Visual Robot Navigation with Omnidirectional Vision

From Local and Map-Based Navigation to Semantic Navigation

DISSERTATION

submitted in partial fulfillment
of the requirements for the degree

Doktor Ingenieur
(Doctor of Engineering)

in the

Faculty of Electrical Engineering and Information Technology
TU Dortmund University

by

Luis Felipe Posada

Medellin, Colombia

Date of submission: 29th August 2018

First examiner: Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram
Second examiner: Univ.-Prof. Dr.-Ing Ralf Mikut

Date of approval: 1st October 2019

"*Dedicated to the memory of my father.*"

Acknowledgments

This thesis means the closing of the dream of studying in Germany that I built since I was a child. The stories I heard from my great-grandparents, German engineers who emigrated to Colombia in the 19th century to work in the gold mines of Titiribí, awakened in me great curiosity about Germany. But perhaps the strongest latent force that pushed me on this journey came from my father, Héctor Posada, an engineer at MIT and a pioneer in Latin America in computer science and telecommunications. Unfortunately, the violent conflict of the '90s in Colombia took his life, but his example, tenacity, and wit would remain within me and later would become my greatest inspiration to study abroad. In honor of him, I dedicate this thesis.

This work would not have been possible without Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram, who gave me the chance to work as a member of the RST department and inspired me with his leadership and professionalism. I am grateful to apl. Prof. Dr. rer. nat. Frank Hoffmann, who introduced me to new exciting topics and fueled me with ideas. A significant part of this thesis is due to his in-depth technical knowledge and advice he so kindly offered me. I extend my thanks to Prof. Dr.-Ing Ralf Mikut, who kindly agreed to review this work as the second examiner.

I am very grateful to all RST colleagues with whom I have worked. Especially Krishna Narayananan and Javier Oliva. Krishna accompanied me from day one, giving me support and feeding me with enriching discussions and suggestions. Javier became one of the most inspiring people I met in Germany. In addition to offering me a great friendship, he taught me how to reach millions of people through the Internet. I am also thankful to Thomas Nierobisch, who was my master thesis advisor and encouraged me to continue my doctoral studies at RST, to Jürgen Limhoff for helping us with the robotic platform and hardware, and to Mareike Leber and Gabriele Rebbe for their endless help the paperwork which made our work more comfortable. I extend my gratitude to other RST members, students, and friends who, unfortunately, due to space, I cannot mention, but offered me help and feedback.

I am very thankful to Colciencias for giving me financial support through the "Francisco José de Caldas" scholarship and the NRW state for a semester scholarship during the early days of this thesis.

I am forever grateful to my mother, Amalia, who was the person that supported me the most through the distance in all aspects during my stay in Germany; also, my siblings Soledad and Camilo for being always there for me, whenever I needed them. Finally, the last words of gratitude are for my partner Melissa for her love, unlimited support, patience, and life knowledge.

Abstract

In a world where service robots are increasingly becoming an inherent part of our lives, it has become essential to provide robots with superior perception capabilities and acute semantic knowledge of the environment. In recent years, the computer vision field has advanced immensely, providing rich information at a fraction of the cost. It has thereby become an essential part of many autonomous systems and the sensor of choice while tackling the most challenging perception problems. Nevertheless, it is still challenging for a robot to extract meaningful information from an image signal (a high dimensional, complex, and noisy data). This dissertation presents several contributions towards visual robot navigation relying solely on omnidirectional vision.

The first part of the thesis is devoted to robust free-space detection using omnidirectional images. By mimicking a range sensor, the free-space extraction in the omniview constitutes a fundamental block in our system, allowing for collision-free navigation, localization, and map-building. The uncertainty in the free-space classifications is handled with fuzzy preference structures, which explicitly expresses it in terms of preference, conflict, and ignorance. This way, we show it is possible to substantially reduce the classification error by rejecting queries associated with a strong degree of conflict and ignorance.

The motivation of using vision in contrast to classical proximity sensors becomes apparent after the incorporation of more semantic categories in the scene segmentation. We propose a multi-cue classifier able to distinguish between the classes: floor, vertical structures, and clutter. This result is further enhanced to extract the scene's spatial layout and surface reconstruction for a better spatial and context awareness. Our scheme corrects the problematic distortions induced by the hyperbolic mirror with a novel bird's eye formulation. The proposed framework is suitable for self-supervised learning from 3D point cloud data.

Place context is integrated into the system by training a place category classifier able to distinguish among the categories: room, corridor, doorway, and open space. Hand engineered features, as well as those learned from data representations, are considered with different ensemble systems.

The last part of the thesis is concerned with local and map-based navigation. Several visual local semantic behaviors are derived by fusing the semantic scene segmentation with the semantic place context. The advantage of the proposed local navigation is that the system can recover from conflicting errors while activating behaviors in the wrong context. Higher-level behaviors can also be achieved by compositions of the basic ones. Finally, we propose different visual map-based navigation alternatives that reproduce or achieve better results compared to classical proximity sensors, which include: map generation, particle filter localization, and semantic map building.

Contents

1 Introduction **1**
1.1 Experimental Platform . . . 2
1.2 Contributions and Outline . . . 2

2 Visual Range Sensor **6**
2.1 Introduction . . . 6
2.2 Related Work . . . 7
2.3 Free-Space Detection with Ensemble of Experts . . . 8
2.3.1 System Architecture . . . 8
2.3.2 Segmentation and Features . . . 10
2.3.3 Experimental Results . . . 15
2.4 Online Self-Supervised Free-Space with Range Image Data . . . 17
2.4.1 System Architecture . . . 17
2.4.2 Validation Data with a Range Camera . . . 18
2.4.3 Experimental Results . . . 20
2.5 Summary . . . 22

3 Uncertainty Representation with Fuzzy Preference Structures **23**
3.1 Introduction . . . 23
3.2 Related Work . . . 24
3.3 Fuzzy Preference Structures . . . 24
3.3.1 Preference Structure and Classification . . . 25
3.3.2 Valued Preferences for Ensembles of Segmentations . . . 27
3.3.3 Experimental Results: Classification of Indoor Scenarios . . . 27
3.4 Ensemble of Experts with Fuzzy Preference Structures . . . 31
3.4.1 Classification Results . . . 31
3.5 Summary . . . 35

4 Semantic Segmentation of Scenes **36**
4.1 Introduction . . . 36
4.2 Related Work . . . 37
4.3 Semantic Segmentation . . . 38
4.4 Features . . . 38
4.4.1 Location . . . 38
4.4.2 Color . . . 39

4.4.3 Local Intensity Distribution . . . 39
4.4.4 Shape . . . 40
4.4.5 Texture . . . 43
4.5 Random Forest Classification . . . 43
4.6 Semantic Segmentation Results . . . 44
4.7 Self-Supervised Semantic Segmentation with Range Images . . . 48
4.8 Summary . . . 48

5 Surface Layout of Scenes **49**
5.1 Introduction . . . 49
5.2 Related Work . . . 50
5.3 Surface Layout Approach . . . 51
5.3.1 Free-space and Vertical Structures . . . 51
5.3.2 Vanishing Points Estimation and Oriented Lines . . . 52
5.3.3 Floor-Wall Boundary Features . . . 55
5.3.4 Histogram of Oriented Gradients . . . 56
5.4 Experimental Results . . . 57
5.5 Summary . . . 58

6 Visual Place Category Recognition **60**
6.1 Introduction . . . 60
6.2 Related Work . . . 61
6.3 System Architecture . . . 62
6.4 Engineered Visual Features . . . 62
6.4.1 Gradient-Based Features . . . 63
6.4.2 Texture-Based Features . . . 64
6.4.3 Color-Based Features . . . 64
6.4.4 Shape-Based Features . . . 64
6.5 Learned Visual Features . . . 65
6.6 Classification . . . 66
6.6.1 Support Vector Machines . . . 67
6.7 Place Recognition Results . . . 70
6.8 Summary . . . 72

7 Behavior-Based Local Semantic Navigation **77**
7.1 Introduction . . . 77
7.2 Related Work . . . 78
7.3 System Architecture . . . 79
7.4 Semantic Segmentation . . . 80
7.5 Door Detection . . . 80
7.6 Navigation Behaviors . . . 81
7.6.1 Goal Point Homing . . . 81
7.6.2 Corridor Centering . . . 82
7.6.3 Obstacle Avoidance . . . 83
7.6.4 Door Passing Behavior . . . 85
7.6.5 High-Level Behaviors . . . 87
7.7 Summary . . . 87

8 Map-Based Navigation and Semantic Mapping **88**
8.1 Introduction . . . 88
8.2 Related Work . . . 89
8.3 Semantic Mapping Framework . . . 90
8.3.1 Free Space Segmentation . . . 90
8.3.2 Inverse Sensor Model . . . 92
8.3.3 Occupancy Grid Mapping . . . 92
8.3.4 Place Classification . . . 93
8.3.5 Semantic Mapping . . . 93
8.4 Semantic Mapping Results . . . 94
8.5 Monte Carlo Localization with Omnidirectional Vision . . . 95
8.6 Robot Localization Results . . . 99
8.7 Summary . . . 99

9 Conclusions and Outlook **101**

Bibliography **103**
Published Papers . . . 103
Contributed Papers . . . 104
References . . . 105

1
Introduction

Robots are migrating faster and faster from typical industrial applications to human environments to perform increasingly complex tasks. This transition requires higher perception capabilities and intelligence. Conventionally, mobile robots have operated using range sensors, which provide robust proximity information to nearby obstacles. However, in recent years, vision has emerged as a more versatile alternative as it provides more information, albeit more complex. This development is further supported by the increasing availability of powerful computer vision systems at affordable costs.

Inspired by the human vision, our most important sensor, we expect the robot of the future to have a deep semantic understanding of the world to dramatically improve human-robot natural communication, navigation, localization, and recognition of surrounding objects. Nevertheless, despite the recent advances, it is still challenging for a robot to extract meaningful information from the image signal (a high dimensional, complex, and noisy data), since:

- Navigation relies on depth information of the scene, which is not provided by a single 2D image.
- Raw image data is rather complex, which implies the need to segment and group objects to extract their relevant features and to aggregate those in a meaningful manner.
- Visual cues such as intensity, texture, and optical flow are highly context-dependent.
- Visual appearance of the environment and objects vary substantially with viewpoint and illumination.

This dissertation focuses on the visual navigation of mobile robots solely through omnidirectional vision. We investigate different aspects of visual navigation, providing the robot with the following capabilities: (i) deriving the free-space in the scene for collision-free navigation, (ii) handling uncertainty in the free-space classification for improving the robot's decision making, (iii) gaining semantic knowledge through classification of image regions, (iv) improving the spatial understanding through layout and structure recovery, (v) advancing place recognition for context-aware navigation, (vi) enhancing local semantic navigation with visual behaviors, and (vii) building a semantic mapping and achieving a purely visual map-based navigation.

Specifically, our work is concerned with omnidirectional-vision, which provides a wide field-of-view or panoramic snapshot of the environment. The advantage of

an omni-view (similar to a laser rangefinder), is that it captures the environment around 360° invariant under rotation. Additionally, the appearance variation between consecutive frames is small, given the low optical-flow due to the wide field-of-view which is beneficial for navigation, map building, and localization. These advantages come with the cost of low resolution, high distortion, and a blind-spot in the image due to the nature of the hyperbolic mirrors. These limitations can be overcome, as we shall see in the contributions derived in the following chapters.

1.1 Experimental Platform

The experimental platform used throughout this work consists of a Pioneer 3DX mobile robot shown in Fig. 1.1. The robot is equipped with a range camera and an omnidirectional camera.

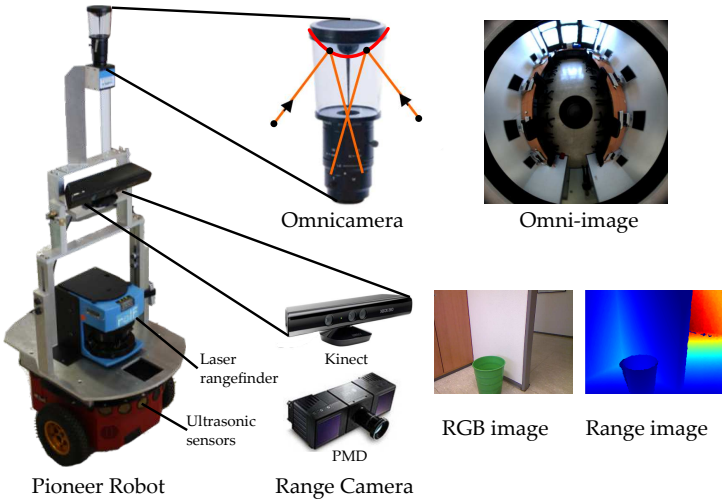


Figure 1.1: Robotic Platform and sensors.

In the experiments conducted in the first two chapters, we employ a Photo-mixer Device Camera (PMD) as the 3D camera. The PMD provides 3D measurements at a 204 x 204 pixel resolution across a 40° x 40° field of view. In the latter chapters, the 3D data is acquired with a Microsoft Kinect, which provides 320 x 240 depth data across a 57° x 43° field of view. The omnidirectional sensor combines a CCD camera with a hyperbolic mirror that has a vertical field of view of 75° that is directed towards the bottom to capture the floor. Fig 1.2 illustrates at the right an omnidirectional image captured on the scene at the left. The robot was positioned on the corridor and objects on both images are labeled to illustrate how the hyperbolic mirror distorts the image, but is able to capture a 360° wide field of view. The robot is also outfitted with a SICK LMS-200 laser rangefinder and 16 Polaroid ultrasonic range sensors. The range data acquired with these proximity sensors are mainly used for validation.

1.2 Contributions and Outline

The organization of the thesis is outlined in Fig 1.3. The first chapters deal with estimating the free-space in omnidirectional images. These chapters constitute a sig-

Figure 1.2: Experimental platform positioned on a scene where the omnidirectional image at the right was captured. The scene and the omnidirectional image are labeled with the same objects to show the wide field of view image formation and distortion introduced by the hyperbolic mirror

nificant building block towards performing a purely visual local navigation and map building without relying on range from proximity sensors. From Chapter 4 onwards, the system incorporates more and more semantic understanding to take full advantage of the vision-based systems necessary to achieve more complex robotics tasks. The outline of the thesis is as follows:

Chapter 2 introduces two novel free-space segmentation approaches for omnidirectional images that are pivotal for local navigation and map building in the last chapters. The first scheme rests upon the fusion of multiple classifications generated from heterogeneous segmentation schemes using a mixture of experts approach. The second scheme employs an online self-supervised scheme that selects the traversable floor region using the optimal segmentation to the appearance of the local environment by cross-validation over 3D scans captured by a 3D range camera.

Chapter 3 investigates the uncertainty inherent in a single free-space segmentation as well as the global uncertainty across multiple segmentation based classifiers. The uncertainty of the classifications is explicitly expressed in terms of preference, conflict, and ignorance utilizing fuzzy preference structures, thereby mapping the problem of classification into the domain of decision-making under fuzzy preferences. We show that the classification error is substantially reduced by rejecting those queries associated with a strong degree of conflict and ignorance.

Chapter 4 extends the binary classifier from Chapter 2, increasing the number of semantic labels for each classified image's region. The labeling distinguishes between the main classes: floor, vertical structures, and clutter. Vertical planes are objects such as walls, boards, and placards. Non-planar objects in the scene are grouped into the class clutter and are objects with multiple surfaces such as furniture, plants, people, or obstacles. We show that the method is suitable for self-supervised learning using the 3D point cloud data of a range camera.

Chapter 5 extends further the semantic segmentation by extracting the spatial layout and surface reconstruction of the omnidirectional image. Knowing the spatial layout of the scene improves scene understanding and enhances object detection. The

proposed system benefits from a cascaded classification, where all vertical structures from the semantic segmentation are analyzed to find their main orientation by fusing orientation features from the HOG transform, floor/wall boundaries, and oriented lines in the image.

Chapter 6 presents a place-category recognition system that distinguishes among the location categories: room, corridor, doorway, and open space. The approach evaluates hand-engineered features compared to learned-from-data representations. Different combinations of heterogeneous features are tested with non-learning ensembles and ensembles that learn the model with supervised learning. The developed place recognition system is a fundamental block for the local semantic navigation and the semantic map building in Chapters 7 and 8.

Chapter 7 introduces a local semantic navigation approach based on visual behaviors solely relying on omnidirectional vision. Four basic behaviors are presented: goal point homing, corridor centering, obstacle avoidance, and door passing. Places in the behavior are described with a semantic category (e.g. room, corridor), while regions in the scene are also labeled with semantic regions (e.g. door, wall, floor). The proposed local navigation has the advantage that it can recover from conflicts, such as activating behaviors in the wrong context. The chapter also shows how to derive high-level behaviors by compositions of basic behaviors.

Chapter 8 deals with two aspects of the map-based navigation solely using omnidirectional vision: (i) map localization, and (ii) semantic map building. The map representation consists of an occupancy grid constructed by a sensor model and its inverse from the segmented local robot's free-space. The sensor model corrects the non-linear distortions of the omnidirectional mirror and outputs a scaled perspective image of the ground plane using a bird's eye mapping. The free-space bird's eye view images constitute the perceptual basis for both the mapping and the localization. Robot localization is achieved with the Monte Carlo method and the semantic map building uses the place category classifier from Chapter 6 to label categories: room, corridor, doorway, and open room. Each place class maintains a separate grid map that is fused with the range-based occupancy grid for building a dense semantic map.

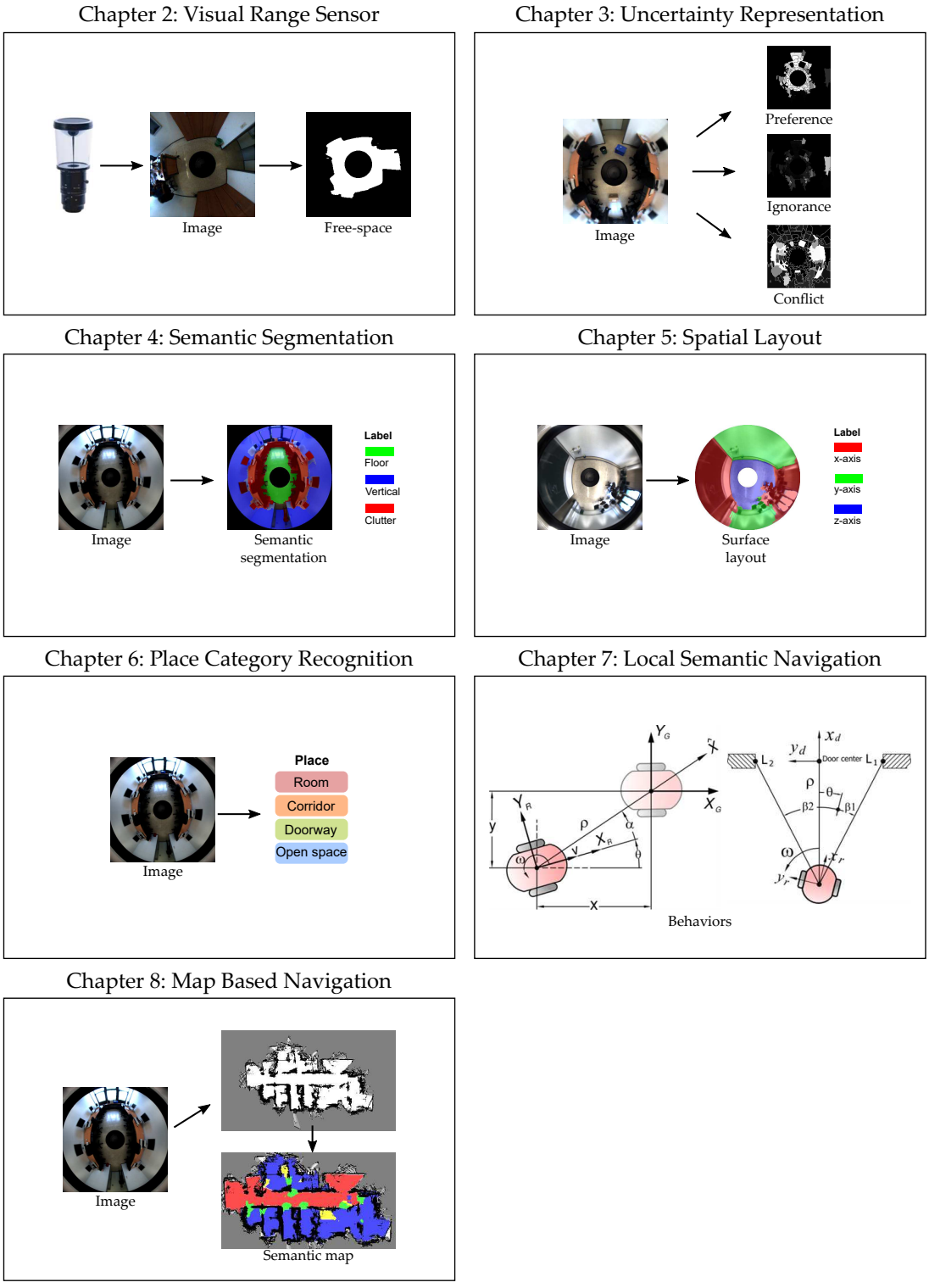


Figure 1.3: Organization of the thesis

2
Visual Range Sensor

2.1 Introduction

Navigation is one of the fundamental skills a mobile robot should master to perform its task autonomously. Robot navigation has conventionally relied on range sensors that provide robust proximity information to nearby obstacles. However, in recent years, vision has emerged as an alternative, providing more information albeit more complex.

Vision, in contrast to range sensors, enables the appearance-based distinction of objects, making it possible to differentiate, for example, ground floor, obstacles, people, walls, corridors, and doors. Notwithstanding, in spite of a large amount of information in an image, pure autonomous visual navigation remains a challenge given its noisy, high-dimensional, and complex data in its signal.

Free-space detection constitutes a significant building block for vision-based robotics as it supports high-level tasks such as localization, navigation, and map building. In the context of mapless navigation, the segmentation of the free space in the local environment replaces or augments proximity sensors and supports a direct mapping from visual cues onto actions characteristic for reactive behaviors like in [Nar+09].

In this chapter, we address the problem of extracting free-space by segmenting the floor obstacle regions in omnidirectional images using two approaches. The first technique rests upon the fusion of multiple classifications generated from heterogeneous segmentation schemes. Each segmentation relies on different features and cues to determine a pixel's class label. The second approach extracts the traversable floor region in the omnidirectional robocentric view by selecting the segmentation optimal to the appearance of the local environment by cross-validation over 3D scans captured by 3D range camera.

While the literature reports several robust floor-obstacle segmentation methods, the novelty of our method lies in the following ideas: Use of the mixture of experts approach [Pol06]. The key idea is to combine multiple segmentations in a way that the overall decision making benefits from the mutual competences of heterogeneous individual classifiers. The analysis confirms that the combination of diverse classifiers achieves a classification rate that far exceeds the performance of any single classifier. Additionally, our approach differs in several important aspects, namely: a) 3D range camera instead of a laser scanner; b) segmentation of the range image in addition to the 2D image; c) multiple representations of the appearance-based floor model; d)

various segmentation algorithms rather than one; e) obstacle and background modeling.

2.2 Related Work

Terrain traversability is a topic already studied by many research teams and has been surveyed recently by Papadakis [Pap13]. The techniques covered are mainly from outdoor robotics, ranging from lidar, stereo, to color based ones. Indoor robotics methods employ similar principles and are addressed under the name free-space detection.

Early free-space detection approaches [LBG97; UN00] employed color histograms to model the appearance of the floor and non-floor objects. Obstacles are classified as regions that differed significantly in appearance to the bottom part of the image (assumed to be the floor). The method in [UN00] improved [LBG97] by implementing an adaptive scheme that continuously updates the floor model based on the recently traversed terrain. The drawback of these histogram-based systems is that the robot is not able to continue navigation if the appearance of the floor changes abruptly. For example, with markers or carpets on an otherwise homogeneous floor. The fast approach in [LV03] also relies on color histograms, thus making it sensitive to ambiguous texture or color.

Martin [Mar06] designs the vision system through evolutionary algorithms to estimate the depth of free space along with different directions, thus mimicking a conventional proximity sensor. Ground patches from planar homographies estimated from corners tracked across multiple views were proposed by [PL01] while [KK04] segments the ground plane by calculating plane normals from motion fields. The segmentation in [Bla+08] rests upon a two-stage K-means clustering using local color as well as texture descriptors. Plagemann *et al.* [Pla+08] employ Gaussian process models of either edge-features or principal components of the image.

Self-supervised detection of traversable terrain gained increasing attention in outdoor robotics. The so-called near-to-far online learning [Dah+06; Gru+07; Kim+06] extracts the ground truth about local flat drivable areas from laser range sensors. A model is generated from the ground truth data to classify remote traversable regions in off-road terrain beyond the short-range training region in front of the vehicle.

More recently, Suger *et al.* [SSB15] developed a system able to project into a 2D occupancy grid the analysis of a 3D-lidar sensor. Delmerico *et al.* [Del+16] present a method that combines an aerial and ground robot for an almost online, self-supervised system. The robot on-air collects data to fully train a small convolutional neural network for terrain classification used by the ground robot. The training is achieved after only one minute of flight.

While the literature reports many robust free-space segmentation methods, very few are designed for omnidirectional vision. The novelty of our method lies in the mixture of experts approach to combine multiple segmentations in a way that the overall decision making benefits from the mutual competences of heterogeneous individual classifiers

2.3 Free-Space Detection with Ensemble of Experts

In this section, we introduce the first approach for free-space segmentation, which fuses multiple classifications generated from heterogeneous segmentation schemes. We follow the mixture of experts approach [Pol06; Pos+10a; Pos+11b] with different segmentations and cues to determine a pixel's class label in omnidirectional images.

2.3.1 System Architecture

The overall system architecture is shown in Fig. 2.1. The upper part outlines the ground truth data generation for classifier training. The lower part depicts the features, segmentation schemes and the ensemble of classifiers. The ground truth labeled data (Fig. 2.2 is generated using a PMD camera that delivers 3D range data at a 204 x 204 pixel resolution across a $40^\circ \times 40^\circ$ field of view. The PMD camera information only generates the ground truth for offline training and is not used at run time.

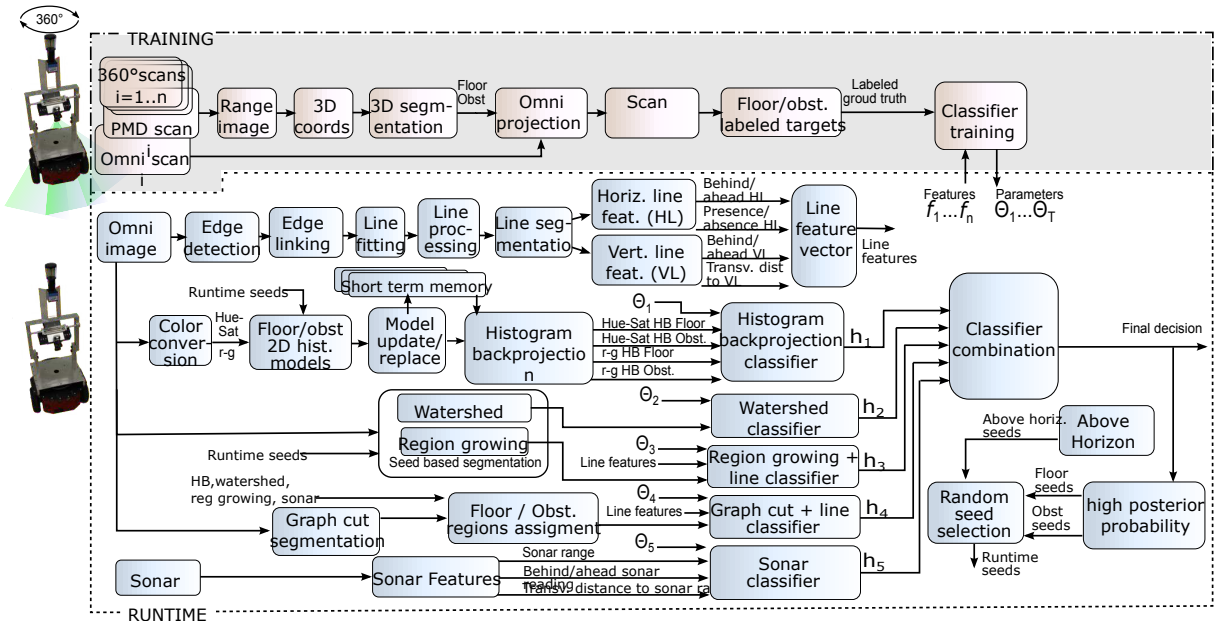


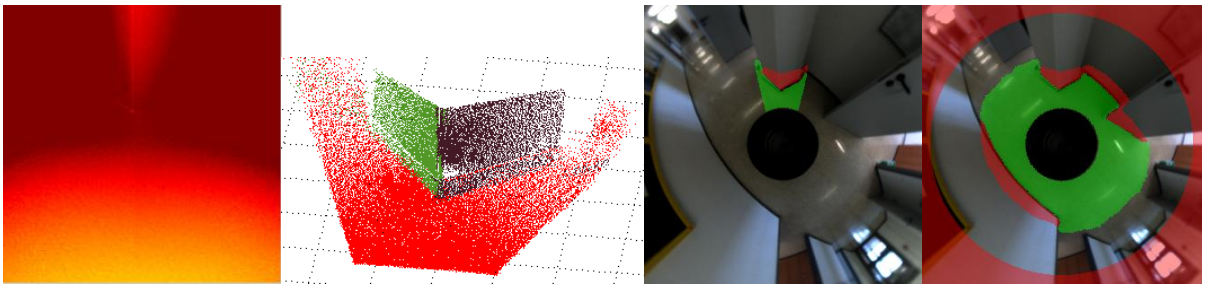
Figure 2.1: System architecture of the free-space detection with ensemble of experts

The depth image of different scenes, such as corridors, open rooms, and confined spaces, is obtained by rotating the robot and PMD camera by 360° and capturing scans at each pose. The 3D data from the PMD is subsequently fitted to planar surfaces using random sampling consensus (RANSAC). Surface normal orientation, distance to the camera, and connected components determine whether a pixel and its associated 3D point are labeled as obstacles or floor.

The target floor-obstacle labels for each omni-image are generated after projecting the 3D labeled points into the omni-view. Furthermore, a complete 360° ground truth mask is generated by the superposition of overlapping scans (10° overlaps) from odometry and scan matching.

Fig. 2.2a-d illustrates the target data classification in the range image. Brightness indicates proximity in the range image. The estimated planes in 3D space are each marked with a different color in the point cloud. The 3D plane segmentations are

projected onto the corresponding sector in the omnidirectional view. The last step is the generation of the complete 360° ground truth mask employing scan matching. At runtime, the sole robotic reactive behavior perceptions are the classification of the omnidirectional image into free space and obstacles.



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

Figure 2.2: Ground truth data generation (a) Range image, (b) Planes fitted from the 3D data with RANSAC, (c) Projection into the omnidirectional view, (d) Ground truth data after 360° scan-matching and considering points over the horizon

Four segmentation schemes generate heterogeneous segmentations: 2D histogram models; marker-based watershed and region growing that rely on seeds and the graph-based segmentation method an unsupervised approach. Further, the robot's built-in sonar sensors provide depth information to the system with low angular resolution. Vertical and horizontal lines features that are strong obstacle indicators contribute further to enhance the region growing and graph-based segmentation by fusing their segmentation with geometric features. Only two segmentations are supported with line features to obtain as heterogeneous as possible features and classifiers. This diversity ensures that the classifiers make different errors at specific instances. This heterogeneity is a fundamental property of ensemble-based classification to reduce the generalization error compared to the individual classifier error rates.

Each classification expert constitutes a naive Bayes classifier that predicts the posterior probability of the class of a single-pixel as either floor or obstacle based upon the features and their likelihoods computed on the training set. The naive Bayes classifier, although it assumes conditional independence of features, achieves reasonable accurate classification at low computational complexity in comparison to more advanced schemes such as Bayesian networks. Notice that for reactive mobile robot navigation, a frame rate of 10Hz is desirable, thus requiring fast real-time image processing and classification.

For combining the expert evidence, we investigate four different combination methods [Pol06]: The stacked generalization approach, where the outputs of individual classifiers serve as inputs to a second level meta-classifier; the behavior knowledge space (BKS) that builds a lookup table that maps the possible combinations of multiple classifier decisions with the observed class frequencies in the training data and finally two non-trainable aggregation rules based on the median and the majority voting. Since the PMD data is not available at runtime, the seed-based segmentation algorithms obtain their seeds from pixels in the previous frame with high-class

confidence. Extracting seeds from the previous frame is reasonable, given that the optical flow between consecutive frames is low due to the wide field of view of the omni-camera and the high frame rate.

2.3.2 Segmentation and Features

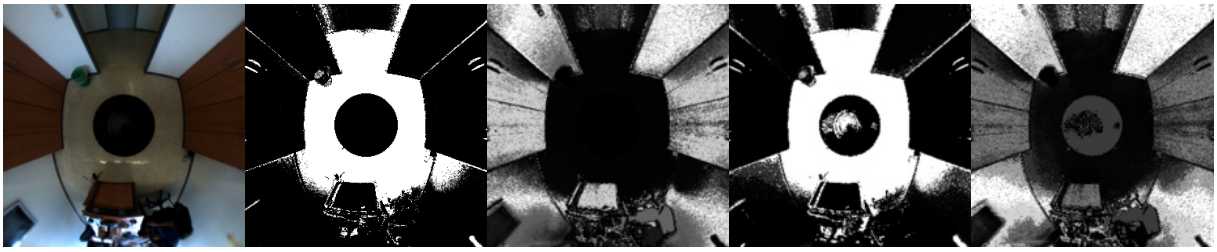
Histogram Backprojection

Histogram Backprojection [SB91] identifies regions in the image which match with a reference model in appearance. The normalized histogram reference model M in a 2D color space is compared with the normalized histogram H of the current frame I and the backprojection image B is computed by

$$B(u,v) = min\left(\frac{M(c(u,v))}{H(c(u,v))}, 1\right), \tag{2.3.1}$$

in which $c(u,v)$ denotes the 2D color of the pixel (u,v). The backprojection image in the range between $[0,1]$ is interpreted as the degree of similarity between the region and the reference model.

Instead of merely generating a single color model of the floor, our approach maintains additional models of the obstacles. Maintaining explicit appearance models of the obstacle regions refines the decision boundary between the floor and obstacles compared to a floor versus rest comparison. The omnidirectional view extends about 15° beyond the horizon line such that pixels above the horizon line certainly do not belong to the floor. Thus, the obstacle model is estimated from pixels above the corresponding horizon circle in the omni-image. The 2D histograms are quantized into 32-32 bins to improve generalization, reduce the dimensionality, and are averaged and normalized A naive Bayes classifier based on histogram backprojection merges the information of the following four input features: The floor and obstacle model backprojection use the Hue-Saturation channel of the HSV color space; accordingly, the floor and obstacle model backprojection of the r-g channel of the RGB (normalized RGB space). Examples of the output of the histogram backprojection segmentation are shown in Fig. 2.3



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

Figure 2.3: (a) Input image (b) Hue-Saturation histogram backprojection with floor model (c) Hue-Saturation histogram backprojection with obstacle model (d) R-g histogram backprojection with floor model (e) R-g histogram backprojection with obstacle model

Dynamic changes in slowly varying environments are addressed by updating the histograms with an exponential moving average (EMA) whereas in rapid changes in

the scene, e.g., changes in the floor surface, a new model is instantiated rather than updating. This situation is identified with histogram-intersection, where a threshold value indicates no matches with the stored histograms. The maximum number of models stored simultaneously in short term memory is limited to five instances of obstacle models and two for the floor to meet computational demands. The EMA is computed with the following equation:

$$\bar{M}(x,y)_t = \alpha M(x,y)_t + (1-\alpha)\bar{M}(x,y)_{t-1} \tag{2.3.2}$$

in which M_t denotes the histogram of the current frame and $\bar{M}_{t-1}$ the previous average. The smoothing factor α is set to 0.6 giving more importance to recent observations. The histogram intersection is given by:

$$d(M_1, M_2) = \sum_{x,y} \min(M_1(x,y), M_2(x,y)) \tag{2.3.3}$$

in which a score of one denotes an exact match and zero a complete mismatch. In practice, a new model is initialized if the histogram intersection between the current frames and the most similar stored models falls below a threshold of 0.4. New histogram models replace the oldest model in case the maximum storage capacity is exceeded. Models that do not match with any of the thirty most recent frames are automatically discarded.

Marker-based Watershed

Marker-based watershed segmentation [RM01] is best understood by interpreting the gradient image as elevation information. The watershed lines coincide with strong image gradients. Pixels in the elevation image are attracted to their local minimum, thus forming so-called basins. Water is flooded evenly from each marker to flood the basins. The process stops once the water level reaches the highest peak in the landscape. Basins connected to the same original marker are merged into a single region. Fig.2.4 shows an example of watershed segmentation seeded with a random percentage of high confidence labeled seeds.

(a) (b) (c)

Figure 2.4: Watershed segmentation: (a) Input image (b) Marker-based watershed with random seeds (c) Resulting watershed segmentation knowing the floor/obstacle seeds selected at points with high posterior probability

Graph-based Segmentation

Graph-based segmentation is described in [FH04]. This method is highly efficient and produces segmentations based on a region comparison function that considers the minimum weight edge in a graph between two regions in measuring the difference between them. The graph is constructed in such a way that image pixels are transformed into a feature space vector that combines the (x, y) pixel location with its RGB color value. Edges in the graph connect points that are close in the feature space. The assignment of whether or not a region belongs to the ground or obstacles, is determined with information from previous segmentations.

Given the unsupervised nature of this algorithm, it is common to find situations where the segmentation grows unbounded. It is, therefore worthwhile to fuse the graph-based segmentation in the classifier with the line features. Fig. 2.5 illustrates the graph-based segmentation, the floor-obstacle assignment and the benefit of using line features to prevent unbounded region growing.

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

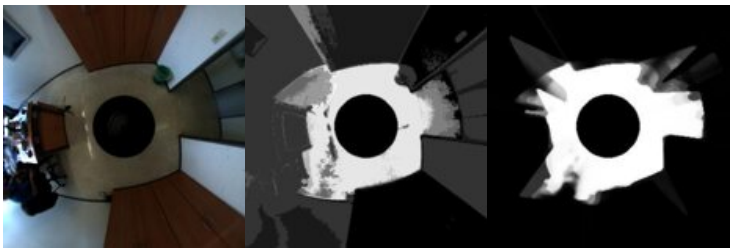
Figure 2.5: Graph-based segmentation: (a) Input image (b) Graph-based segmentation (c) Floor-obstacle assignment to the graph-based regions from the previous segmentations; d) Output of the classifier after merging the line features

Region Growing

Region growing starts with a set of seed points, and neighboring pixels are added based on their similarity in an incremental fashion to a region. Each segmentation originates from a single seed of known class labels. Region growing segmentation is repeated N_i times with different seeds randomly sampled from pixels classified in the previous image with high confidence. After N_i random region growing repetitions, $N_i(u, v)$ counts how often the pixel (u, v) is part of the final region, thus indicating the similarity of that pixel with prototype pixels of class λ_i. It is important to notice that a pixel that is not captured by a particular region growing instantiation might still belong to the corresponding class, as region growing is a local method and only expands to those pixels that are connected with the original seed. In particular, the obstacle regions are often fragmented due to the multitude of objects. Therefore, it is essential to relate the absolute counts to the average number of pixels $< M_{ki} >$ in a single region growing segmentation. A pixel has a membership degree to class λ_i given by

$$\mu_i(u, v) = \max\{\frac{N_i(u, v)}{N_i} \frac{\cap M_i}{\overline{M_i}}, 1\}. \tag{2.3.4}$$

The relative frequency $N_i(u,v)/N_i$ is scaled by the ratio between the size of the union of all segmentations $\cap M_i$ of class λ_i and the average size $\overline{M_i}$ of a single segmented region. The ratio is a coarse estimate of the number of fragmented segments of a class. Therefore, the score of a fragmented class is weighted more strongly in comparison to a class that forms a connected region.

