5

Semantic Modelling of Space

Andrzej Pronobis[1], Patric Jensfelt[1], Kristoffer Sjöö[1], Hendrik Zender[2], Geert-Jan M. Kruijff[2], Oscar Martinez Mozos[3], and Wolfram Burgard[3]

[1] Royal Institute of Technology (KTH), Centre for Autonomous Systems,
Stockholm, Sweden
{pronobis, patric, krsj}@csc.kth.se

[2] DFKI GmbH, Saarbrücken, Germany
{zender, gj}@dfki.de

[3] Albert-Ludwigs-Universität Freiburg, Department of Computer Science,
Freiburg, Germany
{omartine, burgard}@informatik.uni-freiburg.de

5.1 Introduction

A cornerstone for robotic assistants is their understanding of the space they are to be operating in: an environment built by people for people to live and work in. The research questions we are interested in in this chapter concern spatial understanding, and its connection to acting and interacting in indoor environments. Comparing the way robots typically perceive and represent the world with findings from cognitive psychology about how humans do it, it is evident that there is a large discrepancy. If robots are to understand humans and vice versa, robots need to make use of the same concepts to refer to things and phenomena as a person would do. Bridging the gap between human and robot spatial representations is thus of paramount importance.

A spatial knowledge representation for robotic assistants must address the issues of human-robot communication. However, it must also provide a basis for spatial reasoning and efficient planning. Finally, it must ensure safe and reliable navigation control. Only then can robots be deployed in semi-structured environments, such as offices, where they have to interact with humans in everyday situations.

In order to meet the aforementioned requirements, i.e. robust robot control and human-like conceptualization, in CoSy, we adopted a spatial representation that contains maps at different levels of abstraction. This stepwise abstraction from raw sensory input not only produces maps that are suitable for reliable robot navigation, but also yields a level of representation that is similar to a human conceptualization of spatial organization. Furthermore, this model provides a richer semantic view of an environment that permits the robot to do spatial categorization rather than only instantiation.

H.I. Christensen et al. (Eds.): Cognitive Systems, COSMOS 8, pp. 165–221.
springerlink.com

This approach is at the heart of the Explorer demonstrator (cf. Chapter 10), which is a mobile robot capable of creating a conceptual spatial map of an indoor environment. In the present chapter, we describe how we use multi-modal sensory input provided by a laser range finder and a camera in order to build more and more abstract spatial representations.

5.1.1 Related Work

Research in spatial representations for mobile robots has yielded different multi-layered environment models. Vasudevan *et al.* [1] suggest a hierarchical probabilistic representation of space based on objects. The work by Galindo *et al.* [2] presents an approach containing two parallel hierarchies, spatial and conceptual, connected through anchoring. Inference about places is based on objects found in them. Furthermore, the *Hybrid Spatial Semantic Hierarchy* (HSSH), introduced by Beeson *et al.* [3], allows a mobile robot to describe the world using different representations, each with its own ontology.

Other different cognitively inspired approaches to robot navigation convey route descriptions from a technically naïve user to a mobile robot. These approaches need not necessarily rely on an exact global self-localization, but rather require the execution of a sequence of strictly local, well-defined behaviors in order to iteratively reach a target position. Kuipers [4] presents the *Spatial Semantic Hierarchy* (SSH). Alternatively, the *Route Graph* model is introduced by Krieg-Brückner *et al.* [5]. Both theories propose a cognitively inspired multi-layered representation of the *map in the head*, which is at the same time suitable for robot navigation.

Additionally, several approaches on mobile robotics extend metric maps of indoor environments with semantic information. The work by Diosi *et al.* [6] creates a metric map through a guided tour. The map is then segmented according to the labels given by the instructor. Martinez Mozos *et al.* [7] extract a topological semantic map from a metric one using supervised learning. Alternatively, Friedman *et al.* [8] use *Voronoi Random Fields* for extracting the topologies. Although these works use range measurements as main input data, other sensors have been used for similar tasks. Torralba *et al.* [9] use processed images to distinguish between different place categories in the environment. Pronobis *et al.* [10] also use vision to recognize the different places that form an indoor environment. Finally, the combination of different sensory modalities can improve the recognition, as shown in Rottmann *et al.* [11] and Pronobis *et al.* [12]. More detailed review of different approaches to place classification can be found in Section 5.8.

The multi-layered representation presented in this chapter differs from the previous work primarily in the level of integration achieved. First, each of the layers of the representation advances the state of the art in its corresponding area. Second, the advanced techniques are combined into a single, coherent model, representing the world at various levels of abstraction (e.g. metric, topological, semantic, conceptual) based on information coming from multiple sources (vision, range sensors, verbal cues etc.). In particular, the model integrates the approaches of [7] and [12] for the semantic classification of places with visual object search algorithms [13] and the metric mapping based on the M-Space representation [14]. Moreover, the representation is designed for human-robot interaction and the models generated using the aforementioned techniques are combined with a common-sense ontology of an indoor environment. This bridges the gap in spatial understanding between the robot and humans and allows to include information extracted from verbal cues into the representation.

5.1.2 Outline

The rest of this chapter is organized as follows. First, we highlight the background for the research on spatial representations for mobile robots (Section 5.2). Then, we provide an overview of our spatial model (Section 5.3) and describe each of the levels of representation in detail (Sections 5.4–5.6). Finally, we present the algorithms used to augment the representation with semantic object and place information (Sections 5.7 and 5.8, respectively) and report results of performed experiments (Section 5.9). We conclude the chapter with a brief summary in Section 5.10.

5.2 Background

An approach to endowing autonomous robots with a human-like conceptualization of space inherently needs to take into account research in sensor-based mapping and localization for robots as well as findings about human spatial cognition.

Research in cognitive psychology addresses the inherently qualitative nature of human spatial knowledge. In accordance with experimental studies, it is nowadays generally assumed that humans adopt a partially hierarchical representation of spatial organization [15, 16]. The basic units of such a qualitative spatial representation are topological regions [17], which correspond to more or less clearly bounded spatial areas. The borders may be defined physically, perceptually, or may be purely subjective to the human. It has been shown that even in natural environments without any clear physical or perceptual boundaries, humans decompose space into topological hierarchies by clustering salient landmarks [18].

Aside from the functionality of the cognitive map, another relevant question from cognitive science is how people categorize spatial structures. Categories determine how people can interact with, and linguistically refer to entities in the world. Basic-level categories represent the most appropriate name for a thing or an abstract concept. The basic-level category of a referent is assumed to provide enough information to establish equivalence with other members of the class, while distinguishing it from non-members [19, 20]. We draw from these notions when categorizing the spatial areas in the robot's conceptual map. We are specifically concerned with determining appropriate properties that allow us to meaningfully refer to spatial entities in a situated dialogue between the robot and its user.

5.3 Overview of the Spatial Model

This section gives an overview of the multi-layered conceptual spatial model adopted in the CoSy project. The model forms a basis for spatial understanding, reasoning, navigation, and human-robot interaction in the integrated robotic system. In this framework, the space is modeled at different levels of abstraction that range from low-level metric maps for robot localization and navigation to a conceptual layer that provides a human-like space decomposition and categorization. An illustration of the model and the main levels of representation is presented in Figure 5.1.

The lower layers of the model are derived from sensory input. These layers combine a metrical, line-based representation of spatial structure modeling occupied space, and a navigation graph of virtual markers modeling free space. Different methods are used to gradually construct more abstract representations. On higher levels, we regard topological regions and spatially situated objects as the primitive entities of spatial conceptualization. The robot must be able to assign human concepts to such spatial entities in order to meaningfully act in, and talk about, an environment. Many places in indoor environments are designed in a way that makes their structure, general appearance, and spatial layout afford specific actions; corridors and staircases are examples of this. Other places afford more complex actions provided by objects that are located there. For instance, the concept of a living room applies to rooms that are suited for resting. Having a rest, in turn, can be afforded by certain objects, such as couches or TV sets. The representation allows for combining cues provided by the basic geometrical shape, general visual appearance, perceived objects, and possibly situated dialogue to provide reliable semantic descriptions of space.

The rest of this section provides an overview of each of the layers of the spatial representation.

5.3.1 Metric Map

The lowest level of the spatial model is represented by a metric map. The map encodes spatial boundaries in the environment using lines as basic spatial primitives and supports self-localization of the robot. It is anchored to a metric world co-ordinate system, which is also used as a basis for the higher level representations. The positions of lines as well as of a robot on the metric map are established and maintained by a module for Simultaneous Localization

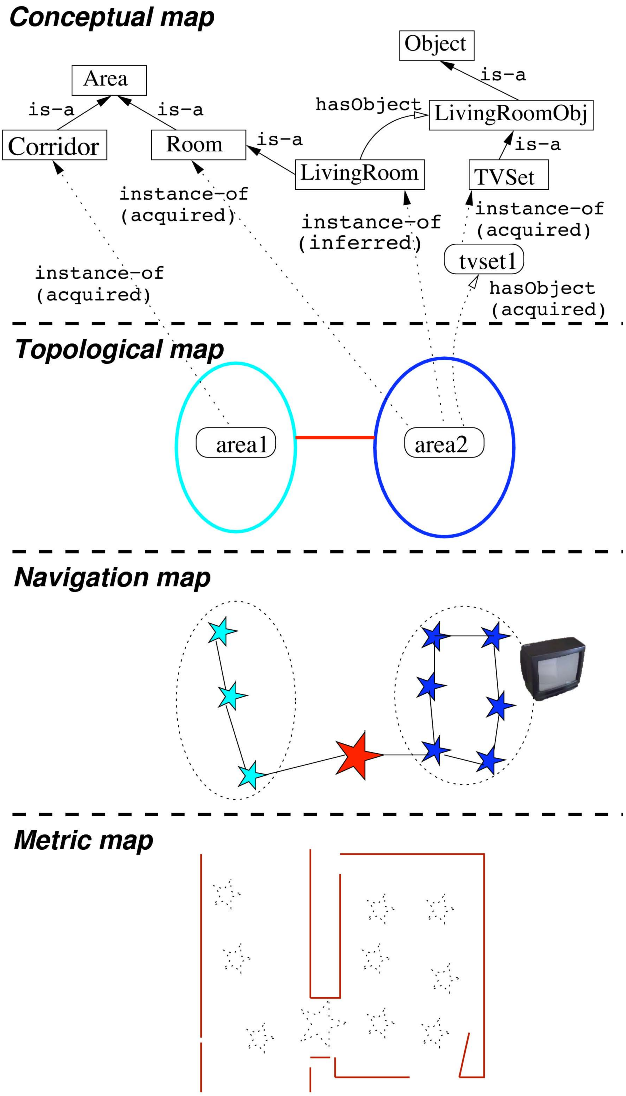


Fig. 5.1. The multi-layered structure of the conceptual spatial model

and Mapping (SLAM) [14]. Section 5.4 gives more details about the applied SLAM algorithm and other approaches that have been investigated.

In comparison to a representation based on an occupancy grid [21], the line-based map does not directly provide a description of the free space but rather of the surfaces in the environment that can be described by lines. However, since the global co-ordinate system of the metric map is purely internal to the robot and humans are not able to easily evaluate quantitative spatial descriptions, the metric map alone is not sufficient to support human-robot dialogues.

5.3.2 Navigation Map

The navigation map provides the next layer of representation, which establishes a model of free space and its connectivity, i.e. reachability. It is based on the notion of a roadmap of virtual free-space markers as described in [22, 23] and implemented as a graph of nodes that are anchored to the metric map. As the robot navigates through the environment, a marker or navigation node is dropped whenever the robot has traveled a certain distance from the closest existing node. Nodes are connected following the order in which they were visited. More information about the navigation graph can be found in Section 5.5.

It is also in the navigation graph that the spatial representation is augmented with semantic information about the environment. First, the semantic category of a place is extracted using a place classification algorithm [12] and stored in the nodes. Doors detected in the environment are represented as doorway nodes and added to the graph. Finally, objects detected by an object search component [13] are also stored on this level of the map. Section 5.7 and Section 5.8 present the algorithms used to detect objects and extract semantic place information from the geometry and appearance of the environment.

5.3.3 Topological Map

The navigation map provides a basis for further, topological abstractions. A topological map consisting of connected areas is built by segmenting the navigation graph into interconnected sets of nodes separated by recognized doors (doorway nodes). This layer of abstraction corresponds to human-like qualitative segmentation of an indoor space into distinct regions (e.g. rooms). On this level, semantic place information extracted and accumulated over entire regions is evaluated to determine appropriate semantic categories for areas in the topological graph. More information about this process can be found in Section 5.5.

5.3.4 Conceptual Map

On the highest level of abstraction, the system is endowed with a conceptual map. The conceptual map builds up a further interpretation of spatial organization. The topological areas together with their place categories form the basic spatial entities. A description logic-based reasoner is used to infer more fine-grained semantic information for the areas. The reasoner integrates knowledge about areas and observed objects with a common-sense ontology of an indoor environment. This ontology represents a taxonomy of areas and objects and the relations between objects and areas. Since there is a strong connection between typical objects found in an area and the semantic category of the area, this layer can also be used to constrain expectations about which objects are likely to be observed, given that the basic-level concept of an area is known (for example through a situated dialogue with a human user). Section 5.6 provides more details about the conceptual map.

5.4 Metric Mapping

This section gives details about the metric mapping algorithms that were investigated in CoSy and used to maintain the metric representation in the spatial model. Metric maps can be represented in many ways. The two most common approaches are based on occupancy grids [24, 25, 26, 27, 28, 29] and features [30, 31, 32, 33, 14]. Occupancy grids discretize the world into cells. Each grid cell holds a value representing the probability that the corresponding area in the environment is occupied. Feature-based maps on the other hand abstract the sensor data into a set of features. In a structured environment – of which most office environments are examples – lines, corners and edges are common features. The features can be parameterized by, for instance, their color, length, width, position, etc. One of the main advantages of this type of representation is that it requires very few assumptions about the world, whereas one has to settle on a set of features to parameterize the map beforehand.

Mapping (building a model of the environment) and localization (finding the position in the environment) are often treated as two separate problems. Maps were made assuming that the position is known and the position is calculated given a map. However, for an autonomous agent that explores an unknown environment these two task are intrinsically linked, and form a chicken-and-egg problem; to perform mapping one needs the position and to perform position one needs the map. This leads to Simultaneous Localization and Mapping (SLAM) which has been a thriving research area for more than a decade.

In this chapter we will focus on feature-based representations. A feature-based map can in general be written

$$\mathcal{M} = \{f_j \mid j = 1, \ldots, M\}, \tag{5.1}$$

where f_j is a feature and M is the number of features in the map.

5.4.1 M-Space

A number of different types of features have been used for mapping. Depending on the type of application the model of the environment is 2D or 3D. For most indoor applications a 2D representation is used; navigation in cluttered environments often requires a 3D representation. When taking the step outdoors the world is less structured and it becomes increasingly likely that the ground is not flat which also calls for a 3D model.

To motivate the work with the so called M-Space representation, let us first consider how to represent a line segment. Such a line segment could for example offer a 2D abstraction of a wall in an indoor environment. Four parameters are needed to fully specify the line segment. On a real robot, the sensors may not be able to detect its end points; even if the end points are within the range of the sensors they are often hard to detect due to occlusions. This implies that a measurement typically only constrains the position of the sensor to a certain distance and relative angle with respect to the wall. In other words, all dimensions of a wall are typically not constrained by one single measurement.

There are a number of ways to represent a line segment. To name a few; slope and intersection ($y = k_y x + m_y$), end points, distance and direction (infinite line [34]), center point, length and orientation. All of these suffer one or more problems, some of which are addressed by the so called SP-model [35].

A characteristic property of SP-model is that each feature element has its own local reference frame. The frame of reference is chosen with the axes along the directions of symmetry. A line, for example, has the x-axis along the direction of the line. A plane will have a normal that coincides with the z-axis. The main advantage of using a local frame is that the description of the uncertainty can be made independent of the global position of the features. This avoids lever-arm effects that can result when for example using direction and orientation to represent a line. The local frames also help to make frame transformations and differentiations thereof more standardized. Another key concept in the SP-model is the so called *binding matrix*, B. The binding matrix is a row-selection matrix. The self-binding matrix selects the DOFs that are not part of the motion symmetry, i.e. the DOFs of a feature that are constrained and have probabilistic information attached to them. The binding matrices offer a machinery for making partial observations of a feature. This is useful, for example, when observing a single point on a line. A limitation with the SP-model is that one has to attach a frame to all features. For some types, such as lines, it is difficult to model the extent, e.g. the length, in a probabilistic way within the SP-model framework. In [36], the length of lines is estimated and modeled but it relies on always detecting both end points at the same time and making a *direct measurement* of the length. An *indirect measurement* is not possible as the origin of the reference frame cannot be observed, not being attached to anything observable, but just defined to be in the middle of the line.

