The OpenGRASP Benchmarking Suite: An Environment for the Comparative Analysis of Grasping and Dexterous Manipulation.

S. Ulbrich, D. Kappler, T. Asfour, N. Vahrenkamp, A. Bierbaum, M. Przybylski, and R. Dillmann

Abstract— **In this work, we present a new software environment for the comparative evaluation of algorithms for grasping and dexterous manipulation. The key aspect in its development is to provide a tool that allows the reproduction of well-defined experiments in real-life scenarios in every laboratory and, hence, benchmarks that pave the way for objective comparison and competition in the field of grasping. In order to achieve this, experiments are performed on a sound open-source software platform with an extendable structure in order to be able to include a wider range of benchmarks defined by robotics researchers. The environment is integrated into the OpenGRASP toolkit that is built upon the OpenRAVE project and includes grasp-specific extensions and a tool for the creation/integration of new robot models. Currently, benchmarks for grasp and motion planningare included as case studies, as well as a library of domestic everyday objects models, and a real-life scenario that features a humanoid robot acting in a kitchen.**

I. INTRODUCTION

Benchmarking as a means of objective comparison and competition amongst researchers is and has always been of great interest in science—not only in robotics research—and it is common practice in the industry, to compare graphics accelerators for instance. It is crucial for research to design well-defined experiments that are reproducible by others research groups [1]. This necessity for similar mechanisms in robotics manifests in projects such as the European research project *BRICS (Best Practice in Robotics)* [2]. Together with the vast number of different techniques to tackle prominent problems in robotics arose the demand for efficient measurement and comparative analysis. A small number of famous competitions is known for complete integrated robotic systems such as the *DARPA Grand Challenges* [3], the *RoboCup* soccer competition [4], or *RoboCup@Home* [5], a competitive scenario for service robots. Unfortunately, participation in such big events is usually limited to a few selected groups, often simply caused by limited resources and the lack of necessary hardware.

Benchmarks that focus purely on the algorithmic parts of a system can be found for various sub-fields of robotic research. In mobile robots, several benchmark suites have been presented that address various of methodological aspects (e.g., trajectory tracking, and static and dynamic path planning) [6], navigation [7] a benchmark toolkit for the comparison of mobile robot modeling and determination of model parameters [8]. The topic "benchmarking" is also discussed in swarm robotics [9], planning [10], human/robot interaction [11] and domestic robotics [12]. The idea to provide reproducibility for groups with limited resources or different hardware has also been presented work the work on "Rat's Life" [13] though in the context of cognitive robots.

However, in the field of robot grasping, that has gained a great significance with the upcoming of many sophisticated anthropomorphic robot systems (e.g., [14], [15]) and dexterous hands such as [16], [17], [18], there is still little competitive effort to be seen. This research is closely related to the target objects and the operating environment, which frequently is desired to be human-centered. Among the few attempts are a benchmark that focuses on classification of daily life activities and provides a related categorization of objects [19] and web-based data bases which provide object models for grasping along with sets of stable grasps, namely the *Columbia Grasp Database* [20], the *KIT ObjectModels Web Database* [21], [22] and the selection of every day objects described in [23]. Metrics are also an important topic and crucial to benchmarking. Such a metric has been presented for scoring the quality and stability of grasps [24].

However, the necessity of the creation of standardized grasp benchmarks becomes increasingly recognized and is discussed among the community of robotics researchers [25], [26], [5], [27]. Probably the biggest obstacle preventing benchmarks is the still missing infrastructure necessary for the development of uniform test cases that could lead to comparative results. That is, only a small number of selected research institutions in the world have access to comparable hardware. A possible way to tackle this problem is to provide ground truth data as presented for the Nao robot [28]. On the other hand, techniques are developed for various different robotic frame works making it hard to let them run in a single environment. This heterogeneity is described in [29] and, since then, new competitors such as the prominent the *Robot Operating System* [30] have became available and gained a lot of popularity.

