# Planning Perception and Action for Cognitive Mobile Manipulators

Andre Gaschler[a] and Svetlana Nogina[a] and Ronald P. A. Petrick[b] and Alois Knoll[a]

[a]fortiss An-Institut Technische Universität München, Munich, Germany;
[b]School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

## ABSTRACT

We present a general approach to perception and manipulation planning for cognitive mobile manipulators. Rather than hard-coding single purpose robot applications, a robot should be able to reason about its basic skills in order to solve complex problems autonomously. Humans intuitively solve tasks in real-world scenarios by breaking down abstract problems into smaller sub-tasks and use heuristics based on their previous experience. We apply a similar idea for planning perception and manipulation to cognitive mobile robots. Our approach is based on contingent planning and run-time sensing, integrated in our "knowledge of volumes" planning framework, called KVP. Using the general-purpose PKS planner, we model information-gathering actions at plan time that have multiple possible outcomes at run time. As a result, perception and sensing arise as necessary preconditions for manipulation, rather than being hard-coded as tasks themselves. We demonstrate the effectiveness of our approach on two scenarios covering visual and force sensing on a real mobile manipulator.

**Keywords:** robot task planning, mobile manipulation

## 1. INTRODUCTION

The task of controlling mobile manipulators in real-world environments is an inherently challenging one since it often requires reasoning about both perception and action under uncertain conditions. In order to address these difficulties, we present an application of a "knowledge of volumes" approach to robot task planning (KVP in short), which combines the power of symbolic artificial intelligence planning with the efficient computation of volumes that serve as an intermediate representation for both robot perception and action.

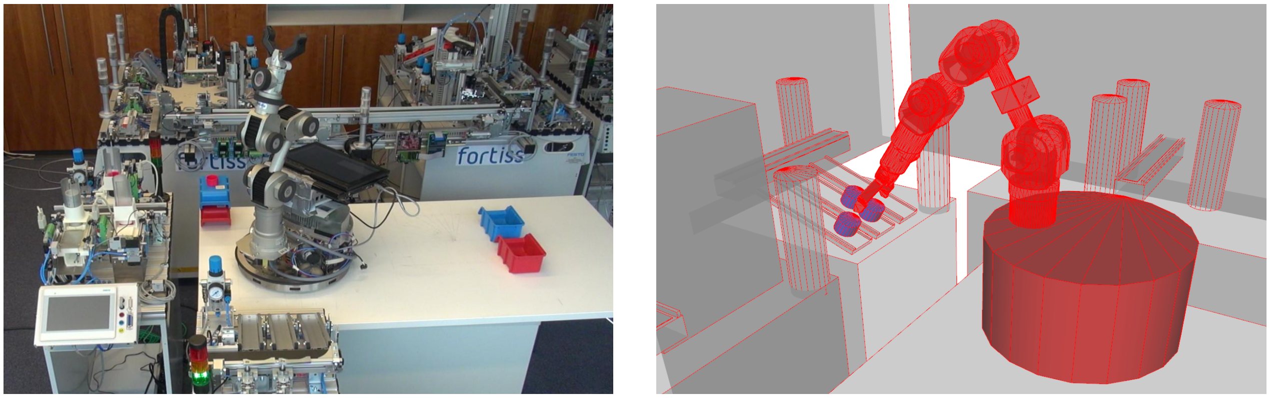
Our KVP approach is guided by two main principles which make it particularly useful for planning robot perception and manipulation with uncertain or incomplete knowledge, real-world geometry, and multiple robots and sensors. First, we use PKS (Planning with Knowledge and Sensing)[1,2] as the underlying symbolic planning system. In contrast to many other off-the-shelf planning engines, PKS operates at the knowledge level and can represent both known and unknown information, enabling it to model sensing actions with clear and concise domain descriptions, and reason in partially known environments that are typical for mobile manipulation. Second, rather than discretising the search space, we represent many geometric predicates—preconditions for perception and manipulation—by continuous volumes, specifically sets of convex polyhedra.[3,4] This notion of volumes serves as a powerful intermediate representation for modelling perception and action, both at the geometric level and at the symbolic level. Our approach is among the first to treat sets of convex polyhedra as such an intermediary representation between continuously-valued robot motions and viewing cones, and discrete symbolic actions, building a combined geometric and symbolic cognitive architecture.

