Supervised semantic labeling of places using information extracted from sensor data [1]

Óscar Martínez Mozos [a,*] Rudolph Triebel [a] Patric Jensfelt [b]
Axel Rottmann [a] Wolfram Burgard [a]

[a] *Department of Computer Science, University of Freiburg,*
D-79110, Freiburg, Germany

[b] *Royal Institute of Technology, Center for Autonomous Systems,*
SE-100 44, Stockholm, Sweden

Abstract

Indoor environments can typically be divided into places with different functionalities like corridors, rooms or doorways. The ability to learn such semantic categories from sensor data enables a mobile robot to extend the representation of the environment facilitating the interaction with humans. As an example, natural language terms like “corridor” or “room” can be used to communicate the position of the robot in a map in a more intuitive way. In this work, we first propose an approach based on supervised learning to classify the pose of a mobile robot into semantic classes. Our method uses AdaBoost to boost simple features extracted from sensor range data into a strong classifier. We present two main applications of this approach. Firstly, we show how our approach can be utilized by a moving robot for an online classification of the poses traversed along its path using a hidden Markov model. In this case we additionally use as features objects extracted from images. Secondly, we introduce an approach to learn topological maps from geometric maps by applying our semantic classification procedure in combination with a probabilistic relaxation method. Alternatively, we apply associative Markov networks to classify geometric maps and compare the results with the relaxation approach. Experimental results obtained in simulation and with real robots demonstrate the effectiveness of our approach in various indoor environments.

Key words: Semantic place classification, Topological maps, Place categorization, Human-robot interaction

[*] Corresponding author. Fax: +49 (761) 203-8007
Email address: omartine@informatik.uni-freiburg.de (Óscar Martínez Mozos).

[1] This work was partially supported by the European Union under FP6-004250-

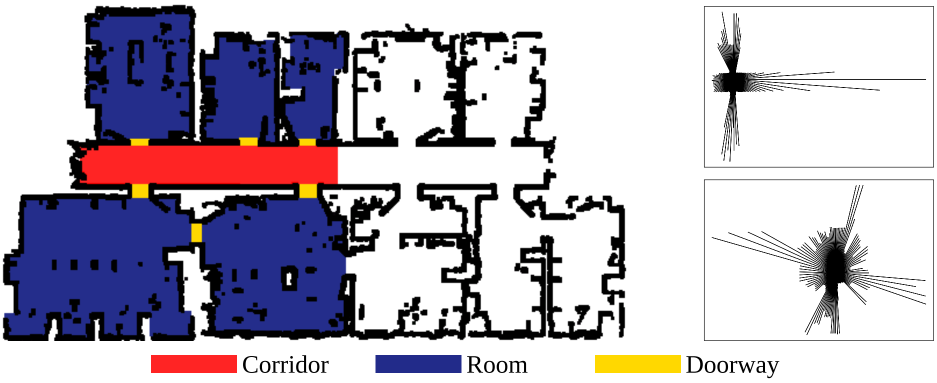

Fig. 1. The left image shows a map of the ground floor of building 52 at the University of Freiburg with part of it classified by hand. The semantic labels are depicted as color/grey scale. The right image depicts some example scans recorded in a corridor (top) and a room (bottom).

1 Introduction

In the past, many researchers have considered the problem of building accurate maps of the environment from the data gathered with a mobile robot. The question of how to augment such maps by semantic information, however, is virtually unexplored. Whenever robots are designed to interact with their users, semantic information about places can improve the human-robot communication. From the point of view of humans, terms like “corridor” or “room” give a more intuitive idea of the position of the robot than using, for example, the 2D coordinates in a map.

Indoor environments, like the one depicted in the left image of Fig. 1, can typically be divided into areas with different functionalities such as corridor, rooms or doorways. These different places usually contain specific geometrical structures and can therefore be distinguished using sensors able to capture this structure, such as laser range finders. As an example, Fig. 1 (left) shows a typical hand-labeled division of the environment into three categories of places with its corresponding labels.

In this work, we address the problem of classifying places of the environment of a mobile robot using mainly range finder data. The key idea is to classify the position of the robot in the environment based on the current laser observation, either obtained by the robot or simulated given the map. Examples for typical observations obtained in an office environment are shown in the right images of Fig. 1. Each observation is classified by applying a sequence of binary classifiers learned with the AdaBoost algorithm [1]. These classifiers are built in a supervised fashion from simple geometric features that are ex-

IP project CoSy and by the German Research Foundation under contract number SFB/TR8.

tracted from the different laser scans. The selection of the AdaBoost algorithm is motivated by the fact that we extract a large set of geometrical features for each laser scan, and we want to select the best ones for the final classification.

We furthermore present two main applications of this approach. Firstly, we show how to classify the different poses of the robot during a trajectory improving the final classification using a hidden Markov model. In this case we additionally use as features objects extracted from images. Secondly, we introduce an approach to learn topological maps from geometric maps by applying our semantic classification to simulated range data in combination with a probabilistic relaxation procedure. Alternatively, we apply associative Markov networks to classify geometric maps and compare the results with the relaxation approach.

The rest of this work is organized as follows. After discussing related work in Section 2, Section 3 presents the sequential AdaBoost classifier. In Section 4 the simple features are described. Section 5 presents the application of a hidden Markov model to the online place classification with a moving robot. Section 6 contains our approach for topological map building. In Section 7 we show experimental results obtained using our methods. Finally, in Section 8 we present some conclusions.

2 Related work

In the past, several authors considered the problem of adding semantic information to places. Buschka and Saffiotti [2] describe a virtual sensor to identify rooms from range data. Koenig and Simmons [3] apply a pre-programmed routine to detect doorways. Finally, Althaus and Christensen [4] use sonar data to detect corridors and doorways. Learning algorithms have additionally been used to identify objects and places in the environment. For example, Anguelov *et al.* [5,6] apply the EM algorithm to cluster different types of objects from sequences of range data and to learn the state of doors. Limketkai *et al.* [7] use relational Markov networks to detect objects like doorways based on laser range data. Finally, Torralba and colleagues [8] apply hidden Markov models for learning places from image data. Compared to these approaches, our algorithm is able to combine arbitrary features extracted from range data to form a sequence of binary strong classifiers to label places. Our approach is also supervised, which has the advantage that the resulting labels correspond to user-defined classes.

On the other hand, different algorithms for creating topological maps have been proposed. Kuipers and Byun [9] extract distinctive points in the map defined as local maxima of a distinctiveness measure. Kortenkamp and Weymouth [10] fuse vision and ultrasound information to determine topologically relevant places. Shatkey and Kaelbling [11] apply a HMM learning approach to learn topological maps. Thrun [12] find critical points in the Voronoi diagram, which minimize the clearance locally. Choset [13] encodes metric and topological information in a generalized Voronoi graph to solve the SLAM problem. Additionally, Beeson *et al.* [14] detect topological places with an extension of the Voronoi graph. Zivkovic *et al.* [15] create a higher level conceptual map with visual landmarks and geometric constraints. Finally, Tapus and Siegwart [16] use fingerprints to create topological maps. In contrast to these previous approaches, the technique described in this paper applies a supervised learning method to identify complete regions in the map like corridors, rooms or doorways that have a direct relation with a human understanding of the environment. This knowledge about semantic labels of places is then used to build topological maps with a mobile robot.