At least for the first problem, a possible solution is to leave the real hardware for benchmarking and rely completely on simulation instead, as suggested by Michel et al. By offering a simulation of the robot cup soccer competition and the Nao robot [31]. That way, the evaluation of individual components becomes be independent of the available hardware. Today, there exists a great number of robotics suits, several of them containing highly-developed simulators [32], [33],

The work described in this paper was partially conducted within the EU Cognitive Systems project GRASP (FP7- 215821) funded by the European Commission.

[30], [34], and a small number among them is designated to grasping [35], [36], [37], [38].

The biggest remaining problem is that grasping and grasp related manipulation is investigated within very heterogeneous environments, that is, on different robots using different middleware and programming languages. Therefore, an interface for the creation of well-defined experiments that can be performed under equal and stable conditions is presented in this paper. The vision driving this work is to encourage scientists world-wide working on the field of robotic grasping and grasp-related manipulation to participate in the process of creating a benchmark suite and feeding the results into a database. This database helps to evaluate many of the algorithmic components developed in their research field, to compete with them, and certainly will also facilitate the choice of the right components when designing new systems. This interface is designed to be extendable and open comes as a part of the open-source OpenGRASP toolkit [37] that builds on top of the OpenRAVE simulator [35]. It consists of four parts:

1) a Python-based interface that glues the simulator to evaluated algorithmic component,
2) a web service collecting and providing results of various benchmarks run by various groups,
3) a set of unified kitchen centered real-life objects
4) and a simulated environment that features a humanoid robot acting in a kitchen.

This architecture will be presented in the remainder of the document. The next section will present the components of the OpenGRASP toolkit including the new benchmark architecture and the GRASP model data-base. Afterwards, the organization of the benchmark architecture will be explained in detail and then, in section IV, two exemplary implementations (one for grasp and one for motion planning) and their results will be presented.

II. The OpenGRASP Toolkit

In robotics, simulation of robotic systems is an essential component in design and planning, and many industrial robot manufactures provide simulators for their robots. The OpenGRASP toolkit [37] is a new simulation environment that is dedicated to grasping. It allows the development and testing of new grasp-related algorithms as well as the modeling of new robots. The simulations are carried out within an improved version of the OpenRAVE simulator [35], which has been enhanced with extended sensor models, interchangeable physics engines and a tool for the creation of new robot models. With OpenGRASP, many of the building blocks required for the benchmarking environment are already available. The following sections highlight some of the most important features.

A. OpenGRASP Robot Editor

Based on the open source 3D modeling tool Blender.org [39], the OpenGRASP Robot Editor is not directly integrated into the simulator itself (see Fig. 1). It allows the convenient creation of new robot models and

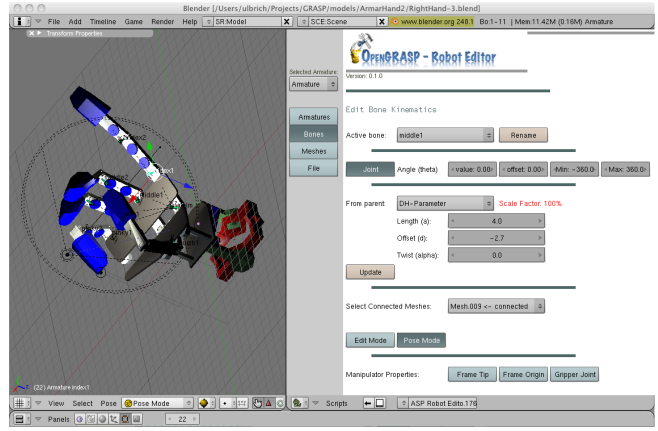


Fig. 1. A screenshot of the OpenGRASP Robot Editor.

the conversion from other file formats, and offers a scientific user interface that gives easy access to the many features of the underlying comprehensive modeling software. The key aspects of this software are:

- *geometric modeling:* The creation of new robots models requires a tool that excels in modeling of the geometric components (i.e., meshes),
- *semantic modeling:* The ability to allow the description of semantic properties, such as definitions of kinematic chains, sensors and actuators, or even specify algorithms,
- *dynamics modeling:* Definition of physical attributes of the robot's elements. At the moment, the focus lies on the dynamics of rigid bodies,
- *conversion:* Robot models usually come in a variety of different file formats and have to be converted first before they can be loaded into the editor.

