International Journal on Artificial Intelligence Tools
Vol. 17, No. 3 (2008) 465–482
© World Scientific Publishing Company

LEARNING TO BEHAVE IN SPACE: A QUALITATIVE SPATIAL REPRESENTATION FOR ROBOT NAVIGATION WITH REINFORCEMENT LEARNING*

LUTZ FROMMBERGER
SFB/TR 8 Spatial Cognition, Project R3-[Q-Shape], Universität Bremen
Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
lutz@sfbtr8.uni-bremen.de

The representation of the surrounding world plays an important role in robot navigation, especially when reinforcement learning is applied. This work uses a qualitative abstraction mechanism to create a 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 faster compared to metrical representations, but also facilitates reusing structural knowledge of the world at different locations within the same environment. Acquired policies are also applicable in scenarios with different metrics and corridor angles. 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.

1. Introduction

In goal-directed navigation tasks, an autonomous moving agent has fulfilled its mission 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. In spatial domains, we usually encounter state spaces with these properties.

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 what has the agent really learned about the world it operates in? Can it use the acquired knowledge at different locations in this environment or even in an unknown one? This depends critically on the chosen spatial representation of the world that is given to the learning system. In the worst case, all the collected knowledge can become useless, for example, when physical map coordinates are used as the system's input and the agent operates in a similar,

*A condensed version of this work was presented at FLAIRS 2007.[1]

but different environment. In this case, a completely new set of action selections will usually be necessary. If the agent wants to cope with the new world, it has to learn everything again from scratch, including basic navigation skills and collision avoidance strategies. As knowledge of the underlying structure of the state space is usually not explicitly acquired, strategies cannot be reused in other scenarios without bigger effort. The agent lacks an *understanding* of the structure 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.[2] In alignment with that, Lane and Wilson argue that navigation tasks in a spatial environment possess a certain structure, which proves to be advantageous during the learning process.[3] 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 work often concentrates on the design of the agent's actions or the internal representation of the value function, we apply a qualitative spatial abstraction directly on the sensory data. Thus, the learning mechanism is not altered and this representation can easily be combined with existing approaches addressing the same problem.

Qualitative spatial representations are an expressive means to describe relations among features in geometrical space. They bear inherent spatial knowledge about the environment, which is advantageous for a learning system: Spatial constraints are internally provided by the input representation and do not need to be acquired separately. We claim that incorporation of structural spatial knowledge to the state representation can enable the agent to develop a generally sensible behavior in space that it can reuse at different locations within the same world or in other environments. The use of this representation will also support the learning process by speeding it up and making it more robust.

The first goal of this work is to provide a qualitative 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 article is organized as follows: After an overview on related work in Sec. 2, we introduce the robot navigation scenario used in this work (Sec. 3). Different aspects of goal-directed navigation behavior are discussed in Sec. 4, and a qualitative representation of space consisting of the relative position of detected landmarks and surrounding line segments to the agent's moving direction is introduced. Experimental results described in Sec. 5 prove that the proposed representation induces fast and stable learning of a policy that also works in modified environments and can even be transferred to totally unknown worlds. This work closes with a summary and an outlook.

2. 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 are used by Braga and Araújo to improve the learning performance,[4] but this approach requires an a-priori existence of a topological map of the environment. 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.[5] Lane and Wilson describe relational policies for spatial environments and demonstrate significant learning improvements.[3] However, their approach runs into problems when non-homogeneities such as walls and obstacles appear. To avoid that shortcoming, they suggest regarding the relative position of walls with respect to the agent, but did not realize this approach yet. Recent work by Mahadevan and Maggioni introduces a method to autonomously construct basis functions for value function approximation based on the structure and geometry of the given problem.[6] 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.

Thrun and Schwartz tried to to find reusable structural information that is valid in multiple tasks.[2] 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. Several other approaches have recently been developed for transferring learned knowledge from one task to another,[7,8] but all of these require an explicit transfer process, while the presented work just enables this property by the choice of the state space representation.