The so called M-Space representation builds on the SP-model[1]. It also attaches a local frame to each feature element and allows for a generic treatment of many types of features. The measurement subspace, or M-space, is

[1] For a detailed description see [14] from which this text is derived.

an abstraction of the measured subspace of the feature space that reflects symmetries and constraints. The idea is that the features are parameterized to fully specify their location and extent (the feature space) but that they can be initialized in a subspace corresponding to the information provided by the sensors.

For example, when representing a line segment the extent is accommodated for in the representation even though only the distance to and the orientation of the line is known initially. We cannot represent the uncertainty with regard to changes in the coordinates along the length of the line by a Gaussian distribution. However, the uncertainty regarding changes perpendicular to the line and regarding the orientation can be approximated by a Gaussian. Let $\delta\mathbf{x}_p$ denote the M-space corresponding to a small change in feature coordinates $\delta\mathbf{x}_f$. Here the subscript, p, stands for small perturbations in the M-space. The actual values of the M-space coordinates, $\mathbf{x}_p$, are never needed or considered. It is only the changes to them that enter into the estimates. These changes are used to make adjustments to the feature coordinates $\mathbf{x}_f$. The uncertainty estimate is an estimate of the distribution of $\delta\mathbf{x}_p$ values around a mean of zero. The adjustments to the feature coordinates are made to maintain this zero-mean. No re-centering step like in the SP-model is required with this view of the uncertainty. The uncertainty is defined in a frame attached to the feature and can be projected into the global frame using the current global coordinates of the feature. The statistics are represented in an analytic way rather than in the strict geometric sense of the SP-model. In most cases, the differences are in the second order corrections to the covariances.

The relation between the feature space coordinates and the M-space coordinates is defined by a projection matrix, $B(\mathbf{x}_f)$, similar to the binding matrix in the SP-model. The projection matrix relates small changes $\delta\mathbf{x}_p$ to small changes $\delta\mathbf{x}_f$. An important difference to the binding matrix is that the projection matrix is a function of the individual feature and changes with time. The rather involved re-centering step in the SP-model is replaced by re-evaluating the projection matrices.

A common issue in feature-based SLAM is that one cannot initialize a feature after the first observation. A single observation typically does not contain enough information to do so reliably. Among the reasons behind this we find for example

- The entire feature is not detected at once.
  - In the case of a line segment, the end points might not have been detected if the line is partially occluded or long.
  - When using monocular vision, only the bearing to the feature can be initialized from a single image.
- Measurements are noisy. Even though a feature is fully observed it is good practice to get a second opinion from new measurement data to reject false measurements.

The M-space representation offers a solution to these problems by allowing the M-space dimensionality to change over time. Features are typically initialized with zero M-space dimensions and with time, as more information is gathered, more dimensions will be added. Consider mapping a wall as a line segment. The life cycle of the line segment might be

1. First detection: feature initialized with 0 M-space dimensions.
2. Re-observed N times: the wall's existence and quality are confirmed. The distance and orientation of the wall added to the M-space.
3. Start point detected: M-space dimensionality goes up to 3
4. End point detected: the feature reaches full dimensionality of 4.

The importance of the ability to let the dimensions of a feature grow over time is well illustrated by a horizontal line feature observed by a camera. A single image does not contain information to pinpoint the location of a feature. The assumption that the line feature is horizontal implies that a single observation will be enough to provide information about the relative orientation of the robot. That is, even if the robot moves parallel under the line and is unable to use triangulation to fix the position of the line in space the observations of the line can help reduce the angular uncertainty of the robot. This is useful in, for example, a corridor where the motion often is parallel to the linear structures found in the ceiling.

5.4.2 Single Camera Bearing Only SLAM

Range sensors such as laser scanners are still the most common sensors for systems that perform mapping and localization in settings where robustness is key e.g. in industrial applications. However, cameras are becoming increasingly interesting as the performance keeps increasing while the price keeps going down due to the large demand from the consumer market e.g. for mobile phones. One of the main advantages of a camera over a range sensor is that the information that it provides is so much richer and not limited to a few hundred distance measurements typically lying in a plane. The hard part is to get the information out from the image.

There is a rich literature on single camera bearing-only SLAM. Most of these use point features extracted from the image to define landmarks in the map [37, 38]. Given standard feature detectors such as SIFT [39], there can be several hundreds of features and thus potential landmarks per frame. A single SIFT descriptor is not discriminative enough in itself, especially in man-made environments where structures like corners give raise to many SIFT points with very similar descriptors. When used for object recognition [39] it is a combination of descriptors extracted from the object that provide the discriminative strength. This idea is used in the vSLAM approach [40] where the SIFT points are used to recognize places. In [41], we present a framework where a few stable (over time and in space) SIFT features are identified and used as landmarks (location features). The rest of the SIFT features are

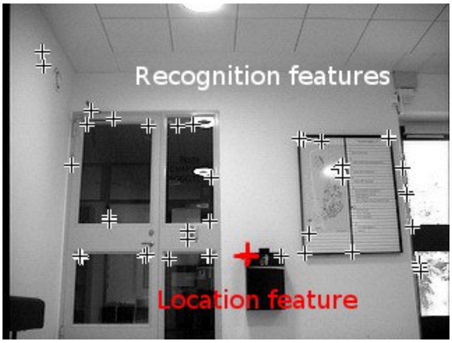


Fig. 5.2. A few stable features are identified and used to define the location of landmarks. The rest of the features are used to improve the matching/recognition of these.

used to strengthen the matching of features (recognition features). This is illustrated in Figure 5.2. When matching against a landmark, the matching is performed not only with the feature defining the position of the landmark but also the rest of the feature extracted from the two frames. This greatly improves the robustness of the matching.

5.4.3 Using Visual Attention for SLAM

Choosing useful landmarks which are easy to track, stable over several frames, and easily re-detectable when returning to a previously visited location is important in order to get a visual SLAM system working. Getting few but good, rather than many and bad landmarks reduces the issue of complexity. In [42, 43, 44], we suggest the application of a biologically motivated attention system [45] to find salient regions in images. Attention systems are designed to favor regions with a high uniqueness such as a red fire extinguisher on a white wall. Such regions are especially useful for visual SLAM because they are discriminative by definition and easy to track and re-detect. We show that salient regions have a considerably higher repeatability than Harris-Laplacians and SIFT key-points. Active gaze control is also used and has been shown to enhance the performance over using a statically mounted camera. The strategy to steer the camera consists of three behaviors: a *tracking* behavior identifies the most promising landmarks and prevents them from leaving the field of view. A *re-detection* behavior actively searches for expected landmarks to support loop-closing. Finally, an *exploration* behavior investigates regions with no landmarks, leading to a more uniform distribution of landmarks. The advantage of the active gaze control is to obtain more informative landmarks (e.g. with a better baseline), a faster loop closing, and a better distribution of landmarks in the environment. Figure 5.3 shows the difference in the robot pose uncertainty when driving the same trajectory with active camera control

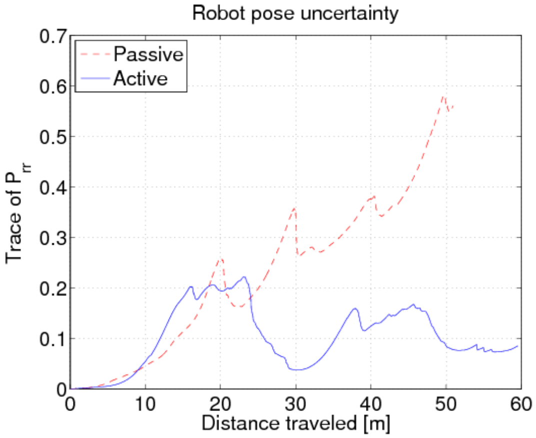


Fig. 5.3. A comparison of the robot pose uncertainty with camera control (active) or without (passive)

and without. This example illustrates that the active camera control allows the robot to reduce the uncertainty by seeing landmarks that it could otherwise not have seen. For more details please refer to [43, 44].

5.4.4 Visual Scans

One of the more popular and successful ways to do metric mapping is to use scan matching [46, 47, 29]. In a feature-based setting the scan can be considered to be a feature which is defined by the scan distances themselves. Building the map boils down to finding the position from which the scans were acquired in such a way that the laser scans align and for a consistent map.

In [48] we present an idea to use so called visual scans in much the same way as laser scans are used in scan matching. Using a stereo camera, a 3D point cloud is calculated by extracting and matching SIFT features in each image. This 3D point cloud forms the visual scan. Aside from the position, each point also has a descriptor saying something about the appearance which can be used for matching.

The map is defined by a number of reference scans. A reference scan is added when there is not enough overlap between the current visual scan and any of the other visual scans. The advantage of this representation is that it gives a very rich description of the environment (a dense point cloud which can have hundreds or even thousands of points) while the estimation problem only needs to deal with the parameters defining the position of the sensor (3 parameters in 2D and 6 in 3D, compared to $3N$ with N points treated independently). An example is shown in Figure 5.4.

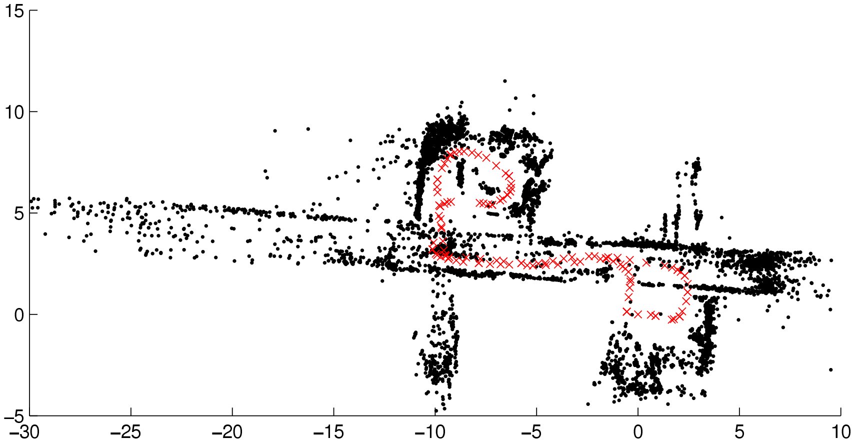


Fig. 5.4. A map with 126 visual reference scans. Together these scans contain 8333 points. This shows how the visual scans help define a dense representation at the same time as providing a low-dimensional estimation problem.

5.5 Navigation and Topological Maps

Above the bottom layer – the metric map – the spatial model contains the navigation and topological maps. As explained above, the metric map encodes boundaries in the environment and is used to ensure safe and reliable navigation and obstacle avoidance. In contrast, the navigation and topological layers encode more abstract information about the space accessible to the robot, particularly important from the functional point of view. This information is encoded in the form of graph-like structures in which the links represent connectivity between spatial entities at different spatial scales. The graph constituting the navigation map consists of nodes representing small unbounded free space regions in the environment. The topological representation, in its turn, models the indoor environment in terms of larger bounded areas connected by detected doors.

The representations play two main roles in the system. First, they discretize the continuous free space into a finite number of spatial units. These units are then used for tasks such as planning or interaction with the robot. For example, the high level task "go to the office", can be translated to the low level action "go to the closest navigation node attached to the topological node representing the office". If this position is occupied the robot can choose to go to the next navigation node. Discretization of space drastically reduces the number of combinations that have to be considered during the planning process.

The second important function of the navigation and topological representations is preserving additional information about the surroundings. Here,

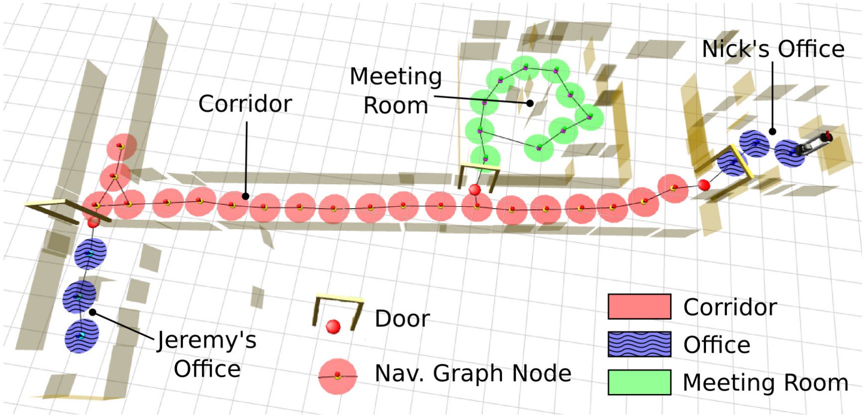


Fig. 5.5. An example of a navigation map overlayed on a metric map. The free space navigation nodes are represented by circles and are assigned to different topological areas based on the separation established by the doorway nodes. The colors of the nodes indicate the functional category of the areas as recognized by a place classification algorithm.

semantic information about places extracted from the sensory input is accumulated and stored together with navigation and topological nodes. Moreover, information about objects found in the environment is tied to nodes in the navigation graph. This is used to link the representations with the conceptual map and allows to refer to places in terms of their functional category or detected objects. As an example, the order "go to the TV", can be processed and translated into "got to the closest node from which you saw the TV".

The rest of this section focuses on the process of generating the navigation and topological representations based on the information encoded in the metric map and additional cues extracted from the sensory input.

5.5.1 Building the Navigation Graph

The navigation map provides the first discretization of the continuous space described by the model. It is represented in the form of a graph built as the robot explores the environment and is based on the notion of a roadmap of virtual free space markers [22, 23]. A free space navigation node is dropped whenever the robot has traveled a certain distance from the closest existing node (approximately 1 meter). Each node is anchored to the metric map and is assigned (x, y) co-ordinates. Nodes are connected following the order in which they were generated. This order is given by the trajectory that the robot follows during the map acquisition process. The final graph serves for planning and autonomous navigation in the already visited part of the environment. An example of a navigation graph overlayed on a metric map is presented in Figure 5.5. This simple representation proved to be very powerful during real-world experiments with the integrated system.

As the robot navigates through the environment, additional information about the surroundings collected on the way is assigned to the closest navigation nodes. Moreover, special doorway nodes are added to the navigation graph at the points where doors are detected in the environment (see Figure 5.5). As explained below, these nodes play an important role in building the spatial model.

5.5.2 Space Segmentation and Topological Graph

The structure of indoor environments allows for introducing larger scale and more abstract representations than the navigation graph. In such environments a room is an important concept. Different rooms can be associated with different owners and functionalities. Moreover, rooms are spatial entities commonly referred to in the natural language. The ability to segment space into rooms becomes crucial for an artificial mobile cognitive system. Therefore, as another layer, the spatial model builds a topological graph consisting of areas and links which represents rooms in the environment and their connectivity.

The structure of the topological graph is built based on the assumption that the transition between two areas happens through a door. This creates a human-like qualitative segmentation of an indoor space into distinct regions. A door detection algorithm is used to generate doorway nodes in the navigation graph whenever the robot passes through a narrow opening. The width of the opening is selected so that it matches typical doorways in the environment. Information about the door opening, such as width and orientation, is stored along with the detected position of the doorway in the doorway node. The doorway nodes are used to segment the navigation graph and assign navigation nodes to areas in the topological graph.

More complex door models such as those in [49, 50] can be used for more robust door detection. However, such models put additional constraints on how doors have to look to be recognized. The only assumption in the model described here is that the door is a narrow opening which the robot passes through. No assumptions are made regarding the door leaf (e.g. swinging or sliding) or special structure around the door. This can be beneficial for a robot that has to operate in different environments. An alternative would be to use a learning approach, such as in [51], where both visual features and the motion of the door are taken into account.

5.5.3 Adding Object Information

Objects and landmarks play an important role in understanding spatial structure. They are important cues that often determine the actions that can be performed in a particular area. Moreover, objects are nameable features commonly used in describing spatial locations. For this reason, visual search algorithms are used in the system to perform autonomous exploration and detection of objects typically found in indoor environments. Detailed information about the algorithms applied for this purpose can be found in Section 5.7.

The presented spatial model incorporates information about objects. This information is later used by the last conceptual layer. First, however, the objects must be tied to their spatial locations. This is the role of the object nodes which are connected with the navigation nodes of the navigation graph. The object nodes store information about the type of the recognized object and its metric location. The nodes are then linked to the closest navigation node.

5.5.4 Adding Semantic Place Information

Many places in an indoor environment can be characterized by semantic categories corresponding to their inherent functionality. Rooms constitute a good example as they can be categorized as offices, kitchens, meeting rooms etc. However, semantic descriptions can also be assigned to smaller regions such as a printer area in a corridor. The semantic place category is usually reflected in the objects located in that place, but also in the general appearance and geometrical layout.

