Learning Wayfinding Heuristics Based on Local Information of Object Maps

Dominik Joho Martin Senk Wolfram Burgard
Department of Computer Science, University of Freiburg, Germany

***Abstract*— In recent years, the problem of inferring and utilizing semantic information has gained considerable interest within the mobile robotics community. In this paper we focus on the problem of how to utilize the local semantic information of objects in a map to solve a navigation task more efficiently. In particular, we consider a wayfinding task and choose a supermarket environment as an example domain. We present an approach allowing a mobile robot to efficiently find the location of a target product in an unknown supermarket and to guide the search by heuristic rules based on information about the objects in the vicinity of the robot. The wayfinding strategy is learned from data by observing optimal search paths in a training set of supermarkets and then applied and evaluated in a previously unseen supermarket. We evaluate different search strategies and also give a comparison to the performance of humans in a real market. Our results demonstrate, that the learned wayfinding heuristics yield significantly shorter search paths than a standard search technique.**

***Index Terms*— Navigation, Wayfinding, Object Maps**

I. Introduction

In recent years, the problem of inferring and utilizing semantic information in the context of mobile robot navigation has gained substantial interest [2, 3, 7, 9, 11, 14, 15]. This is motivated by the observation that mobile robots can benefit from semantic information in various ways, and especially to more efficiently carry out their tasks. For example, semantic maps are envisioned to be more amenable or communicable to humans, which can be an important property in the fields of service robotics and human-robot interaction [15]. Additionally, they allow robots to reason about their environment and can be considered as a major step towards bridging the gap between perception and action.

In this paper we focus on the problem of how to exploit the semantic information in a local map and about the vicinity of the robot to accomplish a wayfinding task more efficiently, than it would have been possible without such information. As an application domain, we consider typical supermarket environments and propose an approach to efficiently find the location of a target product in a previously unseen supermarket. The search is guided by heuristic rules that depend only on local information that is immediately accessible from the robot's current location, such as the visible products or the type of shelves. We chose a supermarket as it is an excellent example of an environment densely populated with many different objects that are arranged in a meaningful way and that exhibit strong spatial dependencies.

We believe that the knowledge utilized during the search for a product can be expressed as a set of heuristic rules

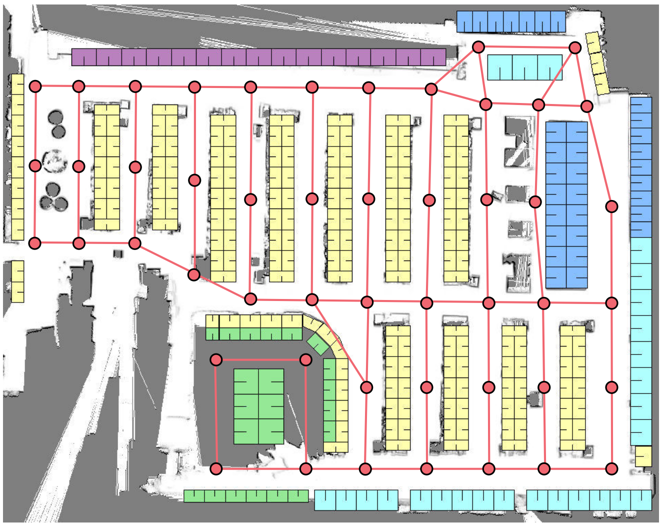


Fig. 1: Example map of a real supermarket environment. In our approach, maps contain shelf locations, shelf types, and product locations. The underlying structure is a graph. The task of the robot is to efficiently find a target product and to guide the search by utilizing the local information given by its current location and the products in its direct vicinity.

and that these rules can be learned from data by observing optimal search behavior. We realize this by learning a decision tree that classifies the outgoing edges of the robot's current location into promising and non-promising directions. The main problem then is to define the relevant edge attributes that are informative enough to guide the search efficiently to the target product.

The definition of the edge attributes is the only domain-specific part of our wayfinding strategy, and our approach is therefore also applicable to different application scenarios, like finding an object in an office or finding buildings or points of interest in a city. As an example, imagine you are in need for medical treatment and you are driving through a city in search for a doctor's practice. At a distance you can see the bright sign of a pharmacy. In this situation, it would certainly be helpful to know that doctors often share the same house with a pharmacy store.

