UNIVERSITÄT DORTMUND
Fachbereich Informatik

Lehrstuhl VIII
Künstliche Intelligenz

Achieving Intelligence in Mobility – Incorporating Learning Capabilities in Real-World Mobile Robots

LS–8 Report 13

M. Kaiser **V. Klingspor** **J. del R. Millán** **M. Accame**
F. Wallner **R. Dillmann**

Dortmund, November 1994

Universität Dortmund
Fachbereich Informatik

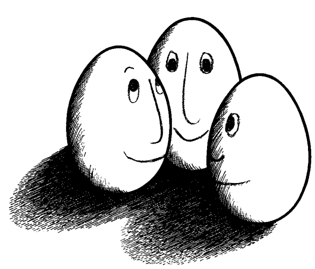
University of Dortmund
Computer Science Department

Forschungsberichte des Lehrstuhls VIII (KI)
Fachbereich Informatik
der Universität Dortmund

ISSN 0943-4135

Anforderungen an:

Universität Dortmund
Fachbereich Informatik
Lehrstuhl VIII
D-44221 Dortmund



Research Reports of the unit no. VIII (AI)
Computer Science Department
of the University of Dortmund

ISSN 0943-4135

Requests to:

University of Dortmund
Fachbereich Informatik
Lehrstuhl VIII
D-44221 Dortmund

e-mail: reports@ls8.informatik.uni-dortmund.de
ftp: ftp://ftp-ai.informatik.uni-dortmund.de/pub/Reports
www:
http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/REPORTS/reports.eng.html

Achieving Intelligence in Mobility – Incorporating Learning Capabilities in Real-World Mobile Robots

LS–8 Report 13

M. Kaiser[1] V. Klingspor J. del R. Millán[2] M. Accame[3]
F. Wallner[1] R. Dillmann[1]

Dortmund, November 1994

Universität Dortmund
Fachbereich Informatik

[1]University of Karlsruhe, IPR, D-76128 Karlsruhe, Germany
[2]European Commission, JRC, I-21020 Ispra, Italy
[3]University of Genova, DIBE, I-16145 Genova, Italy

This paper appeared in:
IEEE-Expert – Special Track on Intelligent Robotic Systems.
Vol. 10, No. 2, April 1995

Achieving Intelligence in Mobility – Incorporating Learning Capabilities in Real-World Mobile Robots

M. Kaiser[1], V. Klingspor[2], J. del R. Millán[3], M. Accame[4], F. Wallner[1], R. Dillmann[1]

[1]University of Karlsruhe, Institute for Real-Time Computer Systems & Robotics, D-76128 Karlsruhe, Germany
[2]University of Dortmund, LS Informatik VIII, D-44221 Dortmund, Germany
[3]European Commission, Joint Research Centre, Inst. for Systems Eng. & Informatics, I-21020 Ispra, Italy
[4]University of Genova, Dept. of Biophysical and Electrical Engineering, I-16145 Genova, Italy

Abstract

This paper presents an integrated approach to the application of Machine Learning techniques for the enhancement of mobile robots' skills. It identifies the learning tasks that can be observed throughout a number of typical applications of mobile robots and puts those tasks into perspective with respect to both existing and newly developed learning techniques. The actual realization of the approach has been carried out on the two mobile robots PRIAMOS and TESEO, which are both operating in a real office environment. In this context, several experimental results are presented.

1 Introduction

Research on mobile robots has always attracted considerable attention from both the Robotics and the AI community. A lot of work has been devoted to the solution of the basic problems involved in their application, such as the planning and execution of collision free motions. In general, control issues are to be taken into account as well as navigation and perception capabilities. Adaptivity and the compliance of the robot to safety requirements are equally important topics.

Currently, mobile robots are leaving the laboratories and entering the real world. In Europe, Japan, and the U.S., they are becoming actual products. Driverless transport systems, usually guided by wire, are employed in automated factories, such as those of the Caterpillar Corporation's factories in Europe. They are used for plant supervision, and increasingly successful in service tasks such as health care. TRC's HelpMate is probably the most popular service robot so far and is used in almost 100 hospitals in the U.S. and Europe.

However, employing mobile robots in the real-world for everyday tasks puts much stronger requirements on their communicativity, adaptivity, and safety. Consequently, the complexity of the robot's control software is dramatically increased. The loop between the robot's perceptions and its actions has to be closed on several levels of abstraction, such as the level of reflexes or the geometrical planning level. Such a hierarchy of increasingly abstract *situation-action rules* requires to build appropriate models of perceptions and actions, that must, however, always be *grounded*, i.e., that can be directly mapped to the robot's basic sensing and motion capabilities. Building these rules means to codify knowledge related to both the task and robot, i.e., to actually *program* the robot.

Additionally, the specification of the actual tasks the robot has to perform should take place on a level that's easily managable by an average user. This requires the robot to learn the user's language, i.e., to learn to transform its own perceptions and actions into symbols the user can understand, and to compile the user's task specification into a set of operational, i.e., directly executable commands. The omnipresent problems of *symbol grounding* and *signal to symbol transformation* become particularly challenging if the robot is adaptive, i.e., if it is changing its behaviour according to both the user's needs and changes in the environment.

Throughout this paper, it will be described how Machine Learning techniques can be employed to solve both problems. In particular, it will be shown how increasingly abstract representations of the robot's perceptions and actions can be built in order to obtain a symbolic description of what the robot knows and is able to do. As this task in its generality is fairly complex, it is necessary to exactly identify those sub-problems that can actually be solved efficiently by employing a learning method, and to separate those for which good classical solutions exist. Therefore, this paper is organized as follows. First, the mobile robots PRIAMOS and TESEO, which have been used to experimentally evaluate the approach, are presented, and the *basic* requirements mobile robots have to fulfill are discussed. To enable a robot to solve a *complex* problem, solutions for several *learning tasks* must be found. These learning tasks are therefore identified and learning techniques appropriate for their solution are presented. Finally, the selected learning techniques and their combination with the conventional components is experimentally evaluated, and conclusions drawn from the several presented lines of research are given.

