

Grounding planning operators by affordances

Christopher Lörken and Joachim Hertzberg
University of Osnabrück, Institut für Informatik
D-49069 Osnabrück, Germany

{cloerken | hertzberg}@informatik.uni-osnabrueck.de
<http://www.inf.uos.de/kbs>

Abstract—This paper introduces a view on planning that is based on interpreting objects not by their identity but by their functionality. Ecological psychologist J.J. Gibson’s concept of *affordances* provides the underlying theory for developing a system that does not distinguish between objects of functional equivalence. It will be argued that the plans resulting from this approach are both less complex and still more flexible and robust in their application while proposing solutions to symbol grounding of objects in the environment and planning operators in execution behaviors. The system will be presented by example.

Keywords: planning, affordance, functional equivalence, symbol grounding, autonomous robot, real-world application

I. MOTIVATION

Controlling an autonomous robot has been among the chief target applications for the field of action planning in AI, starting with STRIPS/SHAKEY [19]. Planning has made considerable progress in the recent past [8], and so has robotics. Then why aren’t smart plan-guided robots all over the place?

A fundamental problem in plan-based robot control, if “plan” is understood to be a state-of-the-art AI plan, is that it would require to solve a symbol grounding [10] problem in two directions. First, the objects and relations that occur in the domain need to be identified from the robot’s sensor data; e.g., in order to move box_A from room X to Y , it needs to be identified and distinguished from, say, box_B and box_C . While this may work well for certain objects, such as for a neighboring room that gets identified by way of robot self-localization in a prior map, it may be very expensive to do for others. Second, the abstract operators contained in a plan need to be grounded in actual control rules or behaviors of the robot. While this has been tackled in several versions in hybrid robot control architectures [2], it is still far from being well-understood.

This paper explores an approach to conceptualizing differently the entities that a plan-based robot perceives in its environment and, hence, the way in which operators are grounded in physical robot behavior. It is based on the notion of *affordances*, as developed by J.J. Gibson. His idea was that

This research is funded by the European Commission’s 6th Framework Programme IST Project MACS [16] under contract/grant number FP6-004381. The Commission’s support is gratefully acknowledged. We thank Andreas Bartel, Frank Meyer, and Andreas Nüchter for their most helpful cooperation and assistance.

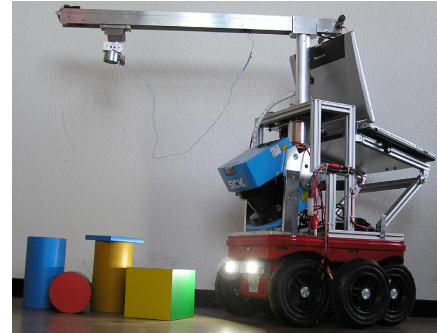


Fig. 1. The Kurt3D robot with a magnetic crane manipulator.

biological beings would perceive important action potentials of their direct environment on an elementary level: e.g., the affordance of “something” in the environment to be lifted by some agent can be perceived directly without prior object segmentation and classification; it may be triggered by a rock, a glass, and a cell phone all alike for, say, a human-sized agent. We will go into some detail in Sec. II. Note that we do not postulate that agent and robot perception has to be based on affordances alone: individual objects, e.g., may play a role in addition, and anchoring [4] may be a method of grounding them in sensor data. We focus on affordances in this paper to make clear the potential that they offer for plan-based robot control.

And here is this potential in a nutshell: Assume a robot is able to perceive some affordance “directly”, i.e., by very cheap algorithms on its sensor data, which do not require object segmentation, classification and reasoning. Assume further that the physical behavior, like lifting, linked to the perceivable affordance is modeled abstractly as an operator in the planning domain description. Then grounding this abstract operator in physical robot behavior would mean to guide the robot into a situation where the sensor data known to trigger the affordance is known or expected to be present, and then rely on the affordance-behavior coupling to guide the physical behavior. So if the robot perception process does include perception of affordances (their number may be large or small), then tailoring the plan operators “in the right way” according to these affordances will solve the grounding problem in both directions. This will be detailed and exemplified in the body of the paper.