Even for humans, the task of efficiently finding a product in a market is not an easy one and presumably human wayfinding [13] is also guided by a set of domain-specific heuristic rules, like “if the goal is to find milk, follow the wall”. We will therefore also compare our technique to the performance of human participants that took part in a field study conducted in a real supermarket [6].

This paper is organized as follows. After discussing related work, Section III introduces our representation of the supermarket environments. Section IV then describes how the search strategy has been learned from data of optimal search paths. In Section V we present the results of an experimental evaluation of various search strategies. We demonstrate that our proposed technique yields significantly shorter search paths than a search strategy that does not take domain-specific information into account.

II. RELATED WORK

Cocora *et al.* [2] investigated the related problem of how to efficiently find the entrance hall in a hotel. They learned a relational navigation policy that utilizes the information about the type of rooms and corridors that are directly connected to the robot's current location. Kollar *et al.* [7] utilized a Markov random field based on statistics of object cooccurrences to reason about the location of a query object when given the locations of other objects in a global map. They subsequently used the inferred likelihood map to plan a search path such that the expected path length is minimized. In contrast to their work, we need to explore the environment and cannot rely upon a priori known locations of certain landmark objects. Galindo *et al.* [3] focused on the problem of how a robot can improve its task planning by relying on semantic information about its domain. In particular, they defined an ontology about typical home-like environments and generated plans to find unseen objects or type of rooms, e.g. a bedroom. Kulyukin *et al.* [8] built a wayfinding assistant for the visually impaired, which can guide persons to a goal location in a supermarket based on RFID measurements. However, for path planning they rely on a given map of the environment and known goal locations.

Learning a wayfinding strategy by observing optimal search paths can also be regarded as a form of learning from demonstration and several authors applied imitation learning techniques to navigation problems. Silver *et al.* [12], for example, used an imitation learning technique to improve outdoor path planning based on overhead terrain data. They learned a cost-function from example paths provided by an imperfect domain-expert. Ziebart *et al.* [16] learned routing preferences of drivers by formulating the problem as an inverse reinforcement problem. They relied on road segment features like speed limits, road type, number of lanes, etc.

The problem addressed in this paper is distinguished from the above mentioned work in the sense that we are facing an environment that is densely populated with a large number of different objects, which are arranged in a meaningful way that can be exploited in order to guide the search for a specific object.

III. MODELLING THE ENVIRONMENT

A supermarket $m \in \mathcal{M}$ contains a set of shelves $\mathcal{S}_m$ and a route graph $\mathcal{G}_m = (V, E)$. Each shelf $s \in \mathcal{S}_m$ is associated with a location $\boldsymbol{\ell}_s = (x_s, y_s)$ and an orientation θ_s. The relation $\textsc{inMarket} \subset \mathcal{S} \times \mathcal{M}$ associates each shelf with its corresponding market. Furthermore, we define a set of shelf types $\mathcal{T} = \{\textsc{Normal}, \textsc{Cooling}, \textsc{Freezer}, \textsc{Counter}, \textsc{Grocery}\}$ and each shelf is associated with exactly one type as defined by the relation $\textsc{type} \subset \mathcal{S} \times \mathcal{T}$. Each shelf contains at least one product and the same product might be placed in several shelves, as defined by the relation $\textsc{inShelf} \subset \mathcal{P} \times \mathcal{S}$. For this, we define a set of 196 products at the granularity of small categories like “sugar”, “pizza, “apple”, “tea”, etc. The relation $\textsc{categOf} \subset \mathcal{P} \times \mathcal{C}$, associates each product with a product category. For this, we define a set $\mathcal{C}$ of 20 product categories with a coarser granularity like “breakfast”, “dairy products”, “vegetables & fruits”, etc.

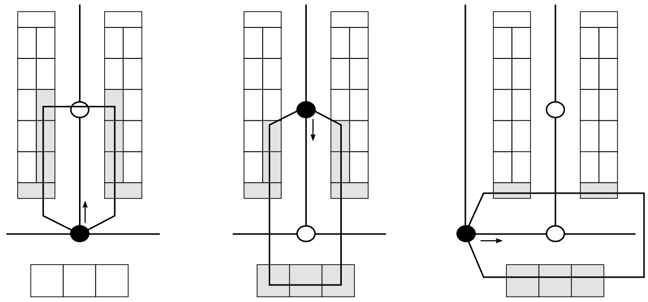


Fig. 2: Three example situations for illustrating the short range visibility. Gray shelves are visible, white shelves are not visible. The location of the robot is indicated by the black node and its orientation is indicated by the arrow.

