

Does It Help a Robot Navigate to Call Navigability an Affordance?*

Joachim Hertzberg, Kai Lingemann, Christopher Lörken, Andreas Nüchter,
and Stefan Stiene

Universität Osnabrück, Institut für Informatik
D-49069 Osnabrück, Germany
{hertzberg,lingemann,cloerken,nuechter,
stiene}@informatik.uni-osnabrueck.de
<http://www.inf.uos.de/kbs/>

Abstract. Gibson’s notion of affordance seems to attract roboticists’ attention. On a phenomenological level, it allows functions, which have “somehow” been implemented, to be described using a new terminology. However, that does not mean that the affordance notion is of help for building robots and their controllers. This paper explores viewing an affordance as an abstraction from a robot-environment relation that is of inter-individual use, but requires an individual implementation. Therefore, the notion of affordance helps share environment representations and theories among robots. Examples are given for navigability, as afforded by environments of different types to robots of different undercarriages and sensor configurations.

1 Background

Among the most basic properties that a mobile robot needs to perceive about its environment is whether it can go someplace, i.e., drive, walk, crawl, climb—whatever its kinematics. State-of-the-art work in indoor robot navigation typically abstracts away from many details of the problem by assuming implicitly that any area that is not perceived as being blocked is navigable. That works as long as you make sure your robot stays clear of staircases, glass doors and mirrors, no drawers out of sensor height are left open, the robot fits under all table-tops, and, of course, the floor is sufficiently flat for its undercarriage. In a nutshell: Out of relatively protected lab environments, it is to-day non-trivial for a mobile robot to determine whether it can physically move to some location in its vicinity.

* Work in parts done in the projects (1) LISA, which is funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept “Research for Tomorrow’s Production” (fund number 02PB2170-02PB2177) and managed by the Project Management Agency Forschungszentrum Karlsruhe, Production and Manufacturing Technologies Division (PTKA-PFT); and (2) MACS, which is funded by the European Commission’s 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission’s support is gratefully acknowledged.

Would it then make sense to work on a Grand Theory of Navigability for mobile robots to solve the problem once and for all? Probably not, as there is no such thing as “objective navigability” in the world—whether or not a particular robot can boldly go where it has never gone before depends on its undercarriage, kinematics, geometry, control, power, and many other parameters. Considering as a zoo of robots a B21 with its small hard wheels, an outdoor Kurt3D with 20 cm diameter rubber wheels, a regular street car, and a walking machine, it is obvious that there are tremendous differences. Navigability is a *relation between a particular individual or class of robots and its environment*, much like Chemero [4] has stressed in interpreting Gibson’s [9] affordances.

Does that mean that navigability, being something utterly subjective for a robot, should be deleted from the robot programmer’s vocabulary? Not either! The concept is generally useful for a mobile robot, just the attribution to some area in space (the concept’s implementation, to use informaticists’ speak) differs among different robots. As an abstraction, it is of general use. For example, it can be communicated among fellow robots, using a uniform meaning, but relying on different implementations (“Is that area navigable for you?”). If some piece of high-level robot control software is to be exchanged between different individuals of robots, then navigability is a good candidate for an abstract concept that can be used uniformly on a high level of programming or modeling, and that may require an individual implementation or “grounding” on every individual robot.

So this informal essay interprets, from a robot designer’s point of view, an affordance as an *abstraction from a robot-environment relation that is of inter-individual use, but requires an individual implementation or grounding*. Whether an affordance is currently present in some environment must be effectively determinable; perceiving it should typically require very little computation, based on available sensor data. The inter-individual use may in particular lie in sharing some abstract, high-level domain model that deals with the abstracted relation. So for example, if navigability of some connected area implies reachability between any two of its positions, then this implication remains true no matter what is navigable for a particular robot.

In the following, we will give an example, using the concept of navigability, of the use and usefulness of this view of affordances. First we give three examples from own previous work for different groundings of navigability; we will point to some other approaches from the literature. Path planning is an example for a general functionality using navigability as a basic concept and delivering different results for different navigability groundings. Sec. 3 discusses our view on the potential view of affordances in robot control. In the end, we summarize our argument in favor of using affordances, in the interpretation just sketched, in robot design.

2 Groundings of Navigability

Every computer program would yield correct results only if applied under the conditions stated in its specification. This holds for robot control systems, too.