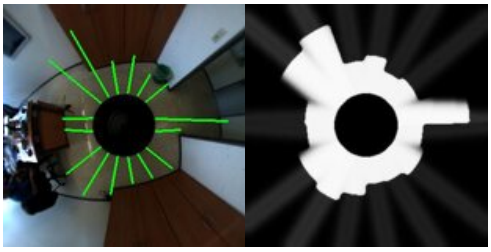


(a) (b) (c)

Figure 2.6: Region growing segmentation: (a)Input image (b) Region growing (c) Classifier output of region growing plus line features

Inverse Sonar Model

The sonar classifier computes the posterior probability of a pixel belonging to a class floor, obstacle based on the observed likelihoods $p(C_i|r,d,a)$. The feature r corresponds to the discretized sonar range reading and is limited to ten discrete values. d is the lateral distance of a pixel to the nearest sonar ray projected into the omni image, and a denotes whether a pixel in the radial direction is in front or behind the nearest sonar range reading. An example of the sonar classifier output is shown in Fig. 2.7. Notice that sonar range readings might be misleading, such as the two range readings in the north-west direction that overestimate the extension of free space.

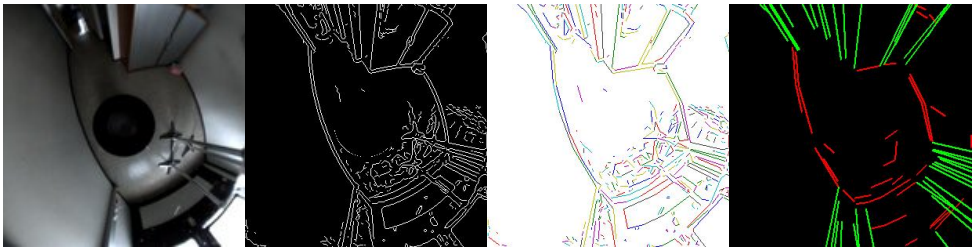


(a) (b)

Figure 2.7: Sonar classifier: (a) Input image, with projected sonar readings (b) Output of the sonar classifier

Segmentation Based on Lines

Edges often correspond to image regions at which the scene exhibits discontinuities in shape, depth, or variation in material properties. In our context, they suffer from the disadvantage of high sensitivity to variations in the scene illumination. Nevertheless,



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

Figure 2.8: Line features: (a) Input image (b) Canny edge detection (c) Edge linking and line fitting (d) Vertical/horizontal lines

edges are strong indicators of the presence of obstacles and provide a valuable cue to enhance the classification of region growing and the graph-based segmentation.

Rather than using merely the edge information at a pixel level, their connectivity is considered. Edges are linked by contour following and subsequently fitted to lines using a split approach. Contours are first globally approximated by a straight line connecting the endpoints. Consequently, they are recursively split at the points of maximum traversal deviation. The resulting lines are classified into vertical and horizontal lines according to their orientation in the omniview. Lines not belonging to either orientation or lines that are too short are discarded. Vertical lines appear radially distributed in the omni-image and in general correspond to typical indoor environment objects such as shelves, cabinets, cupboards or table legs. A horizontal line may indicate the presence of obstacles such as walls, doors or tables.

Four line features are defined accordingly: a) ahead or below a horizontal line; b) Presence or absence of horizontal lines; this is determined for each pixel u, v by verifying if there is an intersection of a horizontal line with the projected line formed by the pixel u, v and the image center; c) ahead or below a vertical line; d) transversal distance to the closest vertical line.

Naive Bayes Classification

The ensemble of experts is constituted by five Naive Bayes classifiers as shown in Fig. 2.1. The likelihood $p(x_i|C_j)$ of a feature x_i given the class C_j is modeled by a Gaussian distribution in the case of continuous features, and by a multinomial distribution in the case of discrete features. The likelihoods of the data and the class priors are estimated from the observed frequencies of classes and features in the training data. The naive Bayes classifier computes the a posteriori probability of the classes $C = \{Floor, Obstacle\}$ according to the likelihood of the conditionally independent features:

$$p(C_j|\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{n} p(x_i|C_j)p(C_j) \tag{2.3.5}$$

in which the normalization factor Z is the evidence of the features $\mathbf{x}$. The ultimate decision boundary is determined by the problem-specific relative costs of false positive and false negative classifications.

2.3.3 Experimental Results

The training data consists of almost two million pixels captured from 500 images of which the true class label is established from the PMD depth information and 3D segmentation. The classification performance is validated on 500 additional unseen images with ground truth obtained from PMD data. The training in the case of stacked generalization follows a k-fold selection in which the training data set is divided into T blocks, in which T denotes the number of classifiers participating in the ensemble. Each single classifier is trained with $T-1$ blocks of data; thereby, each classifier does not see one block of data. The outputs of the classifiers on the unseen block, in conjunction with the ground truth segmentation, constitute the training pairs for the second level meta-classifier. Once the second level classifier is trained, the T classifiers are retrained with the entire training data set. The segmentation accuracy is determined based on the false positive rate FPR, the number of false positives divided by the total number of negatives; and true positive rate TPR, the number of true positives divided by the total number of positives. A false negative is constituted by a floor pixel incorrectly classified as an obstacle, and a false positive is an obstacle pixel incorrectly labeled as floor.

Table 2.1: Classifiers performance

Method	FPR	TPR
Histogram backprojection	0.085	0.889
Watershed	0.042	0.945
Watershed + line features	0.036	0.938
Region growing	0.120	0.718
Region growing + line features	0.044	0.907
Graph segmentation	0.064	0.869
Graph segmentation + line features	0.023	0.917
Sonar	0.045	0.890
Combination with stacked generalization	0.032	0.958
Combination with BKS	0.030	0.912
Combination with median	0.024	0.945
Combination with majority voting	0.024	0.927

Table 2.1 compares the true positive and false negative rates of the output of every single classifier, namely histogram backprojection, watershed, region growing, graph segmentation and sonar. The fusion of vertical and horizontal lines features in region growing and graph segmentation clearly improves the performance of these two classifiers. Using line cues in conjunction with watershed does not improve its accuracy, as watershed itself is already based on edge features. Notice, that watershed and graph cut in isolation are unsupervised segmentation schemes not suitable for classification. They instead inherit the classification from the proper labeling of seeds, which in this case, are provided by stacked generalization. Therefore, their reported accuracy in Table 2.1 has to be attributed to the ensemble of classifiers rather than the watershed and graph-based segmentation itself.

Stacked generalization exhibits the best classification rate in terms of FPR among all classifier ensemble methods. The inferior performance of BKS combination is attributed to the binarization of the posterior probabilities for the generation of the lookup table that results in loss of information that is available to stacked generalization. Even though median and majority voting combination rules require no second-level training, they demonstrate the best performance in terms of FPR. In the context of obstacle avoidance, false-positive classification is more critical since missing an obstacle is potentially more severe than underestimation of the free space. One of the main properties of the median combination is to filter out segmentations that are outliers.

Fig. 2.9 shows the results of four prototypical scenarios, with the corresponding output of the four combination methods. The first scene exhibits misleading specular reflections and shadows that are incorrectly segmented based on color information alone. The vertical line features increase the segmentation performance since the three vertical posts in the bottom right part of the image are correctly segmented from the floor, even though some of the first level classifiers ignore them. The second scenario illustrates a typical office environment with a rather high contrast between floor and obstacles. The specular reflections on the surface are still correctly classified as floor. The third to fifth scenarios illustrate segmentation tasks that exhibit substantial variations in illumination, reflections, and shadows.

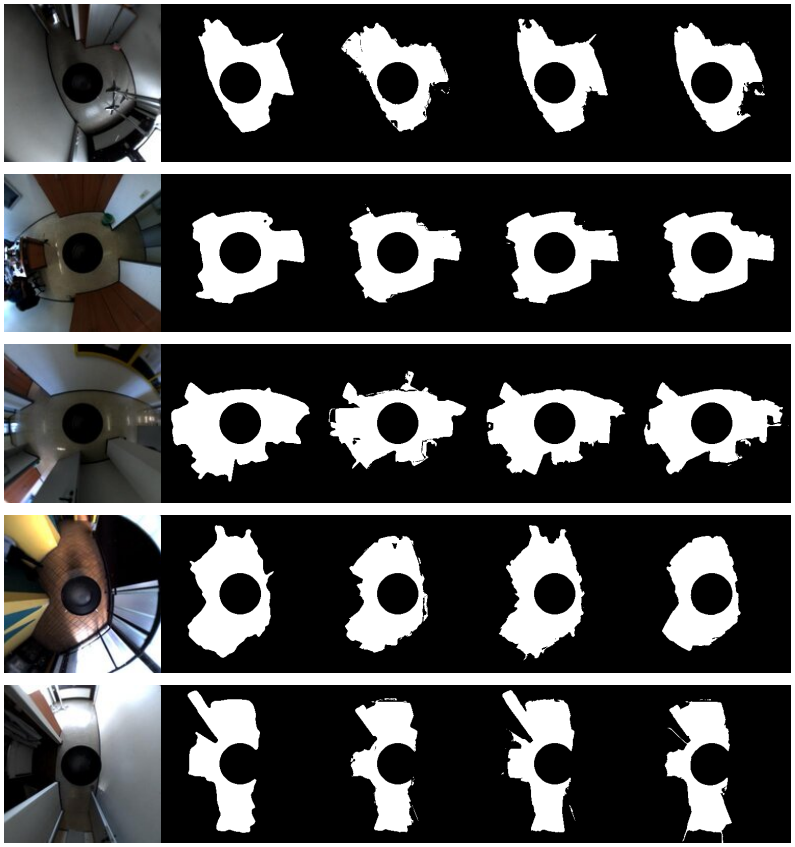


Figure 2.9: Segmentation examples: (a) Input image, (b) Stacked Generalization, (c) Behavior Knowledge Space, (d) Median combining rule, (e) Majority voting.

2.4 Online Self-Supervised Free-Space with Range Image Data

In the previous section, we discussed a segmentation approach based on supervised learning from offline data by fusing with a mixture of experts different heterogeneous segmentations and features. In this section, we explore a self-supervision method [Pos+10b] by online combining 3D information of the Range image to gather more insight into the visual data in the omnidirectional image.

The approach employs a 3D camera to obtain ground truth segmentations in a narrow frontal field of view. The accuracy of alternative segmentation schemes in conjunction with alternative visual cues is evaluated by cross-validation over 3D scans on the frontal view. The combination that performs best in the current context is then applied to segment the omnidirectional view providing 360° depth information.

The range data in the front view provides the seeds and validation data to supervise the appearance-based segmentation in the omniview. The Segmentation relies on histogram backprojection, which maintains separate appearance models for floor, obstacles, and background.

2.4.1 System Architecture

The camera system consisting of a PMD and an Omnidirectional camera is mounted on the Pioneer 3DX mobile robot, as shown in Fig. 2.10. The 40° x 40° field of view of the time-of-flight (ToF) camera and the 75° vertical of the omnidirectional are directed towards the floor and both camera's fields of view intersect in a narrow region in front of the robot.

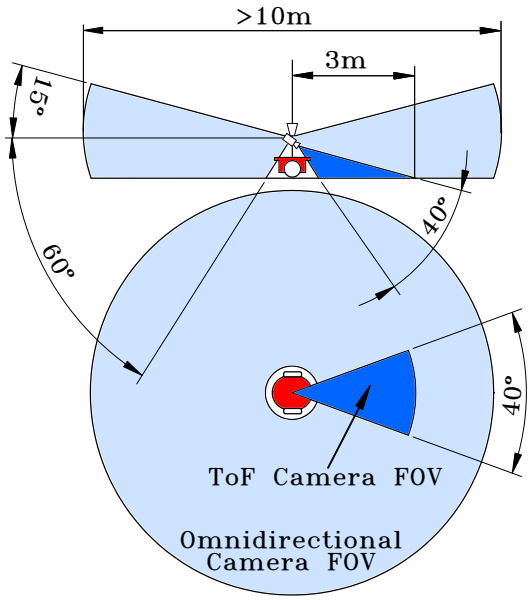


Figure 2.10: Configuration and field of view of the PMD camera and the omnidirectional camera

The overall system architecture is shown in Fig. 2.11. The online segmentation proceeds in four stages: a) range image segmentation into planar regions; b) projection of

these regions into the omnidirectional view; c) omniview segmentation; d) validation and selection of optimal segmentation and cue.

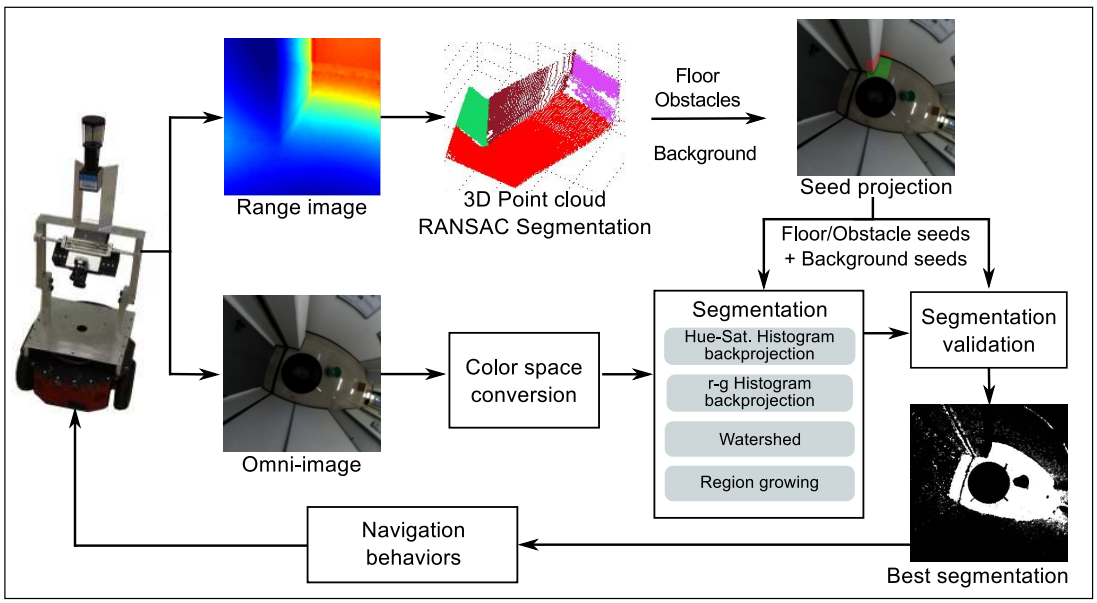


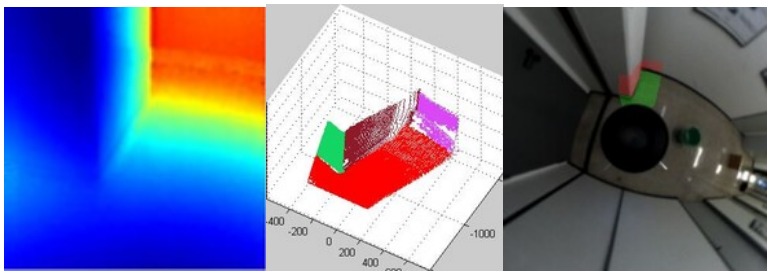
Figure 2.11: System architecture of the online self-supervised free-space detector

2.4.2 Validation Data with a Range Camera

The validation data is generated similar to the technique used in the previous section to obtain the 360° training mask. A 3D scan of the frontal region is segmented into planar surfaces using RANSAC. These segments are then classified according to the orientation of the surface normal, distance to the camera and area into three categories: ground considered as free space, vertical planes and obstacles. Scan points that do not belong to a surface are ignored. These labeled regions are projected as ground truth segmentation into the omnidirectional view. This ground truth allows the selection of the visual cue and segmentation method that is optimal for the appearance and illumination of the local environment. Part of the 3D scan segmentation is utilized as seeds for the subsequent 2D appearance-based segmentation; the remaining data is used to validate the performance of alternative segmentations and cues.

RANSAC is applied iteratively, such that the scan points that belong to the plane with the most inliers are removed from the data set. The next plane is estimated from the residual points until the number of points belonging to the best surface model falls below a threshold. The classified 3D points are projected into the omnidirectional view as shown in Fig. 2.12. It illustrates the range image in which brightness indicates proximity, the estimated and labeled planes in 3D space, and the projection of these labels onto the corresponding sector in the omnidirectional view.

A key aspect of marker or seed-based segmentation algorithms is the selection of initial seeds. The more informative the set of seeds, the better the final segmentation result. Similar to the previous Section, in the online segmentation, we employ four alternatives: segmentation based on color histogram backprojection in two different color spaces (hue-saturation and normalized red-gree), watershed algorithm,



(a) (b) (c)

Figure 2.12: Seeds projection: (a) Range image, (b) Planes fitted from the 3D data with RANSAC, (c) Seeds projected into the omnidirectional view.

and model-based region growing. Although we acknowledge the potential utility of other appearance features such as texture [Bla+08], only color is considered as it is difficult to extract reliable texture due to the low resolution of the omni-image.

No single segmentation scheme alone provides a robust, accurate segmentation across all scenes. The idea is, therefore, to validate the accuracy of segmentation on labeled ground truth data and then to select the method best suited for the current floor texture and illumination. For this purpose, the ground truth segmentation data obtained from the 3D scans is partitioned into training and validation data (Fig 2.13). The training data is used to build the reference model for the histogram backprojection and to provide the initial seeds for marker-based watershed and region growing.

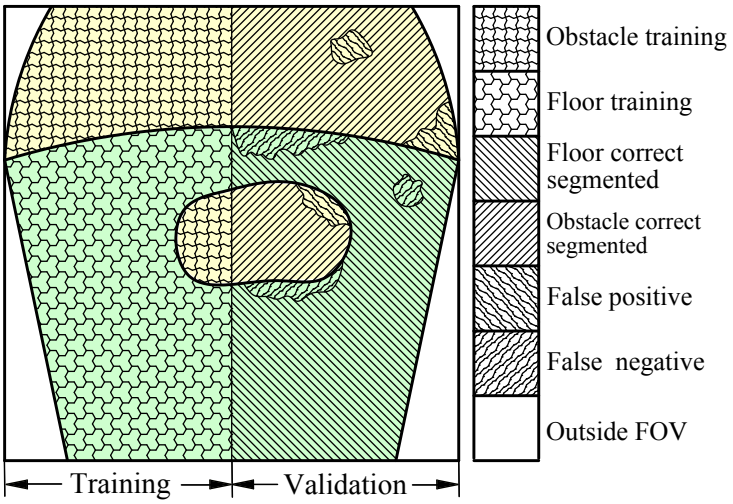


Figure 2.13: Segmentation validation

False positive rate and false negative rate are aggregated into a total classification error

$$f = (2f_p + f_n)/3 \tag{2.4.1}$$

in which false positives are weighted twice as strong since an obstacle miss is potentially more severe in the context of obstacle avoidance.

The segmentation validation utilizes two-fold cross-validation, in which the roles of the training dataset and the validation dataset are reversed and the classification error is averaged over both folds. The segmentation method with the lowest aggregated

classification error is applied to the entire omnidirectional view. In order to save computational resources, the segmentation validation and selection is only repeated every fifth frame. The final segmentation is filtered by a 5x5 median filter in order to eliminate isolated pixels and noise.

2.4.3 Experimental Results

The performance of the system is validated on 500 images with ground truth obtained from PMD data and 30 images in which the actual floor area is segmented by hand. Table 2.2 compares the true positive and false negative rates of watershed and region growing seeded with one percent of labeled pixels as seeds, and histogram backprojection using different models according to the seeds: floor, obstacles, and background. From the results, we can conclude that selecting among several alternative models outperforms using a single floor model. Hue and saturation are more reliable cues compared to normalized color (r and g). On both test sets, the hue saturation classifier with three models achieves true positive (floor classified as floor) rates between $0.85 - 0.9$ if one accepts a false positive (obstacle classified as floor) rate between $0.1 - 0.13$. Selecting the best among all segmentation leads to the value of 0.085 (lowest false positive rate in the hand-labeled set) In the context of navigation, the false positive rate is the critical variable to consider since missing an obstacle is more severe than underestimating the free space.

The classification performance is similar across the PMD validation, and the hand-labeled data set. In fact, the classification error on the hand-labeled data is even lower. We attribute this phenomenon to the fact that the depth-based segmentation is less accurate than hand segmentation. Thus, the test data set itself contains a small fraction of incorrect samples that contribute to the classification error.

The classification performance of watershed and region growing is superior on the PMD test set, with watershed outperforming region growing. It is not surprising that watershed and region growing achieve high classification rates on the PMD test set, as the seeds stem from the same narrow frontal region as the test data. The true generalized classification error of watershed and region growing becomes apparent on the manually labeled data set, in which the test data is uniformly distributed across the omnidirectional view. The dependence of watershed and region growing on a representative set of seeds is a definite disadvantage over more robust histogram backprojection.

Figure 2.14 shows the results of eight prototypical scenarios, together with the corresponding floor segmentation results. The system is robust even with a tiled floor, strong sun reflections, and imperfect lighting conditions. The second-row example shows the ability to adapt to new unseen floor surfaces where floor color and texture abruptly change behind the door.

Table 2.2: Comparison of the segmentation schemes under two data sets

	PMD ground truth		Hand labeled	
Method	FPR	TPR	FPR	TPR
H-S histogram features: F	0.173	0.855	0.089	0.905
H-S histogram features: F|O|	0.123	0.845	0.094	**0.971**
H-S histogram features: F|O|B	0.137	0.845	0.096	0.911
r-g Histogram features: F	0.210	0.828	0.118	0.923
r-g Histogram features: F|O	0.145	0.836	0.105	0.870
r-g Histogram features: F|O|B	0.147	0.825	0.112	0.896
Watershed	**0.021**	**0.948**	0.140	0.749
Region growing	0.097	0.932	0.149	0.676
Histogram all Feat. + Watershed + Region Growing	0.158	0.867	**0.085**	0.904

F: Floor, O: Obstacle, B: Background

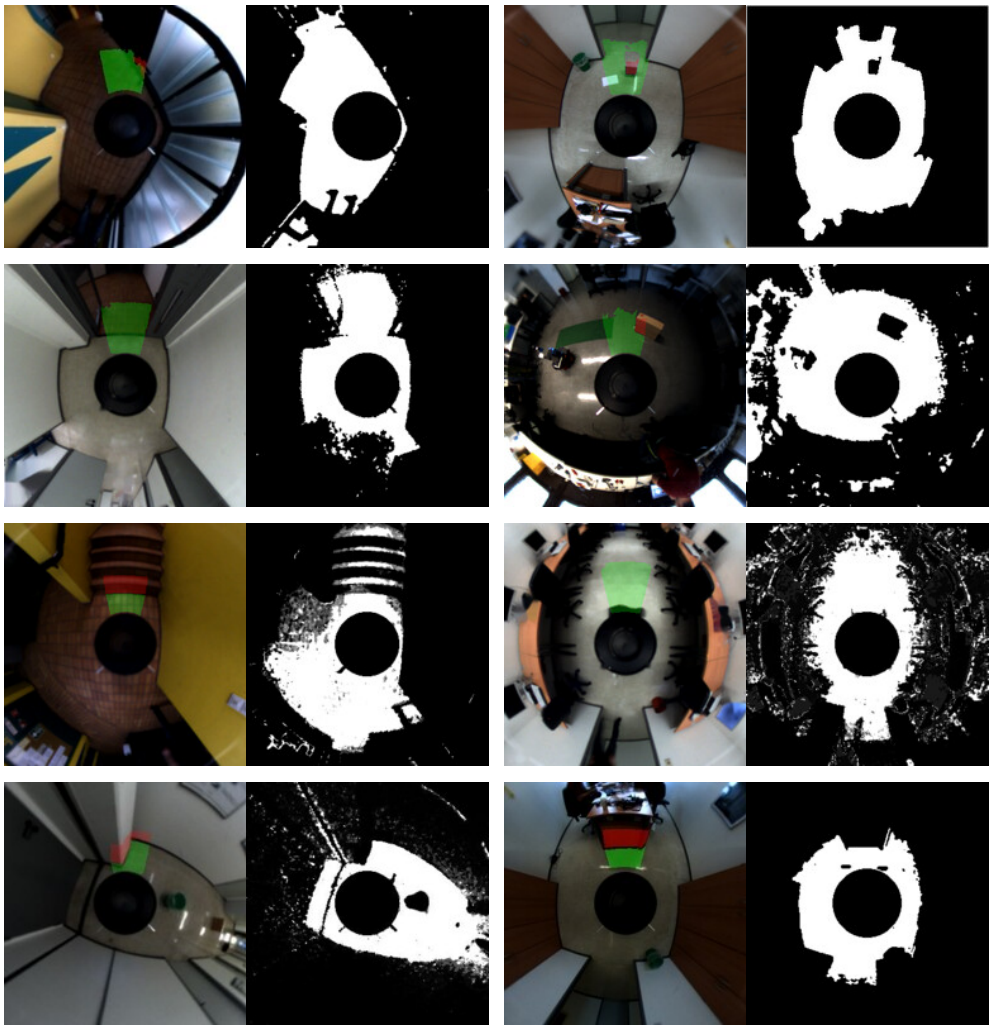


Figure 2.14: Segmentation Examples. The first and third rows show the omnidirectional input image and the projected mask from the 3D segmentation. The second and fourth rows show the output segmentation.

2.5 Summary

In this chapter, we introduced two schemes for detecting free-space in omnidirectional images. The first scheme follows supervised training with the mixture of experts paradigm in which the aggregation of multiple heterogeneous classifier opinions improves the overall classification performance. The classifiers are trained with ground truth segmentations provided by scanning different scenarios around 360° with a PMD 3D camera. Histogram backprojection, marker-based watershed, region growing, and graph-based segmentation provide the core information for the classifier ensemble. Region growing and graph segmentation are augmented by fusing their segmentation with horizontal and vertical lines extracted from edge detection. Moreover, a classifier based on sonar range readings further improves the classification. The expert decisions are combined with stacked generalization, behavior knowledge space or voting. The second proposed scheme is a novel online self-supervised free-space detector, which segments the free-space in omnidirectional images by determining the segmentation optimal for the current situation by cross-validation over ground-truth data provided by a PMD range camera. Both systems are robust for safely navigating in unstructured environments of diverse appearance, texture, and illumination. The systems were tested successfully and validated in robotics runs over several hours.

Several extensions are worth discussing. For instance, implementing more robust segmentation methods with the new advances in convolutional neural networks technology or merging more sensor modalities such as laser or ultrasound range, for a self-supervision in the omniview, with larger range and field-of-view than the 3D camera.

3

Uncertainty Representation with Fuzzy Preference Structures

3.1 Introduction

In the last chapter, we introduced a visual range sensor able to extract the robot's free-space by fusing multiple cues stemming from different segmentation schemes. The key concept was to combine multiple segmentations in such a way that the overall decision-making benefits from the mutual competences of the heterogeneous features. In this chapter, we extend these methods to explicitly model the uncertainty in the classification through fuzzy preference structures [HB08]. In this way, it is possible to establish a relationship between classification learning, fuzzy preference modeling, and decision making, where the problem of classification is mapped into the domain of decision-making under fuzzy preferences.

We investigate the uncertainty inherent in a single segmentation method as well as the global uncertainty across multiple segmentation based classifiers. The results demonstrate that the classification is substantially reduced by rejecting those queries associated with a strong degree of conflict in the case of single segmentations, and ignorance in the case of ensemble decisions.

The classification method combines pairwise decomposition techniques with ideas and tools from fuzzy preference modeling. More specifically, we built upon the approaches [HPB11; Pos+11a], which first decompose a polychotomous classification problem involving m classes into an ensemble of binary problems, one for each ordered pair of classes. The corresponding classifiers are trained on the relevant subsets of the (transformed) original training data. In the classification phase, a new query is submitted to every binary learner. The output of each classifier is interpreted as a fuzzy degree of preference for the first in comparison with the second class. By combining the outputs of all classifiers, one thus obtains a fuzzy preference relation, which is taken as a point of departure for the final classification decision. This way, the problem of classification is effectively reduced to a problem of decision making based on a fuzzy preference relation. Corresponding techniques, which have been investigated quite intensively in the field of fuzzy set theory, hence become amenable to the task of classification. In particular, by decomposing a preference relation into a strict preference, an indifference, and an incomparability relation, this approach

allows one to quantify different types of uncertainty in classification and thereby supports sophisticated classification and postprocessing strategies.

3.2 Related Work

The role of uncertainty in the context of autonomous robot navigation has been long recognized in the robotic community [DS01; DS13; PPP17]. However, the majority of approaches are concerned with behavior design in terms of reactive rule-based fuzzy control. They lack explicit representation of the uncertainty inherent in the perception of the environment attributed to imprecise sensors and the uncertain effect of actions. A fuzzy approach for map building and path planning is investigated in [OUV98; OUV97]. The authors in [NR07] demonstrate that in some scenarios, the fuzzy fusion of sensor data is advantageous in comparison to probabilistic approaches for map building. A comparison of Bayesian, Dempster-Shafer and fuzzy set theory to represent uncertainty in the context of occupancy grid mapping from sonar data is provided in [RP01]. The authors show that in the case of ambiguous range readings, the probabilistic approach is more robust. However, to our best knowledge, none of these approaches distinguishes among the causes of uncertainty, namely lack of evidence or conflicting evidence.

Most realistic classification problems exhibit uncertainty and ambiguity to some extent. This inherent imprecision is either ignored or represented in a probabilistic manner or in terms of a fuzzy model. Nevertheless, most fuzzy classifiers constitute crisp classifiers in that even though they rely on fuzzy inference; they operate with crisp inputs and outputs. Only recently, vague data is explicitly considered in the design and optimization of fuzzy classifiers [SC07].

3.3 Fuzzy Preference Structures

Fuzzy preference structures introduced in [HB08] establish a relationship between classification learning and fuzzy preference modeling and decision making. The problem classification is mapped into the domain of decision making under fuzzy preferences. The resulting structure is a fuzzy relation, which for every pair of labels λ_i, λ_j defines a degree of:

- Preference: which determines the degree to which the classification λ_i is preferred over λ_j,
- Conflict (indifference): the degree to which both labels are in conflict in the sense that both classifications are supported by training data and,
- Ignorance (incomparability): the degree reflects the lack of information supporting either classification.

In summary, preference denotes the preferred classification if the learner is forced to make a classification. Conflict reflects the proximity of an unknown instance to the decision boundary that separates the two classes in case of sufficient training data.

Ignorance reflects the lack of data in case the training set contains neither evidence for λ_i nor λ_j.

3.3.1 Preference Structure and Classification

The idea with Fuzzy preference structures for classification is to interpret the output of a binary classifier $r_{i,j} = M_{i,j}(x)$ for a query x as a preference of class label λ_i in comparison with λ_j. In this sense, the value $r_{i,j} \in [0,1]$ can be interpreted as the degree to which classification λ_i is preferred over λ_j. Denoting $r_{i,j} = \mathcal{R}(\lambda_i, \lambda_j)$, the matrix

$$\mathcal{R} = \begin{bmatrix} 1 & r_{1,2} & \dots & r_{1,m} \\ r_{2,1} & 1 & \dots & r_{2,m} \\ \vdots & & \vdots & \\ r_{m,1} & r_{m,2} & \dots & 1 \end{bmatrix} \tag{3.3.1}$$

is obtained by merging the binary fuzzy classifiers for each possible combination of class labels into a fuzzy preference relation.

We are interested in decomposing the weak preference relation $\mathcal{R}$ into a fuzzy preference structure that describes the strict preference relation $\mathcal{P}$, the indifference relation $\mathcal{I}$, and an incomparability relation $\mathcal{J}$. In principle, any fuzzy t-norm and its dual t-conorm are suitable to obtain preference structure $(\mathcal{P}, \mathcal{I}, \mathcal{J})$ [HB08]. The employed decomposition is given by:

$$\mathcal{P}(\lambda_i, \lambda_j) = r_{i,j}(1 - r_{j,i}) \tag{3.3.2}$$
$$\mathcal{I}(\lambda_i, \lambda_j) = r_{i,j} r_{j,i} \tag{3.3.3}$$
$$\mathcal{J}(\lambda_i, \lambda_j) = 1 - (r_{i,j} + r_{j,i}) \tag{3.3.4}$$

Notice that the definition of incomparability $\mathcal{J}$ coincides with the width of the distribution. At the extreme values $r_{i,j} = 1 - r_{j,i} \in \{0,1\}$ of the weak preference, we obtain a strict strong preference $\mathcal{P}(\lambda_i, \lambda_j) \in \{0,1\}$. Finally, the degree of indifference $\mathcal{I}$ assumes its maximum for $r_{i,j} = r_{j,i} = 0.5$, and the classifiers have no preference for either class.

Uncertainty is investigated on the basis of the classification of a single-pixel supported by a single segmentation. This uncertainty reflects the ambiguity of a particular segmentation in associating a pixel to a particular class based on its color or similarity with neighboring pixels. It provides insight as to which segmentation is best suited for the particular context in terms of appearance and illumination of the environment. In addition, we investigate uncertainty in the context of an ensemble of classifiers across an entire image region. In this case, the image is partitioned into homogeneous regions by unsupervised segmentation with watershed and graph cut. Uncertainty depends on the amount of agreement among the ensemble across all pixels that constitute a region. Significant discrepancies in the classification across the ensemble denote ignorance, whereas conflict is indicated by indifferent preference of individual classifiers on either class.

Valued Preferences for Individual Segmentations

In the following, we detail how the fuzzy preference relation is obtained from data. We assume an ensemble of binary classifiers, in our case, a set of diverse segmentation schemes, in which a learner predicts a score $s_{i,j} \in [0,1]$, which is interpreted as a fuzzy preference for class λ_i over class λ_j. The aggregation of scores over the entire ensemble of classifiers generates the fuzzy preference relation $r_{i,j}$. We first describe the mapping from segmentations onto the scores $s_{i,j}$, which differs with the type of segmentation processing [HPB11].

Histogram Backprojection

Our scheme maintains multiple models $M_k(c(u,v))$ for the classes floor λ_f, obstacle λ_o, and background λ_b. The similarity of the current histogram with a class is given by the maximum similarity with any of its representing models:

$$B_i(u,v) = \max_{k \in \lambda_i} \min \left(\frac{M_k(c(u,v))}{H(c(u,v))}, 1 \right). \tag{3.3.5}$$

Notice, that the similarity measure $B_i(u,v)$ is interpretable as a fuzzy degree of membership of pixel (u,v) to class λ_i. The weak preference among two classes λ_i, λ_j is given by the difference in the similarity of the image with the histogram models representing classes λ_i, λ_j:

$$s_{i,j} = (B_i(u,v))(1 - B_j(u,v)). \tag{3.3.6}$$

Notice that our scheme generates two backprojection images based on the hue-saturation and red-green color spaces. The backprojection images capture the similarity between a pixel and the color histogram of a class λ_i. This observation motivates the following definitions of strong preference, conflict, and incomparability in the case of backprojection:

$$\begin{aligned} \mathcal{P}(u,v)(\lambda_i,\lambda_j) &= B_i(u,v)(1 - B_j(u,v)) && (3.3.7)\\ \mathcal{I}(u,v)(\lambda_i,\lambda_j) &= B_i(u,v) * B_j(u,v) && (3.3.8)\\ \mathcal{J}(u,v)(\lambda_i,\lambda_j) &= 1 - (B_i(u,v) + B_j(u,v)). && (3.3.9) \end{aligned}$$

Considering the extreme cases $B_i(u,v) \in \{0,1\}$, namely the color of a pixel either matches a model $B_i(u,v) = 1$ or does not $B_i(u,v) = 0$. Conflict corresponds to the case in which a pixel matches both models $B_i(u,v) = B_j(u,v) = 1$. Thus there is sufficient data similar to pixel (u,v), but the two classes are highly similar in appearance. Ignorance prevails in case a pixel matches none of the models $B_i(u,v) = B_j(u,v) = 0$, in this case, the color of pixel (u,v) does not appear in the training set, which constitutes a lack of data to provide a competent classification.

Region Growing

Region growing was already described in the previous chapter. It provides the membership degree of a pixel to a class λ_i by

$$\mu_i(u,v) = \max\{\frac{N_i(u,v) \cap M_i}{N_i} \frac{}{\overline{M_i}}, 1\}. \tag{3.3.10}$$

The reasoning is similar to the considerations in histogram backprojection, where a low degree of membership $\mu_i(u,v), \mu_j(u,v) \approx 0$ for both classes is evidence of ignorance, whereas a high degree of membership for both classes $\mu_i(u,v), \mu_j(u,v) \approx 1$ indicates conflict. This leads to the following definitions:

$$\mathcal{P}(u,v)(\lambda_i,\lambda_j) = \mu_i(u,v)(1-\mu_j(u,v)) \quad (3.3.11)$$

$$\mathcal{I}(u,v)(\lambda_i,\lambda_j) = \mu_i(u,v) * \mu_j(u,v) \quad (3.3.12)$$

$$\mathcal{J}(u,v)(\lambda_i,\lambda_j) = 1-(\mu_i(u,v)+\mu_j(u,v)). \quad (3.3.13)$$

3.3.2 Valued Preferences for Ensembles of Segmentations

Until now, we have focused on uncertainty related to a single segmentation. This information allows selecting the segmentation optimal for the current environment. Thus, the one with the lowest ignorance and conflict. However, in order to fully explore the potential of the fuzzy notion of uncertainty in terms of preference, ignorance, and conflict, the weak preferences of multiple classifiers are aggregated across regions in the image. This allows us to obtain sufficient statistics, and the level of strong preference, ignorance, and conflict is established according to the distribution of weak preferences across the region.

Homogeneous regions in the image are obtained using watershed and graph-based segmentation. Both methods operate in an unsupervised manner; hence, they do not consider the class label of a pixel in advance.