In a machine learning context a landmark based qualitative representation was used for example by Busquets et al. within a multi-robot scenario.[9] 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. Qualitative representations of space also play an important role in Kuipers' Spatial Semantic Hierarchy.[10]

3. 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 Fig. 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. This type of setup was previously used to demonstrate a qualitative navigation strategy, which needs the a-priori knowledge of the circular sequence of the colored walls and a separate obstacle avoidance, but does not rely on distance information.[11]

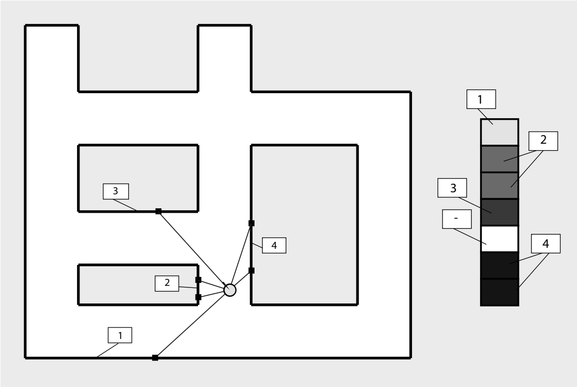


Fig. 1. The navigation task: a robot in a simplified simulated office environment with uniquely distinguishable walls. The lines departing from the robot visualize landmark scans. Detected colors are depicted at the right. The label "–" means that nothing was perceived within the agent's scanning range (the corresponding scan is not drawn). The target location is the right dead end.

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. The turns include a small forward movement; and a small amount of 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 (in this work, a maximum range of 20 times the robot's size is assumed). The goal of the agent is to "find" a certain location within 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 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 not 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 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, for example, just mirrored. So instead of coordinates we concentrate on the agent's *observation* of the environment: A function $\psi : \mathcal{S} \rightarrow \mathcal{O}$ assigns an agent-centered 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 non-injective nature of ψ, the execution of an action can result in the same observation $o \in \mathcal{O}$ before and after its execution, 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, that means, a basic action is repeated as long as the perceived observation $o = \psi(s)$ remains the same. Let $\mathcal{A}_q$ be the set of qualitative actions, then $T(o, a, o) = 0 \ \forall o \in \mathcal{O}, a \in \mathcal{A}_q$.

4. 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:

(i) *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.

(ii) *Generally sensible behavior* regarding the structure of the environment is more or less the same in structurally similar surroundings. A generally sensible behavior in indoor office environments, for example, includes not to crash into walls, avoid superfluous movements, and turn around corners smoothly. 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*.[12]

Both aspects are not completely independent: 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. For example, it helps exploring the environment. Vice versa, there must have been a sensible spatial behavior in every successful goal finding 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 from the overall strategy in a reusable way to prevent the agent from having to learn it again and again within the same or in every new goal finding task in an arbitrary environment.

4.1. *Representing task-specific knowledge*

To represent the necessary knowledge to achieve a goal-directed behavior, an agent-centered representation is appropriate. We define a function $\psi_a : \mathcal{S} \rightarrow \mathbb{R}^n$ which maps the agent's position and orientation (x, y, θ) to a circular order of perceived landmarks. Such representations are, for example, described in the work of Schlieder.[13] In the given setting this can be realized by the color information detected at n discrete angles around the agent, 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, but this mapping is not injective: Every observation c refers to a certain region in the three-dimensional physical position-orientation space. Multiple physical states share the same representation c.

The encoding of a circular order of perceived colors is sufficient to approximately represent the position of the agent within the world and to derive a sequence of actions to reach the goal state. However, it is not sufficient to prevent the robot from collisions. As stated above, the mapping from physical locations to the state representation is not unique, and given the same system input, the consequences of an applied action can differ dramatically and prevent stable learning. This problem is known as *perceptual aliasing.*[14] Every abstraction bears that danger—the less comprehensive the input, the bigger the danger of experiencing perceptual aliasing. It can be shown that the use of eligibility traces permits learning of some problems with perceptual aliasing,[15] but in the scenarios presented in this work the use of eligibility traces alone always resulted in unstable learning behavior with a high risk of collisions, because the same action at the same system state will sometimes result in a collision and sometimes not (see Fig. 2). The representation used so far encodes the agent's position quite well, but does not encode any information about the agent's relation regarding the obstacles, and it does not take the structure of the world into account. It does not support the agent in developing a generally sensible spatial behavior. See Sec. 5 for experimental results using this "color-only" representation.