The nodes V of a route graph $\mathcal{G}_m = (V, E)$ model the decision points in the supermarket and the directed edges define the reachability between decision points. While the reachability could have been modeled with undirected edges, the visibility of the shelves also depends on the current node (the robot's current location) and therefore is defined over directed edges. We use two variants of a visibility relation that defines the shelves that are visible when looking into the direction of a certain edge. The first one is a long range variant $\textsc{shelfVisL} \subset E \times \mathcal{S}$ and the second is a short range variant $\textsc{shelfVisS} \subseteq \textsc{shelfVisL}$. This is motivated by the fact that although certain information, like the type of a shelf, can be determined reliably over long distances, some information can only be determined when one is in close vicinity to a shelf, like for example the products contained within a shelf. Three example situations illustrating the short range visibility are depicted in Fig. 2. On the basis of the two visibility relations we define several other visibility relations, like the visible products

$$\textsc{prodVis} = \{(e, p) \mid \textsc{shelfVisS}(e, s), \textsc{inShelf}(p, s)\}, \quad (1)$$

and the visible product categories

$$\textsc{categVis} = \{(e, c) \mid \textsc{prodVis}(e, p), \textsc{categOf}(p, c)\}. \quad (2)$$

The visibility of shelf types is modeled in such a way, that we can distinguish whether the shelf type is seen in the direct vicinity, or being observed at a further distance:

$$\textsc{typeVisS} = \{(e, t) \mid \textsc{shelfVisS}(e, s), \textsc{type}(s, t)\} \quad (3)$$

$$\textsc{typeVisL} = \{(e, t) \mid \textsc{shelfVisL}(e, s), \textsc{type}(s, t)\}. \quad (4)$$

The proposed wayfinding strategy utilizes the information associated with each edge to decide which edge to follow. In the next section we describe how we learn such a strategy from data by observing optimal search paths.

IV. Learning the Wayfinding Strategy

We are interested in learning a reactive wayfinding strategy that depends only on local information in order to find a certain target product. We therefore classify the outgoing edges of the current node by a decision tree into promising and non-promising directions based on the information associated with each edge. For learning such a decision tree, we first need to define appropriate edge attributes and then generate training data by observing optimal search paths in a training set of supermarkets. To evaluate the strategy, we apply it to a previously unseen market.

A. Defining Edge Attributes

One obvious information, by which the search should be guided, is which products and product categories are visible at a certain edge. If we are searching for coffee and an aisle contains tea, or in general breakfast products, then this edge is certainly a promising candidate. But the decision should also be influenced by additional factors. If we know that the edge has been visited already, we could reject it in order to avoid loops. Also the type of an edge might be of interest, such if an edge belongs to an aisle that follows a wall (wall aisle), because some products, like milk, are only located in such aisles. Likewise, we define main aisles as aisles that follow a main direction in a market and from which many narrow side aisles branch off. Next, it is informative if the robot is approaching certain landmarks in the supermarket, like the entrance, the exit, or the back of the market. Vegetables, for example, are always located near the entrance in our markets. Thus, each optimal search path for finding apples would mostly contain edges that are approaching the entrance. Likewise, frozen food is usually in the back of the market and wine and non-food are near the exit of the market.

We also use statistics about the expected relative product position between the entrance and the exit based on the data of all training markets. The relative position of a shelf s with respect to the location $\boldsymbol{\ell}_{en}$ of the entrance node and the location $\boldsymbol{\ell}_{ex}$ of the exit node of the corresponding market is defined as

$$\mathrm{relPos}(s) = \frac{\|\boldsymbol{\ell}_s - \boldsymbol{\ell}_{en}\|}{\|\boldsymbol{\ell}_s - \boldsymbol{\ell}_{en}\| + \|\boldsymbol{\ell}_s - \boldsymbol{\ell}_{ex}\|} \tag{5}$$

The expected relative position of a *product* is then defined as the average of these values for all shelves that contain this product in the training markets M_t

$$S_p = \{s \mid \text{INSHELF}(p, s), \text{INMARKET}(s, m), m \in M_t\} \tag{6}$$

$$\mathrm{expRelPos}(p) = |S_p|^{-1} \sum_{s \in S_p} \mathrm{relPos}(s). \tag{7}$$

We define a binary edge attribute (No. 222 in Table I) that indicates if the robot would be approaching the expected relative position of the target product by following that edge.

