A Generalizing Spatial Representation for Robot Navigation with Reinforcement Learning

Lutz Frommberger
Universität Bremen
SFB/TR 8 Spatial Cognition, Project R3-[Q-Shape]
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
lutz@informatik.uni-bremen.de

Abstract

In robot navigation tasks, the representation of the surrounding world plays an important role, especially in reinforcement learning approaches. This work presents a qualitative representation of space consisting of the circular order of detected landmarks and the relative position of walls towards the agent's moving direction. The use of this representation does not only empower the agent to learn a certain goal-directed navigation strategy, but also facilitates reusing structural knowledge of the world at different locations within the same environment. Furthermore, gained structural knowledge can be separated, leading to a generally sensible navigation behavior that can be transferred to environments lacking landmark information and/or totally unknown environments.

Introduction

In goal-directed navigation tasks, an autonomous moving agent has solved its task when having reached a certain location in space. Reinforcement Learning (RL) is frequently applied to such tasks, because it allows an agent to autonomously adapt its behavior to a given environment. It has proven to be an effective approach especially in conditions of uncertainty. However, in large and in continuous state spaces RL methods require extremely long training times.

The navigating agent learns a strategy that will bring it to the goal from every position within the world, that is: It learns to select an action for every given observation of the given environment. But usually this strategy cannot be transferred to other scenarios, because knowledge of the underlying structure of the state space is not explicitly acquired. The agent lacks an *understanding* of geometrical spaces.

Thrun and Schwartz claim that it is necessary to discover the structure of the world and abstract from its details to be able to adapt RL to more complex tasks (Thrun & Schwartz 1995). Lane and Wilson argue that navigation tasks in a spatial environment possess a certain structure, which proves to be advantageous during the learning process (Lane & Wilson 2005). The aim of the approach we present in this paper is to enable the agent to profit from this structure by using an appropriate *qualitative* representation for it. While other approaches often concentrate on the design of the agents' actions or the internal representation of the value function, the approach presented in this paper applies abstraction directly on the sensory data.

The first goal of this work is to provide a spatial representation that leads to a small and discrete state space which enables fast and robust learning of a navigation strategy in a continuous, non-homogeneous world. The second goal is to explicitly model structural elements of the environment within this representation to enable the agent to reuse learned strategies in structurally similar areas within the same world and also being able to transfer learned strategies to other, unknown environments.

This paper is organized as follows: First, we introduce the robot navigation scenario used in this work. Then we discuss different aspects of goal-directed navigation behavior and introduce a qualitative representation of space consisting of the relative position of detected landmarks and surrounding line segments. In the following, experimental results prove that the proposed representation induces fast and stable learning of a policy that can also be transferred to unknown environments. After an overview of related work, this paper closes with a summary and an outlook.

The Navigation Task

The task considered within this work is a goal-directed navigation task: An autonomous robot is requested to find a certain location in a simplified office environment (see Figure 1). At the start of the experiment, the world is completely unknown to the agent—no map is given and no other information is provided. The agent is supposed to be capable to determine unique landmarks around it to identify its location. In our experimental setup this requirement is idealized: The goal finding task takes place in a simulated environment which consists of line segments that represent walls. It is assumed that every wall is uniquely distinguishable, making the whole wall a landmark of its own. To represent this, each wall is considered to have a unique color.

The robot is capable of performing three different basic actions: moving forward and turning a few degrees both to the left and to the right. Both turns include a small forward movement; and some noise is added to all actions. There is no built-in collision avoidance or any other navigational intelligence provided. The robot is assumed to be able to perceive walls around it within a certain maximum range. The goal of the agent is to "find" a certain location within

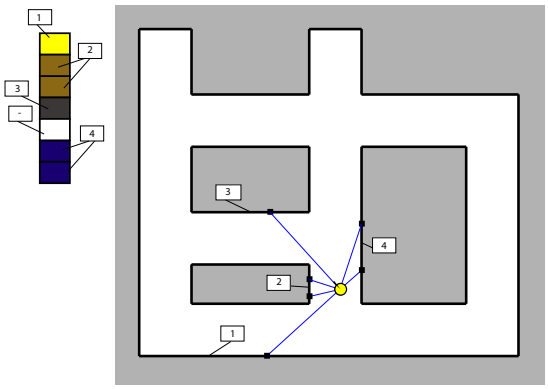


Figure 1: The navigation task: a robot in a simplified simulated office environment with uniquely distinguishable walls. The lines departing from the robot visualize landmark scans. Detected colors are depicted in the upper left. The label "–" means that nothing was perceived within the agent's scanning range. The target location is the right dead end.

the environment and drive towards it.