For example, the navigation control of a mobile robot must of course be in harmony with its physical navigation capabilities—control should steer it only into areas that it can cope with. Often, the specification what a particular robot can and cannot handle physically and in terms of control is left at least partially implicit.

In this section, we will make more explicit the specifications of the autonomous driving capabilities of three different robot-plus-control systems. We will see that this leads to characterizing three different implementations or groundings of what may be labeled the affordance of navigability.

2.1 Navigability as Free Space in 2D

Our first example is a fast (up to 3.5 m/s) indoor robot, as described in detail in [12]. The robot is shown in Fig. 1, left.

As described in [12], its cruise control consists of a simple free-space seeking mechanism: Using very simple fuzzy rules operating on every single beam of a horizontal laser scan, it is determined in which direction is a “virtual roadway” with sufficient clearance (to set the heading), and how far away the nearest obstacle, if any, is in this virtual pathway (to set the velocity). As the involved calculations are utterly simple and the scanner delivers 180° scans with 1° resolution at 77 Hz , fast speeds as 3.5 m/s can still be safely controlled in areas where, at least from time to time, space may be wide open.

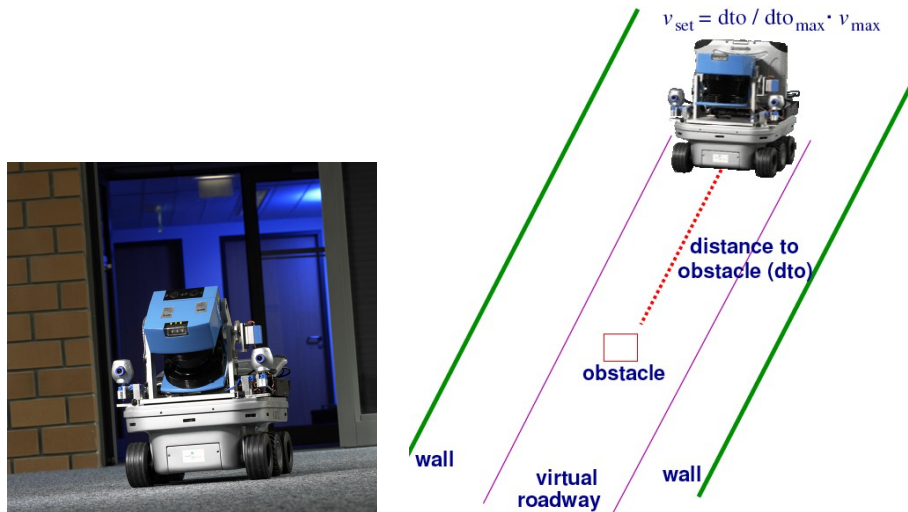


Fig. 1. The Kurt3D robot, images from [12]. Left: The robot platform. During driving, the 3D laser scanner, which is tilted downward in the image, is kept in horizontal position. Right: The “virtual roadway” used for setting velocity and heading. See text for explanations.

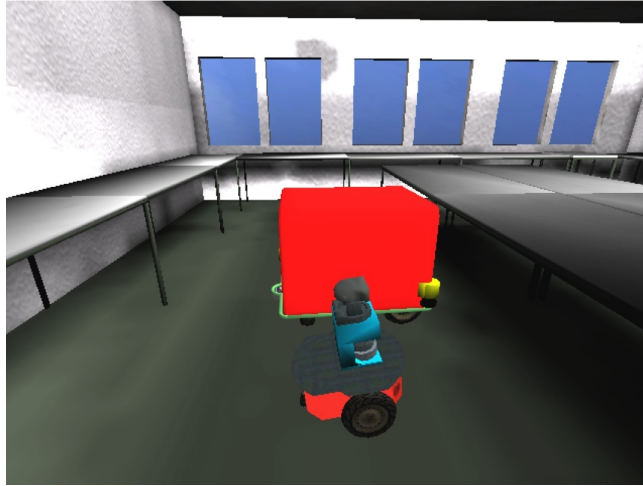


Fig. 2. Navigability in 3D for two different robots, applying the same criterion for their respective sizes, rendering from simulation. The simulated Pioneer in front and its bounding box fits under the table-tops; except for the table-legs, the area is completely navigable. The LISA robot (here the base platform with its correct sizes but without additional rigging, except for the navigation sensors) has table-top height, so it can navigate only in the relatively narrow aisles.

In the context of affordances viewed as relations between an individual and its environment, note that the definition and calculation of the virtual pathway does of course respect the width of the robot. Then what does this robot consider navigable, and what are the assumptions behind it? Like in many indoor lab robots, it is assumed

Space is navigable iff it is unoccupied in the virtual roadway.