Let us emphasize that the current paper’s contribution lies in the way in which domains with affordances present are modeled for planning and in grounding operators. The planning *process* and *algorithms* do not change: In fact, we will show in the paper body that a standard planning system (FF in this case) based on a standard planning domain description in terms of PDDL can still be used, making it possible to profit from latest planning technology. A blend of planning and situatedness has been an issue for long, Agre and Chapman’s work on PENGU [1] being a prominent exemplar. Yet our approach differs from that line of work in that we keep the classical planning process and algorithms, tailoring them such that, by aid of affordances, particular actions in the classical plan may physically run as situated behaviors; the earlier work, in contrast, has put forward that situated plans be entirely different in nature from classical ones.

The paper is structured as follows: Sec. II will describe and interpret the affordance term and start to address its meaning and usefulness for robotics. Sec. III will describe the rationale and the functionality of the planning system and of the closely related execution module and shed light on how operators can actually be grounded by affordances. Sec.s IV and V will describe the developed system including the necessary representations and will present results by example. Finally, Sec. VI summarizes.

II. AFFORDANCES

The term affordance has been coined by ecological psychologist J.J. Gibson almost 30 years ago. He states:

“The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill.” [9, p. 127]

To skip some decades of vivid discussion and different views of Gibson’s concept we will follow the interpretation proposed by A. Chemero [3].¹ He describes an affordance as a *perceivable relation between an environmental object and an agent* that describes some potential for action. In other words, an affordance describes what the agent can do with the object. A stone might, e.g., afford to be thrown or to be used as a paperweight. The affordance itself is hereby subjective to the agent, i.e. the same stone might afford to be liftable for a strong agent, but only afford sitting for a smaller, weaker agent.

We thus interpret affordances as *directly perceivable mappings of environmental features or characteristics to the abilities for interaction* of the particular agent, in our case, of a robot. We therefore see affordances like *liftable*, *pushable*, or *passable* that a robot can perceive in its environment. A robot as the one shown in Fig. 1 that has a magnetic crane as its primary manipulator will, e.g., perceive magnetic objects below a certain size and with a fairly flat top as affording the lift action.

¹For a wrap-up of the discussion and an elaboration on the use of affordance theory for robotics see, e.g., [14].

Gibson’s notion of affordances can thus be seen as a function-centered view on perception. When speaking of affordances, one assumes a system that interprets its surroundings by what it can do with it, how it can interact with it, instead of what things are and if they can be recognized. In other words, we do not need to detect and categorize objects in order to know what to do with them; thus solving the bottom-up problem of symbol grounding. We can furthermore formulate affordance-related operators like, e.g., *lift(l)* for any *l* that is perceived as being liftable. This operator can, in a top-down manner, be grounded in a physical behavior that is being guided by the perceived affordance of liftability.

In the following section, we will describe how standard planning techniques can be used for incorporating the affordance concept and how such a planning system and a robot’s execution control complement each other and benefit from this representation.

III. PLANNING WITH AFFORDANCES

We aim at a system that embeds the expressive power of the affordance concept (as interpreted by Chemero) in standard planning techniques to show that we provide an additional way of representing planable knowledge that does not conflict with existing standards. The approach introduced next will thus, in the end, use the well-known and established planning tools of the Planning Domain Definition Language (PDDL, [17]) and the Fast-Forward (FF) planner by J. Hoffmann [13].

The overall system operates on a knowledge representation or world model based on a coarse symbolic map defining regions in space (see Fig. 2(b)). In contrast to other current systems, we do not insert objects with associated feature lists in that map, but *affordance labels* instead. That means, we assume our robot to explore the environment and perceive, e.g., the affordance of some liftable item.² Then, the map-region in which it has perceived that affordance will be tagged liftable.