The key idea is to assess the ambiguity of a classification given an ensemble of binary scoring classifiers in which each classifier outputs a score $s_{i,j}^k$. Given a set of scoring classifiers the degree of weak preference for λ_i in comparison to λ_j is given by

$$r_{i,j} = \min_k s_{i,j}^k \quad (3.3.14)$$

In the case, where the analysis aggregates statics per region, rather than on the basis of single pixels, the weak preference for a region R_n is defined as

$$r_{R_n i,j} = \min_{(u,v)\in R_n,k} s_{i,j}(u,v)^k \quad (3.3.15)$$

in which the minimum is computed over the pixels that belong to R_n and the set of segmentation based classifiers. As the min operator is rather sensitive to noise and outliers, it is advisable to replace it by the α-quantile of the distribution. In our case, the 20% lowest scores are ignored, and the weak preference is given by the $\alpha = 0.2$ quantile of $s_{i,j}(u,v)^k$. The distribution of scores allows an intuitive interpretation in terms of conflict and ignorance. The width of the distribution of the $s_{i,j}^k$ is a measure of ignorance, e.g., the ignorance vanishes in case all models predict the same score.

3.3.3 Experimental Results: Classification of Indoor Scenarios

The validation data consists of about 480.000 pixels from 24 images captured with a photonic mixer device camera (PMD). The PMD generates the labeled instances for validation, where the true class label is established from the 3D segmentation of the PMD depth information.

Following the procedure from the previous chapter, we obtain ground truth labeled masks by rotating the robot and PMD camera by 360° and capturing scans at each pose. The captured images represent scenes such as corridors, open rooms, and confined spaces. The 3D data from the PMD is fitted to planar surfaces using RANSAC. Surface normal orientation, distance to the camera, and connected components determine whether a pixel and its associated 3D point are labeled as obstacles or floor.

Figs. 3.1-3.3 illustrate the fuzzy preference structures for omnidirectional images of prototypical indoor scenes. In the first scenario in Fig. 3.1, the floor and walls are easily distinguishable from their appearance and the scene is subject to homogeneous illumination. The basic segmentations according to backprojection based on hue and saturation and region growing exhibit no conflict, no ignorance on the floor region, and partial ignorance with respect to the obstacle regions. Ignorance emerges in the brighter areas of the wall for back projection as well as region growing. The combination of preferences across multiple segmentations mediated by watershed and graph cut exhibits a clear preference for either floor or obstacle in most parts of the image. Ignorance only emerges in the darker floor region in the outside corridor and in the shaded floor region at the bottom.

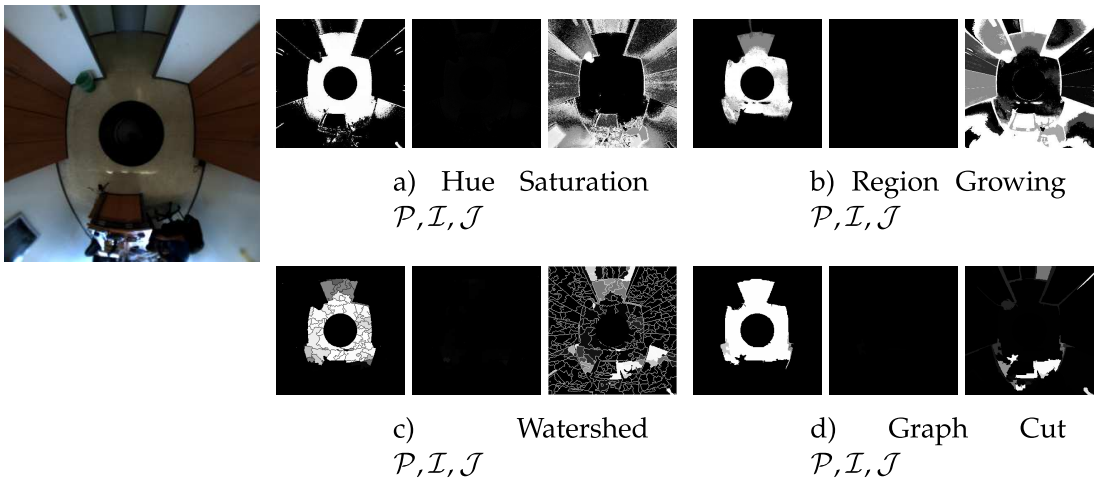
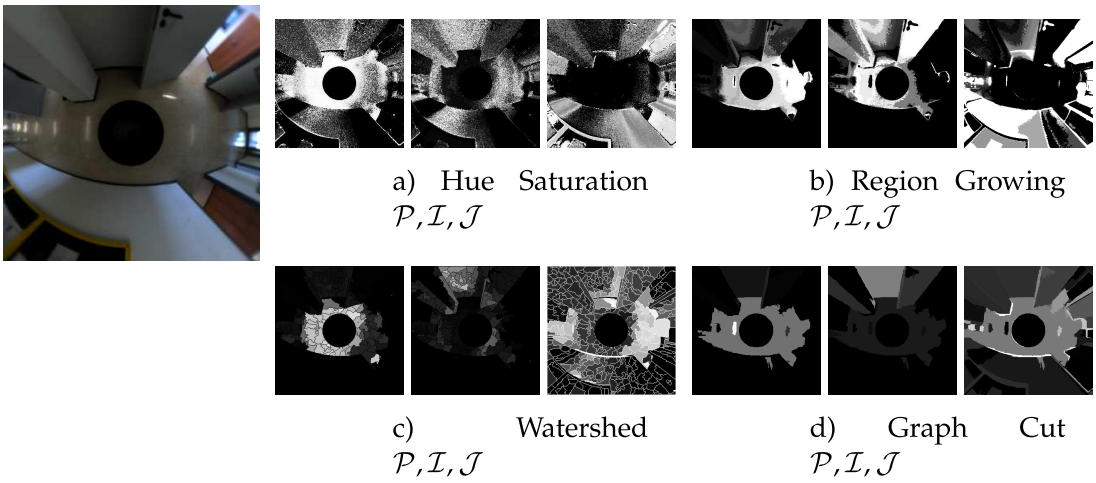


Figure 3.1: Scenario 1 : preference $\mathcal{P}$, conflict $\mathcal{I}$, and ignorance $\mathcal{J}$ for individual and ensemble of segmentations

The second scenario in Fig. 3.2 exhibits substantial ambiguity in particular as the floor and the darker parts of the wall are highly similar in appearance due to insufficient illumination. This ambiguity is apparent in terms of conflict in the individual segmentations based on backprojection and region growing. Conflict is also persistent in the ensemble of segmentations. Watershed and graph cut mainly differ in the granulation of the segmented regions, with watershed resulting in a finer tessellation compared to graph cut. The preference structure also exhibits regions of substantial ignorance, indicating strong variations in weak preferences among the ensemble classifiers.

The third scenario in Fig. 3.3 is characterized by a high level of ignorance partially attributed to the variation in illumination. Ignorance is strong except for the floor region at the top and the large wall region to the right. The appearance of the cluttered and shaded areas does not provide sufficient evidence for a reliable classification.

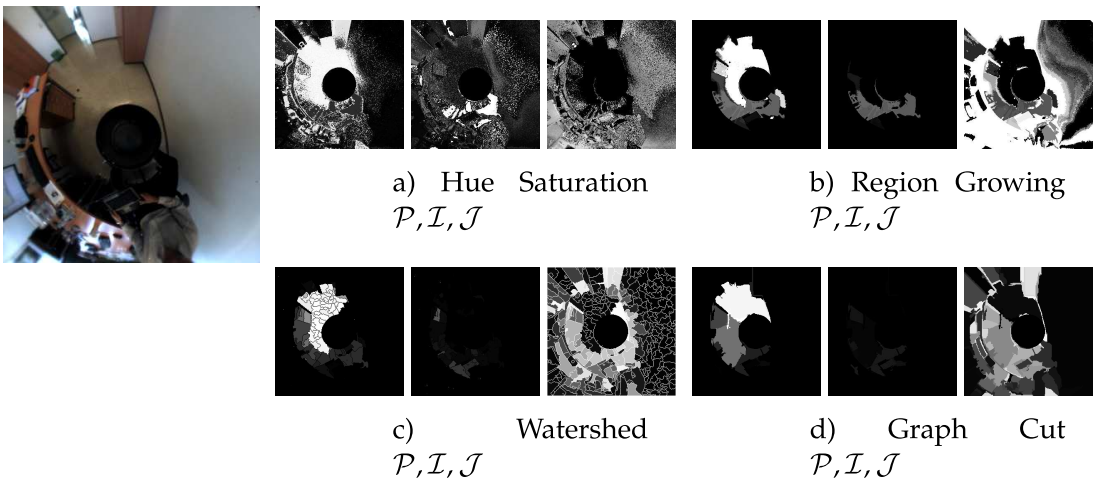


a) Hue Saturation $\mathcal{P}, \mathcal{I}, \mathcal{J}$ b) Region Growing $\mathcal{P}, \mathcal{I}, \mathcal{J}$

c) Watershed $\mathcal{P}, \mathcal{I}, \mathcal{J}$ d) Graph Cut $\mathcal{P}, \mathcal{I}, \mathcal{J}$

Figure 3.2: Scenario 2, preference $\mathcal{P}$, conflict $\mathcal{I}$, and ignorance $\mathcal{J}$ for individual and ensemble of segmentations

Despite the variations in illumination, conflict among the ensemble segmentations is rather minimal.



a) Hue Saturation $\mathcal{P}, \mathcal{I}, \mathcal{J}$ b) Region Growing $\mathcal{P}, \mathcal{I}, \mathcal{J}$

c) Watershed $\mathcal{P}, \mathcal{I}, \mathcal{J}$ d) Graph Cut $\mathcal{P}, \mathcal{I}, \mathcal{J}$

Figure 3.3: Scenario 3, preference $\mathcal{P}$, conflict $\mathcal{I}$, and ignorance $\mathcal{J}$ for individual and ensemble of segmentations

The effect of ignorance and conflict investigated on the 24 images, and 480.000 pixels is reported in Table 1.1. We report the Classification error for the segmentation methods

The effect of ignorance and conflict is summarized in Table 3.1. We report the Classification error for the segmentation methods backprojection with hue and saturation, region growing and the ensemble segmentations with watershed and graph cut. The analysis assumes that the classifier is allowed to reject a decision on those image regions for which the conflict or ignorance is high, thus focusing only on those decisions for which it has substantial confidence. Table 3.1 reports the relative classification error in case the classifier is allowed to reject a fraction $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4\}$ of queries with the highest ignorance or conflict.

It is to be expected that the classification error decreases with higher rejection rates. In the case of backprojection with hue and saturation $C_{hs}, \mathcal{I}_{hs}$, the classification error

Table 3.1: Classification error for hue-saturation backprojection C_{hs}, region growing C_{rg}, watershed C_{ws} and graph cut C_{gc}, classifiers based on the rejection of queries according to conflict $\mathcal{I}$ and ignorance $\mathcal{J}$.

α	$C_{hs}, \mathcal{I}_{hs}$	$C_{hs}, \mathcal{J}_{hs}$	$C_{hs}, \mathcal{J}_{ws}$	$C_{rg}, \mathcal{I}_{rg}$	$C_{rg}, \mathcal{J}_{rg}$
0.4	0.085	0.150	0.074	0.141	0.092
0.3	0.087	0.150	0.088	0.141	0.117
0.2	0.100	0.144	0.101	0.145	0.154
0.1	0.114	0.137	0.121	0.138	0.154
0.0	0.135	0.135	0.135	0.154	0.154

Table 3.1: **(Continued)** Classification error for hue-saturation backprojection C_{hs}, region growing C_{rg}, watershed C_{ws} and graph cut C_{gc}.

α	$C_{rg}, \mathcal{J}_{ws}$	$C_{ws}, \mathcal{I}_{ws}$	$C_{ws}, \mathcal{J}_{ws}$	$C_{gc}, \mathcal{I}_{gc}$	$C_{gc}, \mathcal{J}_{gc}$
0.4	0.068	0.157	0.054	0.150	0.054
0.3	0.077	0.170	0.059	0.170	0.062
0.2	0.090	0.167	0.067	0.167	0.085
0.1	0.119	0.162	0.124	0.187	0.135
0.0	0.154	0.174	0.174	0.194	0.194

is reduced if the classifier is allowed to reject queries according to its notion of conflict. Backprojection with hue and saturation does not capture ignorance well due to the lack of uncorrelated training data such that filtering queries based on ignorance $C_{hs}, \mathcal{J}_{hs}$ even deteriorates accuracy. The situation is reversed for region growing in which case rejecting based on ignorance $C_{rg}, \mathcal{J}_{rg}$ reduces the error rate, whereas rejection based on conflict $C_{rg}, \mathcal{I}_{rg}$ only has a marginal effect. In the case of the region growing classifier, the variation is accomplished by a random selection of seeds resulting in multiple independent classifications.

Ignorance is much easier to detect with the ensemble classifiers, according to equation 3.3.2, in terms of ambiguity across many classifications. Ambiguity emerges in a two-fold manner, namely ambiguity among different segmentation schemes as well as ambiguity in weak preference across the pixels that belong to the same region. The classification error for the watershed segmentation $C_{ws}, \mathcal{J}_{ws}$ is reduced from $e = 0.174$ to $e = 0.067$ if 20% of the queries are rejected and even to $e = 0.059$ in case 30% are rejected. A similar reduction is accomplished in case of ensemble classification based on graph cut segmentation $C_{gc}, \mathcal{J}_{gc}$, namely to $e = 0.085$ at 20% and $e = 0.062$ at 30% rejection rate. It is no surprise that for the ensemble classifiers, the impact on the classification accuracy related to conflict is less pronounced. Notice that the basic error rate without rejection is larger for the ensemble classifiers, in particular, in comparison to the hue saturation classifier. In the ensemble classifiers, a homogeneous classification is assigned to the entire region, whereas, in the case of the single classifiers, each pixel is classified individually. In the case of single-pixel classification, it is

difficult to capture ignorance due to the lack of a sufficient statistical basis. In the case of region-based classification, there are sufficient statistics to estimate conflict and ignorance; however, the accuracy of classifications suffers due to the coarse resolution. This observation motivates us to use the pixel-based segmentations for classification but reject queries based on ignorance captured by the ensemble classifiers. Table 3.1 reports the evolution of error rate for the hue saturation segmentation $C_{hs}, \mathcal{J}_{ws}$ with queries rejected according to ignorance in watershed segmentation. It clearly outperforms rejection based on its internal ignorance $C_{hs}, \mathcal{J}_{hs}$, but is inferior to the ensemble classifier $C_{ws}, \mathcal{J}_{ws}$, even at low rejection rates. In summary, the ensemble classifiers are best able to recognize ignorance whereas conflict is better revealed in the weak preference of single segmentations.

3.4 Ensemble of Experts with Fuzzy Preference Structures

In the previous section, we applied fuzzy preference structures across pixels and superpixels to explicitly represent the uncertainty in terms of preference, conflict, and ignorance. In this section, we built upon [Pos+11a] and extend the ensemble of experts approach from the previous chapter to include uncertainty information from the fuzzy preference structures into the classification.

The overall scheme is illustrated in Fig. 3.4. The first layer relies on three base experts that perform segmentation using: histogram backprojection with hue-saturation channels, histogram backprojection with normalized red-green color space and region growing. The ensemble layer consists of a set of classifiers that fuse the information from each expert. We investigate pixel-wise classifications and aggregation across regions of homogeneous appearance to obtain a segmentation robust to noise and outliers. Two separate appearance models for floor and obstacles are maintained, thus capturing the decision boundary between floor and obstacles in a more refined manner compared to a floor versus rest classification. Two ensembles of classifiers combine the evidence of single segmentation experts for both models. The first ensemble, called *obstacle classifier*, is intended to detect obstacles by fusing the segmentations arising from seeds and histogram models of obstacles, the second ensemble, called *floor classifier*, attempts to recognize floor regions by combining the segmentations using seeds and histogram models of the floor.

Inconsistencies in the predictions are resolved by decomposing the ensemble classifiers into a preference structure. Fig. 3.5 illustrates the pivotal processing steps to generate an obstacle floor classification from the original image of a typical indoor scene.

3.4.1 Classification Results

The training and validation data for evaluating the ensemble of experts with fuzzy preference structures consist of about one million pixels captured from 30 images of which the true class label is established from the PMD depth information and 3D segmentation. Half of the images are used for training and the other half for testing.

Table 3.2 reports the classification error rate after decomposing the *floor classifier* and *obstacle classifier* using fuzzy preference structures. The results show the pixel-wise

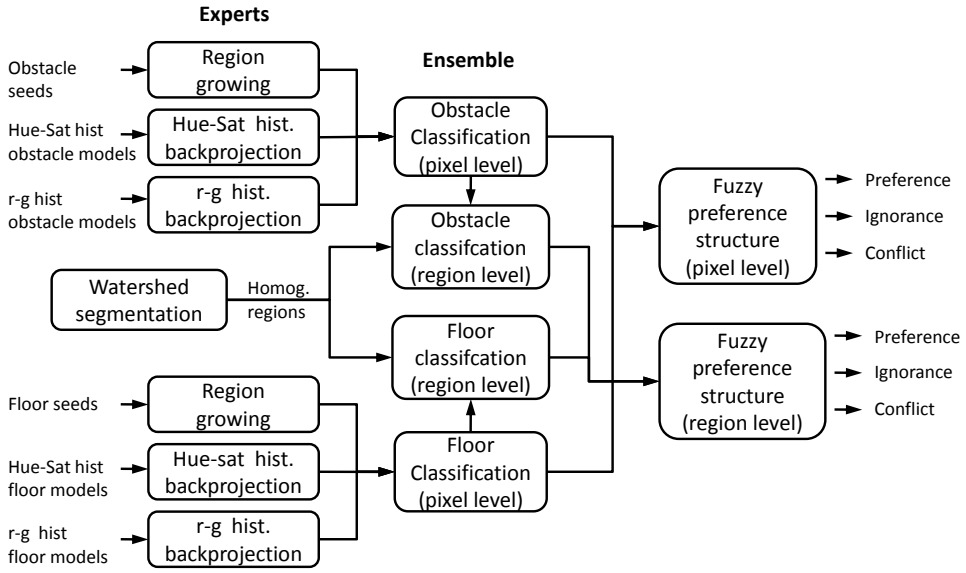


Figure 3.4: Mixture of experts classification with fuzzy preference structures integrated

classification and the region-wise classification with watershed regions. The results confirm the supposed improvement: computing statistics across a region improves the classification accuracy as the error is substantially reduced by rejecting those queries associated with a strong degree of conflict and ignorance.

Table 3.2: Classification error based on the rejection of queries according to conflict and ignorance

Pixel-wise			
Rejection rate	Ignorance 0%	Ignorance 10%	Ignorance 20%
Conflict 0%	0.152	0.140	0.133
Conflict 10%	0.134	0.118	0.107
Conflict 20%	0.128	0.109	0.102
Region-wise			
Rejection rate	Ignorance 0%	Ignorance 10%	Ignorance 20%
Conflict 0%	0.122	0.109	0.091
Conflict 10%	0.111	0.095	0.072
Conflict 20%	0.106	0.086	0.065

Fig. 3.6 illustrates the fuzzy preference structure for omnidirectional images of prototypical indoor scenes using the ensemble of experts. The two first scenarios illustrate a situation in which the robot encounters isolates obstacles of which the initial model or seeds are unknown. Such obstacles exhibit a high level of ignorance reflecting the lack of data to distinguish those novel objects into either floor or obstacle. The third scenario illustrates a scene of ambiguous appearance, which exhibits a substantial amount of conflict due to the similar color of floor and background. The following scenarios depict different levels of texture, illumination, and shadows.

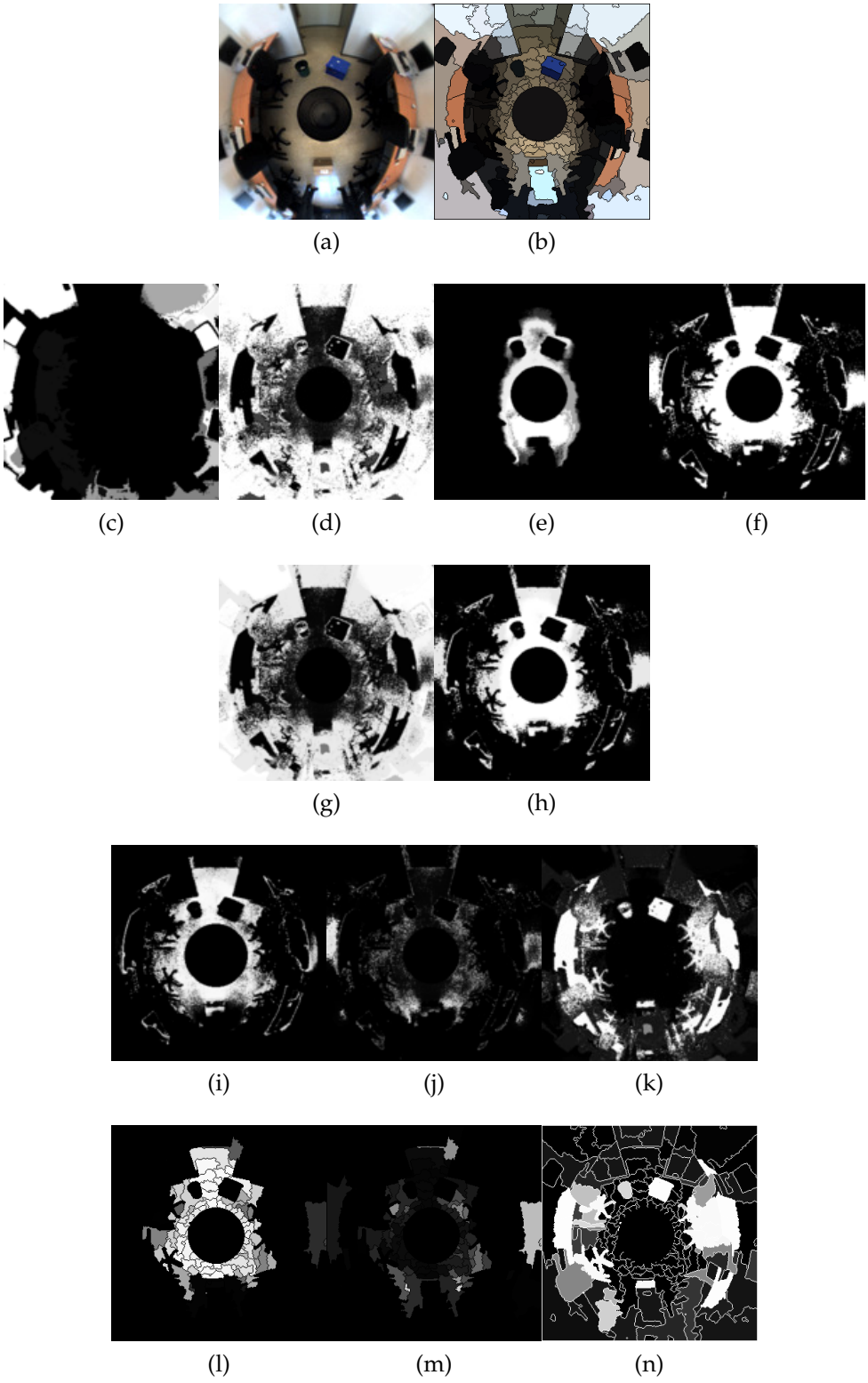
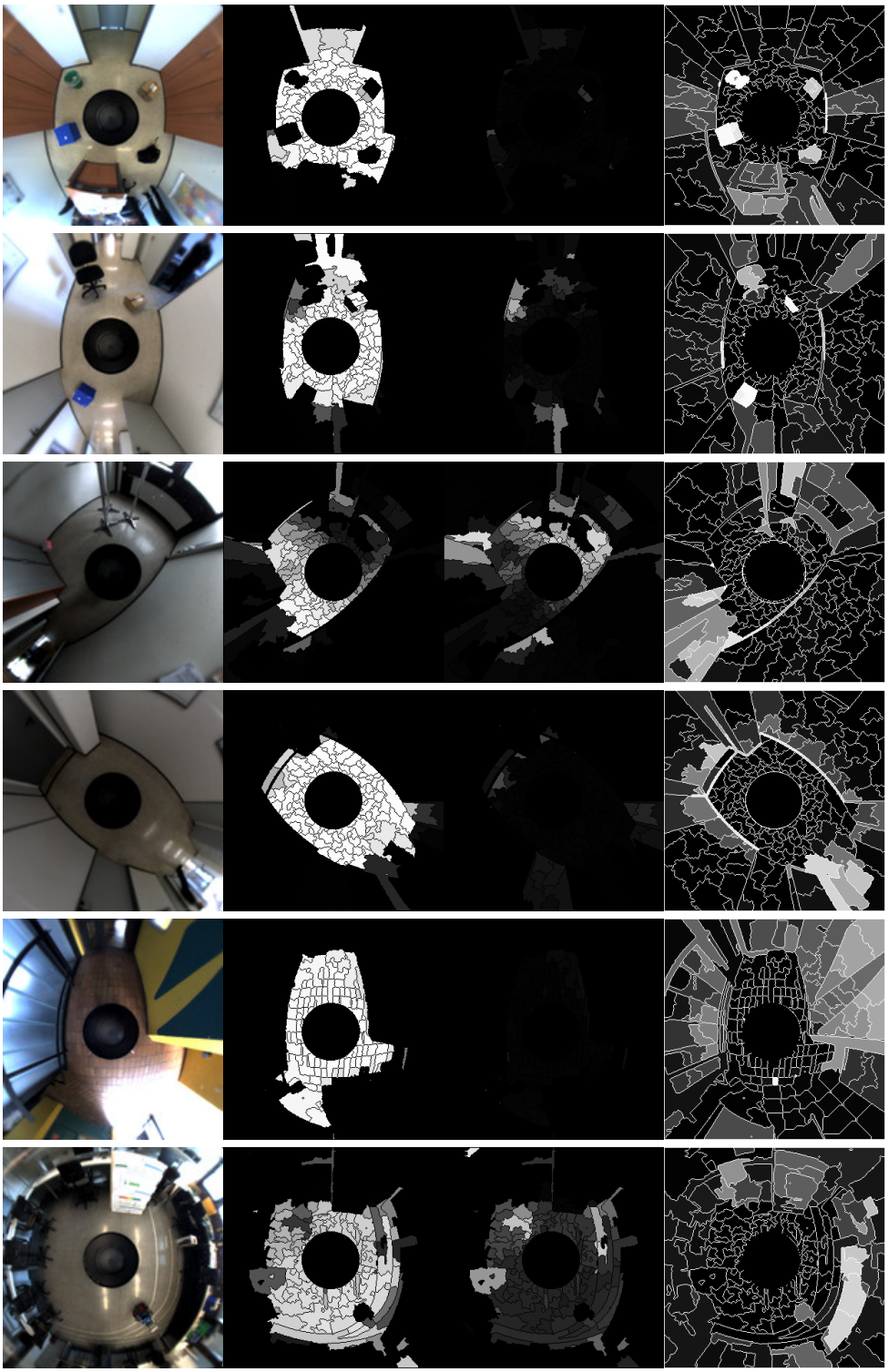


Figure 3.5: Overview of the processing steps in obstacle-floor detection: (a) Input image. (b) marker-based watershed. (c,d) Obstacle segmentations with region growing and histogram backprojection (hue-sat) respectively. (e,f) floor segmentations with region growing and histogram backprojection (hue-sat) respectively. (g,h) Obstacle and floor classifier (i,j,k) Pixel-wise preference, ignorance and conflict. (l,m,n) Preference, ignorance, and conflict processed per watershed regions (watershed lines shown for visualization purposes).



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

Figure 3.6: Classification examples: (a) Input image, b) Preference $\mathcal{P}$, c) Conflict $\mathcal{I}$, d) Ignorance $\mathcal{J}$ (watershed lines shown for visualization purposes)

3.5 Summary

This chapter has presented a floor/obstacle classifier able to represent the implied uncertainty with fuzzy preference structures. The uncertainty of the classifications is explicitly expressed in terms of preference, conflict, and ignorance utilizing fuzzy preference structures.

The preference structure captures the uncertainty inherent to individual segmentations as well as an ensemble of segmentation based classifiers. We also apply this technique to the ensemble of experts paradigm from the previous chapter and fuse heterogeneous segmentations schemes maintaining separate models for floor and obstacles.

The experimental results confirm the initial hypothesis that multiple diverse visual cues in conjunction with an explicit representation of uncertainty attributed to ignorance and conflict are of paramount benefit to vision-based robot local navigation. The method can be extended incorporating additional features, such as ones learned from data and domain specific knowledge to further improve the distinction of novel terrain and obstacles.

4

Semantic Segmentation of Scenes

4.1 Introduction

Robots require a more profound semantic understanding of the environments in which they operate to comprehend human task descriptions in natural language. The ability to reason about objects and their spatial and functional relationship entitles the robot to perform complex tasks autonomously and make human-robot communication more natural.

This chapter presents an approach that segments a scene and provides a semantic label for each region (see Fig. 4.1). The region labeling distinguishes between three main classes: floor, vertical structures, and clutter. The approach does not seek to identify every single object in the scene, but only navigation relevant categories. The category, vertical plane, is composed of objects such as walls, boards, and placards. Non-planar objects with multiple surfaces, e.g., furniture, plants, people or obstacles, are grouped into the class clutter.

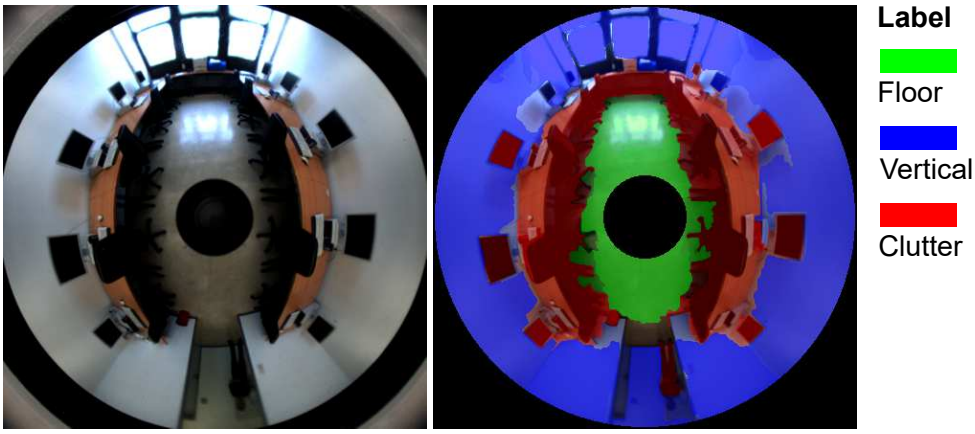


Figure 4.1: Semantic labeling of an indoor omnidirectional image. Three labels are shown: green indicates the floor region ; blue the vertical structures; and red clutter or obstacles

Compared to the approaches presented in previous chapters, the current chapter deals with more sophisticated visual features, and the proposed method is able to classify image regions beyond the classes ground/obstacle. This forms the basis for more advanced analysis such as layout reconstruction, which will be introduced in chapter 5.

Our method has the following characteristics: it is purely vision-based and does not rely on range information or a geometric reconstruction of the scene; it operates on single images and does not depend on a sequence of observations. The method uses a probabilistic representation that allows the fusion of observations into semantic occupancy grid maps (chapter 8). The training of the system, in contrast to more complex models such as Convolutional Neural Networks, does not require a massive dataset. The method is suitable for online learning and self-supervised with a range camera.

4.2 Related Work

The computer vision community has studied for several decades the general problem of semantic segmentation using a diverse number of object categories. The most important methods are surveyed in [Zhu+16; Tho16; GG+17].

Early approaches are based on hand-crafted features, while the current trend is to increasingly rely on highly discriminative features learned from data using deep learning [LBH15]. Deep learning approaches are reborn with the GPU processing era and the availability of huge labeled datasets such as the ImageNet [Rus+15]. The learned features consistently [LSD15; Zhe+15; Che+16] achieve superior results compared to hand features on most of the publicly available datasets. However, training such deep neural network architectures comes at the expense of requiring a massive amount of data for fitting the millions of parameters of these large variance models. Further, to apply deep learning networks to the segmentation task (where currently only small datasets exist), it is required to use the so-called transfer learning [LSD15]. Transfer learning fine-tunes an already pre-trained model (e.g. on object recognition with the ImageNet dataset) to work in another task (e.g. segmentation) by adding additional layers that adapt to the new task.

Engineered features continue to play a role in tasks where labeled data availability is limited, and learning starts from scratch. In this study, we employ engineered features using omnidirectional images where labeled datasets are inexistent. Very few semantic segmentation approaches have been proposed using purely omnidirectional images [Per+07; Pos+13]. The method in [Per+07] presents a system able to discriminate buildings and nature from sub-images extracted from an unwrapped omni-image. The approach on which we build upon [Pos+13] employs unsupervised over-segmentation (similar to [HEH07; SSN09]) to obtain image regions that are labeled with multi-cue statistical learning models.

The robotics community, in counterpart, has focused on labeling mainly objects relevant for robotics tasks such as terrain, sky, vegetation, obstacles, walls, doors, windows, etc. [NH08; Pos+13; Wol+14; Val+16]. Nüchter and Hertzberg [NH08] analyze 3D laser scans to identify floor, walls, and doors. The work of [Wol+14] employ point clouds from an RGBD camera using Randomized Decision Forest. More recently, [Val+16] presented a very successful system for outdoor labeling with a Deep Neural Network. The published dataset of 15.000 images was released to the public.

4.3 Semantic Segmentation

The objective of semantic region classification is to segment and classify different regions: floor, vertical planar surfaces (wall, placards), and isolated objects (e.g. plants, people). The semantic segmentation extracts multiple heterogeneous visual features at the superpixel-level that are labeled by randomized decision trees. The overall system architecture is based on [Pos+13; PHB14] and is shown in Fig. 4.2

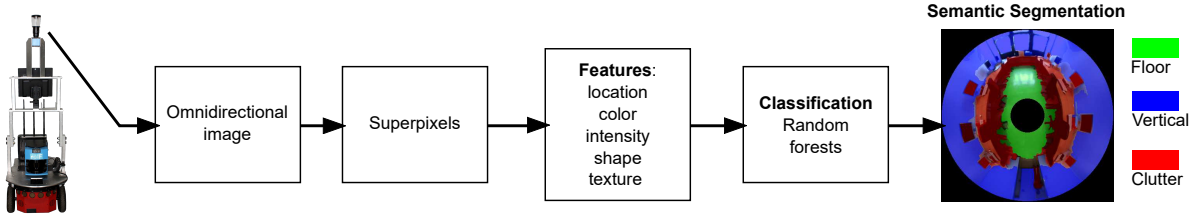


Figure 4.2: System architecture of the semantic segmentation

4.4 Features

Similar to previous approaches [HEH05; DLN05; FVS09; Pos+11a], our vision-based classifier gains statistical support from groups of pixels called superpixels. The objective is to generate superpixels that preserve image structure, but at the same time are computed efficiently. The graph-based segmentation [FH04] fulfills these requirements in the context of our application. It takes on average 200ms to process an image of 440x440 pixels while investigated alternative methods such as the entropy rate segmentation [Liu+11] took five times longer. Superpixels obtained with this method exhibit the best compromise in preserving structure in both simple and complex scenes since it does not need to fix a priori the number of superpixels or balance the superpixel in terms of size. In Fig. 4.3 three examples are shown with different parameter configurations.

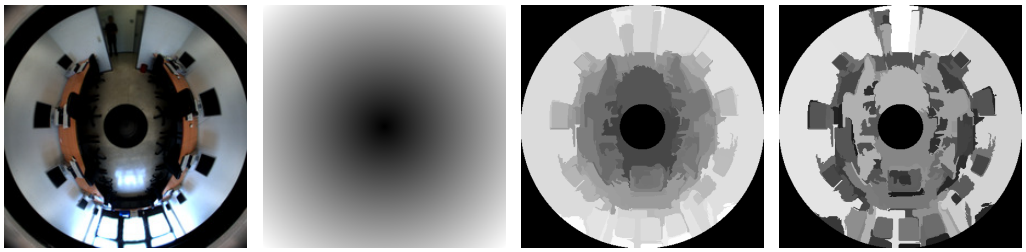
(a) (b) (c) (d)

Figure 4.3: Graph based segmentation [FH04] (a) Input image (b) σ = 0.5, k = 50 (c) σ = 0.5, k = 100 (d) σ = 0.5, k = 200

4.4.1 Location

Location is an important feature in our system since it provides a strong prior to the superpixel class. The omnidirectional camera field of view partially extends beyond

the horizon, and those pixels certainly do not belong to the floor. Likewise, large portions of the floor regions intersect with a blind circular spot in the image center. The spatial information of a superpixel is captured by the features: average, maximum, minimum, first and third quartile, and the standard deviation of the radial distance measured from the image center of all pixels inside the superpixel. All features are normalized with respect to image size to achieve a better image resolution invariability. Fig. 4.4 illustrates some radial distance features computed per superpixel.



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

Figure 4.4: Examples of radial distance features a) Input image b) Radial distance from the image center c) Average distance per superpixel d) Standard deviation per superpixel

4.4.2 Color

The color distribution in a superpixel is captured with three bin local histograms using RGB and HSV colorspaces, and with the mean, standard deviation, and first and third quartile of the RGB and HSV channels. Additionally, two separate color models are generated for the region above the horizon and in the vicinity of the image's center. The models are 2D hue-saturation histograms sampled from the mentioned regions, and histogram backprojection computes the degree of similarity with the reference models. The statistics average, the first and third quartile of the two backprojection images are computed.

4.4.3 Local Intensity Distribution

The local intensity distribution features are computed using entropy, mean, variance, and range of the intensities per superpixel. Entropy is extracted in a $9x9$ window around each pixel and the average value for the whole superpixel. Entropy is useful for capturing the randomness of the distribution and is defined as:

Entropy: $E = -\sum_{i=0}^{L-1} p(x_i) \log_2 p(x_i)$

where x_i is a discrete random variable that denotes intensity levels and $p(x_i)$ is computed from the normalized histogram (sum up to 1) with levels: $i = 0, 1, ..., L-1$.

Mean: $\mu = \sum_{i=0}^{L-1} x_i p(x_i)$

Variance: $\sigma^2 = \sum_{i=0}^{L-1} (x_i - \mu)^2 p(x_i)$

Range: $\max_i x_i - \min_i x_i$

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

Figure 4.5: Local intensity statistics features a) Input image b) Entropy c) Standard deviation d) Range e) Median

4.4.4 Shape

Shape features are computed using image moments [PR92], regional and boundary descriptors, Hu invariants [Hu62], shape topology, and convex hull properties.

