Semantic Labeling of Places

C. Stachniss O. Martínez-Mozos A. Rottmann W. Burgard

University of Freiburg, Dept. of Computer Science, D-79110 Freiburg, Germany

Abstract Indoor environments can typically be divided into places with different functionalities like corridors, kitchens, offices, or seminar rooms. We believe that such semantic information enables a mobile robot to more efficiently accomplish a variety of tasks such as human-robot interaction, path-planning, or localization. In this paper, we propose an approach to classify places in indoor environments into different categories. Our approach uses AdaBoost to boost simple features extracted from vision and laser range data. Furthermore, we apply a Hidden Markov Model to take spatial dependencies between robot poses into account and to increase the robustness of the classification. Our technique has been implemented and tested on real robots as well as in simulation. Experiments presented in this paper demonstrate that our approach can be utilized to robustly classify places into semantic categories.

1 Introduction

In the past, many researchers have considered the problem of building accurate metric or topological 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 be important.

In this paper, we address the problem of classifying places of the environment of a mobile robot using range finder data and vision features. Indoor environments, like the one depicted in Figure 1, can typically be divided into areas with different functionalities such as laboratories, office rooms, corridors, or kitchens. Some of these places have special geometric structures and can therefore be distinguished merely based on laser range data. The types of other places, however, can only be identified according to the objects located at them. For example, a coffee machine can typically be found in the kitchen. To detect such objects, we use vision data acquired by a camera system.

In the approach described here, we apply the AdaBoost algorithm [6] to boost simple features, which on their own are insufficient for a reliable cate-

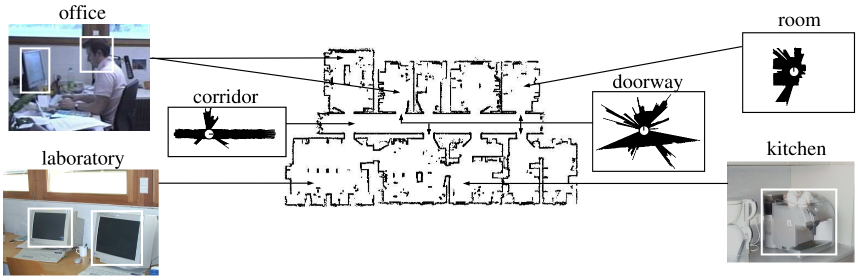

Figure 1. An environment with offices, doorways, a corridor, a kitchen, and a laboratory. Additionally, the figure shows typical observations obtained by a mobile robot at different places.

gorization of places, to a strong classifier for place labeling. Since the original version of AdaBoost provides only binary decisions, we determine the decision list with the best sequence of binary strong classifiers. To take spatial dependencies into account, we apply a Hidden Markov Model (HMM) which estimates the label of the current pose based on the current and the previous outputs of the sequence of binary strong classifiers. Experimental results shown in this paper illustrate that our classification system yields recognition rates of more than 91% or 93% (depending on the number of classes to distinguish between). We also present experiments illustrating that the resulting classifier can even be used in environments from which no training data were available.

In the past, several authors considered the problem of adding semantic information to places. Buschka and Saffiotti [4] describe a virtual sensor that is able to identify rooms from range data. Also Koenig and Simmons [9] apply a pre-programmed routine to detect doorways from range data. Althaus and Christensen [1] use line features to detect corridors and doorways. Some authors also apply learning techniques to localize the robot or to identify distinctive states in the environment. For example, Oore *et al.* [15] train a neural network to estimate the location of a mobile robot in its environment using the odometry information and ultrasound data. Kuipers and Beeson [10] apply different learning algorithms to learn topological maps of the environment.

Additionally, learning algorithms have been used to identify objects. For example, Anguelov *et al.* [2, 3] 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.* [12] use relational Markov networks to detect objects like doorways based on laser range data. Furthermore, they employ Markov chain Monte Carlo to learn the parameters of the models. Treptow *et al.* [19] utilize the AdaBoost algorithm to track a soccer ball without color information. In a recent work, Torralba and colleagues [18] use Hidden Markov Models for learning places from image data.

Compared to the other approaches, our algorithm is able to combine arbitrary features extracted from different sensors to form a sequence of strong classifiers to label places. Our approach is also supervised, which has the advantage that the resulting labels correspond to user-defined classes.

