Efficient Exploration of Unknown Indoor Environments using a Team of Mobile Robots

Cyrill Stachniss[*] Óscar Martínez Mozos Wolfram Burgard

University of Freiburg
Department of Computer Science
Georges-Koehler-Allee 79
D-79110 Freiburg, Germany

Phone: +49 761 203-8024, Fax: -8007 (fax)

{stachnis, omartine, burgard}@informatik.uni-freiburg.de

Keywords: multi-robot exploration, coordination, semantic place information
Mathematics Subject Classification: 68T40

Abstract

Whenever multiple robots have to solve a common task, they need to coordinate their actions to carry out the task efficiently and to avoid interferences between individual robots. This is especially the case when considering the problem of exploring an unknown environment with a team of mobile robots. To achieve efficient terrain coverage with the sensors of the robots, one first needs to identify unknown areas in the environment. Second, one has to assign target locations to the individual robots so that they gather new and relevant information about the environment with their sensors. This assignment should lead to a distribution of the robots over the environment in a way that they avoid redundant work and do not interfere with each other by, for example, blocking their paths. In this paper, we address the problem of efficiently coordinating a large team of mobile robots. To better distribute the robots over the environment and to avoid redundant work, we take into account the type of place a potential target is located in (e.g., a corridor or a room). This knowledge allows us to improve the distribution of robots over the environment compared to approaches lacking this capability. To autonomously determine the type of a place, we apply a classifier learned using the AdaBoost algorithm. The resulting classifier takes laser range data as input and is able to classify the current location with high accuracy. We additionally use a hidden Markov model to consider the spatial dependencies between nearby locations. Our approach to incorporate the information about the type of places in the assignment process has been implemented and tested in different environments. The experiments illustrate that our system effectively distributes the robots over the environment and allows them to accomplish their mission faster compared to approaches that ignore the place labels.

[*] Contact author

1 Introduction

The use of multiple robots is often suggested to have advantages over single robot systems [5, 10]. For example, cooperating robots have the potential to accomplish a task faster than a single robot [19]. A further advantage of robot teams arises from merging overlapping sensor information, which can help to compensate for sensor uncertainty. As a result, the map can be expected to be more accurate. Multiple robots have also been shown to localize themselves more efficiently, especially when they have different sensor capabilities [12, 35].

However, when robots operate in teams there is the risk of interference between them [38, 16]. For example, if the robots have the same type of active sensors such as ultrasound sensors, the overall performance can be reduced due to cross-talk. The more robots that are used, the more time each robot may spend on detours in order to avoid collisions with other members of the team. Efficient exploration techniques that seek to minimize the overall time to complete the task should consider strategies to distribute the robots over the environment and to reduce the number of redundantly explored areas. An illustrating example depicting two real robots exploring an indoor environment with our coordination method is shown in Figure 1.

Several exploration techniques dealing with the problem of appropriate collaboration between robots were presented in the past [4, 21, 34, 44, 47]. Most approaches to multi-robot exploration proceed in the following way. First, a set of potential target locations or target areas is determined. Such target locations are often located close to unknown areas in the environment to allow the robot to observe the unknown space. Secondly, the target locations are assigned to the individual members of the team. The robots then approach those target locations and include their observations obtained along the paths into a map. This process is repeated until the environment has completely been explored. In the context of multi-robot exploration, it is important to achieve a collaboration behavior so that the robots avoid traveling unnecessary long distances, avoid doing redundant work, and avoid interference with other team-mates.

Indoor environments constructed by humans often contain certain structures like corridors with adjacent rooms or offices. However, it is mainly unexplored how robots can utilize such background information to more efficiently solve the exploration task. One of our observations is that the more unexplored target locations are known when assigning targets to robots, the faster the team can explore the environment. This is due to the fact that the robots can be better distributed over the environment. In this way, the amount of redundant work is reduced and interferences occur less likely. It therefore makes sense to focus on areas first which are likely to provide a large number of new target locations in order to obtain a better assignment of targets to robots.

The contribution of this paper is a technique for coordinating a group of robots that enables them to efficiently explore their environment. The goal is to complete the task as fast as possible. Our technique assigns a utility to each target location and follows a decision-theoretic approach to explicitly coordinate the robots. We estimate and utilize semantic information during the collaborative multi-robot exploration. In our approach, the robots get a higher reward for exploring corridors since they typically provide more branchings to unexplored areas like adjacent rooms compared to rooms. This is especially useful in case of large robot teams, because if more target locations are available the robots can be better distributed over the environment. As a result, the

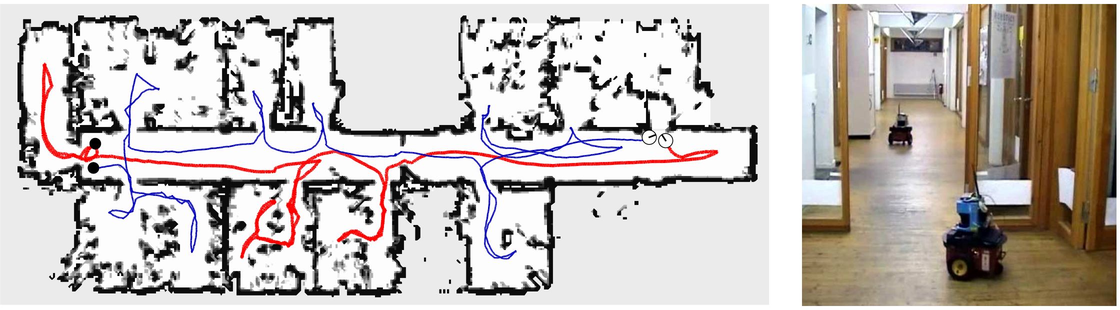


Figure 1: Left: Exploration task with two real robots. The trajectories of the individual robots are plotted in red and blue. Right: Photograph of the two robots during exploration.

exploration mission can be carried out faster. Our technique has been implemented on teams of robots and has been proven to work effectively in different environments. As the experiments demonstrate, our technique significantly reduces the time required to completely cover an unknown environment with a team of robots compared to an approach which lacks the possibility to estimate and integrate semantic information about places in the environment.

The paper is organized as follows. We first explain our technique to estimate semantic labels of places in the environment using the laser range sensor of a robot. In Section 3, we present a hidden Markov model (HMM)-based extension to the labeling approach which improves the classification in the context of exploration. We then propose our coordination technique and describe how to utilize the place information in Section 4. We then present in Section 5 our experimental results and finally discuss related work.

2 Semantic Place Labeling

This section explains how semantic place labels can be obtained with mobile robots based on laser range observations. The goal is to learn a classifier that is able to distinguish corridors from other kinds of indoor structures. To obtain such a classifier, we apply the AdaBoost algorithm introduced by Freund and Schapire [14].

Our classification approach for semantic labeling of places relies on a large set of single-valued features that are calculated for each laser range scan. A laser range scan is a high-dimensional vector containing typically between 180 and 360 individual proximity measurements. Each measurement results from a laser beam that has been emitted from the same location in different orientations.

Working on such vectors directly would result in a high-dimensional classification problem. We therefore extract simple single-valued features from each laser range scan and apply the classification approach based on these features. In general, it is difficult to know *a priori* which features are more discriminative or even if some of them are discriminative at all. One approach to solve this problem is to apply an algorithm that selects an appropriate subset of features which is then used in the classification task. In our work, we use the AdaBoost algorithm to learn a strong classifier. AdaBoost is a boosting algorithm which can also be interpreted as a heuristic

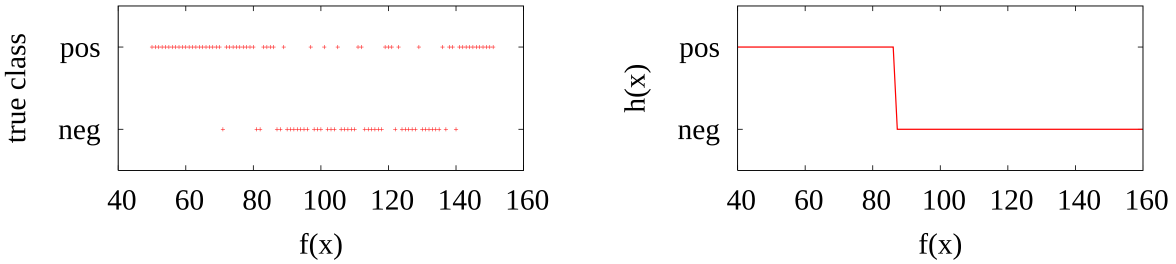


Figure 2: This figure illustrates how the optimal value of θ_j is found. In the left image, the x-value of each data point represents the feature value of a training example and the y-value its true class. By iterating through this list of data points, one can determine the optimal value θ_j for the given training examples. The right image depicts the weak classifier h_j.