One of the roles of the navigation nodes and topological areas in the spatial model is to store semantic information about the places to which they correspond. This information is used to link the lower layers modeling spatially allocated regions with the spatial concepts of the conceptual layer. A specialized component performing multi-modal place classification is used to extract semantic descriptions from the sensory input of a robot. Visual and laser range sensory data acquired in an environment are analyzed and compared to place models in order to produce beliefs about the place categories. In the simplest case, the component can be used to distinguish between two basic place categories: a corridor or a room (e.g. based on the clutteredness and geometric layout). Further specialization can be performed in the conceptual layer based on object information or situated dialogue. However, more specific place categories can also be recognized directly by the place classification system. More information about this process can be found in Section 5.8.

As will be shown through experiments in Section 5.9, the place classification system is able to classify a place with high accuracy given a single data sample (e.g. one image and laser scan) corresponding to only one viewpoint. However, in this case, the task is to provide a reliable and stable label for the whole region covered by a navigation node or a topological area. Since the sensors employed are not omnidirectional, it is necessary to accumulate and fuse the incoming information. However, the data that the robot gathers are not evenly spread over different viewpoints. On the contrary, in many cases the sensors receive a continuous stream of non-informative data (e.g. when the robot is parked close to a wall blocking the view). The system must be able to deal with such problems as temporary lack of informative cues, long-term occlusions or large variability affecting certain viewpoints.

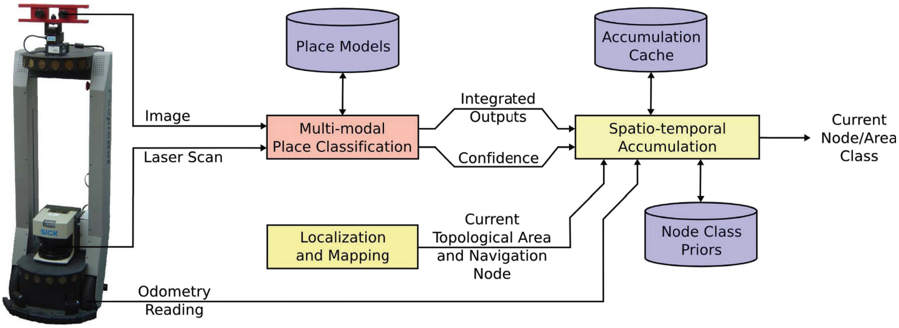


Fig. 5.6. Generating semantic labels for navigation nodes and topological areas using multi-modal place classification

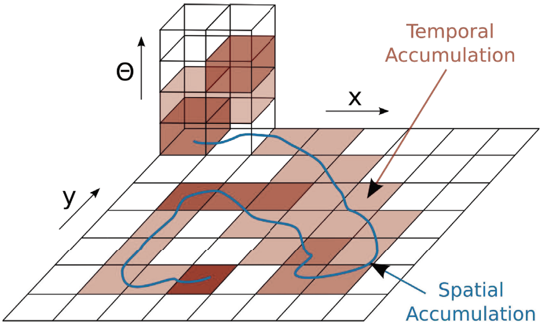


Fig. 5.7. Illustration of the spatio-temporal accumulation process. As the robot explores the environment, the beliefs collected on the way accumulate over time within the bin corresponding to the current pose (x, y, θ) and over space in different bins.

For this reason, the place information produced by the place classification algorithm is accumulated over time and space as presented in Figure 5.6. For each multi-sensory data sample, place classification provides a set of beliefs encoded as a vector of real-valued outputs (see Section 5.8 for details). The confidence of the final decision is also measured and provided by the classification component. The beliefs are fused using a confidence-based spatio-temporal accumulation algorithm. The algorithm relies on information about the current position on the navigation and topological map provided by the localization and mapping system. The spatio-temporal accumulation process is performed within the region covered by the current node and area. When the robot moves to a different node, the collected information is used to update the semantic label attached to the map and saved as a future prior. When the robot enters a location which was already explored, the previously stored beliefs are loaded and can be refined by further exploration.

The principle behind the confidence-based spatio-temporal accumulation algorithm is illustrated in Figure 5.7. As the robot explores the environment, it moves with a varying speed. The robot has information about its own movement provided by the wheel encoders (odometry). As errors accumulate over time, this information can only be used to estimate relative movement rather than absolute position. Although the accurate metric information could be used instead, odometry is sufficient for our application. The spatio-temporal accumulation process creates a sparse histogram along the robot pose trajectory described by the metric position (x, y) and heading (θ). The size of the histogram bins is adjusted so that each bin roughly corresponds to a single viewpoint. Then, as the robot moves, the beliefs about the current semantic category accumulate within the bins. An average of the outputs is calculated in a manner similar to the Discriminative Accumulation Scheme (DAS, [10]) used in the framework of cue integration. This is what we call the temporal accumulation. It prevents a single viewpoint from becoming dominant due to long-term observation. Since each viewpoint observed by the robot will correspond to a different bin, performing accumulation across the bins (this time spatially) allows for generating the final outputs to which each viewpoint contributes equally. In order to exclude most of the misclassifications before they get accumulated, the decisions are filtered based on the confidence value provided by the place classification component. Finally, the best hypothesis is calculated. It is assigned to the navigation node and topological area representing the spatial region over which the accumulation was performed.

5.6 Conceptual Map

The conceptual map provides the link between the low-level maps and the communication system used for situated human-robot dialogue by grounding linguistic expressions in representations of spatial entities, such as instances of rooms or objects. It is also in this layer that knowledge about the environment stemming from other modalities, such as object recognition and dialogue, is anchored to the metric and topological maps.

Based on the work by Zender [52], our system is endowed with a commonsense OWL ontology of an indoor environment (see Figure 5.8) that describes taxonomies (*is-a* relations) of room types and typical objects found therein through *has-a* relations. These conceptual taxonomies have been handcrafted and cannot be changed online. However, instances of the concepts are added to the ontology during run-time. Through fusion of *acquired* and *asserted* knowledge gathered in an interactive map acquisition process [53] and through the use of the *innate conceptual* knowledge, a reasoner can *infer* information about the world that is neither given verbally nor actively perceived. This way linguistic references to spatial areas can be generated.

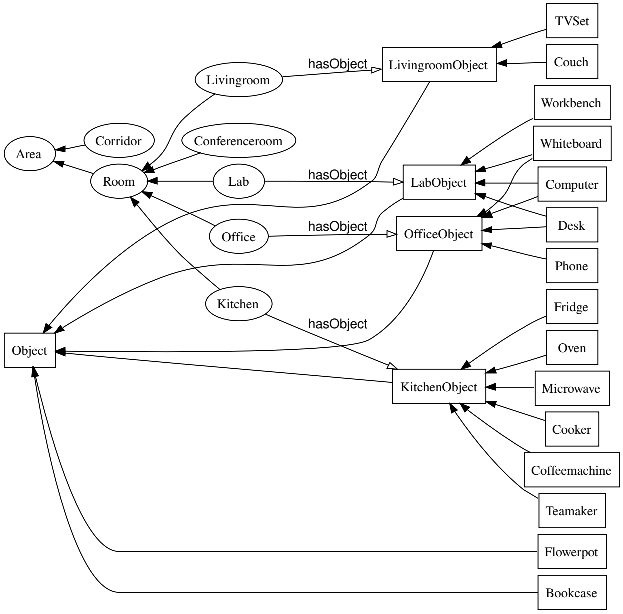


Fig. 5.8. Illustration of a part of the commonsense ontology of an indoor office environment. Solid arrows denote the taxonomical `is-a` relation.

Acquired Knowledge

While the robot moves around constructing the metric and topological maps, our system derives higher-level knowledge from the information in these layers. Each topological area, for instance, is represented in the conceptual map as an ontological instance of the type `Area`. Furthermore, as soon as reliable information about the semantic classification of an area is available, this is reflected in the conceptual map by assigning the area's instance a more specific type. Information about recognized objects stemming from the vision subsystem is also represented in the conceptual map. Whenever a new object in the environment is recognized, a new instance of the object's type, e.g. `Couch`, is added to the ontology. Moreover, the object's instance and the instance of the area where the object is located are related via the `hasObject` relation. This process is shown in Figure 5.1.

Asserted Knowledge

During a guided tour with the robot, the user typically names areas and certain objects that he or she believes to be relevant for the robot. Typical assertions in a guided tour include "You are in the corridor," or "This is the charging station." Any such assertion is stored in the conceptual map, either by specifying the type of the current area or by creating a new object instance of the asserted type and linking it to the area instance with the hasObject relation.

Innate Conceptual Knowledge

We have handcrafted an ontology (Figure 5.8) that models conceptual commonsense knowledge about an indoor office environment. On the top level of the conceptual taxonomy, there are the two base concepts Area and Object. Area can be further partitioned into Room or Corridor. The basic-level subconcepts of Room are characterized by the instances of Object that are found there, as represented by the hasObject relation.

Inferred Knowledge

Based on the knowledge representation in the ontology, our system uses a description logic-based reasoning software that allows us to move beyond a pure labeling of areas. Combining and evaluating acquired and asserted knowledge within the context of the innate conceptual ontology, the reasoner can infer more specific categories for known areas. For example, combining the acquired information that a given topological area is classified as a room and contains a couch with the innate conceptual knowledge given in our commonsense ontology, it can be inferred that this area can be categorized as being an instance of LivingRoom. Conversely, if an area is classified as a corridor and the user shows the robot a charging station in that area, no further inference can be drawn. The most specific category the area instantiates will still be Corridor.

Our method allows for multiple possible classifications of any area because the main purpose of the reasoning mechanisms in our system is to facilitate human-robot interaction. The way people refer to the same room can differ from situation to situation and from speaker to speaker, as reported by Topp *et al.* [54]. For example, what one speaker prefers to call the kitchen might be referred to as the recreation room by another person. Since our aim is to be able to resolve all such possible referring expressions, our method supports ambiguous classifications of areas.

5.7 Object Detection and Recognition

In this section, we discuss how the robot can use active vision for perceiving objects and landmarks in the environment. The process is active in that it is based on active search, primed by interpretations established at other levels of spatial representation. Active vision provides information about objects in the environment. It covers object recognition and determines object pose relative to the world coordinate system adopted by the metric layer in which all other representations are grounded.

The rest of this section gives details about the approach adopted in CoSy for object search and localization (Section 5.7.1) and presents results of an experiment evaluating different methods for object distance estimation (Section 5.7.2).

5.7.1 Object Search and Localization

In our early work [55], the search for objects was performed while exploring the environment to cover space or when guided by the user. This is not sufficient. The user is able to show the robot all objects, but this is a tedious task and does not have the right level of autonomy expected from an autonomous agent. When object search runs in parallel with exploration, it is driven by the laser scanner which has a 180° field of view compared to the camera having about a fourth of that. This means that the camera is not guaranteed to see all parts of the environment.

View planning as a research area is well established. The so called *art gallery problem* [56] is defined as finding a minimal set of viewpoints from which all the parts of the environment can be observed. This problem is akin to the problem of planning for finding objects in the environment. The main difference is that one also needs to take into account the limitations of the observer, i.e. the camera. One of the most important limitations comes from the finite resolution of the camera and the fact that objects have different sizes. Even if a small object is in the field of view it will not be detected if the camera is too far away. Similarly, a large object can typically not be detected if the camera is too close. A system taking these constraints into account is presented in [57, 13]. This system uses a combination of a visual attention mechanism in the form of the RFCH algorithm [58], camera zoom and SIFT recognition [39] for finding the objects. View planning is carried out by selecting views from the nodes in the navigation graph. Briefly, when searching for objects the system first analyzes the map of the environment and performs the view planning. The robot then visits each view point and performs the visual search. The visual attention mechanism tells the systems what parts of the image to investigate further and the system does so by zooming in, thus gathering more pixels from the potential object. When the object is close, SIFT recognition is used to verify the identity of the object. The distance to the object is estimated from visual cues. The distance is used to control the zooming and to estimate the position of the object.

This method relies on the visual attention system not to produce too many views to investigate further. In [59] a method for accumulating over time the available visual evidence for the presence of objects was investigated. This would allow the object recognition algorithm to run in parallel to the exploration and could also handle object detection/recognition algorithms that provide information that is too weak to act on immediately.

The input to the view planning is, besides the objects to search for, a grid representation of the world and a navigation graph. The grid resolution is 0.5m [13]. The grid cells represent possible object locations, and the plan is constructed such that each grid cell is observed from a distance appropriate

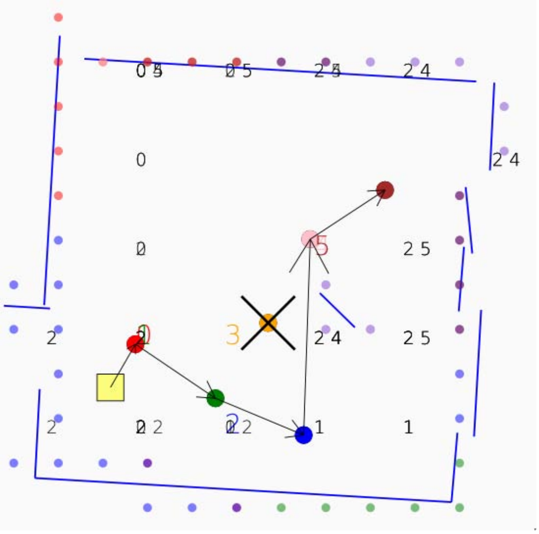
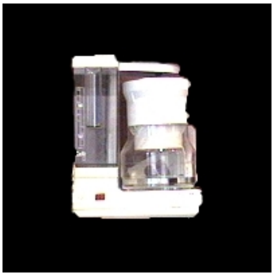


Fig. 5.9. An example of a planned path for object search. In this example, one of the nodes is not used in the plan.



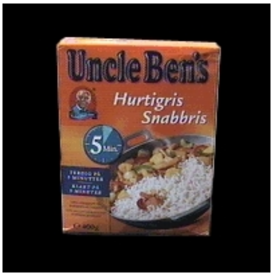


Fig. 5.10. Two of the training images (coffee machine and rice package) provided to the robot beforehand to learn the appearance of the objects

for each object. Figure 5.9 shows an example of a plan from the view planning. There may be several views associated with a node corresponding to different viewing directions.

To represent an object for recognition, the system uses one segmented image of each object in a close-up view. Two examples of training images are shown in Figure 5.10. To be able to plan for detecting the objects and to determine the distance to a detected object, the real world and image size of the object are also stored in the object database.

As a consequence of using a single view for each object, the system can only recognize the object from one side. Using a multi-view representation of the object is a natural extension to this work.

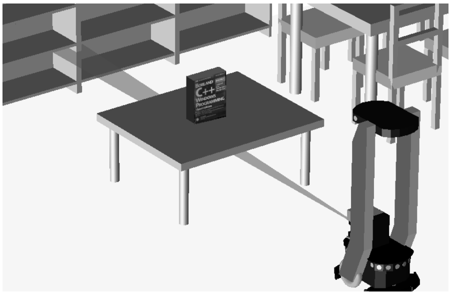


Fig. 5.11. Distance estimation provided by the laser may not be reliable: instead of the distance to the object on the table, the distance to the shelf is measured

In [60], the distance estimate used to determine zoom levels was based directly on the robot's laser sensor. However, the distance provided by the laser is often misleading, as Figure 5.11 shows: the laser sensor is placed about 30cm above the floor and if an object is not at that height, the estimate may be wrong. The approach works only for objects that are placed on the floor or are located close to walls (for example, in a bookshelf). If the distance estimate is wrong, the final zoom may either not be sufficient to make the object occupy enough of the image, or otherwise may be too large causing only a small part of the object to be seen. Furthermore, even if the object is recognized, its estimated position might be inaccurate. To address these issues, we have looked at two alternative ways for distance estimation.

Using the Vote Matrix

Using the RFCH vote matrix for distance estimation consists of measuring how many cells are part of the object and treating the area they occupy in the image as an approximation of the object's size. Here, cells are considered to be associated with a hypothesis if their degree of match is above the threshold and if there is an 8-connected path to the hypothesis with cells of monotonically increasing value. Only the strongest hypothesis and its associated 8-connected cells are taken into account, because it is likely to be the most reliable.

Given the object's actual size stored in the training database, the distance is then computed as:

$$D = \frac{W_{real}\dfrac{W_{im}}{2D_{vote}}}{\tan\left(\dfrac{\alpha}{2}\right)}$$

where D stands for the estimated distance (meters); W_{real}, for the real width of the object (meters); W_{im}, for the width in pixels of the camera image; D_{vote}, for the width in pixels of the bounding box of the cells associated with a hypothesis and α, the horizontal viewing angle. This procedure is fast and approximate, but sufficiently accurate to allow the object search algorithm to assign a valid zoom.

