88% of the trials with an efficiency of 0.76. In the failure cases the robot got stuck in regions with flat gradients. A more advanced gradient descent algorithm could cope with these cases and make the navigation more robust which would also be crucial for applying the method in real world scenarios.

7.2 Future Perspectives for Navigation in Slow Feature Space

The results from the simulator experiments in the previous section have demonstrated the feasibility of performing navigation directly in slow feature space using gradient descent. The following section presents an extension to the proposed method for SFA-navigation which allows to integrate information from higher functions. We furthermore investigate the effect of different velocity distributions within the training area on the resulting representations and the navigation behavior. The preliminary results from the experiments are supposed to serve as an outlook on future research directions and show further potentials of using slow feature representations for navigation. However, a thorough investigation and validation of the presented method and observations is beyond the scope of this thesis.

7.2.1 Navigation with Weighted Slow Feature Representations

For the navigation experiments in the simulator described in the previous section only the slowest two SFA-outputs have been used for navigation. In a static environment the slowest two functions ideally encode the robot's $x-$ and $y-$coordinate as half cosine-/sine-waves [42]. Hence, they change monotonically over space and are orthogonal. In this case, the first two functions are sufficient to represent the position of the robot and the cost function C will have a global minimum. Using more SFA-outputs leads to local minima in the cost surface of C since later functions represent higher modes of previous ones. However, in real world environments the resulting slow feature representations might differ from the theoretical optimal solutions and information from higher functions might be necessary to fully reconstruct the robot's position and perform navigation (cf. 5.1.2).

Weighting the Slow Feature Representations

In order to integrate information from later SFA functions and at the same time prevent the cost function C from having local minima we propose to use a weighting function for the slow feature outputs. The weights should decrease for additional SFA-outputs so that slower ones have a higher impact on the resulting cost value. An intuitive way to select such weights without relying on additional parameters is to use the output signal's slowness value. Here, we use the β-value [11] which is defined as $\beta(s_n) = (1/2\pi)\sqrt{\Delta(s_n)}$, where $\Delta(s_n)$ is the mean of the squared temporal derivative of SFA-output signal s_n. Since the SFA functions are ordered by their slowness the β-value increases for later ones. The output signal of the slowest function encoding information about the robot's x- or y-coordinate will ideally take the form of a half cosine-/sine-wave. The next higher mode of this function will then be equal to a full cosine-/sine-wave so that its corresponding β-value will be twice as high. Therefore, we propose to use the inverse of the β-value as a non-parametric solution to assign decreasing weights to the slow feature outputs used for navigation.

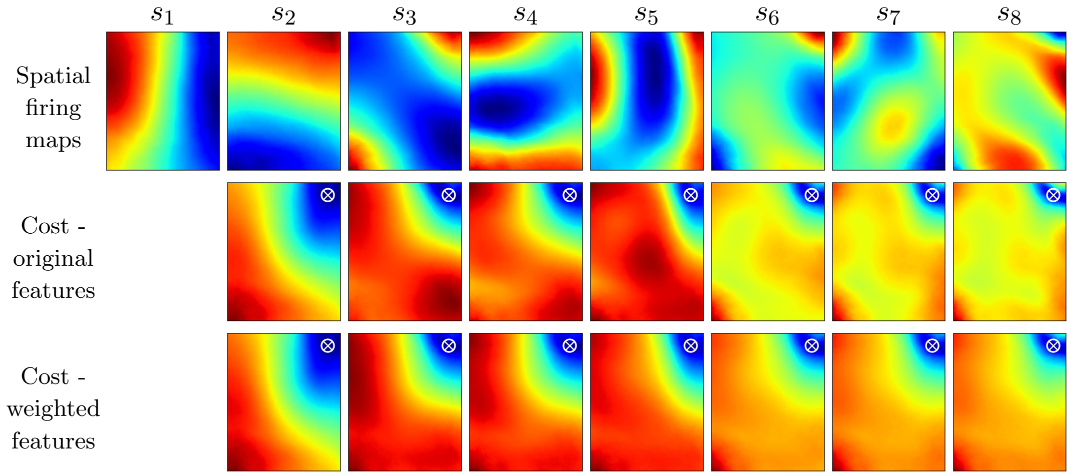


Figure 7.7: Spatial firing maps and cost surfaces. The top row shows the spatial firing maps of the eight slowest SFA-outputs $s_1 \dots s_8$. The first two functions encode the position of the robot on the x- and y-axis which is illustrated by the characteristic gradients along the coordinate axes. Later functions are mixtures and higher modes of previous ones. The cost surfaces show the color coded distance in slow feature space from every position to the target location indicated by the white cross. The dimensionality increases from two SFA-outputs on the left to eight outputs on the right. For the original SFA-outputs the surface of the cost functions develops an increasing number of local minima when more outputs are included (middle row, left to right). Using the inverse of an output's β-value as a weighting factor reduces the impact of faster SFA-outputs so that the general characteristic of the cost surface is preserved (last row, left to right.

Navigation Experiments with Weighted Slow Feature Representations

To investigate the effect of using more than two SFA-outputs on navigation performance we repeated the open field simulator experiment from section 7.1.2 using the original outputs $s_1 \dots s_8$ as well as their values weighted by the inverse of the corresponding β-values. For the navigation experiment using the original slow features the gradient scaling factor was set to $\eta = 1.0$ whereas it was set to $\eta = 0.0015$ using the weighted slow features. The momentum term was set to $\gamma = 0.5$ for both experiments.

