Representing and Selecting Landmarks in Autonomous Learning of Robot Navigation

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. Navigation based on detected landmarks is an important facet of robot navigation. This work investigates into a qualitative representation of landmarks for an autonomous learning task where a robot learns a goal directed navigation strategy with reinforcement learning. We discuss how to build a suitable landmark-based representation. In particular, we focus on selection of landmarks to regard when experiencing a multitude of landmarks, because representing all of them would blow up the state space inappropriately. Thus, we examine strategies for this selection. Furthermore, we introduce a background knowledge based structure-aware landmark selection mechanism to limit landmark observation to the cases where it is really needed.

1 Introduction

Autonomous navigation of mobile robots is an important topic in the field of artificial intelligence. Especially interesting are systems that are able to adapt their behaviors to the needs of the given task and environment: Such a system learns a strategy on its own. In this work we will study an approach utilizing Reinforcement Learning (RL) to learn a strategy that allows a robot (the autonomous agent) to succeed in a goal-directed navigation task: The agent is asked to drive to certain location from any position within the environment. It learns to select an action for every given observation of the world.

In such large and continuous state spaces, RL methods require very long training times. To successfully apply RL to this kind of tasks, an intelligent reduction of the state space is an important prerequisite, and state space abstraction is a key issue to that. In human spatial cognition landmarks are among the most important concepts. Regarding the position of landmarks perceived in the local surroundings of a robot instead of trying to estimate its metrical position within a global map has shown to be successful in many approaches over the last years (e.g., [1,2]), especially when RL is applied [3,4]. Regarding landmarks enables for building agent-centered representations that are invariant of absolute orientation and can build comparably small state-spaces.

In this work, we present a method to specify the agent's position by regarding ordering information of detected landmarks. In particular, we focus on the question how to encode the locations of landmarks in a qualitative spatial representation especially under the assumption of having a multitude of landmarks to cope with. A big number of landmarks leads to high-dimensional representation vectors and blows up state space and learning time. We present strategies to reduce the number of regarded landmarks

C. Xiong et al. (Eds.): ICIRA 2008, Part I, LNAI 5314, pp. 488–497, 2008.

to shrink the state space and enable faster learning of goal-directed strategies. Furthermore, we present a method to utilize background knowledge from structural spatial information to restrict landmark adherence to cases in which it is really needed.

This paper is structured as follows: After an overview on related work in Section 2, Section 3 introduces the learning task and the abstraction paradigm of landmark-enhanced RLPR. Section 4 discusses how to cope with a a multitude of detected landmark and examines landmark selection strategies, including the background knowledge based SDALS mechanism for landmark omission. An experimental evaluation is given in Section 5, followed by a discussion of the results in Section 6 and a conclusion.

2 Related Work

Landmarks play a prominent role in many approaches to robot navigation, for example in the work by Lazanas and Latombe [2] or Prescott [5], who describes relations between landmarks. While these approaches use metrical information, there exist quite a few qualitative approaches to landmark based navigation. An early approach that totally omits metrical measurements and just considers qualitative environment information is the QUALNAV algorithm [1]. It uses ordering information of detected landmarks to approximately encode the robot's position, but, as pointed out by Schlieder [6], it is partly based on wrong assumptions about the robot's position. To cure the problems having arisen in QUALNAV, the so-called *panorama* approach [6] has been introduced. The panorama consists of the circular order of detected landmarks plus a set of virtual landmarks that arise by point reflection of each landmark by 180°. However, building up the panorama representation requires very exact metrical sensory information regarding the angles. Qualitative abstraction is especially valuable when the size of the state space plays an important role, as in reinforcement learning approaches. Busquets et al., for example, successfully utilize fuzzy qualitative information about relations and distances of landmarks in a RL setting [3].

3 A Landmark-Based Spatial Representation

3.1 A Goal Directed Robot Navigation Task

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

To avoid the problems a continuous state space introduces 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 Section 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. There is no built-in collision avoidance or any other navigational intelligence provided.