Using SIFT

SIFT produces a scale parameter for each key point extracted. For each matched pair of key points in the training and recognition image, the quotient of the keys' scale parameter gives an estimate of their relative apparent size and hence their distance, according to:

$$D = \frac{W_{real} \frac{W_{im}}{2W_{tr}} \frac{S_{tr}}{S_{real}}}{\tan\left(\frac{\alpha}{2}\right)}$$

where S_{tr} denotes the scale of the point extracted from the training image; S_{real}, the scale of the point extracted from the recognition image, and W_{tr}, the width of the object in the training image in pixels.

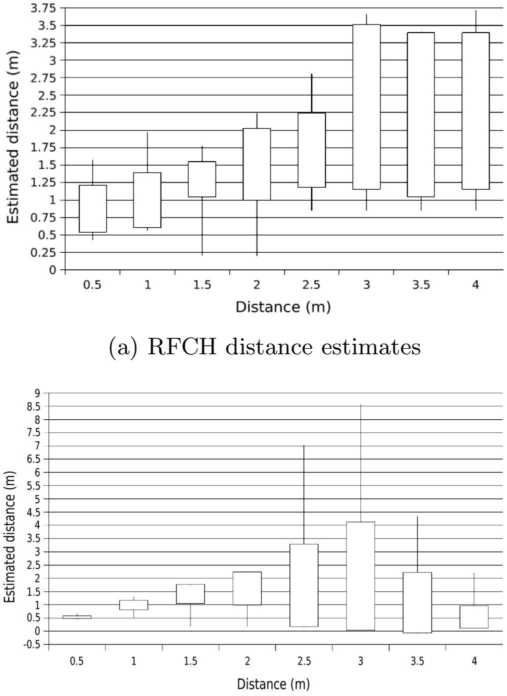
As mismatched key point pairs can produce incorrect scale parameters, the final estimate of the object distance is taken as the median of the distance estimates from all matches. Experiments indicate that an adequate estimate is obtained given 10 or more SIFT matches. With 4 matches or more a passable rough estimate is typically obtained (within about 30%). If there are fewer than 4 matches, the result is likely to be very poor (most likely based on some other structure than the object) and is not used.

The drawback of the above method is that extracting SIFT features from an image is computationally expensive, and using it to guide the zoom process may take too long to be feasible. Another problem is the number of SIFT features required to obtain a robust estimation; when the object is small in the image (i.e. resolved by few pixels), it is unlikely that enough matches will be available.

5.7.2 Object Distance Estimation

Figure 5.12 presents the results of distance estimation using RFCH and SIFT without magnification, performed on five different test objects. As expected, performance deteriorates for both methods at long range, due to the decreased size of the object in the image, and for RFCH also partly to the discretization of the vote cells.

It is notable that the values obtained through both methods tend towards the low end. The reason for this are mainly outliers, erroneously assigned values of 0.5–1m, caused by large background structures being mistaken for a close-up object. Compared to RFCH, SIFT exhibits a far more accurate and dependable estimate at short range. However, its quality rapidly deteriorates



(a) RFCH distance estimates

(b) SIFT distance estimates

Fig. 5.12. Distance estimation results; all objects. Top image RFCH, bottom SIFT. Boxes signify one standard deviation about the average for each distance; lines signify the most extreme values.

at longer distances, as can be seen by inspecting the average value of the estimates beyond 2.5m in Figure 12(b). This is because a certain level of detail is needed to extract SIFT keys. In contrast, RFCH, though most reliable at medium ranges (as demonstrated by the standard deviations in Figure 12(a)), retains the ability at long range to provide very rough approximations, generally adequate for the purpose of selecting a zoom level for the next step. For the final distance estimate, it should be pointed out that SIFT is used – but the magnification of the image will correspond to shifting the diagram in Figure 12(b) into the 0.5m–1m region where the method is most effective.

Figure 5.13 highlights the differences between RFCH and SIFT in distance estimation. Here, for each test image, the absolute error of the distance estimate is compared between the two methods and the percentage of cases where each of the methods gives better estimate is plotted. The graph shows that RFCH becomes more reliable at 2m range or above.

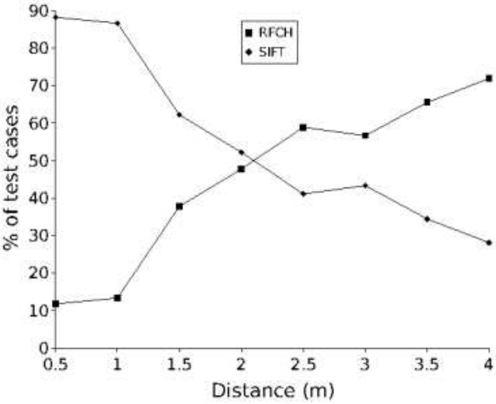


Fig. 5.13. Proportion of instances in which RFCH and SIFT provide the best estimate

5.8 Place Classification

This section presents a multi-modal place classification algorithm able to identify places and recognize semantic place categories. The method effectively utilizes information from different robotic sensors by fusing multiple visual cues and laser range data [12]. The presented approach was used for real-time semantic labeling of the spatial entities represented by the conceptual spatial model.

Place classification, as considered in this section, can be described as a supervised pattern recognition problem of assigning a region in an environment to one of predefined place classes based on multi-modal sensory input and a set of place models. First, the place models are build from a collection of labeled data samples acquired in places belonging to the modeled classes. The models store intrinsic visual and geometric properties of the classes. Then, the algorithm is presented with data samples acquired in one of the same places or in a novel place belonging to one of the same categories, possibly under different conditions and after some time (where the time range goes from some minutes to several months). The goal is to classify correctly as much of the sensory data samples as possible.

The ability to classify places based on their visual and geometric properties is an important competence for a mobile cognitive agent in two fundamental scenarios. First, place classification can be used to recognize previously visited places. In this scenario, place classification becomes a key element of topological and hybrid localization systems, providing them with means for global localization and loop closing [61, 62, 12]. Second, place classification can be used to assign novel places to semantic categories, and thus augment space representations with semantic information [9, 7, 63]. In both cases, this is a challenging classification problem due to large variability and dynamics of real-world environments. First, viewpoint variations cause the sensors to capture different aspects of the same place, which often can only be learned if enough training data are provided. Moreover, real-world environments are usually dynamic and their appearance changes over time. The recognition system must be robust to variations introduced by changing illumination (e.g. during sunny days and at night) and due to human activity (people might appear in the images, objects and furniture can be relocated).

Place classification is a widely researched topic. Purely geometric solutions based on laser range data have proven to be successful for certain tasks [7, 64, 61]. However, the limitations of such solutions inspired many researchers to turn towards vision which nowadays is becoming tractable in real-time applications. The proposed methods employed either perspective [9, 65, 66, 10, 62] or omnidirectional cameras [67, 68, 69, 70]. The main differences between the approaches relate to the way the scene is perceived. Several approaches employ local features, computed from distinct parts of an image [66, 69, 70]. Other use global features, derived from the whole image [67, 68, 9]. Recently, several authors observed that robustness and efficiency can be improved by combining information provided by different visual cues [10, 62] or different sensors, such as a camera and a laser range finder [11, 71, 12].

The algorithm presented here is able to perform robust place classification under different types of variations that occur in indoor environments over a span of time of several months. The method relies on robust descriptors [72, 39, 7] and discriminative classifiers [73, 74] known for their superior generalization abilities. The reliability is further improved by integrating multiple cues and modalities. The system uses different types of visual information provided by global and local image descriptors and geometric cues derived from laser range scans. The cues are combined using a high-level cue integration scheme that learns how to optimally weight each cue [12]. The system is able to measure its own level of confidence and fuse information over time and space in order to provide a reliable decision. Finally, in case of dynamic environments, where the long-term variability cannot be handled by the generalization abilities of the algorithm, the internal representation can be incrementally updated to maintain a stable performance as proposed in [75].

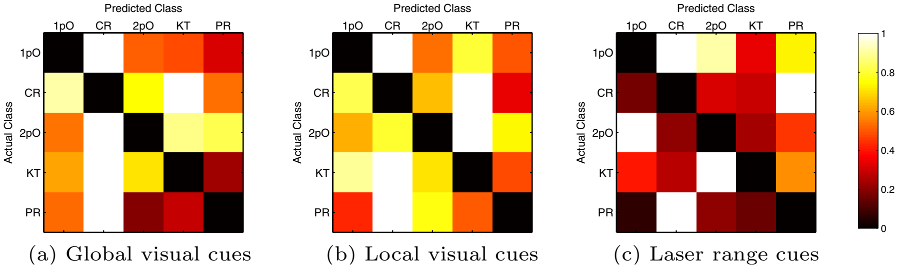
The rest of this section motivates the choice of modalities (Section 5.8.1), provides an overview of the architecture of the place classification system (Section 5.8.2) and gives details about the algorithms used to extract and classify the geometric and visual cues (Section 5.8.3 and Section 5.8.4). Then, the method used for cue integration is described in Section 5.8.5. The section concludes with a discussion on the need for adaptive models for place classification in dynamic environments (Section 5.8.6). All the algorithms were experimentally evaluated on robotic platforms operating in realistic environments. The experiments and obtained results are presented in Section Section 5.9.

5.8.1 Multiple Cues and Modalities for Place Classification

Nowadays, robots are usually equipped with several sensors, typically a laser range finder and a camera (or cameras), providing both geometrical and visual information about the environment. The ability to effectively integrate multiple cues, possibly extracted from multiple sensory modalities, becomes an important feature of a place classification system. First of all, as each sensor usually captures a different aspect of the environment, using multiple cues allows for obtaining more descriptive representation. A laser range scanner can be a valuable source of geometrical information, while vision is necessary if a robot requires a notion of human-like appearance-based concepts. Good descriptive and discriminative abilities along with robustness are the two crucial features of a place classification system with a great influence on its overall performance. The visual sensor is an irreplaceable source of distinctive information about a place. However, this information tends to be noisy and difficult to analyze due to the susceptibility to variations introduced by changing illumination and everyday activities in the environment. At the same time, laser range finders provide much more stable and robust geometric cues. These cues, however, are unable to uniquely represent the properties of different places. This leads to the problem of perceptual aliasing [76]. Clearly, each modality has its own characteristics. Interestingly, the weaknesses of one often correspond to the strengths of the other.

It is important to note that even alternative interpretations of the information obtained by the same sensor can be valuable. In this work we concentrate on two different types of visual cues based on global and local image features. Global features are derived from the whole image and thus can capture general properties of the whole scene. In contrast, local features are computed locally, from distinct parts of an image. This makes them much more robust to occlusions and viewpoint variations, but requires a costly matching process in order to find feature correspondences.

The different properties of the cues result in different performance and error patterns on the place classification task. This is illustrated in Figure 5.14 which shows distributions of errors made by three single-cue place classification algorithms for five different place classes (see Section 5.9.1 for details). It is apparent that each of the cues makes errors according to a different pattern. The cue integration scheme should exploit this fact in order to increase the overall performance. The experimental results reported in Section 5.9.2 show that the performance of a place classification system can indeed be boosted by combining the stability of geometrical solutions with the versatility of different visual cues.



(a) Global visual cues (b) Local visual cues (c) Laser range cues

Fig. 5.14. Distributions of errors made by three single-cue place classification algorithms for five different place classes (1pO - one person office, CR - corridor, 2pO - two persons office, KT - kitchen, PR - printer area). Bright colors indicate the classes most often confused with the actual class. The diagonal elements were removed.

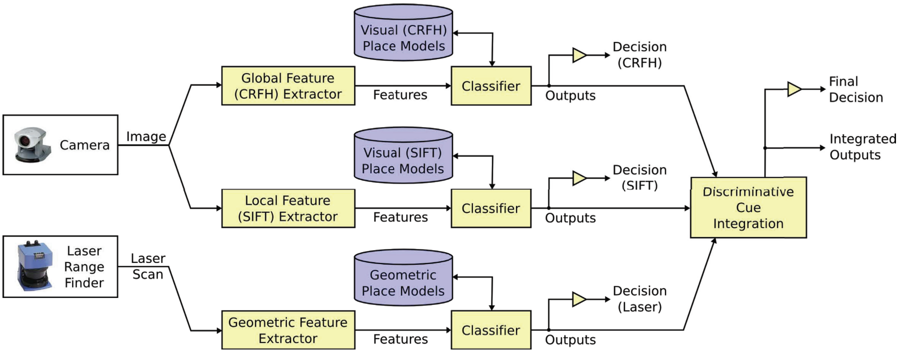


Fig. 5.15. Architecture of the multi-modal place classification system

5.8.2 Architecture of the Place Classification System

The architecture of the place classification system described in this section is illustrated in Figure 5.15. The system relies on two visual cues corresponding to two different types of image features (local based on the SIFT descriptor [39] and global based on the Composed Receptive Field Histograms [72]) as well as simple geometrical cues extracted from laser range scans [7]. The cues are processed independently. For each cue, there is a separate path in the system which consists of two main building blocks: a feature extractor and a classifier. Each classifier produces a set of outputs indicating its soft decision for all place classes. These outputs can be used directly to obtain the final decision separately for each cue. In cases when several cues are available, the single-cue outputs are combined using a high-level discriminative accumulation scheme producing integrated outputs from which the final decision is derived. As was described in Section 5.5.4, the integrated outputs can be accumulated over time and space if the system is used on a mobile platform. Since each of the

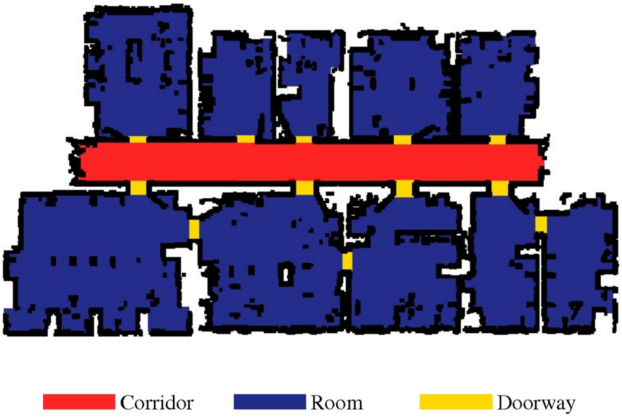


Fig. 5.16. An occupancy grid map built on the ground floor of the building 52 at the University of Freiburg. Some natural divisions can be extracted from this map e.g. corresponding to rooms, doorways and a corridor.

cues is treated independently, the system can decide to acquire and process additional information only when necessary e.g. only in difficult cases. This scheme is referred to as Confidence-based Cue Integration [10].

5.8.3 Laser-Based Place Classification

This section presents our approach to place classification based on geometric features extracted from laser range data. Many places in indoor environments can be distinguished due to their different structure. This structure can be unique for an instance of a place, but can also be characteristic for a whole semantic place category. For example, the bounding box of a corridor is usually longer than that of rooms or hallways. At the same time, rooms are typically smaller than hallways, and also more cluttered than corridors and hallways. As an example, Figure 5.16 shows a typical hand-labeled division of an environment into three categories of places.

As illustrated in Figure 5.15, the place classification algorithm first extracts a set of simple geometrical features from the scan acquired by the range sensor. Figure 5.17 shows an example of a scan taken by a mobile robot in a corridor. Each feature is represented by a numerical value computed from the beams of the scan or from a polygon representing the covered area. Single features alone are not sufficient for reliable places classification. Here, the AdaBoost algorithm is used to boost the simple features into a strong classifier. As is shown in Section 5.9.1, different classification algorithms, such as Support Vector Machines [73], can also be used to successfully derive a place class from the geometrical features. A brief description of the features and the AdaBoost algorithm is given below.

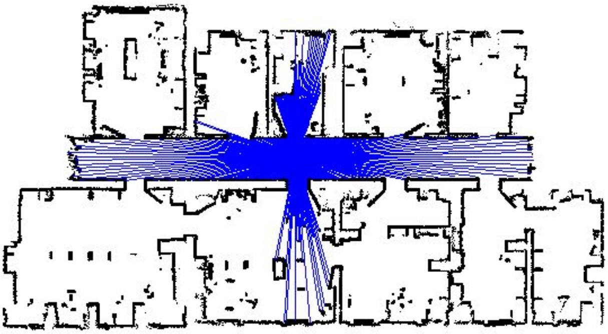


Fig. 5.17. A range scan covering the complete 360° field of view acquired by a mobile robot in a corridor

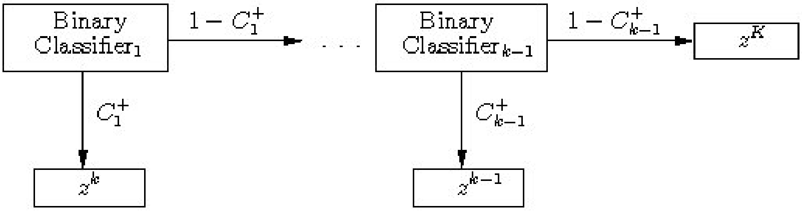


Fig. 5.18. A decision list built for K classes using binary classifiers. The output of each binary classifier is the probability z^k that the classified example belongs to the k-th class.

