Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Supervised Learning of Places from Range Data using AdaBoost

Oscar Martinez Mozos Cyrill Stachniss Wolfram Burgard

University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany

***Abstract*— This paper addresses the problem of classifying places in the environment of a mobile robot into semantic categories. We believe that semantic information about the type of place improves the capabilities of a mobile robot in various domains including localization, path-planning, or human-robot interaction. Our approach uses AdaBoost, a supervised learning algorithm, to train a set of classifiers for place recognition based on laser range data. In this paper we describe how this approach can be applied to distinguish between rooms, corridors, doorways, and hallways. Experimental results obtained in simulation and with real robots demonstrate the effectiveness of our approach in various environments.**

I. 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. For a lot of applications, robots can improve their service if they are able to recognize places and distinguish them. A robot that possesses semantic information about the type of the places can easily be instructed, for example, to "open the door to the corridor, please."

In this work we address the problem of classifying locations of the environment using range finder data. Indoor environments, like the one depicted in Figure 1 can typically be decomposed into areas with different functionalities such as office rooms, corridors, hallways, or doorways. Generally, each of these places has a different structure. For example, the bounding box of a corridor is usually longer than that of rooms and hallways. Furthermore, rooms are typically smaller than hallways and also are more cluttered than corridors or hallways.

The key idea of this paper is to classify the position of the robot based on the current scan obtained from the range sensor. Examples for typical range scans obtained in an office environment are shown in Figure 2. Our approach uses the AdaBoost algorithm [5] to boost simple geometrical scan-features, which on their own are insufficient for a reliable categorization of places, to a strong classifier. Each individual feature is a numerical value computed from the beams of a laser range scan as well as from a polygon representation of the covered area. Since AdaBoost provides only binary decisions, we determine the decision list with the best sequence of binary classifiers. Experimental results shown in this paper illustrate that the resulting classification system can determine the type of the place with a recognition rate of more than 89%. We also present results illustrating that the resulting classifier can even be used in environments from which no training data were available. We also compared our sequential AdaBoost classifier with an extension of AdaBoost to multiple classes. Experiments illustrate that the sequential version provides better results than the multi-class AdaBoost.

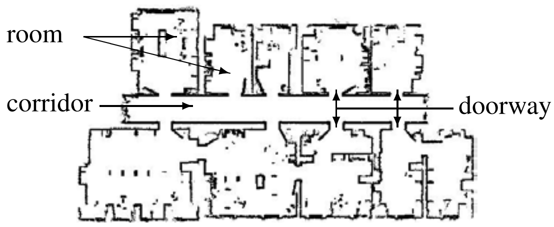

Fig. 1. Example environment containing rooms, doorways and a corridor.

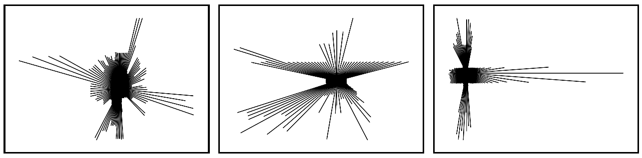

Fig. 2. Example scans recorded in a room, a doorway, and a corridor.

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 [8] use 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.* [13] 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 [9] apply different learning algorithms to learn topological maps of the environment.

Additionally, learning algorithms have been used to identify objects. For example, Anguelov and colleagues [2, 3] apply the EM algorithm to cluster different types of objects from sequences of range data. Treptow *et al.* [18] use the AdaBoost algorithm to track a ball without color information in the context of RoboCup. In a recent work, Torralba and colleagues [17] use Hidden Markov Models for learning places from image data.

Compared to these approaches, our algorithm does not require any pre-defined routines for extracting high-level features. Instead, it uses the AdaBoost algorithm to boost simple features to strong classifiers for place categorization. Our approach is also supervised, which has the advantage that the resulting semantic labels correspond to user-defined classes.

This paper is organized as follows. In the next section we describe the AdaBoost algorithm as well as its extension to multiple classes and our sequential multi-class variant. In Section IV we then introduce our features extracted from laser range scans. Finally, in Section V we present experimental results obtained with our approach.