The given scenario can be formalized as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$ with a continuous state space $\mathcal{S} = \{(x, y, \theta), x, y \in \mathbb{R}, \theta \in [0, 2\pi)\}$ where each system state is given by the robot's position (x, y) and an orientation θ, an action space $\mathcal{A}$ consisting of the three basic actions described above, a transition function $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denoting a probability distribution that the invocation of an action a at a state s will result in a state s', and a reward function $R : \mathcal{S} \rightarrow \mathbb{R}$, where a positive reward will be given when a goal state $s^* \in \mathcal{S}$ is reached and a negative one if the agent collides with a wall. The goal of the learning process within this MDP is to find an optimal policy $\pi : \mathcal{S} \rightarrow A$ that maximizes the reward the agent receives over time.

Applying reinforcement learning on this MDP is anything but trivial. The state space is continuous, resulting in the need to use function approximation to represent the value function. Due to the inhomogeneity of the state space at the position of walls, this approximation is crucial. Furthermore, the the pose of the agent is usually impossible to detect correctly in realistic systems, and the state representation (x, y, θ) is not agent-centered: A learned policy for this MDP will be worthless when applied to an environment that is, e.g., just mirrored. The aim of this work is to find a representation that enables to learn a policy that is applicable to unknown environments as well, resulting in a generally sensible spatial behavior. To achieve that, we concentrate on the agent's *observation* of the environment: A function $\psi : \mathcal{S} \rightarrow \mathcal{O}$ assigns an observation o to every state s. This results in a Partially Observable Markov Decision Process (POMDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, R \rangle$ with $\mathcal{O} = \{\psi(s), s \in \mathcal{S}\}$ being the set of all possible observations in $\mathcal{S}$. We now use this POMDP to approximate the underlying MDP, i.e., we solve the POMDP as if it was an MDP. The quality of the resulting policies clearly depends on how closely ψ can represent the structure of the underlying state space.

Due to the usually non-injective nature of ψ, the execution of an action can result in the same observation $o \in \mathcal{O}$ before and after the action, which is not desirable when using RL. To prevent this, we define a qualitative action behavior: A *qualitative action* is a sequence of identical basic actions $a \in \mathcal{A}$ that lead from a given observation to a different one, i.e., a basic action is repeated as long as the observation remains the same. Let $\mathcal{A}_q$ be the set of qualitative actions, then $T(o, a, o) = 0 \; \forall o \in \mathcal{O}, a \in \mathcal{A}_q$.

Representing General and Task-Specific Spatial Knowledge

To achieve a valuable observation representation, we take a closer look at the given problem. Navigation in space, as performed in the learning examples, can be viewed as consisting of two different aspects:

Goal-directed behavior towards a certain target location depends highly on the task that has to be solved. If the task is to go to a certain location, the resulting actions at a specific place are generally different for different targets. Goal-directed behavior is task-specific. *Generally sensible behavior* regarding the structure of the environment is more or less the same in structurally similar environments. It does not depend on a goal to reach, but on structural characteristics of the environment that invoke some kind of behavior. Generally sensible behavior is task-independent. This distinction closely relates to Konidaris' concepts of *problem-space* and *agent-space* (Konidaris 2006).

Both aspects are not completely independent: A generally sensible spatial behavior does not need a target location, but the other way round it is different: Reaching a target location requires some sort of generally sensible navigation behavior (otherwise the target would not have been reached). Put differently, knowledge of generally sensible navigation behavior is a good foundation for developing goal-oriented strategies. Thus, it is desirable to be able to extract this behavior from the strategy. The aim is to find a representation that divides between the two aspects of navigation behavior in order to be able to single out the general navigation knowledge in a reusable way.

To represent the necessary knowledge to achieve a goal-directed behavior, we define a function $\psi_a : \mathcal{S} \rightarrow \mathbb{N}^n$ which maps the agent's position and orientation (x, y, θ) to a circular order of perceived landmarks. In the given setting this can be realized by the color information detected at n discrete angles around it, resulting in a vector $c = \psi_a(s) = (c_1, \ldots, c_n)$. Every physical state $s \in \mathcal{S}$ maps to exactly one color vector c. This is a compact and discrete *qualitative abstraction* of rich and continuous real world information.