Classification Using AdaBoost

The AdaBoost algorithm, introduced in [74], is one of the most popular boosting algorithms. This algorithm takes as an input a training set of positive and negative examples. On each round, AdaBoost calls a weak learning algorithm repeatedly to select a weak hypothesis. The key idea is to maintain a weight distribution over the training examples. This distribution indicates the importance of the examples at the beginning of the training process and later is controlled by the algorithm. Below, a modified version of the original algorithm is described which outputs a confidence value for each positive and negative classification [77].

The original AdaBoost algorithm was designed for binary classification problems. However, to label places in the environment, we need the ability to handle multiple classes. One way to construct a multi-class classifier is to arrange several binary classifiers into a decision list. Each element of such a list represents one binary classifier which determines if an example belongs to one specific class. In addition, each binary classifier outputs a confidence value C_k^+ for a positive classification of its class k. Figure 5.18 illustrates the structure of the probabilistic decision list.

In the decision list, each test example is fed into the first binary classifier, which outputs a confidence value C_1^+ for a positive classification. Then the example is passed to the next binary classifier. This process is repeated until the last element in the list. The complete output of the decision list is represented by a histogram z. In this histogram, the k-th bin stores the probability that the classified location belongs to the k-th class according to the sequence of classifiers in the decision list. This probability can be computed as follows:

$$z^k = C_k^+ \prod_{j=1}^{k-1}(1 - C_j^+), \tag{5.2}$$

where the confidence value for the last K-th bin is equal to 1. In the multi-cue framework, these probability values are used as the outputs which are integrated by the cue integration function (see Figure 5.15).

One important question in the context of a sequential classifiers is the order in which the individual binary classifiers are arranged. A good strategy is to order the classifiers in increasing order according to their training error rate. Compared to the optimal order, the classifier generated by this heuristic performed only 1.3% worse on average for an application with several classes demonstrated in [78]. In several cases, the sequence generated by this heuristic turned out to be the optimal one.

Simple Features from Sensor Range Data

Let's assume that the mobile robot is equipped with a range sensor covering the 360° field of view. Each laser observation $z = \{b_0, ..., b_{M-1}\}$ contains a set of beams b_i. Each beam b_i consists of a tuple (α_i, d_i) where α_i is the angle of the beam relative to the robot and d_i is the length of the beam. Each training example for the AdaBoost algorithm is just one observation z and its classification y. Thus, the set of training examples is given as

$$E = \{(z_i, y_i) \mid y_i \in Y = \{\text{Room}, \text{Corridor}, \ldots\}\}. \tag{5.3}$$

In this approach, each laser observation is represented by a set of simple geometric features expressed using single real values. All features are rotationally invariant to make the classification dependent only on the (x, y)-position of the robot and not of its orientation. Most of the features are standard geometrical characteristics often used in shape analysis and pattern recognition. We define a feature f as a function that takes as argument one observation and returns a real value: $f : Z \rightarrow \Re$, where Z is the set of all possible observations. Figure 5.19 shows graphically some of these features used. The complete list of features, together with their mathematical definition, can be found in [77].

Common configurations on real mobile robots have only one laser scanner covering the 180° in front of the robot. In these cases the values corresponding to the rear laser scan can be set to zero. A more advanced solution is

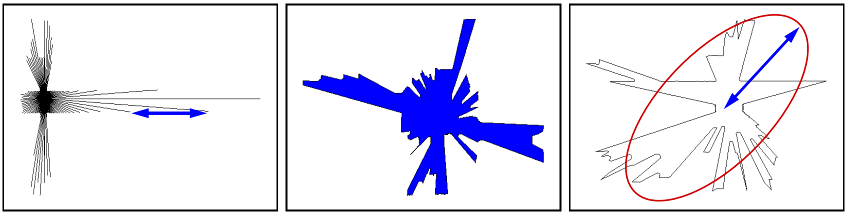


Fig. 5.19. Examples of features generated from laser range data, namely the average distance between two consecutive beams, the perimeter of the area covered by a scan, and the mayor axis of the ellipse that approximates the polygon described by the scan. Here, the laser beams cover a 360° field of view.

to maintain a local map around the robot. This local map can be updated during the movements of the robot, and then used to simulate the rear laser beams [77]. This approximation have shown good results in several indoor experiments [7, 63].

5.8.4 Vision-Based Place Classification

This section describes the visual place classification algorithms proposed in [12] that constitute the two paths of the vision-based channel in the multi-modal system presented in Figure 5.15. Each of the paths is built around a Support Vector Machine (SVM) classifier [79] and a different type of visual feature, global or local, extracted from the same image frame (see Section 5.8.1 for the distinction between the feature types). The global features are represented using a rich global descriptor, Composed Receptive Field Histograms (CRFH, [72]). The local features are based on the Scale Invariant Feature Transform (SIFT, [39]). Both have already been proved successful in the domain of vision-based localization [10, 37, 66].

The rest of the section describes the feature extraction algorithms and sketches the theory behind SVMs which will also form a basis for the cue integration scheme presented in Section 5.8.5.

Global Visual Features: Composed Receptive Field Histograms

CRFH is a multi-dimensional statistical representation of the occurrence of responses of several image descriptors applied to the image. This idea is illustrated in Figure 5.20. Each dimension corresponds to one descriptor and the cells of the histogram count the pixels sharing similar responses of all descriptors. This approach allows to capture various properties of the image as well as relations that occur between them. We tested a wide variety of combinations of image descriptors with several scale levels. On the basis of an evaluation of performance and computational cost, we build the histograms from either first order or second order Gaussian derivative filters applied to the illumination channel at two scales. This resulted in either 4- or 6-dimensional histograms.

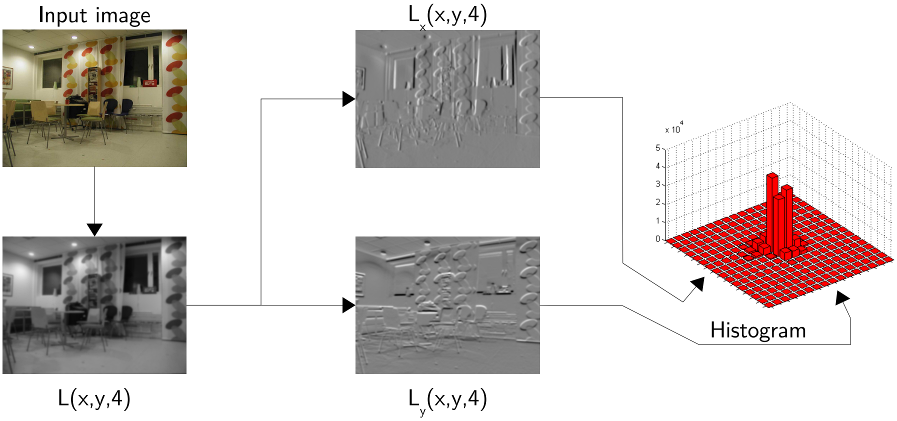


Fig. 5.20. The process of generating multi-dimensional receptive field histograms shown on the example of the first-order derivatives computed at the same scale $t = 4$ from the illumination channel

Multi-dimensional histograms can grow extremely fast if the number of dimensions grows. However, most of the cells are usually empty [72]. Storing only those that are non-zero allows for reducing the amount of required memory and performing operations such as histogram accumulation and comparison efficiently.

In case of SVMs, special care must be taken in choosing an appropriate kernel function which acts as a similarity measure between the feature vectors. In this work, the χ^2 kernel [80] was used for the CRFH descriptors. The χ^2 kernel belongs to the family of exponential kernels, and is given by

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{-\gamma\chi^2(\boldsymbol{x}, \boldsymbol{y})\right\}, \qquad \chi^2(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \frac{||x_i - y_i||^2}{||x_i + y_i||}. \tag{5.4}$$

Local Features: Scale Invariant Feature Transform

The process of local feature extraction consists of two stages: *interest point detection* and *description*. The interest point detector identifies a set of characteristic points in the image that could be re-detected even in spite of various transformations. The role of the descriptor is to extract robust features from the local patches located at the detected points. Here, we used the scale, rotation, and affine invariant interest point detector based on the difference-of-Gaussians (DoG) operator [81] and the SIFT descriptor [39]. Figure 5.21 presents local patches located at the interest points detected in three typical images acquired in an indoor environment.

In case of local features, the similarity between two images is measured by solving the correspondence problem. Thus, in order to couple the local

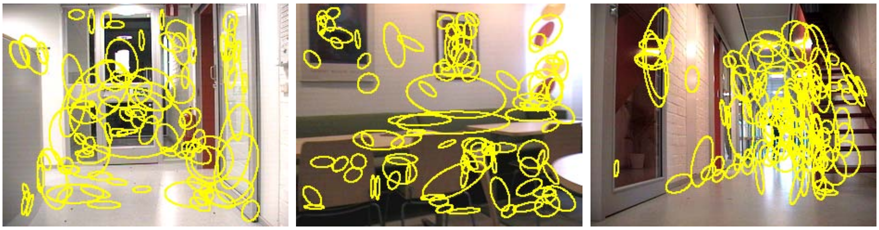


Fig. 5.21. Local patches at the interest points detected in three typical images acquired in an indoor environment. The size of the patches illustrate the scale at which the points were detected.

descriptors with the SVMs, the match kernel proposed in [82] was used. The match kernel is given by

$$K(\boldsymbol{L}_h, \boldsymbol{L}_k) = \frac{1}{n_h} \sum_{j_h=1}^{n_h} \max_{j_k=1,\ldots,n_k} \left\{ K_l(\boldsymbol{L}_h^{j_h}, \boldsymbol{L}_k^{j_k}) \right\}, \tag{5.5}$$

where $\boldsymbol{L}_h, \boldsymbol{L}_k$ are local feature sets and $\boldsymbol{L}_h^{j_h}, \boldsymbol{L}_k^{j_k}$ are two single local features. The sum is always calculated over the smaller set of local features and only some fixed amount of best matches is considered in order to exclude outliers. The local feature similarity kernel K_l can be any Mercer kernel. Here, the RBF kernel based on the Euclidean distance was used for the SIFT features:

$$K_l(\boldsymbol{L}_h^{j_h}, \boldsymbol{L}_k^{j_k}) = \exp\left\{ -\gamma ||\boldsymbol{L}_h^{j_h} - \boldsymbol{L}_k^{j_k}||^2 \right\}. \tag{5.6}$$

Support Vector Machines

Support Vector Machines are a binary discriminative classifier known for their superior generalization abilities. Consider the problem of separating the set of labeled training data $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$ into two classes, where $\boldsymbol{x}_i \in \Re^N$ is a feature vector and $y_i \in \{-1, +1\}$ its class label. Assuming that the two classes can be separated by a hyperplane in some Hilbert space $\mathcal{H}$, then the optimal separating hyperplane is the one which has maximum distance to the closest points in the training set resulting in a discriminant function

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b. \tag{5.7}$$

The classification result is then given by the sign of $f(\boldsymbol{x})$. The values of α_i and b are found by solving a constrained minimization problem, which can be done efficiently using the SMO algorithm [83]. Most of the α_i's take the value of zero; those $\boldsymbol{x}_i$ with nonzero α_i are the “support vectors”. In cases where the two classes are non-separable, the optimization is formulated in such a way that the classification error is minimized and the final solution remains identical. The mapping between the input space and the usually high dimensional feature space $\mathcal{H}$ is done using kernels $K(\boldsymbol{x}_i, \boldsymbol{x})$.

The extension of SVM to multi-class problems can be done in several ways. Three different approaches were used in this work:

1. *Standard one-against-all (OaA) strategy.* If M is the number of classes, M SVMs are trained, each separating a single class from all other classes. The decision is then based on the distance of the classified sample to each hyperplane, and the sample is assigned to the class corresponding to the hyperplane for which the distance is largest.
2. *Modified one-against-all strategy.* In [10], a modified version of the OaA principle was proposed. The authors suggested to use distances to precomputed average distances of training samples to the hyperplanes (separately for each of the classes), instead of the distances to the hyperplanes directly. In this case, the sample is assigned to the class corresponding to the hyperplane for which the distance is smallest. Experiments presented in this paper and in [10] show that in many applications this approach outperforms the standard OaA technique.
3. *One-against-one (OaO) strategy.* In this case, $M(M-1)/2$ two-class machines are trained for each pair of classes. The final decision can then be taken in different ways, based on the $M(M-1)/2$ outputs. A popular choice is to consider as output of each classifier the class label and count votes for each class; the test image is then assigned to the class that received more votes.

In each of the aforementioned cases, the classified sample is processed by a set of binary classifiers. Each of these classifiers produces a value of the discriminant function as defined by Eq. (5.7). In the multi-cue framework, these values are used as the outputs which are integrated by the cue integration function (see Figure 5.15).

Support Vector Machines do not provide any out-of-the-box solution for estimating the confidence of the decision; however, it is possible to derive confidence information and hypotheses ranking from the distances between the samples and the hyperplanes. In order to estimate the confidence of the decision provided by the single-cue place classification algorithms and both confidence estimates and the hypotheses ranking for the final decision of the multi-modal place classification system, we used the distance-based confidence estimation method proposed in [10].

5.8.5 Discriminative Cue Integration

This section describes the SVM-based Discriminative Accumulation Scheme (SVM-DAS) algorithm [12]. The algorithm is used to integrate cues from one or multiple modalities in the place classification system presented in Figure 5.15.

Various cue integration methods have been proposed in the robotics and machine learning community [71, 84, 10, 85, 86, 87]. These approaches can be described according to various criteria. For instance, [88] suggest to classify them into two main groups, *weak coupling* and *strong coupling.* Assuming that each cue is used as input of a different classifier, weak coupling is when the output of two or more independent classifiers are combined. Strong coupling is instead when the output of one classifier is affected by the output of another classifier, so that their outputs are no longer independent. Another possible classification is into *low level* and *high level* integration methods, where the emphasis is on the level at which integration happens. We call *low level integration methods* those algorithms where cues are combined together at the feature level, and then used as input to a single classifier [87, 71]. Another strategy is to keep the cues separated and to integrate the outputs of individual classifiers, each trained on a different cue [85, 84, 12]. We call such algorithms *high level integration methods*, of which voting is the most popular [89]. These techniques are more robust with respect to noisy cues or sensory channels. Moreover, they allow to divide the learning problem into several smaller sub-problems. Additionally, not all cues need always be used and the algorithm can decide on the number of cues that should be extracted for each particular classification task [10].

SVM-DAS is a technique performing weak coupling, high level, non-linear cue integration. For each cue, the method requires training a separate classifier which provides a set of outputs encoding the relation of the classified sample to the place models for the particular cue. The integration is performed by feeding the outputs to a Support Vector Machine. Compared to previous high-level discriminative accumulation methods [84, 10], SVM-DAS gives several advantages. First, it accumulates cues with a more complex, possibly non-linear function, by using the SVM framework and kernels. Such approach makes it possible to integrate outputs of different classifiers such as SVM and AdaBoost. Moreover, it learns the weights for each cue very efficiently from the training data, therefore making it possible to accumulate large numbers of cues without computational problems. At the same time, SVM-DAS preserves the important property of the previous methods to perform correct classification even when each of the single cues gives misleading information.

Suppose, there are P cues and therefore, P single-cue classifiers. Each classifies a single cue $T_p(\boldsymbol{I})$, where $p = 1 \dots P$, extracted from the sensory input $\boldsymbol{I}$. Then, each classifier produces a set of outputs $\{O_h^p(T_p(\boldsymbol{I}))\}_{h=1}^{H_p}$, where H_p defines the number of outputs for the p-th cue. The outputs are used as an input to an SVM, and the parameters of the integration function are learned during the optimization process, for instance using the SMO algorithm [83] (see Section 5.8.4 for a brief overview of the theory behind SVMs). This gives raise to the following integration function of SVM-DAS:

$$O_g^{\Sigma P}(\boldsymbol{I}) = \sum_{i=1}^{n} \alpha_i^g y_i K(\boldsymbol{O}_i, \boldsymbol{O}) + b^g, \; g = 1, \dots, G,$$

where K is the kernel function and $\boldsymbol{O}$ is a vector containing all the outputs for all cues:

$$\boldsymbol{O} = \left[\{O_h^1(T_1(\boldsymbol{I}))\}_{h=1}^{H_1}, \ldots, \{O_h^P(T_P(\boldsymbol{I}))\}_{h=1}^{H_P} \right].$$

The parameters y_i, α_i^g, b^g and the support vectors $\boldsymbol{O}_i$ are inferred from the training data either directly or efficiently during the optimization process. The number of the final outputs G and the way of obtaining the final decision depends on the multi-class extension used with SVM-DAS. We tested the one-against-all extension, for which $G = M$, and the one-against-one extension, for which $G = M(M-1)/2$, where M is the number of classes. In both cases, we observed a very similar performance.