4.2. *A spatial representation of relative position of line segments*

In an office environment, the relations of walls towards each other induce sensible paths inside the world which the agent should learn to follow. To achieve this, it must

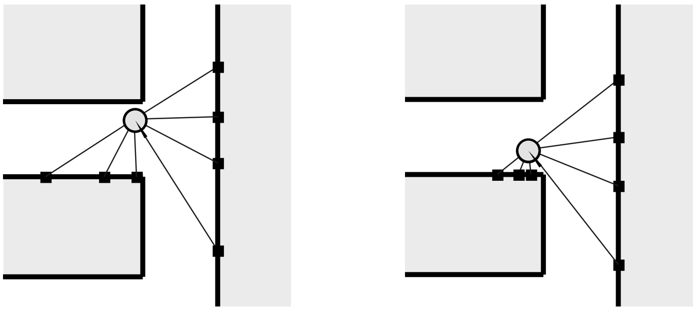


Fig. 2. Perceptual aliasing: The same state representation, different consequences. While receiving the same sequence of detected walls as sensory input, only the right situation will result in a collision when moving forward at the next time step.

be aware where the walls around it are located. 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 the relative positions of certain entities regarding the agent's position and orientation, 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 Fig. 3a). This partition was proposed before to model the movement of extended objects in a qualitative manner.[16]

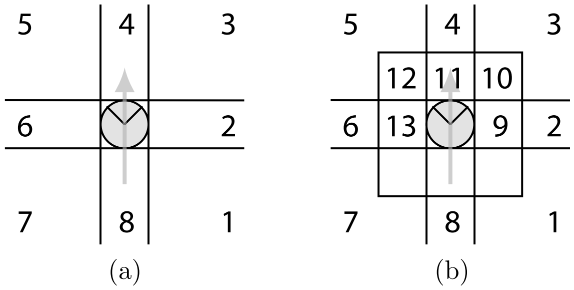


Fig. 3. Neighboring regions around the robot in relation to its moving direction. Note that the regions in the immediate surroundings (b) are proper subsets of $R_1, \ldots, R_8$ (a).

The representation used in this work is a modified version of the *direction-relation matrix.*[17] It is originally defined as

$$dir_{RR}(A, B) = \begin{pmatrix} \mathrm{NW}_A \cap B & \mathrm{N}_A \cap B & \mathrm{NE}_A \cap B \\ \mathrm{W}_A \cap B & 0_A \cap B & \mathrm{E}_A \cap B \\ \mathrm{SW}_A \cap B & \mathrm{S}_A \cap B & \mathrm{SE}_A \cap B \end{pmatrix} \qquad (1)$$

where in our case A is the robot, B is another extended object (in our case a line segment), and NW_A etc. the regions around A. They are named after the cardinal directions: NW, for example, refers to North-West. In Fig. 3, NW_A corresponds to region R_5, N_A to region R_4, and so on. R_0, corresponding to 0_A, is the middle region—the location of the robot. We now define an *overlap status* $\tau(B, R_i)$ of a line segment B regarding a region R_i as follows:

$$\tau(B, R_i) = \begin{cases} 1 & B \cap R_i \neq \emptyset \\ 0 & \text{else} \end{cases} . \tag{2}$$

$\tau(B, R_i)$ 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

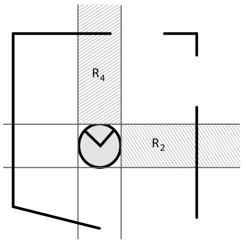
$$\overline{\tau}(R_i) = \sum_{B \in \mathcal{B}} \tau(B, R_i) \tag{3}$$