2 The mobile robots PRIAMOS and TESEO

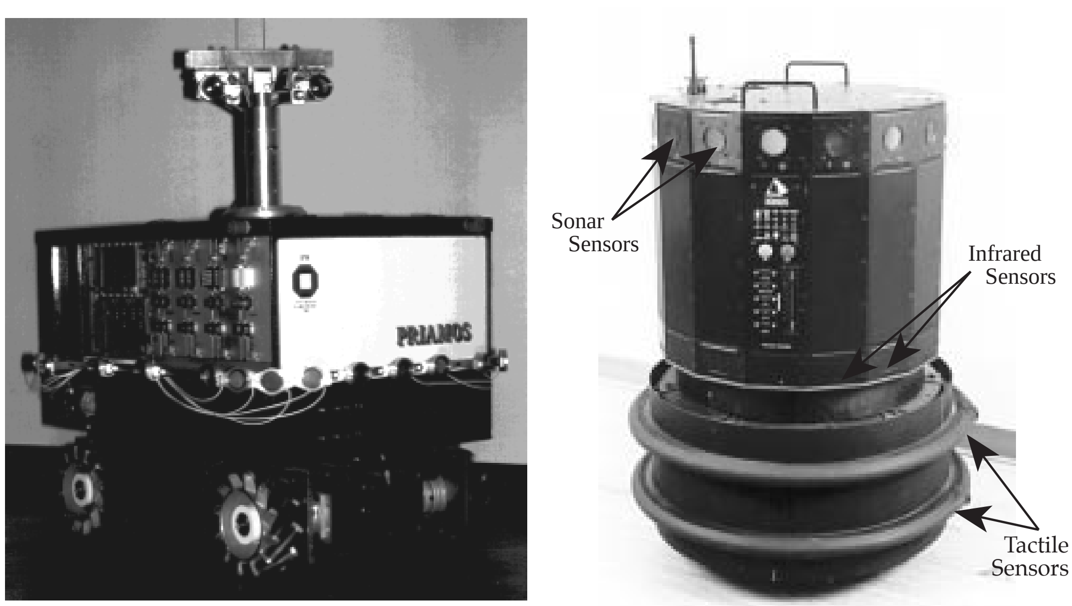


Figure 1: *The mobile robots PRIAMOS (left) and TESEO.*

The mobile robot PRIAMOS (figure 1 left) is used as a platform for experiments on perception, navigation, and on the application of both subsymbolic and symbolic learning techniques for navigation tasks. PRIAMOS is a mobile system with three degrees of freedom, i.e., motion in longitudinal and transverse direction and rotation around the center of the vehicle. This is accomplished by the use of four Mecanum wheels, each one driven separately.

Currently, the robot is equipped with 24 ultrasonic sensors, three of which are mounted at each of the robot's sides, and three at each of the robot's edges. The sensor control and processing system is able to process the input of all 24 sensors up to five times per second. Other sensors, such as a stereo vision system, can be mounted on the upper cover plate using a flexible mounting system.

TESEO is a commercial NOMAD 200 robot. This robot has a radius of 9 inches, a height of 30 inches, and features three independent motors. The first motor translates the three wheels of the robot together. The second one steers the wheels together. The third motor rotates the turret of the robot. The robot can only translate along the forward and backward directions along which the three wheels are aligned. It has a zero gyro-radius, i.e. it can steer around its center. The version of the Nomad 200 in use has three sensory systems, namely tactile, infrared, and ultrasonic (see figure 1 right).

3 Basic requirements for mobile robots

Despite the diversity of mobile robot applications and of the environment the robot usually operates in, there are a some principal capabilities a mobile system must exhibit. The basic task a mobile robot

has to solve is to move safely, i.e., collision-free and goal-oriented in order to arrive at its destination as quickly as possible while avoiding objects and humans in its vicinity. This general task results in two basic requirements. The first is the availability of one or more sensor systems that are capable to perceive environmental information [Elfes, 1987]. To ensure collision-free motion, mobile robots are usually equipped with a set of distance-measuring sensors based on laser, ultrasound, or infrared (see also figure 1). The information provided by these sensors can be processed quickly, hence they are ideally suitable for continuously monitoring the environment in real-time.
The second requirement is the existence of a control link between these sensory systems and the robot's actuators, such that the robot's actions can be altered with respect to its perceptions and, possibly, with respect to the given task. This control link is established via situation-action-rules that might be given in an arbitrary representation (such as a conventional controller, a neural network, or as fuzzy rules).

Why is learning important?

The more complex the robot's tasks become, the more complex situations, actions, and rules will have to be. To handle this increasing complexity, it is desirable to build increasingly abstract representations of the robot's perceptions and actions as well as their correspondance, i.e., the situation-action rules. From the developer's point of view, building these abstract representations in fact means to enable the robot to solve complex tasks, i.e., to actually program it. For a robot user, the highest level representation should represent a symbolic description of the robot's world and action knowledge, i.e., a description that can be used for communication with the robot.
If all situations the robot might encounter can be foreseen at development time, it would be possible, although extremly costly, to equip the robot with a program that is able to handle these situations appropriately. Obviously, this approach becomes intractable as soon as the real world, i.e., noise in perceptions and actions, changing user needs, and a changing environment, have to be taken into account. Hence, the robot should be able to exploit the experiences it gains during operation, i.e., to *learn* from its experiences, in order to generate and maintain abstract models of perceptions, actions, and situation-action rules. This task, however, is a fairly complex one and several subtasks can be identified within it. For each of these subtasks, specific approaches and methodologies that are used are now presented.

4 Learning tasks in mobile robot applications

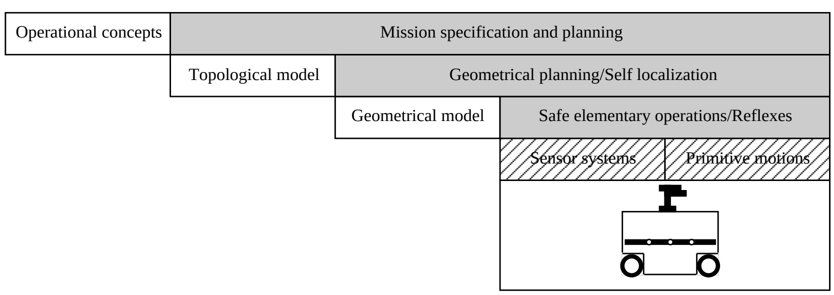


Figure 2: *Principal structure of the robot control system.*

Basically, two different kinds of learning tasks have to be solved. The first kind of tasks is devoted to the generation of an increasingly abstract representation of the robot's environment, i.e., to provide the means to describe *situations* on several levels of abstraction. The first step towards such descriptions is to **build basic features from sensory data**. This is an important aspect especially regarding the application of symbolic learning techniques for the generation of concept descriptions. While the features are given in most Machine Learning applications, the raw distance information that is provided by ultrasonic sensors, such as those used in PRIAMOS and TESEO, is neither appropriate as input for a symbolic learning technique, nor can it directly be used to perform the task of mapping the environment. Therefore, it is necessary to calculate basic features that are relieved from sensor noise and errors introduced by the environment.