In case of SVM-DAS, the nonlinearity is given by the choice of the kernel function, thus in the case of the linear kernel the method is linear. For the experiments reported in this section, we used the non-linear RBF (Gaussian) kernel given by

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{-\gamma ||\boldsymbol{x} - \boldsymbol{y}||^2\right\}. \tag{5.8}$$

5.8.6 Adaptive Place Classification

In most cases, the place classification systems are trained off-line or once they are trained the representation remains static. However, in the real, dynamic world, learning cannot be a single act. It is simply not possible to create a static model which could explain all the variability observed over time. Continuous information acquisition and exchange, coupled with an ongoing learning process, is necessary to provide the system with a valid world representation and preserve stable performance. In artificial autonomous agents constrained by limited resources, continuous learning must be performed in an incremental fashion. It is not feasible to rebuild the internal model from scratch every time new information arrives; neither is it possible to store all the previously acquired data for that purpose. The model must be updated and the updating process must have certain properties. First, the knowledge representation must remain compact and free from redundancy to fit into the limited memory and maintain a fixed computational complexity. Second, the model cannot grow forever even though new information is constantly arriving. The updating process should be able to gradually filter out unnecessary information.

Here, we focus on the scenario in which incremental learning is applied to place models in order to provide adaptability to different types of variations observed in real-world environments. As the experiments described in Section 5.9.2 show, the multi-modal place classification system presented in this chapter is able to cope with illumination and pose changes as well as short-term dynamic variations. Moreover, since it relies on multiple sensors, it can deliver satisfactory results despite dynamic variations that occurred during the period of around 6 months. Still, this variability is clearly affecting the performance of the system. As it is not possible to predict *a priori* how the environment is going to change, the only possible long-term strategy is to update the representation over time, learning incrementally from the new data recorded during use.

To experimentally verify the usefulness of adaptive models for place classification, we implemented and tested the memory-controlled approximate incremental extension of the SVM algorithm proposed in [90]. Approximate techniques [91, 92, 90] seem to be better suited for our problem because, at each incremental step, they discard non-informative training vectors, thus reducing the memory requirements. Other methods, such as [93, 94], instead require storing in memory all the training data. The basic principle behind the memory-controlled method is to combine the fixed-partition incremental extension [92] with an algorithm for controlling the memory growth [95]. Every time a new batch of data becomes available for the learning algorithm, the knowledge stored in previously built model in the form of support vectors is combined with the incoming data and used to train a new model. Then, a support vector reduction algorithm is applied to the model, which filters out redundant information by eliminating those vectors that can be expressed by a linear combination of the others. This permits keeping the model compact and provides the algorithm with forgetting capabilities. For more details, the reader is referred to [90, 75]. The results of the experimental evaluation of the method on place classification data are presented in Section 5.9.3.

5.9 Experiments with Place Classification

This section describes several series of experiments we conducted to evaluate the performance of the place classification algorithms presented in Section 5.8 on both uni-modal and multi-modal data. First, we performed experiments with single-cue place models to verify their properties and test their robustness to different types of variations e.g. introduced by illumination changes and long-term human activity (Section 5.9.1). Then, the evaluation was repeated for systems based on different combinations of cues and modalities to see if the robustness can be improved by cue integration (Section 5.9.2). In the next experiment, we took a different approach and tried to tackle long term variability by using adaptive place models (Section 5.9.3). Finally, we combined multi-modal place classification with a localization and mapping component implementing the first three layers of the spatial model and run an experiment where the task was to build a representation of a novel indoor environment (Section 5.9.4).

5.9.1 Single-Cue Place Classification

This section reports results of two experiments performed using single-cue place classification systems. The aim of the first experiment was to test the

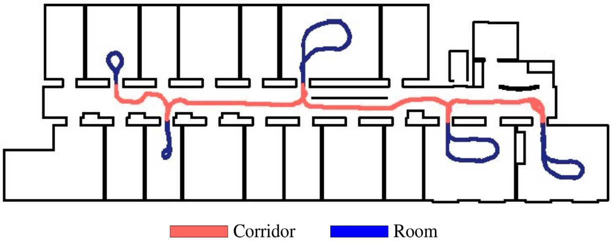


Fig. 5.22. Trajectory followed by the robot during acquisition of the training data for the room vs. corridor classification experiment. Labels were attached to the data based on the position of the robot and are marked on the plot using different colors.

ability of a system relying purely on laser range data to perform classification of places in a typical office environment into two classes: a corridor and a room. The second experiment aimed at evaluating robustness of various cues on the place classification task despite substantial variability that occurred in a realistic indoor environment over the period of several months.

Semantic Place Classification Using Laser Range Data

As explained in Section 5.5.4, providing even basic semantic descriptions, such as a room or a corridor, for regions of space can enhance functionality of a mobile cognitive agent operating in an indoor environment and interacting with a user. In such a scenario, the robot is often facing the user which affects the information captured using the laser range sensor. In order to provide reliable classification during these experiments, we used the approach based on the simple geometric features and the AdaBoost classifier presented in Section 5.8.3. We simulated the rear-view laser scanner by ray-tracing in the local obstacle map. Then, the simulated and the real scans were used together as a 360° laser range finder.

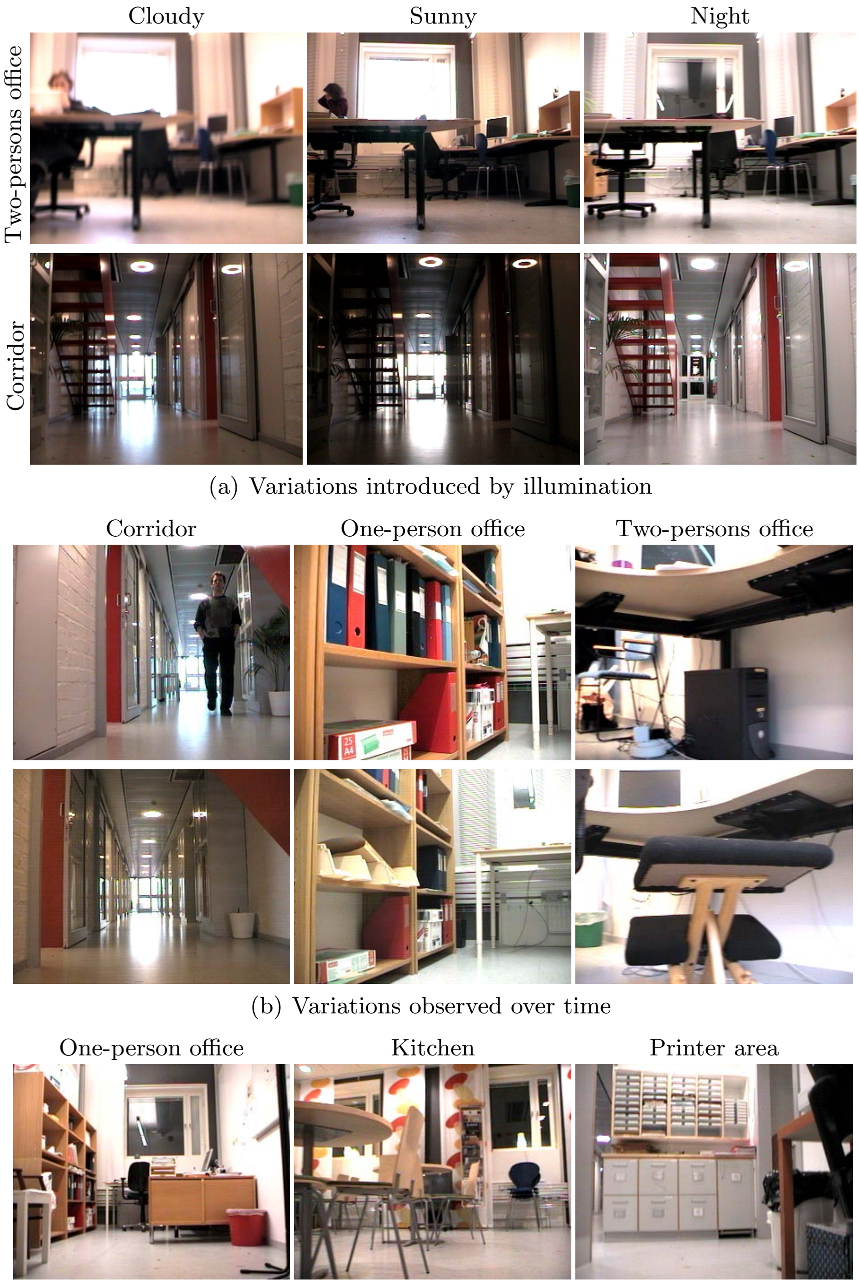
In order to test the method, we used data acquired along trajectories of the robot being driven through rooms and corridors found on two different floors of the CAS/CVAP laboratory at the Royal Institute of Technology in Stockholm, Sweden. To train the classifier, we used the scans acquired on the 6th floor along the trajectory shown in Figure 5.22. The robot was then moved to the 7th floor of the same building, which contains a similar structure. On this floor, we classified two different trajectories established in opposite directions. The classification rates for all the poses of the robot during its movement ranged from 93.18% to 96.8%. A more extensive set of experiments using these approach is shown in [77].

Single-Cue Place Classification Under Large Variability

In this experiment, we tested the robustness of four different single-cue place classification algorithms to different types of variations, such as those introduced by changing illumination or human activity over a long period of time [12]. We evaluated performance of two SVM models trained on global visual features (CRFH, Section 5.8.4) and local visual features (SIFT, Section 5.8.4) as well as SVM and AdaBoost models trained on the laser range cues (here referred to as L-AB and L-SVM, Section 5.8.3). The design of these experiments was partially based on findings from our previous work on visual place classification [65]. A video presenting the setup, experimental procedure and visualization of the results for the original experiments described in [65] can be found in [96].

The evaluation was performed on the IDOL2 database [97, 75]. The database comprises 24 labeled sequences of images at the resolution of 320x240 pixels synchronized with laser scans and odometry data acquired using two mobile robot platforms (PeopleBot and PowerBot) over a time span of 6 months. The acquisition was performed in a five room subsection of a larger office environment, selected in such way that each of the five rooms represented a different functional area: a one-person office (1pO), a two-persons office (2pO), a kitchen (KT), a corridor (CR), and a printer area (PR). Example pictures showing interiors of the rooms are presented in Figure 5.23. The appearance of the rooms was captured under three different illumination conditions: in cloudy weather, in sunny weather, and at night. The robots were manually driven through each of the five rooms while continuously acquiring images and laser scans at a rate of 5fps. The acquisition process was conducted in two phases. Two sequences were acquired using each robot for each type of illumination conditions over the time span of more than two weeks, and another two sequences for each setting were recorded 6 months later. Thus, the sequences captured variability introduced not only by illumination but also natural activities in the environment (presence/absence of people, furniture/objects relocated etc.). It is important to note that, even for sequences acquired within a short time span under similar illumination conditions, variations still exist from everyday activities and viewpoint differences during acquisition. The captured variability is illustrated in Figure 5.23. More detailed information about the database can be found in [98].

We conducted two sets of experiments for each cue on 12 data sequences from the IDOL2 database acquired with the PowerBot (additional experiments can be found in [12]). For each single experiment, we trained the models on one sequence and tested on another. The first set consisted of 12 experiments, performed on different combinations of training and test data acquired closely in time and under similar illumination conditions. Then, we increased the complexity of the problem and performed experiments on 24 pairs of training and test sets, obtained 6 months from each other and under different illumination settings. As a measure of performance we used the percentage of properly classified samples (classification rate) calculated separately for each of the rooms and then averaged with equal weights independently of the number of samples acquired in each room.

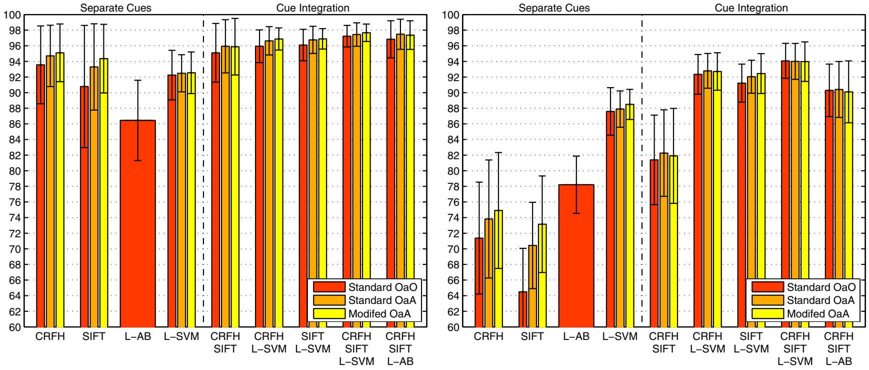


(a) Variations introduced by illumination

(b) Variations observed over time

(c) Remaining rooms (at night)

Fig. 5.23. Examples of pictures taken from the IDOL2 database showing the interiors of the rooms, variations observed over time and caused by activity in the environment as well as introduced by changing illumination



(a) Stable illumination, close in time (b) Varying illumination, distant in time

Fig. 5.24. Results of the experiments evaluating performance of four single-cue place classification systems and systems based on several combinations of multiple cues

In each experiment, we evaluated the performance of all four types of models: CRFH, SIFT, L-AB, and L-SVM. For SVM, we tried the three multi-class extensions described in Section 5.8.4. The results are presented in Figure 5.24a,b (the first four bar groups). First, the results for the three different multi-class extensions are in agreement with [10] - for single cues, the modified one-against-all algorithm gives the best performance independently of the modality on which the classifier was trained. Second, we see that under stable conditions, the vision-based methods outperform the systems based on laser range cues (95.1% for CRFH and 92.5% for L-SVM). It is also apparent that the variations that occurred over the long period of time pose a challenge for both modalities. In this case, vision also suffers from the large variations in illumination which do not affect the geometric cues. Furthermore, we can see that there is a significant difference in performance between the two laser-based solutions in favor of the SVM-based method.

A detailed analysis of the distribution of errors made by all the SVM-based models can be found in Figure 5.14 and Section 5.8.1. The fact that there are large discrepancies between the error patterns indicates that effective cue integration might result in increased performance.

5.9.2 Combining Multiple Cues and Modalities

The experiments described in this section were designed to evaluate performance of the SVM-DAS cue integration scheme and multi-cue place classification system presented in Section 5.8 and [12]. Since SVM-DAS performs high level cue integration, separate models must be trained for each of the combined cues. In this case, we used the models obtained during the single-cue experiments presented in the previous section. Moreover, we used the same

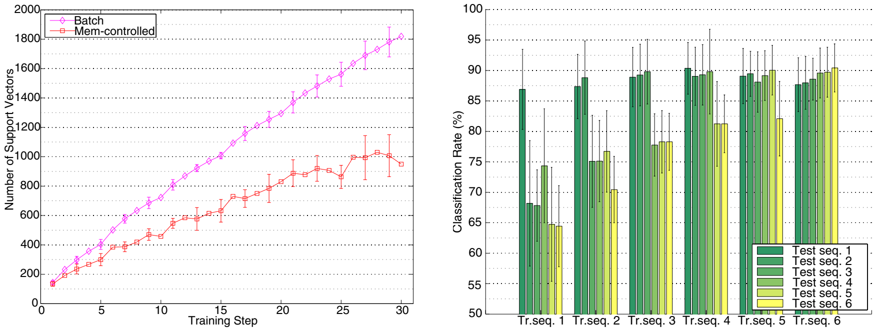
Table 5.1. Average percentages (with standard deviations) of test samples for which all cues had to be used in order to retain the maximal recognition rate

Cues (**Primary cue**)	Percentage of test samples
CRFH + SIFT	29.5±22.1
CRFH + **L-SVM**	32.7±20.3
SIFT + **L-SVM**	33.3±22.4
SIFT + CRFH + **L-SVM**	40.8±21.9

experimental setup, so that the results can be easily compared. A detailed description of all the experiments performed can be found in [12].