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

The spatial structure of a scenario is given by the configuration of line segments. $t(B, R_i)$ tells us where a line segment B is located, but not where it leads to. But the latter information is particularly interesting for anticipatory navigation. A corridor with a left turn, for example, will have connected line segments in all the right and front sectors, but none in freely accessible space. To additionally encode if a line segment B spans from one sector to another, we determine if a line B lies within counter-clockwise adjacent regions R_i and R_{i+1} (for R_8, of course, we need to consider R_1)[a]:

$$\tau'(B, R_i) = \tau(B, R_i) \cdot \tau(B, R_{i+1}) \tag{4}$$

$\tau'(B, R_i)$ 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 Eq. (3). Figure 4 shows an example situation.



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

Fig. 4. Example: RLPR values in an example situation. Region R_2 (right) and R_4 (front) are marked.

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

[a] There is also the possibility of a line being in two non-adjacent regions, if it also cuts the middle sector R_0. Due to minimal impact to the overall result, this is not regarded within this work.

orientation and mid-term planning. But the agent is supposed to react differently when detecting features in its immediate surroundings: If an object is detected there, this information refers to obstacle avoidance and requires a prompt reaction. It can also be utilized to realize certain behaviors, for example, a wall following strategy. So the representation described above is used twice. On the one hand, there are the regions $R_1, \ldots, R_8$ that are limited by the perceptual capabilities of the robot only. On the other hand, bounded subsets of those regions represent the immediate surroundings (see Fig. 3b). The size of the grid defining the immediate surroundings is given a-priori. It is a property of the agent and depends on its size and system dynamics (for example, the robot's maximal speed).

For inducing an appropriate behavior in the immediate surroundings, it is sufficient to determine if there is an object within these regions or not. So while regarding $\overline{\tau}'(R_i)$ for $R_1, \ldots, R_8$, we use $\overline{\tau}(R_i)$ for the $R_9, \ldots, R_{13}$. Also, the regions in the back of the robot are omitted, because the agent cannot move backwards. So the overall representation for agent-space is

$$\psi_b(s) = (\overline{\tau}'(R_1), \ldots, \overline{\tau}'(R_6), \overline{\tau}(R_9), \ldots, \overline{\tau}(R_{13})) \quad . \tag{5}$$

Often it is also sufficient to distinguish if there is at least one obstacle in a certain region or not, i.e., if $\overline{\tau}'(R_i)$ or $\overline{\tau}(R_i)$ equals 0 or not. We express this with a function $\psi'_b : \mathcal{S} \rightarrow \{0,1\}^n$. For all 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

$$\mathcal{O} = \{\psi(s)\} = \{(\psi_a(s), \psi'_b(s))\} \quad . \tag{6}$$

Obviously, this observation space is discrete. Its size $|\mathcal{O}|$ can be approximated by an upper bound. Given the representation in Eq. (6) and a number of seven color scans, $|\mathcal{O}| \leq (C+1)^7 \cdot 2^{11}$ with C being the number of colors perceived. For $C = 22$ (as in the world in Fig. 1), $|\mathcal{O}|$ is almost $7 \cdot 10^{12}$. However, a large number of theoretically possible combinations has no realization in the real word. The RLPR part contributes to $|\mathcal{O}|$ with a factor of just $2^{11} = 2048$, independent of the size of the environment. While learning the task in Fig. 1 with color-RLPR, the agent encountered only about 550,000 different state-action pairs. This is an easily manageable size when using a hashing table.

4.3. *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 want to ensure that the given reinforcement signal for a state-action pair $(\psi(s), a)$ also has an impact on all state-action pairs with the same RLPR part $\psi'_b(s)$, even for yet unknown landmark representations $\psi_a(s)$. Thus, we apply the function approximation method of tile coding (also known as CMACs)[18] to $\psi_a(s)$. Let N be the number of different colors we expect to perceive. We represent the i-th detected color c_i by $\frac{i-1}{N}$, so that $c_i \in [0, 1)$ for all i. If we now choose a tile size of 1 (in each dimension), all color values c_i can be stored in one single tile. Because $\overline{\tau}$ and $\overline{\tau}'$ both map to $\mathbb{N}$ and therefore each value of $\overline{\tau}$ and $\overline{\tau}'$ always maps to a different tile, no function approximation is applied to the RLPR part of the observation. To still be able to differentiate between different colors, we also choose N different tilings. As a result, each update of the policy also affects all system states with the same RLPR representation. This generalizing variant of RLPR is called *Generalizing RLPR* (GRLPR).