Image Moments

The image moments m_{pq}, of order p, q, of a discrete image $I(x, y)$ are computed as:

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y) \tag{4.4.1}$$

The centroid is computed using the zero and first-order moments

$$x_c = m_{10}/m_{00} \; ; \; y_c = m_{01}/m_{00} \tag{4.4.2}$$

The central moments are computed with respect to the centroid to achieve translation invariance

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q I(x, y) \tag{4.4.3}$$

The normalized central moments are normalized with respect to the zeroth moment to achieve invariance to scale.

$$\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma} \; ; \; \gamma = \frac{p+q}{2} + 1 \tag{4.4.4}$$

The two principal axes of the shape region are approximated by an ellipse of equivalent inertia with semi major axis a, semi minor axis b, and orientation φ by:

$$\varphi_a = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \text{ where } \mu_{20} \neq \mu_{02} \tag{4.4.5}$$

$$\begin{aligned} a &= \sqrt{\frac{2\left[\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}\right]}{\mu_{00}}} \\ b &= \sqrt{\frac{2\left[\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}\right]}{\mu_{00}}} \end{aligned} \tag{4.4.6}$$

Hu Invariants

The Hu moment invariants [Hu62] extend the normalized central moments (translation and scale-invariant) to achieve rotation and mirroring invariance. They are constructed with the following seven polynomials using second and third-order moments and combinations:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{4.4.7}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{4.4.8}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{4.4.9}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{4.4.10}$$

$$\begin{aligned}\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{4.4.11}$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{4.4.12}$$

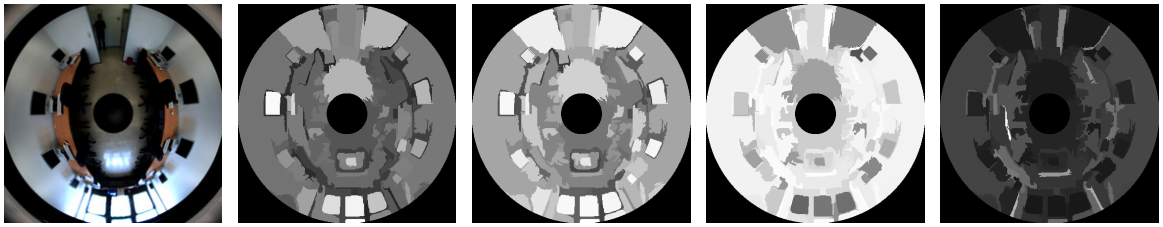
$$\begin{aligned}\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - \\ (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{4.4.13}$$

Regional Descriptors

The regional analysis [GW08; FCCJ09] employs a binary mask extracted from each superpixel forming a region R, its boundary contour C, its area A, and perimeter P. The following features constitute the regional descriptors:

- Area: $A = m_{00}$ is computed as the sum of intensity levels in a region
- Perimeter: P is a scalar that measures arc length around the boundary of a region
- Circularity ratio: $R_c = 4\pi A/P^2$, where the maximal value of 1 is attained when R is circular and $\pi/4$ when it is a square. This descriptor is invariant to scale and rotation.
- Compactness: P^2/A is an alternative method to circularity ratio.
- Area to perimeter ratio: A/P
- Rectangularity: A/A_{bbox}, with A_{bbox}, the area of the bounding box.

- Solidity: A/A_{chull} is scalar specifying the proportion of pixels in the convex hull that are also in the region. A_{chull} is the convex hull area.
- Irregularity: $I(R) = \frac{\pi \max_{(x,y)\in R} \left((x-x_c)^2 + (y-y_c)^2\right)}{A}$ this feature is similar to solidity but considers area (π) of the maximum enclosing circle in the numerator computed by calculating the major chord length in R.
- Major and minor axis length: a, b; Aspect ratio: a/b; and orientation of the major axis: φ_a
- Eccentricity: $E = \sqrt{1 - (\frac{a}{b})^2}$ is a scalar that specifies the eccentricity of the inertia equivalent ellipse of R. With $E \in (0, 1)$. The value of 1 is obtained when both axes have the same length, meaning that the ellipse degenerates to a circle. The value of 0 represents the opposite case, and the ellipse degenerates to a line.
- Relation of the major axis to the perimeter: a/P
- Number of convex hull vertices
- Mean boundary distance: $D_b = \frac{1}{N} \sum d(r, boundary(R))$ where $r \in R$ be a point of R, and the mean distance is computed between the shape's internal points and the shape boundary points.
- Shape complexity measure: $f = A/D_b^2$ can be calculated using the relation between the region area and the mean boundary distance.
- Euler Number: $E = N_C - N_H$ is a scalar that specifies the number of connected components in a region N_C minus the number of holes N_H



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

Figure 4.6: Shape Features a) Input image b) Ratio of pixels in the region to pixels in the total bounding box, (square detector) c) Scalar specifying the proportion of the pixels in the convex hull that are also in the region. Computed as Area/ConvexArea. d) Eccentricity e) Aspect Ratio

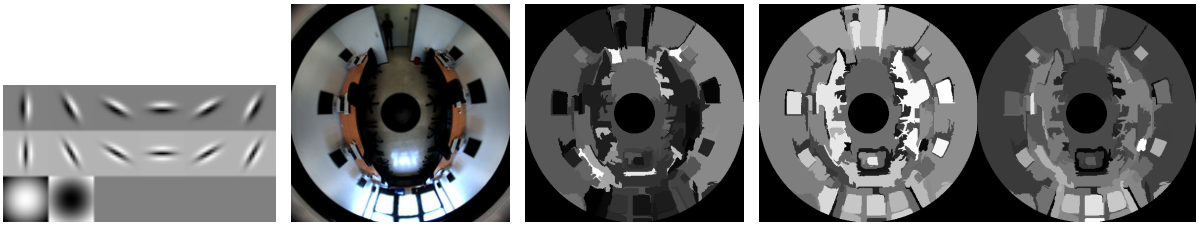
Boundary Descriptors

The boundary analysis [GW08; FCCJ09] starts with the contour of a region as a sequence of pixels $B = \{(x_k, y_k); 1 \leq k \leq K\}$ and the distance from each point to the centroid $D_k = \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2}$. The following boundary features can be extracted:

- Mean radial distance: $D_{mean} = \frac{1}{K}\sum_{k=1}^{K} D_k$
- Standard deviation of the radial distance: $\sigma_r = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(D_k - \mu_r)^2}$
- Minimum and maximum distance between the centroid and boundary points: $D_{min} = \min_k(D_k), D_{max} = \max_k(D_k)$
- Features based on the following relations: D_{max}/D_{min}, D_{max}/D_{mean}, D_{min}/D_{mean}
- Higher-order statistics: skewness and kurtosis

4.4.5 Texture

Texture features are captured by a subset of the maximum response set (MR8) filter banks [VZ05]. The original filter consists of a Gaussian and a Laplacian of Gaussian computed at six different orientations and three scales. Because of computational restrictions, our approach only considers a single scale ($\sigma_x = 1$, $\sigma_y = 3$) (See Fig. 4.7a). Additionally, instead of computing the maximum response for each scale, we capture the maximum response among all filters in a histogram and the mean absolute response of the filters per superpixel. Fig. 4.7 c-d, illustrates the output of three different histogram bins.



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

Figure 4.7: Texture features a) Subset of Maximum Response MR8 Filter Bank with $\sigma_x = 1$ and $\sigma_y = 3$ b) Input image c)-d) Visualization of three different histogram bins that captures the maximum respond among all filters.

4.5 Random Forest Classification

Randomized decision trees or Random Forests for classification[Bre01] consist of an ensemble of T classification trees trained with the bootstrap aggregation technique. With bootstrap, each tree is grown by sampling subsets of the training data with replacement. Additionally, correlated predictors are avoided in the training by picking the best split at each tree node from a subset of m randomly selected features without replacement.

Each tree is fully grown recursively from top to bottom, and majority voting makes a final prediction Y among the predictions of the ensemble. Alternatively, a posterior

probability for each tree $P(Y = c|n_l)$ can be learned from the training data by recording the ratio of the number of times a class c is assigned to a leaf node n_l, to the total number of examples that accumulate at that node.

$$P(Y = c) = \frac{1}{T} \sum_{t=1}^{T} P_t(Y = c|n_{l,t}) \tag{4.5.1}$$

4.6 Semantic Segmentation Results

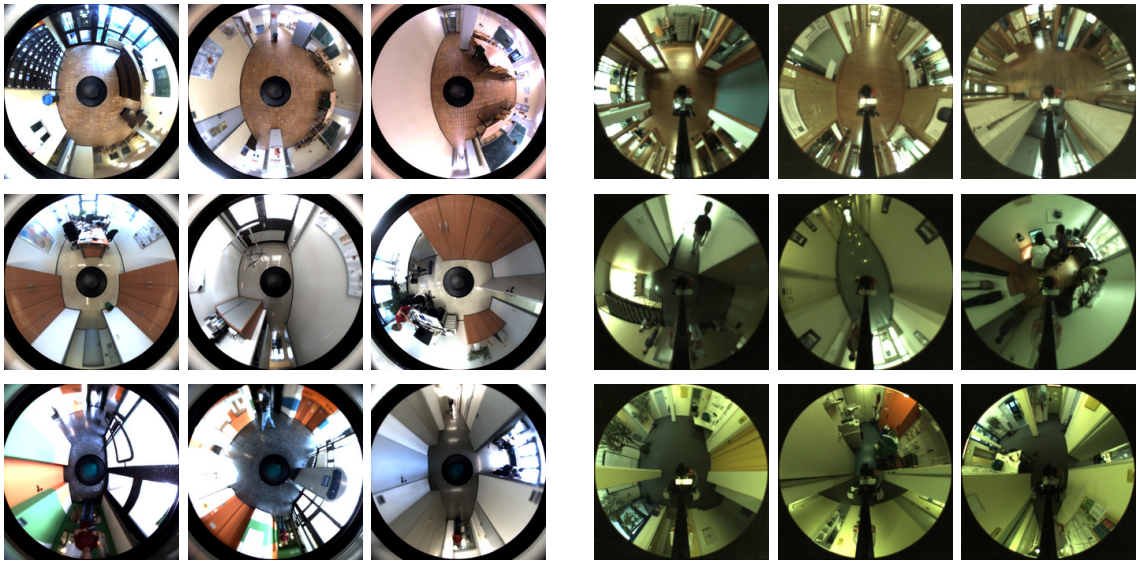
The segmentation algorithm is evaluated on two separated datasets. The first one is shown in Fig. 4.8a and consists of 75 images with their labeled ground truth segmentations [Pos+13]. The images were captured at different locations at the Technical University of Dortmund using the same camera configuration for all of them. In order to assess the generalization ability of the segmentation and to prevent artificial correlation in appearance between training and test images, the entire data set was split into three distinct scenarios (P: physics building, E: electrical engineering building and C: chemistry building). The exemplar scenarios are shown in Fig. 4.8a and correspond to different wings of the complex and exhibit different colors and texture in the floor, walls, and objects.

The second dataset (shown in Fig. 4.8b) consists of 100 manually labeled images captured at four different European cities and using different robot and sensor configurations. The objective of this dataset is to assess the generalization ability to different sensors and configurations, and unknown scenarios. Three of the cities are from the public available COLD dataset [PC09], which provides omnidirectional images from the indoor locations in the University of Freiburg, University of Ljubljana, and the German Research Center for Artificial Intelligence in Saarbrücken. The fourth location, not part of the COLD dataset, corresponds to the Technical University of Dortmund. Each location contributes with 25 manually labeled images.

Unfortunately, no dataset with such segmentations for omnidirectional images is publicly available for comparison or benchmarking. Performance is measured in both datasets pixel-wise with leave-one-out cross-validation. In the first data set, repeatedly, one out of three scenarios is used for testing and the other two datasets for training. In the second dataset, one of the cities is left for testing, while the remaining three are used for training.

Table 4.1 reports the confusion matrix of the tree classes for Dataset I. The overall classification rate of the approach is 86%, whereas Table 4.2 shows the results for Dataset II with an overall accuracy of 80%. The optimal number of trees were chosen with cross-validation and were 300 and 500 for Dataset I and Dataset II, respectively. These values can be seen in Fig. 4.9 by plotting accuracy as a function of the number of trees. Notice that an increase in the number of trees beyond 300 trees for Dataset I shows no further improvement, and no significant improvement is achieved with more than 500 trees for Dataset II.

According to Fig. 4.10, sampling beyond 4 predictors for splitting at each node of the trees does not achieve further performance improvement. Fig. 4.11 reveals the random forest out-of-bag estimates of the importance of the features according to its



(a) Dataset I (TU Dortmund) (b) Dataset II (COLD)

Figure 4.8: Example images from the dataset I and II. (a) Dataset I: Technical University of Dortmund. First row: physics building; Second row: electrical engineering building; Third row: chemistry building (b) Dataset II: Public available COLD dataset. First row: Freiburg; Second row: Ljubljana, Third row: Saarbrücken

	Floor	Vertical	Clutter
Floor	0.79	0.03	0.19
Vertical	0.02	0.92	0.06
Clutter	0.05	0.25	0.70

Table 4.1: Dataset I: Confusion Matrix

	Floor	Vertical	Clutter
Floor	0.77	0.10	0.13
Vertical	0.12	0.85	0.03
Clutter	0.17	0.04	0.79

Table 4.2: Dataset II: Confusion Matrix

mean decrease in accuracy after removing each group of features. Location represents the most important feature, followed by texture.

Several of the semantic segmentation of prototypical indoor scenes with their corresponding ground truths for Dataset I and Dataset II are illustrated on Fig. 4.12.

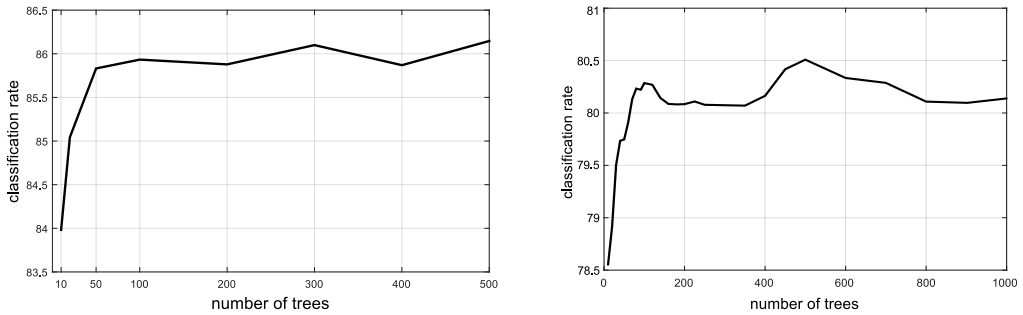


Figure 4.9: Random Forest parameters a) Classification accuracy versus number of trees in the random forest using Dataset I. b) Same as a) with Dataset II

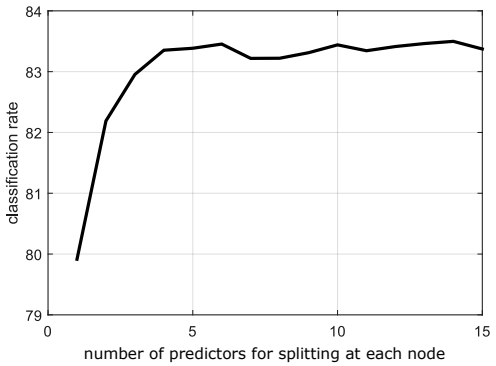


Figure 4.10: Number of predictors sampled for splitting at each node of the trees.

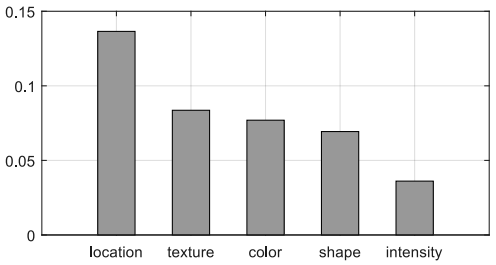
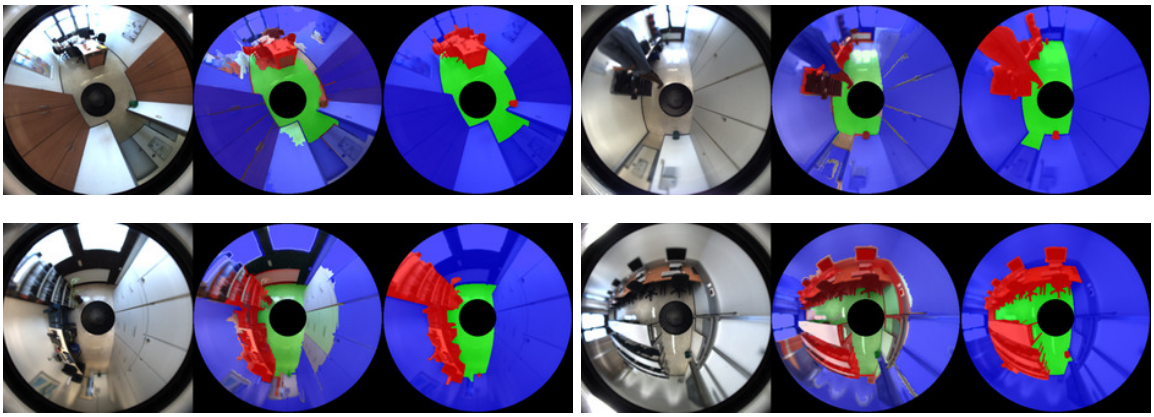
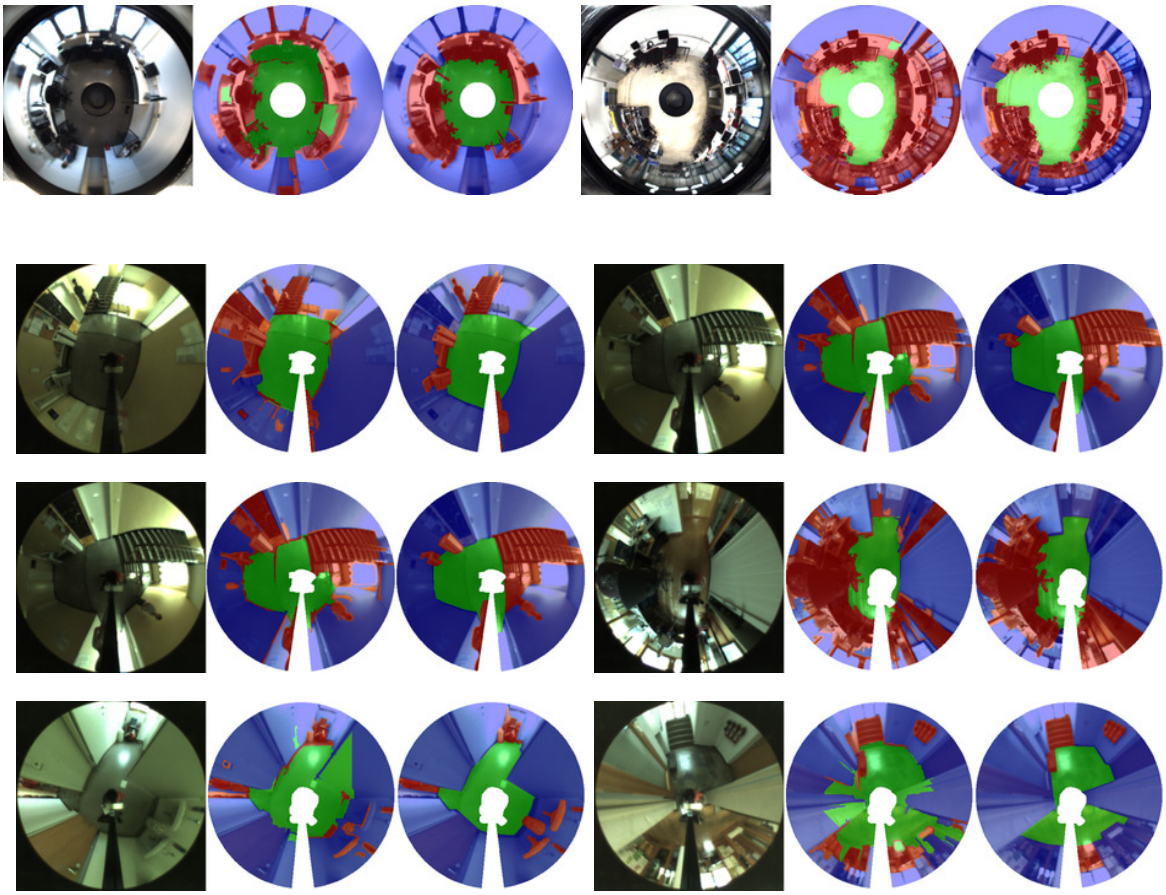


Figure 4.11: Mean decrease in accuracy after removing each group of features.



(a) Dataset I (TU Dortmund)

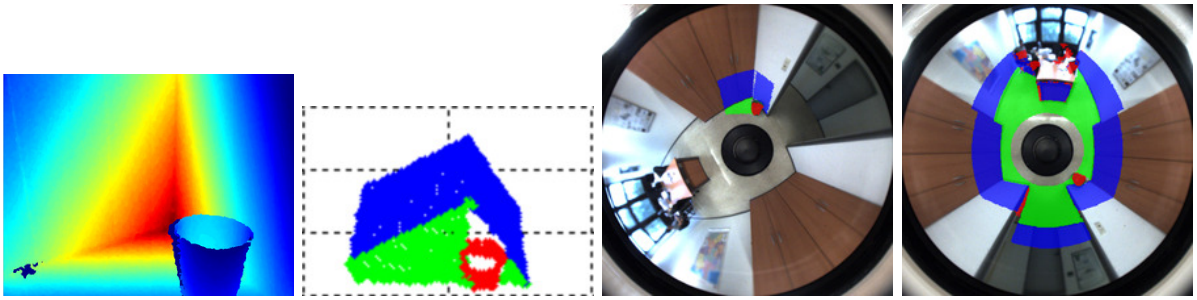


(b) Dataset II (COLD)

Figure 4.12: Semantic classification examples, green color represents floor, blue vertical structures and red obstacles or clutter. Left column: input image; Center column: Classification results (label opacity represents the posterior probability of the associated class); Right column: ground truth.

4.7 Self-Supervised Semantic Segmentation with Range Images

Similar to the system architecture presented in Chapter 2, we can devise a self-supervised system to train the semantic segmentation by labeling the training data using the 3D points from a range camera. The process is illustrated in Fig. 4.13. By rotating the robot 360° and registering the individual frame labeling, we obtain the mask for training the system shown in Fig. 4.13d.



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

Figure 4.13: Self-supervised semantic labeling using range data. a) Range image b) Segmentation of range data from the 3D points of the range image c) Omnidirectional image labeling by projecting the range labeling d) Output after registration frames from a 360° robot rotation

4.8 Summary

This chapter has presented a system that can provide a semantic segmentation of single image regions. The semantic segmentation aggregates multiple heterogeneous features extracted at the superpixel level with a Random Forest classifier and provide a posterior probability of the class. The image regions are classified into the categories: floor, vertical planar surfaces, and isolated objects (e.g. furniture).

The introduced semantic segmentation constitutes the basis for the layout reconstruction, local semantic, and map-based semantic navigation that will be introduced in the next chapters.

The framework is suitable for online self-supervised learning using range data from a time of flight camera. Future work can extend the current system with a larger number of object and scene categories and using a temporal fusion of the perceptions.

5

Surface Layout of Scenes

5.1 Introduction

Humans possess the remarkable ability to understand effortlessly complex scenes very fast [Oli14]. For example, by looking at the scene in Fig. 5.1, we can extract the structure and layout reliably —even by looking at it from different viewpoints. Nevertheless, for automatic systems, this task remains a challenge since the interpretation of a scene is complicated by the fact that 2D images provide no depth information, visual cues are highly context-dependent, and there is uncertainty from noise, occlusion, and illumination in real-world scenarios.

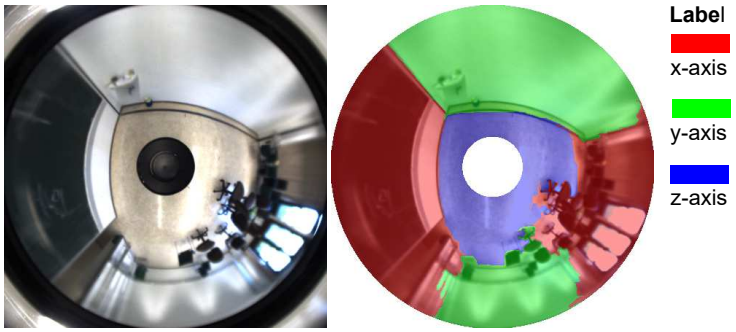


Figure 5.1: Surface Layout labeling of an indoor omnidirectional image. (left) input image, and (right) output of the classifier. Three labels are shown: red indicates structures with normal parallel to the x-axis of the robot coordinate system, green indicates structures aligned with normal corresponding to the y-axis, and blue are regions in the ground plane with normal aligned to the z-axis

Scene understanding through single images is a difficult problem due to ambiguity in the 3D visual reconstruction. An infinite number of 3D structures could project into the same 2D image. The advantage of analyzing human environments is that they are fairly structured. Thus, we can reason about how objects and surfaces are connected by exploiting prior knowledge and the regularities on those environments.

Similarly to humans, we want robotic systems to reason about the environment and execute advanced cognitive tasks such as recognition and navigation. A good starting point towards better scene understanding is to extract the spatial layout configuration. The object detection task becomes more accurate with evidence from the 3D layout and knowledge of the supporting structures and vice versa [SBS10].

While the literature reports several schemes for extracting the spatial layout from surfaces in the scene from monocular vision, very few tackle the problem using omnidirectional images. Existing methods for surface layout extraction with omnidirectional vision [OLNG11; OLNG12] focus on corners and line entities for the structure computation. However, difficulties arise in cluttered environments where those entities cannot be computed reliably. This motivates us in this chapter to address this problem using a different perspective. Our point of departure is statistical learning using a different set of heterogeneous features. The scheme first begins identifying vertical structures in the image with the free-space/vertical structure segmentation. Vertical components are further classified according to their orientation using orientation signals from histograms of oriented gradients, floor/wall boundary features, and oriented line features.

Important monocular cues such as size, location, color, and texture already exploited in the literature [DLN05; HEH07; HC12a] are still useful in omnidirectional systems [PHB14]. Nonetheless, important features for layout extraction, such as oriented lines and vanishing points, need to be computed differently, since the omnidirectional projection geometry distorts those entities, lines in the 3D space project into conics in the omni-image.

This chapter presents an approach that segments a scene into regions and assigns them a category according to their orientation [PVL16]. The region labeling distinguishes between three surface orientation classes. The surfaces are orthogonal to each other and labeled as illustrated in Fig. 5.1. The color red indicates structures with normals parallel to the x-axis of the robot coordinate system; green indicates structures with normals aligned to the y-axis; blue indicates structures with normals aligned to the z-axis (parallel to the ground plane).

5.2 Related Work

The majority of the literature in the surface and spatial layout domain has focused on monocular perspective vision [HEH07; YZM08; LHK09; SSN09; HHF09; HC12b]. Very few references address the omnidirectional vision counterpart [OLNG11; OLNG12]. The last authors exploit line structure in the scene for extracting the layout in indoor setups. Yet overcoming the problems induced from clutter remain an issue, since they hinder the corner extraction necessary in [OLNG11] and the floor-wall boundaries in [OLNG12]. Additionally, the predefined wall layout hypotheses in [OLNG12], limit the recovery of layout configurations outside the predefined set.

Line segment features in [YZM08; Lee+10] remain the main ones for surface layout recovery. The work in [YZM08] uses the geometric regularities in human-made indoor environments and a non-learning approach that finds depth-ordered planes along with dominant directions. The process groups the edges and subsequently fits lines into quadrilaterals based on co-planarity and relative depth.

The work in [LHK09] finds structure by generating the best fitting model to line segments. The approach identifies constraining corners (connections of walls), and several plausible building hypotheses are created and tested against an orientation map. Some difficulties arise in that not all lines or edges lie on a target structure and

floor-wall boundaries are often occluded. These are alleviated assuming symmetry between floor and ceiling, and floor-wall boundaries can be retrieved using the ceil-wall boundaries, which are rarely occluded.

The approaches in [HEH05] and [HEH07] use a larger set of cues such as color, texture, geometry, and position to learn appearance models of geometric classes: parallel to the ground, vertical structures, and sky. Vertical structures are further subdivided according to coarse orientation (left, center, or right/left) and according to the material (porous or solid).

Saxena *et al.* [SSN09] developed a system called Make3D that infers 3D structure from a single image. In contrast to [HEH07], Make3D does not make strong assumptions that scenes consist of the ground plane, vertical walls, or sky. Their system finds superpixels and infers for each of them, plane parameters that capture 3D location, and 3D orientation. Make3D builds upon the depth estimation method in [SSN07], incorporating region properties such as connectedness, co-planarity, and co-linearity into a probabilistic Markov Random Field (MRF) framework.

Delage *et al.* [DLN05] focus on a fully automatic 3D reconstruction of indoor environments from single images. The world is assumed to be Manhattan, meaning that planes are oriented in three orthogonal directions. The 3D ambiguous structure is learned using a Markov Random Field methodology similar to [SSN07], but instead of estimating depth, they estimate whether each point in the image represents a surface or an edge and its orientation.

Haines and Calway [HC12a] follow a different methodology for estimating planar structures. They use a bag of words (BoW) representation trained in urban environments to learn the relationship between appearance and plane structure. K-nearest neighbor determines whether or nor a region is planar and its orientation. Visual primitives in the BoW are computed with HOG extracted at salient points. The issue of different appearance for the same orientation is solved, reducing the dimensionality of histograms and creating a compact representation with the so-called topics in the Latent Semantic Analysis method.

5.3 Surface Layout Approach

The overall system architecture is shown in Fig. 5.2. There are four fundamental blocks: (i) The free-space segmentation block, which labels each image region as free-space/vertical structure and computes the floor boundary features, (ii) The vanishing points extraction and line classification block that assigns orientations according to the three-axis of the robocentric coordinate system, (iii) The histogram of oriented gradients block, which aggregates the gradient distribution per region, and (iv) The surface layout block, which combines all features and assigns an orientation to each vertical structure. The orientation classification is done similar to the previous chapter using Random Forests.

5.3.1 Free-space and Vertical Structures

The first process consists of segmenting the ground floor and the vertical structures using the processing block illustrated in the upper part of Fig. 5.2. The scheme

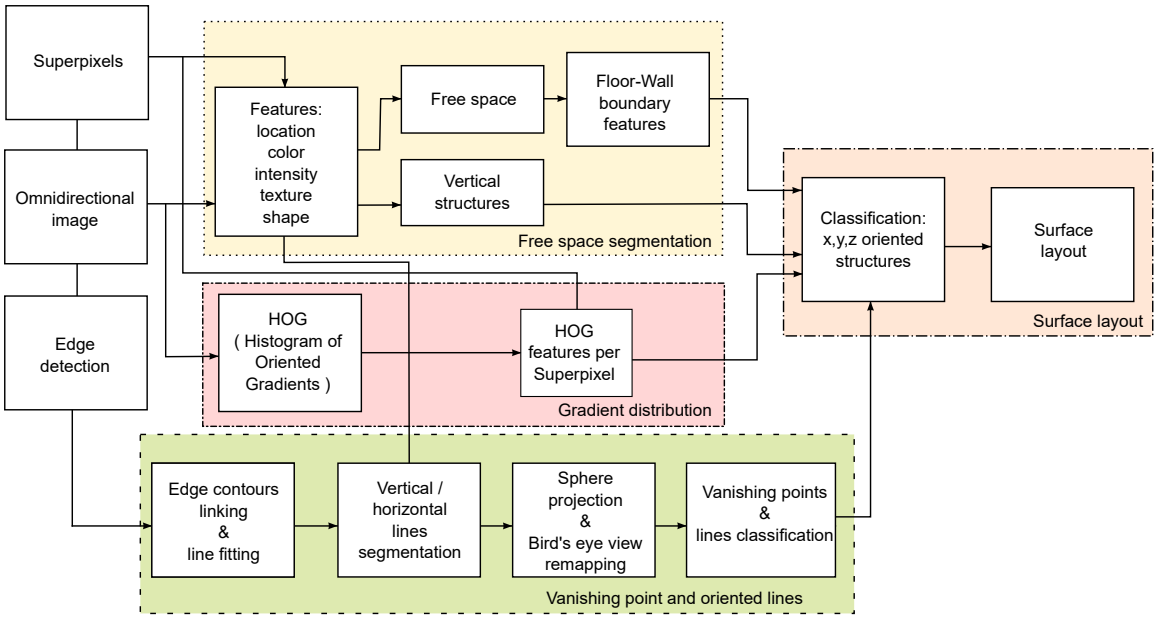


Figure 5.2: Surface layout system architecture

finds the free-space and vertical structures using the heterogeneous cues: location, color, texture, geometry, and shape. The segmentation method is described in detail in chapter 4. Another feasible floor/obstacle segmentation is explained in chapter 2. Vertical structures are all objects above the ground floor e.g. walls, doors, and clutter. The method extracts the layout of the scene assigning those vertical structures with orientations shown in Fig. 5.1.

5.3.2 Vanishing Points Estimation and Oriented Lines

The importance of vanishing points (VPs) for estimating the orientation of structures was already highlighted in the literature [YZM08; LHK09; HEH07; Cri01]. There are different strategies for extracting VPs in omnidirectional images [OLNG12; Baz+12; BP12]. Our approach is slightly different in that it uses a bird's-eye projection to obtain undistorted lines from the omni-image conics. Subsequently to this transformation, the RANSAC algorithm searches for lines imposing an orthogonality constrain.

Our VP extraction scheme makes the following assumptions: i) The sensor is a central omnidirectional system in which all optical rays intersect into a single effective viewpoint, ii) The camera axis is aligned with the ground normal, which is reasonable in most wheeled robots operating indoors, and iii) The world surfaces are assumed to have normals aligned vertically or horizontally following the Manhattan World assumption.

With the above assumptions, boundary lines located in a 3D plane correspond to vertical and horizontal lines in the omni-projection. Vertical lines project into radially distributed lines intersecting into one single point in the image. This point corresponds to the vertical vanishing point (VVP). In our omni-images, the VVP is located in the image center, since images are cropped into a symmetric square with center coincident with the single effective viewpoint.

The VPs of horizontal lines (HVP) cannot be extracted so straightforwardly, since

horizontal lines in the world project into conics in the omni-image given the nature of omni-mirror. Some methods search VPs in the image space, locating the intersection points of conics. Alternatively, the search can be done in a sphere space [Gey00], projecting into the sphere all image entities. Other approaches used the Hough Transform for clustering, with the disadvantage that each VP is found independently and no orthogonality constrain can be ensured. More recently, [Baz+12] clusters the lines with exhaustive search or uses RANSAC with models that ensure orthogonality in the VPs [BP12].

Our VPs extraction scheme models the camera using a sphere. However, it is different from [OLNG12], in that we additionally project the conics in the sphere into lines using a bird's-eye view remapping that removes the distortion (See Fig. 5.3). With the omnicamera aligned to the horizontal ground normal, the HVPs are located with the RANSAC method in the bird's-eye projection. This robust estimator is suitable given the large number of outliers and the necessity of finding several VPs iteratively. The main orthogonal directions are found using 2-line RANSAC, ensuring the orthogonality constraint.

Bird's-Eye View

The bird's-eye view necessary for the VPs computation consists of a scaled perspective projection of the image ground plane. The projection consists of mapping image points into a sphere and projecting them back into a bird's-eye perspective plane.

The uniform model in [SMS06] has the advantage of working with all types of cameras, including a fish-eye camera. This model can relate unit sphere points $[x_s, y_s, z_s]^T$ and the normalized image points $[x_m, y_m]^T$ with the equation:

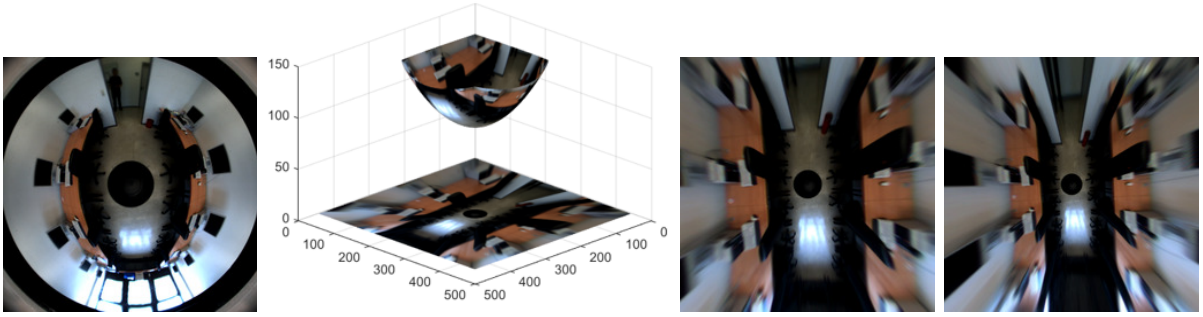
$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} x_m \\ y_m \\ a_0 + a_1\rho + a_2\rho^2 + \cdots + a_N\rho^N \end{bmatrix} \tag{5.3.1}$$

where $a_0 ... a_N$ are the polynomial calibration coefficients, and z_s becomes a function $f(\rho)$ with $\rho = \sqrt{x_m^2 + y_m^2}$. The equations above assume the camera sensor and mirror perfectly aligned, so the normalized image coordinates are proportional to scene points.

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \alpha \begin{bmatrix} x_s \\ y_s \end{bmatrix} \tag{5.3.2}$$

Notice that the constant α is not explicit in equation 5.3.1, since it can be included in $f(\rho)$. Fig. 5.3b shows a visualization example of a hyperbolic mirror camera bird's-eye projection after calibration with [SMS06].

The above formulation is useful if we have access to the omnidirectional camera calibration parameters. Thus, limiting the applicability of this formulation to images, e.g., in the public dataset COLD [PC09]. The COLD dataset provides no mirror parameter data or images capturing calibration patterns. Therefore, to obtain a bird's-eye view on those images, we can opt for a simpler model [RSV; Nay97]. The mirror projection is assumed to be single-viewpoint and orthographic. These assumptions



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

Figure 5.3: Bird's-eye projection a) Input Image b) Bird's-eye mapping into the sphere or paraboloid c) Resulting bird's-eye image with $z = 50$px d) Bird's-eye image with $z = 37$px.

approximate the mirror to a paraboloid with only one parameter: the radius of the paraboloid. Hence, mirror calibration can be virtually obviated.

The bird's-eye perspective projection is computed using polar and azimuthal angles:

$$\theta = \arccos \frac{z}{\sqrt{x_{be}^2 + y_{be}^2 + z_{be}^2}} \tag{5.3.3}$$

$$\phi = \arctan \frac{y_{be}}{x_{be}} \tag{5.3.4}$$

$$\rho = \frac{h}{1 + \cos\theta} \tag{5.3.5}$$

$$\begin{bmatrix} x_{omn} \\ y_{omn} \end{bmatrix} = \begin{bmatrix} \rho \sin\theta \cos\phi \\ \rho \sin\theta \sin\phi \end{bmatrix} \tag{5.3.6}$$

with h, the radius of the circle describing the 90° incidence angle on the omnidirectional camera effective viewpoint, and z defined by the distance between the effective viewpoint and the projection plane in pixels.

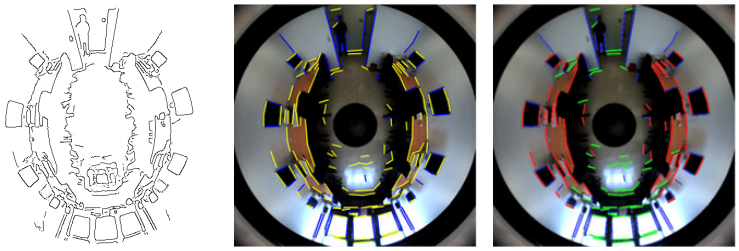
Figs. 5.3c-d show some examples of a bird's-eye projections using the polar azimuthal formulation and the effect of changing the height value.

Oriented Lines

Lines in the scene, which are oriented according to the main vanishing points, provide valuable information about the scene layout structure [YZM08; LHK09; HEH07].