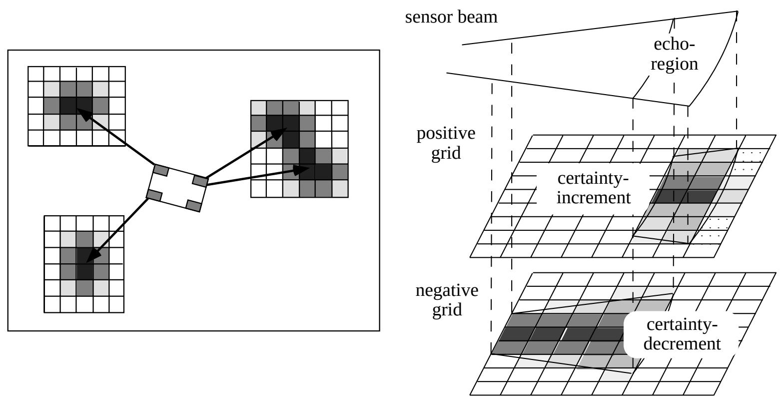


Figure 3: Local grids (left) and generation of positive (indicating occupied areas) and negative (indicating free areas) grids (right).

The procedure used to generate and refine a map of the environment on the base of sonar and infrared measurements extends two approaches that can be found in the literature. Firstly, sonar measurements are collected in occupancy grids [Elfes, 1989] that are local with repect to the actual area of perception (see figure 3). Secondly, as soon as a sufficient degree of safety with respect to the integrated measurements has been achieved, a parametric description of the measured object is generated (see also, for example, [Leonard & Durrant-Whyte, 1992]). More specifically, as every object in the environment is represented by a set of interconnected line segments, generating a parametric description from a certainty grid means to estimate the parameters describing each segment (orientation, distance to the origin) from that grid. The actual estimation procedure selects the cell which features the highest probability of being occupied and successively looks for the neighbour cell which causes the lowest change in probability. During each step, the parameters are re-estimated by means of a Kalman filter. As soon as enough data are available for a reliable estimation, the segment is added to the robot's map.
The choice of the features used for concept learning is crucial, because these features determine the expressiveness of the symbol system. Concepts that cannot be described based by means of existing features cannot be described either with newly built intermediate concepts, since such concepts only simplify the representation, but they do not enlarge the "closure of existing symbols" [Wrobel, 1991]. Therefore, the parametric descriptions obtained from integrating distance measurements are not suitable for learning operational concepts, as they describe situations with respect to the world and not as the robot *itself* perceives them. Basic features as they are required to generate the latter descriptions represent measurements of one or more sensors taken over a finite amount of time. For example, $\mathsf{stable}(s_1,t_1,t_2)$ denotes that sensor s_1 measured a constant distance during the time interval $[t_1, t_2]$. The actual method to construct these features is to **group gradients**, i.e., to collect measurements that show a constant development over time into a single feature. As the quality of the features significantly influences the quality of the concepts learned on the base of these features, a control loop is realized in order to adapt the parameters (e.g., thresholds) of the grouping algorithm. The basic features have to contain exactly those details that are necessary to recognize and distinguish operational concepts. As this property of the features can only be determined by employing them for concept learning, the learning algorithm can be described as follows:

1. Start with a predefined set of parameters.

2. Calculate the basic features based on the function determined by the actual parameters.

3. Try to learn concept descriptions.

4. **If** concept learning succeeds satisfactory **then** stop.

5. **Else** adapt the parameter set according to the evaluation of concept learning to construct more or less abstract features. Iterate at step 2.

The next task is to combine the calculated basic features and to **generate operational concept descriptions on the base of these features.** In particular, this task aims at learning a hierarchy of features which can be expressed by a set of rules. Since lower level features directly calculated from the sensory input occur in rules' premises, an individual higher level feature can be inferred directly from the sensor inputs via a rule chain. The highest level features are concept descriptions, and instances of these concepts are objects that are physically present in the environment. These objects must be linked to the map holding the geometrical information about the world. The generated descriptions are necessarily operational, since they are directly derived from the basic features which are calculated from the sensorial input. The main difference to earlier classification approaches like [DeJong & Mooney, 1986] is the fact that operational concepts do not only use perceptions to classify the object. They also relate these perceptions to the actions that are performed by the robot. To solve this learning task, inductive learning in restricted first order logic [Kietz & Wrobel, 1992] is employed. This learning method yields concept descriptions like

$$\mathsf{stable}(s_1,t_1,t_2)\ \&\ \mathsf{incr\_peak}(s_1,t_2,t_3)\ \&\ \mathsf{stable}(s_1,t_3,t_4)$$
$$\rightarrow \mathsf{s\_jump}(s_1,t_1,t_4).$$

