

Generalization and Transfer Learning in Noise-Affected Robot Navigation Tasks

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

Abstract. When a robot learns to solve a goal-directed navigation task with reinforcement learning, the acquired strategy can usually exclusively be applied to the task that has been learned. Knowledge transfer to other tasks and environments is a great challenge, and the transfer learning ability crucially depends on the chosen state space representation. This work shows how an agent-centered qualitative spatial representation can be used for generalization and knowledge transfer in a simulated robot navigation scenario. Learned strategies using this representation are very robust to environmental noise and imprecise world knowledge and can easily be applied to new scenarios, offering a good foundation for further learning tasks and application of the learned policy in different contexts.

1 Introduction

In goal-directed navigation tasks, an autonomous moving agent has found a solution when having reached a certain location in space. Reinforcement Learning (RL) [1] is a frequently applied method to solve such tasks, because it allows an agent to autonomously adapt its behavior to a given environment. In general, however, this solution does only apply to the problem that the system was trained on and does not work in other environments, even if they offer a similar structure or are partly identical. The agent lacks an intelligent *understanding* of the general structure of geometrical spaces.

The ability to transfer knowledge gained in previous learning tasks into different contexts is one of the most important mechanisms of human learning. Despite this, the question whether and how a learned solution can be reused in partially similar settings is still an important issue in current machine learning research. Recently the term *transfer learning* was used for this research field. In contrast to *generalization*, which describes the ability to apply a learned strategy to unknown instances within the same task, transfer learning tackles generalization ability across different tasks.

In real world applications we frequently experience changes in the environment that modify the characteristics of a task. If the system input is provided by sensors, it is subject to environmental noise, and the amount and type of noise is subject to external conditions like, e.g., the weather. Consequently, applying a learned strategy to the same task under different conditions can also be seen as a process of transfer learning, because the task is not anymore the same as before. The complexity of the transfer process is

heavily influenced by the agent’s perception of the world. *Abstraction* of the state space is a key issue within this context.

In this work we investigate a simulated indoor robotics scenario where an autonomous robot has to learn a goal-directed wayfinding strategy in an unknown office environment. We will show that the choice of an appropriate spatial abstraction for the state space can minimize transfer efforts and enables the agent to learn a strategy that reaches the goal in the same environment under different conditions as well as showing a sensible navigation behavior in totally unknown environments. In particular, we will investigate if the chosen abstraction mechanism can be a means to enable a strategy transfer from an abstract simulation to a real robotics system in the future.

After an overview of related work, Sect. 3 introduces the domain we are investigating and shortly summarizes the RLPR representation used in this work. A study on the influence of environmental noise in the perception of the agent follows in Sect. 4. Section 5 investigates the properties of knowledge transfer under noisy conditions, presents a new algorithm to create a new policy for a target environment, and discusses the remaining challenges. The paper closes with a summary and an outlook.

2 Related Work

While most approaches on robot navigation with RL concentrate on the discrete state space of grid-worlds, there is also work being done in continuous domains. Lane and Wilson describe relational policies for spatial environments and demonstrate significant learning improvements [2]. However, their approach runs into problems when non-homogeneities such as walls and obstacles appear. A landmark based qualitative representation was also used in [3]. For navigation the authors rely on rather complex actions, and obstacle avoidance is handled by a separate component. Transfer to other environments has not been performed within these approaches.

Various work has been done in the domain of transfer learning: Thrun and Schwartz tried to find reusable structural information that is valid in multiple tasks [4]. They introduced so-called skills, which collapse a sequence of actions into one single operation. Konidaris and Barto use the distinction between agent-space and problem-space to build options that can be transferred across tasks [5]. Taylor and Stone present an algorithm that summarizes a learned policy into rules that are leveraged to achieve a faster learning in a target task in another domain [6]. Torrey et al. extract transfer rules out of a source task based on a mapping specifying problem similarities [7]. The latter two approaches have in common that they both rely on external knowledge about the similarities between source and target task. The work we present here focuses on the abstraction method of RLPR [8] that uses this external knowledge to create identical parts of the state space representation for problems sharing the same structure.