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.

The key idea of AdaBoost is to form a strong binary classifier given a set of weak classifiers. The weak classifiers h_j only need to be better than random guessing. Similar to the work of Viola and Jones [43], we construct our weak classifier based on simple, single-value features $f_j \in \mathbb{R}$

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

This weak classifier returns 1 if the training example x is supposed to be a positive example and 0 otherwise. θ_j is a threshold value and p_j is either -1 or $+1$ and thus represents the direction of the inequality. The AdaBoost algorithm determines during the training process for each weak classifier h_j the optimal parameter tuple (θ_j, p_j), such that the number of misclassified training examples is minimized. To achieve this, it considers all possible combinations of p_j and θ_j, whose number is limited since only an finite number N of training examples is given. A training example is defined by the tuple (x_n, y_n) where x_n is the example and $y_n \in \{0, 1\}$ the class x_n belongs to. Using the training examples, (θ_j, p_j) is determined by

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

Figure 2 illustrates the process to compute the optimal value of θ_j. First, one computes for each training example (x_n, y_n) the feature value $f_j(x_n)$ and adds it to a list which is sorted according to that value. Second, one iterates through this list and computes the error of the weak classifier using a value for θ_j that is between the feature value of the current and the next element. The value which provides the highest classification rate is the optimal value for θ_j given the training set.

We compute two sets of simple features for each observation. The first set is calculated using the raw beams $z_{t,i}$, $i = 1, \ldots, M$ in the full range scan z_t. The second set of features is

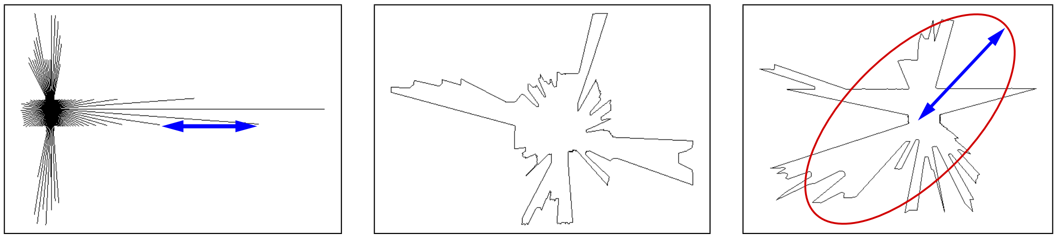


Figure 3: Examples for features generated from laser data, namely the average distance between two consecutive beams, the perimeter of the area covered by a scan, and the length of the major axis of the ellipse that approximates the polygon described by the scan.

calculated from a polygonal approximation $\mathcal{P}(z_t)$ of the area covered by z_t. The vertices of the closed polygon $\mathcal{P}(z_t)$ correspond to the coordinates of the end-points of each beam relative to the robot.

$$\mathcal{P}(z_t) \quad = \quad \{(z_{t,k} \cdot \cos\phi_k, z_{t,k} \cdot \sin\phi_k) \mid k = 1, \ldots, M\}, \tag{3}$$

where ϕ_k is the angle of the k-th beam z_t, k of the observation z_t.

Examples for features extracted from laser range data are depicted in Figure 3. Such features are, for example, the average distance between consecutive beams, the area covered by a range scan, or the perimeter of that area. All our features are rotationally invariant to make the classification of a position dependent only on the (x, y)-position of the robot and not on its orientation. Most of the features are standard geometrical features used in shape analysis [17]. Table 1 and 2 provide a full list of features used by our system to learn classifier for place recognition.

The input to the AdaBoost algorithm is a set of labeled, positive and negative training examples $\{x_n, y_n\}$. In our case, this is a set of laser range observations recorded in a corridor and a second set taken outside corridors. In a series of T rounds, the algorithm repeatedly selects the weak classifier h_j with the lowest error for the training dataset. To do so, AdaBoost uses importance weights that are associated to each example. The importance weight w_n for each example n is updated in each round. The algorithm modifies the importance weights by increasing the weights of the training examples that are incorrectly classified so far.

The optimal parameters (θ_j, p_j) for each weak classifier h_j are also computed using the weighed examples. As a result, a single feature can generate several weak classifiers with different parameters in the individual rounds of the AdaBoost algorithm.

The final strong classifier $\mathcal{H}$ is a weighted majority vote of the best T weak classifiers

$$\mathcal{H}(x) \quad = \quad \begin{cases} 1 & \text{if } \sum_{t=1}^{T} h_t(x) \cdot \alpha_t \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where the value of α_t is computed according to the weighted error rates of the individual weak classifiers. A precise description is given in Algorithm 1. In our system, the resulting strong classifier takes as input a single 360 degree laser range scan recorded by a robot and is able to determine whether the position from which the scan was taken belongs to the class *corridor* or not.

Table 1: Simple features based on the individual beams of a laser range observation 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 [37].
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).

Table 2: Features computed based on the polygon $\mathcal{P}(z)$

1. Area of $\mathcal{P}(z)$.
2. Perimeter of $\mathcal{P}(z)$.
3. Area of $\mathcal{P}(z)$ divided by Perimeter of $\mathcal{P}(z)$.
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 on the Fourier transformation.
7. Major axis of the ellipse that approximates $\mathcal{P}(z)$ using the first two Fourier coefficients.
8. Minor axis of the ellipse that approximate $\mathcal{P}(z)$ using the first two Fourier coefficients.
9. The ratio of the major and minor.
10. Seven invariants calculated from the central moments of $\mathcal{P}(z)$.
11. Normalized feature of compactness of $\mathcal{P}(z)$.
12. Normalized feature of eccentricity of $\mathcal{P}(z)$.
13. Form factor of $\mathcal{P}(z)$.

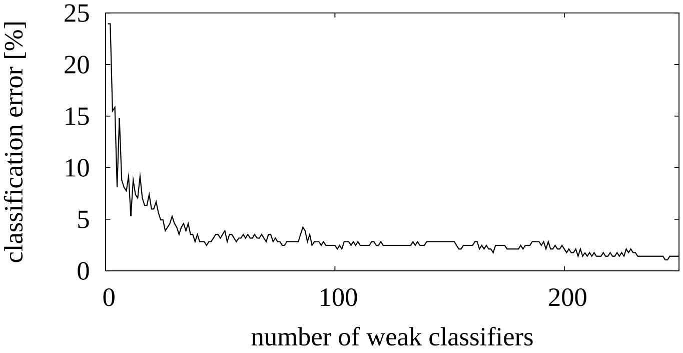


Figure 4: Classification error for a given numbers of weak classifiers (T).

A free parameter is the number T of features used in the classifier. The value of T that allows for accurate classification results can be determined by plotting the number of T versus the classification error on a validation dataset. An example for such a plot is shown in Figure 4. In our current system, we set $T = 100$ since with 100 weak classifiers we are able to accurately determine the semantic class given real sensor observations.

Note that also the angular resolution of the used sensor can have an influence on the performance of the classifier. For different angular resolutions, different feature values can be obtained, compare Eq. (3). For resolutions below 1 degree, we obtained more or less identical results. Between 1 degree and 3 degree, a slight performance loss between 1% and 2% can be observed. For larger angular resolutions, however, the performance breaks down significantly. Thus, a typical laser range finder such as SICK LMS/PLS sensor with a resolution between 0.25 degree and 1 degree is well-suited for obtaining strong classification results with our method.

3 Estimating the Label of a Goal Location

The idea described in the previous section is well-suited to determine the semantic class for the current position of the robot given a laser range scan. Even if the place to classify is not the current pose of the robot, one can simulate a laser range observation in the map of the environment and apply the classifier to the simulated scan. This works well for poses for which surroundings are completely known.

In the context of exploration, however, we are interested in classifying potential targets of the robot. Typically, target locations are located at the frontiers between known and unknown areas. This means that large neighboring areas have not been observed so far which makes it impossible to generate an appropriate observation taken from that location. As we will demonstrate in the experiments, classifying a place at a frontier with the approach presented in the previous section leads to high false classification rates of around 20%. In the following, we therefore introduce a HMM-based technique that takes into account spacial dependencies between nearby locations to obtain a lower error rate for places located at frontiers.