II. The AdaBoost Algorithm

The original AdaBoost algorithm, which has been introduced by Freund and Shapire [5], is a supervised learning algorithm designed to find a binary classifier that discriminates between *positive* and *negative examples*. The input to the learning algorithm is a set of training examples $(x_n, y_n), n = 1, \ldots, N$, where each x_n is an example and y_n is a boolean value indicating whether x_n is a positive or negative example. AdaBoost boosts the classification performance of a simple learning algorithm by combining a collection of weak classifiers to a stronger classifier. Each weak classifier is given as a function $h_j(x)$ which returns a boolean value. The output is 1, if x is classified as a positive example and 0 otherwise. Whereas the weak classifiers only need to be slightly better than a random guessing, the combined strong classifier typically produces good results. To boost a weak classifier, it is applied to solve a sequence of learning problems. After each round of learning, the examples are reweighted in order to increase the importance of those which were incorrectly classified by the previous weak classifier. The final strong classifier takes the form of a perceptron, a weighted combination of weak classifiers followed by a threshold. Large weights are assigned to good classification functions whereas poor functions have small weights.

Throughout this work we apply the variant of the AdaBoost algorithm presented by Viola and Jones [19]. This variant restricts the weak classifiers to depend on single-valued features f_j only. Each weak classifier has the form

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

where θ_j is a threshold and p_j is either -1 or 1 and thus representing the direction of the inequality. The algorithm determines for each weak classifier $h_j(x)$ the optimal values for θ_j and p_j, such that the number of misclassified training examples is minimized. To achieve this, it considers all possible combinations of both p_j and θ_j, whose number is limited since only an finite number of training examples is given:

$$(p_j, \theta_j) \quad = \quad \operatorname*{argmin}_{(\theta_i, p_i)} \sum_{n=1}^{N} |h_i(x_n) - y_n| \quad (2)$$

The resulting algorithm is given in Table I.

TABLE I

The AdaBoost algorithm according to Viola and Jones [19]

- Input: set of examples $(x_1, y_1), \ldots, (x_N, y_N)$.
- Let m be the number of negatives examples and l be the number of positive examples. Initialize weights $w_{1,n} = \frac{1}{2m}, \frac{1}{2l}$ depending on the value of y_n.
- For $t = 1, \ldots, T$:
  1) Normalize the weights $w_{t,n}$ so that $\sum_{n=1}^{N} w_{t,n} = 1$.
  2) For each feature f_j, train a weak classifier h_j.
  3) The error ϵ_j of a classifier h_j is determined with respect to the weights $w_{t,1}, \ldots, w_{t,N}$:
  $$\epsilon_j = \sum_n^N w_{t,n} \left| h_j(x_n) - y_n \right| .$$
  4) Choose the classifier h_j with the lowest error ϵ_j and set $(h_t, \epsilon_t) = (h_j, \epsilon_j)$.
  5) Update the weights $w_{t+1,n} = w_{t,n} \beta_t^{1-e_n}$, where $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ and $e_n = 0$, if example x_n is classified correctly by h_t and 1, otherwise.
- The final strong classifier is given by:
$$h(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \log \frac{1}{\beta_t} h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \log \frac{1}{\beta_t} \\ 0 & \text{otherwise.} \end{cases}$$

III. Multi-class AdaBoost

The standard AdaBoost algorithm has been designed for binary classification problems. To classify places in the environment, however, we need the ability to handle multiple classes. In this section we therefore describe two extensions of AdaBoost for multi-class problems. We first describe the AdaBoost.M2 algorithm, which has been presented by Freund and Shapire [5]. Then we will describe our approach, which uses an optimized sequence of binary classifiers.

A. AdaBoost.M2

Freund and Shapire describe a variant of the AdaBoost algorithm, which is called AdaBoost.M2 and which is able to deal with multiple classes. In AdaBoost.M2 the weak classifiers have an additional argument y which represents the class of the example x.

The key idea of AdaBoost.M2 is to reduce the weak multi-class hypotheses to binary ones and than apply a slightly modified variant of the binary AdaBoost algorithm. To achieve this, each weak classifier $h(x, y)$ is decomposed into K weak binary classifiers according to

$$h(x, y) \quad = \quad h_{j,k}(x) \qquad \text{with} \quad y = k \quad (3)$$

for $k = 1, \ldots, K$. Each weak binary classifier $h_{j,k}(x)$ is learned according to Equation (2) by taking as positive examples those for which $y = k$ and as negative all others. A detailed description of this algorithm can be found in [5, 12].