3 Learning with Qualitative State Space Abstraction

3.1 A Goal Directed Robot Navigation Task

The scenario considered in this work is an indoor navigation task where an autonomous robot learns to find a specified location in an unknown environment, the goal state. This

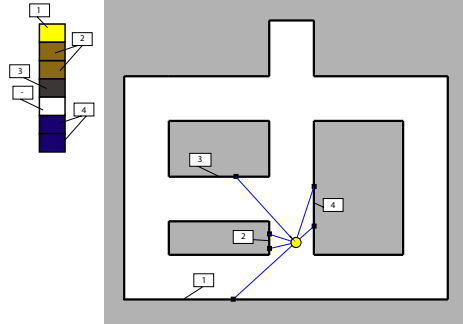


Fig. 1. The navigation task: a robot in a simplified simulated office environment with uniquely distinguishable walls. The lines departing from the robot visualize landmark scans. Detected landmarks are depicted in the upper left. The label “-” means that nothing was perceived within the agent’s scanning range. The target location is the dead end.

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} of navigational actions the agent can perform, 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 $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the reward the agent receives over time.

To avoid the problems of a continuous state space, we consider the agent’s *observation* instead of \mathcal{S} , using a function $\psi : \mathcal{S} \rightarrow \mathcal{O}$ that 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 used function ψ is introduced in Sect. 3.2.

The robot is assumed to be able to perceive walls around it within a certain maximum range. It is capable of performing three different 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. See Fig. 1 for a look on the simulation testbed.

3.2 State Space Abstraction with RLPR

This section shortly introduces the *Relative Line Position Representation* (RLPR) as presented in [8]. RLPR is specifically designed for robot navigation tasks in indoor environments.

The idea behind RLPR is to divide each system state into two separate parts which represent two different aspects of goal-directed agent navigation. *Goal-directed behavior* towards a certain target location depends on the position of the agent within the

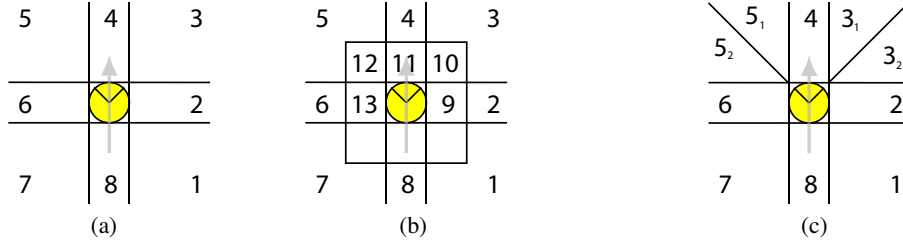


Fig. 2. RLPR: Neighboring regions around the robot in relation to its moving direction. Note that the regions in the immediate surroundings (b) are proper subsets of R_1, \dots, R_8 (a). We present a variation of the representation that divides regions R_3 and R_5 into two parts to assure a higher resolution in this area (c).

world. It can, for example, be encoded by a circular ordering of detected landmarks. (e.g. cf. [9]). In this work, we assume that every wall in the environment can be identified, and recognized walls at n discrete angles around the agent $\psi_l(s) = (c_1, \dots, c_n)$ serve as a representation for goal directed behavior (see Fig. 1). *Generally sensible behavior* regarding the structure of the world is the same at structurally similar parts within and across environments. In contrast to goal-directed behavior it is independent of the goal-finding task to solve. This distinction closely relates to the concepts of *problem-space* and *agent-space* [5]. RLPR is designed to encode the agent-space for generally sensible behavior. It benefits from the background knowledge that the structuring elements for navigation inside of buildings are the walls, which induce sensible paths inside the world which the agent is supposed to follow.

RLPR is a qualitative spatial abstraction of real world sensory data. It encodes the position of line segments perceived by the the agent’s sensory system relative to its moving direction. The representation scheme of RLPR is based on the Direction Relation Matrix [10]. The space around the agent is partitioned into bounded and unbounded regions R_i (see Fig. 2). Two functions $\bar{\tau} : \mathbb{N} \rightarrow \{0, 1\}$ and $\bar{\tau}' : \mathbb{N} \rightarrow \{0, 1\}$ are defined: $\bar{\tau}(i)$ denotes whether there is a line segment detected within a sector R_i and $\bar{\tau}'(i)$ denotes whether a line spans from a neighboring sector R_{i+1} to R_i . See Fig. 3 for an example.