2 The AdaBoost Algorithm

Boosting is a general method for creating an accurate strong classifier by combining a set of weak classifiers. The requirement for each weak classifier is that its accuracy is better than a random guessing. In this work, we apply the AdaBoost algorithm which has originally been introduced by Freund and Schapire [6]. The input to this algorithm is a set of labeled training examples. In a series of T rounds, the algorithm repeatedly selects a weak classifier $h_j(x)$ using a distribution D over the training examples. The selected weak classifier is expected to have a small classification error on the training data. The idea of the algorithm is to modify the distribution D by increasing the weights of the most difficult training examples on each round. The final strong classifier H is a weighted majority vote of the T best weak classifiers.

Throughout this work, we use the approach presented by Viola and Jones [20] in which the weak classifiers depend on single-valued features $f_j \in \Re$. Two kinds of weak classifiers are created in our current system. The first type is defined as by Viola and Jones and has the form

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

where θ_j is a threshold and p_j is either -1 or $+1$ and thus represents the direction of the inequality. We designed a second type of weak classifier which has the form

$$h_j(x) \quad = \quad \begin{cases} p_j & \text{if } \theta_j^1 < f_j(x) < \theta_j^2 \\ -p_j & \text{otherwise,} \end{cases} \tag{2}$$

where θ_j^1 and θ_j^2 define an interval and p_j is either $+1$ or -1 indicating whether examples inside the interval are positive or negative. For both types of weak classifiers, the output is $+1$ or -1 indicating whether the classification is positive or negative. The AdaBoost algorithm determines for each weak classifier $h_j(x)$ the optimal parameters, such that the number of misclassified training examples is minimized.

The AdaBoost algorithm has been designed for binary classification problems. To classify places in the environment, we need the ability to handle multiple classes. To achieve this, we use a sequence of binary classifiers, where each element of such a sequence determines if an example

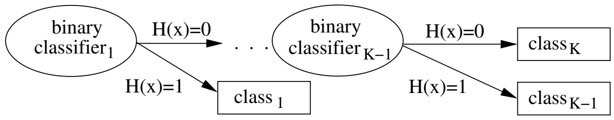

Figure 2. A decision list classifier for K classes using binary classifiers.

belongs to one specific class. If the classifier returns a positive result, the example is assumed to be correctly classified. Otherwise, it is recursively passed to the next element in this list. Figure 2 illustrates the structure of such a decision list classifier.

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. Although this approach is exponential in the number of classes, the actual number of permutations considered is limited in our domain due to the small number of classes. In practice, we found out that the heuristic which sorts the classifiers in increasing order according to their classification rate also yields good results and at the same time can be computed efficiently. In several situations, the sequence generated by this heuristic turned out to be the optimal one or very close to it [16].

3 Features from Vision and Laser Data

In this section, we describe the features used to create the weak classifiers in the AdaBoost algorithm. Our robot is equipped with a 360 degree field of view laser sensor and a camera. Each laser observation consists of 360 beams. Each vision observation consists of 8 images which form a panoramic view. Figure 1 illustrates different images and laser range readings taken in an office environment. Accordingly, each training example for the AdaBoost algorithm consist of one laser observation, one vision observation, and its classification.

Our method for place classification is based on single-valued features extracted from laser and vision data. All features are invariant with respect to rotation to make the classification of a pose dependent only on the position of the robot and not of its orientation. Most of our laser features are standard geometrical features used for shape analysis [8, 17]. Typical examples considered by our system are illustrated in Figure 3. The complete list of laser features is provided by Martínez-Mozos *et al.* [14].