The intrinsic meaning of the affordance concept now allows us to specify an operator *lift(l)*, where *l* represents any suitable object that the robot can actually lift. It is evident that the functional equivalence used here in representing the domain helps make the domain model compact. Take the simple example that all blue objects are liftable for the robot, and all red objects with a flat surface either. This leaves us with two actual instantiations of the affordance *liftable*. If the planner now does not care about these instantiations but instead sends the operator *lift(l)* to the robot’s execution control, the robot can look up in memory which particular instantiations of the affordance *liftable* it knows (namely, exactly those two) and configure its perception to look for the cues that are connected with those instantiations. In this case, the robot would look either for blue color blobs or for red and flat structures and can then ground the *lift* operator on whatever cues it perceives first that meet these requirements.

²The next section explains how affordances get perceived and represented.

Neither is it necessary to recognize objects nor to specify how the operator is being grounded, since it is only at execution time that the robot has to decide by which behavior and object or percept the operator will be implemented.

This has the following benefits:

- 1) The search space for the planner to explore is significantly reduced. Instead of dealing with a potentially large number of operator instances built from all liftable objects (the way in propositional planning), or of representing all these objects by a variable and employing a more complex least-commitment planner (the way in classical partial order planning), we are using just one affordance at planning time.
- 2) The system will be more flexible in execution, as the first appropriate object will be selected in the current environment – be it static or dynamic.
- 3) The actual plan operators need not be changed if the robot gets to know another class of objects that are liftable, e.g., cylinders, as far as the notion of categorized objects is avoided in the domain model. It will only be needed to add another affordance instantiation telling the system that cylindrical objects are liftable as well. Then they will be selected automatically together with the other instantiations for grounding the *lift* operator during execution.

It is to note here that we do not aim at a system that completely avoids the notion of objects. In fact, quite the opposite is the case: objects are very valuable and needed for high-level planning, and we think everything that is worth being anchored as an object should in fact be anchored as an object. But the concept of affordances allows us to neglect the actual object term and the connected workload in generating and maintaining their representation in the world model as well as their additional load on planning for exactly those items where we do not need the item’s identity, e.g., when we do not care if we lift box_A or box_B or even $cylinder_{42}$.

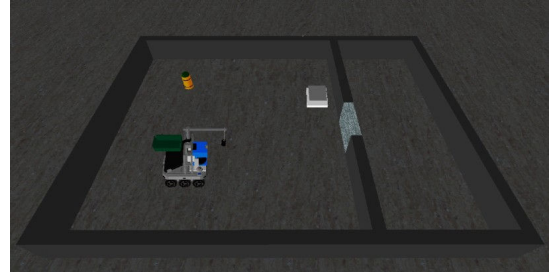
In the rest of this paper, we will, nevertheless, focus on a scenario where we do not need to represent objects at all in order to demonstrate the power and usefulness of the affordance concept. An overall system should of course aim at combining the following affordance extension with standard object-centered representation approaches. Sec. IV will now describe in more detail the underlying representation of both affordances and the general working domain before Sec. V will present the plans that result from this representation.

IV. MODELING AFFORDANCES FOR PLANNING

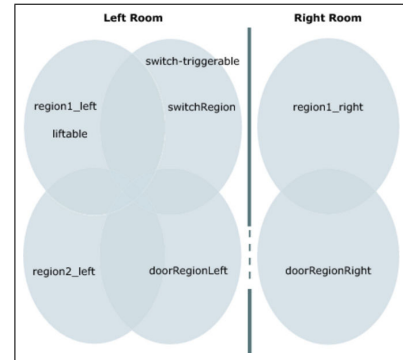
A. Affordance Representation

Being a relation between the environment and the robot, an affordance is represented here in triplet form. The *affordance representation triple* (ART) has the structure:

$$\left\langle \begin{array}{l} \text{cue descriptor,} \\ \text{behavior descriptor,} \\ \text{outcome descriptor} \end{array} \right\rangle \quad (1)$$



(a) Demonstrator scenario.