In previous work, we introduced the general KVP framework,[4,5] and gave details describing the swept volume computation of sets convex sets of polyhedra[3] and planning with contingencies.[6] This work instead focuses on a Mobile Manipulation scenario (see Figure 1) which was chosen to demonstrate several key features of the KVP approach, namely planning with sensor actions (perception), discrete uncertainty (incomplete knowledge), and manipulation.

Further author information: (Send correspondence to A.G.)
A.G.: E-mail: gaschler@cs.tum.edu



A. Gaschler, S. Nogina, R. Petrick, and A. Knoll, "Planning Perception and Action for Cognitive Mobile Manipulators," Proceedings of SPIE Volume 9025 – Intelligent Robots and Computer Vision XXXI: Algorithms and Techniques, J. Röning and D. Casasent, eds., 2014.




(a) We evaluate our robot task planner on a Robotino mobile platform with a Katana 5-DoF manipulator in a factory setting.

(b) Actions have geometric preconditions and effects, with swept volumes of robot motions (red) and object boundaries modelled as sets of convex polyhedra (blue for movable objects, grey for static obstacles).

Figure 1: Mobile Manipulation Scenario

### 1.1 Related Work

Early work on cognitive mobile manipulators date back to systems like Shakey in 1984.[7] Since that time, the field has made significant developments, and cognitive planning architectures have been proposed from different research perspectives, including probabilistic techniques from artificial intelligence,[8] closed-world symbolic planning,[9–11] formal synthesis,[12,13] and sampling-based manipulation planning.[14,15]

A relevant contribution most closely related to our approach is the belief space planner by Kaelbling and Lozano-Pérez,[8,16] which models a belief space of probability distributions over states, making it robust against uncertainty and change. In contrast to belief states, our work instead relies on discrete knowledge and is designed for structured environments with incomplete information and sensing. Furthermore, while Kaelbling and Lozano-Pérez use octrees to represent swept volumes of robot motion, we use sets of convex polyhedra, allowing very efficient collision detection in the deterministic case.[3] In both cases, perception is formulated as a necessary precondition for manipulation, and is not to be hard-coded as a task itself.

Our work also differs from that of Kaelbling and Lozano-Pérez in other important ways. In particular, our approach is novel in using 3D geometric volumes as the underlying representation for symbolic planning and motion planning[5] when combined with an off-the-shelf, general-purpose AI planner supporting deterministic planning with incomplete information and sensing. This allows us to define geometric preconditions for sensing and manipulation actions, involving gain and loss of knowledge, and yields an automatised solution of interleaved sensing and manipulation actions,[6] making it a promising approach for addressing the problem of combined perception and action planning for mobile manipulation. Furthermore, we can easily integrate future improvements from new planning engines that become available in the artificial intelligence community. Finally, our work is demonstrated with a mobile manipulator, and both experimental scenarios were executed on a real Robotino mobile platform with a Katana manipulator, rather than solely in a simulated environment.

## 2. APPROACH

The KVP approach to robot task planning is characterised by two important principles which attempt to cope with the inherent difficulty of reasoning about both symbolic actions and geometric preconditions and effects: the representation of *knowledge* and the representation of *volumes*.[4,5] The notion of knowledge is used to model properties from the planner's belief state, rather than directly representing the world state. This approach is implemented by the knowledge-based symbolic planner PKS (Planning with Knowledge and Sensing),[1,2] which we describe in Section 2.1. Moving down from the symbolic layer, we consider the evaluation of geometric preconditions and effects using the representation of sets of convex polyhedra. Details of this process are given in Section 2.2. The connection between these two levels of representation is make explicit in Section 2.3, with a brief description of our cognitive architecture and software implementation. In the remainder of the paper we then discuss the application of the KVP framework in the evaluation of two task planning scenarios, the Mobile Manipulation and Force Sensing scenarios.