In the case of vision, the selection of the features is motivated by the fact that typical objects appear with different probabilities at different places. For example, the probability of detecting a computer monitor is larger in an office than in a kitchen. For each type of object, a vision feature is defined as a function that takes as argument a panoramic vision observation and returns the number of detected objects of this type in it. This number rep-

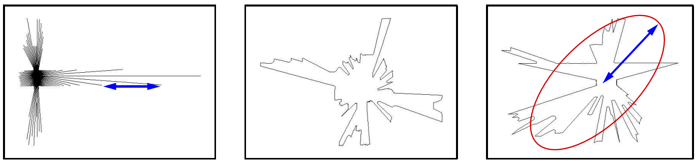

Figure 3. Examples for features generated from laser data, namely the average distance between two consecutive beams, the perimeter of the area covered by a scan, and the mayor axis of the ellipse that approximates the polygon described by the scan.

resents the single-valued feature f_j within AdaBoost according to Eq. (1) and Eq. (2). In our case, we consider monitors, coffee machines, soap dispensers, office cupboards, frontal faces, face profiles, full human bodies, and upper human bodies. An example of such objects is shown in Figure 1. The individual objects are detected using classifiers also trained with AdaBoost and based on the set of Haar-like features proposed by Lienhart *et al.* [11].

In case the observations do not cover a 360 degree field of view, the property of the rotational invariance is lost. In such a situation, we expect that much more training data will be necessary and that the classification will be less robust.

4 Probabilistic Place Recognition

The approach described so far is able to classify single observations only but does not take into account past classifications when determining the type of place the robot is at. However, whenever a mobile robot moves through an environment, the semantic labels of nearby places are typically identical. Furthermore, certain transitions between classes are unlikely. For example, if the robot is currently in a kitchen then it is rather unlikely that the robot ends up in an office given it moved a short distance only. In many environments, to get from the kitchen to the office, the robot typically has to move through a doorway first.

To incorporate such spatial dependencies between the individual classes, we apply a Hidden Markov Model (HMM) and maintain a posterior $Bel(\xi_t)$ about the type of the place ξ_t the robot is currently at

$$Bel(\xi_t) \quad = \quad \alpha P(z_t \mid \xi_t) \sum_{\xi_{t-1}} P(\xi_t \mid \xi_{t-1}, u_{t-1}) Bel(\xi_{t-1}). \qquad (3)$$

In this equation, α is a normalizing constant ensuring that the left-hand side sums up to one over all ξ_t. To implement this HMM, three components need to be known. First, we need to specify the observation model $P(z_t \mid \xi_t)$ which is the likelihood that the classification output is z_t given the actual class is ξ_t. Second, we need to specify the transition model

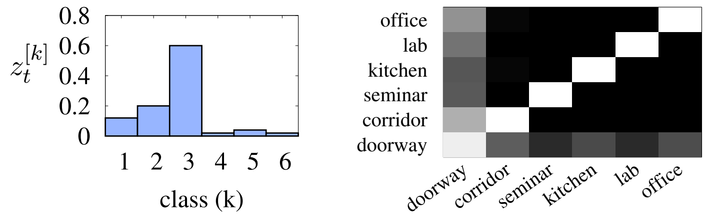

Figure 4. The left image illustrates a classification output z. The right image depicts probabilities of possible transitions between places in the environment. To increase the visibility, we used a logarithmic scale. Dark values indicate low probability.

$P(\xi_t \mid \xi_{t-1}, u_{t-1})$ which defines the probability that the robot moves from class ξ_{t-1} to class ξ_t by executing action u_{t-1}. Finally, we need to specify how the belief $Bel(\xi_0)$ is initialized.

In our current system, we choose a uniform distribution to initialize $Bel(\xi_0)$. Furthermore, the classification output z_t is represented by a histogram, as illustrated in the left image of Figure 4. 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 (compare Figure 2). To compute the individual values for each bin of that histogram, we use the approach by Friedman *et al.* [7]. It determines a confidence value $C \in [0, 1]$ for a positive binary classification with AdaBoost. Let C_k refer to the confidence value of the k-th binary classifier in our decision list. Then, the probability that the location to be classified belongs to the k-th class is given by the k-th bin of the histogram z computed as

$$z^{[k]} = C_k \prod_{j=1}^{k-1} (1 - C_j), \tag{4}$$

whereas for the confidence value C_K, used to compute the last bin ($z^{[K]}$) of the histogram, holds $C_K = 1$ according to the structure of the decision list (compare Figure 2).