We tested the integration method with several combinations of different cues and modalities. The results are reported in Figure 5.24a,b (the last 5 bar groups). First, we combined the two visual cues. We see that the robustness of a purely visual recognition system can be greatly improved by integrating different types of cues, in this case local and global. This can be observed especially for the experiment where the algorithms had to tackle the largest variability. Despite that, the error distributions in Figure 5.14 indicate that we should expect largest gain when different modalities are combined. As we can see from Figure 5.24 this is indeed the case. By combining one visual cue and one laser range cue (e.g. CRFH + L-SVM), we exploit the descriptive power of vision in case of stable illumination conditions and the invariance of geometrical features to the visual noise. Moreover, if the computational cost is not an issue, the performance can be further improved by using both visual cues instead of just one. To test the ability of SVM-DAS to integrate outputs of different classifiers, we combined the SVM models trained on visual cues with AdaBoost model based on geometrical features (L-AB). The method obtained a large improvement in comparison to each of the individual cues. For instance, the recognition rate increased by 12.2% on average in the most difficult case.

Although it is clear that the performance can be significantly improved by using multiple cues, each of the cues introduces additional computational cost. This cost can be significantly reduced by taking the approach presented in [10] which combines confidence estimation methods with high level cue integration. Since, in most cases, decisions based on only one cue are correct, the system could decide to use additional sources of information only when necessary i.e. when the decision based on a single cue is not confident enough. Table 5.1 presents the results of applying the method to the experiments presented in this section. We see that, in general, the decision can be based on the fastest cue (marked with bold font in Table 5.1) and the maximal performance can be retained despite using additional cues only in approximately 35% of cases. Additional cues will be used more often when the variability is large, and rarely for less difficult cases.



(a) Support vectors

(b) Performance of memory-controlled

Fig. 5.25. Average results of the experiments with adaptive place classification: the number of support vectors stored in the model after each step and the classification rates obtained by testing the models after every fifth step with all the available test sets. The training and test sets marked with the same indices were acquired under similar conditions.

5.9.3 Adaptive Place Classification

This section gives an overview of the experimental evaluation of an adaptive place classification model presented in [75]. The experiments were based on the IDOL2 database described in Section 5.9.1 and focused on the ability of the algorithm to adapt to long-term variations. We used the memory-controlled incremental SVM algorithm for training the place models and the visual global features (CRFH) to represent the sensory data. Preliminary experiments showed that the behavior of the algorithm was very similar for the local features.

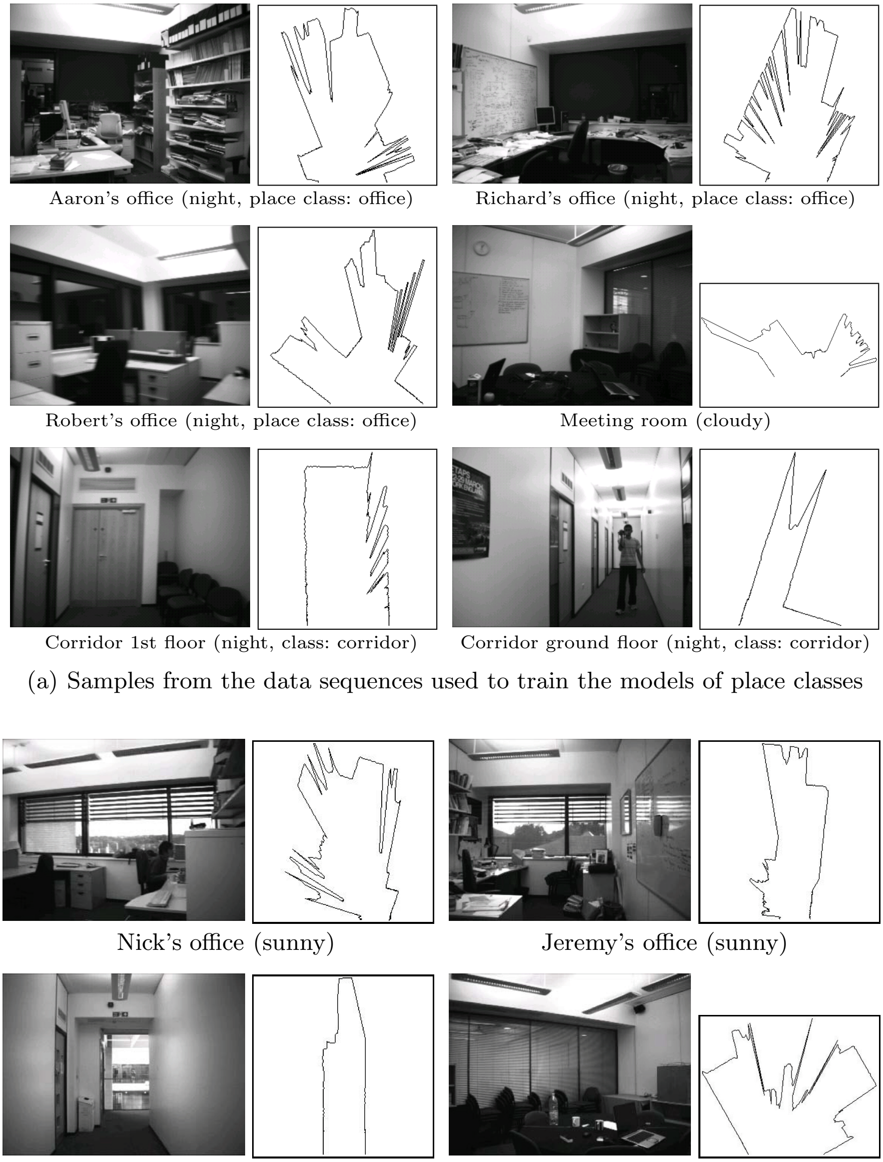
We considered a case where the algorithm needed to incrementally gain robustness to variations introduced by changing illumination and human activities, while at the same time using its adaptation ability to handle long-time changes in the environment. We first trained the system on three image sequences from the database acquired at roughly the same time but under different illumination conditions. Then, we repeated the same training procedure on sequences acquired 6 months later. In order to increase the number of incremental steps and differentiate the amount of new information introduced by each set of data, each sequence was again divided into five subsequences. Thus, in total, there were 30 incremental steps. Since the IDOL2 database consists of pairs of sequences acquired under roughly similar conditions, each training sequence has a corresponding one which could be used for testing. As a measure of performance we used the percentage of properly classified samples (classification rate) averaged over all the rooms.

The experiment was repeated 12 times for different orderings of training sequences and we compared the results of the incremental method to the batch SVM algorithm. Figure 5.25a shows the average amounts of support vectors stored in the models at each incremental step for both methods. Figure 5.25b reports the classification rate measured every fifth step (every time the system completes learning a whole sequence) for the incremental technique. In order to emphasize the need for adaptation as well as to visualize how the learning process affects the performance on the past test data, the figure shows recognition rates for all testing sets used throughout the experiment. By observing the rates for a classifier trained on the first sequence only, we see that the system achieves best performance on a test set acquired under similar conditions. The classification rate is significantly lower for other test sets especially for images acquired 6 months later, even under similar illumination conditions. At the same time, the performance greatly improves when incremental learning is performed on new batches of data. The classification rate decreases for the old test sets; at the same time, the size of the model tends to stabilize and the incremental model is much more compact than the one produced by the batch method. The results presented provide clear evidence of the capability of the discriminative methods to perform incremental learning for vision-based place classification, and their adaptability to variations in the environment.

5.9.4 Semantic Labeling of Space

We performed a real-time experiment to test the multi-modal place classification system together with other components implementing the multi-layered spatial model on a mobile robot platform. The experiment was performed during working hours in a typical office environment. Following the findings of the off-line experiments described in Section 5.9.2, we built the multi-modal place classification system based on visual and laser range cues integrated using SVM-DAS. For efficiency reasons, we used only global features (CRFH) for the vision channel. The system was implemented in the CAST (The CoSy Architecture Schema Toolkit, see Chapter 2) framework and run on a standard 2.5GHz dual-core laptop. The whole experiment was videotaped and a video presenting the setup, experimental procedure and visualization of the results can be found in [99].

The experiment was performed in the building of the School of Computer Science at the University of Birmingham, United Kingdom. The interior of the building consists of several office environments located on three floors. For our experiments, we selected three semantic categories of rooms that could be found in the building: a corridor, an office and a meeting room. In order to train the system, and build place models for these three classes, we first performed acquisition of training data in different parts of the building. To build the model of an office, we acquired data in three different offices: Aaron's office (1^{st} floor), Robert's office (1^{st} floor) and Richard's office (ground floor). To create a representation of the corridor class, we recorded data in 2 corridors, one on the ground floor and one on the 1^{st} floor. The acquisition was performed at night. Finally, to train the model of a meeting room, we used an instance on the 2^{nd} floor. The meeting room belonged to the part of the environment



Aaron's office (night, place class: office)

Richard's office (night, place class: office)

Robert's office (night, place class: office)

Meeting room (cloudy)

Corridor 1st floor (night, class: corridor)

Corridor ground floor (night, class: corridor)

(a) Samples from the data sequences used to train the models of place classes

Nick's office (sunny)

Jeremy's office (sunny)

Corridor 2nd floor (sunny)

Meeting room (sunny)

(b) Samples acquired during the test run

Fig. 5.26. Examples of images and laser scans (synchronized) taken from the data sequences used for training the models of place classes (a) and acquired during the test run (b) in each of the rooms considered during the semantic labeling experiment. The figure illustrates the within-category variations for corridors and offices as well as other types of variability observed for each place class.

where later we conducted the final test. The robot was manually driven around each room and data samples were recorded at the rate of 5 fps. In case of the meeting room, the $1st$ floor corridor as well as Aaron's and Richard's offices, the acquisition was repeated twice. Examples of images and laser scans acquired in each of the rooms can be found in Figure 5.26a.

We trained the place models separately for each modality on a dataset created from one data sequence recorded in each of the rooms. Since one of the advantages of SVM-DAS is the ability to infer the integration function from the training data, after training the models, we trained the integration scheme. We used the additional data sequences acquired in some of the rooms and trained SVM-DAS on the outputs of the uni-modal models tested on these data.

Three days after the training data were collected, we performed a real-time experiment in the lab on the 2^{nd} floor in the same building. The experiment was conducted during the day during sunny weather. The part of the environment that was explored by the robot consisted of 2 offices (Nick's office and Jeremy's office), a corridor and a meeting room. The interiors of the rooms and the influence of illumination can be seen in the images in Figure 5.26b. An automatically generated map of the environment is presented in Figure 5.5.

During the experiment, the task of the robot was to build a multi-layered spatial representation of the environment and semantically label the navigation graph nodes and areas. The only knowledge given to the robot before the experiment consisted of the models of the three place classes: "office", "corridor" and "meeting room". The robot started in Nick's office, and was manually driven through the corridor to Jeremy's office. Then, it was taken to the meeting room where the autonomous exploration mode was turned on. The robot used a frontier-based algorithm based on [100]. After the meeting room was explored, the robot was manually driven back to the Nick's office where the experiment finished. The semantic labeling process was running on-line and the place classification was performed approximately at the rate of 5 times per second. The final semantic map build during the run is shown in Figure 5.5. We can see that the system correctly labeled all the areas in the environment.

The fact that the data were stored allowed for detailed performance analysis of the place classification system, similar to the one presented in Section 5.9.2. The results are displayed in Figure 5.27. When we look at the overall classification rate for all the data samples in the test sequence, we see that vision significantly outperformed laser in this experiment (66% vs. 84%). Still, the performance of the system was boosted by additional 8% compared to vision alone when the two modalities were integrated. The gain is even more apparent if we look at the detailed results for each of the classes (the first three charts in Figure 5.27). We see that the modalities achieved different performance, but also different error patterns, for each class. Clearly, the system based on laser range data is a very good corridor detector. On the other hand, vision was able to distinguish between the offices and the meeting room almost perfectly.

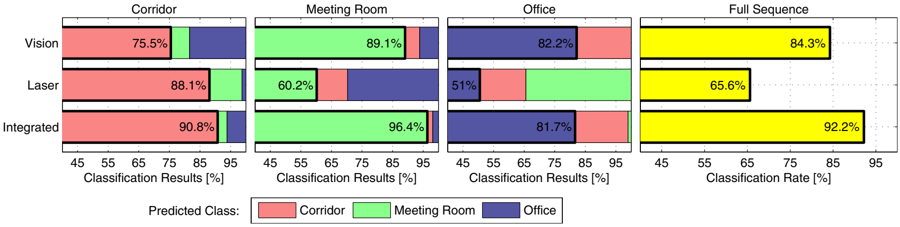


Fig. 5.27. Place classification results obtained on the dataset recorded during the test run. The first three bar charts show the results separately for each place class: "corridor", "meeting room" and "office". The charts show the percentage of the samples that were properly classified (most left bars marked with thick lines), but also how the misclassifications were distributed. The chart on the right presents the percentage of properly classified samples during the whole run.

Finally, the integrated system always achieved the performance of the more reliable modality and for two out of three classes outperformed the uni-modal systems.

5.10 Summary

We set out to create a spatial representation that would help to bridge the gap between how humans and robots represent space to facilitate interaction and support spatial reasoning. In part supported by findings in cognitive psychology and also inspired by such work as by Kuipers [4], we proposed a layered spatial model.

At the lowest level, our representation consists of a metric map that supports navigation and localization. This chapter presented a number of different approaches to how this metric map can be represented and implemented. In the integrated system, the so called M-Space feature representation [14] was used with laser range data. Much of the CoSy research on metric mapping concentrated on investigating methods for a vision-only strategy. This is also where most of the contributions to science in the area of metric mapping are found [48, 41, 14, 44]. However, since the metric map is the foundation of the spatial model and is fundamental for proper functioning of the entire system, reliability had to take priority. The framework used has however been tested in vision-only setups as described in [41, 14].

The navigation map was designed to provide a way of representing the free space in the model. As it provides coarse discretization of space it limits the state space of the path planning tasks and is also extremely useful for storing semantic information. While not a new idea, the navigation graph has been shown in this research to be a powerful representation that supports tasks beyond pure path planning for which it was originally designed [23]. One of the avenues for future work is to investigate how information about the appearance of a place (e.g. detected objects, visual features extracted from a scene or metric place descriptors) can be used to not only introduce semantics into the model, but also support localization. A model that captures the graphical nature of the navigation layer and contains place descriptors and coarse metric information seems like a good candidate for such a joint representation.

The navigation and topological maps allow to segment space into topological regions and associate semantic place information with those regions. A purely geometric method was investigated for categorizing places into rooms and corridors [7]. The experiment showed that the method is able to generate models valid even across different environments. In parallel, research was conducted on vision-based place classification. Extensive experiments demonstrated that places can be recognized and categorized reliably even using a perspective camera with limited field of view and in presence of different types of visual variations [10]. These two novel strands of work were integrated into a joint, multi-modal framework in [12]. This framework was used for semantic place categorization within the integrated system.

The conceptual map corresponds to the highest level of abstraction in the model and provides the link between the spatial model and the communication system used for situated human-robot dialogue. It grounds linguistic expressions in representations of spatial entities, such as instances of rooms or objects. The conceptual also allowed us to derive new knowledge from partial knowledge and a common sense ontology.

Each of the layers in the spatial model plays an important role in the system, providing a basis for different pieces of its functionality. Each layer also advances the state of the art in its corresponding area. As a whole, the model constitutes a versatile, but also coherent spatial representation. Compared to the work by Kuipers [4] our work uses a supervised paradigm and is focused on human-robot interaction. In fact, as will be explained in more detail in Chapter 8, the way we acquire the spatial model is in itself an example of human-robot interaction. This allows knowledge to be exchanged between robot and human during the mapping process which paves the way for a shared representation of space.

As clearly demonstrated in Section 5.9, the integration of information from many different cues and sensory modalities helps to improve the performance and comprehension of space. In a similar way, the layered spatial model provides the means for integrating information across different levels of abstractions. Chapter 10 will explain how the rest of the system interacts with the spatial model in the context of the Explorer scenario system.