## 2.1 Planning with Knowledge and Sensing

Symbolic planning in KVP is provided by the general-purpose PKS planner,[1,2] which can construct plans in the presence of discrete uncertainty. In particular, PKS can model both knowledge gain and knowledge loss, making it well suited for representing sensing and perception actions. Unlike many planners, PKS operates at the "knowledge level" by reasoning about how the planner's knowledge state, rather than the world state, changes due to action. To do this, PKS uses an extended STRIPS-like[17] representation based on a set of five databases, each of which models a particular type of knowledge that can be interpreted formally in a modal logic of knowledge. Unlike planners that work with possible worlds models or representations based on belief states, PKS uses a restricted subset of a first-order language, supporting functions and run-time variables, to help improve the efficiency of its reasoning and plan generation processes.

In this work, we use three of the databases available in PKS: $K_f$, $K_w$, and $K_v$. The $K_f$ database contains facts about the world that are known to the planner, which are used to model the effects of physical actions that change the world. More formally, a formula $\phi \in K_f$ denotes the fact that "the planner knows $\phi$". A second database, $K_w$, models the plan-time effects of knowledge-gathering actions that can return one of two possible values. During planning, a formula $\phi \in K_w$ represents the idea that the planner knows whether $\phi$ or $\neg\phi$ holds, however, the actual binary value will only become known when the plan in executed, i.e., when a physical sensor is used. Finally, the $K_v$ database represents knowledge about function values that will become known at execution time. Therefore, $K_v$ can model the plan-time effects of sensing actions that return general constants (in contrast to $K_w$, which can only model sensors with binary outcomes). For more details on PKS, we refer the reader to the original PKS description[1,2] and its application to robotics.[5]

## 2.2 Geometric Queries with Sets of Convex Polyhedra

In addition to the notion of knowledge, the second principle underlying our KVP approach is the idea of representing all geometric objects and robot manipulators as sets of convex polyhedra.[6] Only with this data structure can we bridge the gap between discrete symbolic and continuous geometric planning, and efficiently evaluate a broad range of geometric preconditions and effects.

In the general case, however, decomposing an arbitrary non-convex mesh into a small (or even minimal) set of convex polyhedra is a challenging problem. Although the exact decomposition problem is known to be NP-hard, Mamou and Ghorbel[18] recently proposed an approximate algorithm that is efficient enough for our type of problem instances, i.e. to decompose CAD models of robots and objects in the scene.

As we previously described in prior work,[3,6] we first apply the mesh simplification algorithm by Garland and Heckbert[19] to reduce overly fine CAD models to $10^4$–$10^5$ triangles. Hierarchical approximate convex decomposition[18] can automatically decompose these models into $\approx 10$ convex models with 20 vertices each, at an accuracy of $\approx 30$ mm, and within a computation time of a few seconds. Decomposition is performed in an iterative manner, searching for "concave" edges on the dual graph of triangles. The search is guided by a mixed cost function, favouring higher local concavity and triangles with a higher aspect ratio. Then, vertices of the mesh are hierarchically segmented into a few convex bodies. After the actual decomposition, another slight mesh simplification may further reduce the number of vertices.

With this representation of robot and object models at hand, we can efficiently perform all geometric queries that are needed in a typical robot task planning scenario (see Figure 1). For collision and inclusion queries between objects or objects and static robots, we can directly use an implementation of the Gilbert-Johnson-Keerthi-algorithm (GJK).[20] For collision queries between objects and swept volumes of continuous robot motions, we first sample robot poses and compute convex decompositions of the involved swept volumes at a quadratic convergence rate;[3] once the swept volumes are available, the problem likewise reduces to pure convex-convex collision checking. As a result, our KVP framework features a very efficient implementation of all geometric queries that arise in our domain.