Given the hierarchical representation of the environment that comprises raw sensorial data as well as operational concepts in terms of rules, the next step is to enable the robot to use these representations. The first learning task regarding this *acquisition of action knowledge* is the **acquisition of a set of safe elementary operations.** These elementary operations express the robot's capabilities on a low level, where they also serve to close the loop between the robot's sensors and its actuators. More specifically, elementary operations are defined as *operations that are realized through a direct coupling between the robot's sensors and its actuators and require a constant focus of attention during execution.* This focus of attention will in most cases be a specific object (such as a recharging station that must be approached, or a door that is to be passed), but it might as well be another robot (if the task is, for instance, tracking), or simply an obstacle that is to be avoided without moving away from the current goal.
Though there might be only a limited number of elementary operations that can be imagined, the realization of each of these operations will have to take the robot's actual environment into account. Therefore, it is desirable to have the robot *learn* this realization, starting from some description of what it is expected to do.
One possibility to learn this realization is to follow the line of reinforcement learning [Barto *et al.*, 1983, Williams, 1992]. The robot then has to learn suitable reactions directly from a feedback signal, which, in the simplest case, is positive as soon as the robot achieves the goal (e.g., it passes the door), and is negative otherwise. However, the use of reinforcement learning in real robotic systems is limited, since it usually requires an extremely long training time and the exploration involved in the learning process, i.e., the execution of random actions in order to explore the action space and to achieve a better solution, is not feasible due to safety reasons.
In this context, to learn safe elementary operations means to associate with every perceived situation the action that maximizes the total reward in the long term, i.e., that finally yields the most efficient realization of the corresponding elementary operation. To solve this task, two different learning rules, namely *temporal difference (TD) methods* [Sutton, 1988] and *associative search (AS)* [Barto *et al.*, 1983, Williams, 1992] are used, where TD methods are employed to predict the total future reward and AS is used to update the situation-action mapping based on the estimation given by TD. In order to allow for quick convergence, incremental learning, and collision avoidance, three extensions are made to the above approaches. First, instead of learning from scratch, a fixed set of *basic reflexes* is employed every time the neural network which represents the situation-action rules fails to provide an action on the base of the current situation. These basic reflexes correspond to previous elemental knowledge about the task (such as *if the goal is left then move to the left*). The new situation-action rule obtained by associating the perceived situation with the basic reflex is subsequently tuned by reinforcement learning. Second, to allow for incremental learning, a resource-allocating procedure is used that builds a modular network. The rules represented in a single module map similar sensory inputs into similar actions and, in addition, they have similar long-term consequences, such that improvements on a module will not negatively alter

other unrelated modules. An elementary operation might therefore consist of several modules, each of which aims at achieving the same long term goal but allows to start from a different situation. The third improvement is the use of a *counter-based scheme* to concentrate the search around the best currently known actions, depending on the specific experience of the robot. This improvement allows to avoid dangerous actions resulting from exploration.

The second option that can be taken is to learn elementary operations from examples. This option is particularly interesting if an elementary operation is linked to a specific operational concept. In this case, the features describing the concept become the conditions that apply before, during, and after the execution of the elementary operation, as in

standing(s_1,t_1,t_2,front_side) & parallel_moving(s_1,t_2,t_3,left_and_right) & standing(s_1,t_3,t_4,back_side)
$\rightarrow$ move_through_door(s_1,t_1,t_4),

such that the operational concepts can be seen as operators for situated actions (see [Ringle, 1993] for a discussion on this topic).

Basically, learning an elementary operation from examples requires to approximate a function mapping the sensorial input to robot motions. The task of function approximation is usually related to methods such as neural networks [Miller *et al.*, 1990], regression trees [Breiman *et al.*, 1984], or the generation of fuzzy rules [Wang & Mendel, 1992]. These methods, however, do in general not yield a result that is easily understandable for a human user. Therefore, under the roof of operational concept,s a specific elementary operation is seen as an *action feature* and learned and represented in the same way as the perceptual features described before.

As the safe elementary operations are local in nature, complex tasks that are to be performed by the robot can only be solved by employing several elementary operations subsequently. To find the best sequence of elementary operations is a typical planning problem. As the main task of a mobile robot is, however, to move from a start to a goal position, this planning problem must be solved in a hierarchical manner. First, planning takes place on the base of the geometrical model of the world in order to generate a path that is physically feasible. Second, if more than one path exists, that one fulfilling a given selection criterion (that might be depending on the choice of the elementary operations) is selected. Third, while the robot is actually moving along the selected path, the appropriate elementary operation should be active at any time.

To perform this kind of hierarchical planning requires knowledge about the world that goes beyond the geometrical world model. In fact, on the base of the geometrical world model only the robot is not able to consider any specific elementary operation at planning time. Therefore, it is necessary to **acquire schemata that express the robot's abilities on a high level**. These schemata, which represent situation-action rules on the highest level of abstraction, capture the correspondance between specific locations, non-geometrical parameters of the world, and elementary operations. In the machine learning community, learning schemata is an application domain of deductive approaches such as EBL or EBG (see [Ellman, 1989] for an overview). Based on background knowledge, these methods transform a particular problem solution into a schema (a macro) or extract heuristics (meta-knowledge) for problem solving. The main difference of the approach presented here is the use of a single entity, the *topological graph*, to represent all schemata. By labeling each edge of the graph with the elementary operation that was employed during the execution of a mission, the initially only geometrical information stored in the topological graph is constantly extended. However, as the execution of different mission will result in labeling a single edge several times, an inductive component is required as well in order to generate a label that is consistent with the complete experience of the robot. In that respect, what is required is a multi-strategy approach, which looks as follows:

1. On the base of the geometrical world model, generate the topological graph, e.g., by employing Voronoi diagrams (see [Hwang & Ahuja, 1992] for an overview on motion planning algorithms).

2. Label each edge with the local geometrical planning with collision avoidance EO.

3. Additionally, initialize the costs of each edge to be the geometrical length of that edge.

4. Whenever, during a mission, a specific elementary operation has been employed while the robot was moving along a specific edge, label that edge with the EO. If the edge was already labeled with

Figure 4: Parameterization of the Canny algorithm.

Firstly, **optimal sensor parameterizations** are to be learned, in order to obtain information of maximum quality. This aspect becomes especially important if a visual sensor is used. Based on a quality evaluation of acquired images, the camera parameters (focus, camera aperture, electronic gain, and black level) are adapted until an acceptable image quality has been obtained. The actual adaptation strategy can be represented in several ways, e.g., as rules [Ooi *et al.*, 1990] or in terms of neural networks that have been trained using a set of a-priori acquired images and the corresponding, manually set camera parameters. In our case, the regulation unit has been realized by means of a three layer feedforward network trained on quality evaluations and acquisition parameters obtained from more than 950 images taken in an office environment. The actual training procedure is an enhanced backpropagation algorithm. At run-time, the network is fed with the quality evaluation of a taken image, from which it calculates a new set of camera parameters.

As the efforts made in the acquisition phase pay only off if the edge detection algorithm is setup appropriately, neural regulation is also employed to parameterize the Canny algorithm [Canny, 1986] used for the edge extraction. The Canny algorithm is mainly a hysteresis threshold done on the gradient magnitude of the image. The variation of the values of the thresholds permits to select the number and the length of the edges, so the network has been designed to regulate the hysteresis mechanism. More specifically, the network has been trained to respond with the correct thresholds when fed with features computed on the base of image's pixelwise gradients (see also figure 4).