3 The AdaBoost algorithm

Our classification approach for semantic labeling of places relies on a large set of single-valued geometrical features that are calculated for each laser scan. If we represent each laser scan with a vector containing all these features, then the resulting classification problem has a high dimension. Furthermore, it is difficult to know a priory which features are more discriminative or even if some of them are discriminative at all. One approach to solve this problem is to use algorithms able to select the best features for the final classification task. In our work, we use the AdaBoost algorithm to learn a strong classifier from a large set of simple features, each represented by a single value. This algorithm can be seen as a heuristic method for selecting the most discriminative features for the final classifier. In this section we explain the key ideas of AdaBoost, and how it is combined with single-valued features.

Boosting is a general method for creating an accurate strong classifier by combining a set of weak classifiers. The requirement to each weak classifier is that its accuracy is better than a random guessing. In this work we will use the Boosting algorithm AdaBoost originally introduced by Freund and Schapire [17]. The input to the algorithm is a set of labeled training examples $(e_n, l_n), n = 1, \ldots, N$, where each e_n is an example and $l_n \in \{+1, -1\}$ is a label indicating whether e_n is positive or negative respectively. In a series of rounds $t = 1, \ldots, T$, the algorithm selects repeatedly a weak classifier $h_t(e)$ using a distribution D_t over the training examples. The selected weak classifier is expected to have a small classification error on the weighted training examples. The idea of the algorithm is to modify the distribution D_t at each round, increasing the weights of the examples which were incorrectly classified by the previous weak classifier. The final strong classifier H is a weighted majority vote of the best T weak classifiers. Large weights are assigned to good weak classifiers whereas poor ones receive small weights. Whereas the weak classifiers only need to be slightly better than a random guessing, the combined strong classifier H typically produces good results.

Throughout this work we will use the approach presented by Viola-Jones [18] in which the weak classifiers depend on single-valued features f_j and have the form

$$h_j(e) = \begin{cases} +1 \text{ if } p_j f_j(e) < p_j \theta_j \\ -1 \text{ otherwise} \end{cases}, \tag{1}$$

where θ_j is a threshold and p_j is either $+1$ or -1 and thus represents the direction of the inequality. On each round t of the algorithm, the values for θ_j and p_j are chosen so that the misclassification on the weighted training examples is minimized (step 3b in Fig. 2). In our case this is achieved by searching for θ_j and p_j such as

$$(\theta_j, p_j) = \operatorname*{argmin}_{(\theta_i, p_i)} \sum_{n=1}^{N} D_t(n) I(h_i(e_n) = l_n), \tag{2}$$

where I is the indicator function, which is 1 if the argument is true, and 0 otherwise, and $D_t(n)$ is the weight assigned to example e_n in the iteration t. The final Adaboost algorithm is shown in Fig. 2 in the generalized form given by Schapire and Singer [1] and modified for the concrete task of this work.

The so far described method is able to distinguish between two classes of examples, namely positives and negatives. In practical applications, however, we want to distinguish between more than two classes. To create a multi-class classifier we follow the approach proposed by Martínez Mozos [19] and create a sequential multi-class classifier using $K-1$ binary classifiers, where K is the number of classes we want to recognize. Each element in the sequence determines if an example pertains to one specific class. If the classification is positive, then the example is assigned the corresponding class. Otherwise, the example is passed to the next element in the sequence. In our current system, we typically consider a small number of classes which makes it feasible to evaluate all potential sequences and choose the best order of binary classifiers. However, we found out that the heuristic which sorts the classifiers in decreasing order according to their classification rate also yields good results and at the same time can be computed efficiently. Compared to the optimal order, the classifier generated by this heuristic for an application with six different classes performed on average only 1.3% worse as demonstrated by Rottmann [20].

```
(1) Input: Set of examples $(e_1, l_1), \ldots, (e_N, l_N)$, where $l_n = +1$ for positive and $l_n = -1$ for negative examples.
(2) Initialize weights $D_1(n) = \frac{1}{2a}$ for $l_n = +1$ and $D_1(n) = \frac{1}{2b}$ for $l_n = -1$, where $a$ and $b$ are the number of positive and negative examples respectively.
(3) For $t = 1, \ldots, T$:
    (a) Normalize the weights $D_t(n) = \frac{D_t(n)}{\sum_{i=1}^{N} D_t(i)}$.
    (b) For each feature $f_j$ train a weak classifier $h_j$ using the distribution $D_t$.
    (c) For each classifier $h_j$ calculate $r_j = \sum_{n=1}^{N} D_t(n) l_n h_j(e_n)$,
        where $h_j(e_n) \in \{+1, -1\}$.
    (d) Choose the classifier $h_j$ that maximizes $|r_j|$ and set $(h_t, r_t) = (h_j, r_j)$.
    (e) Update the weights $D_{t+1}(n) = D_t(n) \exp(-\alpha_t l_n h_t(e_n))$,
        where $\alpha_t = \frac{1}{2} \log(\frac{1+r_t}{1-r_t})$.
(4) The final strong classifier is given by $H(e) = \mathrm{sign}(F(e))$,
    where $F(e) = \sum_{t=1}^{T} \alpha_t h_t(e)$.
```

Fig. 2. The generalized AdaBoost algorithm.

4 Simple features from sensor range data

In this section we describe the features used to create the weak classifiers in the AdaBoost algorithm. We assume that the robot is equipped with a 360° field of view range sensor. Each laser observation $e = \{b_0, ..., b_{M-1}\}$ contains a set of M beams. Each beam b_i consists of a tuple (ϕ_i, ρ_i) where ϕ_i is the angle of the beam relative to the robot and ρ_i is the length of the beam. Each training example for the AdaBoost algorithm consists of one observation e and its classification l. Thus, the set of training examples is given as $\mathcal{X} = \{(e_n, l_n) \mid l_n \in Y\}$, where $Y = \{\text{Room}, \text{Corridor}, \text{Doorway}\}$ is the set of classes. Throughout this paper we assume that the classification of the training examples is given in advance. In practice this can be achieved by manually labeling places in the map or by instructing the robot while it is exploring its environment.

Our method for place classification is based on simple geometrical features extracted from the range scans. We call them *simple* because they are single-valued features. We then define a feature f as a function that takes as argument one laser observation e and returns a real value: $f : E \rightarrow \mathbb{R}$, where E is the set of all possible observations. All our features are standard geometrical features often used in shape analysis [21–25]. Some features are extracted directly from the raw beams in e and others are calculated from a polygonal approximation of the area covered by e. All features are rotationally invariant to make the classification of a pose dependent only on the (x, y)-position of

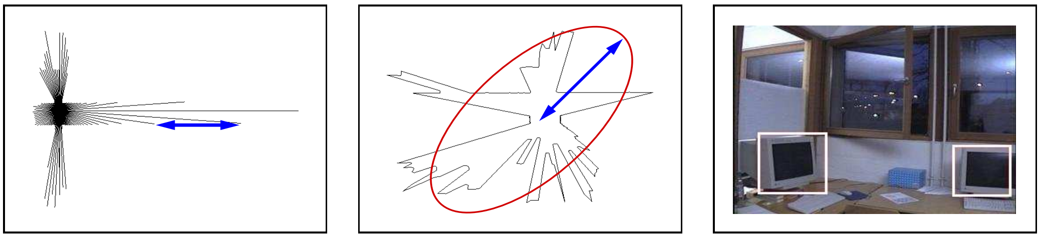

Fig. 3. The two leftmost images show examples for features generated from laser data, namely the average distance between two consecutive beams, and the mayor axis of the ellipse that approximates the polygon described by the scan. The right image shows a vision observation with some objects detected in it.