For the storage of the models, an open, extensible and already widely accepted file format, which supports the definition of at least kinematics and dynamics, has been chosen. This is necessary in order to enable the exchange of robot models between supporting applications, leading to greater flexibility in the selection of appropriate tools. Due to its acceptance as an industry standard, the wide distribution, the now native support for kinematics, and a clear and extensible design, COLLADA in version 1.5 [40] has been selected as the preferred file format in OpenGRASP. At the time of writing, the OpenGRASP Robot Editor produces valid COLLADA documents and experimental support for the import has been added recently. Special annotations to the file that are processed by the OpenRAVE simulator have are also supported.

Having employed this editor, a great number of robot and hand models have already been created so far. Among them are the humanoid *ARMAR-III* [14], the *PA-10* robot, the *shadow hand*, the *Schunk SAH* and *SDH* hands, just to name a few. All of these models can be applied within the simulation and are available for selection within the configuration of the benchmarks.

B. Physics Simulation

The Physics Abstraction Layer (PAL) [41] is a software package created by Adrian Boing that renders the most common physics engines interchangeable. It is an abstraction layer that provides an interface to a number of different physics engines which allows to dynamically switch between them. This functionality adds even more flexibility to the OpenGRASP simulator, offering the possibility to choose the engine with the best performance [42], depending on the specific environment and task. Using this interface, it becomes also possible benchmark the different engines.

The OpenRAVE Physics Engine interface allows the simulator to apply different engines and different collision checkers. OpenGRASP replaces the basic physics engine in OpenRAVE, which is limited to offer an ODE[1] interface, by a new plugin that encapsulates PAL. It is capable of initializing PAL with a specific engine and, thus, eliminates the need to create different plugins.

C. Tactile Sensors

A new tactile sensor plugin provides simulation models of the *Weiss Robotics* DSA 9330 and DSA 9335 tactile matrix sensor modules [43]. The model simulates the mapping of contact locations to the sensor matrix cells of the sensing surface. Further, it allows specification of a linear characteristic between contact force and compression ratio of the deformable sensing surface to emulate pressure readings similar to the real sensor.

D. KIT ObjectModels Web Database

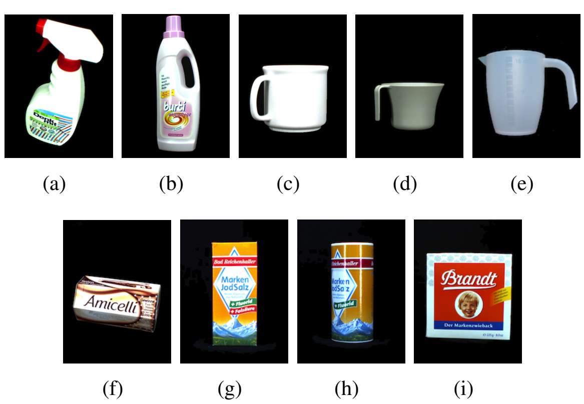


Fig. 2. Different views of every-day objects available in the *KIT ObjectModels Web Database*: a) Spray bottle, b) detergent bottle with handle, c) white cup with handle, d) green cup with handle, e) measure cup, f) chocolate box, g) rectangular salt box, h) cylindric salt box, and i) bread box.

The OpenGRASP benchmark features every-day objects that can be found in a human-centered scenario (i.e., a kitchen) [44]. Among them are objects of different shapes and sizes, colors and textures, and different topology and difficulty to grasp (see Fig. 2). All database entries are available as multiple-view stereo images, mesh data and point clouds. These objects can be selected from the benchmark GUI, are then automatically downloaded from the database, and integrated into the benchmark.