The second task to be solved is to provide the robot with means to correct its estimation of its position in the world. This **localization** task has been extensively studied (see, for instance, [Crowley, 1989, Leonard & Durrant-Whyte, 1992]), and good solutions exist. Also the method employed in this work is a "classical" one based on a prediction of sensor measurements.

5 Learning robots in the real world

All learning tasks described in the previous section are essential to achieve flexibility, good performance, and robustness for a mobile robot operating in the real world. However, their solution in the framework of standard robot control architectures is not straightforward. Usually, these architectures feature distinct handling of object knowledge and action knowledge, with the latter being represented only implicitely in the form of executable code. Since good representations are a key issue for the successful application of machine learning techniques, the experimental results given in this section take this aspect into account.

5.1 Generation of safe elementary operations

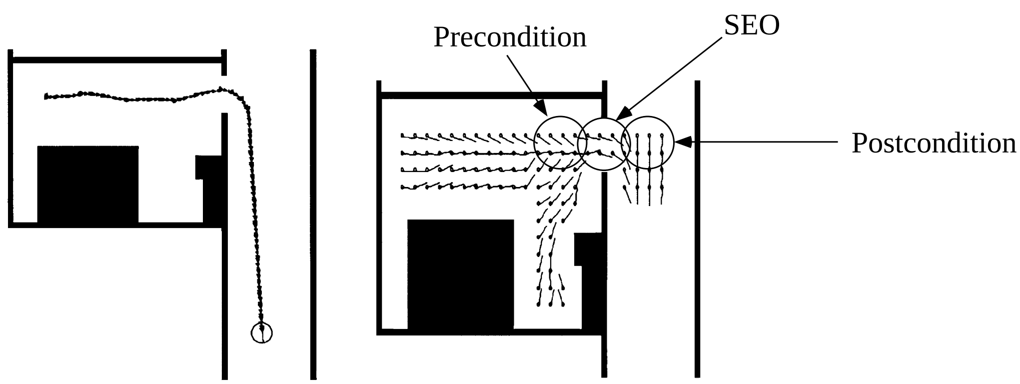


Figure 5: *Robot performance after 10 trials (left) and generalization capabilities/resulting EO (right).*

If an elementary operation such as the passing of a door is to be learned by means of reinforcement learning, a suitable reinforcement signal is the most important thing to provide. In this experiment, the robot was asked to reach a goal position in the corridor, starting from inside an office. This task is solved *very rapidly*. The robot reaches the goal efficiently and without colliding after traveling 10 times from the starting location to the desired goal (see figure 5 left). The total length of the first trajectory is approximately 13 meters while the length of the trajectory generated after the suitable sequence of reactions has been learned is about 10 meters.
Achieving the desired goal implicitely builds the elementary operation move_through_door (see figure 5 right). After the elementary operation has been built, it can be used as an action feature in the frame of an operational concept. Also, it can be assigned to particular locations in the environment.

5.2 Generating operational descriptions

The existence of safe elementary operations alone does not qualify mobile robots for real-world applications. It is important to be able to relate these elementary operations to certain objects (such as a door). Hence, it is necessary to find means to describe objects in a way that is robust and allows on-line matching. Operational concepts which represent objects both in terms of robot actions and sensory inputs serve this purpose.
The performed experiments aimed at learning descriptions for the concepts along_door and through_door. Using a simulated basic environment, 28 traces in a simple room, most of them are paths along or through the doorway in different directions, were generated (see figure 6). For every trace, about 30 measurements were taken by each of PRIAMOS' 24 sensors, yielding 17472 single measurements. From the sensory data, basic features such as stable, increasing, decr_peak were calculated. These features describe the perceptions

	s_line	s_jump	s_convex	s_concave
#rules	62	36	21	10
coverage	91%	82%	60%	87%

	sg_line	sg_jump	sg_convex	sg_concave
#rules	43	36	2	3
coverage	87.7%	89.7%	84.9%	89.0%

	through_door (parallel)	through_door (diagonal)
#rules	2	1
coverage	100%	100%

Table 1: Learning results

of the robot on an abstract level and are the input to the actual learning algorithm. Learning is based on examples given for each concept to be learned. To generate these examples, direct teaching of the robot can be used as well as simulation. The applied learning algorithm, GRDT, is able to syntactically restrict the hypothesis space, i.e., the space of learnable rules, by learning instantiations of user-given rule schemata [Kietz & Wrobel, 1992]. In addition to the examples, background knowledge like information about sensor configurations was given. GRDT starts with the most general hypothesis and specializes it until the given positive and negative examples are sufficiently well described.

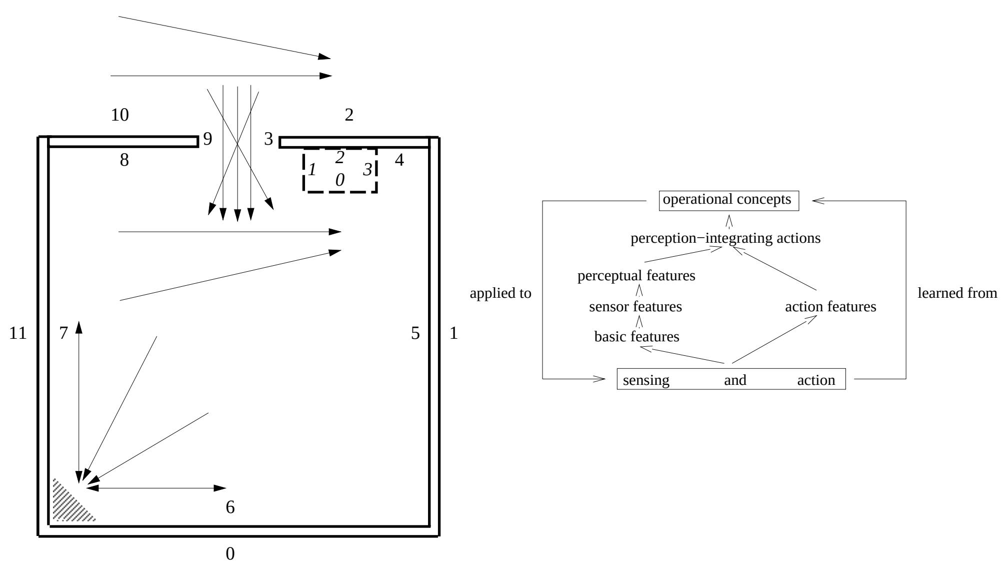


Figure 6: *Room used to generate training examples and corresponding trajectories (left); Hierarchically structured concepts (right).*

Several learning tests for the three levels of perceptual features, single sensor features, sensor group features, and action-integrating perceptual features (see also figure 6) were performed. Table 1 shows the results of learning at all levels. The last line of each part of the table displays the quality of the learned rule set, represented by the quotient of input examples covered by learned rules and all input examples.