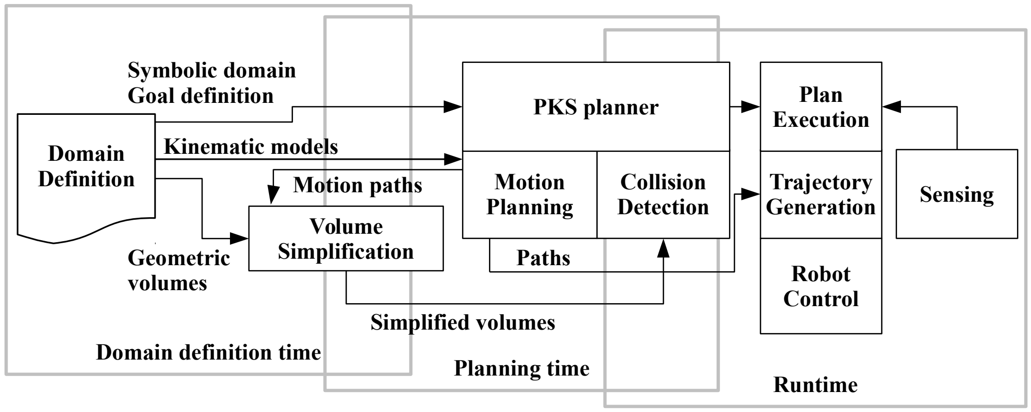



Figure 2: Overview of the implemented KVP software architecture.[5]

### 2.3 KVP Cognitive System Architecture

The software implementation of our KVP robot task planner is divided into several components which are executed or called as library functions, as shown in Figure 2. First, the geometric volumes of robots and objects are simplified and decomposed during the domain definition process. At this time, the concrete problem instance with robot and object positions is not yet known—only the CAD models of all involved volumes, static parameters (such as robot kinematics), and the symbolic domain definition of predicates, actions, and goals are used.

At plan generation time, the PKS planner searches for a sequence of actions whose symbolic and geometric preconditions are satisfied by the planner's knowledge state, and whose execution leads to a state that fulfils the goal conditions. To evaluate geometric preconditions and effects, PKS performs direct function calls to the domain-specific motion planning and collision detection functions. A valid sequence of symbolic actions and robot motion paths is produced as output. At run time, robot motion paths are interpolated by a simple trajectory generator and the physical robots are controlled in real time. KVP also allows run-time sensing and resolution of branched plans,[3] which are demonstrated in the Force Sensing scenario later in Section 3.2.

## 3. EVALUATION

In our evaluation, we implement and test our approach in a Mobile Manipulation scenario with a Robotino omni-directional mobile platform and a Katana 5-DoF manipulator, as depicted in Figure 1a. For comparison, we briefly summarise the results of the Force Sensing scenario from earlier work,[6] as this second scenario shows another interesting interaction between interdependent perception and manipulation actions.

### 3.1 Mobile Manipulation Scenario

In the Mobile Manipulation scenario, a single mobile manipulator must move a new block under a stack of $n$ blocks while keeping the original order (see Figure 4). This task is defined in KVP as a mobile manipulation problem comprising picking, placing, and moving actions. Symbolic action definitions for the PKS planner are given in Figure 3 (with a few helper functions omitted for brevity). In particular, the actions `pickUp` and `putDown` include preconditions that the robot be close to the directed location, which can be achieved by the `moveCloseTo` action. Also, the preconditions and effects of these actions ensure that only one object can be in the robot's gripper at any one time. The predicate `isReachableLocation` is an example of a domain-specific property which is evaluated by invoking the motion planning and collision detection library, providing a link between symbolic planning and motion control. Full details of the domain implementation are described in work by Nogina.[21]