The line extraction process is achieved by following two steps: (i) Edges are extracted with the canny detector and edge pixels are linked by contour following, and (ii) Contours are fit to line segments using the split and merge approach with the Iterative-End-Point-Fit algorithm [Ngu+05].

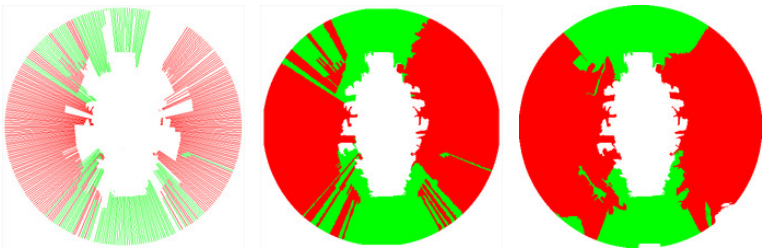
The resulting lines are classified into vertical and horizontal according to their orientation in the omni-view. Lines not belonging to those orientations or non-descriptive lines (e.g. too short) are simply discarded. Horizontal lines are further clustered according to the closest VP. The process is illustrated in Fig. 5.4a-c.



(a) (b) (c)

Figure 5.4: Oriented lines according to the main vanishing points a) Edge image with the Canny edge detector b) Classification as vertical and horizontal lines c) Lines oriented to the three main vanishing points. Red lines follow the x-axis, green the y-axis and blue the z-axis.

The next processing step consists of sampling the clustered oriented lines according to the main VPs, and extend them radially at steps with a fixed angular resolution (See Fig 5.5a). Image regions without line orientation information are extrapolated using the marker-based watershed algorithm [RM01]. Watershed fills all image regions, starting from a set of seed points and floods the landscape, which is represented by the image gradient. The result after watershed is shown in Fig. 5.5b.



(a) (b) (c)

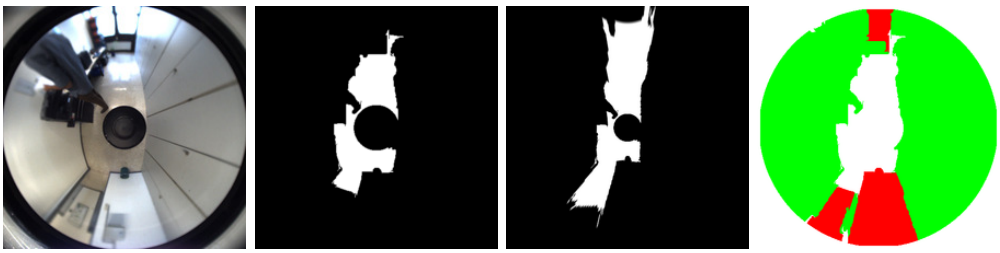
Figure 5.5: Orientation map feature from oriented lines according to the scene's vanishing points a) Each horizontal line is extended radially at a fixed angular resolution b) Watershed extends the map to all image regions c) Orientation map at the superpixel level.

The idea is to average the watershed extrapolation on the homogeneous superpixels regions. This way, we obtain a more robust statistical weighting of orientations in the homogeneous regions. This is illustrated in Fig. 5.5c, where some mistakes in the extrapolation were corrected.

5.3.3 Floor-Wall Boundary Features

The floor-wall boundary is an important feature for recovering the spatial layout in a scene. Indeed, the most important one for estimating shape hypotheses in [OLNG11; OLNG12]. However, the line extraction process to compute the floor/wall boundaries is in both methods hindered by clutter, noise, and lighting conditions. Our method follows a different approach. We start by segmenting the floor region and project the segmentation into the bird's-eye view. This way, lines in the 3D scene captured as conics in the omni-image get undistorted. The floor region in the bird's-eye view

roughly sketches the orientations of the planes in the scene. Fig. 5.6 a-c illustrates the process.



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

Figure 5.6: Orientation map feature computed with the floor wall boundary. a) Input omnidirectional image. b) Floor/Wall segmentation c) Bird's-eye view of the floor/wall segmentation. d) The orientation map using the floor/wall contour orientation aggregated per superpixel.

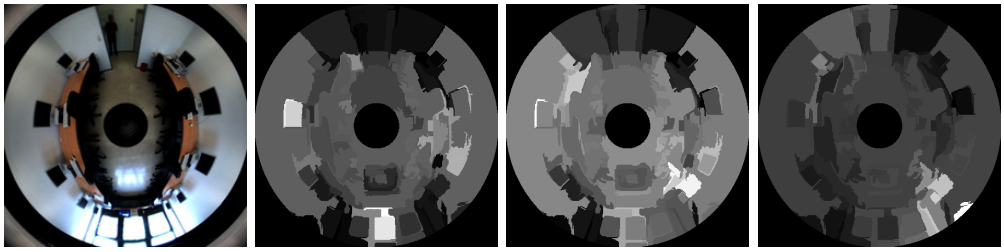
Since clutter generates undesired noisy lines in the boundary approximation, we apply morphological dilation followed by erosion to close gaps and remove noise in the segmentation image. The canny edge detector extracts the boundary contour. Subsequently, the orientation of vertical structures is computed from the obtained borderline. The process that follows is similar to the one explained in the previous section. Boundary-lines are extended radially by sampling the boundary at a fixed angular resolution. Marker-based watershed and the graph-based over-segmentation extend the orientation map to all regions and average the inconsistencies. The pivotal steps are illustrated in Fig. 5.6.

5.3.4 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) is a descriptor able to represent the shape and local appearance by the distribution of gradient orientation in localized portions of an image [DT05]. The gradient is computed by convolving the image with Prewitt kernels without smoothing: [-1 0 1] and $[-1\ 0\ 1]^T$. The image is divided into rectangular regions called cells, where gradient orientation occurrences are accumulated in a histogram of 9 bin orientation values. The unsigned version of HOG spread from 0 to 180°, while the signed spread from 0 to 360°. In this work, we use the unsigned version and weight the contribution of each pixel in the histogram with the value of the gradient itself.

The HOG descriptor, as demonstrated by [DT05] exhibits better invariance to illumination and contrast changes by normalizing the local response over larger spatial regions called blocks. The parameters 8 x 8 pixels per cell, 2 x 2 cells per block, and 9 orientations bins showed very good performance results in the original implementation and are adopted in this work using the publicly available code in the VLFeat open-source library [VF08].

Our HOG descriptors are computed densely for each of the cells and are normalized with the L2-norm of the block. The final HOG descriptor per superpixels gets contributions from all cell descriptors inside the superpixel region. They are normalized with area of the region to be more invariant to scale and size. Fig. 5.7 illustrates the response per superpixel at different orientations.



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

Figure 5.7: Histogram of Oriented Gradients feature. Angles are measured clockwise to the horizontal axis of the image a) Input image b)–d) Average response per superpixel at three random chosen angles (90°, 110°, 130°). Color white represents a high number of orientations coincident with the shown angle.

5.4 Experimental Results

The layout classification scheme is evaluated on 100 manually labeled images on four different locations in Europe. The first three locations are from the publicly available COLD dataset [PC09]. Each location provides 25 images in the following places: the University of Freiburg, the University of Ljubljana, and the German Research Center for Artificial Intelligence in Saarbrücken. The last location consists of 25 images captured at different scenarios at the Technical University of Dortmund.

Performance is measured pixel-wise with leave-one-out cross-validation, meaning, repeatedly one out of four scenarios is used for testing, and the other three for training. The overall classification performance achieved is 76% and the confusion matrix is reported in Table 5.1. The optimal number of trees in the Random Forest ensemble was found to be 500. Beyond this number, the accuracy slightly decreases to a saturation point close to 70%.

Table 5.1: Confusion matrix of the Random Forest classifier with all features

	plane-x	plane-y	plane-z
plane-x	0.75	0.22	0.03
plane-y	0.24	0.73	0.03
plane-z	0.09	0.08	0.83

Table 5.2 shows the effect on the performance after removing each group of orientation features. Removing HOG features decreases the overall accuracy by 2%. Removing the floor/wall boundary features decreases by 3%, the oriented line features by 6%, and removing all three groups of orientation features reduces the performance by 13%. Oriented lines represent the most important feature, followed by floor-wall boundary and HOG.

Fig. 5.8 illustrates some layout classification results at different prototypical indoor scenarios from the COLD dataset and TU Dortmund. The input image and the ground truth are shown as a reference. The classification is performed per superpixel. However, performance evaluation is measured pixel-wise.

Table 5.2: Classifier performance with different sets of features

Classifier + Feature	Accuracy x-axis / y-axis / z-axis / **overall**
RF all Features	0.75 / 0.73 / 0.83 / **0.76**
RF without HOG	0.74 / 0.71 / 0.82 / **0.74**
RF without Wall/Floor	0.74 / 0.71 / 0.80 / **0.73**
RF without Oriented Lines	0.67 / 0.69 / 0.82 / **0.70**
RF w/o (HOG,Wall/Floor,Lines)	0.57 / 0.63 / 0.81 / **0.63**

5.5 Summary

This chapter has presented a system that extracts the spatial layout and main orientation of surfaces in a scene captured with an omnidirectional camera. The system benefits from a cascaded classification, where free-space and vertical structures are first extracted. Subsequently, all vertical structures are analyzed to find their main orientation. The scheme aggregates multiple heterogeneous features computed at the superpixel level with a Random Forest classifier. The fusion of orientation features from the HOG transform, floor/wall boundaries, and oriented lines proved to be very effective in the layout classification. Future work is concerned with extending the number of orientation directions in the scene to go beyond orthogonal planes.

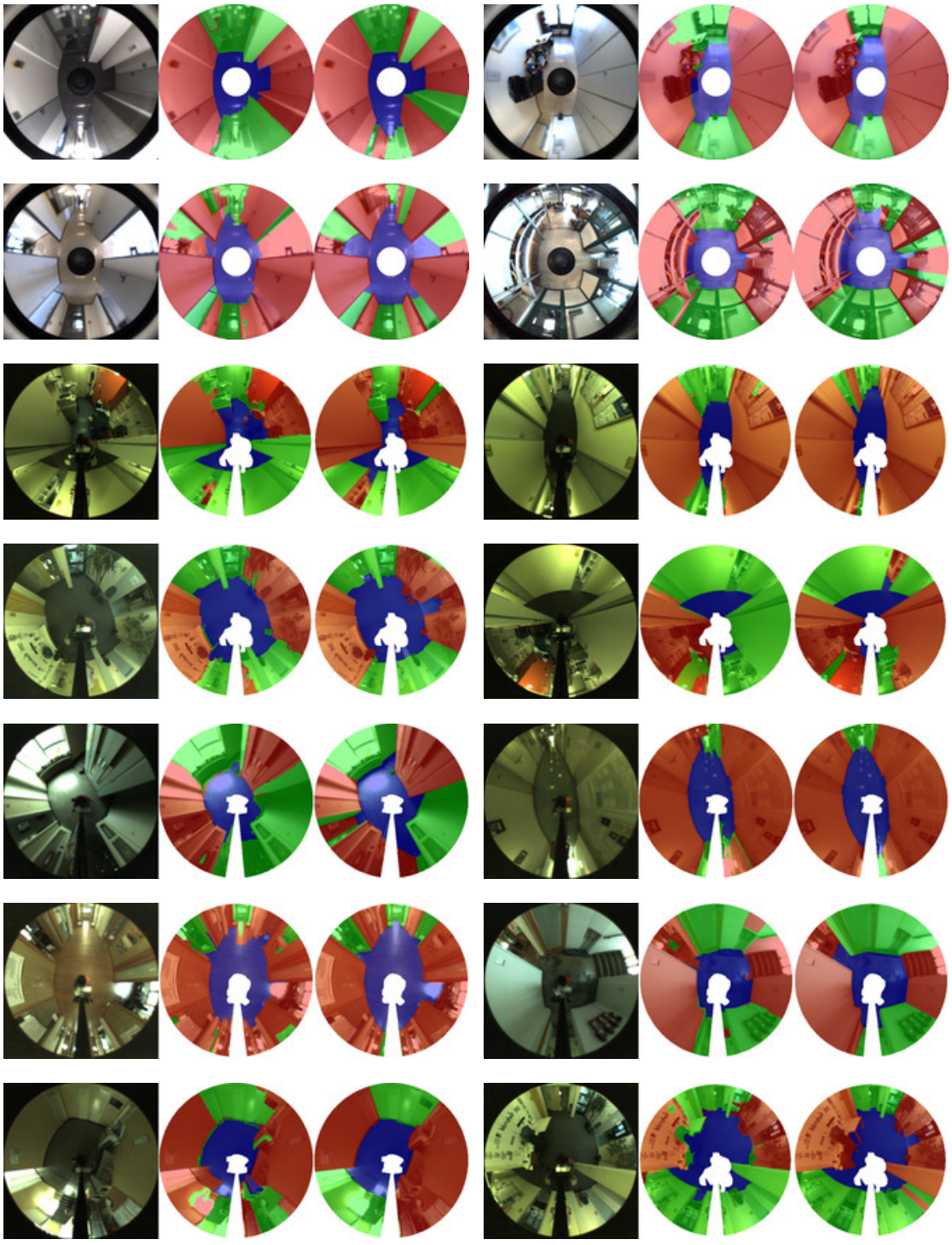


Figure 5.8: Surface layout examples from the TU Dortmund and the COLD Dataset. Three labels are shown: (red) corresponds to the x-axis; (green) y-axis, and (blue) z-axis. Left column is the input image; the center column is the classification results, and the right column corresponds to the ground truth. The top-two rows are examples from the TU Dortmund dataset, and the rest rows below illustrate examples from the COLD dataset.

6

Visual Place Category Recognition

6.1 Introduction

The ability of a robot to reason about the place it occupies and translate sensor readings into human concepts will allow the latter to perform more complex tasks and achieve natural human-robot communication. In this chapter, we investigate the translation of visual cues on omnidirectional images into place categories useful for robot navigation.

In recent years, vision has been increasingly used in robot navigation, since it allows better disambiguation of place categories compared to traditional range sensors. Geometric information extracted from laser rangefinders distance scans [MMSB05] allows only recognition of a limited number of categories. Vision apart of providing appearance information of both scene and its embedded objects [Rot+05] can be used as well as a range sensor to capture the free-space needed in local navigation and obstacle avoidance [Pos+11b].

In this chapter, we propose a place category recognition system using omnidirectional images to distinguish among four location categories: room, corridor, doorway, and open space. We do not seek to recognize all possible scene categories [Zho+14], but rather relevant navigation locations. Humans and animals are our inspiratio; they predominantly rely only on visual information to achieve complex navigation in unstructured environments. We seek to exploit the available visual cues to understand the place, tasks, and environment semantically. In this chapter, though, we concentrate only on place categorization. The results of this chapter will be used in local semantic navigation in chapter 7 and semantic map-based navigation in chapter 8.

Despite the discriminate power of vision, the visual recognition of indoor scenes remains a challenge given: i) the difficult information extraction from the complex raw image data; and (ii) the large intra-class variability in indoor places. Good discriminative appearance features that often perform good in outdoors (e.g. color, texture exhibit little regularity in indoor settings[QT09]).

We study different hand-engineered image representations with good classification performance in the literature, such as GIST, HOG, PHOW, dense SIFT, LBP, color histograms and SSIM. Base classifiers are trained using Support Vector Machines (SVM) and random forest (RF). Several ensemble methods are further studied with the idea of combining multiple decisions in a way that the overall decision making benefits from the mutual competences of heterogeneous individual classifiers. Analysis con-

firms that the combination of diverse classifiers achieves a classification rate that far exceeds the performance of any single classifier.

Additionally, our system also evaluates learned-features using the current technology of convolution neural networks (CNN) and deep learning [GBC16]. Deep learning techniques have advanced very fast, boosted with the era of massive parallelization in GPUs achieving an impressive number of records in the last years on very challenging datasets. All this comes with the drawback of requiring many training examples (sometimes in the order of millions) to fit the huge parameter space of deep neural architectures. Since our data is derived from the relatively small COLD dataset [PC09] and our set comprehends 130 examples of each of the categories: room, corridor, doorway, and open space, we cannot perform a full end-to-end network training. We, therefore, evaluate learned-features from already trained systems on other recognition tasks, such as the ImageNet [Den+09]. Recent studies [SR+14; Xiao2014; Simonyan2014; Ren2015] have shown that generic descriptors extracted from CNN trained in a specific task (e.g. object recognition), perform very well on other recognition tasks (in many cases outperforming hand-tuned features).

6.2 Related Work

One of the first attempts to extract semantic place information in robotics from laser rangefinders for semantic map annotation was presented in [MMSB05]. An extension of this work by [Rot+05] was able to disambiguate better similar place categories with the aid of object recognition within the scene using monocular vision. Hidden Markov models (HMM) have been integrated into various approaches [MMSB05; Rot+05; RMG12] to enhance classification accuracy by integrating spatial dependencies among multiple perceptions.

The majority of the literature body tackles methods with monocular vision. Only a few methods report results using omnidirectional vision [RMG12; Pos+13]. The approach [RMG12] considers a GIST image representation adapted to catadioptric sensors which achieve rotation invariance and HMM for better classification accuracy. The system is cascaded for first recognizing two main classes (places and transitions). Further processing recognizes the sub-classes: corridor, and room within places, and door and stairs from the place transition class. The system presented in this chapter is an enhancement of [Pos+13; Oel+14; PHB14] evaluating a larger set of features, including learned-from-data representations.

In the recent past, the computer vision community has achieved extraordinary progress on the general problem of scene recognition. Techniques such as global features and bag of words with local features, are highly effective in discriminating an impressive, diverse amount of scenes [OT01; LSP06; QT09] Object and scene recognition systems report excellent results using a variety of image features, such as GIST [OT01; QT09; RMG12], histogram of oriented gradients (HOG) [DT05], or densely extracted SIFT features and PHOW [LSP06; QT09].

With the recent advances in the GPU computing era, we have seen a rebirth of convolution neural networks (CNN) [LeC+98]. CNN have achieved an impressive number of records in the last years on very challenging object recognition datasets

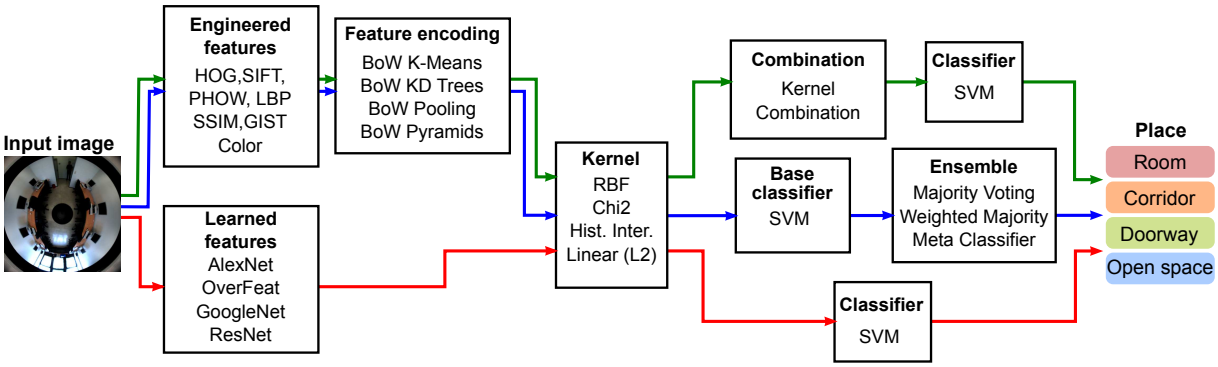


Figure 6.1: System architecture of the place category recognition system

such as the ImageNet [KSH12; Ser+13; SZ14; Sze+15; Ren+15]. The advances are mainly boosted by massive parallelization in GPUs, which allow the computation of very large layer architectures. The learned-features are general enough to be ported to other recognition tasks such as place categorization, by transfer learning or fine-tuning [SR+14; Xia+14; SZ14; Ren+15]. In this work, we evaluate some of those representations on omnidirectional images of places.

6.3 System Architecture

The overall system architecture is shown in Fig. 6.1. The upper part of the figure sketches the recognition using hand-engineered features. Several base classifiers are trained with SVM and their outputs are compared with Ensemble Based Systems [Pol06]. The fusion of features using kernel combination is also evaluated. The part below of the figure shows the recognition with CNN learned-features from the ImageNet dataset. We investigate: (i) whether they outperform the hand-crafted representations, (ii) serve as general purpose features able to generalize to omnidirectional images for place categorization. We are inspired by [SR+14; Xia+14; SZ14], which report good performance of ImageNet learned-features applied to a recognition domain different from the one from which they were derived. However, those studies use monocular cameras and do not consider the effects of the non-linear distortion of omnidirectional cameras, which often require rotation invariant features.

The place categorization is able to recognize the classes: room (R), corridor (C), doorway (D) and open room (O). Our approach classifies using single omnidirectional images rather than accumulate evidence over image sequences using hidden Markov models [RMG12]. In the next sections we describe the features implemented.

6.4 Engineered Visual Features

Our system evaluates several of the most prominent engineered visual features used in general object and scene recognition. Most engineered features try to emulate different aspects of human vision. We present them according to the attribute they emulate, such as: gradient, texture, color, and shape.

6.4.1 Gradient-Based Features

HOG: The histogram of oriented gradients feature [DT05], briefly described in the previous chapter, has achieved great success in many computer vision recognition tasks [LSP06; Fel+10; Kho+12; Xia+14]. HOG captures with 9 bin histograms, gradient orientations in localized portions of the image called cells. HOG achieves invariance to illumination and contrast changes by block normalization. A block consists of a grid of a certain number of cells. In this work, we investigate 2x2 and 3x3 cells per block with a different bag of visual words (BoW) techniques, such as k-means dictionary with the Elkan algorithm [Elk03], matching with 3 spatial pyramid matching [LSP06], k-d trees nearest neighbour search, and max-pooling matching from a spatial pyramid of 2 levels using Locality-Constrained Linear Coding (LLC) [Kho+12].

SIFT: The Scale Invariant Feature Transform (SIFT) [Low04] is a widely used gradient-based feature similar to HOG, but rather than computing gradient orientations at fixed grid locations, it finds keypoints on the image upon which their descriptors are computed. Keypoints are located on extrema over a scale-space search on Difference of Gaussians (DoG) images. DoG is an approximation of the costly to compute Laplacian of Gaussians (LoG) which, in practice behaves as a blob detector. DoG are computed at different pyramid scales, accounting for different blob sizes. Local maxima on DoG images is subsequently searched using pixelwise comparisons between each point and its neighbors as well as neighbors at nearby scales as to check whether the keypoint is better represented at the nearby scales. The SIFT algorithm implements outliers filtering using contrast thresholding and edge thresholding to reject keypoints with low contrast and edges, which exhibit a high DoG value.

The descriptor is computed in a 16x16 pixel neighborhood, which is divided in 4x4 sub-blocks of 4x4 pixels. An 8-bin histogram captures the orientation of each sub-block, thus achieving a 4x4x8 = 128 dimension descriptor.

In the same way, as with the HOG, we employ BoW with k-means dictionary, k-d tree, and pyramid matching with max-pooling and LLC.

Dense SIFT and PHOW: Dense SIFT is a fast computation version of SIFT which employs a flat rather than a Gaussian window. The descriptors are densely extracted at a fixed scale and orientation. The keypoints are sampled at 8 pixels step and with a spatial bin of 4 pixels wide. We compare grayscale dense SIFT versus PHOW [BZM07a], which is a dense SIFT variant extracted at multiple scales and with color information. We report results with 2 scales, using RGB and HSV color channels, and implementing different BoW techniques. Namely, nearest-neighbor search using a k-d tree for fast querying in high dimensional data, classical euclidean search, and max-pooling strategy from a spatial pyramid of 2 levels using LLC coding.

GIST: GIST is a descriptor specially crafted for scene recognition with the idea of capturing [OT01] the spatial structure of scenes. It uses a low dimensional representation with five perceptual dimensions, namely naturalness, openness, roughness, expansion, and ruggedness. In the implementation, dimensions are estimated with spectral and spatial information from Gabor filters in the form of steerable pyramids using 8 orientations and 4 scales. In order to make GIST represent global image properties, the local responses are averaged over 4x4 grids which lead to a 8x4x16 = 512 dimension feature.

6.4.2 Texture-Based Features

Local Binary Patterns: Local Binary Pattern (LBP) [OPH96] is a feature suitable for representing texture, faces, and materials. Since scenes in many situations are well represented by texture [RM04], it is no coincidence that LBP has shown to be useful in scene recognition [Xia+10]. LBP descriptor is constructed by comparison of pixels with their neighbors and recording the sign of differences. In an 8-neighborhood comparison, an 8-bit binary number is obtained. The sum of occurrences of the binary patterns within a window in a histogram leads to the LBP feature vector. In this work, we employ a rotation invariant and multiresolution LBP variant [OPM02], which enhances the original LBP idea by using a circular pattern neighborhood that can vary in size to account for scale. Each pattern is also rotated to the minimum value to achieve rotation invariance.

Local Binary Pattern Histogram Fourier: We also evaluate the variant Local Binary Pattern Histogram Fourier(LBP-HF) [Aho+09], to verify whether the better rotation normalization claimed by the method increases performance with our omni-images. Instead of normalizing rotation locally, LBP-HF computes from non-uniform LBP histograms, a global rotation invariant feature from the Fourier magnitude spectrum.

6.4.3 Color-Based Features

Color Histograms: Large scale scene recognition [Xia+14] has shown that color has a significant correlation with hundreds of different scenes categories. However, color histograms performance is well below gradient-based methods such as HOG or SIFT, since they only record global color composition. Thus, discarding important spatial information which is important to discern images belonging to different categories, but with a similar color distribution. The color histogram features evaluated in this study consist of joint color histograms of 8x8x8 = 512 bins using RGB color space. We evaluate also HSV joint histograms of 10x10x5 = 500 bins and L*a*b with 5x10x10 = 500 bins.

Color Name Feature: We also extract Color Name features [VDW+09], which are more localized models than color histograms to check whether the spatial information increases performance improvement in our problem. Color Name Features use 8,12, and 16-pixel patches with 2 pyramid levels and 4-pixel spacing. We use a dictionary of 256 images and 50 colors. The code implementation is publically available [Kho+12]. Color names are automatically learned from images collected on the Internet using Probabilistic Latent Semantic Analysis (PLSA). This technique allows us to overcome noisy query results in the image retrieval domain and to improve performance.

6.4.4 Shape-Based Features

Self Similarity Images (SSIM): The SSIM descriptor [SI07] computes the internal layout of local self-similarities in images. SSIM can match images having a similar shape and spatial layout while being robust to large appearance variations in color, texture, photometric properties, edges, and small local affine deformations. The descriptor measures internal self-similarity by comparing patches (typically 5x5 pixels) with a larger surrounding (region of 40 pixels radius). The sum of squared differences (SSD)

between patch colors in CIE L*a*b space, leads to a correlation map, which is further discretized using a log-polar representation with 4-radial intervals and 20 angles. The descriptor thus has 80 values and records the maximum value inside each bin. The final processing consists of a $[0-1]$ range normalization. We compute SSIM descriptors with the code found in [CPZ09], which uses a BoW to collect all per patch descriptor occurrences in a histogram before a visual-word mapping. The visual vocabulary is built with k-means clustering with thousands of SSIM descriptors from a large set of images.

6.5 Learned Visual Features

In the previous section, we presented some of the most important hand-engineered features that dominated the last decade in many visual recognition tasks. In the recent past, after the publication of the AlexNet [KSH12] and the advent of the GPU computation era, computer vision has been revolutionized to achieve human-like and above performance in object, place, activity recognition to mention a few.

Learned-features, in contrast, to hand-engineered ones, are derived automatically from data using Convolutional Neural Networks (CNN) [LeC+98]. Some examples are [KSH12] and more recently [Ren+15], which are trained using millions of images from the ImageNet dataset [Den+09].

Convolutional networks, opposed to traditional multilayer perceptron networks, have sparse connectivity linking neurons locally between adjacent layers. Thus, receptive field, parameters, size, and memory are drastically reduced, allowing for inputs of raw pixels from larger images and avoiding prohibitive huge training sets to fit the several order of magnitude parameter space of the densely connected counterparts. CNNs also have the advantage of not discarding the image topology, thus, render them ideal for grid-like data structures such as images. Thorough convolutions (instead of matrix multiplications of traditional networks), parameters are shared per layer.

A non-linearity and pooling on top of the convolutional layer, make them very robust to image variations, such as scale, rotation, distortions. CNN are trained with back-propagation which leads to a feature extractor of increasing order as the layer gets deeper. The learned-features have shown to be general enough to beat state-of-the-art hand-tuned features in visual tasks different from the one they were trained [SR+14; Xia+14; SZ14; Ren+15].

In the following, we summarize and evaluate some of the most significant CNN architectures found in the literature.

AlexNet: AlexNet proposed by Krizhevsky *et al.* [KSH12] is a convolutional neural network trained with 1.2 million images from the ImageNet Dataset [Den+09] to classify 1000 object categories. The network consists of 8 layers: The first 5 are convolutional and the last 3 are fully connected. AlexNet implements in its convolutional layers Rectified Linear Units (ReLU), which converge faster than *tanh*, and in some layers, contrast normalization and max-pooling. Refer to [KSH12] for details. Dropout technique, which randomly drop units and their connections, is used in the first two fully-connected layers. Dropout serves as a mechanism for avoiding over-fitting. The

fully connected layers have 4096 neurons. AlexNet with its 60 million parameters and 650000 neurons, requires different data augmentation to prevent over-fitting. The training data is increased by a factor of 2048 using random cropping, horizontal flipping, and RGB intensities jittering.

CaffeNet: The Caffe reference model (CaffeNet) [Jia+14] is basically the same AlexNet architecture. However, with max-pooling and normalization layers interchanged, and training is not augmented with RGB intensities jittering.

OverFeat: The OverFeat network [Ser+13] has a similar layer architecture to the AlexNet with the following differences: the first and second layers have a larger dimension due to stride of 2 pixels instead of 4. However, max-pooling does not overlap. OverFeat also avoids contrast normalization and trains with crop variations and horizontal flips but no intensity jittering.

Zeiler & Fergus: The network proposed by Zeiler & Fergus [ZF14] is a modification of the AlexNet. They improve it by changing the 11x11 filters to 7x7 in the first layer, and by reducing the stride from 4 to 2. With these changes, the network can reduce aliasing artifacts, and retain more information in the first and second layer features.

VGGNet: VGGNet was one of the first very deep convolutional networks proposed by Simonyan *et al.* [SZ14] achieving state of the art object recognition in the ImageNet challenge. One of the key ideas of VGGNet was to reduce the filter size to 3x3 and pooling to 2x2 , while increasing deep to 16-19 layers. This way, the number of parameters remain similar to a shallower net. The features of the VGGNet recorded an excellent performance on other recognition tasks and generalized well on a wide range of datasets. The end classifier was trained with SVM and without fine-tuning. This network was inspired by AlexNet and was outfitted with ReLU units and without local normalization.

GoogLeNet: GoogLeNet [Sze+15] is a network of 22 layers depth designed to increase depth and width while keeping computations constant. The use of 1x1, 3x3, and 5x5 filters and reducing spatial resolution drastically in the first layers allows it to decrease computations. Thus, GoogLeNet can achieve a deeper layer-architecture. The training follows a similar procedure to AlexNet with 1.2M images from the ImageNet dataset. However, data augmentation additionally to cropping jittering included aspect ratio and photometric distortions.

ResNet: ResNet [Ren+15] is a very deep neural network with 152 layers (8 times larger than VGGNet and 7 times GoogLeNet). ResNet was able to achieve such large depth architecture by implementing learning residual functions that reference the input layers. This learning technique prevents a degradation problem found on large unreferenced architectures. There exist several ResNet models with 51, 101, and 152 layers, with each having increasing accuracy.

6.6 Classification

We investigate linear Support Vector Machines (SVM) and kernelized SVM classifiers. The multi-class SVM is handled with one-vs-all training.

Combination and fusion of base predictors (trained individually with the features above), is also considered to evaluate the performance increase of ensemble-based

systems [Pol06], which idea is to produce a stronger classifier. We consider simple non-learning methods such as: majority voting and weighted voting up to more advanced ensemble methods with meta-classifiers and kernels fusion. We conduct several experiments to conclude which combinations and techniques are more appropriate to our problem.

6.6.1 Support Vector Machines

Support Vector Machines (SVM) [CV95] are called large margin binary classifiers since they learn a hyperplane, which separates classes with the largest margin to the nearest training sample of any class. Given training data in the form of pairs: $\{(\mathbf{x}_i,\ y_i\ );\ i = 1, ..., n\}$ with $y_i \in \{-1, +1\}$, $\mathbf{x}_i \in \mathcal{R}^n$, the goal is to find the hyperplane (assuming separable data) with larger margin, following the constraint:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \tag{6.6.1}$$

where the margin $\frac{2}{\|\mathbf{w}\|}$ is found to be maximum by solving the following optimization problem:

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2, \\ \text{s.t.} \quad &y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \end{aligned}$$

In the case of non-linearly separable data, constraints cannot be satisfied; therefore a soft margin term is introduced,

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i, \\ \text{s.t.} \quad &y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

The positive slack variables $\xi > 0$ and the regularization parameter C relax the constraints by introducing a further cost with the following behavior:

- $\xi_i = 0$ the point is on the correct margin boundary
- $0 < \xi_i \leq 1$ the point is between the margin and the correct side of the hyperplane. (margin violation)
- $\xi_i > 1$ the point lies on the wrong side of the boundary (misclassified).

While the parameter C tries to find a compromise between two conflicting objectives: maximizing the margin and minimizing the number of errors on the training data. A larger C makes constraints hard to ignore; therefore the margin gets narrow, and the sum of error term dominates. A smaller C allows the constraints to be ignored, and the margin gets large since the sum of errors term is negligible.

The optimization with a soft margin can be solved with Lagrange multipliers.

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_i \alpha_i (y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i)) \tag{6.6.2}$$

By setting the partial derivatives of the Lagrangian $\frac{\partial \mathcal{L}}{\partial w_i} = 0$, $\frac{\partial \mathcal{L}}{\partial b_i} = 0$, and $\frac{\partial \mathcal{L}}{\partial \xi_i} = 0$, the problem can be written in its dual form:

$$\begin{aligned} &\max \sum_i \alpha_i - \frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ &\text{s.t.} \quad \sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \end{aligned}$$

solving for $\boldsymbol{\alpha}$, $\mathbf{w}$ becomes,

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i, \tag{6.6.3}$$

and the classifier function can be written as a sum over support vectors:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \tag{6.6.4}$$

Support vectors are vectors $\mathbf{x}_i$ for which $\alpha_i > 0$. The function $f(\cdot)$ is conveniently expressed in term of dot products $\mathbf{x}_i^T\mathbf{x}$, which can be replaced with kernels (see next section). The classifier decision function for prediction is $\text{sign}(f(\mathbf{x}))$.

Nonlinear SVM

SVM can be extended beyond linear models using kernel functions. By mapping the input features with a non-linear function $\phi(\mathbf{x})$, the classifier in equation 6.6.4 becomes:

$$\sum_i \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{6.6.5}$$

where K is a kernel function defined as:

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \tag{6.6.6}$$

and maps feature vectors onto a higher dimensional space with the idea that on the transformed space, the hyperplane separation might become non-linear. The mapping $\phi(\mathbf{x})$ can be very expensive to compute since it can be a very high dimensional vector. However, with kernels, only the inexpensive inner product embedded in K need to be calculated, avoiding the explicit mapping. This shortcut is the so-called kernel trick in the literature.

So that kernels can be expressed as inner products in high dimensional spaces while keeping computational complexity almost unchanged, they must satisfy the Mercer's condition. This condition states that a kernel must be a symmetric function, and the kernel matrix must be positive semidefinite.

In this work, we evaluate the following kernels:

$$K_{lin}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' \tag{6.6.7}$$

$$K_{rbf}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \tag{6.6.8}$$

$$K_{\chi^2}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{n} \frac{2x_i x_i'}{(x_i + x_i')} \tag{6.6.9}$$

$$K_{e\chi^2}(\mathbf{x}, \mathbf{x}') = \exp(-\mu D_{\chi^2}(\mathbf{x}, \mathbf{x}')) \tag{6.6.10}$$

$$K_{hi}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{N} \min(x_i, x_i') \tag{6.6.11}$$

K_{lin} is the lineal kernel, K_{rbf} is the radial basis function kernel (Gaussian), K_{χ^2} is the chi-square distance kernel, $K_{e\chi^2}$ is the exponential chi-square version, and K_{hi} is the histogram intersection kernel.

Equation 6.6.9 is a positive semi-definite chi-square version [VZ12]. While 6.6.10 uses the conventional D_{χ^2} distance formulation.

$$D_{\chi^2}(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \sum_{i=1}^{n} \frac{(x_i - x_i')^2}{(x_i + x_i')} \tag{6.6.12}$$

However, in order to incorporate D_{χ^2} into the SVM, an extended Gaussian kernel [Zha+07] is used and denoted as $K_{e\chi^2}$.

We use the publicly available library code for Support Vector Machines LIBSVM [CL11]. This implementation provides an efficient implementation and Kernel support.

Multiclass SVM

SVM are binary classifiers and need to be extended for solving the multi-class problem. In this work, we use the one-vs-all technique, where K classifiers are trained with a total K number of classes. Each classifier separates a class k of the training examples from the rest of the classes. While predicting the output $\hat{y}$ of an unseen sample $\mathbf{x}$, all classifiers are presented with $\mathbf{x}$, and the larger output score is chosen as the prediction:

$$\hat{y} = \underset{k \in \{1 \ldots K\}}{\operatorname{argmax}} f_k(\mathbf{x}) \tag{6.6.13}$$

In this chapter, we investigate different ensemble strategies. We start with kernel combinations, followed by stacked generalization in which a meta-classifier is trained with the score outputs of the base classifiers. We also investigate simple non-learning rules to check whether performance is similar to the meta-classifiers, which require an extra learning stage.

Kernel Combination

Kernel combination consists of a weighted combination of individual kernels based on accuracy:

$$K_{all} = w_1 K_{Sift} + w_2 K_{Hog} + w_3 K_{Gist} + ... + w_N K_N \tag{6.6.14}$$

We evaluate different kernel weighting strategies for determining w_i ($i = 1..N$). The simplest approach is to normalize according to the accuracy determined thorough cross-validation

$$w_i = \frac{\text{accuracy}_{K_i}}{\sum_{j=1}^{N} \text{accuracy}_{K_j}} \tag{6.6.15}$$

The second approach employs the following heuristic [GA11]:

$$w_i = \frac{\text{accuracy}_{K_i} - \delta}{\sum_{j=1}^{N} (\text{accuracy}_{K_j} - \delta)} \tag{6.6.16}$$

where the accuracy of each kernel is compared with a threshold δ. We set δ to the lowest kernel accuracy, thereby, decreasing the effect of low-performance kernels.