With GRLPR the agent can reuse structural knowledge acquired from previous observations 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 (see Sec. 5.2 for experimental results). 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, and, of course, this environment has also to be structured by line segments.

For the representation of the value function $Q(o, a)$, tile coding with 25 tilings is chosen when using GRLPR in this work ($o \in \mathcal{O}$ is the state representation according to Eq. (6) and $a \in \mathcal{A}_q$ is a qualitative action).

5. Experimental Results

All experiments have been conducted using Watkins' Q(λ) algorithm.[19] 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 $Q(o, a)$ ($o \in \mathcal{O}$, $a \in \mathcal{A}_q$). A positive reward is given when the agent reaches the target location, a negative reward is given when the agent collides with a wall. In all other states, no reinforcement signal is received. In detail, the reward function for a system state o is defined as follows:

$$R(o) = \begin{cases} 1 & \text{if goal state} \\ -1 & \text{if collision} \\ 0 & \text{else} \end{cases} \tag{7}$$

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.

5.1. *Goal finding performance*

In a first experiment, the robot has to solve the goal finding task in the environment depicted in Fig. 1. The agent starts from 20 starting positions equally distributed

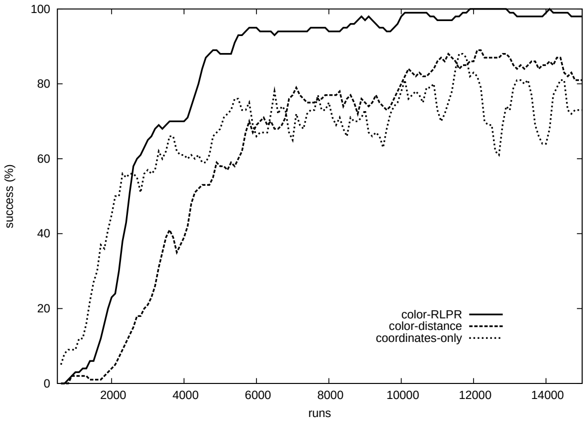


Fig. 5. Number of runs reaching the goal state: Both the coordinate-only and the color-distance approach don't show a stable and successful learning, while color-RLPR is both fast and successful.

inside the corridors. We test color-RLPR and color-GRLPR against a color-only representation given by ψ_a only, and two non- or semi-qualitative representations:

(i) a coordinate-only representation of the original MDP with $s = (x, y, \theta)$

(ii) 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 (if $n = 7$ these are the distances to the detected wall landmarks)

The continuous parts of these state vectors are approximated by using tile coding. As many different parameter values (n, angular size, 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 5 compares the learning success of color-RLPR and the color-distance and coordinate-only 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 6 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 learns even faster than the other two qualitative 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

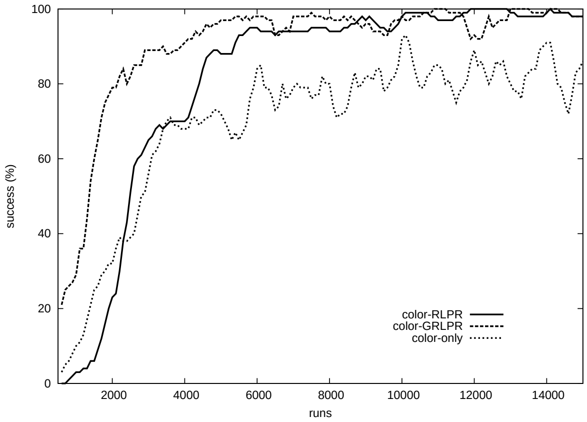


Fig. 6. Comparison of qualitative approaches: GRLPR shows a very fast learning especially in the early training phase. The color-only approach does not lead to a stable learning behavior.

be explained by the contradicting nature of the given world. 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.