The goal condition of this domain is that all objects are placed on a stack in a certain order at a location in the left of the scene (see Figure 4). For evaluation, all but the lowest object are already placed in a stack (Figure 4A), so any solution in this scenario involves moving all objects to a different location (Figure 4B), inserting the new object, and building up the stack in the defined order (Figure 4C). With this formulation,

```
action moveCloseTo(?l : location)
   preconds:
      !K(isCloseTo = ?l) &
      extern*(isReachableLocation(?l))
   effects:
      add(Kf, isCloseTo = ?l)

action pickUp(?o : object, ?l : location)
   preconds:
      K(handEmpty) &
      K(getObjectLocation(?o) = ?l) &
      K(isCloseTo = ?l) &
      K(knownPosition(?o))
   effects:
      del(Kf, handEmpty),
      add(Kf, inHand(?o)),
      del(Kf, knownPosition(?o))
```

```
action putDown(?o : object, ?l : location)
   preconds:
      K(inHand(?o)) &
      K(isCloseTo = ?l) &
      K(knownFreeSpace(?o, ?l)) &
      K(requestedSpaceFor(?l) = ?o)
   effects:
      del(Kf, inHand(?o)),
      add(Kf, handEmpty),
      add(Kf, getObjectLocation(?o) = ?l)
```

Figure 3: MOBILE MANIPULATION scenario: symbolic action definitions.

Table 1: Evaluation of the MOBILE MANIPULATION scenario, where $n$ objects are to be stacked in a given order. Even though this problem may seem simple to solve, finding a correct plan requires a global search, as any deviation from the correct sequence of actions will render the goal infeasible.

| Objects | Total Time | Symbolic Planning | Inverse Kinematics | | Path Planning | |
|---|---|---|---|---|---|---|
| $n$ | [s] | [s] | [s] | Calls | [s] | Calls |
| 1 | 2.2309 | 0.0020 | 0.00012 | 2 | 2.2205 | 6 |
| 2 | 4.5080 | 0.0137 | 0.14031 | 20 | 4.3315 | 18 |
| 3 | 6.4250 | 0.0365 | 0.14213 | 12 | 6.2112 | 30 |
| 4 | 8.4666 | 0.0730 | 0.14138 | 16 | 8.2029 | 42 |
| 5 | 11.0243 | 0.3074 | 0.14769 | 20 | 10.5091 | 54 |
| 6 | 15.0111 | 2.0692 | 0.14471 | 24 | 12.7248 | 66 |
| 7 | 17.2498 | 3.4601 | 0.30019 | 26 | 14.1849 | 77 |
| 8 | 80.7198 | 64.1298 | 0.33280 | 30 | 16.8952 | 91 |
| 9 | $>300$ (timeout) | | | | | |

the MOBILE MANIPULATION scenario is a typical example of Sussman's anomaly,[22] in that the correct solution cannot be found by a local search. As shown in the evaluation in Table 1, the domain becomes infeasible for $n > 8$ on a regular desktop computer due to limited memory. We purposely included this example to show the current limitations of KVP, and to motivate our future work aimed at improving our planning heuristics.

### 3.2 Force Sensing Scenario

In contrast to the MOBILE MANIPULATION scenario, we present a second domain from our previous work[6] in order to demonstrate the interaction between perception and manipulation. In the FORCE SENSING scenario (Figure 5a), a compliant robot manipulator has the ability to grasp, lift, and transfer beverage containers which are located on a table. When a container is lifted, the robot can sense its weight and, from this, reason whether the drink must be held upright in order to prevent spilling. The goal is to transfer all containers to a second table, and the robot may hold its gripper upright during these motions, if required.

Formally, the important actions of the domain are defined as shown in Figure 5b. Only when an object has been confirmed to not be spillable (i.e., $\neg$`isSpillable(?o)` is true) may the robot transfer it without holding it upright, using the action `transferFast`. However, since this knowledge is not available until run time, the planner must build a contingent (branching) plan using the action `senseWeight`, modelled with knowledge recorded in PKS's $K_w$ database, which allows it to reason about the case where an object can be spilled (i.e., `isSpillable(?o)` is true), as well as the case where an object is not spillable. Our software framework (Figure 2) executes the actions in this plan and chooses the appropriate branch to follow based on the run-time results of previous sensing actions. This scenario was implemented and tested on a joint-impedance controlled light-weight 7-DoF robot with a force-controlled parallel gripper, as shown in Figure 5a. The external force of weight of an object was measured by internal torque sensing.