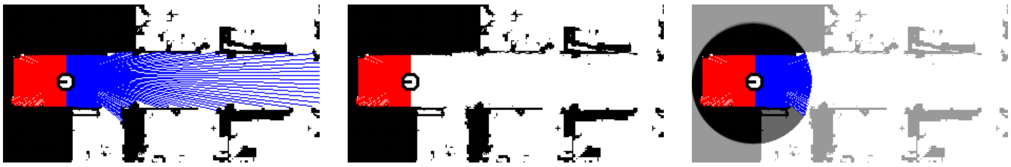

Fig. 4. The left image shows a laser observation covering 360° field of view. In the middle image the range observation covers only 180° in front of the robot. In the right image, the rear beams are simulated using the occupancy information contained inside the shaded area (local map).

the robot and not of its orientation. Typical examples of laser features considered by our system are illustrated in Fig. 3. A detailed list of laser features is contained in [19,26]. All in all we have 321 geometrical features which are fed into the learning algorithm. This large number of features is one of the main reasons to use AdaBoost. As explained in the previous section, this boosting method will allow us to select the most discriminative features for the final classifier.

As already mentioned, our simple features are based on laser observations covering 360° field of view (left image in Fig. 4). However, common configurations on real mobile robots have only a laser covering 180° in front of the robot (middle image in Fig. 4). In these last cases we propose to maintain a local map around the robot when classifying a trajectory. This local map can be updated during the movements of the robot and then used to simulate the rear laser beams (right image in Fig. 4). The classification of trajectories will be explained in Section 5.

4.1 Adding features from vision sensors

As explained before, the features used for the AdaBoost algorithm are single-valued representations of range sensor observations. This set can be extended considering observations from other sensors, like for example cameras. Here, we explain how we calculate a new set of features based on objects extracted from vision observations. The new features will be used in addition to the geometrical ones when classifying a trajectory as will be shown in Section 5.

Each of our vision observations $v = \{i_1, \ldots, i_8\}$ consists of 8 images which form a panoramic view. A vision feature is then defined as a function that takes as argument one vision observation v and returns the sum of located objects in all the images: $f_{o \in O} : V \rightarrow \mathbb{N}$, where V is the set of all possible vision observations and O is the set of all objects that we try to locate, in our case formed by the objects: “monitor on”, “monitor off”, “coffee machine”, “office cupboard”, “frontal face”, “face profile”, “full human body”, and “upper human body”. These new features permit us to distinguish between places that have similar geometrical structure and can only be identified according to the objects found there, like for example, coffee machines in kitchens. The right image of Fig. 3 depicts one example image taken in a laboratory with some objects detected in it. The different objects were detected in the images using Haar-like features based classifiers [27]. To combine the geometrical and the object-based set of features in the AdaBoost algorithm we extend each training example with the vision observations such as $\mathcal{X} = \{(e_n, v_n, l_n) \mid l_n \in Y\}$, where the set of labels Y was extended to six different places, namely rooms, corridor , doorways, kitchen, offices, seminar room and laboratory. The vision features are used in the learning process together with the geometrical ones applying the same AdaBoost algorithm of Section 3 (Rottmann *et. al.* [28]).

5 Probabilistic place recognition

The approach described so far is able to classify single observations only and does not take into account past classifications when determining the class of the current observation. In the particular domain, however, observations obtained at nearby places are typically identical. Furthermore, certain transitions between classes are rather unlikely. For example, if the classification of the current pose is “kitchen” then it is rather unlikely that the classification of the next pose is “office” given the robot moved a short distance only. To get from the kitchen to the office, the robot first has to move through a doorway.

To utilize these dependencies between the individual classes, we use a hidden Markov model (HMM) and maintain a posterior $Bel(l_t)$ about the type of the room $l_t \in Y$ the robot is currently in:

$$Bel(l_t) = \alpha P(z_t|l_t) \sum_{l_{t-1}} P(l_t|l_{t-1}, u_{t-1}) Bel(l_{t-1}). \tag{3}$$

In this equation, α is a normalizing constant ensuring that the left-hand side sums up to one over all l_t. To implement such an HMM, three components

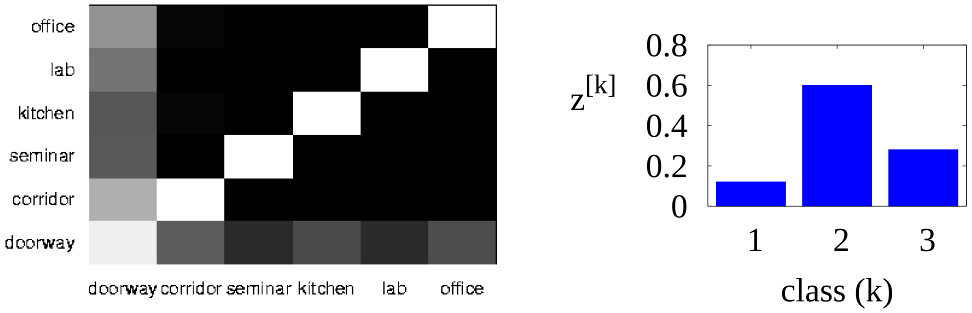

Fig. 5. The left image depicts probabilities of possible transitions between places in the environment. To increase the visibility, we used a logarithmic scale. Dark values indicate low probability. The right image illustrates an example classification output z use for topological map building.

need to be known. First, we need to specify the observation model $P(z_t|l_t)$ which is the likelihood that the classification output is $z_t \in Y$ given the actual class is l_t. Second, we need to specify the transition model $P(l_t|l_{t-1}, u_{t-1})$ which specifies the probability that the robot moves from class l_{t-1} to class l_t by executing action u_{t-1}. Finally, we need to specify, how the belief $Bel(l_0)$ is initialized.

In our current system we choose a uniform distribution to initialize $Bel(l_0)$. To determine the quantity $P(z_t|l_t)$ we generated a statistics about the classification output of the AdaBoost algorithm given that the robot was at a place corresponding to l_t. To realize the transition model $P(l_t|l_{t-1}, u_{t-1})$ we only consider the two actions $u_{t-1} \in \{MOVE, STAY\}$. The transition probabilities were estimated by running 1000 simulation experiments, in which we started the robot at a randomly chosen point and orientation in the environment and commanded it to move 20-50cm forward. This value corresponds to the distance typically traveled by the robot between two consecutive updates of the HMM. The finally obtained transition probability matrix $P(l_t|l_{t-1}, u_{t-1})$ for the action $MOVE$ is depicted in the left image of Fig. 5. As can be seen, the probability of staying in a place with the same classification is higher than the probability of changing the place. Moreover, the probability of moving from one room to a doorway is higher than the probability of moving from a room to a corridor. This indicates that the robot must first cross a doorway in order to reach a different room. Furthermore, the matrix shows a lower probability of staying in a doorway than moving into a room. This is due to the fact that a doorway is usually a small area in which the robot never rests for a longer period of time.