The encoding of a circular order of perceived colors is sufficient to approximately represent the position of the agent within the world but it does not represent any information about the agent's position regarding the obstacles.

A Spatial Representation of Relative Position of Line Segments

As a generally sensible behavior in office environments is affected by the walls, which induce sensible paths inside the

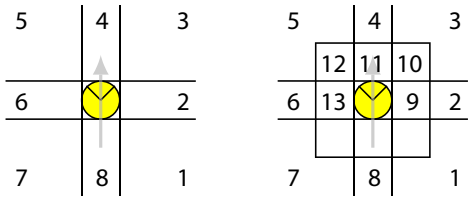
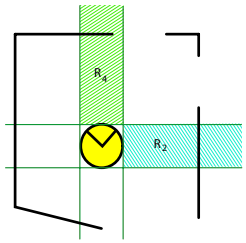


Figure 2: Neighboring regions around the robot in relation to its moving direction. Note that the regions in the immediate surroundings (b) are proper subsets of $R_1, \ldots, R_8$ (a).



$\overline{\tau}(R_1) = 1 \quad \overline{\tau}'(R_1) = 1$
$\overline{\tau}(R_2) = 1 \quad \overline{\tau}'(R_2) = 1$
$\overline{\tau}(R_3) = 3 \quad \overline{\tau}'(R_3) = 0$
$\overline{\tau}(R_4) = 1 \quad \overline{\tau}'(R_4) = 1$
$\overline{\tau}(R_5) = 2 \quad \overline{\tau}'(R_5) = 1$
$\overline{\tau}(R_6) = 1 \quad \overline{\tau}'(R_6) = 1$
$\overline{\tau}(R_7) = 2 \quad \overline{\tau}'(R_7) = 1$
$\overline{\tau}(R_8) = 1 \quad \overline{\tau}'(R_8) = 0$

Figure 3: Example: RLPR values in an example situation. Region R_2 (right) and R_4 (front) are marked.

world, their whereabouts have to be taken into account. In the following we describe a function $\psi_b : \mathcal{S} \rightarrow \mathbb{N}^n$ that maps a system space to an extremely compact representation of the relative positions of lines towards the agent's moving direction. For further reference, it is called *RLPR* (Relative Line Position Representation).

To encode RLPR, we construct an enclosing box around the robot and then extend the boundaries of this box to create eight disjoint regions R_1 to R_8 (see Figure 2a). This representation was proposed to model the movement of extended objects in a qualitative manner (Mukerjee & Joe 1990). The representation used in this work is a modified version of the *direction-relation matrix* (Goyal & Egenhofer 2000). We define a *traversal status* $\tau(B, R_i)$ of a line segment B regarding a region R_i as follows:

$$\tau(B, R_i) = \begin{cases} 1 & B \cap R_i \neq \emptyset \\ 0 & \text{else} \end{cases} \quad (1)$$

$\tau(B, R_i)$ is 1 if a line B cuts region R_i and 0 if not. The overall number of lines in a region R_i therefore is

$$\overline{\tau}(R_i) = \sum_{B \in \mathcal{B}} \tau(B, R_i) \quad (2)$$

with $\mathcal{B}$ being the set of all detected line segments.

For anticipatory navigation, it is particularly interesting also to know where the walls lead to, i.e., which line segments span from one region to another. To additionally encode this, we determine if a line B lies within counterclockwise adjacent regions R_i and R_{i+1} (for R_8, of course, we need to consider R_1):

$$\tau'(B, R_i) = \tau(B, R_i) \cdot \tau(B, R_{i+1}) \quad (3)$$

$\tau'(B, R_i)$ is also very robust to noisy line detection, as it does not matter if a line is detected as one or more segments. The overall number of spanning line segments in a region, $\overline{\tau}'(R_i)$, is derived analogously to (2). Figure 3 shows an example situation.

Special care has to be taken on the immediate surroundings of the agent. The position of detected line segments is interesting information to be used for general orientation and mid-term planning, but obstacles in the immediate surroundings are to be avoided in the first place. So the representation described above is used twice. On the one hand, there are the regions $R_1, \ldots, R_8$ that are bounded by the perceptual capabilities of the robot. On the other hand, bounded subsets of those regions represent the immediate surroundings (see Figure 2b). The size of the grid defining the immediate surroundings is given a-priori. It is a property of the robot and depends on its size and system dynamics (e.g., speed).