This implies, in particular, that no overhanging objects exist for collision out of scan plane height, and that a sufficiently flat floor continues ad infinitum, unless framed by an obstacle perceivable in scanner height.

2.2 Navigability as Free Space in 3D

In the ongoing research project LISA [2], a robot platform of about desktop height with an additional manipulator on top is required to navigate safely in a populated lab environment. This includes tables, desks with drawers, wheeled office chairs, and other everyday objects. Part of the robot's task is to transport liquid samples. Therefore, collision with any object must be avoided under all circumstances. On the other hand, the floor is plane and flat with no steps or dents, just like in the 2D case of the Kurt3D robot before.

Under these requirements, free space in some scan plane is, of course, insufficient for navigability: We need to make sure that collision is avoided over all

points of the robot surface. Given that we don't have to care about the floor, we assume

Space is navigable iff it does not intersect the robot's bounding box in 3D. Additionally, accessibility may be required: The part of space in question is connected to the current robot pose by a path through navigable space.

Again, like in the 2D case, navigability of some area of space may differ with respect to different robot individuals, as their bounding boxes may differ. In this interpretation, it is similar to the configuration space of a navigating robot; this similarity is owed to the example of navigability, not to the concept of affordances. Note that an affordance is something in the robot's perception, whereas the configuration space is objectively given. In particular, it is possible to change the interpretation of navigability in a given robot (if the designer knows what he or she is doing), but it is not possible to change the configuration space just by an act of decision.

Technically, the required 3D geometry information about the environment is calculated on the LISA robot from small laser scanners mounted right above base plate height and pointing upward at an angle. The data fusion takes some time, but given that the velocity has to be small, in particular in the vicinity of objects, the required calculations can all be done online in real time. Figure 2 visualizes the practical difference in navigability of space for the LISA robot (or rather, a planning version of the bare navigation platform) and a Pioneer-type robot in simulation; the physical LISA robot is currently being built.

2.3 Navigability as Afforded by a 3D Surface

Out of flat lab floors, navigability depends, in addition to the absence of obstacles, on the geometry and physical structure of the ground in relation to the robot's undercarriage (wheel diameter, ground clearance, leg size etc.). This becomes an issue, e.g., in a growing number of outdoor robotics projects.

A relatively mild example is an approach to determine drivability of surface for an outdoor version of the Kurt3D robot; for details, please see [15]. The basic approach, adapted from [18], is this: Owing to its 3D laser scanner, Kurt3D builds incrementally a 3D geometry representation of its environment in the form of a scan point cloud; an example of a single scan is given in Fig. 3, left.

In a 3D scan, you can determine a sequence of scan points in the same azimuthal direction in rising vertical angles. If the angle between the cartesian projections of two subsequent measurement points is sufficiently low, then the second point is classified as a ground point. By connecting triplets of sufficiently close ground points (possibly in different azimuthal directions) to triangles, single ground points are enlarged to a ground surface; Fig. 3, middle, gives an example. Note that the ground surface need not be plane, but just "sufficiently plane", as determined by a threshold value of the allowed angle between subsequent points in ground determination.

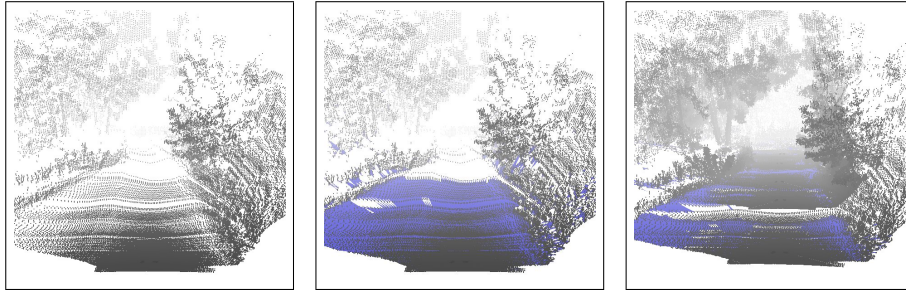


Fig. 3. Finding navigable surface on an outdoor gravel path scene; figure imported from [15]. *Left:* A single outdoor 3D scan. Note that the path is uneven. *Middle:* Areas (triangles) between neighboring surface points all labeled drivable are shaded in blue. Note that the area in front is very dense with surface points, which are all labeled drivable. Note second that there are some disconnected patches of surface points in and behind the path shoulder. *Right:* View into the model from the same virtual view point as before, but with the next scans along the path registered. Sufficiently large areas sufficiently dense with drivable surface points are filled with blue. (Again, the area in front is completely drivable.) Note that the next scan has been taken too far away from the first to connect the drivable surface areas, so some of the objectively drivable path remains unlabeled here for lack of point density.