To determine $P(z_t \mid \xi_t)$, we use the KL-divergence [5] between two distributions. The first distribution is the current classification output z_t. The second one is learned from a statistics: for each class ξ, we compute a histogram $\hat{z}_{1:h}(\xi)$ using h observations recorded within a place belonging to class ξ (here $h = 50$). This histogram $\hat{z}_{1:h}(\xi)$ is obtained by averaging over the individual histograms $\hat{z}_1, \ldots, \hat{z}_h$, which are computed according to Eq. (4). To determine $P(z_t \mid \xi_t)$, we use the KL-divergence $kld(\cdot \parallel \cdot)$ which provides a measure about the similarity of two distributions

$$P(z_t \mid \xi_t) = e^{-kld(z_t \parallel \hat{z}_{1:h}(\xi_t))^2}. \tag{5}$$

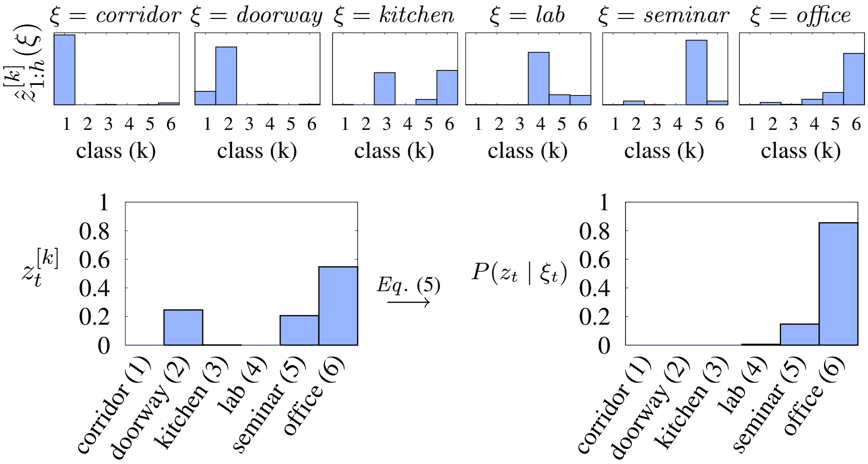

Figure 5. The distributions depicted in the first row show the learned histograms $\hat{z}_{1:h}(\xi)$ for the individual classes (here corridor (1), doorway (2), kitchen (3), lab (4), seminar room (5), and office (6)). The left image in the second row depicts a possible classification output z_t. In the right image, each bar represents the corresponding likelihood $P(z_t \mid \xi_t)$ for the different estimates of ξ_t.

To illustrate the computation of the observation likelihood $P(z_t \mid \xi_t)$ consider Figure 5. The first row depicts examples for the histograms $\hat{z}_{1:h}(\xi)$. The left image in the second row depicts the output z_t of the sequential classifier while the robot was in an office. As can be seen, also the classes doorway and seminar room have a probability significantly larger than zero. This output z_t and the histogram $\hat{z}_{1:h}(\xi_t)$ is than used to compute $P(z_t \mid \xi_t)$ according to Eq. (5). The result for all classes is depicted in the right image in the second row. In this image, each bin represents the likelihood $P(z_t \mid \xi_t)$ for the individual classes ξ_t. As can be seen, the observation likelihood given the robot is in a doorway is close to zero, whereas likelihood given it is in an office is around 90%, which is actually the correct class.

To realize the transition model $P(\xi_t \mid \xi_{t-1}, u_{t-1})$, we only consider the two actions $u_{t-1} \in \{Move, Stay\}$. The transition probabilities were learned in a manually labeled environment by running 1000 simulation experiments. In each run, we started the robot at a randomly chosen point and orientation. We then executed a random movement so that the robot traveled between 20cm and 50cm. These values corresponds to typical distances traveled by the robot between two consecutive updates of the HMM. The finally obtained transition probability matrix $P(\xi_t \mid \xi_{t-1}, u_{t-1})$ for the action $Move$ is depicted in the right image of Figure 4. 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

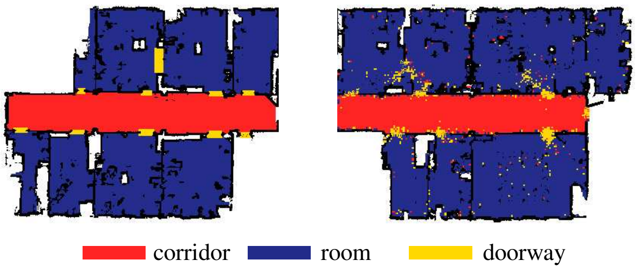

Figure 6. Whereas the left image depicts the training data, the right image shows the classification result on the test set. The training and test data were obtained by simulating laser range scans in the map.

from a room to a doorway is higher than the probability of moving from a room directly to a corridor. This indicates that the robot typically has to cross a doorway first in order to reach a different room. Furthermore, the matrix shows a lower probability of staying in a doorway than staying the the same type of 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.