An important measure of the quality of navigation is the number of collisions during training and test runs (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, color-GRLPR performs noticeably better than color-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.

Table 1. Number of collisions after two trials of 15,000 episodes

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

Note: RLPR approaches show fewer collisions than the metrical ones. Especially GRLPR reduces collisions significantly. In test runs (which do not include random actions), the difference is even higher.

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 color-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.

5.2. *Generalization capabilities*

As the color-(G)RLPR state space representation is agent centered and abstracts from metrical details, learned strategies should also be successful in environments that differ from the original one with respect to dimensions, orientation, and angles of corridors. So we created modified environments that have the same topology (including the same coloring of walls) as the one that the agent has been learning in. Training has been performed for 40,000 episodes in a world very similar to Fig. 1, with the only difference that it was turned by 90° and the left dead end was missing.

Figure 7 shows trajectories of the agent in two modified worlds: In the left one, all distances on the y-axis are scaled down, in the right one the corridors are deformed.

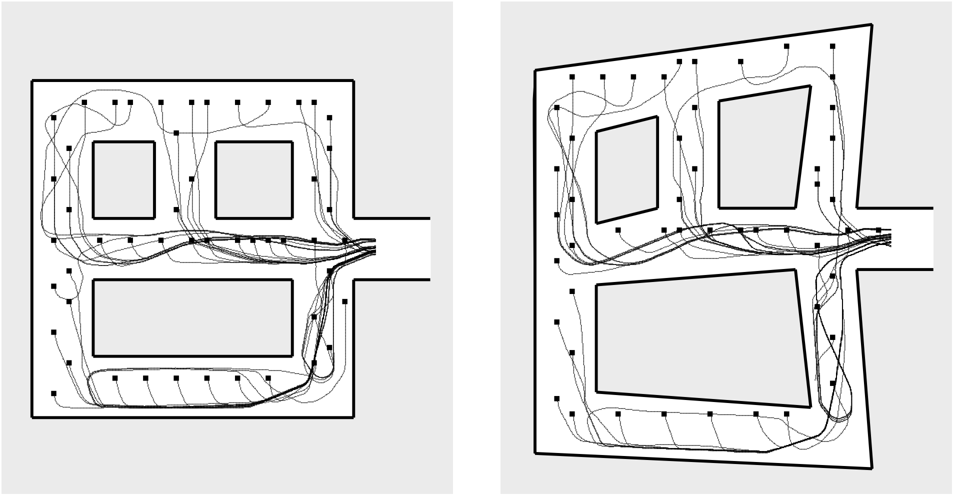


Fig. 7. GRLPR-based trajectories of the agent in worlds with modified distances (left) and deformed corridor angles (right): Learned strategies can be applied successfully.

We tested strategies learned with RLPR and GRLPR. The results are given in Table 2: Both RLPR- and GRLPR-based policies show a successful goal-finding performance in the modified worlds. While in the RLPR-based strategy up to 12%

of the test runs end in a collision, the GRLPR-based one is extremely effective and hardly shows any crashes at all. The agent is able to show a sensible behavior in both of these worlds, also performing U-turns to reach the goal faster.

Table 2. Performance in modified environments

	Goal finding success	
Representation	different dimensions	different angles
RLPR	94 %	88 %
GRLPR	100 %	98 %

Note: The learned policies also succeed in modified environments.

To show that learned knowledge of the structure of the world is not dependent on the goal-oriented landmark information, we must examine how the agent behaves when using a strategy learned with GRLPR in the absence of landmarks or in an unknown world. The knowledge gained from the environment in Fig. 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 caused by discounted rewards) the agent learns to happily run into any dead end and towards walls—at least, the system acquires the structural knowledge where goals are usually located in the environment it was trained in.