B. Sequential AdaBoost

An alternative way to construct a multi-class classifier is to arrange several binary classifiers to a decision list. Each element of such a list is one binary classifier which determines if an example 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

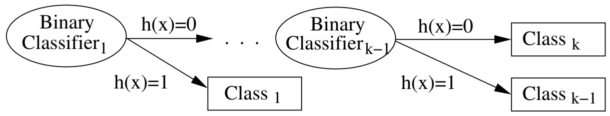

Fig. 3. A decision list classifier for k classes using binary classifiers.

element in this list. Figure 3 illustrates the structure of such a decision list classifier.

One important question in the context of a sequential classifier is the order in which the individual binary classifiers are arranged. This order can have a major influence on the overall classification performance, because the individual classifiers typically are not error-free and classify with different accuracies.

Since the first element of such a sequential classifier processes more data than subsequent elements, it is typically a good strategy to order the classifiers according to their estimated error rate. In general, the problem of finding the optimal order of binary classifiers that minimizes the classification error is NP-hard. In our application, however, we typically are confronted with a small number of classes only so that we can easily enumerate all potential permutations to determine the optimal sequence.

IV. Features from Laser Range Scans For Place Classification

In the previous section we described the key principles of the AdaBoost algorithm for boosting simple features to strong classifiers. It remains to describe the features of the range scans used in our current system. We assume that the robot is equipped with a 360° field of view range sensor. Each 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 consists of one observation z and its classification y. Thus, the set of training examples is given as

$$E \quad = \quad \{(z_i, y_i) \mid y_i \in Y = \{\text{Room}, \text{Corridor}, \ldots\}\} \quad (4)$$

where Y 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. The goal is to learn a classifier that is able to generalize from these training examples and that can later on reliably classify so far unseen places in this environment or even other environments.

As already mentioned, 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. All our features are rotational invariant to make the classification of a pose dependent only on the (x, y)-position of the robot and not of its orientation. Most of our features are standard geometrical features often used in shape analysis [6, 7, 10, 14, 15].

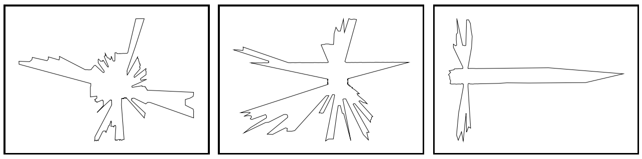

Fig. 4. Polygonal representations of the scans shown in Figure 2.

TABLE II

SET B OF SIMPLE FEATURES OF THE RAW BEAMS IN z

1) The average difference between the length of consecutive beams.
2) The standard deviation of the difference between the length of consecutive beams.
3) Same as 1), but considering different max-range values.
4) The average beam length.
5) The standard deviation of the length of the beams.
6) Number of gaps in the scan. Two consecutive beams build a gap if their difference is greater than a given threshold. Different features are used for different threshold values.
7) Number of beams lying on lines that are extracted from the range scan [16].
8) Euclidean distance between the two points corresponding to the two smallest local minima.
9) The angular distance between the beams corresponding to the local minima in feature 8).

We define a feature f as a function that takes as argument one observation and returns a real value: $f : Z \to R$, where Z is the set of all possible observations.

Two sets of simple features are calculated for each observation. The first set B is calculated using the raw beams in z. The second set P of features is calculated from a polygonal approximation $\mathbf{P}(z)$ of the area covered by z. The vertices of the closed polygon $\mathbf{P}(z)$ correspond to the coordinates of the end-points of each beam b_i of z relative to the robot.

$$\mathbf{P}(z) \quad = \quad \{(d_i \cos \alpha_i, d_i \sin \alpha_i) \mid i = 0, \ldots, M-1\} \quad (5)$$

The polygonal representations of the laser range scans depicted in Figure 2 are shown in Figure 4. Tables II and III list the individual features used by our system to learn a strong classifier for place recognition.

TABLE III

SET P OF SIMPLE FEATURES OF $\mathbf{P}(z)$

1) Area of $\mathbf{P}(z)$.
2) Perimeter of $\mathbf{P}(z)$.
3) Area of $\mathbf{P}(z)$ divided by Perimeter of $\mathbf{P}(z)$.
4) Mean distance between the centroid to the shape boundary.
5) Standard deviation of the distances between the centroid to the shape boundary.
6) 200 similarity invariant descriptors based in the Fourier transformation.
7) Major axis Ma of the ellipse that approximates $\mathbf{P}(z)$ using the first two Fourier coefficients.
8) Minor axis Mi of the ellipse that approximate $\mathbf{P}(z)$ using the first two Fourier coefficients.
9) Ma/Mi.
10) Seven invariants calculated from the central moments of $\mathbf{P}(z)$.
11) Normalized feature of compactness of $\mathbf{P}(z)$.
12) Normalized feature of eccentricity of $\mathbf{P}(z)$.
13) Form factor of $\mathbf{P}(z)$.