Furthermore, we calculate the average Euclidean distance $\mathrm{prodDist}(p_i, p_j)$ for each pair (p_i, p_j) of products based on their locations in the training markets. If we denote by $P_{i,j}$ the visible products that are associated with an outgoing edge $e_{i,j}$ from the current node v_i to a possible successor node v_j, then the average product distance of this edge to the target product p_t is defined as

$$\mathrm{avrgProdDist}(e_{i,j}, p_t) = |P_{i,j}|^{-1} \sum_{p \in P_{i,j}} \mathrm{prodDist}(p_t, p) \tag{8}$$

We define an indicator attribute (attribute No. 223) that is set to true if an edge has the lowest average product distance of all outgoing edges of the current node, and thus can be considered to be the most promising edge with respect to the expected product distances. Likewise, we define an attribute that uses the path distance on the route graph between products instead of the Euclidean distance (attribute No. 224). As it is not easy to decide beforehand whether the path distance or the Euclidean distance is a more reliable indicator for product distances we use both attributes and let the learning algorithm decide which one to use during the induction of the tree. A complete list of all attributes can be seen in Table I.

TABLE I: The attributes that are used to characterize an edge. All attributes are binary. In the experimental evaluation we test different combinations of subsets (a–d) of these attributes.

Att. Subset	Att. No.	Description
a	1	Edge already visited
a	2–197	Product $p_i \in \mathcal{P}$ visible
a	198–217	Product of category $c_i \in \mathcal{C}$ visible
a	218	Shelf of type NORMAL visible (short range)
a	219	Shelf of type COOLING visible (short range)
a	220	Shelf of type FREEZER visible (short range)
a	221	Shelf of type COUNTER visible (short range)
b	222	Leads to expected relative position
b	223	Has smallest avrg. Euclidean dist. to product
b	224	Has smallest avrg. path dist. to product
c	225	Current node belongs to a main aisle
c	226	Next node belongs to a main aisle
c	227	Next node belongs to a wall aisle
c	228–230	Leads to entrance, exit, or back of the market
d	231	Shelf of type COOLING visible (long range)
d	232	Shelf of type FREEZER visible (long range)
d	233	Shelf of type GROCERY visible (long range)

B. Generating Training Data

We use a fixed set of 15 target products. These are the same products that human participants had to find in a field study conducted in the very same supermarket in which we will evaluate our strategy. We learn a separate decision tree for each of these 15 target products.

We determine for each node in a training supermarket the shortest path to a given target product. Each node of the optimal path corresponds to a local decision for taking a certain outgoing edge (the one that leads to the next node of the optimal path) and for rejecting all other outgoing edges of that node. This way, each optimal search path contributes a set of positive and negative examples of edges to be taken or not, respectively. The positive and negative examples of all paths for all starting positions in all training supermarkets then constitute the training data for learning the decision tree for a given target product.

As there might exist more than one optimal path from a starting location to the target location, we search for more than just a single shortest path to generate training data. Additionally, as the decision points are placed manually, there might be small differences between nearly optimal paths. From a topological point of view, these paths would still qualify as optimal paths. We therefore accept all paths as optimal, which are no longer than a given small threshold when compared to the actual shortest path.

C. Decision Tree Learning and Pruning

We use the well known ID3 algorithm [10] to learn a decision tree. For convenience, we restate the basic idea of the algorithm. The tree is constructed top-down and each node is associated with a set of positive e_p and negative e_n examples and a set A of yet untested attributes. At each node an attribute $a \in A$ is chosen that maximizes the information gain

$$G(a) = I(|e_p|, |e_n|) - \sum_{v \in a} \frac{|e_{p(v)}| + |e_{n(v)}|}{|e_p| + |e_n|} I(|e_{p(v)}|, |e_{n(v)}|) \tag{9}$$

where

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n} \tag{10}$$

denotes the information entropy and $e_{p(v)}$ and $e_{n(v)}$ are the set of positive and negative examples, respectively, where attribute a has the value v. If the examples of a node belong to one class only, the node becomes a leaf node with the respective class. If no other attributes are left, the class of a leaf node is defined by the majority vote of the associated examples.