The third weighting we evaluate is proposed by Xiao *et al.* [Xia+14], where the weighting is the accuracy with respect to the median of the accuracies weighted to the fourth power

$$w_i = \left(\frac{\text{accuracy}_{K_i}}{\text{median}(\{\text{accuracy}_{K_1}, \text{accuracy}_{K_2}, \ldots, \text{accuracy}_{K_N}\})} \right)^4 \tag{6.6.17}$$

Non Learning Combinations

Majority Voting: There are different variants of majority voting classifiers (e.g. unanimous voting, simple-majority, plurality voting) [Pol06]. Unanimous voting requires all voting to agree. Simple-majority needs at least one more than half the number of votes. In this work, we consider plurality voting (or simply majority voting), where the chosen class is the one that receives the largest number of votes.

$$C(x) = \text{mode}\{h_1(x), h_2(x), \ldots, h_k(x)\} \tag{6.6.18}$$

Weighted Majority Voting: Knowing that some base classifiers are more accurate, we can weight their decisions to try improve overall performance,

$$C(x) = \underset{i}{\text{argmax}} \sum_{j=1}^{C} w_j I(h_j(x) = i) \tag{6.6.19}$$

where $I(\cdot)$ is the indicator function. Similar to the kernel combination (equation 6.6.16), cross-validation accuracy is employed as a weighting factor. By normalizing the weights, we can ensure they add up to one.

6.7 Place Recognition Results

The dataset for this study consists of 520 images with ground truth labels. There are 130 images of each of the classes: corridor, room, doorway, and open space (See

Fig. 6.2). The images were selected at different building locations from the COLD database [PC09] and from the Technical University of Dortmund. Special care was taken to choose instances with minimum overlap among the scenes; thus, evaluating better the true generalization ability of the system as training and testing are less correlated.

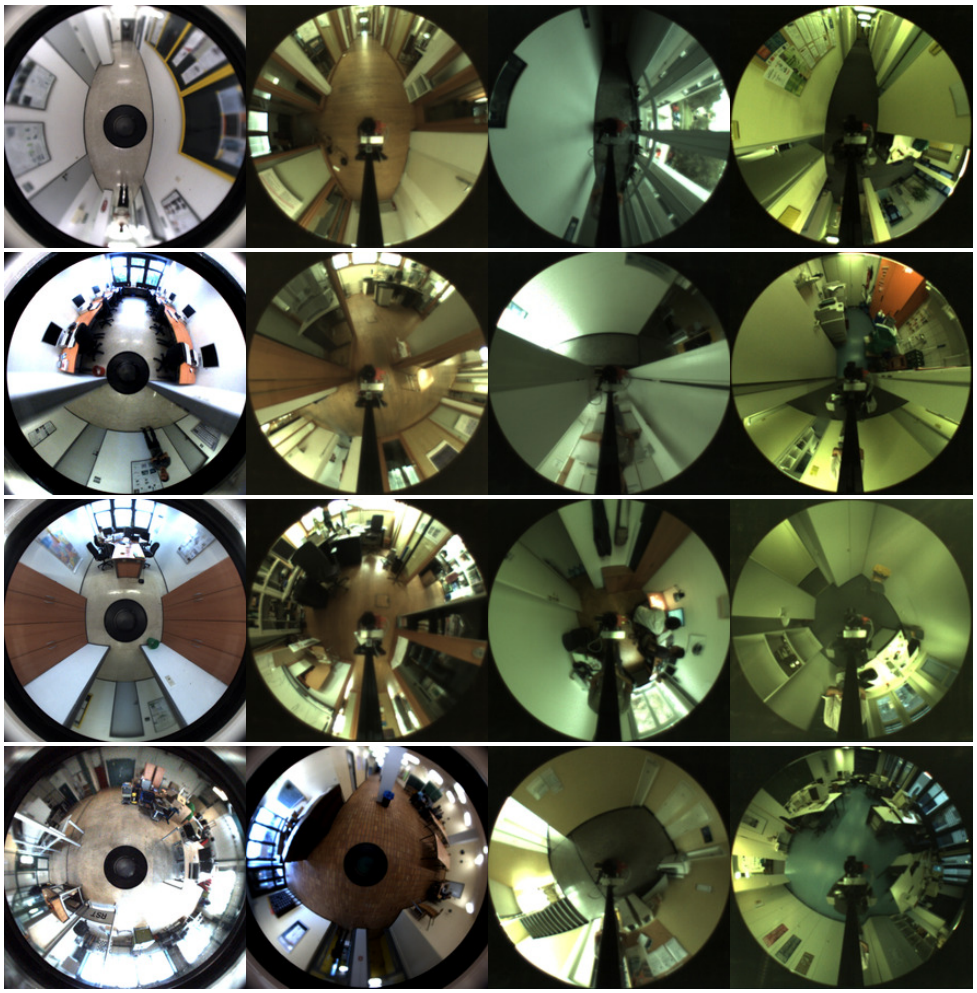


Figure 6.2: Examples of images of Dortmund (first column) and COLD dataset (Freiburg, Saarbrucken, Ljubljana). The first row shows examples of the corridor category; second row: doorway; third row: room; and the last row: open space.

The distribution of classes was equally extracted from the locations: Freiburg, Saarbrücken, Ljubljana, and TU Dortmund. The exception was the open-space category, which had very few examples in the COLD dataset. Therefore, 80% of open-space data comes from TU-Dortmund images.

The examples were randomly partitioned into 50% training and 50% testing with the holdout method five times. Results report the average classification accuracy over these five repetitions. To avoid the over-fitting in such a small dataset, especially with feature vectors of large dimensions, we augment the training data with more images. Each image was rotated 4 times 45°, and was mirrored horizontally and vertically, thereby, enlarging the training data 6 times.

Table 6.1 summarizes the classification performance of the base classifiers using

SVM for each of the engineered features and the corresponding best performing kernel. Table 6.2 is shown for comparison purposes and evaluates the same features under linear SVMs. PHOW features perform best, achieving 88% overall accuracy with a kernelized SVM and 85% using a linear SVM. In general, Kernelized SVM outperforms linear SVM in the majority of experiments. DenseSIFT and HOG3x3 features are close below PHOW. The results show that gradient-based methods dominate the upper positions (PHOW, HOG, SIFT).

GIST is the only gradient technique without BoW, and 86% accuracy is very close to HOG3x3 with BoW. Texture features (LBP) are in the middle of the ranking and also do not employ the BoW framework. SSIM close to 82% accuracy provides an alternative information source to gradient.

Color (between 55-75% accuracy) is well below the other features and does not provide as a good representation for places. Using different variants of BoW does not seem to have a contribution as we notice that the classical vector quantization with k-means achieves very similar results to the k-d tree representation.

The combination of base-classifiers with engineered-features with the non-learning combinations: majority voting, weighted voting, and the median is summarized in Table 6.3. Accuracy improves in all tested combinations compared to single base classifiers. It scales up to 89.9% combining different base classifiers and features. Weighted majority has slightly better results compared to majority voting. Median combination performs slightly below voting strategies.

Table 6.4 shows the results of SVM with Kernel combinations. Kernel combination achieves the best accuracy of all systems evaluated with an overall accuracy of 90.8% combining the features PHOW, denseSIFT, HOG, and GIST. We do not notice a significant improvement by using any particular weighting formula. The Kernel weighting raised to the fourth power in equation 6.6.17 leads to almost the same results as with equations: 6.6.15, and 6.6.16.

The learned-from-data features evaluation is summarized in Table 6.5. We notice that these general-purpose features exhibit good generalization in the experiments, although they were learned in a completely different recognition task setup. The best learned-feature is the ResNet with 152 layers architecture and achieves 80.4 % accuracy. Thus, it is comparable with denseSIFT using a linear SVM. In general, the learned-features in this experiment perform below the engineered features. The reason can be partly attributed to the fact that engineered features were tuned using omnidirectional images of the same recognition task, whereas the learning in the learned-features was carried out with monocular images and using a different recognition task (object recognition). We expect the performance of the learned-features to increase significantly by fine-tuning them with a larger set of omnidirectional images and using examples of place category recognition.

6.8 Summary

This chapter has presented a place category recognition system able to classify the locations: room, corridor, doorway, and open-space in omnidirectional images. The developed recognition system is pivotal in more advanced semantic-based robot nav-

Table 6.1: Kernelized SVM Classifiers with Best Performance per Feature

Name	Feature	Kernel	Accuracy in % C | D | O | R | **Overall**)
C1	PHOW Grayscale + BoW with k-d tree	$K_{e\chi^2}$	91.4 | 85.2 | 88.0 | 88.6 | **88.3**
C2	PHOW HSV + BoW with L2 distance	$K_{e\chi^2}$	86.8 | 81.9 | 91.7 | 90.8 | **87.7**
C3	PHOW RGB + BoW with L2 distance	$K_{e\chi^2}$	90.5 | 83.1 | 86.8 | 89.5 | **87.5**
C4	Dense SIFT + BoW with L2 distance	$K_{e\chi^2}$	88.6 | 80.6 | 88.6 | 90.5 | **87.1**
C5	HOG 3x3 + BoW with Max Pooling + LLC	K_{rbf}	88.6 | 81.2 | 89.8 | 87.1 | **86.7**
C6	GIST	K_{rbf}	84.6 | 84.9 | 86.2 | 88.3 | **86.0**
C7	HOG 2x2 + BoW with L2 distance	$K_{e\chi^2}$	88.0 | 80.3 | 86.8 | 88.0 | **85.8**
C8	Sparse SIFT + BoW with L2 distance	K_{hi}	86.5 | 79.4 | 85.5 | 85.8 | **84.3**
C9	LBP	K_{rbf}	88.3 | 80.0 | 83.1 | 82.5 | **83.5**
C10	LBP Fourier	K_{rbf}	88.3 | 79.7 | 82.8 | 83.1 | **83.5**
C11	SSIM	K_{rbf}	82.8 | 79.7 | 81.5 | 83.1 | **81.8**
C12	Color Name 256	K_{rbf}	76.3 | 62.5 | 86.5 | 76.6 | **75.5**
C13	Color Name 64	K_{rbf}	74.5 | 61.8 | 79.4 | 73.5 | **72.3**
C14	Color Histogram RGB	$K_{e\chi^2}$	74.5 | 53.2 | 80.0 | 72.0 | **69.9**
C15	Color Histogram HSV	$K_{e\chi^2}$	69.2 | 54.2 | 78.8 | 66.5 | **67.2**
C16	Color Histogram L*a*b	$K_{e\chi^2}$	52.3 | 43.7 | 69.5 | 53.2 | **54.7**

Table 6.2: Linear SVM Classifiers with Best Performance per Feature

Name	Feature	Kernel	Accuracy in % C | D | O | R | **Overall**)
L1	PHOW Grayscale + BoW with k-d tree	K_{lin}	88.3 | 85.8 | 81.2 | 87.1 | **85.6**
L2	PHOW HSV + BoW with k-d tree	K_{lin}	85.2 | 79.4 | 86.8 | 90.2 | **85.4**
L3	PHOW RGB + BoW with k-d tree	K_{lin}	88.3 | 82.2 | 82.2 | 88.3 | **85.2**
L4	HOG 3x3 + BoW with Max Pooling + LLC	K_{lin}	85.2 | 78.8 | 84.6 | 83.7 | **83.1**
L5	HOG 2x2 + BoW with L2 distance	K_{lin}	85.8 | 77.2 | 83.7 | 82.2 | **82.2**
L6	Sparse SIFT + BoW with L2 distance	K_{lin}	82.5 | 79.1 | 83.4 | 83.7 | **82.2**
L7	Dense SIFT + BoW with L2 distance	K_{lin}	83.4 | 74.5 | 83.1 | 82.2 | **80.8**
L8	SSIM	K_{lin}	78.8 | 80.0 | 75.7 | 79.1 | **78.4**
L9	LBP	K_{lin}	84.3 | 70.8 | 72.9 | 77.2 | **76.3**
L10	GIST	K_{lin}	76.0 | 73.2 | 67.4 | 76.3 | **73.2**
L11	LBP Fourier	K_{lin}	75.7 | 67.4 | 77.8 | 68.6 | **72.4**
L12	Color Name 256	K_{lin}	71.4 | 53.2 | 80.9 | 66.8 | **68.1**
L13	Color Name 64	K_{lin}	60.0 | 48.0 | 74.2 | 58.5 | **60.2**
L14	Color Histogram RGB	K_{lin}	72.6 | 17.2 | 72.0 | 64.3 | **56.5**
L15	Color Histogram HSV	K_{lin}	62.5 | 21.5 | 65.5 | 60.0 | **52.4**
L16	Color Histogram L*a*b	K_{lin}	46.5 | 35.4 | 50.8 | 50.8 | **45.8**

Table 6.3: Ensemble Performance Results with Non-Learning Combinations

Name	**Classifier**	**Combination**	**Accuracy in %** C | D | O | R | **Overall**)
NL1	PHOW(C1) + Dense SIFT(C4)	Median	83.7 | 84.6 | 84.9 | 88.0 | **85.3**
		Majority	96.3 | 81.5 | 89.5 | 86.5 | **88.5**
		Weighted Majority	91.4 | 85.2 | 88.0 | 88.6 | **88.3**
NL2	PHOW(C1) + Dense SIFT(C4) + HOG(C5)	Median	91.1 | 84.3 | 91.4 | 89.8 | **89.2**
		Majority	92.6 | 83.4 | 90.8 | 89.8 | **89.2**
		Weighted Majority	92.3 | 84.0 | 91.1 | 89.8 | **89.3**
NL3	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6)	Median	87.4 | 86.2 | 90.8 | 88.9 | **88.3**
		Majority	94.8 | 84.9 | 91.4 | 88.3 | **89.8**
		Weighted Majority	93.5 | 84.9 | 90.8 | 89.2 | **89.6**
NL4	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6) + LBP(C9)	Median	90.8 | 86.5 | 91.4 | 89.5 | **89.5**
		Majority	94.5 | 85.2 | 90.2 | 89.5 | **89.8**
		Weighted Majority	93.5 | 85.2 | 90.8 | 89.8 | **89.8**
NL5	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6) + LBP(C9) + SSIM(C11)	Median	88.3 | 85.8 | 89.8 | 88.3 | **88.1**
		Majority	94.8 | 84.9 | 90.8 | 88.3 | **89.7**
		Weighted Majority	94.2 | 84.9 | 91.1 | 89.5 | **89.9**
NL6	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6) + LBP(C9) + SSIM(C11) + COLOR(C12)	Median	89.8 | 86.8 | 92.0 | 88.9 | **89.4**
		Majority	92.6 | 84.3 | 91.1 | 89.2 | **89.3**
		Weighted Majority	92.6 | 84.3 | 91.1 | 89.5 | **89.4**
NL7	Best 5: C1 + C2 + C3 + C4 + C5	Median	91.1 | 85.5 | 91.4 | 90.2 | **89.5**
		Majority	92.0 | 84.6 | 90.5 | 90.2 | **89.3**
		Weighted Majority	91.4 | 84.9 | 90.5 | 90.2 | **89.2**
NL8	All Kernels: C1 + C2 + ... + C16	Median	87.4 | 86.5 | 91.4 | 90.2 | **88.8**
		Majority	91.1 | 83.1 | 90.5 | 91.1 | **88.9**
		Weighted Majority	91.1 | 83.1 | 90.5 | 91.1 | **88.9**

Table 6.4: Kernel Combination Performance Results

Name	Kernels	Combination	Accuracy in % C | D | O | R | Overall)
KC1	PHOW(C1) + Dense SIFT(C4)	Eq. 6.6.15	88.3 | 84.0 | 90.5 | 90.8 | **88.4**
		Eq. 6.6.16	91.4 | 85.2 | 88.0 | 88.6 | **88.3**
		Eq. 6.6.17	88.6 | 84.0 | 90.2 | 90.8 | **88.4**
KC2	PHOW(C1) + Dense SIFT(C4) + HOG(C5)	Eq. 6.6.15	88.9 | 84.6 | 91.1 | 92.0 | **89.2**
		Eq. 6.6.16	89.2 | 85.2 | 88.3 | 90.8 | **88.4**
		Eq. 6.6.17	89.2 | 84.6 | 91.1 | 92.0 | **89.2**
KC3	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6)	Eq. 6.6.15	92.0 | 87.7 | 91.7 | 92.0 | **90.8**
		Eq. 6.6.16	89.5 | 83.7 | 91.1 | 91.4 | **88.9**
		Eq. 6.6.17	92.0 | 87.7 | 91.7 | 91.4 | **90.7**
KC4	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6) + LBP(C9)	Eq. 6.6.15	92.3 | 86.8 | 90.8 | 91.1 | **90.2**
		Eq. 6.6.16	92.0 | 87.1 | 92.3 | 91.1 | **90.6**
		Eq. 6.6.17	92.3 | 86.8 | 90.8 | 91.1 | **90.2**
KC5	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6) + LBP(C9) + SSIM(C11)	Eq. 6.6.15	92.3 | 86.5 | 92.3 | 91.4 | **90.6**
		Eq. 6.6.16	92.6 | 86.5 | 91.1 | 90.8 | **90.2**
		Eq. 6.6.17	92.3 | 86.5 | 92.3 | 91.4 | **90.6**
KC6	PHOW(C1) + Dense SIFT(C4) + HOG(C5) + GIST(C6) + LBP(C9) + SSIM(C11) + COLOR(C12)	Eq. 6.6.15	91.7 | 84.6 | 91.4 | 90.5 | **89.5**
		Eq. 6.6.16	92.3 | 86.5 | 92.0 | 91.4 | **90.5**
		Eq. 6.6.17	92.3 | 85.8 | 91.4 | 90.5 | **90.0**
KC7	Best 5: (C1 + C2 + C3 + C4 + C5)	Eq. 6.6.15	89.5 | 83.1 | 90.8 | 90.8 | **88.5**
		Eq. 6.6.16	89.8 | 84.0 | 89.8 | 91.1 | **88.7**
		Eq. 6.6.17	89.5 | 83.4 | 90.8 | 90.8 | **88.6**
KC8	All Kernels: C1 + C2 + ... + C16	Eq. 6.6.15	87.4 | 80.6 | 90.2 | 90.2 | **87.1**
		Eq. 6.6.16	89.2 | 84.9 | 92.0 | 91.7 | **89.5**
		Eq. 6.6.17	88.3 | 84.3 | 92.3 | 91.4 | **89.1**

Table 6.5: Performance Results with Learned-features

Name	Feature	Kernel	Accuracy in % C | D | O | R | **Overall**)
CNN1	ResNet 152	K_{rbf}	80.6 | 84.9 | 80.3 | 75.7 | **80.4**
CNN2	CaffeNet	K_{rbf}	81.8 | 69.5 | 78.5 | 80.0 | **77.5**
CNN3	ResNet 50	K_{kl2}	80.9 | 73.2 | 81.2 | 72.9 | **77.1**
CNN4	AlexNet	K_{rbf}	81.2 | 67.4 | 77.8 | 77.8 | **76.1**
CNN5	ResNet 101	K_{rbf}	75.1 | 78.2 | 77.8 | 72.3 | **75.8**
CNN6	GoogLeNet	K_{rbf}	80.6 | 69.8 | 72.3 | 75.1 | **74.5**
CNN7	Zeiler & Fergus	K_{kl1}	80.3 | 66.8 | 65.8 | 76.6 | **72.4**
CNN8	OverFeat	K_{rbf}	71.4 | 75.7 | 77.2 | 60.3 | **71.2**
CNN9	VGGNet	K_{kl1}	75.1 | 64.3 | 68.3 | 74.8 | **70.6**

igation. As we shall see in the following chapters, the place category information will allow the robot to build a semantic map and perform a more complex local navigation, which exploits the place context. In this chapter, we also evaluated several hand-engineered features and learned-from-data representations. For our particular task with such a small omnidirectional images dataset, the engineered features showed better results. However, the potential of learned-features is promising, since they were able to generalize to omni-images , although they were learned in a completely different task and using monocular images. Similar to previous chapters, the place category recognition also benefits from the combination of heterogeneous features. We analyzed non-learning and learning combinations. The latter performed better by including an extra learning state.

7

Behavior-Based Local Semantic Navigation

7.1 Introduction

Mobile robots are being shifted very quickly from controlled setups in industries to unstructured human environments to perform progressively more complex tasks. In fact, service robotics represents the fastest growing market in the robotics sector with new exciting applications emerging every year in diverse areas such as inspection, surveillance, assistance, rescue, and entertainment. Nevertheless, fully autonomous robots dealing with complex human-robot interaction still require profound semantic understanding to comprehend human navigation task descriptions in natural language (e.g. follow the corridor and enter the room at the right doorway) and better understand the world and its enclosed objects.

Autonomous navigation is one of the fundamental capabilities that a robot must achieve. It involves global navigation (localization and path planning), and local navigation (for collision avoidance). Proximity sensors such as laser range finders have been used effectively over the past two decades to perform both types of navigation. However, understanding the environment semantically from range is limited to recognizing a few semantic categories [MMSB05]. This limitation motivates the use of visual data. In this work, we employ a wide field-of-view sensor that provides an omnidirectional camera. Similar to a range laser finder, it captures the environment globally and invariant under robot's rotation, but provides a much richer semantic information.

In this chapter, we propose a framework for local robot navigation (i.e. navigation not involving maps) based on visual behaviors that extract the information from omnidirectional vision. In semantic navigation, rather than navigating with mere metric pose instructions, robots require much more complex recognition and navigation capabilities. This type of navigation is inspired by humans, where places are not described in terms of coordinates in a global map, but with semantic data. The current robot's place is given a semantic category (e.g. room, corridor), while regions in the scene are also labeled with semantic regions (e.g. door, wall, floor). The advantage of using this representation is that the system can recover from errors occurring from conflicts while activating behaviors in the current context. For example, activating a

corridor-centering behavior inside a room.

The local semantic navigation scheme allows the robot to reach its targets by activating behaviors that can be parsed directly from natural language (e.g. get out of the room). Starting points and goals are described by places, rather than by poses. This kind of local navigation requires no prior map knowledge but relies on different visual recognition modules such as place categorization, semantic segmentation, free-space detection, and door detection. Each module assists in different basic visual navigation behaviors.

7.2 Related Work

The literature reports several schemes to extract semantic information from visual data in mobile robotics [KG15]. Most systems either recognize places semantically [Rot+05; RMG12; Sün+16] or label portions of the image with semantic labels [NH08; SBB12]. Nevertheless, very few systems present a coherent navigation framework integrating the available semantic data.

In the context of map-based semantic navigation, [Tse+05] proposes a hybrid navigation system for humans that combines classical geometric navigation with semantic information using an ontology for modeling environmental concepts, rules, relationships, and preferences of users. The authors in [URD11] propose an environmental model for robot navigation that can handle metric, topological, and semantic features for performing planning and reachability analysis given start and end-points (called semantic positions). The proposed ontology enables the robot to reach the positions using a graph, whose edges translate to robot actions.

Within the framework of local navigation, the autonomous city explorer robot [Bau+09] is a very successful system that could travel about 1.5km in downtown Munich, finding its way without prior map or GPS signals merely by asking pedestrians for heading information. The city explorer robot shares the focus on reliable and safe local navigation between intermediate navigation points with our system. Our system differs in that the start and end-points are described by semantic places (as in [URD11]), rather than following directions pointed out by pedestrians.

Similar to our system that is based on visual behaviors, Narayanan *et al.* propose to learn the behaviors using supervised learning to draw the relationships between perception and the geometry of the environment [Nar+10; Nar+11a; Nar+12; Nar+13]. The learning performance by additionally classifying the scenario as corridor, open space, and cluttered environments, and learn more specific behaviors per current scenario is discussed in [Nar+11b].

In an effort to interpret human language and apply corresponding robot's motor actions, the work on [Mat+13] describes from the natural language perspective how to parse the commands, and their system can translate English commands into sequences of desired actions. While their system operates with good results on simulation, the handling of complex data from real robots is still open to research and needs verification.

7.3 System Architecture

The overall system architecture of the local semantic navigation, partly inspired by [PHB14], is illustrated in Fig. 7.1. The system is divided into four modules. The *place category recognition* module presented in chapter 6 assigns the current robot's location with four place labels: room, corridor, doorway, and open space. Place categorization plays an important role in our system since it gives the robot context information for activating the behaviors and allows the robot to recover from conflicting behaviors.

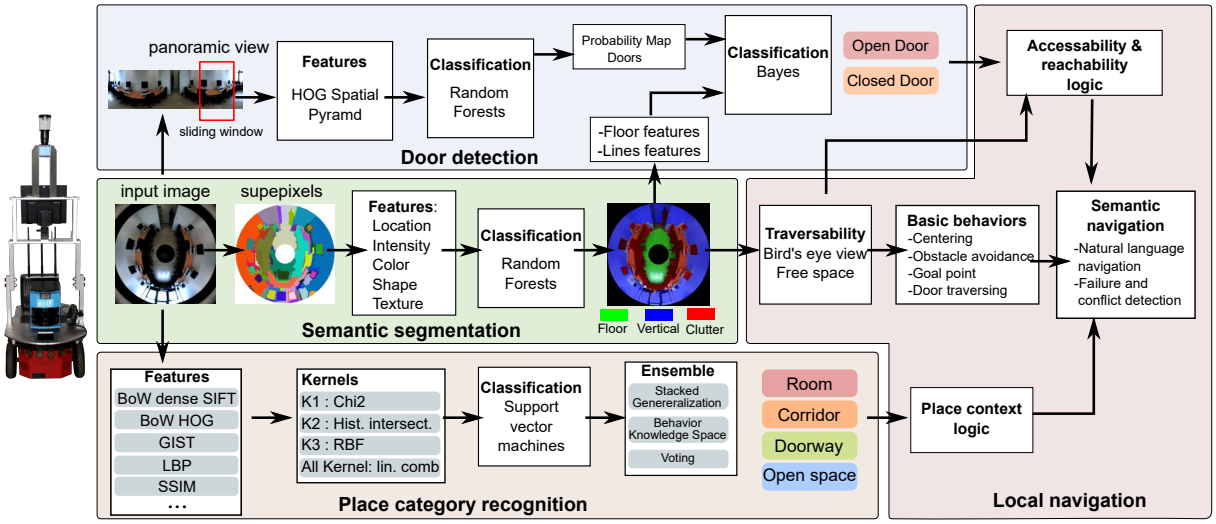


Figure 7.1: Local navigation system architecture

The *semantic segmentation* module from chapter 4 identifies relevant navigation objects and regions in the image such as floor, walls, vertical structures, and obstacles in the local environment.

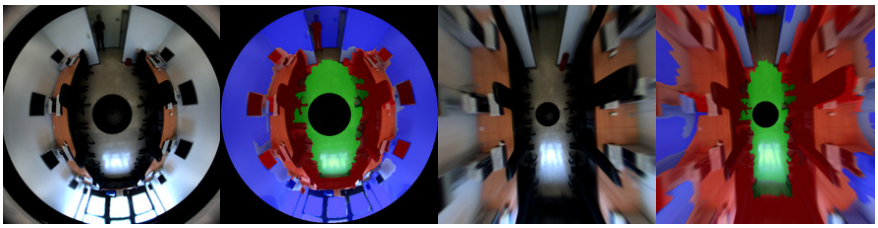
Doors are very important objects in indoor navigation since they are located at transition points between places. The *door recognition* module can identify doors in the image and detect whether they are open or closed. Doors cannot be readily extracted from the semantic segmentation module since they often overlap with other classes in the segmentation (floor and walls). To counter that, the door detection employs the bounding box representation, which in combination with a sliding window, outputs a probability map of door locations on a constructed panorama. As opposed to an omniview, where doors are radially distributed in the image; in the panorama, all doors are oriented vertically. Thus, this avoids using specially crafted rotation invariant descriptors as in [RMG12].

The *local navigation* module is composed of different navigation behaviors. The high-level behaviors (e.g. get out/ enter the room) are constituted by several basic behaviors such as goal point, obstacle avoidance, and door traversing which are activated sequentially with behavior arbitration. The semantic navigation gives the robot context for activating the behaviors. The system checks the start point and endpoint context. This logic allows the system to decide whether it is feasible or not the activation. This way the system can avoid activating, for example, a corridor following behavior when no corridor is present or a door-passing behavior without a doorway.

Section 7.6 describes the behaviors in detail.

7.4 Semantic Segmentation

The *semantic segmentation* module follows the approach presented in chapter 4. Figures 7.2 a-b show the output of the segmentation. The images are divided into superpixels where features: location, color, texture, shape, and line features are computed and are the inputs to the semantic classifier. The Random Forest classification method predicts with high accuracy the classes: ground region, clutter, and planar structures illustrated in the figure with labels: green, red, and blue, respectively.



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

Figure 7.2: Semantic segmentation and Bird's-eye mapping a) Input image b) Semantic segmentation c)-d) Corresponding bird's-eye views

The omnidirectional image contains a substantial amount of distortion given the nature of the hyperbolic mirror. To better assess the traversability of the terrain, distortions in the free-space image are corrected with the bird's-eye mapping presented in chapter 5 (See Fig. 7.2 c-d. The free-space in the bird's eye projection corresponds to a scaled version of the real metric scenario. Thereby, distances to obstacles can be measured conveniently with Euclidean distance.

7.5 Door Detection

The *door detection* module is based on the histogram of oriented gradients (HOG) descriptor [DT05]. HOG (explained in chapter 6) records the distribution of local orientations in the gradient image and is useful for object detection where an object's appearance is predominately described by shape. HOG is robust to background noise, illumination changes and clutter. In this work, we use the spatial pyramid extension [BZM07b] using three pyramid levels. Since its bounding box can well represent a door, we search in a constructed panorama image with a fixed-width sliding window from left to right. The panorama image increases the classification performance since HOG is not rotation invariant. The descriptors computed at each pyramid level are stacked and inputted to a random forest classifier using a probabilistic output learned from the training. Furthermore, we train a meta-classifier for determining whether the door is open or closed. This meta-classifier includes additional features that better capture the door surface. We compute mean, variance, entropy, solidity, MR8 texture and floor continuity (See chapter 4 for more details). An output example of the door detection classifier is shown in Fig.7.3.

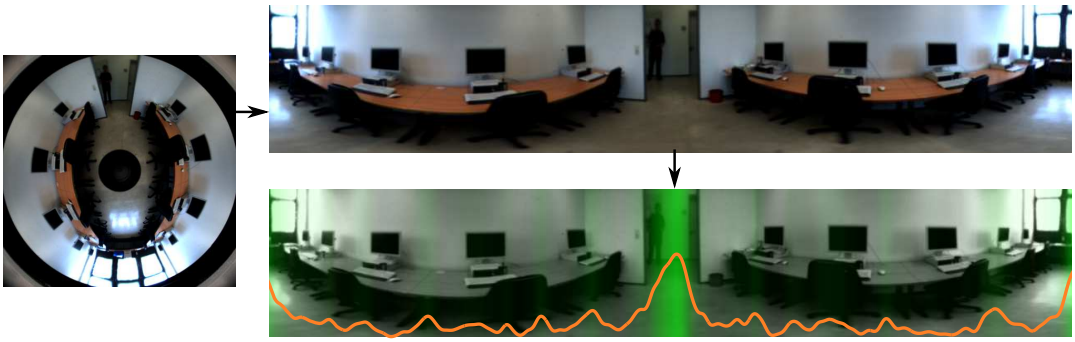


Figure 7.3: Door detection example. Left: Input omnidirectional image. Top: Panorama image. Bottom: Probability output of the door classifier represented with the opacity of the green mask and the superimposed orange line.

7.6 Navigation Behaviors

The behaviors' starting and ending points are described by semantic places: {*room, corridor, doorway, hallway*}. The low-level behaviors: {*obstacle avoidance, goal point, corridor following, door traversing*} are the building blocks of more sophisticated high-level behaviors such as: {*enter, get_out, traverse, follow*}. Each behavior is defined by the triplet <*start point, behavior, end point*>. The system first checks if the behavior can be activated in the current place context by querying the *place recognition* module and reporting the status if the behavior can be executed in the current location. Robust navigation between start and goal is accomplished by arbitration of the basic behaviors. The behavior *obstacle avoidance* is always present in the arbitration to avoid collisions. Once the target goal is reached, the robot reports the status comparing the current end place with the commanded one.

7.6.1 Goal Point Homing

The goal point behavior drives the robot from a current pose (x, y, θ) towards the goal pose (x', y', θ') using a feedback controller similar to [Ast99]. Fig. 7.4 illustrates the variables involved in the goal computation. The system is transformed to polar coordinates to obtain a smooth control law with the following relations:

$$\begin{aligned} \rho &= \sqrt{(x'-x)^2 + (y-y')^2} \\ \gamma &= \text{atan2}(y'-y, x'-x) - \theta \\ \phi &= \theta' - \theta + \gamma \end{aligned} \tag{7.6.1}$$

velocity and rotational velocity signals follow the control law:

$$\begin{aligned} v &= K_\rho \rho \\ \omega &= K_\gamma \gamma + K_\phi \phi \end{aligned} \tag{7.6.2}$$

in which the gains K_ρ, K_γ and K_ϕ are tuned to achieve a smooth convergence and $K_\rho > 0$, $K_\gamma - K_\rho < 0$ and $K_\phi < 0$ ensure local stability.

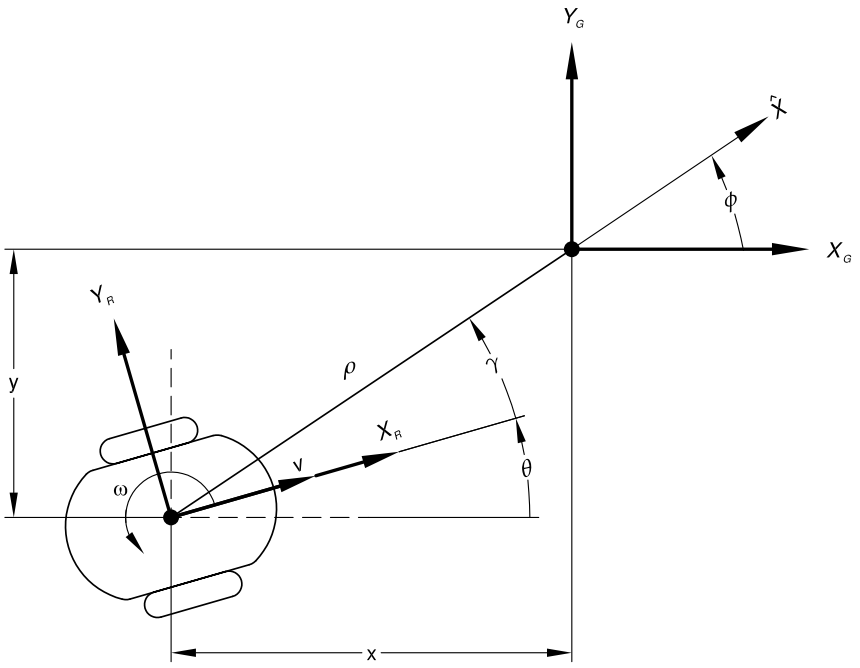


Figure 7.4: Goal point behavior involved variables.

7.6.2 Corridor Centering

This behavior is supposed to align the robot's heading with the orientation of the walls and centering the robot in the middle of a corridor. The robot's heading error and lateral offset are reflected in the distribution of the ground in robocentric coordinates. For this purpose, the free-space region is characterized by image moments, which approximate the free space distribution with an ellipse with major axis and minor axis and orientation φ. Fig. 7.5 sketches the involved variables in the centering behavior, where the segmented image is approximated with an ellipse. Considering a look ahead point A positioned on the major axis at a distance d from the centroid, the angles α and β are easily computed and describe the current robot's offset and misalignment to the corridor centerline at A.

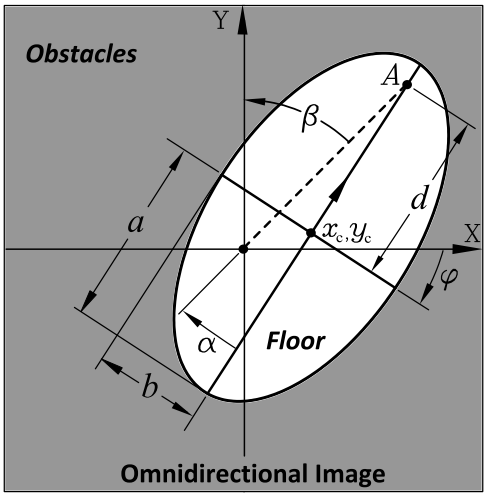


Figure 7.5: Corridor centering behavior variables

The image moments presented in chapter 4 allows us to calculate the major and minor axis and the orientation as follows:

$$\varphi = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \tag{7.6.3}$$

$$\begin{aligned} a &= \sqrt{\frac{2\left[\mu_{20}+\mu_{02}+\sqrt{(\mu_{20}-\mu_{02})^2+4\mu_{11}^2}\right]}{\mu_{00}}} \\ b &= \sqrt{\frac{2\left[\mu_{20}+\mu_{02}-\sqrt{(\mu_{20}-\mu_{02})^2+4\mu_{11}^2}\right]}{\mu_{00}}} \end{aligned} \tag{7.6.4}$$

It is assumed that the semi major axis coincides with the orientation of the corridor and that the displacement of the centroid from the image center captures the lateral displacement of the robot within the corridor. The lateral and orientation errors are computed with respect to a look-ahead point A located on the major axis at a distance d from the centroid. The angle

$$\alpha = -\varphi - \beta \tag{7.6.5}$$

between the vectors that connect the image center and the centroid of the floor with the look-ahead point, reflects the lateral offset. The angle

$$\beta = \arctan\left(\frac{x_c + d\cos(\varphi+\pi/2)}{y_c + d\sin(\varphi+\pi/2)}\right) \tag{7.6.6}$$

between the robot's current heading and the major axis directly captures the orientation error.

In experiments with the pioneer 3DX robot, a distance d in pixels corresponding to 2m in the ground showed good centering results. The turn rate that aligns the robot with the corridor middle line is computed from a similar stabilizing error feedback proportional control law of the goal point behavior, but with translational velocity constant.

$$\begin{aligned} v &= v_0 \\ \omega &= K_\alpha\alpha + K_\beta\beta \end{aligned} \tag{7.6.7}$$

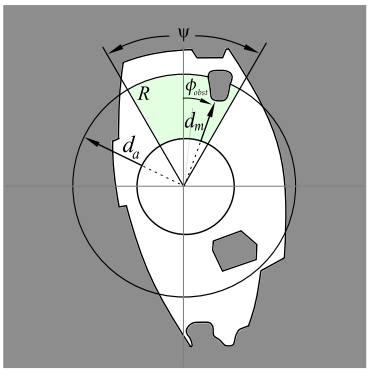
The robot's translational velocity remains constant v_0 during the centering maneuver. The gains $K_\alpha < 0$ and $K_\beta < 0$ are tuned to achieve a smooth convergence.

7.6.3 Obstacle Avoidance

This behavior monitors the frontal region of the omnidirectional image. The avoidance region R is defined by the aperture angle ψ and the activation distance d_a as illustrated in Fig. 7.6. If a non-floor pixel denoting an obstacle is detected inside R, the behavior is activated. The perception is aggregated into the distance d_m and heading ϕ_{obst} of the closest obstacle. The perception is mapped onto a motor action

$$\begin{aligned} v &= v_a d_m / d_a \\ \omega &= \text{sign}(\omega_o)\omega_a \sin\phi_{obst} \end{aligned} \tag{7.6.8}$$

The sign of the initial turn rate ω_o depends on whether the obstacle is in the right or left half-plane, and the robot then turns in the opposite direction. A turn flag



(a)

Figure 7.6: Obstacle avoidance behavior variables

sign(ω_o) memorizes this turning direction, which is maintained throughout the entire avoidance maneuver. The relative heading of the obstacle modulates the magnitude of the turn rate. The constants v_a and ω_a denote the avoidance translational and rotational velocities.