3.2 Qualitative State Space Abstraction

The next two paragraphs shortly introduce the *landmark-enhanced Relative Line Position Representation* (landmark-enhanced RLPR) as presented in [4]. RLPR performs abstraction directly and only on the state space representation and creates a very small and discrete state space. It has been shown that this representation outperforms coordinate- and distance based representations regarding robustness and learning speed.

The key idea behind RLPR is to divide each 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 world, which can ideally be encoded with the help of landmarks. *Generally sensible behavior* regarding the structure of the world is the same at structurally similar places 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* (goal-directed behavior) and *agent-space* (generally sensible behavior) [7].

We now create a bipartite observation vector $\psi(s) = (\psi_l(s), \psi_r(s))$ where $\psi_l(s)$ denotes the problem space representation and $\psi_r(s)$ the agent space representation. We will first shortly introduce how to represent the latter one.

Representing Agent Space. RLPR is designed to encode agent-space for generally sensible behavior. It benefits from the background knowledge that the structuring elements for navigation inside of buildings are 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 space around the agent is partitioned into bounded and unbounded regions R_i (see Fig. 1). Two functions $\overline{\tau} : \mathbb{N} \rightarrow \{0, 1\}$ and $\overline{\tau}' : \mathbb{N} \rightarrow \{0, 1\}$ are defined: $\overline{\tau}(i)$ denotes whether there is a line segment detected within a sector R_i and $\overline{\tau}'(i)$ denotes whether a line spans from a neighboring sector R_{i+1} to R_i.

$\overline{\tau}_i$ is used for bounded sectors in the immediate vicinity of the agent (R_9 to R_{13} in Fig. 1(b)). Objects that appear there have to be avoided in the first place. The position of detected line segments in R_1 to R_8 (Fig. 1(a)) is interesting information to be used for general orientation and mid-term planning, so $\overline{\tau}'$ is used for R_1 to R_8.

Summed up, the agent space representation $\psi_r(s)$ is defined as

$$\psi_r(s) = (\overline{\tau}'(R_1), \ldots, \overline{\tau}'(R_6), \overline{\tau}(R_9), \ldots, \overline{\tau}(R_{13})) \tag{1}$$

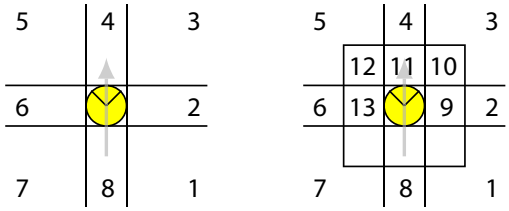

Fig. 1. RLPR: Neighboring regions around the robot in relation to its moving direction. Note that the regions in the immediate surroundings (b) are proper subsets of $R_1, \ldots, R_8$ (a).

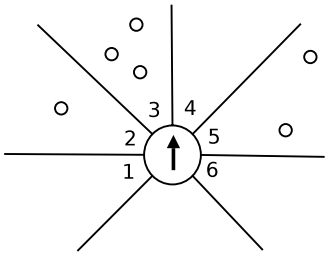

Fig. 2. Six sectors around the robot. One landmark is detected in sector 2, three in sector 3, and two in sector 5.

Representing Problem Space. In [4], problem space is encoded by a vector of uniquely identified walls around the agent, represented by a certain color. It is a very strong assumption that all walls can be correctly identified. Thus, we now investigate how problem space can be represented using point-based landmarks.

3.3 Representing Point Based Landmarks

At any point in time, the agent is surrounded by a varying number of detected landmarks. Each landmark b has a certain distance d_b to the agent and an angle ϕ_i towards its moving direction. The sequence $(d_1, \phi_1), \ldots, (d_n, \phi_n)$ of n detected landmarks exactly describes the position of the agent in an egocentric frame of reference. That is, every sequence $(d_1, \phi_1), \ldots, (d_n, \phi_n)$ maps to exactly one position (x, y, θ) of the agent. However, the state space is still continuous, and very similar states (that is, positions that are very close to each other) still have a different, yet overly exact representation.