Results As expected the surface of cost function C contains an increasing number of local minima when using more SFA-outputs of the learned representation. Weighting the outputs by the inverse of their β-value decreases the impact of later functions, which potentially represent higher modes of previous ones, and preserves the overall characteristic of the SFA gradients. The spatial firing maps of the SFA-outputs $s_1 \dots s_8$ as well as the cost surfaces from the original and weighted outputs are illustrated in Fig. 7.7.When the original slow features are used for navigation the success rate amounts to 0.98 for the slowest two outputs with an efficiency of 0.94. The success rate decreases significantly for more than three outputs and the target is reached only in 16% of the trials if all eight outputs are used. In the navigation experiments with the slow features that have been weighted by the inverse of their β-values the best performance is achieved when using the first three outputs with a success rate of 0.98 and an efficiency of 0.92. Navigation with an increasing number of slow feature outputs leads to a slight decrease in the performance while the target was still reached in 86% of the trials with an efficiency of 0.86 using all eight SFA-outputs. The resulting navigation performances for the original and the weighted slow features are illustrated and compared in Fig. 7.8.

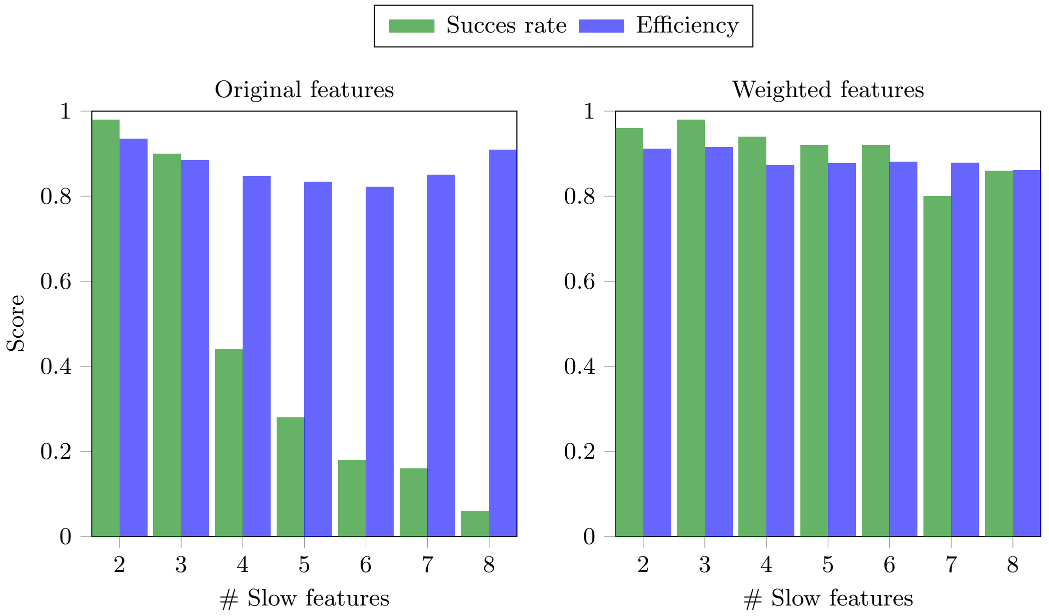


Figure 7.8: Navigation results for an increasing number of SFA-outputs. With the original features the navigation performance is best when using the first two slowest features with a success rate of 0.98 and an efficiency of 0.94. The success rate drops significantly with additional SFA-outputs and amounts to 0.16 when using the outputs of the eight slowest functions. Navigation performance with the weighted slow features is only slightly reduced when increasing the number of outputs. The best performance is achieved for three SFA-outputs with a success rate of 0.98 and an efficiency of 0.92. Navigation with eight slow feature outputs is successful in 86% of the trials with an efficiency of 0.86.

Discussion

The results from the experiment have shown that navigation in slow feature space performing gradient descent breaks down when more than two slow feature outputs are used without weighting. Later SFA-outputs usually represent higher modes of previous ones which leads to local minima in the cost surface which leads to a significant drop in performance. Using the inverse of an outputs' β-value $\beta(s_n)$, which is a measure of its slowness, as a weighting factor is an intuitive and non-parametric way to include information from higher functions and at the same time to reduce the emergence of local minima. Using all eight weighted slow features the virtual robot was still able to reach the target in 86% of the trials.

7.2.2 Implicit Optimization of Traveling Time

Usually there exist many possible paths one could choose in order to navigate from the current position to a given target location. The selection of a viable path is in general based on some optimization criteria depending on the specific scenario. A prey animal will not follow the direct path to a food source leading through an open field but instead prefer to make a detour to be covered by bushes. When steering a large vehicle on a construction site one might want to minimize the risk of a collision and thus consider the distance to obstacles for path planning. In most scenarios, however, the criteria for an optimal path is the distance, the time of travel or a weighted combination of both.

The results from the navigation experiments in an open field scenario from section 7.1.2 have demonstrated that the trajectories obtained by performing gradient descent in slow feature space are close to the optimal ones given by the direct distance. In the experiment the robot drove with a constant velocity throughout the whole area. In real world scenarios, however, the robot might pass regions with different conditions of the underground e.g. grass, sand or asphalted street during exploration of the environment. Thus, the velocity might vary for different regions of the environment depending on the underground. In such a scenario the direct path to a target might not be optimal in terms of traveling time.

It has been observed that dogs seem to consider the difference in velocities for running and swimming when planning a trajectory [122]. In an experiment a ball was thrown from the shore into a lake and it has been measured at which point the dog decided to stop running on the shore and jumped into the water to swim the remaining distance. Instead of directly starting to swim to the ball, which would have been the shortest path, or running along the shore until being on the same level as the ball the dog instead chose a transition point which was near to the theoretically optimum w.r.t. time.

We assume that variations in the velocity within different regions of the environment will be reflected in the slow feature outputs and thus affect the navigation behavior resulting in an implicit optimization of traveling time. For the theoretical optimal solutions the variance of the resulting SFA functions is equally distributed over time [42]. Therefore, the variance over space will be larger within low velocity regions where the distance traveled per time step is small. Following the slow feature gradient a mobile robot should thus navigate around low velocity regions and stay within high velocity regions if the difference is significantly large. Thereby, the traveling time is implicitly minimized.