$\bar{\tau}_i$ is used for bounded sectors in the immediate vicinity of the agent (R_9 to R_{13} in Fig. 2(b)). Objects that appear there have to be avoided in the first place. The position

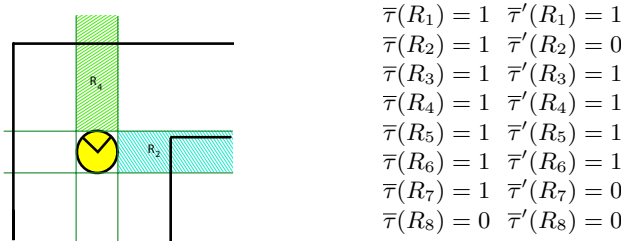


Fig. 3. RLPR values in an example situation. Region R_2 (right) and R_4 (front) are marked.

of detected line segments in R_1 to R_8 (Fig.2(a)) is interesting information to be used for general orientation and mid-term planning, so $\bar{\tau}'$ is used for R_1 to R_8 .

Combined with a vector of seven detected landmarks (c_1, \dots, c_7) , the overall state space representation function ψ using RLPR is

$$\psi(s) = (\psi_l(s), \psi_r(s)) = (c_1, \dots, c_7, \bar{\tau}'(R_1), \dots, \bar{\tau}'(R_6), \bar{\tau}(R_9), \dots, \bar{\tau}(R_{13})) \quad (1)$$

$\bar{\tau}'(R_7)$ and $\bar{\tau}'(R_8)$ are irrelevant here, because the agent cannot move backwards, so they are not included into the state space representation within this work.

Figure 2(c) shows a new variant of the RLPR partition that divides the regions in the front left and right of the agent into two parts and increases the resolution in this area. This variant proved to be beneficial especially for transfer tasks into environments lacking landmark information. All the experiments in Sect. 5 have been conducted using it, the others use the representation given in (1).

Because RLPR performs abstraction directly and only on the state space representation, it can be combined with every underlying learning approach. RLPR creates a very small and discrete state space and is therefore suitable for easy to handle table based value function representations. It has been shown that this representation outperforms coordinate- and distance based representations by robustness and learning speed [8].

GRLPR (Generalizing RLPR) is a technique to enable applicability of RLPR-based representations to structurally similar regions within the same or across different learning tasks. The function approximation method of tile coding [11] (which is also known as CMACs) is applied to the landmark part $\psi_l(s)$. A tile size big enough that the whole landmark space of N landmarks can fit within one tile ensures that each update of the policy affects all system states with the same RLPR representation. As an effect, the agent can reuse structural knowledge acquired within the same learning task and the learned policy is immediately applicable to different learning tasks in the same environment and even to new, unknown worlds (ad-hoc strategy transfer). A discussion of the transfer properties of GRLPR based strategies follows in Sect. 5.2.

4 RLPR and Environmental Noise

4.1 Unreliable Line Detection

The question how RLPR-based strategies behave under different conditions and if they can also be transferred from simulation to a real system puts a focus on the behavior under environmental noise. An abstraction method like RLPR clusters several different percepts to a single observation o . RLPR offers a very strong abstraction from detected line segments to a few binary features, so one could argue that an erroneous classification may have severe consequences.