Algorithm 1 The AdaBoost algorithm

```
Require: Input: set of examples $(x_1, y_1), \ldots, (x_N, y_N)$.
$k$ = number of negatives examples
$l$ = number of positive examples.
for $n = 1, \ldots, N$ do
    if $y_n = 1$ then
        $w_{1,n} = \frac{1}{l}$
    else
        $w_{1,n} = \frac{1}{k}$
    end if
end for
for $t = 1, \ldots, T$ do
    Normalize the weights $w_{t,n}$ so that $\sum_{n=1}^{N} w_{t,n} = 1$.
    for all features $f_j$ do
        Train a weak classifier $h'_j$ for the feature $f_j$.
        Compute the error $\epsilon'_j$ of a classifier $h'_j$according to
            $$\epsilon'_j = \sum_{n=1}^{N} w_{t,n} \cdot \left| h'_j(x_n) - y_n \right| .$$
    end for
    Determine the weak classifier with the lowest error:
            $$(h_t, \epsilon_t) = \operatorname*{argmin}_{(h'_j, \epsilon'_j)} \epsilon'_j$$
    $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$
    for $n = 1, \ldots, N$ do
        $w_{t+1,n} = w_{t,n} \cdot \beta_t^{1 - |h_t(x_n) - y_n|}$
    end for
    $\alpha_t = \log \frac{1}{\beta_t}$
end for
The final strong classifier is given by
            $$\mathcal{H}(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} h_t(x) \cdot \alpha_t \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise,} \end{cases}$$
return $\mathcal{H}$
```

In our approach, we generate a potential target location for each group of frontier cells lying on the same frontier. This process is repeated for each frontier. As an example, the left image of Figure 5 depicts a potential target location extracted for the right-most frontier (the targets for the other two frontiers are not shown in that image).

Due to the structure of man-made environments, the semantic class does not change randomly between nearby poses. Therefore, it makes sense to consider a smoothing or filtering between places located close together. To do so, we generate a short virtual trajectory to the desired goal location. We then simulate a laser range observation within the partially known map along the virtual trajectory. Whenever the ray-casting operation used to simulate a beam reaches an unknown cell in the grid map, the virtual sensor reports a maximum-range reading. We then apply a hidden Markov model (HMM) and maintain a posterior $Bel(L_x)$ about the type L_x of the place x the virtual sensor is currently at

$$Bel(L_x) \quad = \quad \eta \cdot p(o_x \mid L_x) \cdot \sum_{L_{x'}} p(L_x \mid L_{x'}) \cdot Bel(L_{x'}). \tag{5}$$

In this equation, o_x is the result of the AdaBoost-based classifier for the observation taken from place x and η is a normalizing constant ensuring that the left-hand side sums up to one over all semantic labels.

To implement this HMM, three components need to be known. First, we need to specify the observation model $p(o_x \mid L_x)$ which is the likelihood that the classification output is o_x given the actual class is L_x. In our current system, the observation model has been learned based on 5,000 observations, simulated at randomly chosen locations in different environments combined with the corresponding ground truth labeling. Thus, the likelihoods for the individual situations (corridor and non-corridor) can be directly obtained by counting.

Second, we need to specify the transition model $p(L_x \mid L_{x'})$ which defines the probability of a movement from class $L_{x'}$ to class L_x. To determine this model, we evaluated trajectories of robots obtained during exploration in different environments. Again, we can directly compute $p(L_x \mid L_{x'})$ by counting the transitions between corridor and non-corridor locations on that trajectories given the (manually generated) ground truth labeling for the training environments.

Furthermore, we need to specify how the belief $Bel(L_{start})$ is initialized. In our current system, we choose a uniform distribution, which means that all classes (here *corridor* and *non-corridor*) have the same likelihood.

Finally, we have to describe how the virtual trajectory is generated. The endpoint of the trajectory is the frontier cell to be classified. Since locations that have fewer unknown grid cells in their surroundings can typically be classified with a higher success rate, the other positions on those trajectory should be as far away from the unknown locations as possible. Therefore, we apply the Euclidian distance transformation [31] with respect to unknown and occupied cells in the local area of the frontier. We then select the pose in the free space within that local area with the highest distance to unknown areas. An A* planner is used to generate the virtual trajectory to the target location. An illustrating example is depicted in Figure 5.

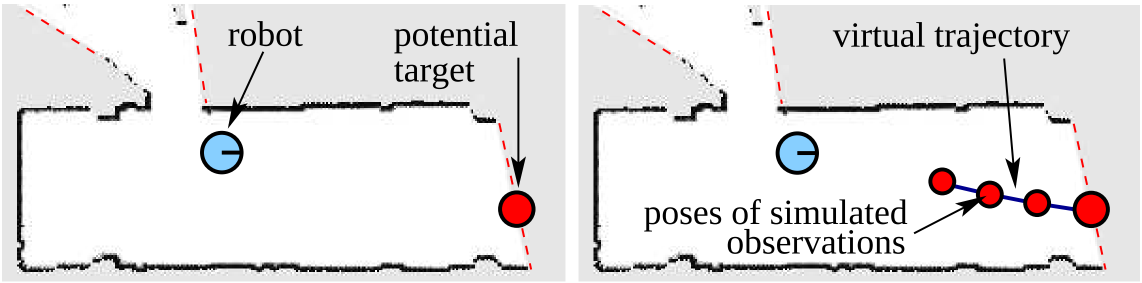


Figure 5: This figure illustrates the generation of the virtual trajectory used for the HMM filtering. The left image depicts the current location of the robot, the frontiers (dashed lines), and a potential target location to be evaluated. To do so, the robot generates a virtual trajectory as shown in the right image and simulates observations at several positions located on the trajectory. These sequence of observations is used as the input of the HMM to obtain a more robust classification result.

4 Efficient Multi-Robot Exploration

The goal of collaborative multi-robot tasks is to share the load between the members of a team in order to accomplish the task faster. As we will discuss in the related work section, different approaches exist that assign target locations to robots using job-shop-scheduling techniques, bidding algorithms, or decision theoretic approaches. In the approach described here, we modify our previous work [4] so that the place information is integrated in the coordination process.

To reduce the overall time needed to explore an unknown environment, the robots need to select appropriate target locations. Target locations should fulfill three constraints so that guiding a robot to such locations provides novel and relevant information about the environment. First, the robot should be able to gather information about so far unknown areas of the environment. Second, the robots should avoid redundant work. Sending multiple robots to the same target location, for example, is often a waste of resources since one robot can solve the task nearly as efficiently as multiple robots. Third, the time needed by a robot to reach a desired target location should be small. Our approach addresses these three issues when assigning target locations to the robots. The first aspect in considered when identifying potential target location for the robots. We place potential targets locations on the frontiers between known and unknown areas in the model of the environment. Given the grid map representations, frontiers cells [44] can be extracted easily by inspecting the neighboring cells in the grid.

Given a set of potential target locations (frontiers), we proceed as follows. For each robot i in a team, we compute the cost V_t^i to each target location t based on the distance to be traveled to reach that location. To avoid that several robots focus on the same frontier, we introduce a utility U_t for each target location t which is discounted after being assigned to one robot. In this way, the robots get distributed over the environment and do not focus on the same local area. Additionally, target locations which can potentially be observed by other robots already assigned to nearby targets are discounted.