Experiments

As a first proof of concept we tested the navigation behavior with slow feature representations directly learned from the coordinates along a trajectory. This should be equivalent to the representations learned from corresponding images in the ideal case [42]. We performed two experiments where the environment is divided into a low and a high velocity region. In the first experiment the environment is split vertically and the start and target locations are within the different velocity regions (see Fig. 7.9a). In the second experiment a circular area with low translational speed is in the center of the environment surrounded by the high velocity region (see Fig. 7.10a). Here, the navigation task is to get from one side of the circle to the opposite site such that the direct path leads through the low velocity region. In the high velocity regions the translational movement is three times higher than in the low velocity regions. The training trajectory consists of 5000 samples along line segments with a random orientation.
In the first experiment, where the environment is vertically split into a low and high velocity region, the optimal trajectory is determined by setting up an equation for the time of travel w.r.t. the start and target positions, the velocities and the transition point between the regions. The optimal transition point is given by the zero crossing of the first derivative. The optimal trajectory for the second experiment was determined by applying A* to a discretized grid. To account for the different velocities the distance estimations have been weighted by a factor of three which is equivalent to the ratio of the velocities. For the navigation experiments we used the slowest eight SFA-outputs weighted by the inverse of their β-values.

Results The different velocities clearly affect the resulting SFA-functions which is illustrated in the spatial firing maps. Since the variance of the SFA-outputs is equally distributed over time most of it is concentrated in the low velocity regions. The maps of the first four SFA-outputs from both experiments are shown in Fig. 7.9b and Fig. 7.10b respectively. In the first experiment, where the low and high velocity regions are separated by a vertical line, the trajectory closely follows the fastest route which is defined by the optimal transition point w.r.t. the start and target locations and the respective velocities (see Fig. 7.9c). In the second experiment the gradients flow around the low velocity region located in the center of the environment. The trajectory resulting from gradient descent thus leads around this region and is close to the optimal one obtained with A*. The trajectory is shown in Fig. 7.10c.

Discussion

If the constitution of the underground changes within the working area the maximum achievable velocity can be affected. For such a scenario the direct path to a target location might not be the optimal one in terms of traveling time. The preliminary

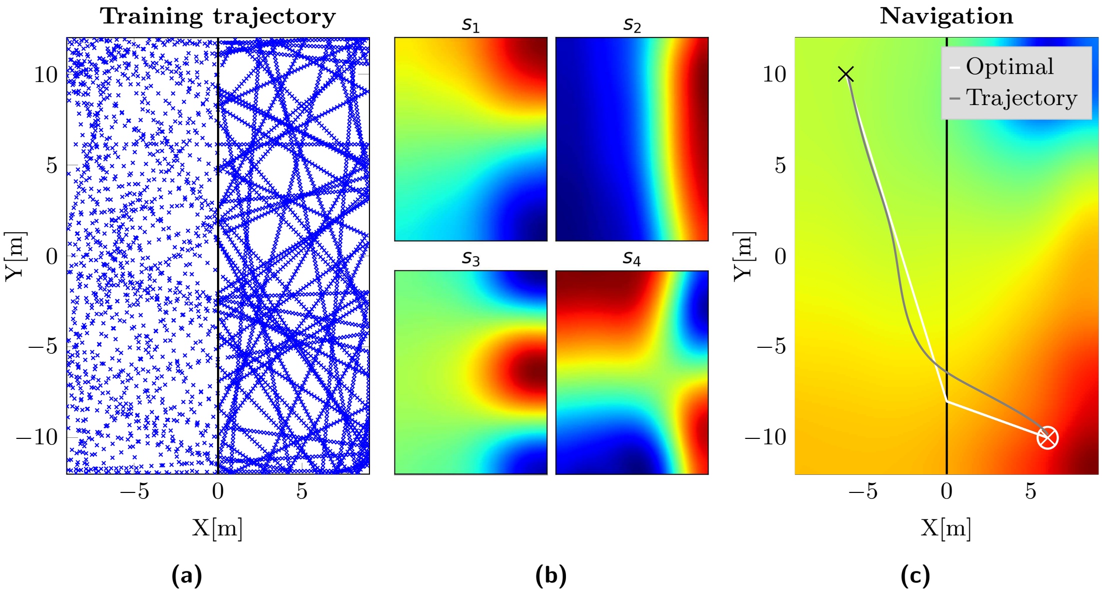


Figure 7.9: (a) During the training phase the velocity in the left half is three times higher than in the right half. The separation between the high and low velocity regions is indicated by the black line. (b) Spatial firing maps of the four slowest functions show distortions caused by a higher amount of variance in the region with a lower velocity. (c) The trajectory resulting from gradient descent, indicated by the gray line, closely follows the optimal one with minimal traveling time indicated by the white line.

results from the experiments, using the coordinates from a random training trajectory, have shown that regional differences in the velocity are reflected in the slow feature representations confirming the theoretical derivations in [42]. The variance over space of the resulting SFA-outputs is higher for low velocity regions. Following the slow feature gradients thus leads to trajectories that tend to stay within the high velocity regions resulting in an implicit optimization of the traveling time.
However, it has to be further investigated if the results can be reproduced if the SFA-model is trained with real world images. In this case information about the velocity of a mobile robot is embedded in the high dimensional image data and their perception depends on the spatial layout of the scene, i.e. the distance to objects.