In a real robotics system, environmental data is mostly provided by a 2D laser range finder (LRF). After that a line detection algorithm (as for example described in [12]) will extract line segments from the collected distance values. Data provided by an LRF is subject to noise: A line segment in the real world (l) will frequently be detected as two or more separate, shorter line segments (l_1, \dots, l_n) . However, RLPR is very robust towards such errors: $\bar{\tau}$ (used for R_9, \dots) regards the existence of features in an area, so

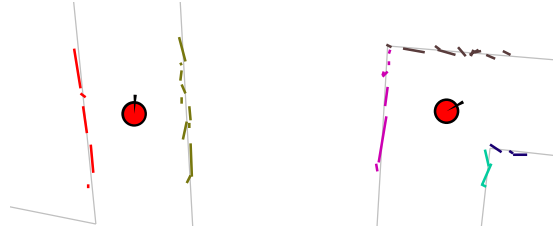


Fig. 4. The noise model: The real walls (thin lines) are detected as several small chunks (thick lines). Depicted is a noise level of $\rho = 20$.

misclassifications are only to be expected in borderline cases. $\overline{\tau}'$ delivers a wrong result only if the “hole” between l_i and l_{i+1} is exactly at one of the region boundaries.

To evaluate this, we applied a noise model to the line detection in the simulator. Depending on a parameter ρ , $k \in \{1, \dots, \rho + 1\}$ holes of 20 cm each are inserted into each line segment l of a length of 5 meters, i.e., a wall with a length of 5 meters will in the average be detected as $\frac{1}{2}(\rho + 1) + 1$ line segments (for comparison, the robot’s diameter is 0.5 meters). Additionally, the start and end points of the resulting line segments l_1, \dots, l_k are then relocated into a random direction. The result is a rather realistic approximation of line detection in noisy LRF data (see Fig. 4).

For this experiment and all subsequent ones described in this work we used Watkins’ $Q(\lambda)$ algorithm [13]. During training, the agent uses an ϵ -greedy policy with $\epsilon = 0.15$ for exploration. 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 $\alpha = 0.05$, 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. Within this section, the state space representation from (1) was used with GRLPR generalization. In all the experiments in this work, the world in Fig.1 was used for training.

Figure 5 (upper graphs) shows how the learning success is affected by different noise values ρ . Of course, as the noisy perception influences the transition probabilities T , learning time increases with the value of ρ and learning success decreases somewhat. But up to $\rho = 20$ a success rate of 95% in tests can be reached after 40,000 learning runs. Regarding collisions, the differences between the noise levels are even smaller compared to the differences with regard to learning speed. Even with $\rho = 20$ (which means more than 11 detected line segments per 5 meter in the average), the agent performs comparably well.

The sensibility of the line detection to group points together is a parameter to the algorithm and can be appropriately adjusted. For RLPR, it is better to aberrantly detect two adjacent line segments as a single one, because it does not matter to the navigation behavior as long as the gap is not that big that the robot could pass through it. Thus, a simple trick can reduce the impact of noise: Each line can be prolonged by a few centimeters to reduce the probability of holes. Figure 5 (lower graphs) shows that for $\rho = 20$ such prolongation leads to a significant improvement and a result comparable to very low noise levels.