As the aim is to recognize and classify sensed objects, the learned rules must be tested with the original measurements. The coverage of the intermediate concepts sg_jump and sg_line is 64 % in both cases. This is a really good result, since these concepts are derived in two steps via sensor features and the coverage of sensor features is around 80 to 90 %. From these intermediate concepts, the system derives through_door in all given cases. Nonetheless, one of the rules derives the goal concept through_door in some additional cases. However, learning this rule could be easily suppressed by changing the confirmation criterion.

5.3 Mapping and localization

The self-localization and map-building tasks use similar features as the learning of operational concept descriptions, but employ a different approach. The biggest problem that has to be dealt with for self-localization as well as for the generation of a useful, i.e., accurate map, are the uncertainties and inaccuracies that arise from the imperfectness of the world, the robot, and its sensors. The adopted approach combines the integration of the robot's movements over time (resulting in an estimation of the current position which suffers from the uncertainties in the internal sensors) with the use of (static) reference objects that serve as beacons for the position estimation. Obviously, unknown objects or objects whose position is not known to the robot cannot be used for position estimation. Hence, recalibrating the robot in a totally unknown environment is only possible by some sort of backtracking, using the initial position and measurements as reference values.

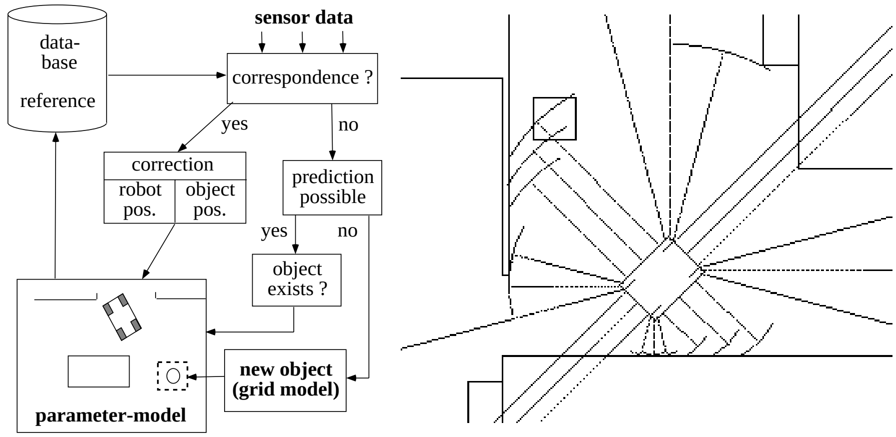


Figure 7: *Application of local model and geometrical world model (left) and screendump from ultrasonic sensor prediction (right).*

For the actual update of the robot's position estimation, basic object features (such as line segments) must be calculated from the sensorial input. These features are matched with the knowledge contained in the geometrical map. The basic operation is to derive a parameterized description of a perceived object and to match this description with the geometrical world model (see also figure 7 left). This method allows for significant reduction of the positioning error. For example, after moving over a distance of four meters, including a right angle turn, the position error of PRIAMOS in (x, y, θ) without self-localization was $(17\text{ cm}, 9\text{ cm}, 0.20^o)$. Applying the localization method yielded a significant improvement towards $(0.6\text{cm}, 3.5\text{cm}, 0.006^o)$. To test the object localization, a known object inside a room was moved by $(40\text{ cm}, 40\text{ cm}, 9.0^o)$. The object localization method started with these values and was able to reduce the uncertainty in the object's position to $(6\text{ cm}, 6\text{ cm}, 1.0^o)$.

5.4 Regulation of sensor parameters

In addition to features calculated from the sensory input of PRIAMOS' and TESEO's distance sensors, visual information can be used to verify predictions made on the base of the aforementioned rules. Also, visual input is necessary in case the prediction and the robot's world model differ (e.g., if a door has been closed without notifying the robot, see also figure 8).

To this aim, it is necessary that the vision system is able to deliver information that is reliable under changing environmental conditions. The key to achieve this reliability is the regulation of the image acquisition and processing parameters. The automatic vision system designed for this task performs an iterative adjustment of the image acquisition parameters (see figure 8).

After the image has been acquired, the visual sensing system generates a symbolic description of the perceived scene by means of the neurally regulated Canny algorithm (see figure 9). This description can then be used for robot localization as well as for extending the world model and the operational concepts towards the third dimension.

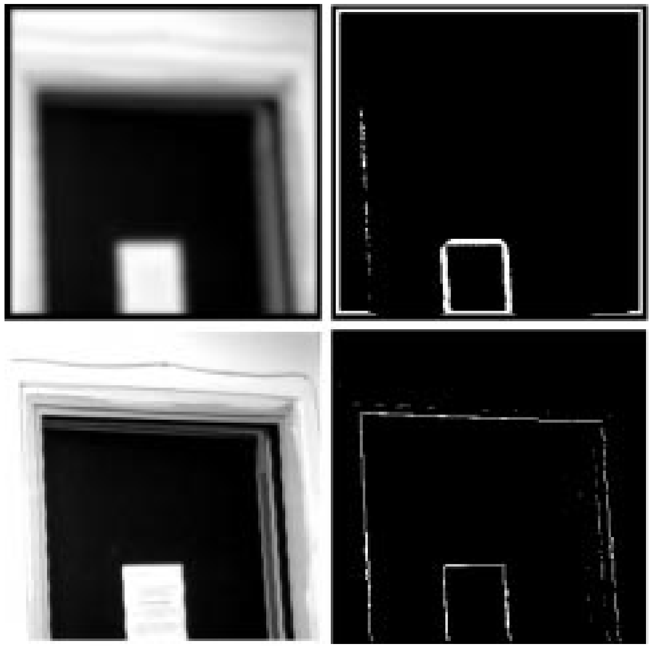


Figure 8: *Upper part of an open door with corridor behind: a) image acquired with random regulation; b) resulting edges after processing; c) image acquired after regulation loop (one-shot); d) final processed data.*