7.3 Conclusion

We presented a straightforward and efficient method for navigating directly in slow feature space using gradient descent. The slow feature representations are learned for a specific environment in an offline learning phase where the robot randomly samples the environment. After the unsupervised learning step a navigation direction can be

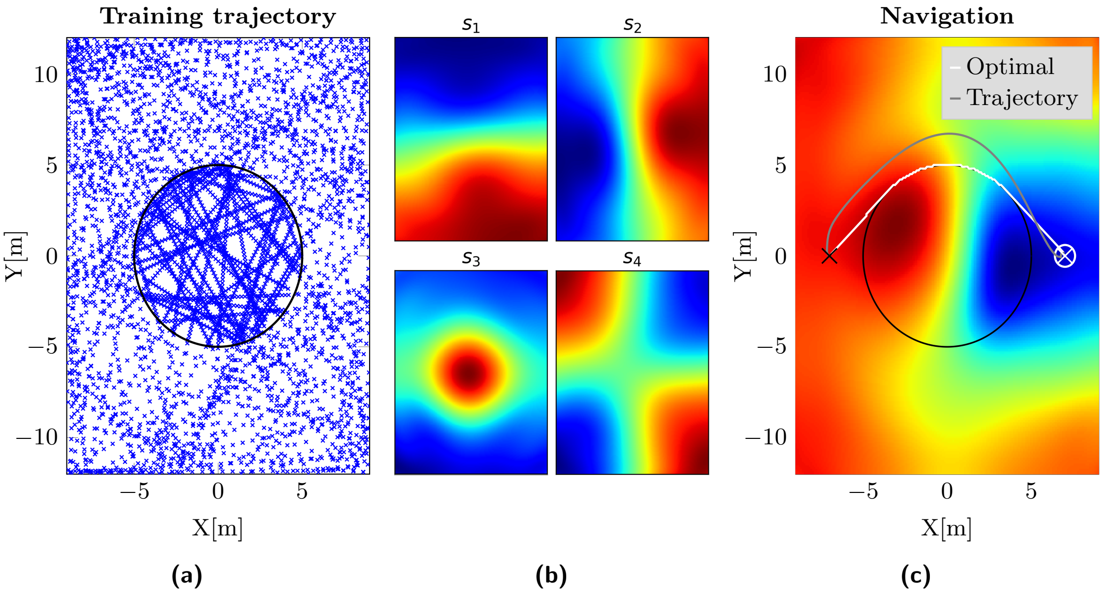


Figure 7.10: (a) During the training phase the velocity in the inner circle is three times lower than in the surrounding region. The black circular line indicates the transition from the low to the high velocity region. (b) Since the amount of variance of the SFA-outputs is higher in the low velocity region the spatial firing maps of the four slowest functions show distortions. (c) The resulting trajectory, illustrated by the gray line, completely leads around the low velocity region and is close the optimal one indicated by the white line.

obtained very efficiently from three close-by evaluations of the cost-function which computes the distance in slow feature space to the value at the target location. Information about obstacles is implicitly encoded in the learned slow feature representations which is reflected in the resulting gradients. Hence, circumnavigating obstacles requires no explicit planning of the trajectory but is accomplished by simply following the steepest gradient. In the simulator experiments the robot reached the target in almost 100% of the trials in an open field scenario and in 88% of the trials when the target location was behind an obstacle using the two slowest SFA-outputs. In the failure cases the robot got stuck in regions with flat gradients. A more advanced gradient descent algorithm could cope with these cases and make the navigation more robust which would also be crucial for applying the method in real world scenarios.

In addition to the fundamental approach of navigation by gradient descent in slow feature space we also presented some preliminary results on further perspectives regarding the use of additional features for navigation and the implicit optimization of traveling time.

In cases where the learned slow representations deviate from the optimal solutions it might be necessary to include information from later functions in order to fully reconstruct the position of the robot. To avoid the emergence of local minima of the cost function C resulting from higher modes we used the inverse of an outputs' β-value to weight the slow feature representations. The β-value is a measure of an output's temporal variation and thus an intuitive and non-parametric way to obtain feasible weights. The results from the navigation experiment with eight slow feature outputs have shown that the weighting drastically improves the robustness of the gradient based navigation as it preserves the general characteristics of the cost surface when using more than the two slowest SFA-outputs.

In the experiments with slow feature representations learned directly from the coordinates of a random training trajectory we have shown that differences in the velocities within the environment are reflected in the SFA-outputs. The resulting trajectories are close to optimal w.r.t. traveling time since the SFA-gradients preferably lead through high velocity regions.

Although the initial experiments demonstrated the feasibility of the approach it has to be validated in real world experiments in future work. For the application of the method in real world scenarios it might be beneficial to move the robot along the estimated gradient direction with a fixed step size. The accuracy of the learned SFA representation could be estimated using the unsupervised metric learning method described in section 5.3 to set the minimal step size accordingly. This way the gradient estimations could become more robust to noise in the image data and consequently in the SFA-outputs.

8 Summary and Conclusion

This thesis approached the fundamental problems of self-localization, the creation of robust environmental representations and navigation with a mobile robot using vision as the only sensory input. The proposed methods build upon a biologically motivated model for rat navigation based on unsupervised Slow Feature Analysis (SFA). The model extracts a spatial representation of the environment by directly processing the visual input from a mobile robot in a hierarchical SFA-network. The use of an omnidirectional vision system allows to learn orientation invariant representations of the robot's location by modifying the perceived image statistics through additional simulated rotational movement. The resulting SFA-outputs encode the position of the robot as slowly varying features while at the same time being invariant to its orientation.

The model was first validated in a simulator environment and then compared to state-of-the-art visual SLAM methods in real world indoor and outdoor experiments. Although the model is conceptually simple, in the sense that it is based on a single unsupervised learning rule, the presented experiments have proven that the learned SFA representation enables a precise localization with accuracies that are on par or even superior compared to the state-of-the-art SLAM methods. To enable the integration of ego-motion estimates from odometry and to communicate the learned representations to a potential user in real world application scenarios we introduced a method for the unsupervised learning of a mapping from slow feature to metric space. Capturing odometry-based distance measurements and the corresponding slow feature outputs for points along several straight line trajectories allows to obtain two independent estimates for each point. The line parameters and the weights for the mapping can be learned simultaneously by minimizing the difference between both point estimates. An alternative approach for learning spatial representations from tracked landmark views instead of using the whole panoramic images was proposed. The resulting localization performance is comparable to the original model while the alignment of the views to a canonical orientation largely reduces the training cost. Using multiple marker views has been shown to further improve localization accuracy and allows to deal with occluding objects. However, the transfer of the approach to real world scenarios requires a robust method for the detection and identification of suitable landmarks which might be tackled in future work.