To abstract from this differences we aim at providing a qualitative representation that roughly encodes the whereabouts of landmarks with respect to the agent and its moving direction. Around the robot we partition the space into circular sectors with equal angular size (see Fig. 2). The space in the back of the robot can remain unconsidered. Every landmark that is detected by the sensory system of the agent can now be mapped to exactly one sector in the plane (or none, if it is detected in the back). In the scope of this work, we use eight sectors ranging from -140° to 140°.

4 Selecting from Multiple Landmarks

When using a partition as described in the previous section we end up with a set of landmarks for each sector. The size of each of these sets corresponds to the number of landmarks detected in this sector. Let us say that we partition the world into $n \in \mathbb{N}$ sectors and k_i $(1 \leq i \leq n)$ landmarks are detected in each sector i, $k_i \geq 0$. So we create a sequence of n sets:

$$\psi_l(s) = (L_1, L_2, \ldots, L_n) \quad , \quad L_i = l_{i_1}, l_{i_2}, \ldots, l_{i_{k_i}} \tag{2}$$

with l_{i_j} being detected landmarks in sector L_i. For the case of $k_i = 0$ (that is: no landmark is detected in sector i), $L_i = \emptyset$. Note that each landmark can map to exactly one sector, such that $l_{i_j} = l_{i'_{j'}}$ only holds if $i = i'$ and $j = j'$.

In theory, a set L_i can become arbitrarily large. In malicious environments it may contain all landmarks available in the world, but even in "normal" scenarios the number of landmarks in one sector can be higher than just a few in certain cases. The biggest problem of arbitrary lengths of the input set is that representations with many landmarks within one sector will not carry too much information anymore, as representations with slight variations within L_i will most likely not correspond to very different locations within the world, but to locations very close to each other. So we end up with an arbitrarily big number of different state representations for very similar situations—and that is exactly what we want to avoid, as this blows up the state space dramatically.

Let us look at an example: Fig. 3 shows a world with different distributions of landmarks, ranging from a small to a very high number of landmarks, and also one world with landmarks only at intersection points. We now take a look at the learning success

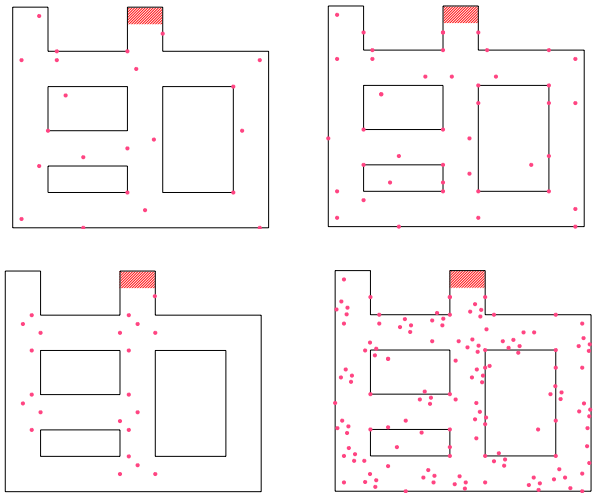

Fig. 3. Four environments with different distribution of landmarks, represented by dots: few landmarks (top left), many landmarks (top right), landmarks placed only at intersections (bottom left), and and absurdly high number of landmarks distributed all over the place (bottom right). The goal area is marked in all environments.

when applying RL with landmark-enhanced RLPR as introduced in Section 3 to learn to navigate from starting positions all over the world to the marked goal area.

For all the experiments within this work we used Watkins' Q(λ) algorithm [8]. During training, the agent uses an ϵ-greedy policy with $\epsilon = 0.15$ for exploration, that is, 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.

Success graphs of this experiment are shown in Fig. 4. Learning success critically depends on the number of landmarks—the more there are, the longer learning times are observed. With few landmarks the learning performance is extremely good. In particular it gets obvious that landmarks only at intersections are sufficient to succeed in the task. For the huge number of landmarks in the fourth world, a stable success rate of 100% has not been reached yet even after 50,000 learning episodes.