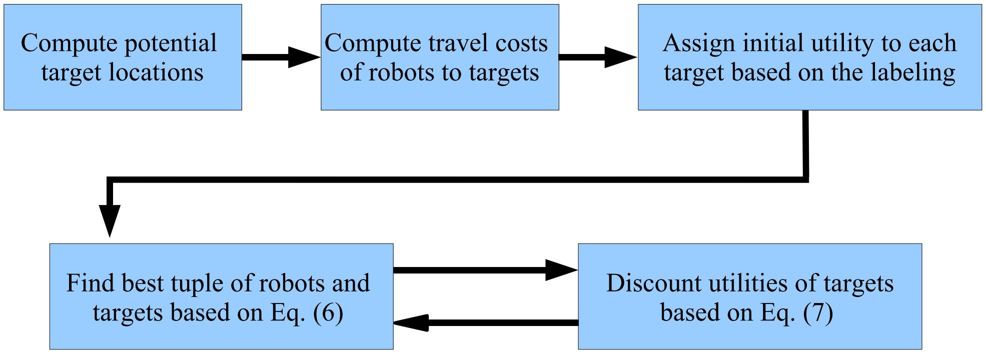

Compute potential target locations
Compute travel costs of robots to targets
Assign initial utility to each target based on the labeling
Find best tuple of robots and targets based on Eq. (6)
Discount utilities of targets based on Eq. (7)

Figure 6: Diagram illustrating the procedure to coordinate the team of robots

The target assignment procedure is an iterative process, graphically illustrated in Figure 6. In each iteration, the tuple (i, t), where i is a robot and t a frontier cell, with the best overall evaluation given the utility U_t and the travel cost V_t^i is chosen

$$(i, t) \quad = \quad \underset{(i',t')}{\operatorname{argmax}} \left(U_{t'} - V_{t'}^{i'} \right). \tag{6}$$

One then recomputes the utilities of all frontier cells by discounting all frontiers that are likely to be observed by robot i when approaching target t as

$$U_{t'} \quad \leftarrow \quad U_{t'} - P_{vis}(t, t'), \tag{7}$$

where $P_{vis}(t, t')$ describes the probability that the target t' can be observed by robot i moving to t. In our approach, this probability density is approximated by a linear function.

These two step process is repeated for the remaining robots. The approach works in a centralized fashion but can also deal with limited communication. Typically, one robot calculates the assignments. In case the whole team splits up into several teams due to the restricted communication range, one member of each sub-team becomes the coordinator (for example based on a priority scheme) and executes the target assignment procedure. See [4] for aspects on limited communication.

The knowledge about the semantic labels is integrated into the utility function. All places which are supposed to provide several branchings to adjacent places are initialized with a high utility. In our current implementation, all corridor locations get a γ times higher initial utility (U_{init}) compared to all other potential target locations. In this way, the robots prefer targets in corridors and eventually make slight detours to explore them first. To determine the actual value of γ, we performed exploration runs in different environments with varying γ. We figured out that we obtained the best results using a γ-value of around 5. Algorithm 2 depicts the resulting coordination technique used in our current system.

Our approach distributes the robots in a highly efficient manner over the environment and reduces the amount of redundant work by taking into account visibility constraints between targets and their semantic labels. The labels are used to focus the exploration on unexplored corridors,

Algorithm 2 Target Assignment Algorithm Using Semantic Place Labels.

```
Determine the set of frontier cells.
Compute for each robot $i$ the cost $V_t^i$ for reaching each frontier cell $t$.
Estimate for each frontier cell $t$ the semantic labeling $L_t$ (according to Section 3).
Set the utility $U_t$ of all frontier cells $t$ to $U_{init}(L_t, n)$ according to their semantic labeling $L_t$
and the size $n$ of the team (see text below).
while there is one robot left without a target point do
    Determine a robot $i$ and a frontier cell $t$ which satisfy:
    $(i, t) = \mathrm{argmax}_{(i',t')} \left(U_{t'} - V_{t'}^{i'}\right)$.
    Reduce the utility of each target point $t'$ in the visibility area according to
    $U_{t'} \leftarrow U_{t'} - P_{vis}(t, t')$.
end while
```

because they typically provide more branchings to adjacent rooms than other places. The high number of branchings results in a higher number of potential target locations that are available in the assignment process. This typically leads to a more balanced distribution of robots over the environment. As we will demonstrate in the experiments, the integration of such semantic labels helps to reduce the overall exploration time of multi-robot exploration approaches for large robot teams.

Please note that for very small teams of robots we do not achieve a reduction of the exploration time using our technique. This fact can be explained by considering the single-robot exploration scenario. In this case, it makes no sense to focus on exploring the corridors first, since the robot has to cover the overall environment with its sensor. Moving through the corridors first will in general lead to an increased trajectory length and in this way will increase the overall exploration time. We observed this effect for robot teams smaller than five robots.

To prevent a loss of performance compared to approaches which do not consider semantic place information for small robot teams, we trigger the influence of the semantic place information depending on the size of the team. We linearly decrease the influence γ for teams smaller than 10 robots. The linear interpolation of the influence of the semantic labels is encoded in the utility function $U_{init}(L_t, n)$ in Algorithm 2. Here, n denotes the number of robots.

5 Experiments

This section is designed to evaluate the improvements of our multi-robot coordination technique which makes use of semantic place information. Due to the high number of robots in the team, we evaluated our collaboration technique only in simulation experiments.

5.1 Performance Improvement using Semantic Place Information

The first experiment has been carried out in the map of the Fort Sam Huston hospital which is depicted in the top image of Figure 7. This environment contains a long horizontal corridor,

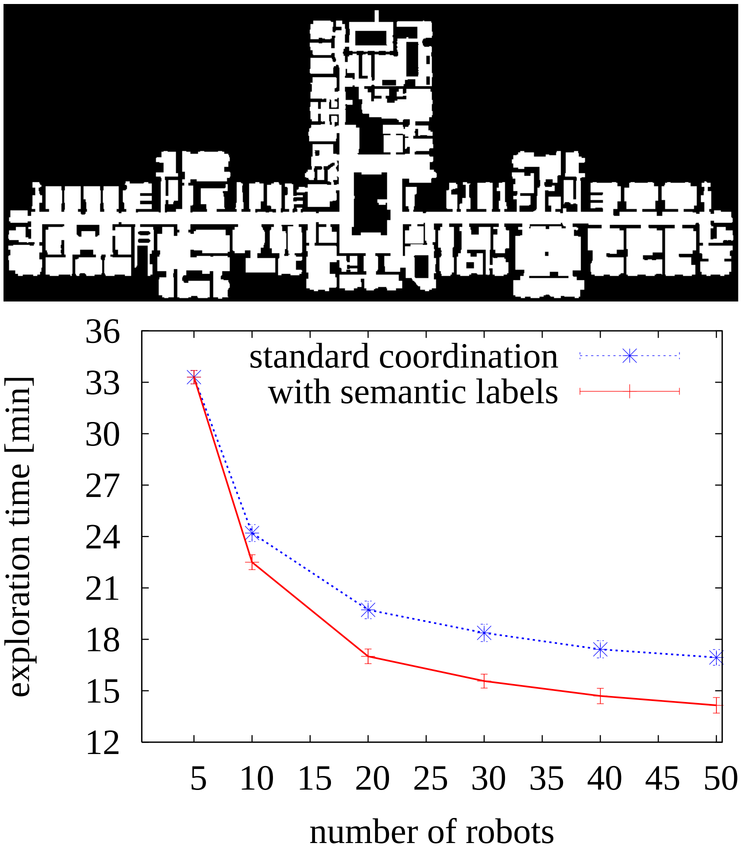


Figure 7: Coordination results obtained in the Fort Sam Huston hospital map employing the coordination strategy with and without the use of semantic place labels.

vertical corridors, and several rooms adjacent to the corridors.

We varied the size of the robot team from 5 to 50 robots and applied the coordination technique with and without taking into account semantic information about places. The lower plot in Figure 7 depicts the result of the exploration experiment by plotting the exploration time versus the number of robots. The error bars in that plot indicate the 95% confidence level. As can be seen, our technique significantly outperforms the collaboration scheme that does not consider the place information. This significant reduction of exploration time is due to the fact that the robots focus on exploring the corridors first. As a result, a big number of frontiers emerges due to typically numerous adjacent rooms. Especially in the context of large teams, this results in a better distribution of robots over the environments and thus speeds up the overall exploration process. This effect can be observed in Figure 8. The graphs plot the number of available target locations over time during an exploration task carried out using the Fort Sam Houston map. During the assignment process, most of the time the number of available target locations is higher compared to our previous approach. This leads to a better assignment of target locations to robots and as a result the amount of redundant work is reduced.