[1]Open Dynamics Engine

E. The Kitchen Scenario

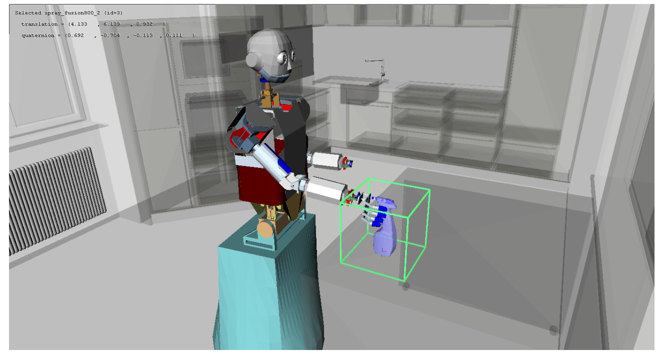


Fig. 3. Screenshot of the kitchen scenario from the Simulator.

A scenario is required for the grasp benchmarks which reflects a real-life situation which involves a real robot. For this reason, OpenGRASP proposes (and provides) a kitchen environment complete with furniture, realistic objects (see Sec. II-D) and a model of a real humanoid robot [14] with an anthropomorphic full-finger hand (see Fig. 3 for an actual screenshot of the scene). It is suggested that benchmarks use this standardized environment, optionally replacing the robot by their own hardware. However, it will also be possible for the community to define completely new scenarios reflecting different situations and places for new benchmarks.

III. The OpenGRASP Benchmark

The OpenGRASP Benchmark consists of four components:

1) *real-life scenarios* (for instance, the presented kitchen environment together with the graspable objects and the humanoid robot, see Sec. II-E),
2) a *web-service* that provides test cases, scenarios, robot models, as well as records the results, if requested. A "high-score" that displays the results of all participants of a benchmark is available on the web server,
3) a *control software* that communicates with the web-service and controls and monitors the simulation,
4) a *software interface* that serves as a connection/compatibility layer between the participant's observed algorithm, the benchmark, and the simulator.

That way, the architecture fulfills an adaption of the *model-view-controller* design pattern, see Fig. 4.

A. The Controller Software

The preferred way of extending and controlling the OpenGRASP simulation is through the access of OpenRAVE's Python interface. The controller software is consequently completely written in Python and starts OpenRAVE transparently in the background. On startup, it connects to the web service provided with the OpenGRASP Benchmark and fetches information about all available registered test

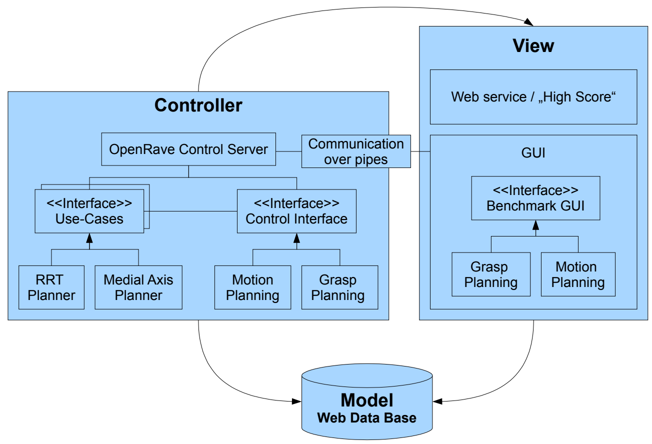


Fig. 4. The benchmarking frame work is structured according to the *model-view-controller* design pattern. The model consists of the web database and the local data storage. The GUI, the web "high-score", and the OpenRAVE interface form the view. Within the benchmark control software hierarchy, interfaces for test cases in each category of benchmarks are provided. For a new benchmark categories, a new "Benchmark GUI" and "Control Interface" has to be implemented. Special test cases (the more common case, e.g., the medial axis grasp planner) are derived from the "Use-Cases" interfaces classes.