A technique to avoid overfitting in decision tree learning is to prune the learned tree. In the experimental section we will therefore also investigate the influence of two pruning techniques, namely a simple restriction on the maximum depth of the tree (max-depth-pruning, MDP) and reduced error pruning (REP) [1]. In MDP every subtree that has its root node at a given depth of the original tree will be collapsed into a leaf node. For REP we need to divide the training data set into an induction set, which is used during induction of the decision tree, and a pruning set, which is used to evaluate which part of the tree should be pruned. REP then replaces any subtree with a leaf node if this does not lead to a higher classification error on the pruning data set.

The learned decision tree is then used to guide the search for the target product by classifying each outgoing edge of the robot's current location into promising and non-promising directions. It may happen that more than one edge will be classified as a promising direction. In this case, we choose randomly among the promising candidates.

V. Experimental Evaluation

The supermarket data was collected in real supermarkets. Three of the supermarkets were used as a training set for learning the decision trees and the fourth for evaluating the search strategies. We first introduce two random strategies – a random walk and a random exploration strategy – and then discuss some variants of our proposed strategy based on decision trees. Next, we explain how the the data of the field study with human participants has been assessed and finally provide quantitative results of a comparison of all of these strategies.

A. Random Walk and Random Exploration Strategies

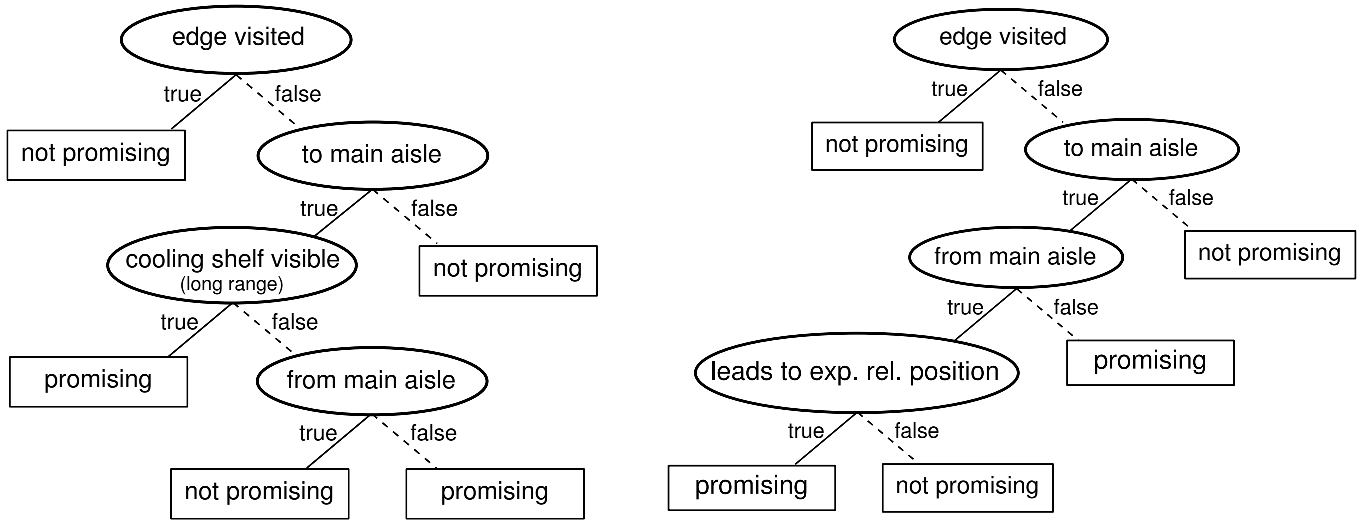
We consider two random strategies of which the first one is a plain random walk. The random walk strategy serves as a lower bound in the comparison – any reasonable strategy should do better than a random walk. The second strategy is a random exploration strategy that can be considered as an improved random walk which avoids already visited edges by randomly selecting only among the unvisited edges. If all outgoing edges of a node have been visited already, an edge will be chosen that leads to the nearest node with at least one unvisited edge. If a search technique does not perform better than a random exploration technique, it obviously is not able to utilize domain-specific information, which is the ambition of our strategy.