For inducing an appropriate behavior in the immediate surroundings ($R_9, \ldots$), it is sufficient to determine if there is an object or not. So while regarding $\overline{\tau}'(R_i)$ for $R_1, \ldots, R_8$, we use $\overline{\tau}(R_i)$ for the immediate surroundings. Also, the regions in the back of the robot are omitted, because the agent cannot move backwards:

$$\psi_b(s) = (\overline{\tau}'(R_1), \ldots, \overline{\tau}'(R_6), \overline{\tau}(R_9), \ldots, \overline{\tau}(R_{13})) \quad (4)$$

Often it is sufficient to distinguish if $\overline{\tau}'(R_i)$ or $\overline{\tau}(R_i)$ equals 0 or not, resulting in $\psi'_b : \mathcal{S} \rightarrow \{0, 1\}^n$. For the experiments in this work, we used ψ'_b.

To combine knowledge about goal-directed and generally sensible spatial behavior, we now build a feature vector by concatenating the representation of detected colors and RLPR (color-RLPR), so the observation space is

$$\mathcal{O} = \{(\psi_a(s), \psi'_b(s))\} \quad (5)$$

This observation space is discrete. Its size $|\mathcal{O}|$ can be approximated by an upper bound. Given the representation in (5) and a number of 7 color scans, $|\mathcal{O}| \leq (C + 1)^7 \cdot 2^{11}$ with C being the number of colors perceived. For $C = 20$, $|\mathcal{O}|$ is bigger than $3 \cdot 10^{12}$. However, a large number of theoretically possible combinations has no realization in the real word. The RLPR part enhances the size by a factor of just 2048, independent of the size of the environment. While learning the task in Figure 1 with color-RLPR, the agent encountered only about 550,000 different observations.

GRLPR: Generalizing RLPR

To achieve a generalizing behavior that abstracts from the concrete landmark information and just considers the structural information given by RLPR, we apply the function approximation method of tile coding (Sutton 1996) to the representation. We choose a tile size big enough that the whole color space of N colors can fit within one tile, so that each update of the policy affects all system states with the same RLPR representation. We choose a tile size of 1 and make sure that each color representation $c_i \in [0, 1)$ (i.e., the i-th detected color c_i is represented by $\frac{i-1}{N}$). With this encoding, all colors can be stored within one tile, and no function approximation is applied to the RLPR part of the representation. To still be able to differentiate between different colors,

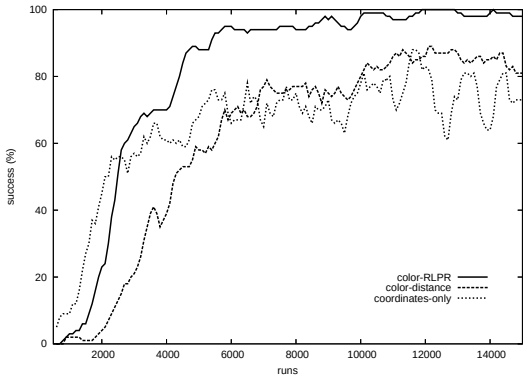


Figure 4: Number of runs reaching the goal state: Both the coordinate and the color-distance approach don't show a stable and successful learning, while color-RLPR is both fast and successful.

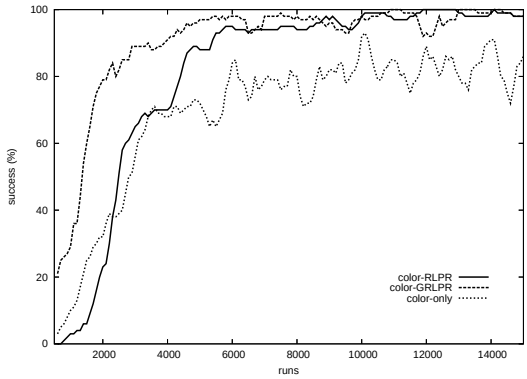


Figure 5: GRLPR shows a very fast learning especially in the early training phase. The color-only approach does not lead to a stable learning behavior.

we also choose N tilings. As an effect, the agent can reuse structural knowledge acquired within the same learning task. Furthermore, the learned policy is immediately applicable to new environments even if they are completely unknown and no distinguishable landmarks are present in the new worlds or none that have ever been perceived before. The only thing to assure is that perceived colors in the new world are not assigned numbers that have been used in the training task. This generalizing variant of RLPR is called *Generalizing RLPR* (GRLPR).