## 4. CONCLUSION

In this paper, we present an application of our "knowledge of volumes" approach to robot task planning (KVP) in the field of mobile manipulation. In particular, we describe how a cognitive mobile robot can reason about knowledge gain and knowledge loss, allowing us to formulate perception and manipulation actions with both symbolic and geometric preconditions and effects. Our KVP framework solves this type of problem using a combination of knowledge-level planning and geometric predicate evaluation with sets of convex polyhedra.

## ACKNOWLEDGMENTS

The authors would like to thank Quirin Fischer and Sören Jentzsch for their help with the implementation and evaluation, and Markus Rickert for his comments on the manuscript. This research was supported in part by the European Union's 7th Framework Programme through the JAMES project (Joint Action for Multimodal Embodied Social Systems, grant no. 270435, `http://www.james-project.eu/`).

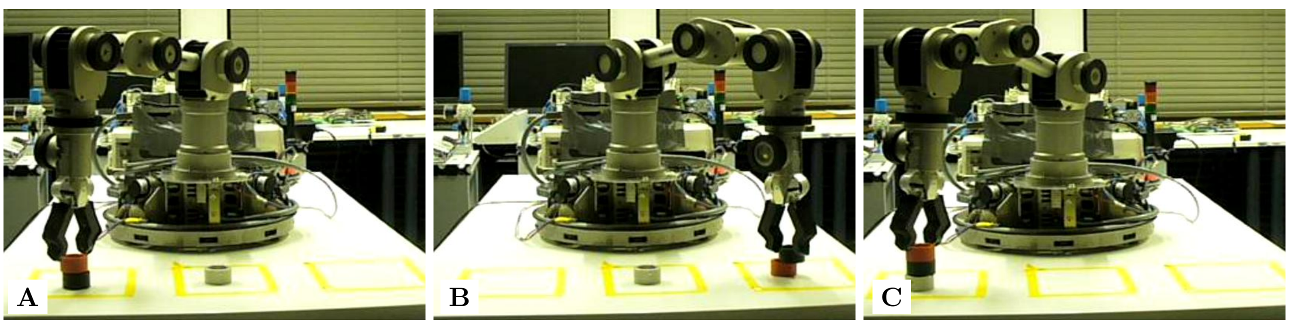
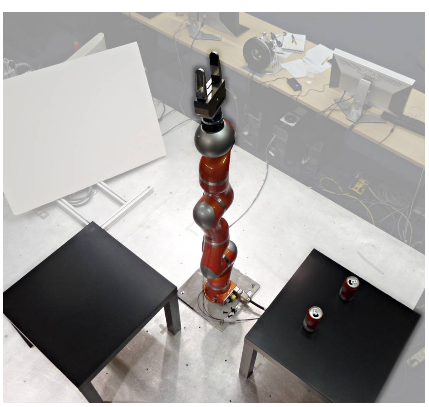



Figure 4: The MOBILE MANIPULATION scenario was implemented on a mobile manipulator with $n = 3$.



(a) Implementation

```
action transferFast(?o : object)
   preconds:
      K(¬isSpillable(?o))
      K(isGrasped(?o)) &
      K(¬isRemoved(?o)) &
   effects:
      add(K_f, isRemoved(?o))

action senseWeight(?o : object)
   preconds:
      K(isGrasped(?o))
   effects:
      add(K_w, isSpillable(?o))
```

(b) Example PKS actions

Figure 5: In the FORCE SENSING scenario, a compliant robot manipulator can sense if beverage containers are filled by weighing them, and can hold them upright while moving to prevent spilling, unless they are known to be completely empty or not opened.[6]

## REFERENCES


[1] Petrick, R. P. A. and Bacchus, F., "A Knowledge-Based Approach to Planning with Incomplete Information and Sensing," in [*Proceedings of the International Conference on Artificial Intelligence Planning and Scheduling (AIPS)*], 212–221 (2002).

[2] Petrick, R. P. A. and Bacchus, F., "Extending the knowledge-based approach to planning with incomplete information and sensing," in [*Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*], 2–11 (2004).