References

1. Vasudevan, S., Gachter, S., Berger, M., Siegwart, R.: Cognitive maps for mobile robots an object based approach. In: Proceedings of the IROS 2006 Workshop: From Sensors to Human Spatial Concepts, Beijing, China (2006)
2. Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernández-Madrigal, J., González, J.: Multi-hierarchical semantic maps for mobile robotics. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Alberta, Canada (2005)
3. Beeson, P., MacMahon, M., Modayil, J., Murarka, A., Kuipers, B., Stankiewicz, B.: Integrating multiple representations of spatial knowledge for mapping, navigation, and communication. In: Interaction Challenges for Intelligent Assistants, AAAI Spring Symposium, Stanford, CA, USA (2007)
4. Kuipers, B.: The Spatial Semantic Hierarchy. Artificial Intelligence 119, 191–233 (2000)
5. Krieg-Brückner, B., Röfer, T., Carmesin, H.-O., Müller, R.: A taxonomy of spatial knowledge for navigation and its application to the Bremen autonomous wheelchair. In: Freksa, C., Habel, C., Wender, K.F. (eds.) Spatial Cognition 1998. LNCS (LNAI), vol. 1404, pp. 373–397. Springer, Heidelberg (1998)
6. Diosi, A., Taylor, G., Kleeman, L.: Interactive SLAM using laser and advanced sonar. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2005), Barcelona, Spain (2005)
7. Mozos, O.M., Triebel, R., Jensfelt, P., Rottmann, A., Burgard, W.: Supervised semantic labeling of places using information extracted from laser and vision sensor data. Robotics and Autonomous Systems Journal 55(5), 391–402 (2007), http://cognitivesystems.org/CoSyBook/chap5.asp#MartinezMozos07a
8. Friedman, S., Pasula, H., Fox, D.: Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India (2007)
9. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: Proceedings of the International Conference on Computer Vision, ICCV 2003 (2003)
10. Pronobis, A., Caputo, B.: Confidence-based cue integration for visual place recognition. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA (2007), http://cognitivesystems.org/CoSyBook/chap5.asp#pronobis07iros
11. Rottmann, A., Mozos, O.M., Stachniss, C., Burgard, W.: Place classification of indoor environments with mobile robots using boosting. In: Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA, USA, pp. 1306–1311 (2005), http://cognitivesystems.org/CoSyBook/chap5.asp#rottmann05aaai
12. Pronobis, A., Martínez Mozos, O., Caputo, B.: SVM-based discriminative accumulation scheme for place recognition. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008), Pasadena, CA, USA (2008), http://cognitivesystems.org/CoSyBook/chap5.asp#pronobis08icra

13. López, D.G., Sjö, K., Paul, C., Jensfelt, P.: Hybrid laser and vision based object search and localization. In: Proceedings of the International Conference on Robotics and Automation, ICRA 2008 (2008), http://cognitivesystems.org/CoSyBook/chap5.asp#Galvez08a
14. Folkesson, J., Jensfelt, P., Christensen, H.: The m-space feature representation for slam. IEEE Transactions on Robotics 23(5), 1024–1035 (2007), http://cognitivesystems.org/CoSyBook/chap5.asp#Folkesson07a
15. Stevens, A., Coupe, P.: Distortions in judged spatial relations. Cognitive Psychology 10, 422–437 (1978)
16. McNamara, T.: Mental representations of spatial relations. Cognitive Psychology 18, 87–121 (1986)
17. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. Fundamenta Informaticae 46, 1–29 (2001)
18. Hirtle, S.C., Jonides, J.: Evidence for hierarchies in cognitive maps. Memory and Cognition 13, 208–217 (1985)
19. Brown, R.: How shall a thing be called? Psychological Review 65(1), 14–21 (1958)
20. Rosch, E.: Principles of categorization. In: Rosch, E., Lloyd, B. (eds.) Cognition and Categorization, pp. 27–48. Lawrence Erlbaum Associates, Hillsdale (1978)
21. Moravec, H.P.: Sensor fusion in certainty grids for mobile robots. AI Magazine 9, 61–74 (1988)
22. Latombe, J.C.: Robot Motion Planning. Academic Publishers, Boston (1991)
23. Newman, P., Leonard, J., Tardós, J., Neira, J.: Explore and return: Experimental validation of real-time concurrent mapping and localization. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2002), Washington, D.C., USA, pp. 1802–1809 (2002)
24. Moravec, H., Elfes, A.: High resolution maps from wide angle sonar. In: Proceedings of the International Conference on Robotics and Automation (ICRA 1985), pp. 116–121 (1985)
25. Schiele, B., Crowley, J.L.: A comparison of position estimation techniques using occupancy grids. In: Proceedings of the International Conference on Robotics and Automation (ICRA 1994), vol. 2, pp. 1628–1634 (1994)
26. Burgard, W., Fox, D., Henning, D., Schmidt, T.: Estimating the absolute position of a mobile robot using position probability grids. In: Proceedings of the National Conference on Artificial Intelligence (AAAI 1996), Portland, Oregon, USA, pp. 896–901 (1996)
27. Duckett, T., Nehmzow, U.: Mobile robot self-localisation using occupancy histograms and a mixture of gaussian location hypotheses. Robotics and Autonomous Systems 34(2–3), 119–130 (2001)
28. Hähnel, D., Schulz, D., Burgard, W.: Map building with mobile robots in populated environments. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2002), vol. 1, pp. 496–501 (2002)
29. Hähnel, D., Burgard, W., Fox, D., Thrun, S.: An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2003), vol. 1, pp. 206–211 (2003)
30. Leonard, J.J., Durran-Whyte, H.F., Cox, I.J.: Dynamic map building for an autonomous mobile robot. The International Journal of Robotics Research 11(4), 286–298 (1992)

31. Arras, K., Vestli, S.: Hybrid, high-precision localisation for the mail distributing mobile robot system mops. In: Proceedings of the International Conference on Robotics and Automation (ICRA 1998), pp. 3129–3134 (1998)
32. Dissanayake, M.G., Newman, P., Clark, S., Durrant-Whyte, H., Corba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation 17(3), 229–241 (2001)
33. Frese, U., Schrder, L.: Closing a million-landmarks loop. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China (2006)
34. Arras, K., Tomatis, N., Siegwart, R.: Multisensor on-the-fly localization using laser and vision. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2000), Takamatsu, Japan, pp. 462–476 (2000)
35. Tardós, J.: Representing partial and uncertain sensorial information using the theory of symmetries. In: Proceedings of the International Conference on Robotics and Automation (ICRA 1992), vol. 2, pp. 1799–1804 (1992)
36. Castellanos, J.A., Tardós, J.D.: Mobile Robot Localization and Map Building: A Multisensor Fusion Approach. Kluwer Academic Publishers, Dordrecht (1999)
37. Se, S., Lowe, D.G., Little, J.: Vision-based mobile robot localization and mapping using scale-invariant features. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2001), Seoul, Korea (2001)
38. Elinas, P., Sim, R., Little, J.J.: σSLAM: Slam: Stereo vision slam using the rao-blackwellised particle filter and a novel mixture proposal distribution. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2006), Orlando, FL (2006)
39. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
40. Goncalves, L., di Bernardo, E., Benson, D., Svedman, M., Ostrovski, J., Karlsson, N., Pirjanian, P.: A visual front-end for simultaneous localization and mapping. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2005), Barcelona, Spain, pp. 44–49 (2005)
41. Jensfelt, P., Kragic, D., Folkesson, J., Björkman, M.: A framework for vision based bearing only 3D SLAM. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2006), Orlando, FL (2006)
42. Frintrop, S., Jensfelt, P., Christensen, H.I.: Attentional landmark selection for visual slam. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China (2006)
43. Frintrop, S., Jensfelt, P.: Active gaze control for attentional visual SLAM. In: Proceedings of the International Conference on Robotics and Automation, ICRA 2008 (2008), http://cognitivesystems.org/CoSyBook/chap5.asp#Frintrop08a
44. Frintrop, S., Jensfelt, P.: Attentional landmarks and active gaze control for visual SLAM. IEEE Transactions on Robotics, special Issue on Visual SLAM 24 (2008), http://cognitivesystems.org/CoSyBook/chap5.asp#Frintrop08b
45. Frintrop, S.: Vocus: A visual attention system for object detection and goal-directed search. Ph.D. thesis, University of Bonn (July 2005)
46. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2d range scans. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR 1994), pp. 935–938 (1994)

47. Gutmann, J.-S., Konolige, K.: Incremental mapping of large cyclic environments. In: Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation (CIRA 1999), pp. 318–325 (1999)
48. Bertolli, F., Jensfelt, P., Christensen, H.I.: Slam using visual scan-matching with distinguishable 3D points. In: Proceedings of the International Conference on Intelligent Robots and Systems, IROS 2006 (2006), `http://cognitivesystems.org/CoSyBook/chap5.asp#Bertolli06a`
49. Jensfelt, P.: Approaches to mobile robot localization in indoor environments. Ph.D. thesis, Signal, Sensors and Systems (S3), Royal Institute of Technology, SE-100 44 Stockholm, Sweden (2001)
50. Tapus, A., Ramel, G., Dobler, L., Siegwart, R.: Topology learning and recognition using bayesian programming for mobile robot navigation. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2004), Sendai, Japan, pp. 3139–3144 (2004)
51. Anguelov, D., Koller, D., Parker, E., Thrun, S.: Detecting and modeling doors with mobile robots. In: Proceedings of the International Conference on Robotics and Automation, ICRA 2004 (2004)
52. Zender, H.: Learning spatial organization through situated dialogue, Master's thesis, Dept. of Computational Linguistics, Saarland University, Saarbruecken, Germany (2006)
53. Topp, E.A., Christensen, H.I.: Tracking for following and passing persons. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Alberta, Canada (2005)
54. Topp, E.A., Hüttenrauch, H., Christensen, H., Severinson Eklundh, K.: Bringing together human and robotic environment representations – a pilot study. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China (2006)
55. Kruijff, G.-J.M., Zender, H., Jensfelt, P., Christensen, H.I.: Situated dialogue and spatial organization: What, where... and why? International Journal of Advanced Robotic Systems, Special Issue on Human-Robot Interaction 4(1), 125–138 (2007), `http://cognitivesystems.org/CoSyBook/chap5.asp#kruijff07jars`
56. Lee, D.T., Lin, A.K.: Computational complexity of art gallery problems. IEEE Transactions on Information Theory 32(2), 276–282 (1986)
57. López, D.G.: Combining object recognition and metric mapping for spatial modeling with mobile robots. Master's thesis, Royal Institute of Technology (July 2007)
58. Ekvall, S., Kragic, D.: Receptive field cooccurrence histograms for object detection. In: Proceedings of the International Conference on Robotics and Automation, IROS 2005 (2005)
59. Sjö, K., Paul, C.: Object localization using bearing only visual detection. In: Proceedings of the 10th International Conference on Intelligent Autonomous Systems (2008), `http://cognitivesystems.org/CoSyBook/chap5.asp#Sjoe08b`
60. Ekvall, S., Kragic, D., Jensfelt, P.: Object detection and mapping for service robot tasks. Robotica: International Journal of Information, Education and Research in Robotics and Artificial Intelligence 25(2), 175–187 (2007)
61. Brunskill, E., Kollar, T., Roy, N.: Topological mapping using spectral clustering and classification. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2007), San Diego (2007)

62. Siagian, C., Itti, L.: Biologically-inspired robotics vision monte-carlo localization in the outdoor environment. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA (2007)
63. Zender, H., Mozos, Ó.M., Jensfelt, P., Kruijff, G.-J.M., Burgard, W.: Conceptual spatial representations for indoor mobile robots. Robotics and Autonomous Systems 56(6), 493–502 (2008), http://cognitivesystems.org/CoSyBook/chap5.asp#zender08ras_fs2hsc
64. Buschka, P., Saffiotti, A.: A virtual sensor for room detection. In: Proceedings of the International Conference on Intelligent Robots and Systems, IROS 2002 (2002)
65. Pronobis, A., Caputo, B., Jensfelt, P., Christensen, H.I.: A discriminative approach to robust visual place recognition. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China (2006), http://cognitivesystems.org/CoSyBook/chap5.asp#pronobis06iros
66. Filliat, D.: A visual bag of words method for interactive qualitative localization and mapping. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2007), Roma, Italy (2007)
67. Ulrich, I., Nourbakhsh, I.: Appearance-based place recognition for topological localization. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2000), San Francisco, CA, USA (2000)
68. Blaer, P., Allen, P.: Topological mobile robot localization using fast vision techniques. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2002), Washington, DC, USA (2002)
69. Murillo, A.C., Guerrero, J.J., Sagues, C.: Surf features for efficient robot localization with omnidirectional images. In: Proceedings of the the International Conference on Robotics and Automation (ICRA 2007), Roma, Italy (2007)
70. Valgren, C., Lilienthal, A.J.: Incremental spectral clustering and seasons: Appearance-based localization in outdoor environments. In: Proceedings of the International Conference on Robotics and Automation (ICRA 2008), Pasadena, CA, USA (2008)
71. Tapus, A., Siegwart, R.: Incremental robot mapping with fingerprints of places. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Alberta, Canada (2005)
72. Linde, O., Lindeberg, T.: Object recognition using composed receptive field histograms of higher dimensionality. In: Proceedings of the International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK (2004)
73. Vapnik, V.: Statistical Learning Theory. Wiley and Son, New York (1998)
74. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
75. Luo, J., Pronobis, A., Caputo, B., Jensfelt, P.: Incremental learning for place recognition in dynamic environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA (2007), http://cognitivesystems.org/CoSyBook/chap5.asp#luo07iros
76. Kuipers, B., Beeson, P.: Bootstrap learning for place recognition. In: Proceedings of the 18th National Conference on Artificial Intelligence, AAAI 2002 (2002)

77. Mozos, O.M.: Semantic place labeling with mobile robots, Ph.D. thesis, University of Freiburg, Freiburg, Germany (July 2008)
78. Rottmann, A.: Bild- und laserbasierte klassifikation von umgebungen mit mobilen robotern, Master's thesis, University of Freiburg, Department of Computer Science (2005) (in German)
79. Cristianini, N., Taylor, J.S.: An Introduction to SVMs and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
80. Chapelle, O., Haffner, P., Vapnik, V.: SVMs for histogram-based image classification. Transactions on Neural Networks 10(5)
81. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. International Journal of Computer Vision 66(3)
82. Wallraven, C., Caputo, B., Graf, A.: Recognition with local features: the kernel recipe. In: Proceedings of the International Conference on Computer Vision, ICCV 2003 (2003)
83. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods: Support Vector Learning, pp. 185–208. MIT Press, Cambridge (1999)
84. Nilsback, M.E., Caputo, B.: Cue integration through discriminative accumulation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004 (2004)
85. Poggio, T., Torre, V., Koch, C.: Computational vision and regularization theory. Nature 317
86. Triesch, J., Eckes, C.: Object recognition with multiple feature types. In: Proceedings of the International Conference on Artificial Neural Networks, ICANN 1998 (1998)
87. Matas, J., Marik, R., Kittler, J.: On representation and matching of multicoloured objects. In: Proceedings of the International Conference on Computer Vision, ICCV 1995 (1995)
88. Clark, J., Yuille, A.: Data fusion for sensory information processing systems. Kluwer Academic Publishers, Dordrecht (1990)
89. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley, Chichester (2001)
90. Pronobis, A., Caputo, B.: The more you learn, the less you store: Memory-controlled incremental SVM, IDIAP-RR 51, IDIAP (2006), `http://cognitivesystems.org/CoSyBook/chap5.asp#pronobis06idiap`
91. Domeniconi, C., Gunopulos, D.: Incremental support vector machine construction. In: Proceedings of the International Conference on Data Mining, ICDM 2001 (2001)
92. Syed, N.A., Liu, H., Sung, K.K.: Incremental learning with support vector machines. In: Proceedings of the International Joint Conferences on Artificial Intelligence, IJCAI 1999 (1999)
93. Cauwenberghs, G., Poggio, T.: Incremental and decremental support vector machine learning. In: Advances in Neural Information Processing Systems, NIPS 2000 (2000)
94. Orabona, F., Castellini, C., Caputo, B., Luo, J., Sandini, G.: Indoor place recognition using online independent support vector machines. In: 18th British Machine Vision Conference (BMVC 2007), Warwick, UK (2007)
95. Downs, T., Gates, K.E., Masters, A.: Exact simplification of support vector solutions. Journal of Machine Learning Research 2

96. Pronobis, A., Caputo, B., Jensfelt, P., Christensen, H.I.: A discriminative approach to robust visual place recognition (2006), http://cognitivesystems.org/cosybook/videos.asp#robVisPR
97. The KTH-IDOL2 database, http://cognitivesystems.org/cosybook/datasets.asp#idol
98. Luo, J., Pronobis, A., Caputo, B., Jensfelt, P.: The KTH-IDOL2 database. Tech. Rep. CVAP304, Kungliga Tekniska Hoegskolan, CVAP/CAS (October 2006), http://cognitivesystems.org/CoSyBook/chap5.asp#luo06kth
99. Pronobis, A.: Multi-modal semantic labeling of space (2008), http://cognitivesystems.org/cosybook/videos.asp#MMsemLab
100. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, pp. 146–151 (1997)