Experimental Results

All experiments have been conducted using Watkins' Q(λ) algorithm (Watkins 1989). During training, the agent uses an ϵ-greedy policy with $\epsilon = 0.15$. This means, that at each time step the agent performs a random action with a probability of ϵ, otherwise it executes the action a which yields the highest rating according to $Q(o, a)$ ($o \in \mathcal{O}$, $a \in \mathcal{A}$). A positive reward is given when the agent reaches the target location, a negative reward is given when the agent collides with a wall. Test runs without random actions (and without learning) have been performed after every 100 training episodes. A step size of $\alpha = 0.2$, a discount factor of $\gamma = 0.98$, and $\lambda = 0.9$ was used in all the trials. A learning episode ends when the agent reaches the goal state, collides with a wall, or after a certain number of actions.

Goal Finding Performance

In a first experiment, the robot has to solve the goal finding task in the environment depicted in Figure 1. The agent starts from 20 starting positions equally distributed inside the corridors. We test color-RLPR and -GRLPR against a color-only representation given by ψ_a only, and two non- or semi-qualitative representations: First, a coordinate-only representation of the original MDP with $s = (x, y, \theta)$, and second, a color-distance representation consisting of the color vector $\psi_a(s)$ and a number of distance values to the nearest obstacles acquired at n discrete angles around the agent. The continuous parts of the state vectors are approximated using tile coding. As many different parameter values (n, angular distance, number of tilings, size of tiles) have to be tried for the metrical approaches, we only regard the best performing combinations as representatives in this section.

Figure 4 compares the learning success of color-RLPR and the color-distance and coordinate representation. With color-RLPR, the agent reaches the goal after about 10,000 learning episodes and keeps a stable success rate afterward. In contrast, both metrical approaches fail to show a stable behavior and even fail to reach 100% success within the first 15,000 learning episodes. The coordinate based approach learns fast in the beginning, but gets extremely unstable afterward. Generally, for both metrical approaches, depending on the choice of parameters, the results are either unstable, or stable, but unsuccessful.

Figure 5 shows the success graphs of the non-metric approaches. The color-only representation learns as fast as the coordinate based approach, but is also comparably unstable, even if slightly more successful. Due to the smaller observation space, it also shows a faster learning than color-RLPR in the beginning. Color-GRLPR, however, learns faster than the other two approaches in the early training phase. This indicates that GRLPR benefits from its generalizing behavior and empowers the agent to reuse structural spatial knowledge gained in already visited parts of the environment. The relatively long period of learning until reaching 100% can be explained by the contradicting nature of the given environment. From certain starting points, it is necessary to first turn right and then turn left at the same type of intersection. So if the agent once has learned to perform a certain action at the first intersection, this strategy works against reaching the goal at the second one, and further training effort is required to cope with this contradiction[1].

An important measure of the quality of navigation is the number of collisions during training (see Table 1). Compared to the coordinate and color-distance approach, this number is reduced by more than 10% in training and about 50% in the test runs when using color-RLPR. Furthermore,

[1]This contradicting nature is also the reason why learning without landmark information does not work in this world.

Representation	Collisions Training	Test
coordinates	9635	2095
color-distance	10586	1800
color-RLPR	8513	1095
color-GRLPR	3986	424
color-only	21388	2385

Table 1: Number of collisions after two trials of 15,000 episodes: RLPR approaches show fewer collisions than the metrical ones. Especially GRLPR reduces collisions significantly.

color-GRLPR performs noticeably better than RLPR. This indicates that the generalization ability leads to a sensible navigation behavior rather early. Of course the color-only approach, that does not cope with distance notions at all, shows the highest number of collisions.

Regarding the distance traveled to reach the goal is a hard issue, because an optimal path cannot be determined. The shortest path leads very close around corners and walls and, due to noise and perceptual ambiguity, frequently results in collisions. The system tries to balance out between a short and a safe path, so the shortest path will (and shall) never be learned. However, simulation trends show that the actions needed to reach the goal continuously decrease over 250,000 runs when using color-RLPR or -GRLPR.