5 Experiments

The approach described above has been implemented and tested on a real robot as well as in simulation. The robots used to carry out the experiments were an ActivMedia Pioneer 2-DX8 equipped with two SICK laser range finders as well as an iRobot B21r robot which is additionally equipped with a camera system.

The goal of the experiments is to demonstrate that our simple features can be boosted to a robust classifier of places. Additionally, we analyze whether the resulting classifier can be used to classify places in environments for which no training data were available. Furthermore, we demonstrate the advantages of utilizing the vision information to distinguish between different rooms like, e.g., kitchens, offices, or seminar rooms. Additionally, we illustrate the advantages of the HMM filtering for classifying places with a moving mobile robot. Finally, we briefly present results comparing our sequential AdaBoost classifier with a multi-class variant of AdaBoost, called AdaBoost.M2 [6]. Throughout our experiments, the term classification result refers to the most likely class reported by the HMM or respectively by the sequence of binary classifiers.

5.1 Results with the sequential classifier using Laser Data

The first experiment was performed using simulated data from our office environment in building 79 at the University of Freiburg. The task was to distinguish between three different types of places, namely rooms, door-

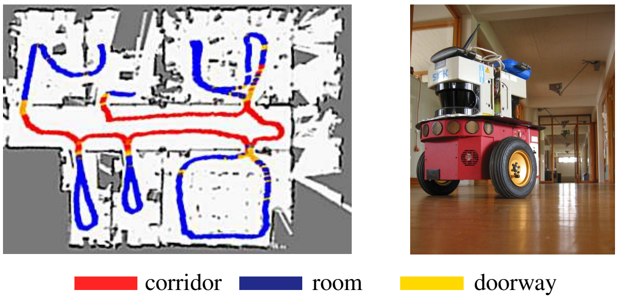

Figure 7. The left image depicts a trajectory of a robot and the corresponding classifications based on real laser data. The robot used in this experiment is depicted in the right image.

ways, and a corridor based on laser range data only. In this experiment, we solely applied the sequential classifier without the HMM filtering. For the sake of clarity, we separated the test from the training data by dividing the overall environment into two areas. Whereas the left part of the map contains the training examples, the right part includes only test data (see Figure 6). The optimal decision list for this classification problem in which the robot had to distinguish between three classes is room-doorway. This decision list correctly classifies 93.9% of all test examples (see right image of Figure 6). The worst configurations of the decision list are those in which the doorway classifier is in the first place. This is probably due to the fact, that doorways are hard to detect and typically most parts of a range scan obtained in a doorway cover the adjacent rooms or the corridor. Note that we obtained similar success rates with alternative training and test sets.

The next experiment has been carried out with a real mobile robot that we manually steered through the environment. We used the same classifier as in the previous experiments. The trajectory including the corresponding classification results as well as the mobile robot are depicted in Figure 7. As can be seen from this figure, the learned classifier yields a robust labeling also for real robot data.

Additionally, we performed an experiment using a map of the entrance hall at the University of Freiburg which contained four different classes, namely rooms, corridors, doorways, and hallways. The optimal decision list is corridor-hallway-doorway with a success rate of 89.5%.