6 Topological map building

A second application of our semantic place classification is the extraction of topological maps from geometric maps. Throughout this section we assume that the robot is given a map of the environment in the form of an occupancy grid [29]. This geometric map can be obtained using SLAM techniques [30]. The approach for topological map building presented here is an off-line method and it uses simulated laser scan observations for the semantic classification. Vision observations are not taken into account due to the difficulty on simulating objects in images having only a 2D geometric map of the environment.

6.1 Probabilistic relaxation labeling

After obtaining an occupancy grid of the environment, our approach determines for each unoccupied cell of such a grid its semantic class. This is achieved by simulating a range scan of the robot given it is located in that particular cell, and then labeling this scan into one of the semantic classes using the sequential classifier presented in Section 3. This results in an occupancy map with a semantic label in each free cell. As we will see in the experimental Section 7, the final maps usually contain some errors in the classification. To smooth the final classification of each cell we apply a probabilistic relaxation labeling method introduced by Rosenfeld *et al.* [31]. This method changes (or maintains) the label of a cell according to the labels of its neighborhood.

The probabilistic relaxation labeling problem is defined as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph consisting of nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Let furthermore $\mathcal{L} = \{l_1, \ldots, l_L\}$ be a set of labels. We assume that every node v_i stores a probability distribution about its label which is represented by a histogram P_i. Each bin $p_i(l)$ of that histogram stores the probability that the node v_i has the label l. Thus, $\sum_{l=1}^{L} p_i(l) = 1$. For each node v_i, $\mathcal{N}(v_i) \subset \mathcal{V}$ denotes its neighborhood which consists of the nodes $v_j \neq v_i$ that are connected to v_i. Each neighborhood relation is represented by two values. Whereas the first one describes the compatibility between the labels of two nodes, the second one represents the influence between the two nodes. The terms $\mathcal{R} = \{r_{ij}(l, l') \mid v_j \in \mathcal{N}(v_i)\}$ defines the compatibility coefficients between the label l of node v_i and the label l' of v_j. And $\mathcal{C} = \{c_{ij} \mid v_j \in \mathcal{N}(v_i)\}$ is the set of weights indicating the influence of node v_j on node v_i.

Given an initial estimation for the probability distribution over labels $p_i^{(0)}(l)$ for the node v_i, the probabilistic relaxation method iteratively computes estimates $p_i^{(r)}(l),\ r = 1, 2, \ldots$, based on the initial probabilities $p_i^{(0)}(l)$, the compatibility coefficients $\mathcal{R}$, and the weights $\mathcal{C}$ in the form

$$p_i^{(r+1)}(l) = \frac{p_i^{(r)}(l)\left[1 + q_i^{(r)}(l)\right]}{\sum_{l'=1}^{L} p_i^{(r)}(l')\left[1 + q_i^{(r)}(l')\right]}, \tag{4}$$

where

$$q_i^{(r)}(l) = \sum_{j=1}^{M} c_{ij}\left[\sum_{l'=1}^{L} r_{ij}(l,l')p_j^{(r)}(l')\right]. \tag{5}$$

Note that the compatibility coefficients $r_{ij}(l,l') \in [-1,1]$ do not need to be symmetric. A value $r_{ij}(l,l')$ close to -1 indicates that label l' is unlikely at node v_j when label l occurs at node v_i, whereas values close to 1 indicate the opposite. A value of exactly -1 indicates that the relation is not possible and a value of exactly 1 means that the relation always occurs.

Probabilistic relaxation provides a framework for smoothing but does not specify how the compatibility coefficients are computed. In this work, we apply the coefficients as defined by Yamamoto [32]

$$r_{ij}(l,l') = \begin{cases} \frac{1}{1-p_i(l)}\left(1 - \frac{p_i(l)}{p_{ij}(l|l')}\right) & \text{if } p_i(l) < p_{ij}(l \mid l') \\ \frac{p_{ij}(l|l')}{p_i(l)} - 1 & \text{otherwise} \end{cases}, \tag{6}$$

where $p_{ij}(l \mid l')$ is the conditional probability that node v_i has label l given that node $v_j \in \mathcal{N}(v_i)$ has label l'. Each of the values $p_i(l)$ and $p_{ij}(l \mid l')$ are pre-calculated only once and remain the same during the iterations of the relaxation process. Thus, the coefficients $\mathcal{R}$ remain the same as well.

So far we described the general method for relaxation labeling. It remains to describe how we apply this method for spatial smoothing of the classifications obtained by our AdaBoost classifier. To learn a topological map, we assume a given two-dimensional occupancy grid map in which each cell $m_{(x,y)}$ stores the probability that it is occupied. We furthermore consider the eight-connected graph induced by such a grid. Let $v_i = v_{(x,y)}$ be a node corresponding to a cell $m_{(x,y)}$ from the map. Then we define a neighborhood $N_8(v_{(x,y)})$ using the 8-connected cells to $v_{(x,y)}$ as described in [21].

For the initial probabilities $p_{(x,y)}^{(0)}(l)$, we use an alternative representation of the classification output of the decision list described in Section 3. In this case the classification output is represented by an 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 our decision list. To compute the individual values for each bin of that histogram, we use the approach by

Friedman *et al.* [33] which determines a confidence value $C^+ \in [0,1]$ for a positive output of the binary classifier

$$C^+ = P(l = +1 \mid e) = \frac{e^{F(e)}}{e^{-F(e)} + e^{F(e)}}, \tag{7}$$

where $F(e)$ is the output of the AdaBoost algorithm according to Fig. 2. Let C_k^+ refer to the confidence value of the k-th binary classifier in our decision list. The probability that the location belongs to the k-th class is given by the k-th bin of the histogram z computed as

$$z^{[k]} = C_k^+ \prod_{j=1}^{k-1} (1 - C_j^+). \tag{8}$$

Note that the confidence value C_K^+ which is used to compute the last bin $z^{[K]}$ of the histogram is $C_K^+ = 1$ so that $\sum_{k=1}^{K} z^{[k]} = 1$. The right image of Fig. 5 illustrates an example of the histogram z corresponding to a typical classification output.

Continuing with the relaxation method, our set $\mathcal{L}$ is composed by the labels corridor, room, doorway, and wall. For each node $v_{(x,y)}$ in the free space of the occupancy grid map, we calculate the expected laser scan by ray-casting in the map. We then classify the observation and obtain a probability distribution z over all the possible places according to Equation (8). The classification output z for each pose (x, y) is used to initialize the probability distribution $P_{(x,y)}^{(0)}$ of node $v_{(x,y)}$. For the nodes lying in the free space, the probability $p_{(x,y)}^{(0)}(wall)$ of being a wall is initialized with 0. Accordingly, the nodes corresponding to occupied cells in the map are initialized with $p_{(x,y)}^{(0)}(wall) = 1$.