B. Variants of the Decision Tree Strategy

In total, we evaluate five variants of our proposed strategy based on decision trees. The first four variants differ by the set of attributes they are allowed to use. We start from a simple variant, which uses only subset 'a' of the attributes (see Table I), while the three subsequent variants can use increasingly more attributes (including subsets 'b', 'c', and 'd'). The resulting decision trees are not pruned in any way and therefore might be prone to overfitting. We therefore also investigate the influence of two pruning techniques. We tried several alternatives by restricting the maximum depth of the trees to different levels (MDP) or by applying reduced error pruning (REP) or a combination of both to any of the four attribute variants. We found the best variant to be a combination of both pruning techniques applied to a tree that uses the full set of attributes. We first applied MDP using a maximum depth of four and then additionally applied REP. To do so, the training data set was split into an induction set (75% of the data) and a pruning set (25% of the data). Two examples of learned and pruned decision trees can be seen in Fig. 3.

C. Field Study with Human Participants

A field study involving 38 human participants was conducted in a real supermarket [6]. The participants had to find the same 15 products that we used as target products. As the supermarket in which the study took place is the same market that we used as a model for our evaluation market, we can directly compare the path distances of the human participants to the path distances traveled by the robot in the simulated environment. In order to assure that we have a metrically comparable model of the real market, we first built an occupancy grid map of the supermarket using a laser-based FastSLAM implementation [4] and then placed the shelves according to the grid map, as can be seen in Fig. 1. The product placement in our virtual market also resembles the



(a) Decision tree for finding yogurt.

(b) Decision tree for finding UHT milk.

Fig. 3: Two examples of pruned decision trees that have been learned from optimal search paths in the training supermarkets. The trees use the attribute variant (a,b,c,d,pruned) mentioned in Table II.

product locations in the real market. The participants were tracked using a RFID-based localization technique [5] and the resulting trajectories were than mapped upon the route graph for a fair comparison with the path distances of the simulated robot.

D. Evaluation of the Different Strategies

The human participants had to find the 15 products in a given order, and so the location of a found target product will be the starting location for the search for the next target product. Therefore, each target product is associated with a certain starting location and we evaluate the *simulated* wayfinding strategies for the same 15 pairs of starting location and target product.

As a performance measure we consider the length of a complete search path, that is the path length of a search for all 15 products. We simulated 5000 search trials for the random strategies and 1000 search trials for strategies based on the decision trees. We only have a sample size of 26 complete searches of the human participants, because some search sub-trials (for a single product) have been cancelled if the search took too long or the participants gave up. This introduces a slight bias to the comparison for the benefit of the human participants, because the *simulated* search trials were not cancelled if they took "too long". Nevertheless, we think that the available data of the human search paths still constitutes a usable basis for a comparison.

In Table II we plot the mean and standard deviation of the search path lengths. We performed a one-tailed paired t-test[1] and found all improvements indicated by the means to be significant at the 0.01 level, except for the difference between the decision trees with attribute combinations (a) and (a,b,c).

[1] If the sample sizes differed, we used the sample size of the smaller sample. We also applied Welch's t-test, which is applicable for unequal sample sizes and unequal variances, and got the same results regarding the statistical significance at the 0.01 level.

TABLE II: Mean and standard deviation (SD) of the overall search path lengths for different search strategies. For further comparison we list the length of the optimal path and the path length ratio defined as the average path length of a strategy divided by the length of the optimal path.

Strategy	Search Path Length		Ratio	Samples
	Mean (km)	SD (km)		
Random Walk	7.559	2.290	30.6	5000
Random Exploration	1.953	0.299	7.9	5000
Dec. Tree (a)	1.609	0.263	6.5	1000
Dec. Tree (a,b)	1.425	0.193	5.8	1000
Dec. Tree (a,b,c)	1.620	0.257	6.6	1000
Dec. Tree (a,b,c,d)	1.717	0.238	7.0	1000
Dec. Tree (a,b,c,d,pruned)	1.176	0.211	4.8	1000
Human Participants	0.565	0.110	2.3	26
(Optimal Path)	(0.247)	–	1.0	–

As expected, the random walk yields extremely long search paths that are on average 30.6 times longer than the optimal path. The random exploration strategy already yields much better search paths that are on average 7.9 times longer than the optimal path. This can be further improved to a ratio of 5.8 when the search is guided by our proposed strategy based on the unpruned decision trees. We get even better results, reducing the ratio to 4.8, when we use pruned decision trees. This seems to suggest that the unpruned decision trees overfit the data of the three training supermarkets. Humans achieved the best results with a ratio of 2.3.