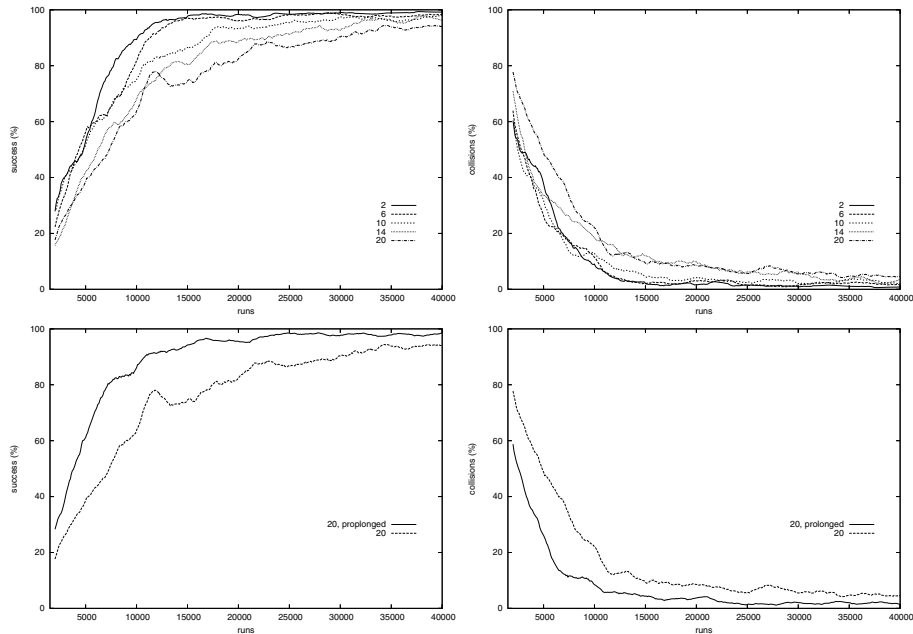


Fig. 5. Learning with unreliable line detection. Upper row: Learning success (left) and collisions (right) with different noise levels ρ . Even under heavy noise, the system learns successfully (but somewhat slower) with a low number of collisions. Lower row: Line segment prolongation significantly improves the learning behavior at a noise level of $\rho = 20$.

4.2 Unreliable Landmark Detection

Also landmark detection can be (and usually is) influenced by distortion. In another experiment we tested how the structural information of RLPR helps the system to cope with very unreliable landmark detection. The noise model implied here is the following: Each single landmark scan c_i returns the correct landmark detection with a probability of σ , so the probability that (c_1, \dots, c_7) is detected correctly is σ^7 . For $\sigma = 95\%$ the overall probability to detect the whole vector correctly is around 70%, 3% for $\sigma = 60\%$, and only 0.2% for $\sigma = 40\%$. In case of a failure, no landmark is detected.

Figure 6 shows the learning success for values of $\sigma = 95, 85, 60, 40$ and 20%. The higher the failure in landmark detection, the lower is the rate at which the agent reaches the goal. However, even with $\sigma = 60\%$, the goal is reached in about two third of the test runs after 40,000 episodes of training. This is still a very good value, because the goal area is located behind a crossing, and reaching it is subject to a decision, which is hard to draw when the agent is that uncertain of where it is. Furthermore, the number of collisions is more or less negligible after very few runs, almost independent of σ . This shows that GRLPR does its job in enabling the agent to learn a good general navigation behavior even in the absence of utilizable landmark information.

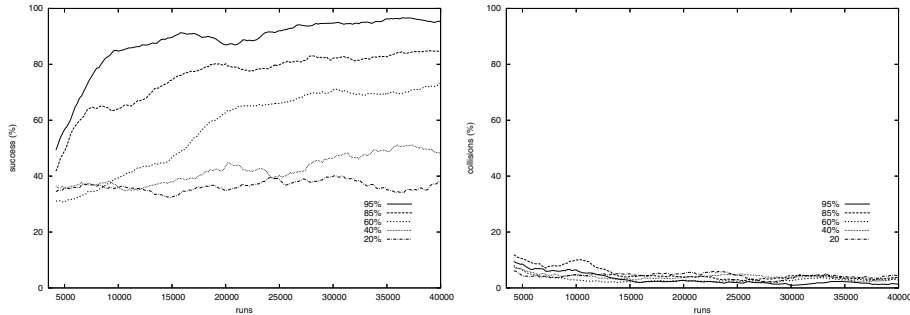


Fig. 6. Learning with unreliable landmark information for different levels of input distortion σ : The learning success of reaching the goal (left) decreases with higher values of σ , because the agent is uncertain of where it is; but is still comparably high even for very high distortion levels. The number of collisions (right) is hardly affected by landmark distortion.

Overall, the results in noisy environmental condition show a very robust behavior of the learned strategies and are a strong hint that strategies learned in simulation may also be successfully applied to a real robot system.

5 Knowledge Transfer to Unknown Environments

5.1 Ad-Hoc Strategy Transfer with Noisy Training Data

It has been shown in [8] that strategies learned with GRLPR can be transferred to different scenarios than the one the system was trained in without any further effort (ad-hoc strategy transfer), even if the new environments are differently scaled, contain different corridor angles or even lack any landmark information. This is because the RLPR representation of the new world is identical and the tile coding approximation of the value function returns a reasonable Q-value also for landmarks that have never been perceived before.