Each of the weights $c_{ij} \in \mathcal{C}$ is initialized with the value $\frac{1}{8}$, indicating that all the eight neighbors v_j of node v_i are equally important. The compatibility coefficients are calculated using Equation (6). The values $p_i(l)$ and $p_{ij}(l \mid l')$ are obtained from statistics in the given occupancy grid map corresponding to previously labeled training data as shown in Fig. 1.

6.2 Region extraction and topological mapping

After the relaxation process, complete regions are extracted from the classification map using adjacent cells having the same labels. This is done using the algorithm by Rosenfeld and Pfaltz [34]. A topological graph is then constructed in which each node represents a region and each edge represents a connection between regions. We finally apply a heuristic region correction step to the topological map to increase the classification rate:

(1) We mark each region corresponding to a room or a corridor whose size does not exceed a given threshold of 1m^2 compared to the training set as classification error and assign the label of one of its connected regions to it.
(2) We mark each region labeled as doorway whose size does not exceed a given threshold of 0.1m^2 square meters or that is connected to only one region as false classification and assign the label of one of its connected regions to it.

6.3 Semantic classification of maps using associative Markov networks

In this section we present an alternative method for classifying and smoothing the corresponding labeling for creating topological maps. The main idea of the approach is to treat the classification of the poses in the geometric map as a collective classification problem [35]. Similarly to the relaxation labeling approach described in Section 6.1, collective methods also take into account the neighborhood of a cell when assigning a label to it. However, collective classification uses this influence in both ways: the label of a cell is influenced by its neighborhood and, at the same time, the neighboring labels of a cell are influenced by its label. In contrast, the relaxation labeling process uses only the neighboring information to change a certain label, but the neighborhood is not influenced.

One popular method for the task of collective classification are relational Markov networks (RMNs) [36]. In addition to the labels of neighboring points, RMNs also consider the relations between different objects. E.g., we can model the fact that two classes A and B are more strongly related to each other than, say, classes A and C. This modeling is done on the abstract class level by introducing clique templates [35]. Applying these clique templates to a given data set yields an ordinary Markov network (MN). In this MN, the result is a higher weighting of neighboring points with labels A and B than of points labeled A and C. Additionally, each node in the network is associated a set of features.

The whole process of labeling is composed of two steps. First, a supervised learning process is used to learn the parameters of the RMN used as a training set. Second, a new network is classified using these parameters. This last step is also called inference. In this work, we will use a special type of RMNs known as associative Markov networks (AMNs). Efficient algorithms are available for learning and inference in AMNs (for more detail see [37]).

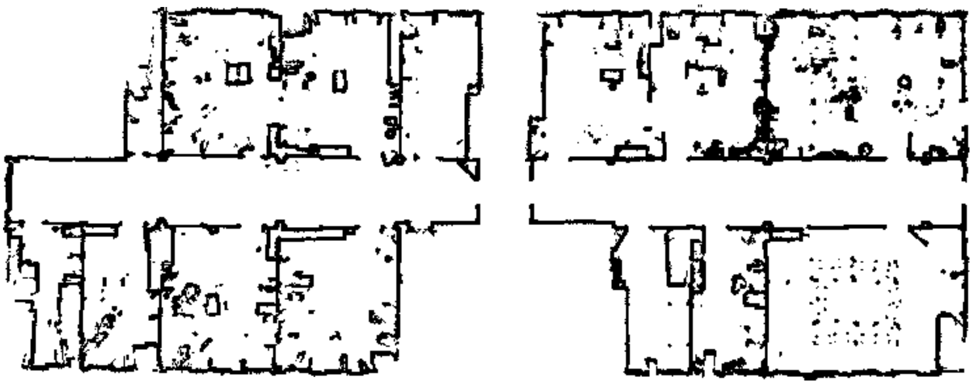

Fig. 6. Complete map of the ground floor of building 79 at the University of Freiburg. The map is divided into two parts for learning and classification purposes.

In our case we create an AMN in which each node represents a cell in the geometric map. Each node is given a semantic label corresponding to the place in the map (corridor, doorway or room). We also create a 8-neighborhood for each cell. Furthermore, a set of features is calculated for each cell. These features correspond to the geometric ones extracted from a simulated laser beam as explained in Section 4. To reduce the number of features during the training and inference steps, we select a subset of them. This selection is done using the AdaBoost algorithm [19].

7 Experiments

The approaches described above have been implemented and tested on real robots as well as in simulation. The robots used to carry out the real experiments were an ActivMedia Pioneer 2-DX8 equipped with two SICK lasers, an iRobot B21r robot equipped with a camera system and a laser, and an ActivMedia PowerBot equipped with a front laser. To simulate the laser scans in the different maps we used the software CARMEN [38]. This simulator adds gaussian noise to the laser measurements. In addition, it generates random readings and maximum readings with predefined probabilities.

The goal of the experiments is to demonstrate that our simple features can be boosted to a robust classifier of places. We first present experiments in which independent poses of the robot are classified into semantic classes. Next we show the two main applications of this approach. Firstly, we show an experiment in which the poses of the robot during a trajectory are classified using HMMs. Secondly, we present experiments where the toplogical map is created starting from a geometric map. Finally, we present some results in which we learned a classifier in different indoor environments and then we apply this classifier to unknown ones.

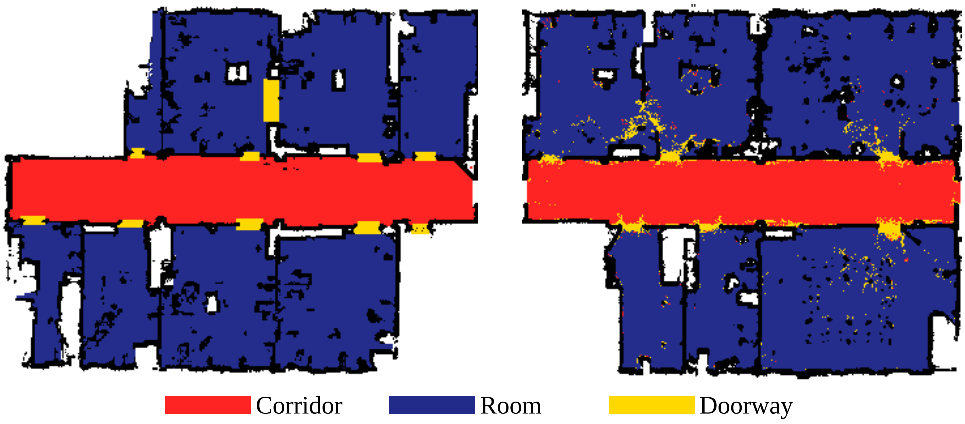

Fig. 7. The left image depicts the training data. The right image shows the test set with a classification rate of 97.3%. The training and test data were obtained by simulating the laser range scans in the corresponding grid maps.

7.1 Semantic classification of robot poses

The first experiment was performed using simulated data from our office environment in the building 79 at the University of Freiburg (Fig. 6). To obtain the laser observations we simulated in each free position in the map the corresponding laser scans. The task was to classify the poses of the robot into three different types of places, namely room, doorway, and corridor based on laser range data only. In this experiment we classified each pose independently without any filtering. The left part of the floor (Fig. 6 left) was used for training purposes, whereas the right part (Fig. 6 right) was used as test set. Fig. 7 shows the labeled training data together with the classification results on the test set. The optimal decision list for this classification problem, in which the robot had to distinguish between three classes, was room-doorway. This decision list correctly classifies 97.3% of all test examples (right image of Fig. 7).