We will now investigate what to do if the number of detected landmarks is too big for high-performance learning. A solution to this problem is a stronger abstraction. We claim that a certain maximal amount of landmarks is sufficient for encoding the position of the agent. We reach this goal by applying an *aspectualization* on L_i, that is, we omit certain elements of L_i. A set L_i is abstracted to a set $L'_i \subseteq L_i$ with $|L'_i| \leq |L_i|$ and $|L'_i| \leq l_{\max}$ with $l_{\max} \in \mathbb{N}$ being the maximal number of landmarks we allow to be represented within a sector.

4.1 Strategies for Landmark Selection

Several possibilities exist to realize this aspectualization. In the following we will investigate some strategies to decide which landmarks to choose and which to drop from the state representation. We call the sequence $L_1, \ldots, L_n$ the *original observation* and $L'_1, \ldots, L'_n$ the *reduced observation*.

Random Landmark Selection. The easiest way of selecting a subset of detected landmarks in a sector is to choose the landmarks by random. This approach ensures a

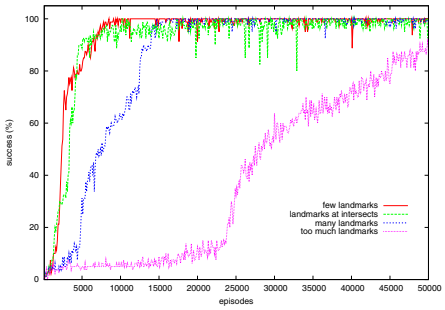

Fig. 4. Comparison: Goal finding success in environments with a different amount of landmarks

maximal number of $n \cdot l_{\text{max}}$ landmarks within the feature vector. However, all combinations of landmarks are equally distributed with respect to the frequency of observation. For example, if we perceive one configuration of k landmarks within a sector i from one and the same position, we will experience $\binom{k}{l_{\text{max}}}$ different observations L_i at this position over time. If $k > l_{\text{max}}$ for a larger number of states, then random landmark selection will most probably even increase the state space, making the strategy counterproductive.

Known Landmarks First. Another drawback of random landmark selection is that every landmark is treated equally, regardless of its relevance for the navigation process. Instead it would be beneficial to regard the "important" landmarks only by preferring landmarks that have been perceived in an earlier episode. This prevents the inclusion of rarely observed landmarks. However, after some time, a significantly high percentage of landmarks will be known and this method will converge to random landmark selection. Furthermore, the success of this method is hard to predict and depending on the environment. Summed up, the success of this approach is doubtful, and its effect imponderable, so the use of the "known landmark first" approach is discouraged.

Close Landmarks First. Alternatively, we could prefer those landmarks with the smallest distance to the robot. In contrast to the "known landmarks first" strategy the relevance of landmarks for being regarded in landmark selection now depends on the location of the robot. This inhibits the effect that a landmark is overseen just because too many known landmarks are around—the landmark will become a selection candidate as soon as the robot approaches it. Another important advantage of this strategy is that it establishes a unique mapping within the observation function, that is, each position of the robot is mapped to exactly one reduced observation: $\forall\,(x, y, \theta)\,\exists_1\, L'_1, \ldots, L'_n$.

4.2 SDALS: Structural Analysis for Landmark Selection

If we again take a lot at the "landmarks at intersects" curve in Fig. 4, it is obvious that only a few landmarks are sufficient, as long as they are appropriately located. In this case, the landmarks are exclusively located at corridor intersections. We call these locations *decision points*. If there is no decision to take, for example in a corridor, the structural information of the walls—the agent space—is sufficient to decide how to navigate. It does not matter to know which corridor the agent is exactly in to learn an appropriate action: going forward until the next intersection. Thus, landmark information is not necessary there. Furthermore, we gain a generalization effect, because structurally identical locations now share the same observation.