(b) Symbolic map.

Fig. 2. Scenario and map. Coarse localization in qualitative regions is sufficient for affordance-based planning. The scenario picture is taken from the MACSim simulator by METU-KOVAN [5]. Two rooms are separated by a wall with a sliding door, which can be operated by putting something on the switch by the door.

The *cue descriptor* part contains information like, e.g., color or shape; the *behavior descriptor* refers to the physical action that is afforded for the robot; and the *outcome descriptor* describes the perceivable effect expected of that action. Staying with the earlier example, a *liftable* ART may thus represent that all blue objects are liftable, e.g., with a crane or an arm. The actual ARTs may either be handcoded or acquired during an exploration phase of the system in which the robot tries out its different actions on various environmental objects (cf., e.g., [7], [21]).

Let us assume here that the robot is able to detect the cues described in the different cue descriptor parts of its ARTs within its environment; the details regarding these procedures are described elsewhere (cf. e.g., [20]).

Note, however, that we distinguish the *abstract affordance type liftable* from its different instantiations, the ARTs, that belong to that type. So the two different instantiations mentioned in the previous section of blue or red and flat liftable objects correspond to two distinct ARTs of the same abstract affordance type liftable.

B. Experiment Setup and World Model

We chose a simple demonstration scenario consisting of two rooms, a separating wall with a door, and a switch that

has to be weighted for the door to open (see Fig. 2(a)). To represent this setup and the current state of the world, we introduce the map depicted in Figure 2(b). It divides the room structure into 6 labelled regions, where these are not meant to be separated sharply, but may overlap.

The entries in this map that have to be maintained by the system are whether an instance of a known affordance type has been perceived while being located in one of the regions. Most importantly, upon perceiving some affordance, i.e., upon detecting some ART's cue descriptor, the robot tags the map region with a label stating that it has perceived that affordance type, e.g., *liftable* in Fig. 2(b), in that region. Note that the map does neither contain the ART whose cues have been detected, nor the presence of particular cues in that region. The map is concurrently being updated while the robot is moving about its environment.

C. Domain Description

Having introduced the representation of affordances and the underlying symbolic world model, we now move on to the actual planning side of the system. As mentioned earlier, the planning domain itself is completely modeled in PDDL, using the standard planner FF [13] for planning.

The domain description, which we provide only partially here for reasons of space, does not differ much from a common PDDL description. However, some predicates are affordance predicates reflecting the fact that an abstract affordance type has been perceived within a particular region.

Listing 1 Predicates

```
(:types region room switchRegion doorRegion)
(:predicates
  (robotAt ?region - region)
  (inRoom ?region - region ?room - room))
  (hasLiftedSomething)
  (liftable ?region - region )
  (switch-triggerable ?region - switchRegion)
  (passable ?startRegion ?targetRegion - doorRegion))
```

Listing 1 defines, next to some simple world fact predicates, the atomic formulas *liftable*, *switch-triggerable*, and *passable*. Each of these encodes the robot's percept of an according affordance type within a particular map-region. So, e.g., *liftable(region2_left)* would be contained in the situation description (see listing 4), if the robot has previously perceived the cues of one or more ARTs belonging to the affordance type *liftable*. *switch-triggerable* relates to the robot's impression that it can put something on the switch, while *passable* describes the affordance of something like a gap in a wall that can be passed through. These predicates thus code the general impression of the robot to have perceived some affordances within the regions.

Due to the nature of the affordance concept, one can use this symbolically encoded knowledge about the possibility for action that a certain area provides, for modeling operators that exploit this knowledge by using it as a precondition.

Listing 2 Lift - Action

```
(:action lift
  :parameters (?region - region)
  :precondition
    (and
      (robotAt ?region)
      (liftable ?region)
      (not (hasLiftedSomething)))
  :effect
    (and
      (hasLiftedSomething)
      (not (liftable ?region))))
```

Listing 2 shows a lift operator whose precondition requires being in a region where the liftability affordance has been perceived.