7.2 Classification of trajectories using hidden Markov model filtering

The following experiment was performed using real laser and vision data obtained in the building 79 at the University of Freiburg (Fig. 6). The main goal of the experiment is to show how HMMs improve the classification on trajectories. In addition, we show how vision data can be included in a real experiment. For this purpose we use the set of features composed by the geometrical features extracted from the laser and the objects extracted from the images as explained in Section 4. In this experiment six different types of places were learned from the training examples, namely offices, doorways, a laboratory, a kitchen, a seminar room, and a corridor. We trained our classifier in the right part of the building 79 because here all six classes are represented

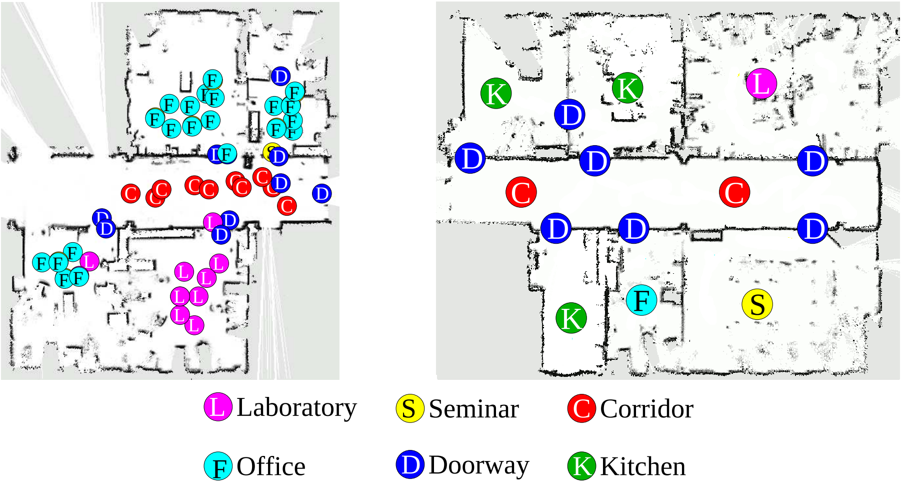

Fig. 8. The right image shows the training map hand-labeled with six different places. Here a big spot indicates the label for the complete area. The left image shows the classification using HMM of a typical trajectory in the other part of the floor. In this image each small spot indicates the classification of the robot while staying in that position.

(right image of Fig. 8). We then classified a typical trajectory along the left part of the floor applying a HMM as explain in Section 5. In this part of the floor there is no real kitchen or seminar room. The result of the classification is shown in the left image of Fig. 8. The classification rate was 91.2% using the decision list corridor-doorway-kitchen-lab-seminar. The classification rate decreases to 75.4 % if no HMM is applied.

7.3 Building topological maps

In the next experiment we create a topological map of the right part of the floor of building 79 at the University of Freiburg (Fig. 6) from its geometric map. The first step was to classify every free cell into one of the three places corresponding to door, corridor and room using only features extracted from simulated laser scans. The resulting classification was previously shown in the right image of Fig. 7. After applying our probabilistic relaxation method and heuristics (Section 6) we obtained the final toplogical map shown in Fig. 9. As can be seen the two left rooms above the corridor are detected as only one region. This is due to the fact that no correct doorway was detected in between.

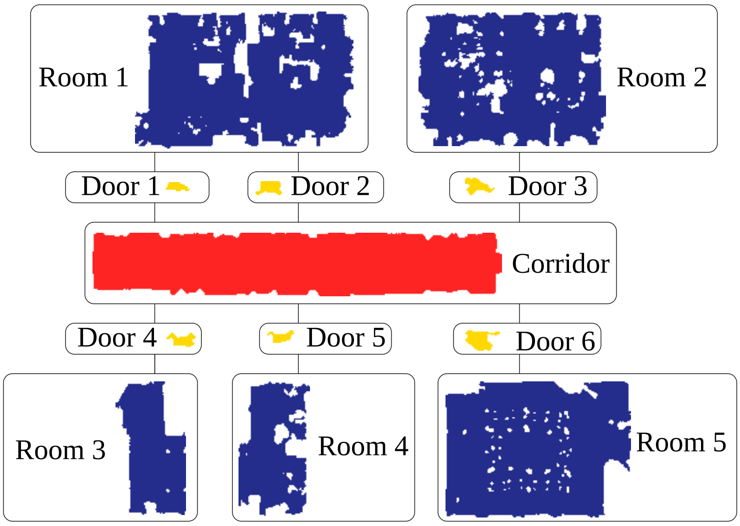

Fig. 9. Toplogical map from the right part of the floor of building 79.

7.4 Transferring the classifiers to unknown environments

These experiments are designed to analyze whether a classifier learned in a particular environment can be used to successfully classify the places of a new environment. To carry out the first experiment, we trained our sequential classifier simulating the laser scans in the left part of the map of Fig. 1, which corresponds to the building 52 at the University of Freiburg. The resulting classifier was then evaluated on scans simulated given the map of the Intel Research Lab in Seattle (Fig. 10 left). The resulting classification is shown in the right image of Fig. 10. Although the structure of the Intel Lab is very different from the one of the building 52, the resulting classification still seems to be consistent with the one possibly done by a human.

In a second experiment we learned a classifier using laser scans simulated in the maps of the buildings 79 and 52 at the University of Freiburg (Figs. 1 and 6). We then extract the topological map of the building denoted as "SDR site B" in the Radish [39] repository (Fig. 11 left), using the learned classifier together with the methods explained in Section 6. The result is shown in the right image of Fig. 11. For a more detailed description of this experiment we refer the reader to [26].

7.5 Learning semantic maps using associative Markov networks

In this experiment we show some preliminary results when applying AMNs to geometric maps as explained in Section 6.3. For this purpose we classified the right map of the building 79 at the University of Freiburg (Fig. 6 right). For the training process we used the left hand-labeled part of the map (Fig. 7

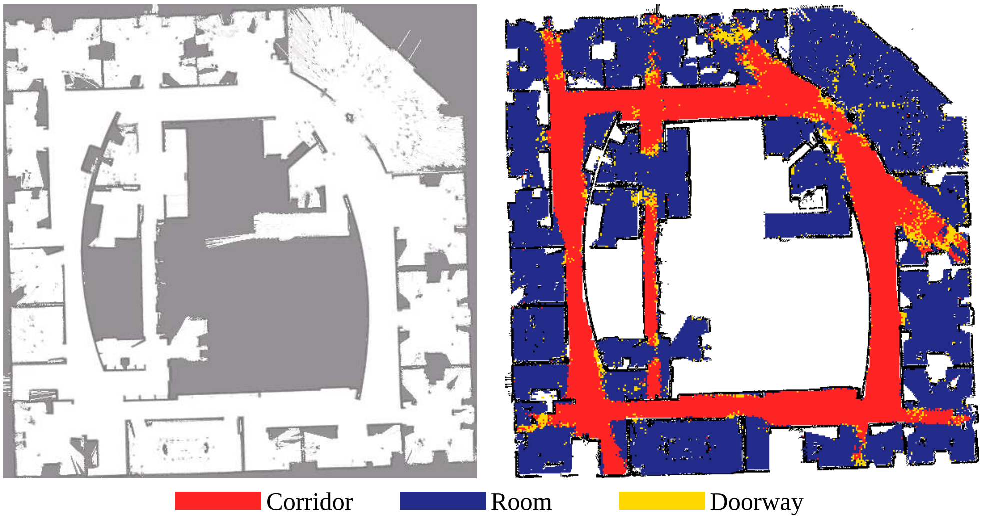

Fig. 10. The left map depicts the occupancy grid map of the Intel Research Lab and the right image depicts the classification results obtained by applying the classifier learned from the environment depicted in Fig. 1 to this environment.