As the results so far provided transferred strategies that were trained under optimal environmental conditions we tested the transfer of a policy that was trained with noisy line detection. We trained the system in the world depicted in Fig. 1 with GRLPR both with a perfect line detection and with a noise parameter $\rho = 10$. The learned strategies have then been applied to another world (Fig. 7). Without environmental noise in training, the robot moves much closer to the walls compared to the noise-affected strategy which keeps the agent predominantly in the middle of the corridors and creates movement patterns that look well from a human perspective. The strategy of the noise-free learning task uses features in the bounded RLPR regions ($R_9...$) for navigation, while the noise-affected agent learned to avoid them: the agent plays for safety. Thus, noise in training does not only not disturb the agent in learning a transferable strategy, it proves being beneficial in keeping it away from possible danger and provides sensible trajectories within unknown worlds.

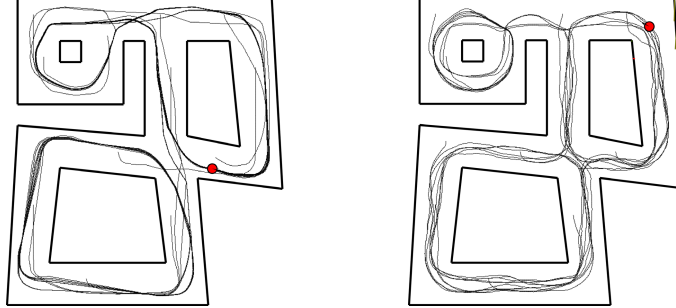


Fig. 7. Ad-hoc strategy transfer under noisy line detection: Trajectories in an unknown environment, learned with GRLPR with perfect line detection (left) and with distorted line detection with a noise parameter $\rho = 10$ (right). The noise-affected strategy shows a reasonable navigation behavior and keeps the robot safely in the middle of the corridors.

5.2 A Strategy Transfer Algorithm

GRLPR based strategies, however, need to be carefully checked when being transferred to a target scenario. Not all policies perform well in new worlds, even if they showed a perfect behavior in the source problem. We identified two drawbacks in the ad-hoc concept of GRLPR knowledge transfer, which are caused by the way generalization is achieved: First, in the target environment, the system is confronted with completely unknown landmark vectors and usually returns a reasonable reward. However, the reward is not independent of the new landmark vector, and different landmark inputs may result in different actions, even if the RLPR representation is the same. Second, because the system is trained with discounted rewards, Q-values are high for states near the goal state and decrease with distance from it. As tile coding sums up values in different tiles, the states in the vicinity of the goal have the biggest impact on the generalized strategy.

To get rid of these problems, we present a new method to generate a strategy for generally sensible navigation behavior in an observation space \mathcal{O}' without landmark information with $\mathcal{O}' = \{\psi_r(s)\}$, $s \in \mathcal{S}$. It works both on policies learned with and without GRLPR generalization; but in all the experiments conducted we used non-GRLPR based policies in the source task. Given a learned policy π with a value function $Q_\pi(o, a)$ ($o \in \mathcal{O}$, $a \in \mathcal{A}$), we achieve a new policy π' with $Q_{\pi'}(o', a)$ ($o' \in \mathcal{O}'$, $a \in \mathcal{A}$) with the following function:

$$Q_{\pi'}(o', a) = \frac{\sum_{c \in \{\psi_l(s)\}} (\max_{b \in \mathcal{A}} (|Q_\pi((c, o'), b)|))^{-1} Q((c, o'), a)}{\#\{(c, o'), a) | Q_\pi((c, o'), a) \neq 0\}} \quad (2)$$

This is just a weighted sum over all possible landmark observations (in reality, of course, only the visited states have to be considered, because $Q(o, a) = 0$ for the others, so the computational effort is very low). It is averaged over all cases where the Q-value is not zero ($\#$ describes a cardinality operator over a set here). The weighting factor scales all values according to the maximum reward over all actions to solve the mentioned problem of states far from the goal having lower Q-values.