Summed up, the use of the proposed (G)RLPR representations shows a faster and more stable learning compared to non- or semi-qualitative approaches and the number of collisions is reduced significantly. Moreover, the RLPR based approaches don't require parameter fiddling.

Generalization Capabilities

To test the generalization abilities of a learned policy with GRLPR, we must examine how the agent behaves when using the learned strategy in the absence of landmarks or in an unknown world. The knowledge gained from the environment in Figure 1, however, is not too helpful to achieve a generally sensible spatial behavior, because the goal is in a dead end near a wall, and as a result (because states near the goal have bigger impact) the agent learns to happily run into any dead end and towards walls.

So the agent was trained in a more homogeneous environment (see Figure 6, but note that also this environment requires contradicting decisions at the same type of intersection). After learning with GRLPR for 40,000 episodes, the landmark information is turned off, so that the agent perceives the same (unknown) color vector regardless of where it is. Figure 6 shows the resulting trajectories from 20 starting positions, using the strategy that took the least number of steps to the goal. The agent is able to navigate collision-free and perform smooth curves, fully exploring the environment. This generalized spatial strategy is acquired very fast: After only 200 episodes of learning, a test run without landmark information results in fairly smooth trajectories exploring all of the world, and collisions can be observed only

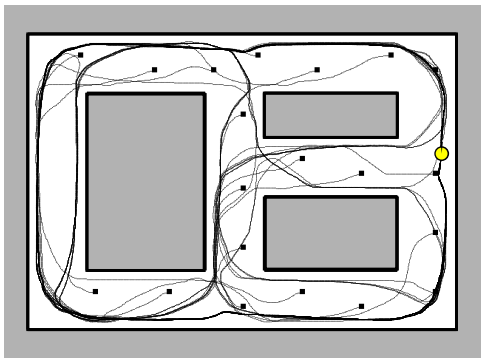


Figure 6: Trajectories of the agent in the same environment with no landmark information available. Learning was performed using GRLPR. The small dots mark starting positions. The agent shows a sensible behavior and moves smoothly and collision-free.

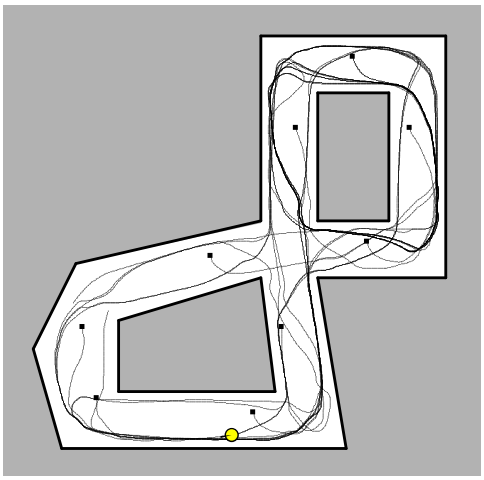


Figure 7: Trajectories of the agent in an unknown environment without landmarks, using the strategy learned in the world depicted in Figure 6 with GRLPR.

when starting from 2 out of the 20 starting positions. The learned policy can also successfully be transferred to absolutely unknown environments. Figure 7 shows the agent's trajectories in a landmark-free world it has never seen before with different corridor angles and structural elements, successfully following the strategy gained in the prior experiment without any modification.

Finally, we show that the "outer" sectors of RLPR $(R_1, \ldots, R_8)$ are essential for building a generalizing representation. When just using the sectors in the immediate surroundings $(R_9, \ldots, R_{13})$ for building the observation, the given goal-seeking task can be learned even faster than with full RLPR. In the absence of landmarks within the same world, however, the policy already fails. Because of the missing structural information, the agent's strategy restricts to collision avoidance, resulting frequently in endless turnings around itself. Moreover, the trajectories when driving around curves are longer than with full RLPR.