When an instance of this operator is being reached during execution, the robot will retrieve all liftability ARTs from its memory, configure its perception to look out for any of the cues represented in the cue-descriptors of the ARTs, and eventually select a suitable liftable object within that region to lift. It will ground the *lift*-operator elegantly in the next object that actually affords lifting for the robot, and on the appropriate physical behavior to lift it.

Listing 3 presents another example for an affordance-based operator, namely, *trigger-switch*, that is insofar different from the last example, as it contains the assumption of a perceivable affordance on the operator's effect.

Listing 3 Trigger-Switch - Action

```
(:action trigger-switch
  :parameters (?doorRegion - doorRegion
    ?otherDoorRegion - doorRegion
    ?switchRegion - switchRegion)
  :precondition
    (and
      (robotAt ?switchRegion)
      (hasLiftedSomething)
      (switch-triggerable ?switchRegion)
      (not (= ?doorRegion ?otherDoorRegion)))
  :effect
    (and
      (passable ?doorRegion ?otherDoorRegion)
      (passable ?otherDoorRegion ?doorRegion)
      (not (switch-triggerable ?switchRegion))
      (not (hasLiftedSomething))))
```

This is of course needed for proper planning precisely because a closed door does *not* afford passing. But as that means that *passable* would not be contained in the map, as it would be false at planning time, the system needs the knowledge of how to generate that fact or, generally speaking, of what new affordance is expected to result from an action.

In some sense, this contradicts the original affordance idea of Gibson, who claimed that affordances are immediately there to be perceived and that one does not reason about them. While this view, due to the computational complexity of perception, has already been questioned in the robotic literature (cf. e.g. [15], [14]), it does not even require more expert knowledge than needed in standard object-feature based planning systems. Such a system would simply model that the door object has, as a result of the trigger-switch operator, the feature of being open, and that feature would in turn be a precondition for the drive-through operator.

D. Different Types of Operators

As stated above, the affordance concept offers a valuable way to extend existing planning representations with operators that make use of the perceivable functional equivalence of objects. The introduced system is, however, really an extension, not a substitution. Even this simple example scenario includes operators that should not be formulated as depending on affordances. In this case, these operators are not connected to objects, but to the basic navigation skills of the robot. Above, it has always been assumed that the robot already was in a particular region in order to demonstrate the affordance-based operator grounding. Of course, one could argue that flat floor affords navigability for the robot; in fact, there are some approaches focussing on affordance-use for navigation (e.g. [6], [18], [21]). In our case, where each region in the same room is approachable from each other region, an *approach* operator does, however, neither have to be affordance-based nor would it be beneficial, since adding affordance percepts for navigability to each region in the map does not contain any valuable information whatsoever (cf. the more general discussion of this matter in [12]).

V. RESULTS

A. Planning

The previous section has provided excerpts of the developed affordance-based domain description, which can be used as the input to an off-the-shelf planner.

Listing 4 shows an exemplary problem definition for this domain that specifies the goal for the robot to get into the right room. In order to make things interesting, the door between the two rooms is closed, but the robot has already perceived liftability in region2_left and the affordance of a triggerable switch in the switchRegion.

Listing 5 shows the resulting plan as it has been generated by FF, omitting some implementation specific parameters. As one can see, the plan tells the robot to move to the region where it knows that something liftable is present. Then, in that region, the plan tells to just lift up *something liftable*, to carry it to the switch, open the door, and eventually act upon the passable affordance to drive to the other room. The time for computing this plan (a fraction of a second) is negligible compared to the time for physically executing the plan on the real robot.