Though we did not achieve the same performance as humans, the results clearly indicate, that the utilization of wayfinding heuristics in the form of decision trees leads to significantly shorter search paths when compared to uninformed wayfinding strategies. The random exploration strategy performed significantly worse than our approach, because it is not able to take domain-specific background knowledge into account, which is the advantage of our proposed technique.

VI. Conclusions

We proposed a wayfinding strategy which is guided by heuristic rules and that is able to find a product in a supermarket significantly faster than a standard search strategy. We furthermore showed how such heuristics can be learned from data by observing optimal search behavior in a training set of supermarkets. We then applied and evaluated these heuristics in a previously unseen supermarket. Additionally, we compared our technique to the performance of human participants that took part in a field study in a real supermarket.

The results of our experiments demonstrated that our technique yields search paths that are on average 4.8 times longer than the optimal path, while an exploration strategy only achieves a ratio of 7.9 times the optimal path length. The exploration strategy is an example of a standard uninformed search strategy which is not able to exploit domain-specific background knowledge. The significantly shorter path lengths of our strategy highlights the benefits of utilizing the semantics of the robot's local environment when searching for the target object.

The proposed strategy depends only on local information, which is directly accessible from the robot's current location, such as the visible products, product categories, shelf types, etc. Thus, no book-keeping efforts are necessary except for a recognition of previously traversed edges. This makes the proposed strategy not only effective, but also very efficient.

Acknowledgment

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under contract number SFB/TR 8 Spatial Cognition (R6-[SpaceGuide]). The field study involving human participants in the real supermarket was carried out by Christopher Kalff and colleagues of the Centre for Cognitive Science at the University of Freiburg, Germany.

References

[1] Leonard A. Breslow and David W. Aha. Simplifying decision trees: A survey. *The Knowledge Engineering Review*, 12(1):1–40, 1997.

[2] Alexandru Cocora, Kristian Kersting, Christian Plagemann, Wolfram Burgard, and Luc De Raedt. Learning relational navigation policies. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.

[3] Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008.

[4] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. on Robotics*, 23(1):34–46, 2007.

[5] Dominik Joho, Christian Plagemann, and Wolfram Burgard. Modeling RFID signal strength and tag detection for localization and mapping. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3160–3165, Kobe, Japan, 2009.

[6] Christopher Kalff and Gerhard Strube. Where is the fresh yeast? The use of background knowledge in human navigation. In *Spatial Cognition 2008: Poster Presentations*, pages 17–20, Freiburg, Germany, 2008.

[7] Thomas Kollar and Nicholas Roy. Utilizing object-object and object-scene context when planning to find things. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2168–2173, Kobe, Japan, 2009.

[8] Vladimir Kulyukin, Chaitanya Gharpure, John Nicholson, and Grayson Osborne. Robot-assisted wayfinding for the visually impaired in structured indoor environments. *Autonomous Robots*, 21(1):29–41, 2006.

[9] Benson Limketkai, Lin Liao, and Dieter Fox. Relational object maps for mobile robots. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2005.

[10] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[11] Ananth Ranganathan and Frank Dellaert. Semantic modeling of places using objects. In *Proc. of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[12] David Silver, J. Andrew Bagnell, and Anthony Stentz. High performance outdoor navigation from overhead data using imitation learning. In *Proc. of Robotics: Science and Systems (RSS)*, 2008.

[13] Philip A. Titus and Peter B. Everett. Consumer wayfinding tasks, strategies, and errors: An exploratory field study. *Psychology and Marketing*, 13(3):265–290, 1996.

[14] Shrihari Vasudevan and Roland Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 56(6):522–537, 2008.

[15] Hendrik Zender, Oscar Martínez Mozos, Patric Jensfelt, Geert-Jan M. Kruijff, and Wolfram Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.

[16] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, 2008.