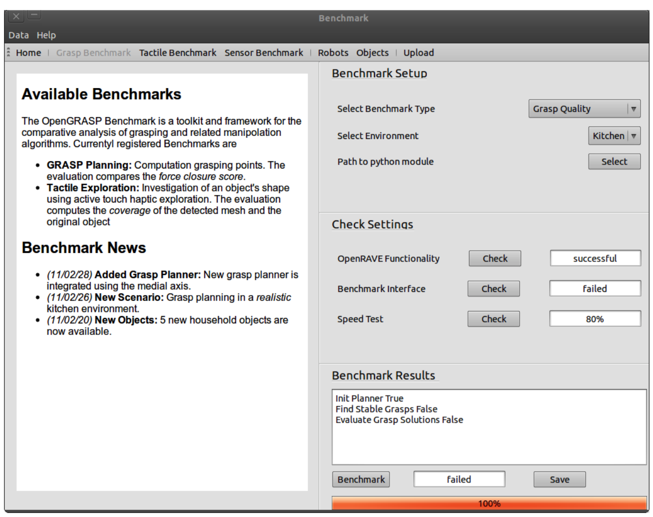


Fig. 5. A screenshot of the graphical user interface provided by the controller software of the benchmarking suite.

environments, benchmarks, models and user high-scores. A graphical user interface is created by the controller, which allows to select, setup and launch the benchmark (see Fig. 5). All benchmarks are organized in categories according to their domains. For instance, there are categories for grasp manipulations, such as motion planning (see Sec. IV-B) and grasp planning itself (see Sec. IV-A). Each of them can be configured to use one of the standardized and predefined benchmark environments, a robot manipulator, and a set of graspable objects. After a successful test run, the results can be chosen to be uploaded to the web-service for future reference or simply be displayed for personal information.

B. The Software Interface

There are several ways to evaluate an algorithm/technique in the OpenGRASP Benchmark. The most direct way is to provide it in form of an OpenRAVE plugin—either written as a C++ plugin or a compatible python script. It is planned to provide the same functionality for scripts written in MATLAB® or Octave. In case that the evaluand is none of the above, a glue layer is required. OpenRAVE's API is well documented, so a C-library, for instance, can be included easily within a plugin that acts as an adapter. Other interfaces, such as algorithms written for the *Robot Operating System (ROS)* [30] are supposed to be connected more easily. An example is described where an independent software is controlled through a network connection. Another example is presented in Sec. IV-B, where a C++ library is encapsulated in an python interface, so that it can be adressed directly by the bechmarking framework.

Currently, there exist two benchmark categories yet: one for *motion planning* and one for *grasp planning*. The benchmarks in this categories will be described in more details in Sections IV-A and IV-B. Given that the OpenGRASP benchmark is intended as a community effort, more benchmark categories and test cases are expected to follow. For every new test case, the author has to write an specific controller that inherits (in the sense of OOP) test routines and properties from the interface provided by the top-level controller in the same benchmark category. That is, authors integrating new algorithms do not have to implement the evaluation routines. This guarantees that all tests are evaluated in a similar manner, thus conserving objectiveness. New interfaces are automatically detected by the controller software and are can then distributed via the web service.

IV. BENCHMARKS

The OpenGRASP Benchmark is an development stage, and currently features two main test cases. However, due to its open design and extensible design, the list will be easily enlarged by new benchmarks that are requested by the community.

A. Grasp Planners

The grasp planner benchmark allows planning algorithms to compete against each other. Grasp planning aims at finding poses of the hand relative to the object and vectors of hand joint angles that, together, represent force closure grasps. That is, grasps where the object cannot move inside the hand if external forces and torques are applied. In this benchmark, grasp planning for objects of known shape is presented. The grasp planner presented by Berenson et al. [24] is integrated in OpenRAVE and serves as the reference in the presented example. Its competitor is the grasp planner based on the *"medial axis"* presented by Przybylski et al. [45], [46].

The two planners use different approaches to generate candidate grasps which are then tested for force closure. The grasp planning method by Berenson [24] uses surface normals of the object as approach directions for the hand toward the object, and a user-defined number of roll angles of the hand around the approach direction are tested. On the other hand, the grasp planner presented in [45], [46] generates candidate grasps by analyzing symmetry information of the objects contained in their medial axes.