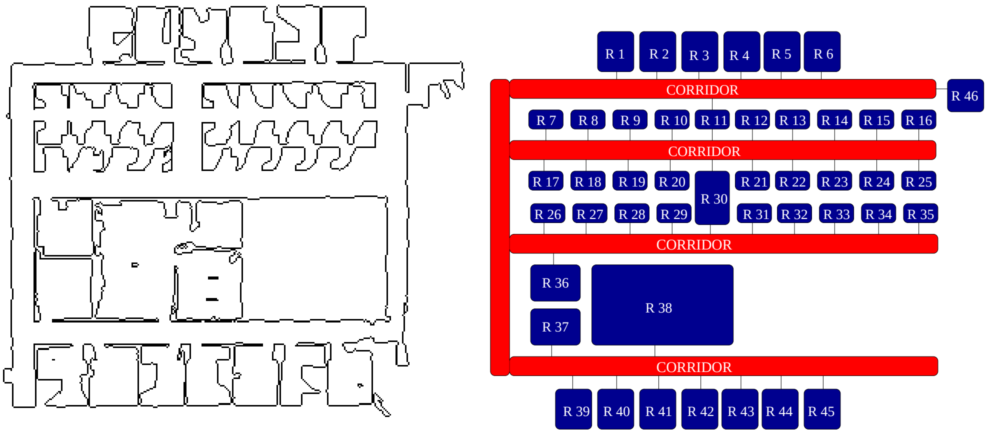

Fig. 11. The left image shows the map of the building "SDR site B". The right image depicts the corresponding topological map. The rooms are numbered left to right and top to bottom with respect the map in the left. For the sake of clarity, the corridor-node is drawn maintaining part of its region structure

left). In this experiment we reduce the resolution of the maps to 20cm. The reason for this is that the original resolution of 5cm generates a huge network which exceeds the memory resources of our computers during the training step of the corresponding AMN. The left image of Fig. 12 shows the results with a classification rate of 98.8% using AMNs. We compared this method with the classification obtained using our sequential AdaBoost together with the probabilistic relaxation procedure. The right image of Fig. 12 depicts the classification results. In this case, only 92.1% of the cells were correctly classified. As we can see, one consequence of changing the resolution to 20cm, is that the classification rate decreases (compare to right image of Fig. 7). We think that this is due to the worse quality of the simulated beams in such a granulated

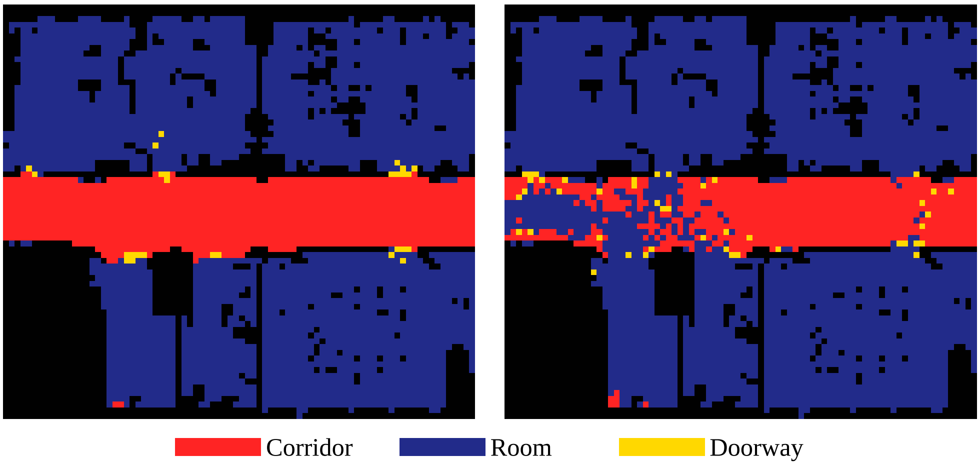

Fig. 12. The left image depicts a classification of 98.8% of the building 79 at University of Freiburg using AMNs. The right image shows the classification of the same building using the sequential AdaBoost classifier together with the probabilistic labeling method. In this case the classification rate was 92.1%. The training and test data were obtained by simulating laser range scans in the left map of Fig. 6 with a resolution of 20cm.

map. However, AMNs seem to be more robust to changes in resolution giving better classifications results.

7.6 Laser observations with restricted field of view

In this experiment we show the results of applying our classification methods when the laser range scan has a restricted field of view. No image data was used. We first steered a PowerBot robot equipped with only a front laser along the 6th floor of the CAS building at KTH (right to left). The trajectory is shown in the top image of Fig. 13. The data recorded in this floor was used to train the AdaBoost classifier. We then classified a trajectory on the 7th floor in the same building. We started the trajectory in an opposite direction (left to right). The rear beams were simulated using a local map. The resulting classification rate of 84.4% is depicted in the bottom image of Fig. 13.

8 Conclusions

In this paper, we presented an approach to classify different places in the environment of a mobile robot into semantic classes. Our algorithm uses simple geometric features extracted from a single laser range scan and applies the AdaBoost algorithm to form a binary strong classifier. To distinguish between more than two classes, we use a sequence of strong binary classifiers arranged

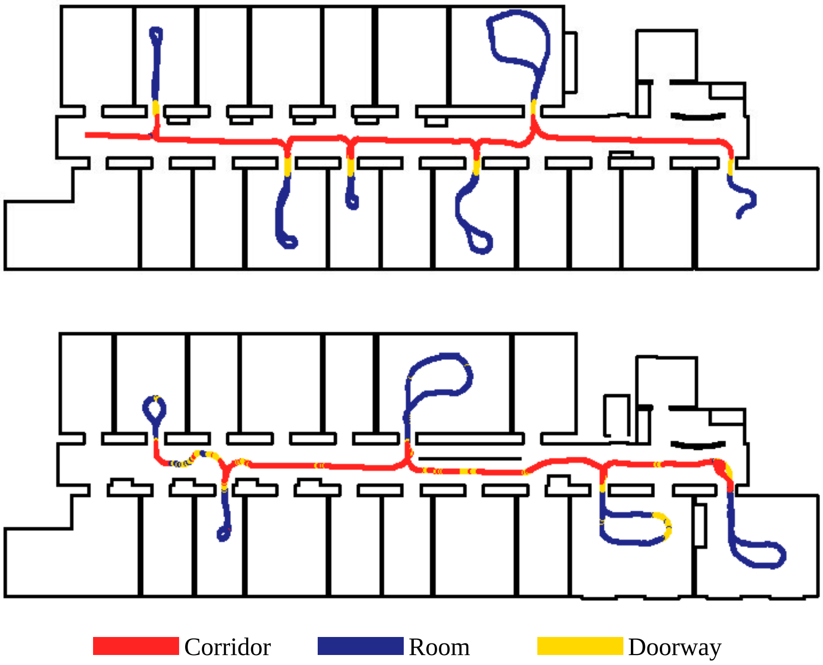

Fig. 13. The top image shows the training trajectory on the 6th floor of the CAS building at KTH. The bottom image depicts the labeling of the trajectory of the 7th floor using only a front laser with a classification rate of 84.4%. The map shown is for informative purposes only and does not represent exactly the environment in which the experiments were carried out.

in a decision list. Additionally we showed how to add features extracted from camera images.