So the agent was trained in a more homogeneous environment without dead ends and a target area within a corridor (see Fig. 8). But even if looking simple, also this environment requires contradicting decisions at the same type of intersection to reach the goal state. 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. Also, no goal state is defined anymore, so that the agent only stops after a certain number of steps or in case of a collision. Figure 8 shows the resulting trajectories from 20 starting positions, using the strategy that took the least number of steps to the goal in the training environment. The agent is able to navigate collision-free and performs smooth curves, fully exploring the world. Most of the time it uses a follow-the-wall strategy, a commonly-used strategy in robotics. This generalized spatial behavior 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 when starting from 2 out of the 20 starting positions.

One could argue that the RLPR representation alone would be sufficient to learn a generally sensible navigation. However, this is only true for tasks that don't require contradicting actions at the same environmental structure. As mentioned before, this is not even the case in such a simple scenario as the one in Fig. 8. Also, goals are necessary for achieving this behavior: If there would not be a goal state with a positive reward, the learning process would only minimize negative reward, which will most likely be achieved by endless turning in place.

GRLPR-based strategies also succeed in totally unknown scenarios: Figure 9

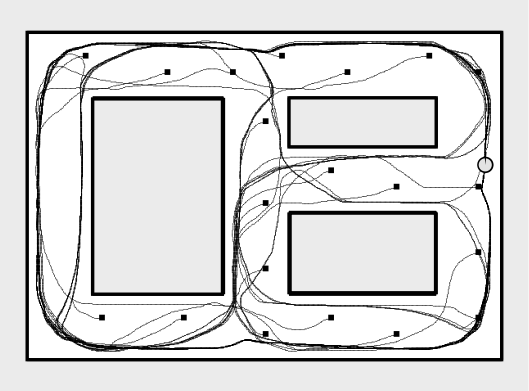


Fig. 8. Trajectories of the agent in the same environment with no landmark information available. Learning was performed using GRLPR. The small dots mark starting positions (the area without starting positions at the left was the target area in the training runs). The agent shows a sensible behavior and moves smoothly and collision-free.

shows the agent's trajectories in a landmark-free world it has never seen before with different corridor angles and structural elements. The robot is successfully applying the policy gained in the prior experiment without any modification.

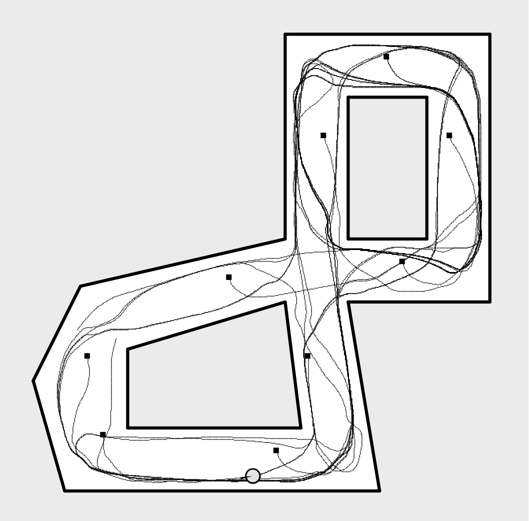


Fig. 9. Trajectories of the agent in an unknown environment without landmarks, using the strategy learned in the world depicted in Fig. 8 with GRLPR.

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 fails. Because of missing structural information, the agent's strategy restricts to collision avoidance, which frequently results in endless turnings around itself. Moreover, the trajectories when driving around curves are longer than with full RLPR. RLPR does not only encode information needed to prevent the agent from crashes, it represents the structure of the corridors the robot is operating in. So the learned generally sensible navigation strategy is more than a low-level collision avoidance and really implements an anticipatory behavior.

6. 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 RLPR 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 needed for learning. 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.

Acquired policies can also successfully be applied to scenarios with different metrics and corridor angles. 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 acquired strategies can be used as background knowledge for new learning tasks in unknown environments and therefore allow for speeding up learning. The system will also be enhanced by the ability of using "real" landmarks instead of the rather abstract concept of colored walls. This will allow for incorporating external background knowledge (for example verbal route directions) into the learning process, which will benefit from the qualitative manner of the presented spatial representation. Furthermore we will 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. We would also like to thank Christian Freksa, Diedrich Wolter, and Holger Schultheis for comments on this work, and Fabian Sobotka for his valuable help in implementation and evaluation.