In long-term outdoor scenarios, environmental effects like dynamic objects, different daytimes, weather conditions and seasonal changes drastically impact the appearance of a place and thus pose a severe problem for vision based mapping and localization methods. We proposed a method for predicting the robustness of visual features which are commonly used in localization and mapping scenarios but might also serve to create alternative image representation for SFA-learning. A classification model was trained with cross seasonal images from corresponding places in order to discriminate between stable and unstable features. Experimental results have shown an increased performance in cross season feature matching compared to the conventional feature selection based on the feature detector response alone. Since the model can be easily incorporated into the standard feature processing pipeline for stable feature selection it is applicable in a broad range of approaches. A further performance improvement might be achieved by the use of lighting invariant descriptors (e.g. [18, 79]).

As an alternative approach for obtaining robust environmental representations we presented a unified approach which is solely based on the invariance learning capabilities of the SFA-model. First, we tackled the problem of slowly changing environmental variables during training which might interfere with the spatial coding. The identification of loop-closures in the training trajectory allows to change the perceived image statistics by re- inserting images of the same place from the past in the temporally ordered image sequence. Thereby, the perceived variation of environmental effects is increased and the unsupervised SFA-learning algorithm is provided with a self-generated supervisory signal regarding its slowness objective. Results from the experiments have demonstrated that feedback from loop-closures improves robustness especially for changing lighting conditions.

In order to learn invariant representations for long-term robust outdoor localization we extended the approach to recordings along the same trajectory in different conditions. Establishing dense position correspondences between recordings in different conditions allows to create a training sequence where the perceived environmental condition changes faster than the position. This requires the SFA-model to learn representations that are invariant w.r.t. environmental changes in order to extract the slowly varying position. Results from simulator and real world experiments have shown that the model learns an increasingly invariant representation of the environment using data sets from different conditions. It needs to be investigated in future work in which way condition invariance and orientation invariance learning can be combined in an optimal way. It would also be interesting to explore the generalization capabilities of the slow features learned in lower layers to unseen environments from the same domain.

A novel method for efficient navigation in slow feature space using gradient descent was presented. The slow feature representations are learned for a specific environment in an offline learning phase. After the unsupervised learning step a navigation direction can be obtained very efficiently from three close-by evaluations of the cost-function which computes the distance in slow feature space from the current to a target location. Obstacles are implicitly encoded in the learned slow feature representations and are reflected in the resulting gradients. Hence, circumnavigating obstacles is accomplished by simply following the SFA gradients and requires no explicit trajectory planning. Using the first two slowest SFA-outputs for navigation in a simulator environment, the target was reached in almost 100% of the trials in an open field scenario and in 88% of the trials when the target location was behind an obstacle. A more advanced gradient descent algorithm might resolve the failure cases where the robot got stuck in regions with flat gradients.

To account for deviations from the theoretical optimal solutions in real world navigation scenarios it might be necessary to use additional SFA-outputs for gradient estimation. However, the simple integration of additional SFA-outputs will inevitably lead to local minima in the cost function due to higher modes of previous ones. Therefore, we used the inverse of an outputs' β-value to weight the slow feature representations for gradient estimation. The β-value is a measure of an output's temporal variation and thus an intuitive and non-parametric way to obtain feasible weights. Results from the simulator experiment have demonstrated robust navigation with up to eight slow feature outputs. The preliminary results from experiments with different velocity distributions in the environment suggest that these differences are encoded in the learned slow feature representations and lead to trajectories that implicitly optimize for traveling time. Although the simulator experiments demonstrated the feasibility of gradient descent based SFA navigation it remains to be validated in real world experiments in future work.

Research during the last decades has made great progress in the fields of visual localization and mapping, long-term robustness and navigation. However, these problems have often been approached individually, not considering the system as a whole. Geometric methods based on sparse feature matching or semi-dense image alignment represent the current state-of-the-art in terms of localization and mapping accuracy but do not consider long-term robustness. Furthermore, due to their sparseness the created environment representations are not suitable for trajectory planning. Methods achieving long-term robustness generally trade off localization accuracy for improved invariance using less specific feature representations or constrain the problem of localization and mapping to certain types of trajectories. This thesis has shown that the conceptually simple approach of unsupervised SFA learning can serve as a basis to implement methods for all aspects of mobile robot navigation. Considering the promising results achieved in this early stage of research it might become a viable alternative to the established methods in the future.

Bibliography

[1] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[2] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 510–517, 2012.

[3] A. Angeli, S. Doncieux, J. Meyer, and D. Filliat. Visual topological SLAM and global localization. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, pages 4300–4305, 2009.

[4] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5297–5307, 2016.

[5] A. Arleo and W. Gerstner. Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, 83(3):287–299, Aug 2000.

[6] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987.

[7] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics Automation Magazine*, 13(3):108–117, 2006.

[8] J. Barraquand and J.-C. Latombe. Robot Motion Planning: A Distributed Representation Approach. *I. J. Robotics Res.*, 10(6):628–649, 1991.

[9] A. Barrera and A. Weitzenfeld. Biologically-inspired robot spatial cognition based on rat neurophysiological studies. *Autonomous Robots*, 25(1):147–169, Aug 2008.

[10] H. Bay, T. Tuytelaars, and L. J. V. Gool. SURF: Speeded Up Robust Features. In *ECCV, Austria*, pages 404–417, 2006.

[11] P. Berkes and L. Wiskott. Slow feature analysis yields a rich repertoire of complex cell properties. *Journal of Vision*, 5(6):9, 2005.

[12] W. Böhmer, S. Grünewälder, Y. Shen, M. Musial, and K. Obermayer. Construction of approximation spaces for reinforcement learning. *Journal of Machine Learning Research*, 14(1):2067–2118, 2013.

[13] F. Bonin-Font, A. Ortiz, and G. Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296, 2008.