5.5 Coordination and exploitation of global experience

The elementary operations, the reliable sensor systems that they require, the basic tools for navigation such as mapping and self localization, and operational, i.e. on-line applicable rules to detect key situations and task-related objects that have been described so far are key components of a mobile robot that has to operate in the real world. However, to use these components efficiently, the robot must take the global context and the specified tasks into account, i.e., it must plan its movements and define conditions for the activation of particular elementary operations, sensor systems, and operational concept descriptions. The actual experiences the robot gains while performing a mission are represented by means of the topological graph (see figure 10). These experiences comprise the best topological path with respect to geometrically given start and goal, the detection of a specific object by means of the perceptual features describing an operational concept, and the actual costs and elementary operation associated to a particular edge. Whenever the robot is requested to move perform a particular task (e.g., to move to (1000,800,90), then pass through the first door on the left side and report the position), a first planning

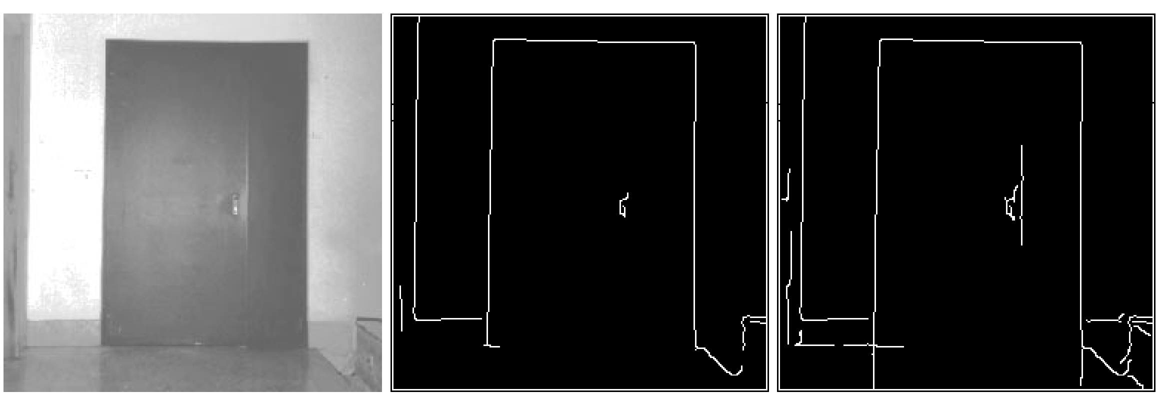


Figure 9: *Door: a) image acquired after regulation loop; b) neural based edge extraction; c) fixed value edge extraction.*

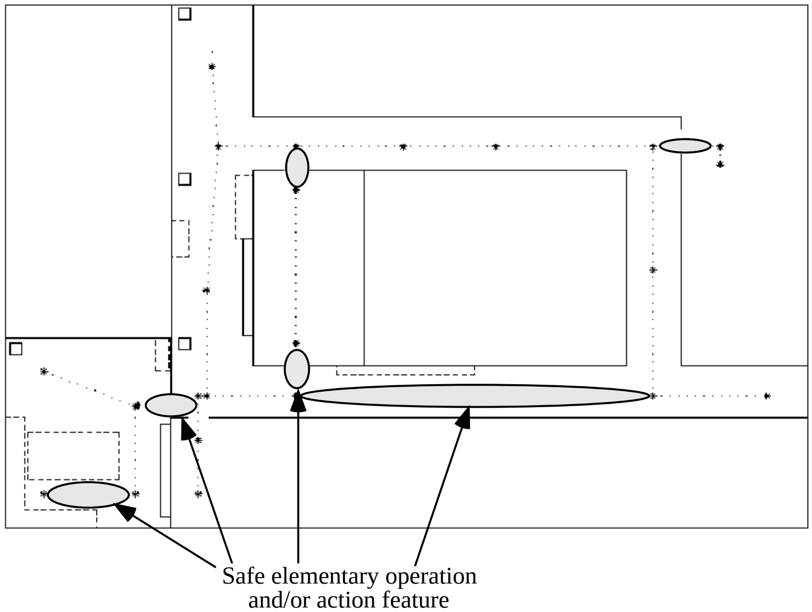


Figure 10: *Geometrical map of the office environment used for experiments with topological graph and indications of associated operations.*

step is performed on the base of the topological graph. A *branch-and-bound* algorithm selects the best path, based on the edge cost estimations. During the actual execution of the path, the robot uses the elementary operation associated to a particular edge. In case no specific EO has been assigned to an edge, local geometrical planning takes place (see also figure 11).

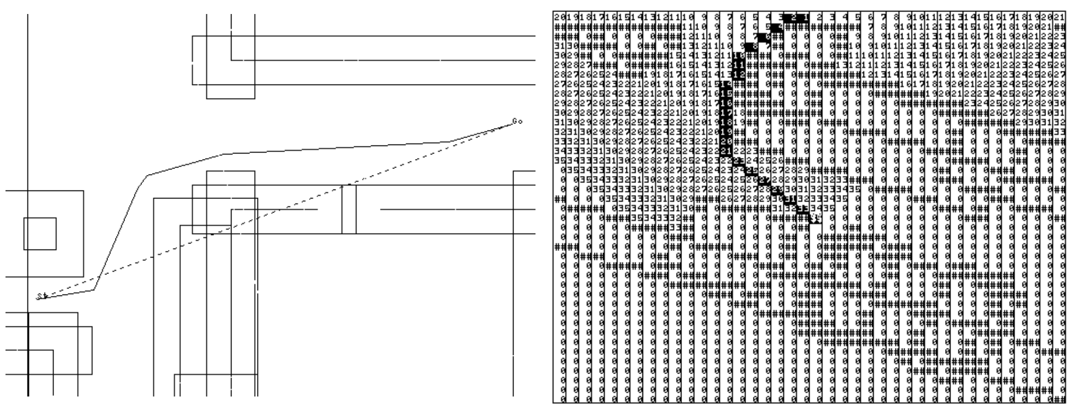


Figure 11: A geometric planning example. Shown is a ground plan of an office environment as well as the result of a planning cycle. (#: obstacles; $$: robot position; filled squares indicate the path)