A nice property of strategies created like this is that, in contrast to the GRLPR based strategies, they can be analyzed before applying them. It is easy to count the observation states in \mathcal{O} that contributed to $Q_{\pi'}(o', a)$ to get an idea how big the data basis for an observation is. It is also possible to define a confidence measure on the new strategy π' .

$$\text{conf}(o') = \begin{cases} \frac{1}{\#\mathcal{A}} \sum_{a \in \mathcal{A}} (\max_{b \in \mathcal{A}} (Q_{\pi'}(o', b)) - Q_{\pi'}(o', a)) & Q_{\pi'}(o', a) \neq 0 \ \forall a \in \mathcal{A} \\ 0 & \text{else} \end{cases} \quad (3)$$

gives a measure for an observation o' of how certain the strategy is in choosing the best action according to the Q-value at this state over another one. If there is no information collected for an action a yet, the result is defined as 0. Of course there are situations where two or more actions are equally appropriate, so this measure only makes sense when summed up over the whole observation space for the whole strategy:

$$\text{Conf}(\pi') = \frac{\sum_{o' \in \mathcal{O}} \text{conf}(o')}{\#\{o' | \text{conf}(o') \neq 0\}} \quad (4)$$

$\text{Conf}(\pi)$ can be used to compare strategies before applying them to another world. When comparing the confidence values of a task learned with and without line detection distortion, for example, $\text{Conf}(\pi')$ converged to 0.434 for the undistorted case and to 0.232 in the same world with a distortion of $\rho = 10$.

However, strategies that are created with (2) don't prove to be completely successful and even show similar results as GRLPR based strategies: Some of them work perfectly well in the target environment, and some don't and lead to collisions frequently. Even after introducing the weight factor, the differences of the output of the value function Q still seem to be too high.

A solution to this problem is to totally abstract from the rewards and only regard which action the agent will greedily be chosen in a particular state and which one has the least reward expectation. We then sum up over the decisions, not over the expected reward. We modify (2) accordingly:

$$Q_{\pi'}(o', a) = \frac{\sum_{c \in \{\psi_i(s)\}} \phi_b((c, o'), a)}{\#\{Q((c, o'), a) \neq 0\}} \quad (5)$$

$$\phi_b(s, a) = \begin{cases} 1 & \text{if } Q(o, a) = \max_{b \in \mathcal{A}} Q(o, b) \\ -1 & \text{if } Q(o, a) = \min_{b \in \mathcal{A}} Q(o, b) \\ 0 & \text{else} \end{cases}$$

This mechanism proves to create strategies that are robustly applicable to new environments. Figure 8 shows that the strategies perform very well even under line detection distortion in the target world. In contrast to GRLPR based strategies or those created with (2), experiments show that you can pick any strategy after an arbitrary learning time and will not experience outliers that don't succeed at all. Strategies with higher confidence values perform better, and $\text{Conf}(\pi')$ asymptotically increases over learning time. In contrast to that, GRLPR based strategies show a tendency to perform worse after longer training. Furthermore, the transferred strategies prove to be robust against

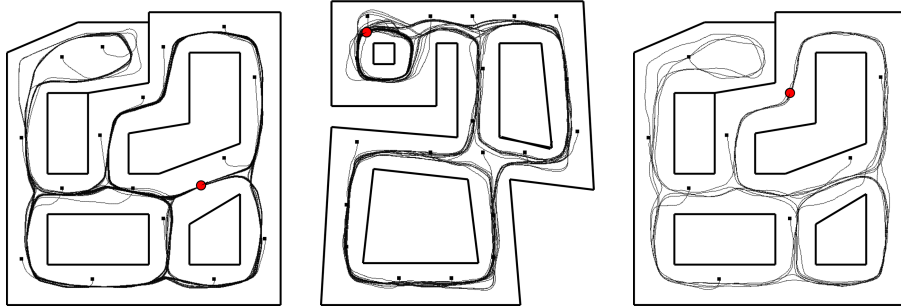


Fig. 8. Trajectories of policies transferred with (5) to unknown environments: The agents keeps in the middle of corridors and shows a sensible navigation behavior even at line detection noise levels in the target environment of $\rho = 1$ (left), $\rho = 10$ (middle) or $\rho = 20$ (right).

noise and do not, like frequently observable in GRLPR based ones, show a behavior that navigates closely to walls. Without noise, the agent navigates collision-free, and also for high values of ρ , collisions rarely occur.