[14] O. Booij, B. Terwijn, Z. Zivkovic, and B. J. A. Kröse. Navigation using an appearance based topological map. In *2007 IEEE International Conference on Robotics and Automation, ICRA 2007, 10-14 April 2007, Roma, Italy*, pages 3927–3932, 2007.

[15] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robotics*, 32(6):1309–1332, 2016.

[17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 778–792, 2010.

[18] N. Carlevaris-Bianco and R. 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, sep 2014.

[19] W. Churchill and P. M. Newman. Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation. In *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pages 4525–4532, 2012.

[20] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós. Mapping large loops with a single hand-held camera. In *Robotics: Science and Systems III, June 27-30, 2007, Georgia Institute of Technology, Atlanta, Georgia, USA*, 2007.

[21] M. Collett, L. Chittka, and T. Collett. Spatial memory in insect navigation. *Current Biology*, 23(17):R789 – R800, 2013.

[22] M. Cummins and P. M. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *I. J. Robotics Res.*, 30(9):1100–1123, 2011.

[23] M. J. Cummins and P. M. Newman. FAB-MAP: probabilistic localization and mapping in the space of appearance. *I. J. Robotics Res.*, 27(6):647–665, 2008.

[24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005.

[25] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 1403–1410, 2003.

[26] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, 2007.

[27] F. Dayoub and T. Duckett. An adaptive appearance-based map for long-term topological localization of mobile robots. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 22-26, 2008, Acropolis Convention Center, Nice, France*, pages 3364–3369, 2008.

[28] F. Dellaert and M. Kaess. Square root SAM: simultaneous localization and mapping via square root information smoothing. *I. J. Robotics Res.*, 25(12):1181–1203, 2006.

[29] A. Dosovitskiy, J. T. Springenberg, M. A. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. pages 766–774, 2014.

[30] G. Dudek and M. R. M. Jenkin. *Computational principles of mobile robotics.* Cambridge University Press, 2000.

[31] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): part I. *IEEE Robotics Automation Magazine*, 13(2):99–110, 2006.

[32] E. Eade and T. Drummond. Scalable monocular SLAM. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 469–476, 2006.

[33] N. Einecke, J. Deigmöller, K. Muro, and M. Franzius. Boundary wire mapping on autonomous lawn mowers. In *Field and Service Robotics, Results of the 11th International Conference, FSR 2017, Zurich, Switzerland, 12-15 September 2017*, pages 351–365, 2017.

[34] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22(6):46–57, 1989.

[35] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II*, pages 834–849, 2014.

[36] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 1449–1456, 2013.

[37] A. N. Escalante and L. Wiskott. How to solve classification and regression problems on high-dimensional data with a supervised extension of slow feature analysis. *Journal of Machine Learning Research*, 14(1):3683–3719, 2013.

[38] A. N. Escalante and L. Wiskott. How to solve classification and regression problems on high-dimensional data with a supervised extension of slow feature analysis. *Journal of Machine Learning Research*, 14(1):3683–3719, 2013.

[39] A. N. Escalante and L. Wiskott. Improved graph-based SFA: Information preservation complements the slowness principle. *CoRR*, abs/1601.0, 2016.

[40] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, jun 1981.

[41] P. Földiák. Learning Invariance from Transformation Sequences. *Neural Computation*, 3(2):194–200, 1991.

[42] M. Franzius, H. Sprekeler, and L. Wiskott. Slowness and Sparseness Lead to Place, Head-Direction, and Spatial-View Cells. *PLoS Computational Biology*, 3(8):1–18, 2007.

[43] M. Franzius, N. Wilbert, and L. Wiskott. Invariant object recognition and pose estimation with slow feature analysis. *Neural Computation*, 23(9):2289–2323, 2011.

[44] F. Fraundorfer, C. Engels, and D. Nistér. Topological mapping, localization and navigation using image collections. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29 - November 2, 2007, San Diego, California, USA*, pages 3872–3877, 2007.

[45] F. Fraundorfer and D. Scaramuzza. Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications. *Robotics Automation Magazine, IEEE*, 19(2):78–90, jun 2012.

[46] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[47] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[48] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, 94(3):335–360, 2011.

[49] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual slam. *Machine Vision and Applications*, 21(6):905–920, Oct 2010.

[50] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587, 2014.

[51] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Intell. Transport. Syst. Mag.*, 2(4):31–43, 2010.

[52] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Robotics: Science and Systems III, June 27-30, 2007, Georgia Institute of Technology, Atlanta, Georgia, USA*, 2007.

[53] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, September, 1988*, pages 1–6, 1988.

[54] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[55] R. I. Hartley and P. F. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[56] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[57] J. Heinly, E. Dunn, and J. Frahm. Comparative evaluation of binary features. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*, pages 759–773, 2012.

[58] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artif. Intell.*, 17(1-3):185–203, 1981.

[59] K. J. Jeffery and J. M. O'Keefe. Learned interaction of visual and idiothetic cues in the control of place field orientation. *Experimental Brain Research*, 127(2):151–161, 1999.

[60] W. Y. Jeong and K. M. Lee. CV-SLAM: a new ceiling vision-based SLAM technique. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2-6, 2005*, pages 3195–3200, 2005.

[61] E. Johns and G.-Z. Yang. Dynamic scene models for incremental, long-term, appearance-based localisation. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 2731–2736, 2013.

[62] E. Johns and G.-Z. Yang. Feature Co-occurrence Maps: Appearance-based localisation throughout the day. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 3212–3218, 2013.

[63] M. Kaess, A. Ranganathan, and F. Dellaert. isam: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, 2008.

[64] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.

[65] A. Kawewong, N. Tongprasit, S. Tangruamsub, and O. Hasegawa. Online and incremental appearance-based SLAM in highly dynamic environments. *I. J. Robotics Res.*, 30(1):33–55, 2011.

[66] A. Kelly. A 3d space formulation of a navigation kalman filter for autonomous vehicles. Technical Report CMU-RI-TR-94-19, Carnegie Mellon University, Pittsburgh, PA, May 1994.