5.2 Transferring the Classifiers to New Environments

The second experiment is 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 this experiment, we trained our sequential classifier in the left half of the map shown in Figure 1. In the right half of this environment, our approach was able to correctly classify 92.1% of all places. The resulting classifier was then evaluated on scans simulated

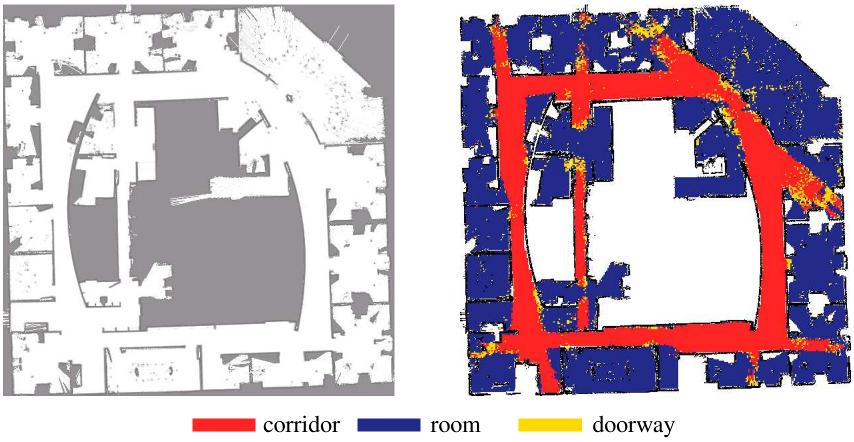

Figure 8. The left map depicts the occupancy grid map of the Intel Research Lab and the right image depicts the classification results obtained by applying the classifier learned from the environment depicted in Figure 1 to this environment. The fact that 82.2% of all places could be correctly classified illustrates that the resulting classifiers can be applied to so far unknown environments.

given the map of the Intel Research Lab in Seattle depicted in Figure 8. For these scans the classification rate decreased to 82.2%. This indicates that our Algorithm yields good generalizations which can also be applied to correctly label places of so far unknown environments. Note that a success rate of 82.2% is quite high for this environment, since even humans typically cannot consistently/correctly classify the places in this environment.

5.3 Improving Robustness using HMM Filtering

The third experiment was performed using real laser and vision data obtained in a typical office environment, which contains six different types of places, namely offices, doorways, a laboratory, a kitchen, a seminar room, and a corridor. The true classification of the different places in this environments is shown in Figure 9.

The classification performance of the classifier on a typical real data test set is shown in in left image of Figure 10. The classification rate in this experiment is 73.7%. If we additionally apply the HMM to estimate the type of the place, the classification rate increases up to 90.9%. The labeling obtained with the HMM is shown in the right image of Figure 10.

A further experiment was carried out using test data obtained in a different part of the same building. We applied the same classifier as in the previous experiment. Whereas the sequential classifier yields a classification rate of 75.4%, the combination with the HMM generated the correct answer in 91.2% of all cases. A two-sample t-test applied to the classification results obtained along the trajectories for both experiments showed that the improvements introduced by the HMM are significant. Furthermore,

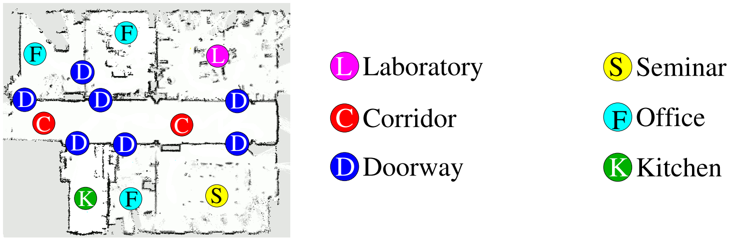

Figure 9. Ground truth labeling of the individual areas in the environment.

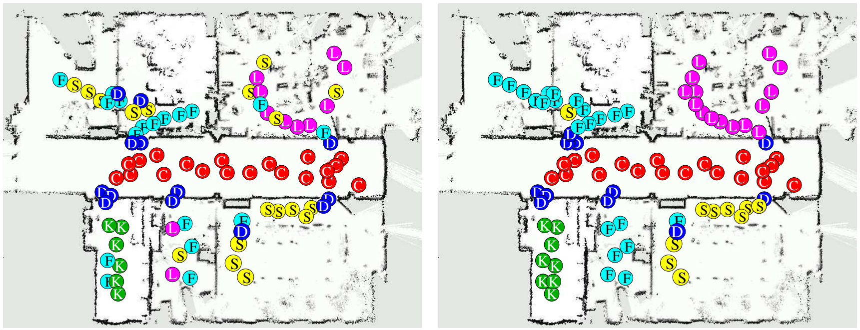

Figure 10. The left image depicts a typical classification result for a test set obtained using only the output of the sequence of classifiers. The right image shows the resulting classification in case a HMM is additionally applied to filter the output of the sequential classifier.

we classified the same data ignoring the vision information and based only on the laser features. In this case, only 54.4% could be classified correctly without the HMM. The application of the HMM increases the classification performance to 66.7%. These three experiments illustrate that the HMM seriously improves the overall rate of correctly classified places.