The time required to compute this is marginal, consisting of elementary calculations on local data. In particular, it is negligible compared to the time required to register neighboring 3D scans, which, in turn, is small compared to the ~ 5 s required to record a 3D scan in the current version of Kurt3D. Navigability is handled quite explicitly here. In full detail, we have the following assumption:

Space is navigable iff it corresponds to a part of surface labeled drivable in the respective 3D scan, and the robots's bounding box positioned there does not intersect with points in the 3D scan. Again, accessibility may be required in addition.

2.4 Uses of Navigability

It is no artifact to be concerned about navigability, as can be seen in other work in the literature. As online processing of 3D geometry information is recently coming into play in robotics, it is natural to care about it. [18] is an example; we borrow the term *navigability map* from [8] (where it was called traversability map).

Reliable online determination of navigability under challenging conditions had to be pushed to some extreme in the DARPA Grand Challenge [7]. Consequently, a large part of the design effort of participating teams has gone into designing the sensor configuration and the respective algorithms for road detection (given that the Grand Challenge requires following some desert road rather than driving cross country). See [17] as an example.



Fig. 4. Tough instances of navigability decisions. Photos from the RoboCup Rescue competition, Osaka, Japan, 2005.

Navigability may even be more of a challenge in current RoboCup Rescue competitions [13], where part of the task is to navigate across extremely cluttered and ragged areas. Fig. 4 gives an impression. To our knowledge, there is no work explicitly in this context on determining navigability automatically. The reason is that current robots are mostly tele-operated in competitions and that the emphasis lies currently on physical maneuverability rather than autonomous control. So it is in fact up to the operator to decide about navigability, based on what he or she perceives on the remote user interface. Anyway, the setting presents a challenge for determining navigability, which will have to be done autonomously in the end.

Turning from these somewhat exotic examples back to mundane settings, note finally that differences in navigability in different robots have very practical consequences. Consider path planning. Standard methods work on 2D maps, see, e.g., [16, Sec. 6.2.1] and tools available in [1]. Planned paths are executable only if this map is in fact a 2D projection of the robot’s navigability map; if navigability in free space in 2D is all that is needed for a particular application, the two map types would coincide.

Fig. 2 has introduced an environment where considering 3D information does become relevant: Starting from identical positions and heading towards identical target positions, the two robots may have to plan and follow completely different paths, or there may even be no executable path for the LISA robot, whereas the Pioneer may safely execute one. Fig. 5 gives an example for this environment where different robots have to find different paths, based on their different navigability maps.

3 Discussion

Then what does all that tell us about using affordances in robot control? Obviously, all cited work, including our own, has been done without affordances in mind for the respective robots, and there is no point in re-labeling existing control code as providing this and that affordance for this and that robot. So the answer to the question in the title of this paper is of course: No! But the concept of affordances, as exemplified by navigability in the previous section, could help us in another way write better robot controllers.