Furthermore, we observed a reduction of interferences between robots when they plan their

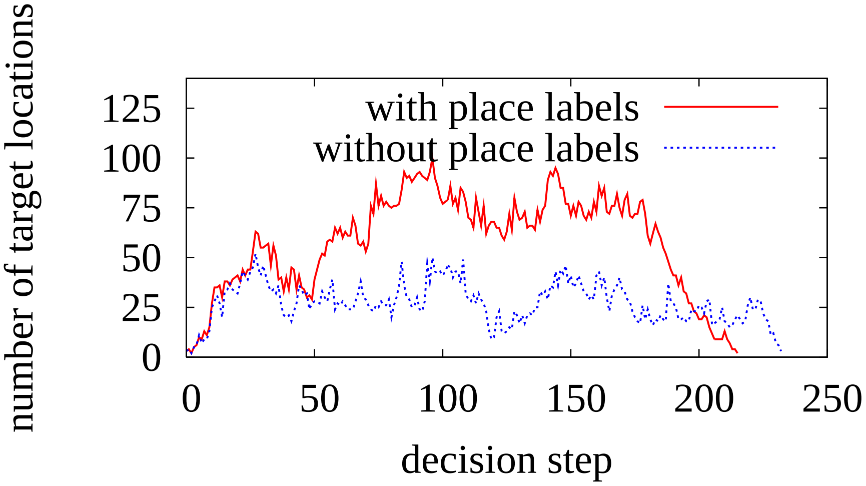


Figure 8: The number of potential target locations at the different decision steps during exploration.

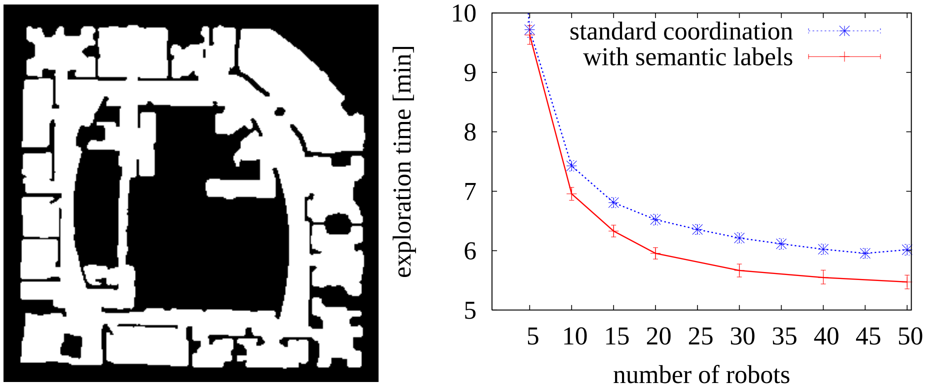


Figure 9: Results obtained in the Intel Research Lab.

paths through the environment. The interferences lead to a lower speed of the robots, since they often block their paths. Therefore, reducing the number of interferences allows the robots to accomplish their task faster. In our experiments, we observed a reduction of robot-robot interferences of up to 20%.

We performed similar experiments in different environments, for example in the Intel Research Lab depicted in the left image of Figure 9. The result is comparable to the previous experiment and again the knowledge about the semantic categories of places allows the robots to complete the exploration task more efficiently. The actual evolution of the exploration time in this experiment is depicted in the right plot of Figure 9. The same holds for experiments carried out using the floor plan of the DLR building shown in Figure 10.

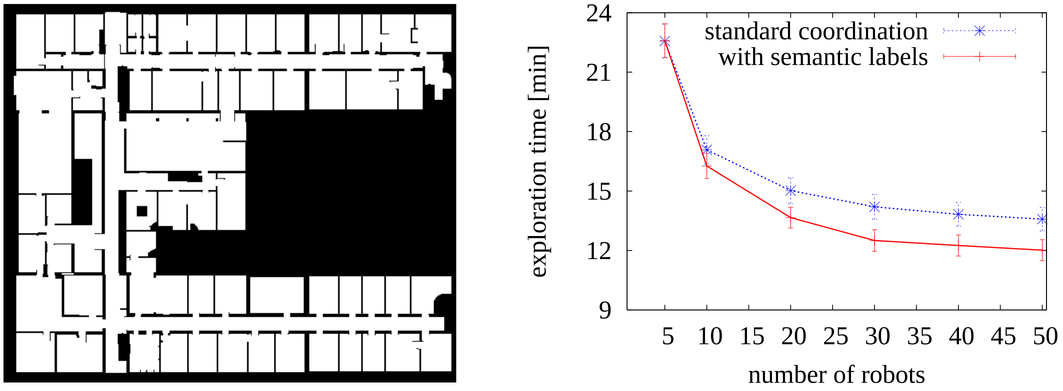


Figure 10: Floor plan of the German Aerospace Center (DLR) and the corresponding results of our exploration system.

5.2 Influence of Noise in the Semantic Place Information

In the experiments presented above, we assumed that the robots are able to correctly classify the different target location into the semantic categories. This assumption, however, is typically not justified. In this experiment, we evaluate the performance of our approach for different classification error rates. We evaluated the exploration time for a classifier which randomly misclassified 5%, 10%, and 15% of the places. Figure 11 depicts a plot comparing the different error rates. As can be seen, even at a high error of 10%, our approach significantly outperforms the coordination technique that ignores the semantic information. When the error of the classification exceeds 15%, the exploration time is still reduced, although this result is not significant anymore.

5.3 Applying a Trained Classifier in New Environments

This experiment is designed to illustrate that it is possible to train a classifier in an environment and transfer it to a new environment with a substantially different geometric structure. Obviously, the performance of the classifier decreases in such situations. The results we obtain, however, still provide a sufficiently accurate classification. Figure 12 shows two labeled maps. The one in the first row was labeled manually and used to learn the classifier using AdaBoost. For the environment depicted in the lower image, we simulated an observation for each grid cell and than used the trained classifier to label the positions. As can be seen, the spacial structures are quite different but the classification is good except of a small areas which are wrongly classified. Large parts of the misclassified areas in this experiment are located at the ends of the corridors. This is mainly due to the fact that large parts of the area covered by scans recorded at these locations actually cover a corridor.

We then used this classification result to perform an exploration task. The results of this experiment are depicted in Figure 13. The figure plots the time needed to explore the environment using our approach with the true labels, with the labels estimated by our classifier, and without using place information at all. As can be seen, there is only a small time overhead when using the

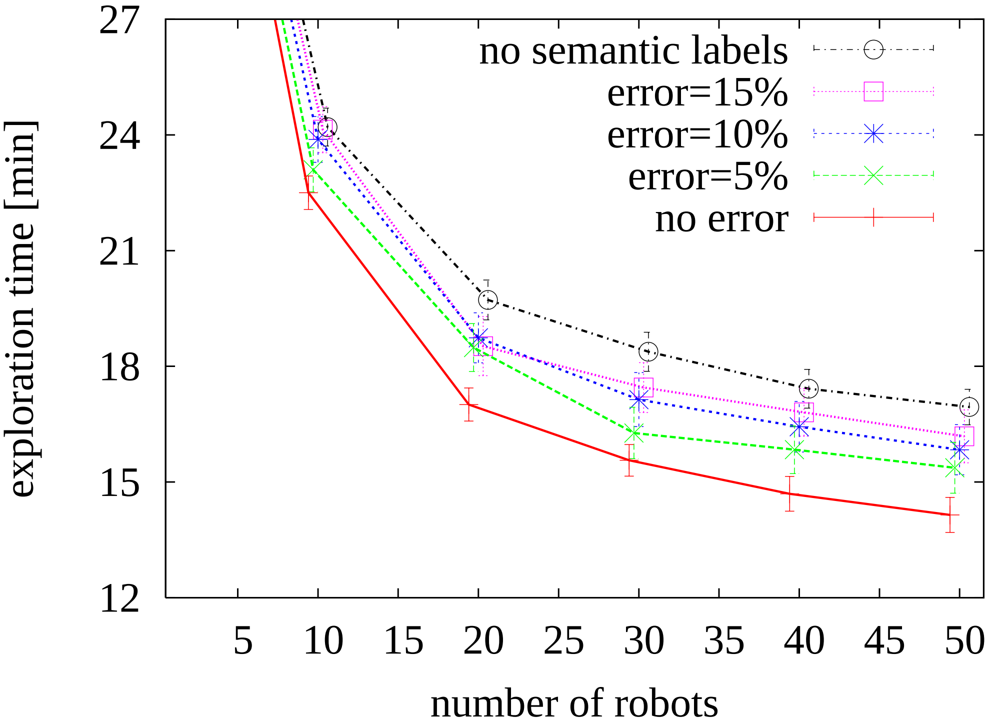


Figure 11: Exploration results with wrongly labeled places.