We now present a landmark selection approach called *Structural Decision Aware Landmark Selection* (SDALS). The main idea of SDALS is to ignore any problem space information at states that require unique actions because of agent space. In our case, landmark information is omitted when the structure of the surrounding walls alone triggers a certain action, for example, to go left when reaching the wall at a left turn.

RLPR (see Section 3.2) offers a unique possibility to distinguish structurally different states, that is, it subsumes locations with the same corridor structure (for example, a right turn) under the same RLPR values $\psi_r(s)$. To identify RLPR representations that require a unique action, we rely on background knowledge from an earlier learning task. It is possible to derive a *generalized strategy* π' with a Q-function $Q_{\pi'}$ that

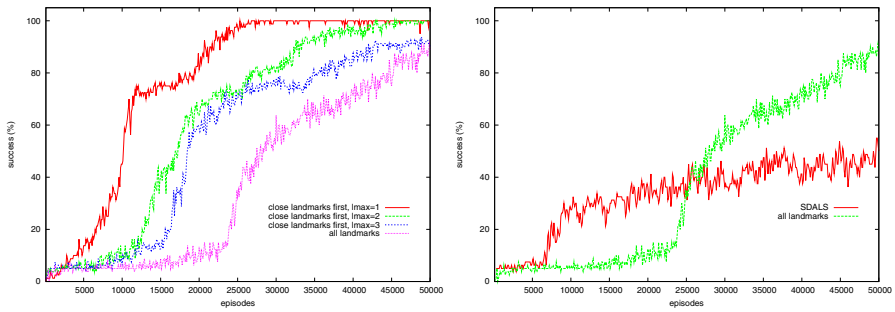

Fig. 5. Left: Comparison of "close landmarks first" vs. "all landmarks". Especially for low values of $l_{\max}$, "close landmarks first" shows a significant improvement. Right: Effect of the SDALS approach: After a very strong first learning phase, the learning progress disappears.

operates just on RLPR values $o' = \psi_r(s)$ derived from policies learned with landmark-enhanced RLPR [9]. This generalized strategy works in any environment, regardless of the concrete task to solve and delivers a generally sensible navigation action for any state. Furthermore it is possible to derive a *confidence value* $\mathrm{conf}(o')$ for any RLPR representation $o' = \psi_r(s)$ [9]:

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

In other words, $\mathrm{conf}(o')$ is a measure for an RLPR observation o' of how "certain" the policy is in choosing the best action according to the Q-values.

We now choose a set D of RLPR observations with the highest confidence values from a generalized strategy of a prior experiment. D contains those RLPR observations where one action is most strongly preferred over the other according their Q-values. States with this RLPR information are no decision points. SDALS utilizes this information to build a reduced observation $\psi_s(s)$ by ignoring landmark information at any state s where $\psi_r(s) \in D$:

$$\psi_s(s) = \begin{cases} (\psi_l(s), \psi_r(s)) & \text{if } \psi_r(s) \notin D \\ (\emptyset, \psi_r(s)) & \text{else} \end{cases}$$

SDALS only regards landmarks at places that have not shown unique decisions according to their RLPR representations in prior runs. This shrinks the state space effectively.

5 Experimental Evaluation

In this section we test the landmark selection strategies presented before. We concentrate on the scenario with the huge number of landmarks (Fig. 3). As expected, random landmark selection and "known landmarks first" do not show learning progress after 50,000 episodes in a scenario with this amount of landmarks.

Fig. 5 (left) shows the success of the "close landmarks first" strategy: It leads to a significant improvement in learning speed. While the "all landmarks" representation does not reach a stable success rate of 100% after 50,000 learning episodes, "known landmark first" manages to do so after about 25,000 ($l_{\max} = 1$) or 38,000 episodes ($l_{\max} = 2$). For $l_{\max} = 3$ the strategy shows a rapid learning in the early phase, but converges slower towards 100% success. Summed up, one landmark per sector is perfectly enough to fulfill the task when using "close landmarks first".