6 Discussion

Most of the several learning techniques presented so far aim at building increasingly complex representations of a mobile robot's perceptions and actions. Although some of the approaches are based on quite different methodologies, the solution of all tasks that were investigated required specific attention with respect to these key issues:

Complexity and evaluation: One of the most important results obtained from this research is the necessity to accurately analyze the application domain before applying any kind of learning, in order to determine *learnable* subtasks, i.e., in order to obtain subtasks for which examples can be generated in sufficient numbers without much effort. Related to this observation is the aspect of evaluation. Small-scale tasks usually feature well-defined evaluation criteria (such as the quality of egdes, the number and coverage of generated rules, etc.). Large scale tasks, like the acquisition of schemata, are difficult, if not impossible to evaluate on an objective basis. Mostly, the evaluation will have to be relative to some a-priori status, i.e., *does the system do what is desired, and how much effort would probably be necessary to realize these capabilities without learning?*

Knowledge acquisition: To overcome the knowledge acquisition bottleneck is one of the major tasks involved in any application of Machine Learning. However, a typical effect that was discovered during our research was that the necessity to acquire knowledge about the application was replaced by the necessity to acquire knowledge *about* the learning methodology and knowledge *for* the learning methodology, e.g., in terms of examples. While the latter problem is the less critical the less complex the individual learning tasks are, the former can only be solved by developing more robust and easier to handle learning algorithms, and by explicitely considering real-world conditions during their design.

Application area: The application area of mobile robotics proved to be very suitable for the evaluation of several kinds of Machine Learning algorithms. This might be due to the fact that it features two well-defined problem areas, namely mapping and navigation, that can relatively easily be formulated on several levels of abstraction. Also, the actions the robot can perform in this environment are limited. To achieve this level of granularity, e.g., with respect to an assembly robot, would require to severly restrict the actual capabilities of the machine.

7 Summary and further work

If mobile robots are to be employed in the real world, they must be able to gain profit from their experience. They must be able to adapt to their environment and to the specific conditions of use as they are given by the application and the user. In such a scenario, learning becomes evident. A learning robot can relieve the robot designer from cumbersome programming tasks, and it can support the robot user to customize itself for his or her specific needs. Throughout this paper, it has been shown that several areas of machine learning research, both devoted to subsymbolic and to symbolic learning techniques, can provide substantial help in designing learning robots.
However, the effort involved in obtaining the described results is still high. Most of the employed algorithms are still too difficult to be handled by unexperienced users and must be made much more robust and transparent. The results they produce must be presented in such a way that they can easily be understood and, possibly, even verified automatically. Only then it can be expected that complex learning systems, such as learning robots, will eventually be useful for everyday applications.

Acknowledgement

This work has been supported by the ESPRIT Project 7274 "B-Learn II". It has been performed at the Instituto de Cibernètica, Prof. Luis Basañez and Prof. Carme Torras, Universitat Politecnica de Catalunya, Spain, at the Lehrstuhl Informatik VIII, Prof. Dr. Katharina Morik, Department of Computer Science, University of Dortmund, Germany, at the Department of Biophysical and Electronic Engineering, Prof. G. Vernazza, Universita di Genova, Italy, and at the Institute for Real-Time Computer Systems & Robotics, Prof. Dr.-Ing. U. Rembold and Prof. Dr.-Ing. R. Dillmann, Department of Computer Science, University of Karlsruhe, Germany. The authors would like to thank all the people involved in the presented work and the anonymous reviewers for their helpful comments.

References

[Barto *et al.*, 1983] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 835–846, 1983.

[Breiman *et al.*, 1984] L. Breiman, J.H. Friedman, R.A. Olsen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth and Brooks, 1984.

[Canny, 1986] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679 – 698, 1986.

[Crowley, 1989] J. L. Crowley. World modelling and position estimation for a mobile robot using ultrasonic ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 674–680, 1989.

[DeJong & Mooney, 1986] G. DeJong and R. Mooney. Explanation-based learning - an alternative view. *Machine Learning*, pages 145 – 176, 1986.

[Elfes, 1987] A. Elfes. Sonar based real world mapping and navigation. *IEEE Transactions on Robotics and Automation*, 3(3):249–265, 1987.

[Elfes, 1989] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, pages 46–57, June 1989.

[Ellman, 1989] T. Ellman. Explanation-based learning: A survey of programs and perspectives. *ACM Computing Surveys*, 21(2), 1989.

[Hwang & Ahuja, 1992] Y. K. Hwang and N. Ahuja. Gross motion planning – a survey. *ACM Computing Surveys*, 24(3):219 – 293, 1992.

[Kietz & Wrobel, 1992] J.-U. Kietz and S. Wrobel. Controlling the complexity of learning in logic through syntactic and task-oriented models. In Stephen Muggleton, editor, *Inductive Logic Programming*, chapter 16, pages 335 – 360. Academic Press, London, 1992. Also available as Arbeitspapiere der GMD No. 503, 1991.

[Leonard & Durrant-Whyte, 1992] J. J. Leonard and H. F. Durrant-Whyte. *Directed sonar sensing for mobile robot navigation.* Kluwer Academic Publishers, Boston, London, Dordrecht, 1992.

[Miller *et al.*, 1990] W. T. Miller, R. S. Sutton, and P. J. Werbos. *Neural networks for control.* The MIT Press, 1990.

[Ooi *et al.*, 1990] K. Ooi, K. Izumi, M. Nozaki, and I.Takeda. An advanced autofocus system for video camera using quasi conditioning reasoning. *IEEE Transactions on Consumer Electronics*, 36(3):526 – 529, 1990.

[Ringle, 1993] Martin D. Ringle, editor. *Special Issue: Situated Action*, volume 17 of *Cognitive Science.* 1993.

[Sutton, 1988] R. S. Sutton. Learning to predict by methods of temporal difference. *Machine Learning*, 3:9 – 44, 1988.

[Wang & Mendel, 1992] L.-X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(22):1414 – 1427, 1992.

[Williams, 1992] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, pages 229 –256, 1992.

[Wrobel, 1991] S. Wrobel. Towards a model of grounded concept formation. In *Proc. 12th International Joint Conference on Artificial Intelligence*, pages 712 – 719, Los Altos, CA, 1991. Morgan Kaufman.