5.4 Comparison of the Sequential Classifier with AdaBoost.M2

Our current system uses a sequence of strong binary classifiers arranged in a decision list. To evaluate this approach, we compared it to AdaBoost.M2 [6], which is a multi-class variant of AdaBoost. In all our experiments, the optimal sequential classifiers performed better than AdaBoost.M2. Table 1 provides a quantitative analysis of the classification performance for three different environments. As can be seen, our sequential AdaBoost classifier yields better results than the AdaBoost.M2 algorithm. A more detailed comparison between both algorithms can be found in the work by Martínez-Mozos [13].

We also evaluated the performance of the system when the order of the binary strong classifiers is chosen according to their classification rate. Compared to the optimal order, the classifier generated by the heuristic for six different classes performed in average only 1.3% worse.

Table 1. Classification results for different classifiers.

Environment	Seq. Classifier %	AdaBoost.M2 %
depicted in Figure 1	92.1	91.8
depicted in Figure 6	93.9	83.8
Univ. of Freiburg, entrance hall	89.5	82.3

6 Conclusion

In this paper, we presented a novel approach to classify different places in the environment of a mobile robot into semantic classes, like rooms, hallways, corridors, offices, kitchens, or doorways. Our algorithm uses simple geometric features extracted from a single laser range scan and information extracted from camera data and applies the AdaBoost algorithm to form a strong classifier. To distinguish between more than two classes, we use a sequence of strong binary classifiers arranged in a decision list. We furthermore use a Hidden Markov Model to take into account the spatial dependencies between places. Experiments carried out on a real robot as well as in simulation illustrate that our technique is well-suited to reliably label places in different environments. Further experiments illustrate that a learned classifier can even be applied to so far unknown environments. Finally, we compared our sequential AdaBoost classifier to AdaBoost.M2, a multi-class variant of the AdaBoost algorithm. In our experiments, the sequential classifier always outperformed AdaBoost.M2.

We belive that these results open new directions for future research. First, semantic labels can be used to facilitate loop-closing actions during exploration and SLAM. Furthermore, the extraction of labels will support natural interaction with users.

Acknowledgment

This work has partly been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 (A3) and by the EC under contract number FP6-004250-CoSy.

References

[1] P. Althaus and H.I. Christensen. Behaviour coordination in structured environments. *Advanced Robotics*, 17(7):657–674, 2003.

[2] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and 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.

[3] D. Anguelov, D. Koller, Parker E., and S. Thrun. Detecting and modeling doors with mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2004.

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

[5] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.

[6] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

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

[8] R.C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing Inc., 1987.

[9] S. Koenig and R. Simmons. Xavier: A robot navigation architecture based on partially observable markov decision process models. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, pages 91–122. MIT-Press, 1998.

[10] B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI)*, 2002.

[11] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM, 25th Pattern Recognition Symposium*, 2003.

[12] B. Limketkai, L. Liao, and D. Fox. Relational object maps for mobile robots. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1471–1476, Edinburgh, Scotland, 2005.

[13] O. Martínez-Mozos. Supervised learning of places from range data using adaboost. Master's thesis, University of Freiburg, Department of Computer Science, 2004.

[14] O. Martínez-Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using adaboost. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1742–1747, Barcelona, Spain, 2005.

[15] S. Oore, G.E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9(3):683–699, 1997.

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

[17] J.C. Russ. *The Image Processing Handbook*. CRC Press, 1992.

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

[19] A. Treptow, A. Masselli, and A. Zell. Real-time object tracking for soccer-robots without color information. In *Proc. of the Europ. Conf. on Mobile Robots (ECMR)*, 2003.

[20] P. Viola and M.J. Jones. Robust real-time object detection. In *Proc. of IEEE Workshop on Statistical and Theories of Computer Vision*, 2001.