Strategies $Q_{\pi'}$ that were transferred to \mathcal{O}' and only rely on the RLPR part of the representation can then be used as a basis for new goal-directed learning tasks in unknown worlds: If the value $Q_{\pi'}((c, o'), a)$ of the policy to learn is undefined at an observation $(c, o') \in \mathcal{O}$ (because this observation has not been visited yet), it can be initialized with $Q_{\pi'}(o', a)$, which provides the learning agent with a good general navigation behavior from scratch.

5.3 Remaining Challenges and Limitations

The algorithm presented in (5) performs very well to create generally sensible navigation strategies for different indoor environments. However, the learned policies do not completely prevent the agent from collisions when environmental noise is applied. Collisions may occur in two situations: First, when corridors are getting narrower and the agent tries to perform an U-turn or the agents is driving around a very steep curve, collisions may occur. The training has been performed in a very regular environment, so the learned turning strategy may bring the robot closer to a wall than it was used to in training. Second, and more frequently, the agent crashes when it is directly heading into a corner. In this case, the RLPR representation signals obstacles everywhere in front of the robot. Due to the agent's motion dynamics that don't include turning in place or even driving backwards, the robot is doomed in a dead-end it cannot escape anymore.

Usually, the transferred policy hinders the agent to reach such situations, but under conditions of noise this cannot be prevented completely. To avoid such trouble, it is suggested to provide (possibly negatively rewarded) actions that allow for escaping from such situations, especially when planning to apply learned strategies from simulation to a real robotics platform.

6 Conclusions and Future Work

In this work we investigated into generalization and transfer properties of robot navigation strategies learned in simulation with RL while using the abstraction method of RLPR. We showed that RLPR is a sensible and robust abstraction and enables learning both with distorted line segment and landmark detection. A new algorithm was presented for transferring goal-directed navigation strategies to generally sensible navigation strategies that can be transferred to arbitrary environments. The results prove to be robust and consistent, and the navigation policies enable successful navigation in unknown worlds even under heavy environmental noise.

Future work will have to provide a thorough analysis of the impact of the confidence measure for transferred strategies. Regarding the promising results, a next step will be to show that strategies learned in simulation can be transferred to a real robot.

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

1. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. In: Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA (1998)
2. Lane, T., Wilson, A.: Toward a topological theory of relational reinforcement learning for navigation tasks. In: Proceedings of FLAIRS 2005 (2005)
3. Busquets, D., de Mántaras, R.L., Sierra, C., Dietterich, T.G.: Reinforcement learning for landmark-based robot navigation. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) AAAMAS 2002. LNCS (LNAI), vol. 2636, Springer, Heidelberg (2003)
4. Thrun, S., Schwartz, A.: Finding structure in reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 7 (1995)
5. Konidaris, G.D., Barto, A.G.: Building portable options: Skill transfer in reinforcement learning. In: Proceedings of IJCAI 2007 (January 2007)
6. Taylor, M.E., Stone, P.: Cross-domain transfer for reinforcement learning. In: Proceedings of ICML 2007, Corvallis, Oregon (June 2007)
7. Torrey, L., Shavlik, J., Walker, T., Maclin, R.: Skill acquisition via transfer learning and advice taking. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 425–436. Springer, Heidelberg (2006)
8. Frommberger, L.: A generalizing spatial representation for robot navigation with reinforcement learning. In: Proceedings of FLAIRS 2007, Key West, FL (May 2007)
9. Schlieder, C.: Reasoning about ordering. In: Kuhn, W., Frank, A.U. (eds.) COSIT 1995. LNCS, vol. 988, pp. 341–349. Springer, Heidelberg (1995)
10. Goyal, R.K., Egenhofer, M.J.: Consistent queries over cardinal directions across different levels of detail. In: Tjoa, A.M., Wagner, R., Al-Zobaidie, A. (eds.) Proceedings of the 11th International Workshop on Database and Expert System Applications, pp. 867–880 (2000)
11. Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse tile coding. In: Advances in Neural Information Processing Systems, vol. 8 (1996)
12. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems* (1997)
13. Watkins, C.: Learning from Delayed Rewards. PhD thesis, Cambridge University (1989)