[3] Gaschler, A., Petrick, R. P. A., Kröger, T., Khatib, O., and Knoll, A., "Robot Task and Motion Planning with Sets of Convex Polyhedra," in [*Robotics: Science and Systems (RSS) Workshop on Combined Robot Motion Planning and AI Planning for Practical Applications*], (2013).

[4] Gaschler, A., Petrick, R. P. A., Khatib, O., and Knoll, A., "A knowledge of volumes approach to robot task planning," (2014). Under review.

[5] Gaschler, A., Petrick, R. P. A., Giuliani, M., Rickert, M., and Knoll, A., "KVP: A Knowledge of Volumes Approach to Robot Task Planning," in [*IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*], (2013).

[6] Gaschler, A., Petrick, R. P. A., Kröger, T., Knoll, A., and Khatib, O., "Robot Task Planning with Contingencies for Run-time Sensing," in [*IEEE Intl Conf on Robotics and Automation (ICRA) Workshop on Combining Task and Motion Planning*], (2013).

[7] Nilsson, N. J., "Shakey The Robot," Tech. Rep. 323, AI Center, SRI International (Apr. 1984).

[8] Kaelbling, L. P. and Lozano-Pérez, T., "Unifying Perception, Estimation and Action for Mobile Manipulation via Belief Space Planning," in [*IEEE International Conference on Robotics and Automation*], (2012).

[9] Cambon, S., Alami, R., and Gravot, F., "A hybrid approach to intricate motion, manipulation and task planning," *Intl. Journal of Robotics Research* **28**(1), 104–126 (2009).

[10] Plaku, E. and Hager, G. D., "Sampling-based motion planning with symbolic, geometric, and differential constraints," in [*IEEE Int. Conference on Robotics and Automation*], (2010).

[11] Dornhege, C., Gissler, M., Teschner, M., and Nebel, B., "Integrating symbolic and geometric planning for mobile manipulation," in [*IEEE Intl. Workshop on Safety, Security & Rescue Robotics*], 1–6 (2009).

[12] Kress-Gazit, H. and Pappas, G. J., "Automatically synthesizing a planning and control subsystem for the DARPA Urban Challenge," in [*Automation Science and Engineering (CASE). IEEE International Conference on*], 766–771 (2008).

[13] Cheng, C.-H., Geisinger, M., Ruess, H., Buckl, C., and Knoll, A., "Game Solving for Industrial Automation and Control," in [*IEEE International Conference on Robotics and Automation*], 4367–4372 (2012).

[14] Zacharias, F., Borst, C., and Hirzinger, G., "Bridging the gap between task planning and path planning," in [*IEEE/RSJ International Conference on Intelligent Robots and Systems*], 4490–4495 (2006).

[15] Barry, J. L., *Manipulation with Diverse Actions*, PhD thesis, Massachusetts Institute of Technology (2013).

[16] Kaelbling, L. P. and Lozano-Pérez, T., "Integrated Task and Motion Planning in Belief Space," *International Journal of Robotics Research* **32**(9–10), 1194–1227 (2013).

[17] Fikes, R. E. and Nilsson, N. J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence* **2**, 189–208 (1971).

[18] Mamou, K. and Ghorbel, F., "A simple and efficient approach for 3D mesh approximate convex decomposition," in [*IEEE International Conference on Image Processing (ICIP)*], 3501–3504 (2009).

[19] Garland, M. and Heckbert, P. S., "Surface simplification using quadric error metrics," in [*Proceedings Computer graphics and interactive techniques*], 209–216 (1997).

[20] Gilbert, E. G., Johnson, D. W., and Keerthi, S. S., "A fast procedure for computing the distance between complex objects in three-dimensional space," *Robotics and Automation, IEEE Journal of* **4**(2), 193–203 (1988).

[21] Nogina, S., *Task Planning for a Mobile Manipulation Scenario*, master's thesis, Technische Universität München (2013).

[22] Sussman, G. J., *A computational model of skill acquisition*, PhD thesis, MIT (1973).