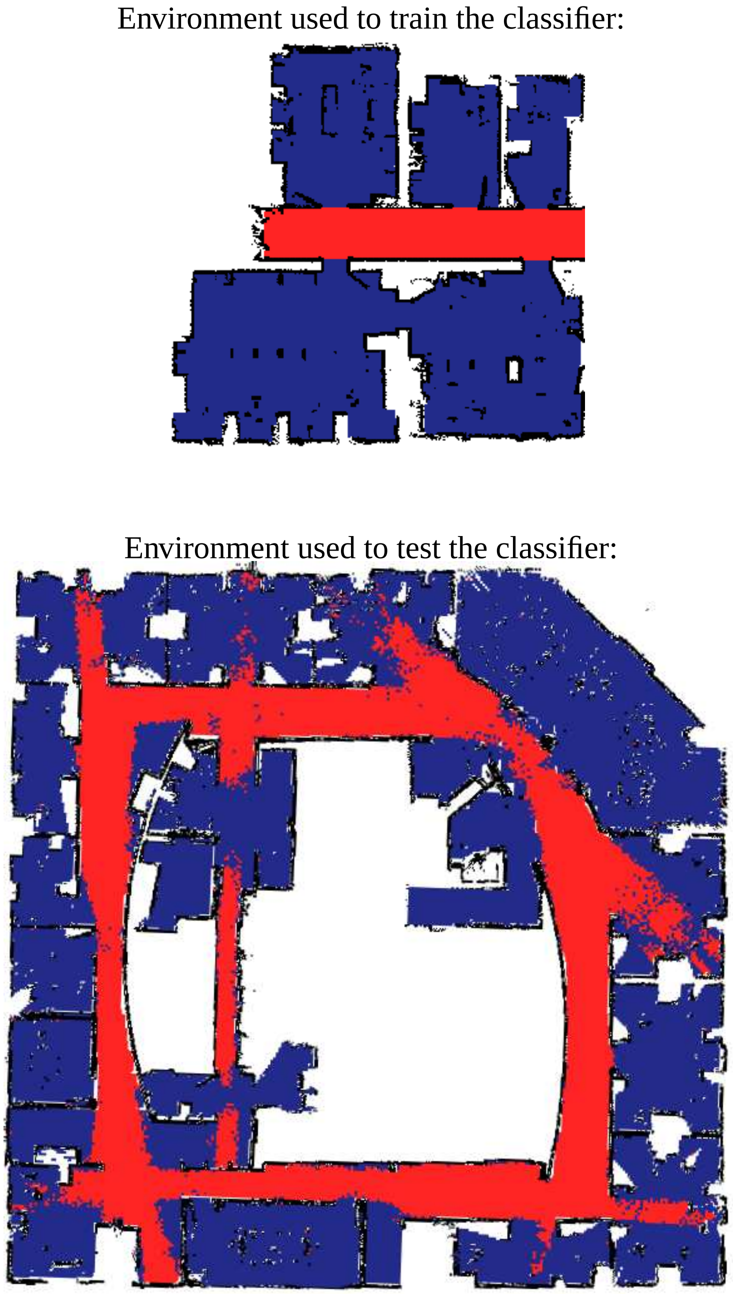
estimated labels compared to the true ones. This indicates that even transferring such a classifier to unknown environments provides a speed-up in the context of multi-robot exploration.

5.4 Improvements of the HMM Filtering and Error Analysis of the Classifier

In this experiment, we want to analyze the actual error of our place classification system and illustrate the improvements of the HMM filtering. To do so, we labeled an environment, trained a corridor classifier using AdaBoost, and used a test set to evaluate the success rate. Whenever a single full 360 degree laser range scan was available, we obtained highly accurate classification results in different office environments. In this case, the error-rate was typically between 2% and 4%.

Figure 14 depicts the result of our classifier depending on the number of invalid readings caused by unknown grid cells close to frontiers. The *x*-axis shows the size of a continuous block of maximum range measurements (here with an angular resolution of the laser of 1 degree). As can be seen, if only half of the observations are available, the classification error rate is between 18% and 19%.

First, we determined the success rate of directly classifying frontier cells without using HMM filtering. In this case, the average classification rate was in average 81.2%. By considering the exploration speed-up depending on the classification rate depicted in Figure 11, such a high error



Environment used to train the classifier:

Environment used to test the classifier:

Figure 12: The training examples for the classifier were trained in the map shown in the top image. In contrast to this, the lower image shows the resulting classification output. The classification for each place was performed based on a laser range scan simulated at the corresponding location in the map. As can be seen, even if the structure of the environment is significantly different, the classification output is reasonable. Red corresponds to corridor locations, blue to rooms.

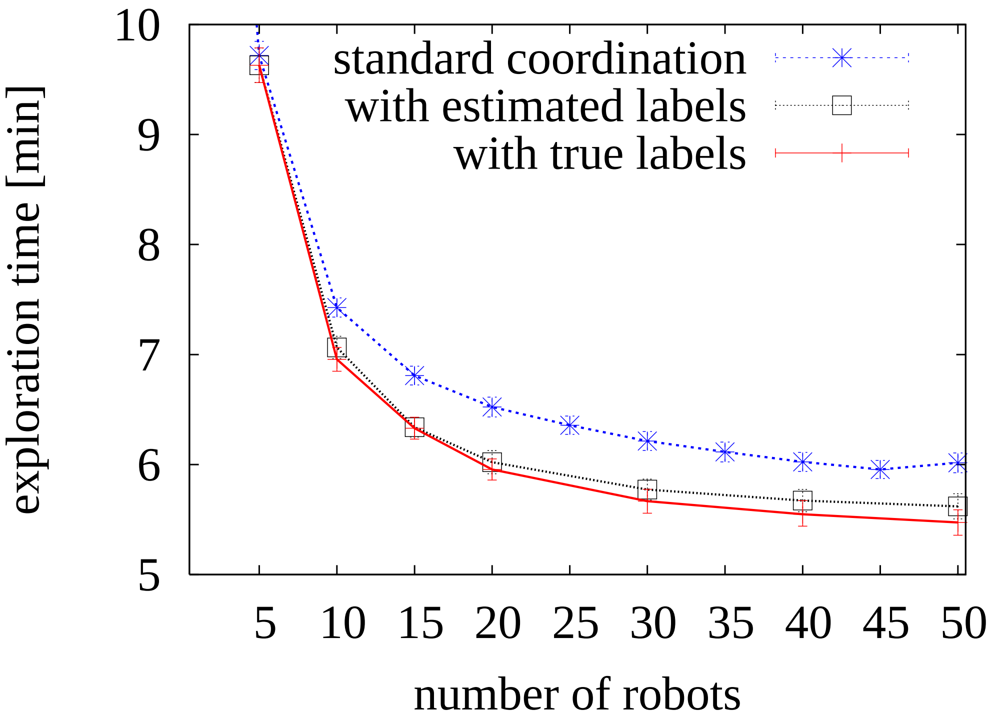


Figure 13: Results obtained in simulation using the Intel Research Lab data.

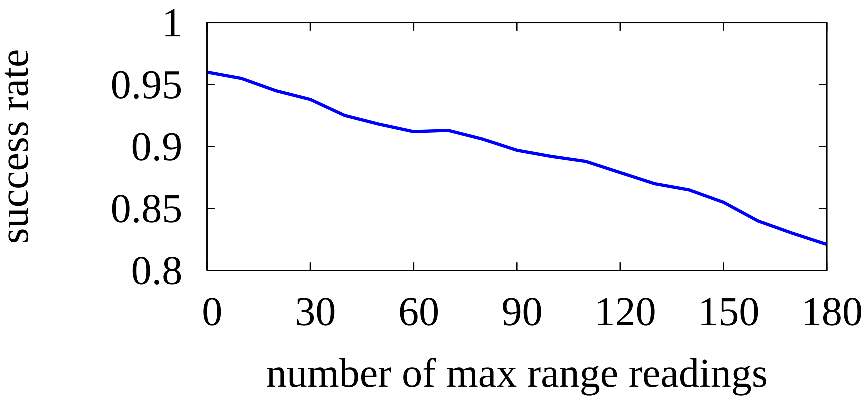


Figure 14: This plot illustrates the classification performance of the standard classifier depending on how many consecutive beams of a 360 degree observation (1 degree angular resolution) are maximum range readings.