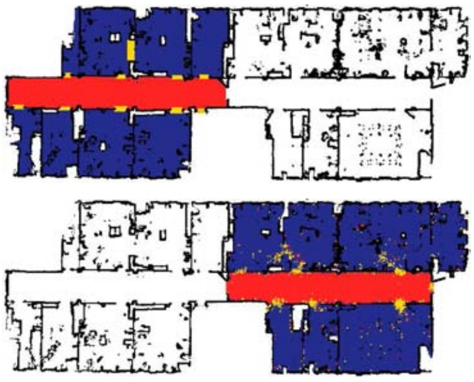

Fig. 5. The top image show the training data used to train the classifiers and the lower image show the classified test data of our sequential AdaBoost algorithm.

V. Experiments

The approach described above has been implemented and tested on a real robot as well as in simulation using the Carnegie Mellon Robot Navigation Toolkit (CARMEN) [11]. The robot used to carry out the experiments was an ActivMedia Pioneer 2-DX8 equipped with two SICK laser range finders (see left image of Figure 7). 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. We first describe the results obtained with the sequential version of AdaBoost. In the next experiment we analyze how well a mobile robot can utilize the resulting classifier. Additionally, we present an experiment illustrating that a classifier can be applied to robustly classify places in a completely new environment. Finally, we present results comparing our sequential AdaBoost with AdaBoost.M2.

One important parameter of the AdaBoost as well as the AdaBoost.M2 algorithm is the number of weak classifiers T used to form the final strong classifier. All in all we formulated 302 simple features, each of them with one free parameter, which is determined in the learning phase according to Eq. (2). AdaBoost even uses features multiple times with different parameters. Thus, much more than these 302 simple features are available to form the strong classifier. We performed several experiments with different numbers of weak classifiers and analyzed the classification error. Throughout our experiments, we found that 100 weak classifiers provide the best trade-off between the error rate of the classifier and the computational cost of the algorithm. Therefore we used this value in all the experiments presented in this paper.

A. Results with Sequential AdaBoost

The first experiment was performed using data from our office environment in building 79 at the University of Freiburg. This environment contains three different types of places, namely rooms, doorways, and a corridor. In this experiment we used the sequential classifier shown in Figure 3. For the sake of clarity we give a result obtained by separating the environment into two parts. The left half of the environment contains the training examples (see Figure 5, top image), and the right half of the environment was then used as a test set. Note that we obtained similar success rates as described below in further experiments with alternative training and test sets. According to Table IV the optimal decision list for this classification problem is room-doorway. This decision list correctly classifies 93.94% of all test examples. The classification results are also depicted as colored/grey-shaded areas in the lower image of Figure 5. This illustrates, that our approach is well-suited to classify places according to a single laser range scan.

TABLE IV

Percentage of correctly classified examples for the 6 configurations of a sequential multi-class classifier.

Classifier Sequence	Correct Classifications %
room-doorway	93.94
room-corridor	93.31
corridor-room	93.16
doorway-corridor	80.68
doorway-room	80.49
corridor-doorway	80.10

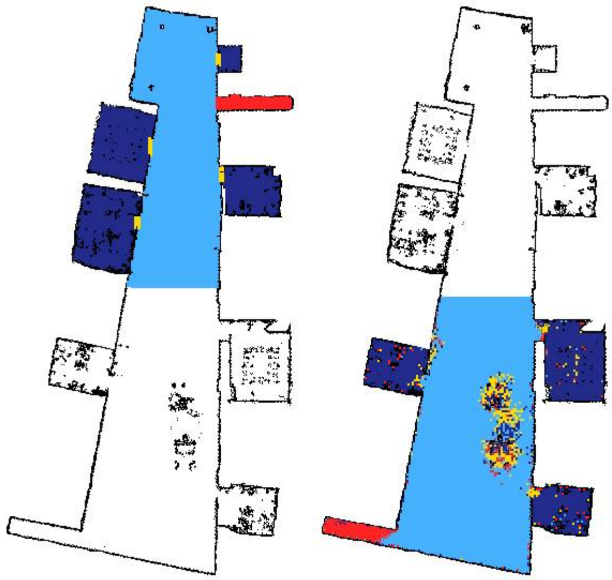

Fig. 6. The left image show the training data used to train the classifiers and the right image the classification results of our sequential AdaBoost classifier.