References

1. Lutz Frommberger. A generalizing spatial representation for robot navigation with reinforcement learning. In *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2007)*, pages 586–591, Key West, FL, May 2007. AAAI Press.
2. Sebastian Thrun and Anton Schwartz. Finding structure in reinforcement learning. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems: Proceedings of the 1994 Conference*, volume 7. MIT Press, Cambrige, MA, 1995.
3. Terran Lane and Andrew Wilson. Toward a topological theory of relational reinforcement learning for navigation tasks. In *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2005)*, 2005.
4. Arthur P. S. Braga and Aluizio F. R. Araújo. A topological reinforcement learning agent for navigation. *Neural Computing and Applications*, 12:220–236, 2003.
5. Robert Glaubius, Motoi Namihira, and William D. Smart. Speeding up reinforcement learning using manifold representations: Preliminary results. In *Proceedings of the IJCAI Workshop "Reasoning with Uncertainty in Robotics"*, Edinburgh, Scotland, July 2005.
6. Sridhar Mahadevan and Mauro Maggioni. Value function approximation with diffusion wavelets and laplacian eigenfunctions. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems: Proceedings of the 2005 conference*, volume 18, pages 843–850. MIT Press, Cambridge, MA, 2006.
7. Matthew E. Taylor and Peter Stone. Cross-domain transfer for reinforcement learning. In *Proceedings of the Twenty Fourth International Conference on Machine Learning (ICML 2007)*, Corvallis, Oregon, June 2007.
8. Lisa Torrey, Jude Shavlik, Trevor Walker, and Richard Maclin. Skill acquisition via transfer learning and advice taking. In *Proceedings of the Seventeenth European Conference on Machine Learning (ECML'06)*, pages 425–436, Berlin, Germany, September 2006.
9. Dídac Busquets, Ramon López de Mántaras, Cales Sierra, and Thomas G. Dietterich. Reinforcement learning for landmark-based robot navigation. In *Proceedings of the Autonomous Agents and Multiagent Systems Conference (AAMAS 2002)*, pages 841–843, Bologna, Italy, 2002.
10. Benjamin Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.
11. Ralf Röhrig. *Repräsentation und Verarbeitung von qualitativem Orientierungswissen.* PhD thesis, University of Hamburg, 1998.
12. George D. Konidaris. A framework for transfer in reinforcement learning. In *Proceedings of the ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning*, Pittsburgh, PA, USA, June 2006.
13. Christoph Schlieder. Reasoning about ordering. In *Proceedings bof COSIT'95*, volume 988 of *Lecture Notes in Computer Science*, pages 341–349. Springer, Berlin, Heidelberg, 1995.
14. S. D. Whitehead and D. H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7(1):45–83, 1991.

15. Paul Crook and Gillian Hayes. Learning in a state of confusion: Perceptual aliasing in grid world navigation. In *Proceedings of Towards Intelligent Mobile Robots (TIMR 2003)*, UWE, Bristol, August 2003.
16. Amitabha Mukerjee and Gene Joe. A qualitative model for space. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI)*, pages 721–727, Boston, MA, August 1990. Morgan Kaufmann.
17. Roop K. Goyal and Max J. Egenhofer. Consistent queries over cardinal directions across different levels of detail. In A. M. Tjoa, R. Wagner, and A. Al-Zobaidie, editors, *Proceedings of the 11th International Workshop on Database and Expert System Applications*, pages 867–880, Greenwich, UK, September 2000. IEEE Computer Society.
18. Richard S. Sutton. Generalization in reinforcement learning: Successful examples using sparse tile coding. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference*, volume 8, pages 1038–1044. MIT Press, Cambrige, MA, 1996.
19. Christopher Watkins. *Learning from Delayed Rewards.* PhD thesis, Cambridge University, 1989.