Related Work

Much effort has been spent to accomplish improvements regarding the training speed of reinforcement learning in navigation tasks, and consideration of the structure of the state space has been found to be an important means to reach that goal. Topological neighborhood relations can be used to improve the learning performance (Braga & Araújo 2003), but this approach requires an a-priori existence of a topological map of the environment. Thrun and Schwartz tried to to find reusable structural information that is valid in multiple tasks (Thrun & Schwartz 1995). They introduced so-called skills, which collapse a sequence of actions into one single operation. Their algorithm can only generalize over separate tasks, not over different states within the same one. Glaubius *et al.* concentrate on the internal value-function representation to reuse experience across similar parts of the state space. They use pre-defined equivalence classes to distinguish similar regions in the world (Glaubius, Namihira, & Smart 2005). Lane and Wilson describe relational policies for spatial environments and demonstrate significant learning improvements (Lane & Wilson 2005). However, their approach runs into problems when non-homogeneities such as walls and obstacles appear. To avoid that shortcoming, they also suggest regarding the relative position of walls with respect to the agent, but did not realize this approach yet. Recent work by Mahadevan & Maggioni introduces a method to autonomously construct basis functions for value function approximation based on the structure and geometry of the given problem (Mahadevan & Maggioni 2006). This is a very beneficial approach for the task that is learned, but in general the learned knowledge cannot be transferred to different environments without further effort.

In a machine learning context a landmark based qualitative representation was used for example within a multi-robot scenario (Busquets *et al.* 2002). The authors partition the world into six circular sectors and store qualitative distance information for every landmark in every sector. For navigation, however, they rely on rather complex actions, and obstacle avoidance is handled by a separate component.

Conclusion and Outlook

Solving a goal-directed robot navigation task can be learned with reinforcement learning using a qualitative spatial representation purely using the ordering of landmarks and the relative position of line segments towards the agent's moving direction. The proposed representation generates a small and discrete state space, even if the world is continuous. It results in a fast and stable learning of the given task and outperforms metrical approaches that rely on function approximation in learning success, speed, stability, and number of collisions. It also reduces the number of parameters. Structural information within the environment is made part of the state representation and can be reused within the same learning task, which facilitates a faster learning and reduces collisions significantly. Furthermore, the use of GRLPR enables to reuse knowledge gained in structurally similar parts of the world and even permits to transfer the learned strategy directly to environments lacking landmark information and/or totally unknown environments without further effort: The agent learns not only a task-dependent strategy, but acquires a generally sensible behavior in geometrical spaces. Different aspects of spatial information (landmark-based goal directed knowledge and structural knowledge about the world) are clearly separated in the representation, permitting to only regard one aspect of it.

Future work will show that the acquired strategy can be used as background knowledge for new learning tasks in unknown environments and therefore allows for speeding up learning. We will also investigate how to learn two separate policies for goal-oriented and generally sensible behavior in a hierarchical learning architecture. Strategies learned in simulation will also be ported to a real robot platform.

Acknowledgments

This work was supported by the DFG Transregional Collaborative Research Center SFB/TR 8 "Spatial Cognition". Funding by the German Research Foundation (DFG) is gratefully acknowledged.

References

Braga, A. P. S., and Araújo, A. F. R. 2003. A topological reinforcement learning agent for navigation. *Neural Computing and Applications* 12:220–236.

Busquets, D.; de Mántaras, R. L.; Sierra, C.; and Dietterich, T. G. 2002. Reinforcement learning for landmark-based robot navigation. In *Proc. of AAMAS*.

Glaubius, R.; Namihira, M.; and Smart, W. D. 2005. Speeding up reinforcement learning using manifold representations: Preliminary results. In *Proc. of the IJCAI Workshop "Reasoning with Uncertainty in Robotics"*.

Goyal, R. K., and Egenhofer, M. J. 2000. Consistent queries over cardinal directions across different levels of detail. In Tjoa, A. M.; Wagner, R.; and Al-Zobaidie, A., eds., *Proc. Workshop on Database and Expert System Applications*.

Konidaris, G. D. 2006. A framework for transfer in reinforcement learning. In *Proc. of the ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning*.

Lane, T., and Wilson, A. 2005. Toward a topological theory of relational reinforcement learning for navigation tasks. In *Proc. of FLAIRS*.

Mahadevan, S., and Maggioni, M. 2006. Value function approximation using diffusion wavelets and laplacian eigenfunctions. In *Neural Information Processing Systems*.

Mukerjee, A., and Joe, G. 1990. A qualitative model for space. In *Proc. of AAAI*.

Sutton, R. 1996. Generalization in reinforcement learning: Successful examples using sparse tile coding. In *Advances in Neural Information Processing Systems*, volume 8.

Thrun, S., and Schwartz, A. 1995. Finding structure in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 7.

Watkins, C. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, Cambridge University.