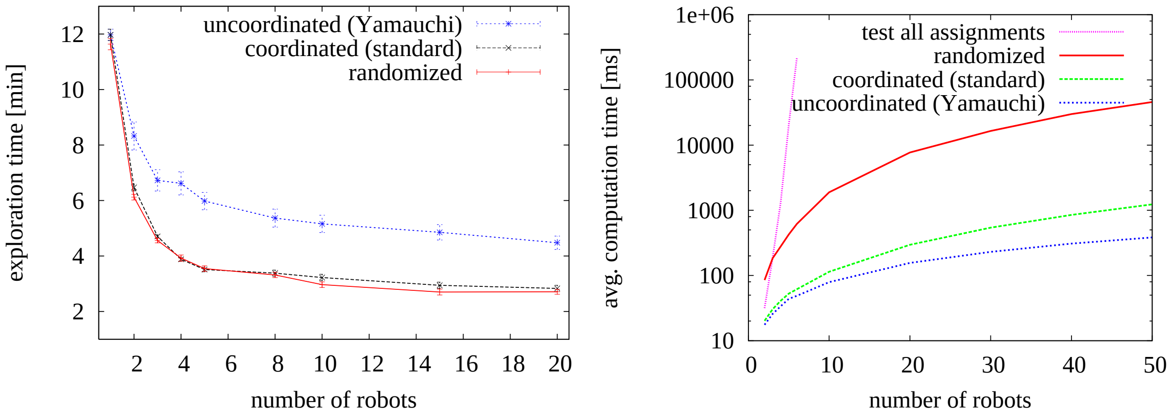


Figure 15: Left: Typical exploration performance in a corridor environment depending on the assignment strategy. Right: Comparison of the runtime of the different approaches (Pentium-4 CPU, 2.8 GHz).

rate is not sufficient to obtain an significant speed-up.

Second, we applied our HMM-based filtering approach that generates virtual trajectories towards frontiers and in this way incorporates the spatial dependencies between the nearby locations. As a result, we obtained an average success rate of 92.8%. This is a good result considering that we obtained an average success rate in this scenario of 96.2% if all observations are perfectly known (see Figure 14). This fact illustrates that the HMM is an useful tool to improve the place labeling especially if not the full 360 degree range scan is available. It allows us to estimate the semantic labels with a comparably low error rate. In this way, our technique can be used to significantly speed up multi-robot exploration by considering semantic information about places in the environment.

In sum, our experiments demonstrate that semantic place information can significantly reduce the exploration time even under classification errors.

5.5 Efficiency of the Coordination Approach

Given the coordination approach depicted in Algorithm 2, one might argue that the presented technique is a greedy method for making assignments and thus suboptimal. However, for finding the optimal solution, an exponential number of assignments has to be evaluated which is infeasible in practice. An alternative to evaluating all possible assignments is a randomized approach that starts with the result of our method and then randomly exchanges target locations to seek for a better assignment (compare [4, 39]).

It is worth noting that such a randomized optimization strategy usually yields slightly better results than our coordination technique although the improvement is not significant. This slight improvement, however, comes with a significantly increased computational cost, see Figure 15. Given these results, we argue that the coordination approach described in this paper is an appropriate method in practice since it leads to an efficient coordination behavior requiring comparably small computational resources.

6 Related Work

The various aspects of the problem of exploring unknown environments with mobile robots have been studied intensively in the past. Many approaches have been proposed for exploring unknown environments with a *single robot* [6, 9, 11, 18, 26, 45, 46]. Most of these approaches guide the robot to the closest unexplored area. These techniques mainly differ in the way the environment is represented. Popular representations are topological [6, 26], metric [11], or grid-based [45, 46]. Furthermore, there is a serious amount of theoretical work providing a mathematical analysis of the complexity of exploration strategies including comparisons for single robots [1, 7, 8, 24]. Additionally, Lee and Recce [28] provide an experimental analysis of the performance of different exploration strategies for one mobile robot.

There are approaches which address the problem of coordinating two robots. The work presented by Bender and Slonim [3] theoretically analyzes the complexity of exploring strongly-connected directed graphs with two robots. Roy and Dudek [36] focus on the problem of exploring unknown environments with two robots and present an approach allowing the robots with a limited communication range to schedule rendezvous. The algorithms are analyzed analytically as well as empirically using real robots.

Also various aspects of the problem of exploring unknown environments with multiple mobile robots have been studied intensively in the past. For example, Rekleitis *et al.* [33, 35] focus on the problem of reducing the odometry error during exploration. They separate the environment into stripes that are explored successively by the team of robots. Whenever one robot moves, the other robots are kept stationary and observe the moving robot, a strategy similar to the presented by Kurazume and Shigemi [27]. Whereas this approach can significantly reduce the odometry error during the exploration process, it is not designed to distribute the robots over the environment. Rather, the robots are forced to stay close to each other in order to remain in the visibility range. Thus, using these strategies for multi-robot exploration one cannot expect that the exploration time is significantly reduced.

Koenig *et al.* [23] analyze different terrain coverage methods for ants which are simple robots with limited sensing and computational capabilities. Furthermore, there has been research on how to deal with limited communication in the context of multi-robot exploration [4, 34].

Yamauchi [44] presented a technique to learn maps with a team of mobile robots. In this approach, the robots exchange information about the map that is continuously updated whenever new sensor input arrives. To acquire knowledge about the environment, all robots move to the closest frontier cell. Is has been shown by Burgard *et al.* [4] that coordination methods that assign robots to targets given the assignments of the team mates outperform the technique of Yamauchi.

The coordination technique presented is this paper is an extention of [4]. We also discount the utility of target locations if they are visible from a goal location already assigned to a robot. In contrast to [4], the approach presented in this paper estimates and incorporates background knowledge about environmental structure into the goal point assignment procedure. As shown in the experimental section, this knowledge allows a team of robots to more efficiently coordinate their actions and thus to finish the exploration mission in a shorter period of time.

Zlot and colleagues [47] proposed an architecture for mobile robot teams in which the exploration is guided by a market economy. They consider sequences of potential target locations for each robot and trade tasks between the robots using single-item first-price sealed-bid auctions. Such auction-based techniques have also been applied by Gerkey and Matarić [15] to efficiently solve the task allocation problem with a group of robots. These approaches do not incorporate the information that assigned target locations might be observed by other robot. Nevertheless, after training for a significant amount of time such methods are likely to show a similar performance to the so-called 'randomized' method presented in Burgard *et al.* [4] in which targets locations and robots are reassigned in case a more efficient solution can be obtained.

Stroupe *et al.* presented the MVERT-approach [41], which uses a greedy method that selects robot-target pairs based on proximity. The goal of the action selection is to maximize cooperative progress toward mission goals. Matarić and Sukhatme [30] consider different strategies for task allocation in robot teams and analyze the performance of the team in extensive experiments.

Howard [20] presented an approach to learn grid maps with multiple robots that explicitely considers the pose uncertainty of the vehicles – an aspect that is not considered in our work. He applies multiple Rao-Blackwellized particle filters to estimate the joint posterior about the map of a robot and its trajectory. Whenever two robots meet, he establishes a corresponding point in the maps which allows him to build a common model of the environment with all robots. Ko *et al.* [21] present an approach that uses the Hungarian method to compute the assignments of frontier cells to robots. In contrast to our work, Ko *et al.* mainly focuses on finding a common frame of reference in case the start locations of the robots are not known. In contrast to this, our approach assumes that the robots start with a known relative offset.

The semantic labels used to improve multi-robot coordination can be seen as background knowledge about spacial structures. Fox *et al.* [13] presented a technique which aims to learn background knowledge in typical indoor environments and later on use that knowledge for map building. They apply their approach to decide whether the robot is seeing a previously built portion of a map, or is exploring new terrain. This is an important information for reliably closing loops. In contrast to Fox *et al.*, our approach improves the coordination between robots and thus leads to a more efficient exploration strategy.

This paper is an extention of a previous publication [40]. In our current paper, we presented our approach to learn semantic place labels in more detail. Additionally, we carried out experiments that illustrate that our classifier can be learned in one environment and then used in a different one. This is an important prerequisite to use this technique in the context of exploring unknown environments. Furthermore, this paper presents a more detail experimental evaluation of the overall approach.

Due to the best of our knowledge, there is no work that investigates how semantic information about places in the environment can be used to optimize the collaboration behavior of a team of robots. The contribution of this paper is an approach that estimates and explicitly uses semantic information to more efficiently spread the robots over the environment. This results in an more balanced target location assignment with less interferences between robots. As a result, the overall time needed to cover the whole environment with the robots' sensors can be significantly reduced.