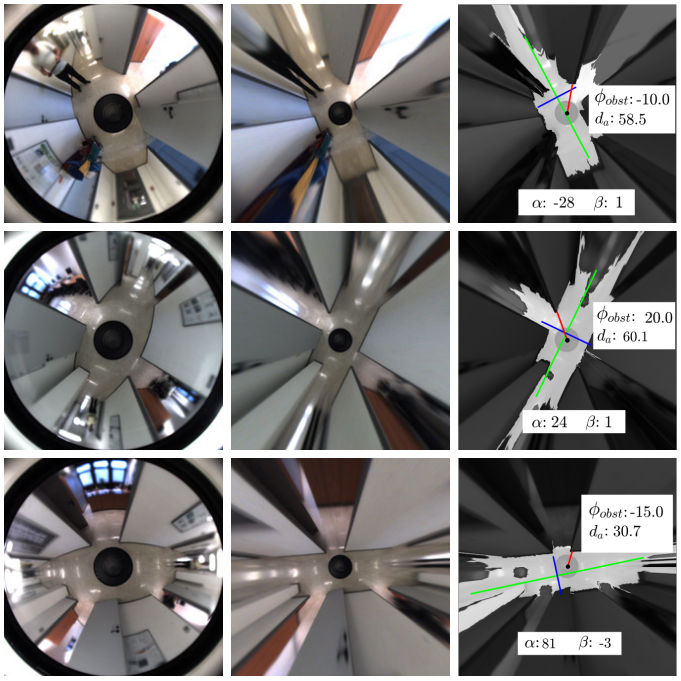


Figure 7.7: Examples showing the computed obstacle avoidance and corridor following variables. First column: input image. Second column: bird's-eye view. Third column: Bird's-eye view of the free-space with the behavior variables.

The obstacle avoidance and corridor following computed variables for three random images taken from different autonomous navigation experiments can be visualized in Fig. 7.7. In the three sequences, the robot could follow the corridor, and avoid obstacles when the activation distance was below a threshold distance of 1 meter. The third example shows an image where the segmentation misclassifies a column and weights the ellipse major line. In practice, this effect can be reduced by weighting the pixels more near the center, where the segmentation and bird's eye projection are

more reliable.

7.6.4 Door Passing Behavior

Detected doors similar to [Pos+09] are tracked by estimating the robot's current pose (x, y, θ) with respect to the midpoint of the door poles. This pose is recovered using bearing only localization, which triangulates door features from multiple captures taken at different viewpoints. The built-in odometer estimates the relative robot motion between consecutive viewpoints. Since measurements and motion are both subject to noise and errors, the door pose is estimated with an extended Kalman filter (EKF). Once the robot is aligned with the door's front, a door traversing behavior is activated using visual servoing to control the robot's turn rate such that both doorposts remain equilateral in the omnidirectional view.

The state prediction of the EKF relies on the odometry motion model, which describes the relative robot motion between two consecutive poses by three basic motions: an initial rotation δ_{rot1} followed by a straight motion δ_{trans} and a final rotation δ_{rot2}. Fig. 7.8 illustrates the involved variables in the bearing only localization. The odometry model predicts the relative robot motion between consecutive states [TBF05]:

$$\mathbf{x}_t^- = \begin{bmatrix} x_t^- \\ y_t^- \\ \theta_t^- \end{bmatrix} = \begin{bmatrix} x_{t-1}^+ \\ y_{t-1}^+ \\ \theta_{t-1}^+ \end{bmatrix} + \begin{bmatrix} \delta_{trans} \cos(\theta_{t-1} + \delta_{rot1}) \\ \delta_{trans} \sin(\theta_{t-1} + \delta_{rot1}) \\ \delta_{rot1} + \delta_{rot2} \end{bmatrix} \tag{7.6.9}$$

The superscript (-) denotes a priori estimate of the process model, and the superscript (+) indicates a posteriori estimate after the correction step.

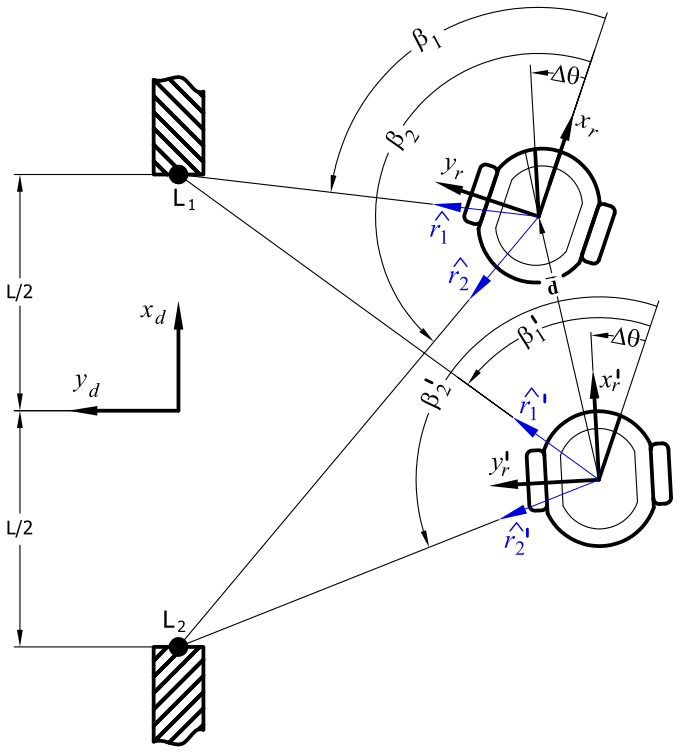


Figure 7.8: EKF bearing localization with respect to the door

The door coordinate frame $\langle x_D, y_D \rangle$ shown in Fig. 7.8 is located at the center of the door and the two robot poses are represented with the frames $\langle x_R^{'}, y_R^{'} \rangle$ and $\langle x_R, y_R \rangle$. With simple trigonometric relations is possible to obtain a formula for the door pose based on $k_{1,2}$ (the magnitudes of the vectors $\mathbf{r}_{1,2}$) as:

$$\begin{bmatrix} x_d \\ y_d \\ \theta_d \end{bmatrix} = \begin{bmatrix} (k_1 \cos(\beta_1) + k_2 \cos(\beta_2))/2 \\ (k_1 \sin(\beta_1) + k_2 \sin(\beta_2))/2 \\ \arctan\left(\frac{k_1 \sin(\beta_1) - k_2 \sin(\beta_2)}{k_1 \cos(\beta_1) - k_2 \sin(\beta 2)}\right) \end{bmatrix} \tag{7.6.10}$$

Notice that in the figure $\mathbf{r}_{1,2}$ are represented with their unitary vectors $\hat{\mathbf{r}}_{1,2}$. Finally, the robot pose with respect to the door frame can be predicted using the bearing angles:

$$\mathbf{z}_t = \begin{bmatrix} -x_d \cos(\theta_d) - y_d \sin(\theta_d) \\ x_d \sin(\theta_d) - y_d \cos(\theta_d) \\ -\theta_d \end{bmatrix} \tag{7.6.11}$$

In the correction step of the Kalman filter, the posteriori state estimate is obtained as $\mathbf{x}_t^+ = \mathbf{x}_t^- + K_t(\mathbf{z}_t - \mathbf{x}_t^-)$. In which the Kalman gain K_t depends on the ratio of measurement and process covariance. The Kalman filter is initialized based on the first two consecutive measurements of door post bearings.

At short distances from the door, the Kalman filter is no longer applied, and the door passing behavior is performed using visual servoing. The reason for this is that depth information becomes unreliable at close range and is not needed for guiding the robot through the door. Fig 7.9 shows the geometry and variables in the door traversing behavior. The robot traverses the door at a constant velocity by centering itself to the continuously tracked doorposts. The visual servoing controls the robot's turn rate, such both doorposts remain equilateral ($\beta_1 = \beta_2$) in the omnidirectional view.

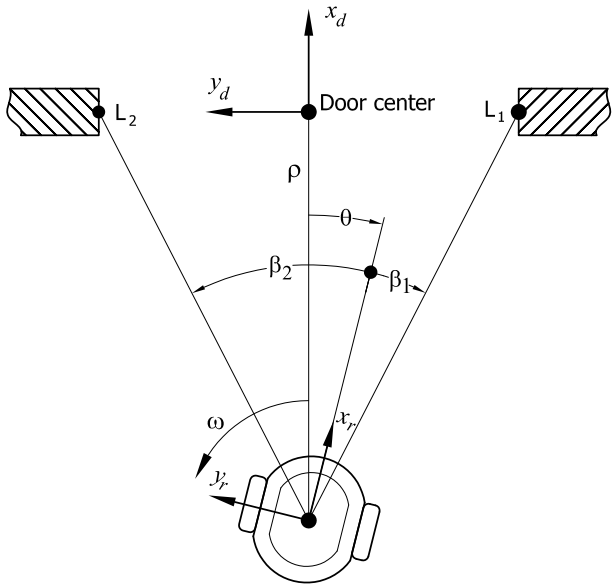


Figure 7.9: Door traversing behavior

7.6.5 High-Level Behaviors

In this section, we show how high-level behaviors can be achieved using compositions or combinations of basic behaviors to perform more specific tasks. Some examples include: *enter room, get out room, follow corridor until next door, follow hallway, etc*. A robot, for instance standing in a corridor, could activate an *enter room* behavior by following the sequence: First, detect a door. If the door center is not aligned with the robot center, activate the corridor following behavior to move the robot until alignment using the position estimates from the EKF door localization. Subsequently, once the robot is aligned with the door, rotate it until the robot's front coincides with the door opening. Then, activate the door traversing behavior with visual servoing to keep the robot in the middle of the door poles. Once the robot is placed at the other side of the door, query the place recognition module to detect whether the location corresponds to a room and reports the status. Any of the navigation behaviors can be combined with stop conditions by querying any of the modules. For example, a corridor following behavior can be activated with a stop condition when the robot reaches a door. Likewise, a hallway following behavior could be active until a wall is encountered.

7.7 Summary

This chapter has presented a local navigation system based on several basic visual behaviors that exploit semantic visual data. Four key modules allow the robot to navigate between semantic positions by activating either basic behaviors or compositions to perform higher-level navigation.

The free-space module estimates the robot's local drive-able regions upon all basic behaviors rely. The door detection module allows the robot to transition between indoor places and provides landmarks for local localization and traversing doors. The place category recognition module is crucial for determining the context and avoid conflicting behavior activations.

Finally, the navigation module describes the local navigation behaviors mapping visual data into motor actions. Four basic behaviors are presented, goal point homing, corridor centering, obstacle avoidance, and door passing.

Different future work opportunities can extend the presented local navigation by integrating the basic behaviors into a more elaborated navigation ontology and extending the system to handle more object and place categories as well as more high-level behaviors.

8

Map-Based Navigation and Semantic Mapping

8.1 Introduction

Map-based navigation is one of the fundamental competences of a robot. To be truly autonomous, a robot must localize itself to know its position anytime in the environment. This process requires abstracting the environment with a suitable map representation. In this chapter, we deal with occupancy grid maps built from omnidirectional images data. We employ occupancy grid maps for Montecarlo localization and extend it further with deeper semantic knowledge annotating each map's location into a semantic map. Semantic mapping has emerged in recent years as a very active research area [KG15], expanding robot capabilities significantly, allowing them to operate under more complex tasks and with better human-robot communication.

Robot navigation has been traditionally tackled relying on geometric information from range sensors. Laser 2D scanners and sonars have played a crucial role in map-based navigation during the past two decades and still today with new lidar 3D point-cloud technologies. However, in recent years, with the need for better perceptual capabilities and richer representations of more complex robotic tasks, the navigation problem has integrated increasingly more semantics captured from imaging devices.

This chapter introduces a semantic mapping and localization framework solely based on visual information from omnidirectional cameras. The map representation consists of an occupancy grid map, which is one of the most popular mapping techniques currently in robotics. Occupancy grids are robust, easy to create, and update but with the shortcoming that consume large memory space, which is problematic for very large environments.

Very few approaches [San+11; YL13] have been proposed able to build purely visual-based occupancy grids, which require a proper estimation of the robot's free space. The main motivation for using omnidirectional sensors lies in that they provide a global view of the robot's surroundings with a single image capture. This advantage comes with the drawback of limited resolution and distortion of omnidirectional images in comparison to conventional perspective vision sensors. Nonetheless, in map building, the lack of high resolution does not play a crucial role in reliable detection of free-space. The occupancy map building approach employs the free space

transformed region into a bird's-eye view which generates distortion-free local maps.

8.2 Related Work

Vision-based map building and localization are subjects already studied for a long time, and the main techniques are surveyed in [FPRARM15; GFO15]. The majority of schemes focus on monocular sensing devices employing different image features for building either a metric or topological representation. Feature-based methods have the advantage that can provide a compact and rich description of the environment in terms of appearance and shape. However, they are not capable of a dense geometric reconstruction as with occupancy grid maps [Thr02b].

Purely visual-based occupancy grid map building has been solved using different approaches and sensors. Monocular cameras provide no depth data; therefore, the mapping system must estimate depth either from a learned regressor [Pla+08] or thorough free-space segmentation [San+11]. Stereo vision mapping systems [AG07] provide depth but require textured scenes and known camera parameters to compute disparity. Some proposed systems combine different information and sensor modalities. Braillon *et al.* [Bra+06] fuse two partial occupancy grids from stereo vision and optical flow and [AG07] combine an occupancy grid with a feature-based map.

Omnidirectional vision methods are close to our system by sharing the advantages and limitations of the wide field of view. Plagemann *et al.* [Pla+10] project the omnidirectional image into a panoramic view to avoid polar indexing and implement a Gaussian process regression to mimic a laser rangefinder (Similar to the visual sonar [LV03] and closest color transitions distance [Men+06]). The mapping system in [Mac+10] combines feature-based visual odometry (computed inside an annulus in the omniview close to the center) with ray-casting obstacle distances on a segmented floor omni-image. More recently, [LLD15] combines feature-based visual SLAM to initially compute accurate poses with a later 3D semi-dense reconstruction for visibility reasoning. These steps result in a 2D occupancy grid map.

In the context of visual localization with omnidirectional images, the majority of approaches have tackle problem feature-based [Men+04; MGS07; Cou+08; LPB12; KUM13; Pay+14; LWF15]. A few approaches are able to localize on dense maps (e.g. occupancy grids) [Men+06; Pla+10; YL13]. These schemes have adapted common computer vision and robotics workflows to work on large distorted omni-images such as: global features from Fourier Transform [Men+04; Pay+14]; laser rangefinder emulation (with color changes [Men+06] ,with features and Gaussian regression [Pla+10]; and gradient changes [YL13] local features matching (pyramid matching kernels with SURF) [MGS07], combination of global and local descriptors [Cou+08] and bag of words using a Vector of Locally Aggregated Descriptors (VLAD) [KUM13].

Paya *et al.* [Pay+14] compare several popular global appearance descriptors such as Fourier Signature, Principal Components Analysis, Histogram of Oriented Gradients and Gist. Lourenco *et al.* [LPB12] proposed a system that localizes with monocular images, which query omnidirectional visual maps with adapted SIFT descriptors. The method in [LWF15] employs feature tracking and an estimator that fuse inertial and odometry measurements.

The main semantic mapping approaches proposed in the literature are surveyed in [KG15] An early attempt of building a richer map representation consisted of annotating occupancy maps with place labels coming from a classifier trained with laser data encoded into different range-features [MMSB05]. An extension of this work by [Rot+05] improved the disambiguation of similar place categories by integrating object recognition within the scene with a monocular camera.

3D laser scans have been used to build semantic agricultural maps [WB10] and labeling of objects instead of places [NH08; SBB12]. More recently, [Sün+16] have build an occupancy grid map that combines the advantages of the Bayes filter implemented in the occupancy grid algorithm to capture temporal coherence of the sensor data with the latest advances of place classification using convolutional neural networks. The novelty of their work lies in the fact that their system can learn online new semantic classes. This is a limitation of conventional computer vision classification, which considers isolated images and its output is limited to the classes of the dataset where they were trained (closed-set limitation).

Related to our work with omnidirectional images, Rituerto *et al.* [RMG12] present a system that can label indoor topological maps using a rotation-invariant GIST [OT01] descriptor adaptation. Their system employs a cascaded classifier to recognize first places and transitions, and further, to discriminate them into corridors, rooms, doors, and stairs. The difference with our system is that can build a dense semantic map with richer image features useful for localization A related problem tackled by [ZBK07] consists of grouping images associated to similar rooms and locations using clustering according to convex places such as rooms but assigns no category to the places.

8.3 Semantic Mapping Framework

The overall system architecture of the semantic mapping framework was presented on [Pos+18] and is illustrated in Fig. 8.1. The system is divided into five modules: (i) *free space* in charge of segmenting the floor region (ii) *inverse sensor model* that computes the probability of occupancy given the current omni-image floor segmentation (iii) *occupancy grid mapping* in charge of updating the occupancy probability of each grid cell with the Bayes filter (iv) *place category recognition* for assigning a semantic place to each perception, and (v) *semantic mapping* that annotates the range-based occupancy grid with the recognized semantic places maintaining a grid map for each place class.

8.3.1 Free Space Segmentation

The *free space* module oversees finding free-space with a binary floor/obstacle classifier. We built upon chapter 4, Fig. 8.2 (b) and (c) illustrate an example of a superpixel image and its corresponding floor segmentation. Another feasible segmentation can be found in chapter 2, which fuses multiple segmentations from different cues with an ensemble of experts.

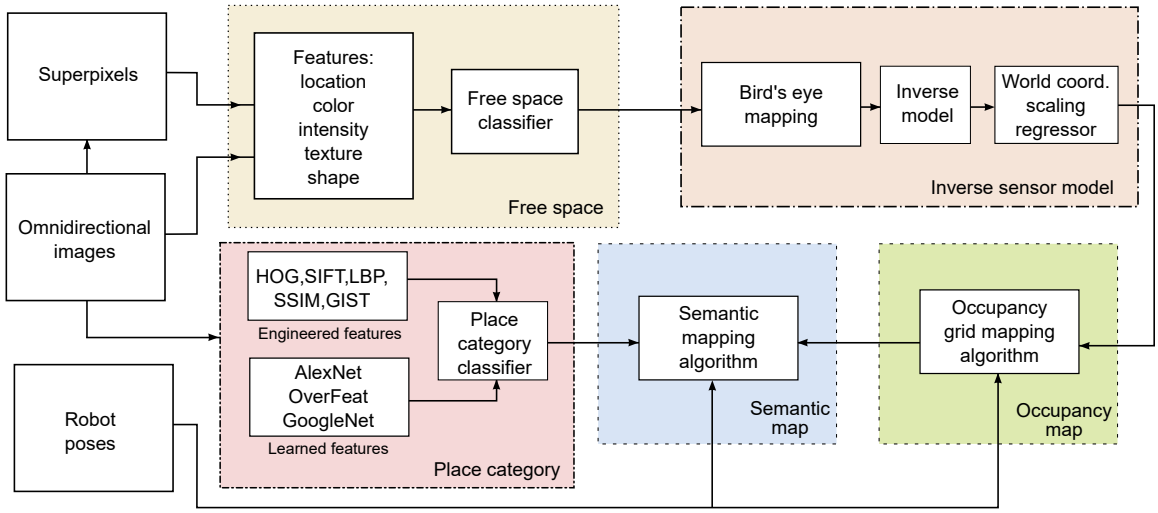


Figure 8.1: System architecture of the semantic mapping framework

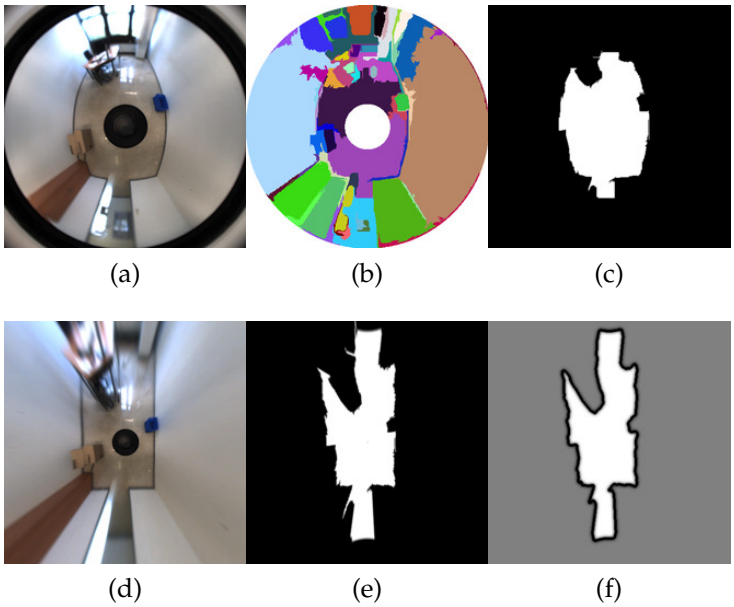


Figure 8.2: Inverse sensor model a) Input Image b) Superpixels for semantic classification c) Floor Segmentation d) Bird's-eye projection of the input image e) Bird's-eye view of the free space f) Inverse Sensor model: gray color (0.5) represents unknown occupancy, white color represents the free space, while color black corresponds to the occupied space. Notice the Gaussian distribution around the occupied regions

8.3.2 Inverse Sensor Model

The *inverse sensor model* is a key module that computes the probability of occupancy conditioned on the floor segmentation. The process starts by correcting the non-linear distortion induced by the omni-camera mirror. The bird's eye transformation introduced in chapter 5 is used for undistorting the image and projecting the floor into a virtual plane, as illustrated in Fig. 8.2 (d)-(e). The second step of the inverse model consists of assigning occupancy values for all grid cells as follows: free space is represented with white pixels (unoccupied cells), whereas border edges indicate occupied cells and are represented with black. Beyond the edges, the occupancy information is unknown. Fig. 8.2 (f) depicts the inverse model cell assignation with the above assumptions. Free space regions are assigned with 0, border regions with 1, and unknown occupancy with a value of 0.5.

The final step of the inverse model is to transform image coordinates to map coordinates with a trained robust regression model [RL87]. We fit the regressor with ground truth data from a laser rangefinder. For each laser ray, we sample in its direction, the free-space segmentation range value in pixels (See Fig. 8.3). We train the model with 50 images and 180 scans for a total of 9000 training instances.

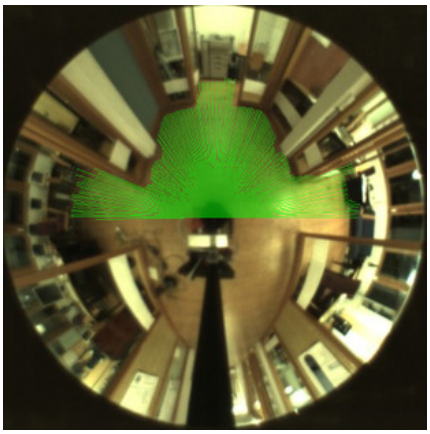


Figure 8.3: Laser ranger scans projected onto the omnidirectional image

8.3.3 Occupancy Grid Mapping

The map building process follows the occupancy grid algorithm [Thr02b], where a posterior over the map is computed using a Bayes filter and a 2D cell map representation. Each grid cell m_i has a binary value of occupancy of 0 or 1 (corresponding to free or occupied space).

The mapping problem consists of calculating the posterior over maps $p(m_i|z_{1:t}, x_{1:t})$, with $z_{1:t}$ the measurements and $x_{1:t}$ the poses up to time t. The *log-odds* formulation of the Bayes filter [TBF05] is very convenient allowing faster computations by using additions instead of multiplications and avoiding numerical instabilities when dividing by probabilities close to zero. With $odds(x) = \frac{p(x)}{1-p(x)}$, the binary Bayes filter for occupancy grid can be written as:

$$\log \frac{p\left(m_i \mid z_{1:t}, x_{1:t}\right)}{1-p\left(m_i \mid z_{1:t}, x_{1:t}\right)} = \log \frac{p\left(m_i \mid z_t, x_t\right)}{1-p\left(m_i \mid z_t, x_t\right)} + \log \frac{1-p\left(m_i\right)}{p\left(m_i\right)} + \log \frac{p\left(m_i \mid z_{1:t-1}, x_{1:t-1}\right)}{1-p\left(m_i \mid z_{1:t-1}, x_{1:t-1}\right)}, \tag{8.3.1}$$

which can be simplified using the terms $l_{t,i}$ and $l_{t-1,i}$ to:

$$l_{t,i} = \log \frac{p\left(m_i \mid z_t, x_t\right)}{1-p\left(m_i \mid z_t, x_t\right)} + \log \frac{1-p\left(m_i\right)}{p\left(m_i\right)} + l_{t-1,i} \tag{8.3.2}$$

The estimation is done recursively, and all cells are updated with newly available sensor information. The probability of occupancy is calculated from the *log-odds* as:

$$p\left(m_i \mid z_{1:t}, x_{1:t}\right) = 1 - \frac{1}{1+\exp\left\{l_{t,i}\right\}} \tag{8.3.3}$$

The probability $p(m_i)$ is the prior of occupancy and has a typical value of 0.5 (unknown occupancy); the probability $p(m_i|z_t, x_t)$ is called the inverse sensor model with values computed using the model derived above.

8.3.4 Place Classification

The place categorization module labels each image with the categories: room (R), corridor (C), doorway (D), and open room (O). Our approach classifies single omnidirectional images rather than accumulating evidence over image sequences [RMG12]. The place categorization analysis starts with extracting visual features with good scene description performance. The place category recognition follows the method introduced in chapter 6.

8.3.5 Semantic Mapping

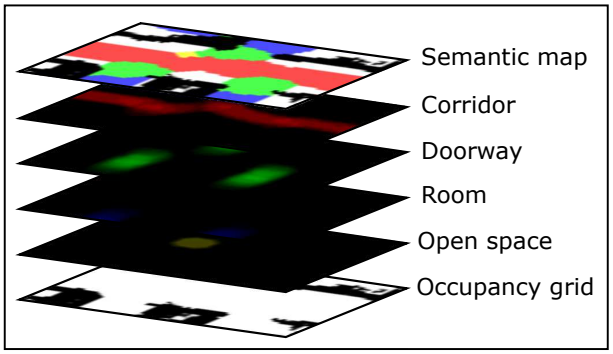


Figure 8.4: Semantic map building. Each place class maintains a separated grid map parallel to the occupancy grid map.

The *semantic mapping* module operates similar to the occupancy grid mapping. For each place class, a separated grid map is maintained and updated with the Bayes

filter equation 8.3.1 to ensure the temporal consistency of the perceptions. Instead of recording occupancy, the semantic map updates the probability of each cell being the current place category. Updates are made only in free-cells and cells within a circle of the current location. A Gaussian modulates the updating to give more weight to the cells near the current pose and decrease it on those further away. Fig. 8.4 illustrates the semantic mapping process with each place class having a separated grid map parallel to the occupancy map.

8.4 Semantic Mapping Results

The dataset in this study consists of a sensor data sequence of 2145 images acquired at the Freiburg University location from the public available COLD database [PC09]. Odometry and robot poses are available; therefore, no SLAM pre-processing for estimating the robot poses is necessary. Fig. 8.5 sketches the Freiburg floor plan and the trajectory followed by the robot while acquiring the sensor data.

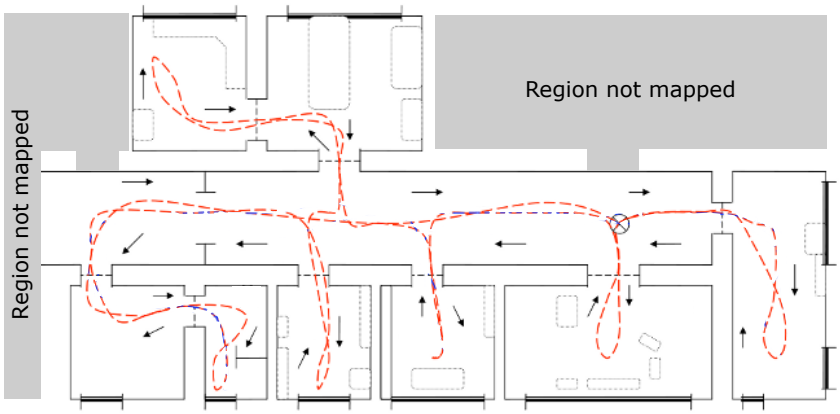


Figure 8.5: Sketch of the Freiburg floor plan (COLD Dataset). The dotted red line indicates the sequence followed by the robot while acquiring the sensor data

The map building process with omnidirectional images as the only sensor data is illustrated in Fig. 8.6. At $t = 0$ (first image on the left), the occupancy is unknown, and all grid elements are assigned with $p = 0.5$ (color gray). After acquiring the first sensor data at $t = 1$ (second image), the occupancy changes according to equation 8.3.2, updating cells covered by the regions of the local map. The map continues to be updated from each incoming sensor data until frame 2145 (last image). Notice that areas beyond the perception are not altered, whereas the probabilities of cells inside the omniview are updated according to the local free space in the inverse sensor model.

A comparison of the vision-based map versus the one constructed with a 2D laser scanner is shown in Fig. 8.7. Notice that the laser rays reach far beyond the mapped areas of the visual method. The visual-based map also shows some phantom artifacts in the map caused by data misalignment (odometry and images). This is especially noticeable at points of fast robot rotation where the optical flow becomes large. The mismatch of images and odometry was visually confirmed by plotting all sensor data

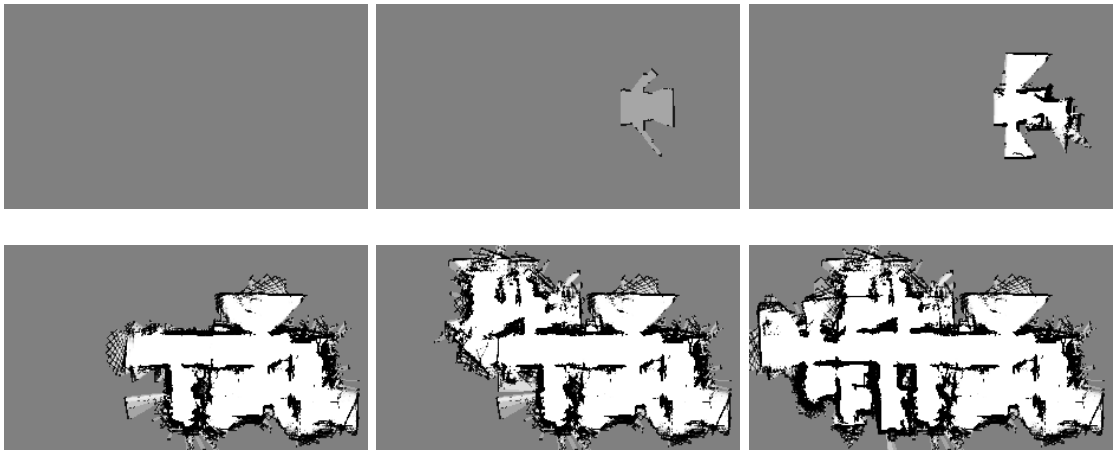
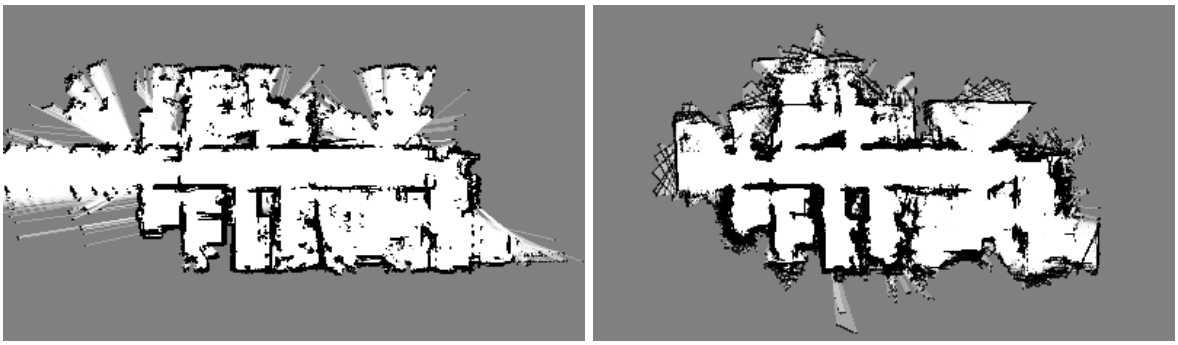


Figure 8.6: Incremental steps while building the occupancy grid map of Freiburg University from the COLD dataset. The first image (top-left) corresponds to $t = 0$. The following steps shown correspond to $t = 1, 100, 1000, 1500$ until last frame at $t = 2745$ (bottom-right image)

(odometry, laser, and images) frame by frame. The laser constructed map is not affected by this effect since laser and odometry data match perfectly.



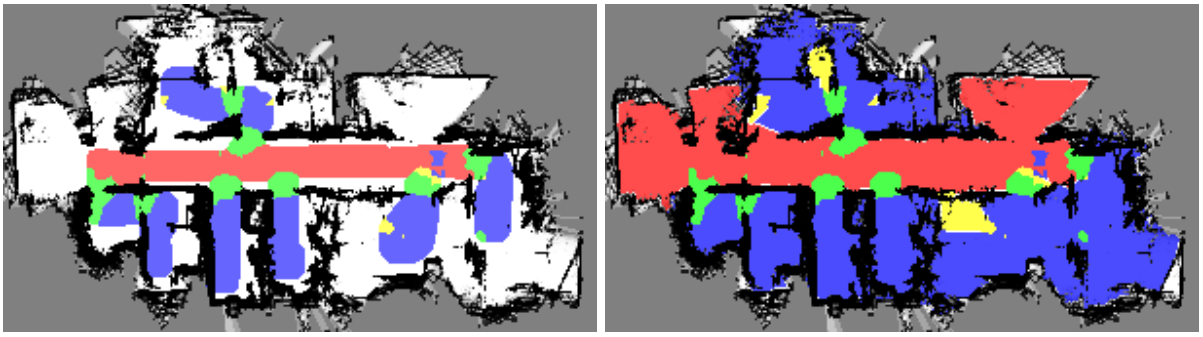
(a) (b)

Figure 8.7: Comparison of the map building using a laser rangefinder vs using visual data. a) Map built using 2D laser scans (shown as reference) c) Map built using the proposed method with omnidirectional images.

The resulting semantic map of the COLD dataset Freiburg sequence is shown in Fig. 8.8. The map on the left corresponds to the semantic labels annotated within a circle outlined from each point of the robot's trajectory. The map on the right extends the semantic labels to the nearest un-occupied cells with a region growing algorithm. Notice that the robot to build a more accurate semantic map needs to visit all map places to avoid miss-classifications with the label expansion.

8.5 Monte Carlo Localization with Omnidirectional Vision

This section describes the localization scheme followed by our robot. The approach rest upon a sensor model of the omnivision segmentation that relies on scan matching and provides the basis for the Monte Carlo localization. Probabilistic-based local-



(a) (b)

Figure 8.8: Semantic map building. Red: corridor, Blue: room, Green: doorway, Yellow: open spaces a) Semantic labels superimposed on top of the map b) Labels expanded beyond the robot trajectory with region growing.

ization methods have consistently outperformed alternative methods since they can better handle the uncertainty of real-world such as sensor noise and aliasing, actuator noise, errors, and abstraction in the models. Their advantages lie in being able to represent robot pose as a probability distribution over all poses, instead of choosing the best estimate.

To localize the robot, we should estimate the robot's pose in a map m using data from the omnimages and the tracked motion with odometry. We employ the particle filter localization scheme [Thr+01; Thr02a], also referred in the literature as Monte Carlo Localization.

We first consider the general formulation of Markov localization (Localization using the Bayes filter) before reviewing the particle filter localization, which is a special case of Markov localization. The Bayes filter allows computing the posterior belief distribution $bel(x_t)$ of a robot pose x_t, where $bel(x_t)$ is an abbreviation of the posterior:

$$bel(x_t) = p(x_t|u_{1:t}, z_{1:t}) \tag{8.5.1}$$

$bel(x_t)$ at time t is conditioned on the controls u and measurements z.

The Bayes filter allows the computation of $bel(x_t)$ from a former state $bel(x_{t-1})$ recursively using the Markov assumption, where x_t is a function only of a previous state x_{t-1} and a recent actions u_t and the measurements z_t.

$$Bel(x_t) = \eta p(z_t|x_t, m) \int p(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1} \tag{8.5.2}$$

The state transition probability $p(x_t|u_t, x_{t-1})$ is referred as *motion model* and serves for the prediction step in the Bayes filter, whereas $p(z_t|x_t, m)$ is *measurement model* used to compute the probability of a measurement given the location of a robot in the map.

Particle Filters

Particle filters are a non-parametric implementation of the Bayes filter and represent $Bel(x_t)$ with a discrete set of M weighted particles

$$X_t = \left\{ < x_t^{[1]}, w_t^{[1]} >, < x_t^{[2]}, w_t^{[2]} >, ..., < x_t^{[M]}, w_t^{[M]} > \right\} \tag{8.5.3}$$

The particle filter localization algorithm estimates the true pose of the robot in four steps.

1. *Initialization*: the initial belief $bel(x_o)$ is obtained with a set of samples $\{< x_o^{[i]}, w_o^{[i]} >\}$ drawn according to a uniform distribution over the map with homogeneous importance factors $w = 1/M$

2. *Prediction*: The prediction step computes the integral part of equation 8.5.2, which is the product of two distributions. $bel(x_{t-1})$ is the prior probability assigned to the state x_{t-1}, and $p(x_t|u_t, x_{t-1})$ is the probabilistic *motion model* that describes the probability that the control u_t induces the transition from x_{t-1} to x_t. Each sample in the prediction step is drawn from the motion model.

3. *Update*: The update step assigns an importance weight to each particle. This is achieved by computing the left multiplier in equation 8.5.2, which is the *measurement model* $p(z_t|x_t, m)$. The likelihood of z_t given x_t and the knowledge of the map m determine the weight w of each particle.

4. *Resampling*: this step generates a new set of particles around previous particles that have a large weight. That means that particles with high likelihood have a higher probability to being re-sampled than particles with low likelihood. After the resampling process, particle weights become homogeneous $w = \frac{1}{N}$. Thus, computational resources are allocated across the localization space according to the probability density.

Motion Model

The particle filter localization has the advantage that it is not necessary to compute $p(x_t|u_t, x_{t-1})$ explicitly. Instead, $p(x_t|u_t, x_{t-1})$ is sampled for arbitrary poses x_t, x_{t-1} and u_t using the *odometry motion model* [TBF05]. Odometry can estimate motion, but its value is only available after robot movement. This poses a problem in robotics control or planning, but not in our case with localization and mapping.

The *odometry motion model* considers odometry data as controls u_t (which are technically measurements). This simplification has the advantage of keeping the state space small and achieves superior results when compared to the common used *velocity motion model*. The velocity model, additional to the problems of slippage and geometry inconsistencies, suffers from the mismatch between the actual actuated motion and the motion computed from mathematical models.