Table IV also contains the classification results of the other five potential sequential classifiers. As can be seen from this tables, the worst configurations are those in which the doorway classifier is in the first place. Also the corridor-doorway classifier does not perform well. The best configurations are corridor-room, room-doorway, and room-corridor.

Additionally, we performed an experiment using a map containing four different classes, namely rooms, corridors, doorways, and hallways. The training set and the resulting classifications are shown in Figure 6. The optimal decision list is corridor-hallway-doorway with a success rate of 89.52%.

TABLE V

ERROR IN THE TRAINING DATA FOR THE INDIVIDUAL BINARY CLASSIFIERS LEARNED FROM THE MAP DEPICTED IN FIGURE 6.

Binary Classifier	Training error %
corridor	0.7
hallway	0.7
room	1.4
doorway	1.5

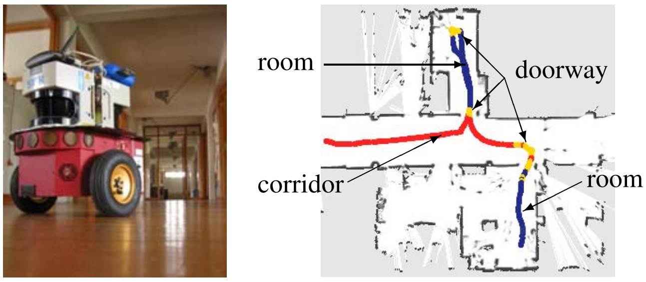

Fig. 7. Classification results are obtained with a mobile robot, shown in the left image, moving through our office environment. Colors/grey levels in the right image indicate the classification of the corresponding places on the trajectory.

Table V contains the error rates of the individual binary classifiers on the training data. The error-rates differ between .7% and 1.5%. The binary doorway-classifier yields the highest error. We believe that this due to several reasons. First, a doorway typically is a very small region so that only a few training examples are available. Furthermore, if a robot stands in a doorway the scan typically covers nearby rooms or corridors which make it hard to distinguish the doorway from such places.

B. Place Recognition with a Moving Robot

In this experiment we use the best classifier for our office building (see Table IV) to classify the current pose of a mobile robot. We installed our Pioneer2-DX8 robot in our office building and steered it through the corridor, different rooms, and several doorways. While the robot was moving we logged its trajectory and the classifications obtained for the different range scans. The result is depicted in Figure 7. Again, the different colors/grey levels of the points on the trajectory indicate the classification of the corresponding scan. As can be seen, the robot reliably identifies the type of the place. Only a few places show wrong classifications. These failures are mostly caused by clutter in the environment which make the sequential room-doorway classifier believe that the current place is a doorway.

C. Transferring the Classifiers to New Environments

The next 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 AdaBoost on the map shown in Figure 1. In this environment our approach was able to correctly classify 92.1% of all places. The resulting classifier was then evaluated on scans simulated given the map of the Intel Research Lab in Seattle. For these scans the classification rate decreased to 82.23% (see Figure 8). This

Fig. 8. Classification results obtained by applying the classifier learned for the environment depicted in Figure 1 to the map of the Intel Research Lab in Seattle. The fact that 82.23% of all places could be correctly classified illustrates that resulting classifiers can be applied to so far unknown environments.

TABLE VI

CLASSIFICATION RESULTS FOR DIFFERENT INDOOR MAPS AND DIFFERENT CLASSIFIERS.

Map depicted in	Sequential Classifier %	AdaBoost.M2 %
Figure 1	92.10	91.83
Figure 5	93.94	83.89
Figure 6	89.52	82.33

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 82.23% is quite high for this environment, since even humans typically do not consistently/correctly classify the places in this environment.

D. Comparison of the Sequential AdaBoost with AdaBoost.M2

Our current system described above uses a sequence of strong binary classifiers arranged in a decision list. In this experiment we compare this approach to AdaBoost.M2, which is a multi-class variant of AdaBoost.

In all experiments our sequential classifiers performed better than AdaBoost.M2. To see the difference, Figures 9 and 10 show typical results obtained with our sequential approach (left image) and AdaBoost.M2 (right image). Table VI shows 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.

Note that we also considered organizing the binary classifiers in a decision tree rather than restricting them to a decision list. In various experiments, however, we found that the tree-structure does not yield improvements over the sequential decision lists, at least in the domain given here.