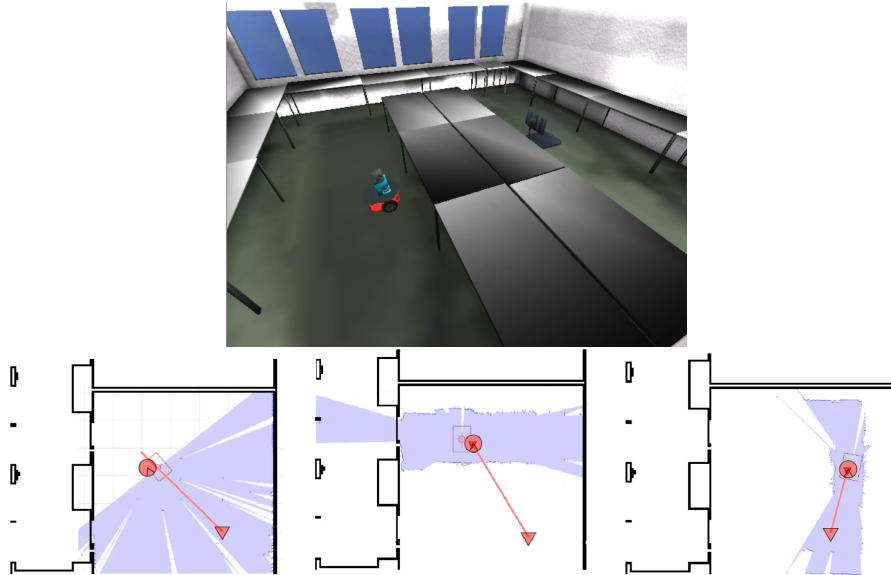


Fig. 5. An example for differences in navigability for different robots in the identical environment. **Top:** The problem is to drive from the shown pose to the target location right to the chair by the tables. **Bottom left:** The Pioneer robot shown in the top image may go straight below the tables. In light blue is the space momentarily perceived as navigable. **Bottom middle:** The taller LISA robot finds the direct path blocked (by the table-tops). **Bottom right:** After some exploration, the target position becomes part of perceived navigable space.

There is a recent trend in robotics to augment classical occupancy maps or geometry maps with certain semantic categories. This has been called *semantic mapping*, e.g., in [14], and it bears many resemblances to the classical AI problem of scene understanding or recognition [10, Sec. 16.6], to recent work on cognitive vision [5], and to symbol grounding [11] in full generality or in its more modest variant of perceptual anchoring [6].

Navigability maps, as presented in the previous section, are instances of semantic maps, if these are defined as *geometric maps that augment the geometric information by labels of data in the map* [14, p.2] – the labeling here being in terms of affordances like “navigable”. Interestingly, this label is attached to no segmented object, but to part of the environment, namely, to space nearby that appears to be navigable according to the (mostly implicit) navigability definition. For the historically minded, this seems to resemble Gibson’s statements that perceiving an affordance does require no object classification (and therefore, no prior segmentation).

Also somewhat in Gibson’s spirit, navigability is a relation between robot and environment that is defined in terms of action on the robot’s side. In happy contradiction to Gibson’s view, making a local navigability *map* out of recent perceptions does of course have a gist of classifying parts of the environment,

or of *reifying* a perceived relation into a classification, to use the term from knowledge representation.

What happens here is this: The robot designer has typically put much effort into making very fast the data processing that leads to the decision “navigable” or not for the particular robot – or graspable, pushable, kickable, whatever the robot’s purpose in life requires. By reifying the response of these carefully tuned perception processes, a small constituent of a symbolic scene description is generated. This generation runs on-line on-board the robot, and it comes practically for free, given it is done anyway in the robot control. In a nutshell, we are solving an instance of the symbol grounding problem here *en passant*, the symbol being the reified robot-environment relation “navigable”. Given that symbol grounding is known to be one of the deep, big, and hard AI problems: Why is it so easy here?

The answer is: We have turned the symbol grounding problem upside down (or upside up insofar as it was upside down before). Rather than attempting to recognize a given symbol in the sensor data, we have started from some sensor data processing that our robot can do efficiently and reliably, as its inner control cycle relies on it, and we have then labeled its result with a telling symbol, e.g., navigability.

Is this cheating? No, as we don’t claim that the robot has “invented” some new concept by itself – all creativity and insight remains on the robot designer’s side.

Is it good for something? Yes, if you accept (like all the researchers in semantic mapping appear to do) that symbolic environment models may help improve robot performance, communication, robustness and engineering, then a symbol like navigability appears to be as good as others – maybe even better, as it is closely connected with the robot performance. Our argument here is to build a robot environment ontology using reified affordances, like navigability, as part of the ground concepts. Other examples of environment categories for a mobile robot might be

Recognizability: The presence of some minimal amount of reliable features or a stable appearance for its particular sensor configuration and on-line processing capabilities;

Speedability: The property of some area of being traversed with high speed (high clearance and low curvature requirements permitting);

Odometrability: The property of some area (which has to be physically traversed to determine this) of producing a low error for the used forward kinematic model on which odometry is based (high grip, low curvature needs)