A series of authors addressed the problem of learning environmental structures and semantic information with mobile robots. Such methods, however, have not been applied to coordinate robots during exploration. For example, Koenig and Simmons [22] use a pre-programmed routine to detect doorways from range data. Althaus and Christensen [2] 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.* [32] 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 [25] apply different learning algorithms to learn topological maps of the environment. Finally, Torralba and colleagues [42] use hidden Markov models for learning places from image data. In our work, we apply a technique originally proposed by Martínez Mozos *et al.* [29]. This technique uses simple features extracted from laser range scans to train a set of classifiers and in this way are able to label a place given a single 2D laser range observation. The leaning approach applies AdaBoost [14] to boost the simple features, which on their own are insufficient for a reliable categorization of places.

7 Conclusion

In this paper, we addressed the problem of efficiently coordinating a team of mobile robots that has to explore an unknown environment. We assume that the robots are equipped with laser range finders to perceive theirs surroundings and that they always know their relative positions. The capability to coordinate the individual robots is important for efficient exploration since in this way redundant work and the risk of interferences between robots can be reduced. Since indoor environments are constructed by humans, they typically consist of structures like corridors and rooms. The knowledge about the type of place at a target location offers the potential to improve the distribution of robots over the environment and thus to reduce redundant work as well as the risk of interference between the robots. The reason for this it that corridors typically provide a high number of branchings to adjacent rooms and therefore lead to a high number of potential target locations during exploration. The more unexplored target locations are available when assigning the targets to robots, the faster the team can explore the environment. This is due to the fact that the robots can be better distributed over the environment.

Our proposed technique to coordinate the robots takes into account the type of place at potential target locations. It integrates this information into the assignment procedure of target locations to robots. We use a decision theoretic procedure that additionally considers the expected travel time as well as the utility of a target location based on visibility constraints to other, already assigned target locations. As a result, the overall exploration time can be reduced compared to collaboration approaches that ignore semantic place information. To autonomously estimate the semantic place information, we learn a classifier using the AdaBoost algorithm. We furthermore apply a hidden Markov model to consider spacial dependencies between nearby locations. This allows us to reduce the error in the process of estimating the type of place. Our approach has been implemented and tested in extensive simulation runs with up to 50 robots. Experiments presented in this paper illustrate that a team of robots can complete their exploration mission in a significantly shorter period of time using our approach. Furthermore, we believe that utilizing semantic information during exploration is not restricted to the exploration technique presented here and that it can be easily integrated into other coordination approaches.

One way of extending this work, would be to learn the place labels in an unsupervised fashion. In this way, the system might be able to determine on its own what kind of spacial structures are useful for coordinated exploration and does not rely on manually defined labels.

Acknowledgment

This work has partly been supported by the DFG under contract number SFB/TR-8 and by the EC under contract number FP6-004250-CoSy. The Fort Sam Houston map was obtained from Radish, thanks go to Richard Vaughan for providing this data. Furthermore, the authors would like to thank Dirk Haehnel for providing the Intel Research Lab dataset and Udo Frese for the DLR floor plan.

References

[1] S. Albers, K. Kursawe, and S. Schuierer. Exloring unknown environments with obstacles. *Algotithmica*, 32:123–143, 2002.

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

[3] M. Bender and D. Slonim. The power of team exploration: two robots can learn unlabeled directed graphs. In *Proc. of the 35rd Annual Symposium on Foundations of Computer Science*, pages 75–85, 1994.

[4] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–378, 2005.

[5] Y.U. Cao, A.S. Fukunaga, and A.B. Khang. Cooperative mobile robotics: Antecedents and directions. *Journal of Autonomous Robots*, 4(1):7–27, 1997.

[6] H. Choset. Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, 2001.

[7] X. Deng, T. Kameda, and C. Papadimitriou. How to learn in an unknown environment. In *Proc. of the 32nd Symposium on the Foundations of Computational Science*, pages 298–303. IEEE Computer Society Press, Los Alamitos, CA, 1991.

[8] X. Deng and C. Papadimitriou. How to learn in an unknown environment: The rectilinear case. *Journal of the ACM*, 45(2):215–245, 1998.

[9] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865, 1991.

[10] G. Dudek, M.. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Journal of Autonomous Robots*, 3(4):375–397, 1996.

[11] T. Edlinger and E. von Puttkamer. Exploration of an indoor-environment by an autonomous mobile robot. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1278–1248, Munich, Germany, 1994.

[12] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Collaborative multi-robot localization. In *Proc. of the 23rd German Conference on Artificial Intelligence*, pages 325–340. Springer Verlag, 1999.

[13] D. Fox, J. Ko, K. Konolige, and B. Stewart. A hierarchical bayesian approach to the revisiting problem in mobile robot map building. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, Siena, Italy, 2003.

[14] 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.

[15] B.P. Gerkey and M.J. Matarić. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758– 768, 2002.

[16] D. Goldberg and M.J. Matarić. Interference as a tool for designing and evaluating multi-robot controllers. *Journal of Robotics & Autonomous Systems*, 8:637–642, 1997.

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

[18] H.H. González-Baños, E. Mao, J.C. Latombe, T.M. Murali, and A. Efrat. Planning robot motion strategies for efficient model construction. In *Proc. Int. Symp. on Robotics Research (ISRR)*, pages 345–352, Snowbird, UT, USA, 2000.

[19] D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige. Many robots make short work. *AI Magazine*, 18(1):55–64, 1997.

[20] Andrew Howard. Multi-robot simultaneous localization and mapping using particle filters. In *Robotics: Science and Systems*, pages 201–208, Cambridge, MA, USA, 2005.

[21] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3238, Las Vegas, NV, USA, 2003.

[22] 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.

[23] S. Koenig, B. Szymanski, and Y. Liu. Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence*, 31:41–76, 2001.

[24] S. Koenig, C. Tovey, and W. Halliburton. Greedy mapping of terrain. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Seoul, Korea, 2001.

[25] B. Kuipers and P. Beeson. Bootstrap learning for place recognition. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, 2002.

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

[27] R. Kurazume and N. Shigemi. Cooperative positioning with multiple robots. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1250–1257, Munich, Germany, 1994.

[28] D. Lee and M. Recce. Quantitative evaluation of the exploration strategies of a mobile robot. *International Journal of Robotics Research*, 16(4):413–447, 1997.

[29] 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.

[30] M.J. Matarić and G. Sukhatme. Task-allocation and coordination of multiple robots for planetary exploration. In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, pages 61–70, Budapest, Hungary, 2001.

[31] A. Meijster, J.B.T.M. Roerdink, and W.H. Hesselink. *Mathematical Morphology and its Applications to Image and Signal Processing*, chapter A General Algorithm for Computing Distance Transforms in Linear Time, pages 331–340. Kluwer Academic Publishers, 2000.

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

[33] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proc. of International Joint Conference in Artificial Intelligence (IJCAI)*, volume 2, pages 1340–1345, 1997.

[34] I. Rekleitis, V. Lee-Shue, A. Peng New, and H. Choset. Limited communication, multi-robot team based coverage. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3462–3468, New Orleans, LA, USA, 2004.

[35] I. Rekleitis, R. Sim, G. Dudek, and E. Milios. Collaborative exploration for the construction of visual maps. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Maui, HI, USA, 2001.

[36] N. Roy and G. Dudek. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Journal of Autonomous Robots*, 11(2):117–136, 2001.

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

[38] M. Schneider-Fontan and M.J. Matarić. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14(5):815–822, 1998.

[39] C. Stachniss. *Exploration and Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Department of Computer Science, April 2006.

[40] C. Stachniss, O. Martínez-Mozos, and W. Burgard. Speeding-up multi-robot exploration by considering semantic place information. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 1692–1697, Orlando, FL, USA, 2006.

[41] A.W. Stroupe, R. Ravichandran, and T. Balch. Value-based action selection for exploration and mapping with robot teams. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 4090–4197, New Orleans, LA, USA, 2004.

[42] 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)*, Nice, France, 2003.

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

[44] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proc. of the Second International Conference on Autonomous Agents*, pages 47–53, Minneapolis, MN, USA, 1998.

[45] B. Yamauchi, A. Schultz, and W. Adams. Integrating exploration and localization for mobile robots. *Adaptive Behavior*, 7(2):217–229, 1999.

[46] A. Zelinsky, R. Jarvis, Byrne J., and S. Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robots. In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, pages 533–538, Tokyo, Japan, 1993.

[47] R. Zlot, A.T. Stenz, M.B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Washington, DC, USA, 2002.