E. Important Weak Features

We furthermore analyzed the importance of the individual weak features in the final strong classifier. Table VII lists the

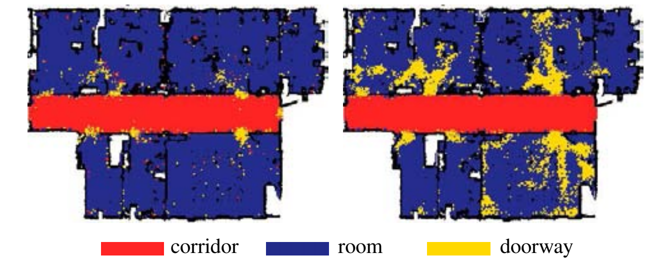

Fig. 9. The left image show the classified test data of our sequential AdaBoost algorithm in building 79. The right image depicts the result obtained with AdaBoost.M2. As can be seen, the error of AdaBoost.M2 is much higher compared to our approach.

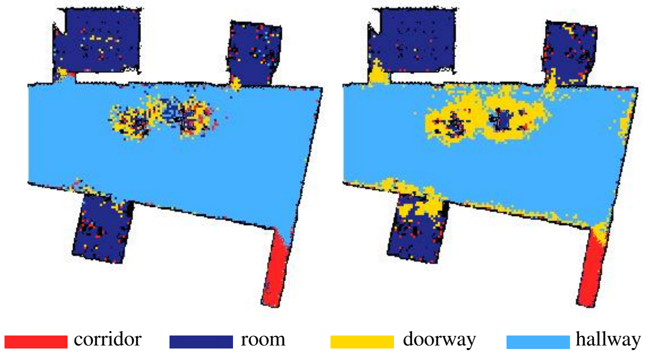

Fig. 10. Classification results of our sequential AdaBoost algorithm (left image) and AdaBoost.M2 (right image). Again the sequential approach outperforms AdaBoost.M2.

seven best features for each binary classifier with the leftmost feature the most important. In this table an entry B.i represents the i-th feature for raw beams in z (Table II), whereas an entry P.j represents the j-th feature of the polygon $\mathbf{P}(z)$ (Table III). Note that often identical features occur. These features differ in their threshold values and their weight, which is assigned by AdaBoost. As the table shows, several features like the average difference between consecutive beams (feature B.1) appears to be quite important. Furthermore, the number of gaps (feature B.6), which represents how cluttered the environment is, appears quite often. Whereas feature P.1, which corresponds to the area of the polygon, is most important for the detection of hallways, the feature B.8, which measures the distance between the smallest local minima in the range scan, has the highest weight in the classifier for doorways.

VI. Conclusion

In this paper we presented a novel approach to classify different places in the environment into semantic classes, like rooms, hallways, corridors, and doorways. Our technique uses simple geometric features extracted from a single laser range scans and applies the AdaBoost algorithm to form a strong classifier. To distinguish between more than two classes we use a sequence of binary classifiers arranged in a decision list. Experiments carried out on a real robot as well as in simulation illustrate that our technique is well-suited to classify places in different environments even without training the classifier for each environment. Furthermore we compared our sequential AdaBoost to AdaBoost.M2, a multi-class variant of the AdaBoost algorithm. In our experiments the sequential classifier always outperformed AdaBoost.M2.

TABLE VII

The best five feature for each binary classifier.

binary classifier	seven best features
corridor	B.6, B.1, P.7, P.6, P.6, B.1, B.1
room	P.2, B.1, P.4, P.6, P.7, B.6, P.5
doorway	B.8, B.1, B.9, B.4, B.2, P.6, B.1,
hallway	P.1, B.1, B.8, P.1, P.12, P.6, B.1

Acknowledgment

This work has partly been supported by the German Science Foundation (DFG) under contract number SFB/TR-8 (A3) and by the EC under contract number FP6-004250-CoSy. Furthermore, we would like to thank Steffen Gutmann for providing us the map shown in Figure 1.

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] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory (Eurocolt)*, 1995.

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

[7] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Addison-Wesley Publishing Inc., 1992.

[8] 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.

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

[10] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8), 1998.

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

[12] O. Martínez Mozos. Supervised learning of places from range data using adaboost. Master's thesis, University of Freiburg, Freiburg, Germany, 2004.

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

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

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

[16] D. Sack and W. Burgard. A comparison of methods for line extraction from range data. In *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2004.

[17] 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.

[18] 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.

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