We presented two applications of our approach. Firstly, we performed an online classification of the positions along the trajectories of a mobile robot by filtering the classification output using a hidden Markov model. Secondly, we presented a new approach to create topological graphs from occupancy grids by applying a probabilistic relaxation labeling to take into account dependencies between neighboring places when improving the final classifications. Additionally, we presented some preliminary results applying associative Markov networks to geometric maps to obtain a semantic classification of the poses.

Experiments carried out using real robots as well as in simulation illustrate that our technique is well-suited to reliably label places in different environments. It allows us to robustly separate different semantic regions and in this way it is able to learn topologies of indoor environments. Further experiments illustrate that a learned classifier can even be applied to so far unknown environments.

References

[1] R. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) (1999) 297–336.

[2] P. Buschka, A. Saffiotti, A virtual sensor for room detection, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2002, pp. 637–642.

[3] S. Koenig, R. Simmons, Xavier: A robot navigation architecture based on partially observable markov decision process models, in: D. Kortenkamp, R. Bonasso, R. Murphy (Eds.), Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems, MIT-Press, 1998, pp. 91–122.

[4] P. Althaus, H. Christensen, Behaviour coordination in structured environments, Advanced Robotics 17 (7) (2003) 657–674.

[5] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, S. Thrun, Learning hierarchical object maps of non-stationary environments with mobile robots, in: Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI), 2002, pp. 10–17.

[6] D. Anguelov, D. Koller, P. E., S. Thrun, Detecting and modeling doors with mobile robots, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2004, pp. 3777– 3784.

[7] B. Limketkai, L. Liao, D. Fox, Relational object maps for mobile robots, in: Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), Edinburgh, Scotland, 2005, pp. 1471–1476.

[8] A. Torralba, K. Murphy, W. Freeman, M. Rubin, Context-based vision system for place and object recognition, in: Proc. of the Int. Conf. on Computer Vision (ICCV), 2003.

[9] B. Kuipers, Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, Journal of Robotics & Autonomous Systems 8 (1991) 47–63.

[10] D. Kortenkamp, T. Weymouth, Topological mapping for mobile robots using a combination of sonar and vision sensing, in: Proc. of the Twelfth National Conference on Artificial Intelligence, 1994, pp. 979–984.

[11] H. Shatkey, L. Kaelbling, Learning topological maps with weak local odometric information, in: Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), 1997.

[12] S. Thrun, Learning metric-topological maps for indoor mobile robot navigation, Artificial Intelligence 99 (1) (1998) 21–71.

[13] H. Choset, Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization, IEEE Transactions on Robotics and Automation 17 (2001) 125–137.

[14] P. Beeson, N. K. Jong, B. Kuipers, Towards autonomous topological place detection using the extended voronoi graph, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2005.

[15] Z. Zivkovic, B. Bakker, B. Kröse, Hierarchical map building using visual landmarks and geometric constraints, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2005.

[16] A. Tapus, R. Siegwart, Incremental robot mapping with fingerprints of places, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2005, pp. 2429– 2434.

[17] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: Proc. of the European Conference on Computational Learning Theory (EuroCOLT), 1995, pp. 23–37.

[18] P. Viola, M. Jones, Robust real-time object detection, in: Proc. of IEEE Workshop on Statistical and Theories of Computer Vision, 2001.

[19] O. Martínez Mozos, C. Stachniss, W. Burgard, Supervised learning of places from range data using adaboost, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), Barcelona, Spain, 2005, pp. 1742–1747.

[20] A. Rottmann, Bild- und laserbasierte klassifikation von umgebungen mit mobilen robotern, Master's thesis, University of Freiburg, Department of Computer Science, in German (2005).

[21] R. Gonzalez, P. Wintz, Digital Image Processing, Addison-Wesley Publishing Inc., 1987.

[22] R. Haralick, L. Shapiro, Computer and Robot Vision, Addison-Wesley Publishing Inc., 1992.

[23] S. Loncaric, A survey of shape analysis techniques, Pattern Recognition 31 (8).

[24] J. O'Rourke, Computational Geometry in C (2nd Edition), Cambridge University Press, 1998.

[25] J. Russ, The Image Processing Handbook, CRC Press, 1992.

[26] O. Martínez Mozos, W. Burgard, Supervised learning of topological maps using semantic information extracted from range data, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 2772–2777.

[27] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, in: Die Deutsche Arbeitsgemeinschaft fuer Mustererkennung (DAGM), 2003, pp. 297–304.

[28] A. Rottmann, O. Martínez Mozos, C. Stachniss, W. Burgard, Place classification of indoor environments with mobile robots using boosting, in: Proc. of the National Conference on Artificial Intelligence (AAAI), Pittsburgh, PA, USA, 2005, pp. 1306–1311.

[29] H. P. Moravec, Sensor fusion in certainty grids for mobile robots, AI Magazine (1988) 61–74.

[30] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT Press, 2006.

[31] A. Rosenfeld, R. A. Hummel, S. W. Zucker, Scene labeling by relaxation operations, IEEE Trans. Systems. Man. Cybernet 6 (6) (1976) 420–433.

[32] H. Yamamoto, A method of deriving compatibility coefficents for relaxation operators, Compt. Graph. Image Processing 10 (1979) 256–271.

[33] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Annals of Statistics 28 (2) (2000) 337–407.

[34] A. Rosenfeld, J. L. Pfaltz, Sequential operations in digital picture processing, Journal of the Association for Computing Machinery 13 (4) (1966) 471–494.

[35] S. Chakrabarti, B. Dom, P. Indyk, Enhanced hypertext categorization using hyperlinks, in: SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, ACM Press, New York, NY, USA, 1998, pp. 307–318.

[36] B. Taskar, P. Abbeel, D. Koller, Discriminative probabilistic models for relational data, in: Proc. Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI), Edmonton, Canada, 2002.

[37] R. Triebel, K. Kersting, W. Burgard, Robust 3D scan point classification using associative markov networks, in: ”Proc. of the International Conference on Robotics and Automation(ICRA)”, 2006, pp. 2603–2608.

[38] M. Montemerlo, N. Roy, S. Thrun, Perspectives on standardization in mobile robot programming, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2003.

[39] A. Howard, N. Roy, Radish: The robotics data set repository.
URL http://radish.sourceforge.net/