Listing 4 Problem Definition

```
(define (problem macs-prob)
  (:domain macs-example)
  (:objects
    region1_left region2_left region1_right - region
    switchRegion - switchRegion
    doorRegionLeft doorRegionRight - doorRegion
    rightRoom leftRoom room )
  (:init
    (robotAt region1_left)
    (inRoom region1_left leftRoom)
    (inRoom region2_left leftRoom)
    (inRoom switchRegion leftRoom)
    (inRoom doorRegionLeft leftRoom)
    (inRoom region1_right rightRoom)
    (inRoom doorRegionRight rightRoom)
    (liftable region2_left)
    (switch-triggerable switchRegion))
  (:goal
    (robotAt region1_right)))
```

Listing 5 FF Generated Plan

```
0: APPROACH-REGION region2_left
1: LIFT region2_left
2: CARRY switchregion
3: TRIGGER-SWITCH switchregion
4: APPROACH-REGION doorregionleft
5: CHANGE-ROOM doorregionleft doorregionright
6: APPROACH-REGION region1_right
```

B. Execution

Given a plan as in Listing 5, the execution module only has to select the operators sequentially and execute them step by step. In case of the type of non affordance-related operators as, e.g., *approach*, the system's execution module can simply go on by triggering whatever control routines are implemented to carry out the desired action. If the operator is affordance-related, i.e., if it has the percept of an affordance as precondition, the execution module has to retrieve all ARTs that belong to the affordance type specified in the operator's precondition, and to configure the robot's perception to look out for their cues. It so selects the next object that provides the required cues and grounds that operator on that object by using it for interaction as specified in the ART's behavior descriptor. Hereby, it does neither matter which particular object it is, be it box_A or box_B nor which particular ART has been detected, be it the one for liftable boxes or the one for liftable cylinders.

The resulting plan execution is thus more flexible than the one based on object-centered domain representations. The reason is that the same operator may get grounded in multiple physical actions in the environment, depending on which object gets chosen at execution time. The choice emerges from the actual interplay of perception, behavior,

and environment dynamics (if any) during execution; given that there is no need to distinguish between different objects offering the affordance in question, this non-commitment is perfectly acceptable. The execution has a strong flavor of opportunism [11] in the sense that it will jump to opportunities that may not even have been present at planning time. The region model proposed for the knowledge representation does furthermore add robustness to the system since the robot just moves to a coarse region where it had perceived the affordance in question earlier. If the objects that resulted in the percept of the affordance have moved within that region because of some environment dynamics, the execution will not even take notice as the robot will simply interact with the next available and suitable object.

www.inf.uos.de/~cloerken/videos.html contains annotated videos from robot plan execution runs in a simulator.

VI. SUMMARY AND NEXT STEPS

The presented system proposes a new view on operator grounding as an extension to robot planning approaches. Objects are represented not by their identity but rather by the abstract kind of interaction possibility that they afford to the robot, i.e., by their affordance relation. A symbolic formalization of this concept has been introduced using only the common tools of PDDL, and results have been presented with an off-the-shelf planner, namely, FF.

Combined with the introduced way of grounding operators during execution, the resulting plans yield a simplification and narrowed search space compared to propositional planning or classical partial order planning approaches.

We have demonstrated that the resulting system is flexible and robust in terms of plan execution, as the interaction objects are not predefined but instead selected and grounded in a behavior during the execution phase.

The system scales easily with newly introduced objects, since they only necessitate new ARTs to be learned, but no change of the planner. Facing a physically different robot, the planning domain description has only to be adopted if that robot can achieve qualitatively different actions. That means, there is no need for alterations if the robot, e.g., simply uses an arm to lift things instead of a crane. What changes is only the interpretation of the affordance of liftability, i.e., the actual ART that would contain different cues for liftable objects for a robot with a crane or an arm. New operators only have to be designed for outcomes that the old robot was unable to achieve, e.g., throwing with an arm manipulator.

One of the most interesting next steps is to reintroduce the notion of objects to the affordance-based planning approach. As stated above, the presented system is meant to augment the existing approaches, as it tries to limit some of the complexity today's planning systems have to cope with. It is, nevertheless, crucial for a planner in general to be able to tell objects apart: For example, it may be of concern not to use the expensive vase to trigger the switch. The interesting question when combining the object-centered and affordance-based representations is thus where and how to draw the line between those objects that need to be anchored

and identified, and those that can be interpreted via their functional equivalence.