First tests reveal that SDALS provides a rapid learning improvement in the early learning phase, boosting the success to 30% and more after about only 5,000 episodes—that is better than any other selection strategy. The reason for this is the generalization effect: Structurally identical locations are subsumed under one single representation. Also, the number of the agent's collisions during training is reduced by more than 30% with SDALS, and in contrast to other strategies, the robot shows a negligible number of collisions in learned policies very early. However, after the strong early learning phase the learning performance shows very slow progress (see Fig.5 (right)).

6 Discussion

A thorough analysis of the SDALS experiments reveals that the agent learns a collision-free navigation strategy very fast, but fails to take the right decision to perform the last turn before reaching the goal state. It turns out that this area is very poorly explored, many states at the important turning point (the crossing) remain almost unvisited. The reason is that the robot mostly takes more or less identical ways through the corridors which he passes through without noticing landmarks thanks to SDALS.

So the generalizing effect of SDALS that enables a rapid learning of a generally sensible navigation behavior and high success rates in the early learning phase obviously leads into the exploration-exploitation dilemma. The bad performance of the other landmark selection strategies results in a higher exploration at decision points and the agent gathers knowledge which can be utilized later. SDALS, however, gets stuck in generally good, but not goal-oriented navigation patterns, and the simple ϵ-greedy exploration does not compensate this.

For this reason we need a better exploration behavior especially at the changeover between decision points and non-decision points. These places can easily be detected by SDALS. Future work has to investigate into that. Also it should be considered to abstract from Q-values (where the confidence values base on) and regard the actions emerging from them instead.

In general, too much knowledge is not always beneficial. Detecting more landmarks than needed requires a selection strategy, and it turns out that, when chosen appropriately, one landmark per sector is perfectly enough for the task at hand.

7 Conclusion

In this work we showed how a complex goal-directed robot navigation task in a continuous state space can be learned fast and efficient with the help of spatial abstraction. Detected landmarks can be represented qualitatively by sorting them to a circular order

of sectors around the robot. This allows for a suitable representation of problem space. In cases where too much landmarks are observed the choice of the landmark with the lowest distance to the robot shows being an efficient strategy for landmark selection.

The SDALS approach enables the agent to only consider landmarks when a decision has to be made. It is based on a structural analysis of generalized background data from previous tasks. At the current state of work, no 100% success in solving the goal finding task can be achieved yet. However, SDALS enables for a rapid early learning success and acquisition of a collision-free navigation very fast.

References

1. Levitt, T.S., Lawton, D.T.: Qualitative navigation for mobile robots. Artificial Intelligence 44, 305–360 (1990)
2. Lazanas, A., Latombe, J.C.: Landmark-based robot navigation. Algorithmica 13(5), 472–501 (1995)
3. Busquets, D., de Mántaras, R.L., Sierra, C., Dietterich, T.G.: Reinforcement learning for landmark-based robot navigation. In: Falcone, R., Barber, S., Korba, L., Singh, M.P. (eds.) AAMAS 2002. LNCS (LNAI), vol. 2631, pp. 841–843. Springer, Heidelberg (2003)
4. Frommberger, L.: Learning to behave in space: A qualitative spatial representation for robot navigation with reinforcement learning. International Journal on Artificial Intelligence Tools 17(3), 465–482 (2008)
5. Prescott, T.J.: Spatial representation for navigation in animats. Adaptive Behavior 4(2), 85–125 (1996)
6. Schlieder, C.: Representing visible locations for qualitative navigation. In: Carrete, N.P., Singh, M.G. (eds.) Qualitative Reasoning and Decision Technologies, Barcelona, Spain, pp. 523–532 (1993)
7. Konidaris, G.D., Barto, A.G.: Building portable options: Skill transfer in reinforcement learning. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI) (January 2007)
8. Watkins, C.: Learning from Delayed Rewards. PhD thesis, Cambridge University (1989)
9. Frommberger, L.: Generalization and transfer learning in noise-affected robot navigation tasks. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS (LNAI), vol. 4874, pp. 508–519. Springer, Heidelberg (2007)