[67] A. Kendall and R. Cipolla. Geometric loss functions for camera pose regression with deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6555–6564, 2017.

[68] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2938–2946, 2015.

[69] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, Missouri, USA, March 25-28, 1985*, pages 500–505, 1985.

[70] G. Klein and D. W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Sixth IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, 13-16 November 2007, Nara, Japan*, pages 225–234, 2007.

[71] L. Kneip, M. Chli, and R. Siegwart. Robust real-time visual odometry with a single camera and an IMU. In *British Machine Vision Conference, BMVC 2011, Dundee, UK, August 29 - September 2, 2011. Proceedings*, pages 1–11, 2011.

[72] K. Konolige and J. Bowman. Towards lifelong visual maps. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, MO, USA*, pages 1156–1163, 2009.

[73] K. P. Körding, C. Kayser, W. Einhäuser, and P. König. How are complex cell properties adapted to the statistics of natural stimuli? *Journal of neurophysiology*, 91(1):206–212, 2004.

[74] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[75] T. Kühnl, F. Kummert, and J. Fritsch. Monocular road segmentation using slow feature analysis. In *IEEE Intelligent Vehicles Symposium (IV), 2011, Baden-Baden, Germany, June 5-9, 2011*, pages 800–806, 2011.

[76] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G^{2}o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 3607–3613, 2011.

[77] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30(1-2):39–64, 2000.

[78] H. Lategahn, J. Beck, B. Kitt, and C. Stiller. How to learn an illumination robust image feature for place recognition. In *2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, June 23-26, 2013*, pages 285–291, 2013.

[79] H. Lategahn, J. Beck, and C. Stiller. DIRD is an illumination robust descriptor. In *2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, June 8-11, 2014*, pages 756–761, 2014.

[80] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller. Urban localization with camera and inertial measurement unit. In *2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, June 23-26, 2013*, pages 719–724, 2013.

[81] R. Legenstein, N. Wilbert, and L. Wiskott. Reinforcement Learning on Slow Features of High-Dimensional Input Streams. *PLoS Computational Biology*, 6(8):1–13, 2010.

[82] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2548–2555, 2011.

[83] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct 2003.

[84] M. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 700–708. 2017.

[85] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3431–3440, 2015.

[86] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[87] S. M. Lowry, M. J. Milford, and G. F. Wyeth. Transforming morning to afternoon using linear regression techniques. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 3950–3955, 2014.

[88] S. M. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. D. Cox, P. I. Corke, and M. J. Milford. Visual place recognition: A survey. *IEEE Trans. Robotics*, 32(1):1–19, 2016.

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

[90] E. Malis and M. Vargas. Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, INRIA, 2007.

[91] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[92] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vision Comput.*, 22(10):761–767, 2004.

[93] C. McManus, W. Churchill, W. P. Maddern, A. D. Stewart, and P. Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 901–906, 2014.

[94] C. McManus, B. Upcroft, and P. Newman. Scene signatures: Localised and point-less features for localisation. In *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, 2014.

[95] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro. Image-based monte carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004.

[96] J.-A. Meyer and D. Filliat. Map-based navigation in mobile robots. II. A review of map-learning and path-planning strategies. *Cognitive Systems Research*, 4(4):283–317, 2003.

[97] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[98] M. Milford and R. Schulz. Principles of goal-directed spatial robot navigation in biomimetic models. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 369(1655), 2014.

[99] M. Milford and G. Wyeth. Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System. *IEEE Trans. Robotics*, 24(5):1038–1053, 2008.

[100] M. Milford and G. Wyeth. Persistent Navigation and Mapping using a Biologically Inspired SLAM System. *I. J. Robotics Res.*, 29(9):1131–1153, 2010.

[101] M. Milford, G. Wyeth, and D. Prasser. RatSLAM: a Hippocampal Model for Simultaneous Localization and Mapping. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*, pages 403–408, 2004.

[102] M. Milford and G. F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*, pages 1643–1649, 2012.

[103] R. Möller and A. Vardy. Local visual homing by matched-filter descent in image distances. *Biological Cybernetics*, 95(5):413–430, 2006.

[104] R. Möller, A. Vardy, S. Kreft, and S. Ruwisch. Visual homing in environments with anisotropic landmark distribution. *Auton. Robots*, 23(3):231–245, 2007.

[105] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada.*, pages 593–598, 2002.

[106] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 1151–1156, 2003.

[107] J. M. M. Montiel, J. Civera, and A. J. Davison. Unified inverse depth parametrization for monocular SLAM. In *Robotics: Science and Systems II, August 16-19, 2006. University of Pennsylvania, Philadelphia, Pennsylvania, USA*, 2006.

[108] R. Muller and J. Kubie. The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. 7:1951–68, 08 1987.

[109] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015.

[110] D. Murray and J. J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171, Apr 2000.

[111] T. Naseer, M. Ruhnke, C. Stachniss, L. Spinello, and W. Burgard. Robust visual SLAM across seasons. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, pages 2529–2535, 2015.

[112] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust visual robot localization across seasons using network flows. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 2564–2570, 2014.

[113] P. Neubert, N. Sünderhauf, and P. Protzel. Appearance change prediction for long-term navigation across seasons. In *2013 European Conference on Mobile Robots, Barcelona, Catalonia, Spain, September 25-27, 2013*, pages 198–203, 2013.

[114] P. Neubert, N. Sünderhauf, and P. Protzel. Superpixel-based appearance change prediction for long-term navigation across seasons. *Robotics and Autonomous Systems*, 69:15–27, 2015.

[115] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2320–2327, 2011.

[116] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.

[117] D. Nistér, O. Naroditsky, and J. R. Bergen. Visual odometry. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), 27 June - 2 July 2004, Washington, DC, USA*, pages 652–659, 2004.

[118] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Journal of Intelligent and Robotic Systems*, 61(1-4):287–299, 2011.

[119] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[120] J. O'Keefe and J. Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, 1971.