A second interesting question is how to distinguish operators formally that are meaningfully affordance-related from those that are not. We believe that this operator-affordance relation is meaningful exactly in those cases when an operator can be grounded by exploiting the affordance. And we believe that these are always those operators that directly describe an action; like e.g., lifting a box but not attending a meeting or getting coffee. While a cup of coffee affords drinking, going to the coffee machine in order to get it is just a necessary planning step before the affordance can be exploited. The means for formalizing this interpretation still have to be evaluated.

REFERENCES

- [1] P. Agre and D. Chapman. Pengi: An implementation of a theory of action. In *Proc. AAAI-87*, pages 268–272. Morgan Kaufmann, 1987.
- [2] R. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.
- [3] A. Chemero. An outline of a theory of affordances. *Ecological Psychology*, 15(2):181–195, 2003.
- [4] S. Coradeschi and A. Saffiotti. An introduction to the anchoring problem. *Robotics and Auton. Syst.*, 43(2-3):85–96, 2003. Special issue on perceptual anchoring. Online at <http://www.aass.oru.se/Agora/RAS02/>.
- [5] E. Şahin. Middle East Technical University, Department of Computer Engineering, 2007. <http://www.kovan.ceng.metu.edu.tr>.
- [6] P. Doherty. Linköpings universitet, Department of Computer and Information Science, 2007. <http://www.ida.liu.se>.
- [7] G. Fritz, L. Paletta, R. Breithaupt, E. Rome, and G. Dorffner. Learning predictive features in affordance-based robotic systems. In *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (IROS 2006)*, pages 3642–3647, Beijing, China, October 2006.
- [8] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.
- [9] J. J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Hillsdale, 1979.
- [10] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
- [11] B. Hayes-Roth and F. Hayes-Roth. A cognitive model of planning. *J. Cogn. Sci.*, 3:275–310, 1979.
- [12] J. Hertzberg, K. Lingemann, C. Lörken, A. Nüchter, and S. Stiene. Does it help a robot navigate to call navigability an affordance? In *Towards Affordance-Based Robot Control*, volume 4760 of *LNAI*, pages 16–26. Springer, 2008.
- [13] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *J. Art. Intell. Research*, 14:253–302, 2001.
- [14] C. Lörken. Introducing affordances into robot task execution. In *Publications of the Institute of Cognitive Science (PICS)*, volume 2-2007. University of Osnabrück, Osnabrück, Germany, May 2007.
- [15] K. F. MacDorman. Grounding symbols through sensorimotor integration. *J. Robotics Soc. of Japan*, 17(1):20–24, 1999.
- [16] MACS. Webpage, 2004. <http://www.macs-eu.org>.
- [17] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. Pddl – the planning domain definition language. Technical Report CVC TR-98-003/DCS TR-1165, Yale University, October 1998.
- [18] R. R. Murphy. Case studies of applying Gibson's ecological approach to mobile robots. *IEEE Trans. Systems, Man, and Cybernetics*, 29(1):105–111, 1999.
- [19] N.J. Nilsson. Shakey the robot. Technical Report TN 323, SRI International, April 1984.
- [20] L. Paletta, G. Fritz, F. Kintzler, J. Irran, and G. Dorffner. Learning to perceive affordances in a framework of developmental embodied cognition. In *Proc. Intl. Conf. Development and Learning, ICDL 2007*, London, UK, July 2007.
- [21] E. Uğur, M. R. Doğar, M. Çakmak, and E. Şahin. The learning and use of traversability affordance using range images on a mobile robot. In *Proceedings of IEEE Intl. Conf. on Robotics and Automation (ICRA 07)*, pages 1721–1726, Rome, Italy, April 2007.