Many more of the same spirit are envisable. Clearly, an environment ontology using such concepts would be different from one based exclusively on human-centered concepts. But note that these concepts are well-defined, given that they rely on the algorithms and calculations that are part of the human-made robot design. They offer themselves as a basis for an inter-individual part of a domain ontology (which will also normally be human-made): Assuming every robot has its own implementation of these environment categories based on the

respective affordances, any higher-level symbolic theory in terms of them will then be grounded in the individual representations as the affordances induce. For example, based on a reification of the navigability affordance into a predicate *Navigable* with the respective *agent* and the perceived navigable *region* as arguments, we can use the following inter-individual axiom

$$\text{Navigable}(\textit{agent}, \textit{region}) \wedge \text{In}(\textit{loc}, \textit{region}) \rightarrow \text{Canmoveto}(\textit{agent}, \textit{loc})$$

for deducing that *agent* can move to location *loc* (assuming the intuitive meanings of all terms invented here). Note that the *Canmoveto* predicate naturally involves the agent. In harmony with the navigability affordance, it has to be agent-dependent, since it allows for different agent-specific implementations of the underlying action – may it be driving, walking, crawling, or whatever. The *In* predicate is agent-independent, of course, representing a general spatial relation. (It may be necessary or useful to give the *Navigable* and *Canmoveto* predicates another argument for time or situation, as they may be time-varying. This is out of the scope of the simple example here.)

There are singular examples in the literature where properties like the ones named have been used for tackling particular problems in robot control; for example, something similar to speedability has been used in [3] for improving the estimation of the time needed for completing a list of delivery tasks. Environment theories or ontologies using such reified robot-environment relations, whose individual grounding comes nearly for free in every individual robot controller, do not seem to exist yet.

4 Conclusion

So it does *not* help a robot navigate to call navigability an affordance – not too surprising. However, every efficient robot control system includes highly tuned sensor data processing modules whose purpose it is to perceive within short control cycle times particular robot-environment relations that are necessary for the robot’s intended function: navigability has been the running example for such a relation in this essay.

It may or may not be in Gibson’s spirit to call these relations affordances. Anyway, their reifications seem to offer some potential for a particular form of symbol grounding in robot control – in fact, the grounding is given, all that needs to be done is define fitting and useful robot domain theories or ontologies making use of the respective concepts. We have no such robot domain theory or ontology yet. We are just convinced they would make sense.

References

1. The player project (2006), <http://playerstage.sourceforge.net/>
2. Project LISA (2006), <http://www.inf.uos.de/kbs/LISA.html>
3. Belker, T., Beetz, M., Cremers, A.B.: Learning of plan execution policies for indoor navigation. *AI Communications* 15(1), 3–16 (2002)

4. Chemero, A.: An outline of a theory of affordances. *Ecological Psychology* 15(2), 181–195 (2003)
5. Christensen, H.I., Nagel, H.-H. (eds.): *Cognitive Vision Systems*. LNCS, vol. 3948. Springer, Heidelberg (2006)
6. Special issue: Perceptual anchoring. *J. Robotics and Autonomous Systems* 43(2-3) (2003)
7. DARPA. Grand challenge 2005 – harnessing american ingenuity (2005), <http://www.grandchallenge.org/>
8. Doherty, P.: Personal communication (2006)
9. Gibson, J.J.: *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA (1979)
10. Ginsberg, M.: *Essentials of Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA (1993)
11. Harnad, S.: The symbol grounding problem. *Physica D* 42, 335–346 (1990)
12. Lingemann, K., Nüchter, A., Hertzberg, J., Surmann, H.: About the control of high speed mobile indoor robots. In: *ECMR 2005. Proc. 2nd Europ. Conf. Mobile Robotics*, pp. 218–223 (2005)
13. NIST. Urban search and rescue robot competitions and arenas (2006), robotarenas.nist.gov/competitions.htm
14. Nüchter, A.: *Semantische dreidimensionale Karten für autonome mobile Roboter*. PhD thesis, Bonn University, Inst. f. Informatics (May 2006)
15. Nüchter, A., Lingemann, K., Hertzberg, J.: Extracting drivable surfaces in outdoor 6D SLAM. In: *ISR 2006. 7nd Intl. Symposium on Robotics and 4th German Conf. Robotik 2006* (2006)
16. Siegwart, R., Nourbakhsh, I.R.: *Introduction to Autonomous Mobile Robots*. MIT Press, Cambridge, MA (2004)
17. Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., Mahoney, P.: Stanley: The robot that won the DARPA Grand Challenge. *J. Field Robotics* 23(9), 661–692 (2006)
18. Wulf, O., Arras, K.O., Christensen, H.I., Wagner, B.A.: 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In: *ICRA 2004. Proc. IEEE Int. Conf. Robotics and Automation*, pp. 4204–4209 (April 2004)