The robot motion between two poses in the *odometry motion model* is decomposed using three basic movements: a rotation followed by a translation and another rotation. The sampling algorithm outputs a random x_t according to the distribution $p(x_t|u_t, x_{t-1})$ considering different error parameters. A detailed description can be found in the book [TBF05].

Measurement Model

The measurement model $p(z_t|x_t, m)$ computes the likelihood of observing the measurement z_t given the robot pose x_t and the knowledge of the map m. In the particle filter, this probability is proportional to the weight or importance factor of each particle $w_t^{[i]} = p(z_t|x_t^{[i]}, m)$. The importance factor is calculated using map matching of a local occupancy grid map m_{local} to the global map m. Fig. 8.9 illustrates the expected scan (m_i), which corresponds to an ideal noise-free scan of the obstacles according to the map at the particle's pose. The normalized cross-correlation or Baron's correlation is widely used as a measure of amount of matching.

$$\rho_{m,m_{local}} = \frac{1}{N-1} \frac{1}{\sqrt{\sigma_m^2 \sigma_{m_{local}}^2}} \sum_i (m_i - \mu_m)(m_{local,i} - \mu_{m_{local}}) \tag{8.5.4}$$

$$\mu_m = \frac{1}{N} \sum_i m_i, \;\; \mu_{m_{local}} = \frac{1}{N} \sum_i m_{local,i} \tag{8.5.5}$$

$$\sigma_m^2 = \frac{1}{N-1} \sum_i (m_i - \mu_m)^2, \; \sigma_{m_{local}}^2 = \frac{1}{N-1} \sum_i (m_{local,i} - \mu_{m_{local}})^2 \tag{8.5.6}$$

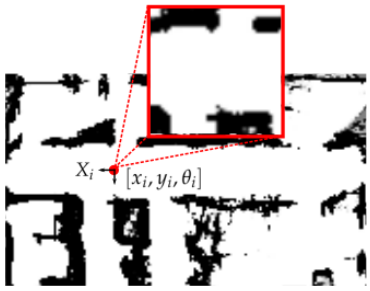


Figure 8.9: Particle's expected scan at pose (x_i, y_i, θ_i)

The output values are in the range $[-1, 1]$. Therefore, they are shifted to positive values and scaled to sum up to one.

Scan matching has the advantage that it considers the free-space directly within the local map rather than computing features or scans endpoint, thus including additional information present in the local occupancy grid maps.

Resampling

The low variance sampler [TBF05] is a resampling strategy based on a sequential stochastic process. Instead of generating independently random numbers while selecting the particles, the low variance algorithm computes a single random number r and selects a sample with replacement. r lies between 0 and $1/M$, where M is the number of samples. The number $1/M$ is added repeatedly to r, and a particle is selected if it corresponds to the result of the addition.

8.6 Robot Localization Results

Figure 8.10 shows a localization sequence of 220 sensor instances acquired at the Technische Universität Dortmund [Oli10]. The robot's localization state is shown at different stages. The environment presents some difficulties for the localization due to substantial perceptual aliasing caused by the geometric similarity of locations along the hallway. The blue circle and the green square correspond to the *true* and *estimated* location of the robot respectively. We assess the robustness of the localization scheme with an odometry motion model that exhibits substantially more noise than the actual robot's odometry. The robot's true location is recovered from odometry data, which is, in this particular case, is highly accurate as the robot travels along a straight path down the corridor.

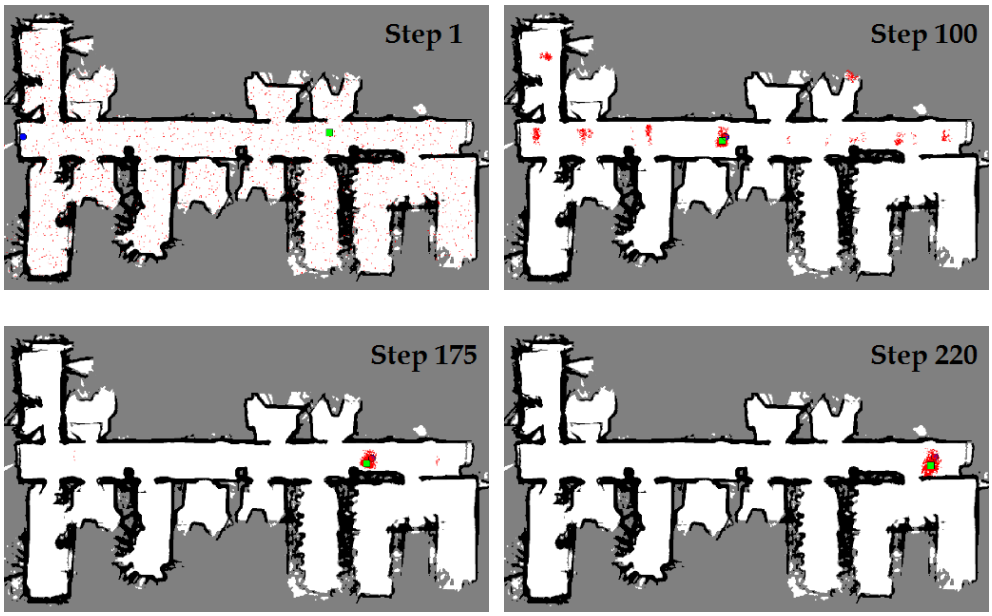


Figure 8.10: Localization using a Particle Filter with omnidirectional images. The sequence consists of 220 images and its correspoinding occupancy grid map at the Technische Universität Dortmund [Oli10]. The green square represents the computed robot's location, while the blue dot is the real robot's position.

The estimated locations are calculated by clustering a total of 1500 particles. The cluster with the most particles determines the robot's location. Thus, the estimated location switches from cluster to cluster until the majority of particles belong to only one cluster. In the shown experiment, the robot is approximately localized after 75 steps corresponding to a distance of around nine meters.

8.7 Summary

This chapter has presented a semantic mapping and localization framework able to operate solely with omnidirectional visual data. The proposed sensor model mimics a range a sensor in that it segments the local robots free space. The sensor model also corrects the non-linear distortions of the omnidirectional mirror and outputs a scaled

perspective image of the ground plane using bird's-eye mapping. These free space segmented bird's eye view images constitute the perceptual basis for both mapping and the localization.

Map building is achieved by discretizing the environment into a grid, in which the state of each cell is described by its probability of being occupied or empty. The occupancy grid algorithm updates the probabilities of the global map.

The obtained occupancy grid map, constitute the basis of the Monte Carlo localization and the semantic map building. The semantic map employs the place category classifier from chapter 6 to label categories: room, corridor, doorway, and open room. Each place class maintains a separate grid map that is fused with the range-based occupancy grid for building a dense semantic map.

Experiments with real robot sensor data for localization and map building confirm the utility of omnidirectional vision as a universal sensor in mobile robotics.

9

Conclusions and Outlook

This thesis has covered some of the essential components of a robot navigation system solely based on omnidirectional vision. We have shown robust methods tested in real-life experiments, ranging from local to map-based navigation. In some of the tasks, our results are comparable to classical proximity sensors, such as laser rangefinders or ultrasound sensors. However, as robots require more semantic information from complex tasks, the more evident the need for computer vision, which provides a much richer representation of the environment.

In the first chapters, we developed different free-space segmentation schemes to mimic with a camera the functionality of a range sensor. Two different approaches were proposed. The first scheme rests upon the fusion of multiple segmentation schemes. Each segmentation relies on a particular feature to determine a pixel's class label. The classifiers were trained with ground truth provided by a 3D camera. The second approach employed an online self-supervised scheme that selects the traversable floor region in the omnidirectional view using the optimal segmentation to the local environment by cross-validation over 3D scans captured by a 3D camera.

Uncertainty in the free-space classifiers was handled with Fuzzy Preference Structures. We investigated the uncertainty inherent in a single free-space segmentation, as well as the global uncertainty across multiple segmentation based classifiers. The uncertainty explicitly expressed in terms of preference, conflict, and ignorance utilizing Fuzzy Preference Structures. We showed that the classification error could be substantially reduced by rejecting queries associated with a strong degree of conflict and ignorance.

In an effort to incorporate more semantics into the system, we enhanced the binary obstacle/floor classifier with a semantic classifier. The proposed scheme considers a larger feature set and labels each image's region with the classes: floor, vertical structures, and clutter. The scheme aggregates multiple heterogeneous features computed at the superpixel level with a Random Forest classifier. The method is suitable for self-supervised learning using the 3D point cloud data of a range camera.

The semantic segmentation was further enhanced to provide the spatial layout and surface reconstruction of the scene in an omnidirectional image. Evidence of supporting structures and the 3D layout can significantly enhance the scene understanding and object detection performance. The proposed method considers a novel bird's eye view formulation and benefits from a cascaded classification, where the free-space and vertical structures are first extracted. Subsequently, all vertical structures are ana-

lyzed to find their main orientation. The fusion of orientation features from the HOG transform, floor/wall boundaries, and oriented lines proved to be very effective in the layout classification experiments.

One of the fundamental capabilities of a service robot is to recognize the place where it is. In the thesis, a place category recognition was proposed to distinguish among the location categories: room, corridor, doorway, and open space. The system evaluated hand-engineered features and learned from data representations. The place category classifier showed improvement from the combination of heterogeneous features using both non-learning and learning combination methods.

Previously obtained semantic information was integrated into a local navigation framework. We explored local semantic navigation by deriving several visual behaviors. Four basic behaviors were presented: goal point homing, corridor centering, obstacle avoidance, and door passing. Current place knowledge described with a semantic category (e.g. room, corridor, doorway) gives the context to the behaviors. While regions in the scene labeled with semantic regions (e.g. floor, door, wall, floor) provide the constraints for the motion, the derived visual behaviors employ all this available semantic information. The advantage of the proposed navigation is that the system can recover from conflicting errors (e.g. activating a behavior in the wrong context). Higher-level behaviors can be achieved by compositions of basic behaviors to perform more complex behaviors.

The last part of the thesis faced two aspects of map-based navigation: (i) map-based localization solely based on omnidirectional vision, and (ii) building semantic maps with omnidirectional vision. The map representation consisted of an occupancy grid built with a sensor model of the omni-view able to correct the non-linear distortions of the hyperbolic mirror and output a scaled perspective image of the ground plane using the bird's-eye mapping. The free-space segmented bird's-eye view constituted the perceptual basis for both the mapping and the localization. Localization was achieved with Monte Carlo localization and the semantic map building employs the place category classifier from Chapter 6 to label the place categories. Each place class maintains a separate grid map that is fused with the range-based occupancy grid to build a dense semantic map.

The work presented in this thesis has shown promising results for vision as the sensor of choice for robot navigation. We believe the results can be enhanced significantly in the future providing larger image datasets to take advantage of the faster computation hardware coming out every day. Currently, the new era of GPU computing has seen flourishing new deep learning methods, which can learn the huge parameter space of very large architectures that generalize to very complex recognition tasks. The manual deriving of specialized features, therefore, becomes an obsolete task.

Currently, in the context of omnidirectional images, the datasets remain small. Improvement can be achieved using transfer learning and fine-tuning from general-purpose features of an already trained model. A future work direction can also investigate learning with self-supervision from 3D using the proposed methods from some of the chapters of this work. This way, it is possible to generate automatically labeled data for feeding the data-hungry deep neural networks.

Bibliography

Published Papers

This thesis is based on the following previously published papers by the author:

[HPB11] **F. Hoffmann, L. F. Posada, and T. Bertram:** "Fuzzy Preference Structures for Floor-Obstacle Classification in Omnidirectional Images". In: *World Conference on Soft Computing 2011 (WConSc)*. 2011.

[PHB14] **L. F. Posada, F. Hoffmann, and T. Bertram:** "Visual semantic robot navigation in indoor environments". In: *41st International Symposium on Robotics. (ISR/Robotik)*. VDE. 2014.

[Pos+09] **L. F. Posada, T. Nierobisch, F. Hoffmann, and T. Bertram:** "Image Signal Processing for Visual Door Passing with an Omnidirectional Camera". In: *International Conference on Computer Vision Theory and Applications*. 2009.

[Pos+10a] **L. F. Posada, K. K. Narayanan, F. Hoffmann, and T. Bertram:** "Robuste Boden-Hindernis Segmentierung durch ein Ensemble von Experten fur die Navigation mobiler Roboter". In: *Proceedings 20. Workshop on Computational Intelligence*. 2010.

[Pos+10b] **L. F. Posada, K.K. Narayanan, F. Hoffmann, and T. Bertram:** "Floor Segmentation of Omnidirectional Images for Mobile Robot Visual Navigation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010.

[Pos+11a] **L. F. Posada, K. K. Narayanan, F. Hoffmann, and T. Bertram:** "Detecting Free Space and Obstacles in Omnidirectional Images". In: *International Conference of Intelligent Robotics and Applications (ICIRA)*. 2011.

[Pos+11b] **L. F. Posada, K. K. Narayanan, F. Hoffmann, and T. Bertram:** "Ensemble of experts for robust floor-obstacle segmentation of omnidirectional images for mobile robot visual navigation". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011.

[Pos+13] **L. F. Posada, K. K. Narayanan, F. Hoffmann, and T. Bertram:** "Semantic Classification of Scenes and Places with Omnidirectional Vision". In: *6th European Conference on Mobile Robots*. 2013.

[Pos+18] **L. F. Posada, A. Velasquez-Lopez, F. Hoffmann, and T. Bertram:** "Semantic Mapping with Omnidirectional Vision". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018.

[PVL16] **L. F. Posada and A. Velasquez-Lopez:** "Spatial layout and surface reconstruction from omnidirectional images". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016.

Contributed Papers

The author also contributed to the following related papers during the realization of this thesis:

[Nar+09] **K. K. Narayanan, L. F. Posada, F. Hoffmann, and T. Bertram:** "Imitation Learning for Visual Robotic Behaviors". In: *Proceedings 19. Workshop on Computational Intelligence*. 2009.

[Nar+10] **K. K. Narayanan, L. F. Posada, F. Hoffmann, and T. Bertram:** "Robot Programming by Demonstration". In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. 2010.

[Nar+11a] **K. K. Narayanan, L. F. Posada, F. Hoffmann, and T. Bertram:** "Scenario and context specific visual robot behavior learning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011.

[Nar+11b] **K. K. Narayanan, L. F. Posada, F. Hoffmann, and T. Bertram:** "Situated learning of visual robot behaviors". In: *International Conference of Intelligent Robotics and Applications (ICIRA)*. 2011.

[Nar+12] **K. K. Narayanan, L. F. Posada, F. Hoffmann, and T. Bertram:** "Machine Interfaces for Intuitive and Effective Demonstrations of Mobile Robot Behaviors". In: *Proceedings 22. Workshop on Computational Intelligence*. 2012.

[Nar+13] **K. K. Narayanan, L. F. Posada, F. Hoffmann, and T. Bertram:** "Acquisition of Behavioral Dynamics for Vision Based Mobile Robot Navigation from Demonstrations". In: *6th IFAC Symposium on Mechatronic Systems*. Hangzhou, China, 2013.

[Oel+14] **M. Oeljeklaus, L. F. Posada, F. Hoffmann, and T. Bertram:** "Analyse globaler Bildmerkmale zur Klassifikation von Verkehrsszenen". In: *Proceedings 24. Workshop on Computational Intelligence*. 2014.

References

Resources by other researchers are summarized in the following list:

[AG07] **F. Andert and L.s Goormann:** "Combined grid and feature-based occupancy map building in large outdoor environments". In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2007.

[Aho+09] **T. Ahonen, J. Matas, C. He, and M. Pietikäinen:** "Rotation invariant image description with local binary pattern histogram fourier features". In: *Scandinavian Conference on Image Analysis*. 2009, pp. 61–70.

[Ast99] **A. Astolfi:** "Exponential stabilization of a wheeled mobile robot via discontinuous control". In: *Journal of Dynamic Systems, Measurements, and Control* 121 (1999), pp. 121–126.

[Bau+09] **A. M. Bauer, K. Klasing, T. Xu, S. Sosnowski, G. Lidoris, Q. Mühlbauer, T. Zhang, F. Rohrmüller, D. Wollherr, K. Kühnlenz, and M. Buss:** "The Autonomous City Explorer project". In: *ICRA*. 2009.

[Baz+12] **J. C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon:** "Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment". In: *I. J. Robotic Res.* 31.1 (2012), pp. 63–81.

[Bla+08] **M. R. Blas, M. Agrawal, A. Sundaresan, and K. Konolige:** "Fast color/texture segmentation for outdoor robots". In: *Int. Conf. on Intelligent Robots and Systems (IROS)*. 2008.

[BP12] **J. C. Bazin and M. Pollefeys:** "3-line RANSAC for Orthogonal Vanishing Point Detection". In: *IROS*. 2012.

[Bra+06] **C. Braillon, K. Usher, C. Pradalier, JL. Crowley, and C. Laugier:** "Fusion of stereo and optical flow data using occupancy grids". In: *Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2006.

[Bre01] **L. Breiman:** "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[BZM07a] **A. Bosch, A. Zisserman, and X. Munoz:** "Image classification using random forests and ferns". In: *ICCV*. 2007.

[BZM07b] **A. Bosch, A. Zisserman, and X. Munoz:** "Representing shape with a spatial pyramid kernel". In: *ACM International Conference on Image and Video Retrieval*. 2007.

[Che+16] **L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille:** "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *arXiv: 1606.00915* (2016).

[CL11] **C.C. Chang and C.J. Lin:** "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011), 27:1–27:27.

[Cou+08] **J. Courbon, Y. Mezouar, L. Eck, and P. Martinet:** "Efficient hierarchical localization method in an omnidirectional images memory". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2008.

[CPZ09] **K. Chatfield, J. Philbin, and A. Zisserman:** "Efficient Retrieval of Deformable Shape Classes using Local Self-Similarities". In: *Workshop on Non-rigid Shape Analysis and Deformable Image Alignment, ICCV*. 2009.

[Cri01] **A. Criminisi:** "Single-view metrology". In: *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer, 2001, pp. 69–105.

[CV95] **C. Cortes and V. Vapnik:** "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[Dah+06] **H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and Bradski G.:** "Self-supervised monocular road detection in desert terrain". In: *Robotics Science and Systems Conference*. 2006.

[Del+16] **J. Delmerico, A. Giusti, E. Mueggler, L. M. Gambardella, and D. Scaramuzza:** ""On-the-spot training" for terrain classification in autonomous air-ground collaborative teams". In: *International Symposium on Experimental Robotics*. 2016.

[Den+09] **J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei:** "ImageNet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2009.

[DLN05] **E. Delage, H. Lee, and A. Y. Ng:** "Automatic Single-Image 3D reconstruction of Indoor Manhatan World Scenes". In: *Int. Symp. Robotic Research*. 2005.

[DS01] **D. Driankov and A. Saffiotti:** *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Studies in Fuzziness and Soft Computing. Springer, 2001.

[DS13] **D. Driankov and A. Saffiotti:** *Fuzzy logic techniques for autonomous vehicle navigation*. Vol. 61. 2013.

[DT05] **N. Dalal and B. Triggs:** "Histograms of oriented gradients for human detection". In: *CVPR*. 2005.

[Elk03] **C. Elkan:** "Using the triangle inequality to accelerate k-means". In: *International Conference on Machine Learning*. 2003.

[FCCJ09] **L. da Fontoura Costa and R. M. Cesar Jr.:** *Shape Classification and Analysis: Theory and Practice*. CRC Press, 2009.

[Fel+10] **P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan:** "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9 (2010), pp. 1627–1645.

[FH04] **P. F. Felzenszwalb and D.P. Huttenlocher:** "Efficient Graph-Based Image Segmentation". In: *International Journal of Computer Vision* 59.2 (2004), pp. 167–181.

[FPRARM15] **J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha:** "Visual simultaneous localization and mapping: a survey". In: *Artificial Intelligence Review* 43.1 (2015), pp. 55–81.

[FVS09] **B. Fulkerson, A. Vedaldi, and S. Soatto:** "Class segmentation and object localization with superpixel neighborhoods". In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 670–677.

[GA11] **M. Gönen and E. Alpaydın:** "Multiple kernel learning algorithms". In: *The Journal of Machine Learning Research* (2011).

[Gey00] **K. Geyer C.and Daniilidis:** "A unifying theory for central panoramic systems and practical implications". In: *Computer Vision—ECCV 2000*. Springer, 2000, pp. 445–461.

[GFO15] **E. Garcia-Fidalgo and A. Ortiz:** "Vision-based topological mapping and localization methods: A survey". In: *Robotics and Autonomous Systems* 64 (2015), pp. 1–20.

[GG+17] **A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez:** "A Review on Deep Learning Techniques Applied to Semantic Segmentation". In: *arXiv:1704.06857* (2017).

[Gru+07] **G. Grudic, J. Mulligan, M. Otte, and A. Bates:** "Online Learning of Multiple Perceptual Models for Navigation in Unknown Terrain". In: *6th International Conference on Field and Service Robotics*. 2007.

[GW08] **R. C. Gonzalez and R. E. Woods:** *Digital Image Processing*. Prentice-Hall, 2008.

[HB08] **E. Hüllermeier and K. Brinker:** "Learning valued preference structures for solving classification problems". In: *Fuzzy Sets and Systems* 159 (18 2008), pp. 2337–2352.

[HC12a] **O. Haines and A. Calway:** "Estimating Planar Structure in Single Images by Learning from Examples." In: *ICPRAM (2)*. 2012, pp. 289–294.

[HC12b] **O. Haines and A. Calway:** "Estimating Planar Structure in Single Images by Learning from Examples". In: *ICPRAM*. 2012, pp. 289–294.

[HEH05] **D. Hoeim, A. A. Efros, and M. Hebert:** "Geometric context from a Single Image". In: *CVPR*. 2005.

[HEH07] **D. Hoeim, A. A. Efros, and M. Hebert:** "Recovering Surface Layout from an Image". In: *International Journal of Computer Vision* 75.1 (2007).

[HHF09] **V. Hedau, D. Hoeim, and D. A. Forsyth:** "Recovering the spatial layout of cluttered rooms". In: *ICCV*. 2009.

[Hu62] **M. Hu:** "Visual pattern recognition by moment invariants". In: *IRE transactions on information theory* 8.2 (1962), pp. 179–187.

[Jia+14] **Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell:** "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv: 1408.5093* (2014).

[KG15] **I. Kostavelis and A. Gasteratos:** "Semantic mapping for mobile robotics tasks: A survey". In: *Robotics and Autonomous Systems* 66 (2015), pp. 86–103.

[Kho+12] **A. Khosla, J. Xiao, A. Torralba, and A. Oliva:** "Memorability of Image Regions". In: *Advances in Neural Information Processing Systems (NIPS)*. 2012.

[Kim+06] **D. Kim, J. Sun, S. Min, O. James, M. Rehg, and A. F. Bobick:** "Traversability Classification Using Unsupervised On-Line Visual Learning for Outdoor Robot Navigation". In: *In Proc. of Int. Conf. on Robotics and Automation (ICRA)*. 2006.

[KK04] **Y. Kim and H. Kim:** "Layered ground floor detection for vision-based mobile robot navigation". In: *Int. Conf. on Robotics and Automation (ICRA)*. 2004.

[KSH12] **A. Krizhevsky, I. Sutskever, and G. E. Hinton:** "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[KUM13] **H. Korrapati, F. Uzer, and Y. Mezouar:** "Hierarchical visual mapping with omnidirectional images". In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2013.

[LBG97] **L. M. Lorigo, R. A. Brooks, and W. E. L. Grimsou:** "Visually-guided obstacle avoidance in unstructured environments". In: *Int. Conf. on Intelligent Robots and Systems (IROS)*. 1997.

[LBH15] **Y. LeCun, Y. Bengio, and G. Hinton:** "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[LeC+98] **Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner:** "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[Lee+10] **D. C. Lee, A. Gupta, M. Hebert, and T. Kanade:** "Estimating Spatial Layout of Rooms using Volumetric Reasoningabout Objects and Surfaces". In: *NIPS*. 2010.

[LHK09] **D. Lee, M. Hebert, and T. Kanade:** "Geometric reasoning for single image structure recovery". In: *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. 2009.

[Liu+11] **M.Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa:** "Entropy rate superpixel segmentation". In: *CVPR*. 2011.

[LLD15] **R. Lukierski, S. Leutenegger, and A. J. Davison:** "Rapid free-space mapping from a single omnidirectional camera". In: *2015 European Conference on Mobile Robots (ECMR)*. IEEE. 2015.

[Low04] **D. G. Lowe:** "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[LPB12] **M. Lourenço, V. Pedro, and J. P. Barreto:** "Localization in indoor environments by querying omnidirectional visual maps using perspective images". In: *International Conference on Robotics and Automation (ICRA)*. IEEE. 2012.

[LSD15] **J. Long, E. Shelhamer, and T. Darrell:** "Fully convolutional networks for semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[LSP06] **S. Lazebnik, C. Schmid, and J. Ponce:** "Beyond Bags of Features Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *CVPR*. 2006.

[LV03] **S. Lenser and M. Veloso:** "Visual sonar: fast obstacle avoidance using monocular vision". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. 2003.

[LWF15] **Y-H. Li L.and Liu, K. Wang, and M. Fang:** "Estimating position of mobile robots from omnidirectional vision using an adaptive algorithm". In: *IEEE transactions on cybernetics* 45.8 (2015), pp. 1633–1646.

[Mac+10] **V. Macchia, S. Rosa, L. Carlone, and B. Bona:** "An Application of Omnidirectional Vision to Grid-based SLAM in Indoor Environments". In: *ICRA 2010 Workshop on Omnidirectional Robot Vision*. IEEE. 2010.

[Mar06] **C. M. Martin:** "Evolving visual sonar: Depth from monocular images". In: *Pattern Recognition Letters* 27.11 (2006), pp. 1174–1180.

[Mat+13] **C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox:** "Learning to parse natural language commands to a robot control system". In: *Experimental Robotics*. Springer. 2013, pp. 403–415.

[Men+04] **E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro:** "Image-based Monte Carlo localisation with omnidirectional images". In: *Robotics and Autonomous Systems* 48.1 (2004), pp. 17–30.

[Men+06] **E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello:** "Omnidirectional vision scan matching for robot localization in dynamic environments". In: *IEEE transactions on robotics* 22.3 (2006), pp. 523–535.

[MGS07] **A. C. Murillo, J. J. Guerrero, and C. Sagues:** "Surf features for efficient robot localization with omnidirectional images". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2007.

[MMSB05] **O. Martinez Mozos, C. Stachniss, and W. Burgard:** "Supervised Learning of Places from Range Data using Adaboost". In: *ICRA*. 2005.

[Nay97] **S. K. Nayar:** "Catadioptric omnidirectional camera". In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE. 1997, pp. 482–488.

[Ngu+05] **V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart:** "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics". In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2005.

[NH08] **A. Nüchter and J. Hertzberg:** "Towards semantic maps for mobile robots". In: *Robotics and Autonomous Systems* 56.11 (2008), pp. 915–926.

[NR07] **S. Noykov and C. Roumenin:** "Occupancy grids building by sonar and mobile robot". In: *Robotics and Autonomous Systems* 55 (2007), pp. 162–175.

[Oli10] **J. Oliva:** "A Mobile Robotics Framework for Omnidirectional Vision Based Localization and Mapping aided by a remote Human Machine Interface". M.Sc. Thesis. Germany: Technische Universität Dortmund, 2010.

[Oli14] **A. Oliva:** "Scene Perception". In: *New Visual Neurosciences*. Ed. by John S. Werner and Leo. M. Chalupa. Mit Press, 2014. Chap. 51, pp. 725–732.

[OLNG11] **N.D. Ozisik, G. Lopéz-Nicolás, and J.J. Guerrero:** "Scene structure recovery from a single omnidirectional image". In: *ICCV Workshops*. 2011.

[OLNG12] **J. Omedes, G. Lopez-Nicolas, and J.J. Guerrero:** "Omnidirectional Vision for Indoor Spatial Layout Recovery". In: *Intelligent Autonomous Systems Conference*. 2012.

[OPH96] **T. Ojala, M. Pietikäinen, and D. Harwood:** "A comparative study of texture measures with classification based on featured distributions". In: *Pattern recognition* 29.1 (1996), pp. 51–59.

[OPM02] **T. Ojala, M. Pietikainen, and T. Maenpaa:** "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.7 (2002), pp. 971–987.

[OT01] **A. Oliva and A. Torralba:** "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope". In: *International Journal of Computer Vision* 42 (2001), pp. 145–175.

[OUV97] **G. Oriolo, G. Ulivi, and M. Vendittelli:** "Fuzzy maps: A new tool for mobile robot perception and planning". In: *Journal of Robotic Systems* 14.3 (1997), pp. 179–197.

[OUV98] **G. Oriolo, G. Ulivi, and M. Vendittelli:** "Real-time map building and navigation for autonomous robots in unknown environments". In: *IEEE Transactions on Systems, Man and Cybernetics Part B* 28 (3 1998), pp. 316–333.

[Pap13] **P. Papadakis:** "Terrain traversability analysis methods for unmanned ground vehicles: A survey". In: *Engineering Applications of Artificial Intelligence* 26.4 (2013).

[Pay+14] **L. Payá, F. Amorós, L. Fernández, and O. Reinoso:** "Performance of global-appearance descriptors in map building and localization using omnidirectional vision". In: *Sensors* 14.2 (2014), pp. 3033–3064.

[PC09] **A. Pronobis and B. Caputo:** "COLD: COsy Localization Database". In: *The International Journal of Robotics Research (IJRR)* 28.5 (May 2009), pp. 588–594.

[Per+07] **M. Persson, T. Duckett, C. Valgren, and A.J. Lilienthal:** "Probabilistic Semantic Mapping with a Virtual Sensor for Building/Nature detection". In: *CIRA*. 2007.

[PL01] **N. Pears and B. Liang:** "Ground plane segmentation for mobile robot visual navigation". In: *Int. Conf. on Intelligent Robots and Systems (IROS)*. 2001.

[Pla+08] **C. Plagemann, F. Endres, J. Hess, C. Stachniss, and W. Burgard:** "Monocular range sensing: A non-parametric learning approach". In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. 2008.

[Pla+10] **C. Plagemann, C. Stachniss, J. Hess, F. Endres, and N. Franklin:** "A nonparametric learning approach to range sensing from omnidirectional vision". In: *Robotics and Autonomous Systems* 58.6 (2010), pp. 762–772.

[Pol06] **R. Polikar:** "Ensemble Based Systems in Decision Making". In: *IEEE Circuits and Systems Magazine* 6.3 (2006), pp. 21–45.

[PPP17] **A. Pandey, S. Pandey, and DR. Parhi:** "Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review". In: *Int Rob Auto J* 2.3 (2017).

[PR92] **J. Prokop and Anthony P. Reeves:** "A survey of moment-based techniques for unoccluded object representation and recognition". In: *Graphical Models and Image Processing*. 1992, pp. 438–460.

[QT09] **A. Quattoni and A. Torralba:** "Recognizing indoor scenes". In: *CVPR*. 2009.

[Ren+15] **S. Ren, K. He, R. Girshick, and J. Sun:** "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[RL87] **Peter J Rousseeuw and Annick M Leroy:** *Robust regression and outlier detection*. John wiley & sons, 1987.

[RM01] **J. BTM. Roerdink and A. Meijster:** "The Watershed Transform: Definitions, Algorithms and Parallelization Strategies". In: *Fundamenta Informaticae* 41.1-2 (2001), pp. 187–228.

[RM04] **L. W. Renninger and J. Malik:** "When is scene identification just texture recognition?" In: *Vision research* 44.19 (2004), pp. 2301–2311.

[RMG12] **A. Rituerto, A.C. Murillo, and J.J. Guerrero:** "Semantic labeling for indoor topological mapping using a wearable catadioptric system". In: *Robotics and Autonomous Systems* (2012).

[Rot+05] **A. Rottmann, O. Martinez Mozos, C. Stachniss, and W. Burgard:** "Semantic Place Classification of Indoor Environments with Mobile Robots Using Boosting". In: *AAAI*. 2005.

[RP01] **M. Ribo and A. Pinz:** "A comparison of three uncertainty calculi for building sonar-based occupancy grids". In: *Robotics and Autonomous Systems* 35 (3-4 2001), pp. 201–209.

[RSV] **S. Roebert, T. Schmits, and A. Visser:** "Creating a bird-eye view map using an omnidirectional camera". In: Citeseer.

[Rus+15] **O. Russakovsky, J. Deng, H. Su, J.Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei:** "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[San+11] **A. M. Santana, K. RT. Aires, R. MS. Veras, and A. AD. Medeiros:** "An Approach for 2D Visual Occupancy Grid Map Using Monocular Vision". In: *Electronic Notes in Theoretical Computer Science* 281 (2011), pp. 175–191.

[SB91] **M. J. Swain and D. H. Ballard:** "Color indexing". In: *International Journal of Computer Vision* 7 (1991), pp. 11–32.

[SBB12] **J. Stückler, N. Biresev, and S. Behnke:** "Semantic mapping using object-class segmentation of RGB-D images". In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2012.

[SBS10] **M. Sun, Y. Bao, and S. Savarese:** "Object Detection with Geometrical Context Feedback Loop." In: *British Machine Vision Conference (BMVC)*. Citeseer. 2010.

[SC07] **L. Sanchez and I. Couso:** "Advocating the use of imprecisely observed data in genetic fuzzy systems". In: *IEEE Transactions on Fuzzy Systems* 15 (2007), pp. 551–562.

[Ser+13] **P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun:** "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv: 1312.6229* (2013).

[SI07] **E. Shechtman and M. Irani:** "Matching Local Self-Similarities across Images and Videos". In: *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*. 2007.

[SMS06] **D. Scaramuzza, A. Martinelli, and R. Siegwart:** "A Toolbox for Easily Calibrating Omnidirectional Cameras". In: *IROS*. 2006.

[SR+14] **A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson:** "CNN features off-the-shelf: an astounding baseline for recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014, pp. 806–813.

[SSB15] **B. Suger, B. Steder, and W. Burgard:** "Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3d-lidar data". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015.

[SSN07] **A. Saxena, J. Schulte, and A. Y. Ng:** "Depth Estimation Using Monocular and Stereo Cues". In: *IJCAI*. 2007, pp. 2197–2203.

[SSN09] **A. Saxena, M. Sun, and A. Y. Ng:** "Make 3D: Learning 3D Scene Structure from a Single Still Image". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (2009).

[Sün+16] **N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford:** "Place categorization and semantic mapping on a mobile robot". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 5729–5736.

[SZ14] **K. Simonyan and A. Zisserman:** "Very deep convolutional networks for large-scale image recognition". In: *arXiv: 1409.1556* (2014).

[Sze+15] **C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich:** "Going deeper with convolutions". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[TBF05] **S. Thrun, W. Burgard, and D. Fox:** *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[Tho16] **M. Thoma:** "A survey of semantic segmentation". In: *arXiv: 1602.06541* (2016).

[Thr+01] **S. Thrun, D. Fox, W. Burgard, and F. Dellaert:** "Robust Monte Carlo localization for mobile robots". In: *Artificial intelligence* 128.1-2 (2001), pp. 99–141.

[Thr02a] **S. Thrun:** "Particle filters in robotics". In: *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc. 2002.

[Thr02b] **S. Thrun:** "Robotic Mapping: A Survey". In: *Exploring Artificial Intelligence in the New Millenium*. Ed. by G. Lakemeyer and B. Nebel. Morgan Kaufmann, 2002.

[Tse+05] **V. Tsetsos, C. Anagnostopoulos, P. Kikiras, P. Hasiotis, and S. Hadjiefthymiades:** "A human-centered semantic navigation system for indoor environments". In: *IEEE International Conference on Pervasive Services*. 2005.

[UN00] **I. Ulrich and I. Nourbakhsh:** "Appearance-Based Obstacle Detection with Monocular Color Vision". In: *Proc. AAAI 2000*. Austin, TX, 2000.

[URD11] **K. Uhl, A. Roennau, and R. Dillmann:** "From structure to actions: Semantic navigation planning in office environments". In: *IROS 2011 Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment*. 2011.

[Val+16] **A. Valada, G. Oliveira, T. Brox, and W. Burgard:** "Deep Multispectral Semantic Scene Understanding of Forested Environments using Multimodal Fusion". In: *The 2016 International Symposium on Experimental Robotics (ISER 2016)*. 2016.

[VDW+09] **J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus:** "Learning color names for real-world applications". In: *IEEE Transactions on Image Processing* 18.7 (2009), pp. 1512–1523.

[VF08] **A. Vedaldi and B. Fulkerson:** *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/. 2008.

[VZ05] **M. Varma and A. Zisserman:** "A statistical approach to texture classification from single images". In: *International Journal of Computer Vision* 62.1-2 (2005), pp. 61–81.

[VZ12] **A. Vedaldi and A. Zisserman:** "Efficient additive kernels via explicit feature maps". In: *IEEE transactions on pattern analysis and machine intelligence* 34.3 (2012), pp. 480–492.

[WB10] **U. Weiss and P. Biber:** "Semantic place classification and mapping for autonomous agricultural robots". In: *IROS Workshop on Semantic Mapping and Autonomous, Knowledge Acquisition*. 2010.

[Wol+14] **D. Wolf, M. Bajones, J. Prankl, and M. Vincze:** "Find my mug: Efficient object search with a mobile robot using semantic segmentation". In: *arXiv: 1404.5765* (2014).

[Xia+10] **J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba:** "Sun database: Large-scale scene recognition from abbey to zoo". In: *Int. Conference on Computer vision and pattern recognition*. 2010, pp. 3485–3492.

[Xia+14] **J. Xiao, K. A Ehinger, J. Hays, A. Torralba, and A. Oliva:** "Sun database: Exploring a large collection of scene categories". In: *International Journal of Computer Vision* (2014), pp. 1–20.

[YL13] **Jun-Yu Yang and Feng-Li Lian:** "Scanning-based floor region identification from omnidirectional images for mobile robot localization". In: *SICE Annual Conference*. IEEE. 2013.

[YZM08] **S. X. Yu, H. Zhang, and J. Malik:** "Inferring Spatial Layout from A Single Image via Depth-Ordered Grouping". In: *In CVPR Workshop on POCV*. 2008.

[ZBK07] **Z. Zivkovic, O. Booij, and B. J. A. Kröse:** "From images to rooms". In: *Robotics and Autonomous Systems* 55.5 (2007).

[ZF14] **M.D Zeiler and R. Fergus:** "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. 2014.

[Zha+07] **J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid:** "Local features and kernels for classification of texture and object categories: A comprehensive study". In: *International journal of computer vision* 73.2 (2007), pp. 213–238.

[Zhe+15] **S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. HS Torr:** "Conditional random fields as recurrent neural networks". In: *IEEE International Conference on Computer Vision*. 2015.

[Zho+14] **B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva:** "Learning deep features for scene recognition using places database". In: *Advances in neural information processing systems*. 2014, pp. 487–495.

[Zhu+16] **H. Zhu, F. Meng, J. Cai, and S. Lu:** "Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation". In: *Journal of Visual Communication and Image Representation* (2016).