[121] E. Olson, J. J. Leonard, and S. J. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, May 15-19, 2006, Orlando, Florida, USA*, pages 2262–2269, 2006.

[122] T. J. Pennings. Do dogs know calculus? *College Mathematics Journal*, 34:178–182, 2003.

[123] E. Pepperell, P. I. Corke, and M. J. Milford. All-environment visual place recognition with SMART. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 1612–1618, 2014.

[124] A. Philippides, B. Baddeley, K. Cheng, and P. Graham. How might ants use panoramic views for route navigation? *Journal of Experimental Biology*, 214(3):445–451, 2011.

[125] A. Ranganathan, S. Matsumoto, and D. Ilstrup. Towards illumination invariance for visual localization. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 3791–3798, 2013.

[126] A. D. Redish. *Beyond the cognitive map: From place cells to episodic memory.* The MIT Press, 1999.

[127] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016.

[128] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6517–6525, 2017.

[129] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[130] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, pages 430–443, 2006.

[131] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. ORB: an efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2564–2571, 2011.

[132] E. Save, L. Nerad, and B. Poucet. Contribution of multiple sensory information to place field stability in hippocampal place cells, 2000.

[133] D. Scaramuzza and F. Fraundorfer. Visual Odometry : Part I: The First 30 Years and Fundamentals. *IEEE Robotics Automation Magazine*, 18(4):80–92, Dec 2011.

[134] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Trans. Robotics*, 24(5):1015–1026, 2008.

[135] G. Schindler, M. A. Brown, and R. Szeliski. City-scale location recognition. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[136] G. Sibley, C. Mei, I. D. Reid, and P. M. Newman. Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment. *I. J. Robotic Res.*, 29(8):958–980, 2010.

[137] R. Sim, P. Elinas, M. Griffin, and J. J. Little. Vision-based SLAM using the rao-blackwellised particle filter. In *In IJCAI Workshop Reasoning with Uncertainty in Robotics (RUR), Edinburgh, Scotland*, pages 9–16, 2005.

[138] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha. Tracking moving devices with the cricket location system. *Proceedings of the 2nd international conference on Mobile systems applications and services MobiSYS 04*, 1:190, 2004.

[139] R. Smith, M. Self, and P. Cheeseman. Autonomous robot vehicles. chapter Estimating Uncertain Spatial Relationships in Robotics, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[140] S. M. Smith and J. M. Brady. SUSAN - A new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[141] S. Song and M. Chandraker. Robust scale estimation in real-time monocular SFM for autonomous driving. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1566–1573, 2014.

[142] K. Souhila and A. Karim. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems*, 4(1):2, 2007.

[143] J. Stone and A. Bray. A learning rule for extracting spatio-temporal invariances. *Network: Computation in Neural Systems*, 6(3):429–436, 1995.

[144] T. Stone, M. Mangan, P. Ardin, and B. Webb. Sky segmentation with ultraviolet images can be used for navigation. In *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, 2014.

[145] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2352–2359, 2011.

[146] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-time monocular SLAM: why filter? In *IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010*, pages 2657–2664, 2010.

[147] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Scale drift-aware large scale monocular SLAM. In *Robotics: Science and Systems VI, Universidad de Zaragoza, Zaragoza, Spain, June 27-30, 2010*, 2010.

[148] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Visual SLAM: why filter? *Image Vision Comput.*, 30(2):65–77, 2012.

[149] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, pages 4297–4304, 2015.

[150] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems XI, Sapienza University of Rome, Rome, Italy, July 13-17, 2015*, 2015.

[151] H. Tanaka, Y. Sumi, and Y. Matsumoto. A high-accuracy visual marker based on a microlens array. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4192–4197, Oct 2012.

[152] H. Tanaka, Y. Sumi, and Y. Matsumoto. A solution to pose ambiguity of visual markers using moiré patterns. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3129–3134, Sept 2014.

[153] K. Tateno, F. Tombari, I. Laina, and N. Navab. CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6565–6574, 2017.

[154] J. Taube, R. Muller, and J. Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435, 1990.

[155] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[156] R. B. Tilove. Local obstacle avoidance for mobile robots based on the method of artificial potentials. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 566–571 vol.1, may 1990.

[157] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - A modern synthesis. In *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, held during ICCV '99, Corfu, Greece, September 21-22, 1999, Proceedings*, pages 298–372, 1999.

[158] C. Valgren and A. J. Lilienthal. SIFT, SURF and seasons: Long-term outdoor localization using local features. In *Proceedings of the 3rd European Conference on Mobile Robots, EMCR 2007, September 19-21, 2007, Freiburg, Germany*, 2007.

[159] C. Valgren and A. J. Lilienthal. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010.

[160] R. Wehner, B. Michel, and P. Antonsen. Visual navigation in insects: coupling of egocentric and geocentric information. *Journal of Experimental Biology*, 199(1):129–140, 1996.

[161] L. Wiskott. Learning invariance manifolds. *Neurocomputing*, 26-27:925–932, 1999.

[162] L. Wiskott and T. Sejnowski. Slow Feature Analysis: Unsupervised Learning of Invariances. *Neural Computation*, 14(4):715–770, 2002.

[163] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4353–4361, 2015.

[164] J. Zeil, M. I. Hofmann, and J. S. Chahl. Catchment areas of panoramic snapshots in outdoor scenes. *J. Opt. Soc. Am. A*, 20(3):450–469, Mar 2003.

[165] Z. Zhang, C. Forster, and D. Scaramuzza. Active exposure control for robust visual odometry in HDR environments. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3894–3901, 2017.

[166] Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):436–450, 2012.

[167] B. Zhou, À. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 487–495, 2014.

[168] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6612–6619, 2017.

[169] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 391–405, 2014.

[170] T. Zito, N. Wilbert, L. Wiskott, and P. Berkes. Modular toolkit for Data Processing (MDP): a Python data processing framework. *Front. Neuroinform.*, 2(8), 2009.