Both planners test candidate grasps by placing the hand at an initial pose where it collides with the object. Then they retract it along the approach direction until it is no longer in collision with the object. Now the fingers of the hand close around the object. Finally, the contact points between the object and the robot hand are determined, and the force closure score is computed.

In the benchmark, a number of candidate grasps is generated with both methods for the set of objects described in Sec. II-D and two different robot hands: The *Barrett hand* and the *ARMAR-III* hand; the complete kitchen scenario is not necessary for this benchmark. The generation of candidate grasps was restricted to grasps where the palm is in direct contact with the object. Results are presented in Tab. I and II. The numbers of generated candidate grasps per object and the percentage of force closure grasps is displayed. The benchmark contains the element which is responsible of the actual evaluation of the results created by the two grasp planners, in terms of the force closure grasp. That way, it is ensured that the same evaluation criterion is applied to both planners.

TABLE I
ARMAR-III HAND: CANDIDATE GRASPS TESTED AND PERCENTAGE OF FORCE-CLOSURE (FC) GRASPS FOUND

Objects	MA-based planner Candidates	MA-based planner FC	Surface normals planner Candidates	Surface normals planner FC
Bread box	296	99.0%	842	16.5%
Prismatic box	516	99.8%	520	43.8%
Salt box	408	85.8%	408	41.2%
Salt can	1072	80.0%	616	48.4%
Detergent	1022	87.6%	928	38.5%
Spray	590	77.5%	762	29.5%
White cup	732	81.1%	456	48.2%
Green cup	514	51.8%	496	52.0%

TABLE II
BARRETT HAND: CANDIDATE GRASPS TESTED AND PERCENTAGE OF FORCE-CLOSURE (FC) GRASPS FOUND

Objects	MA-based planner Candidates	MA-based planner FC	Surface normals planner Candidates	Surface normals planner FC
Bread box	296	66.9%	842	11.8%
Prismatic box	516	20.2%	520	19.2%
Salt box	408	94.4%	408	47.3%
Salt can	1072	91.4%	616	44.2%
Detergent	1022	49.5%	928	25.8%
Spray	590	44.1%	762	13.9%
White cup	732	63.8%	456	37.5%
Green cup	514	34.8%	496	36.1%

B. Motion Planning

This benchmark offers a standarized interface for comparing motion planning algorithms in the context of grasping and manipulation. Here, the focus lies on sampling-based approaches (e.g., RRT [47] or PRM [48]) but an extension to other algorithms, such as potential field or grid-based approaches, is possible. The design of the benchmark allows to include any motion planning library, as long as python calls can be processed. Since most libraries are based on C++, python wrappers (e.g., *boost.python* [49]) can be used to enable C++ calls from within the benchmarking framework. In the following, this is demonstrated by extending the C++ library Simox [38] so that it can be accessed by python code. To guarantee, that the same data is used for all evaluations, the kinematic definitions as well as the 3D models of the robot, the environment, and the objects are extracted from the openRAVE framework and passed to a Simox import filter.

The results of the benchmarking scene are verified with the underlying OpenRAVE framework, whereas the following metrics are used for evaluation:

1) *Setup:* The time needed to setup the planner, including all neccessary steps for loading and data preperation. Usually this step could be performed on robot startup and thus it is not considered as part of the motion planning process.
2) *Planning:* The time needed to perform the motion planning.
3) *Post-Processing:* This measure represents the duration of potential post-processing steps (e.g., path smoothing).
4) *Correctness:* When using sampling-based planners, usually discrete collision detection (DCD) is applied for validating path segments. The required parameter, defining the step size between two samples, directly affects the planning time as well as the correctness of the results. Even with small step sizes there still remains a probability that a collision was missd by the DCD methods. Hence, the correctness of the resulting motion is evaluated by applying DCD methods with a tiny step size parameter. In future releases, continuous collision detection (CCD) methods will be integrated in order to guarantee the correctness of the results [50].
5) *Length:* The length of the resulting path in configuration space. In case grasping or manipulation motions are benchmarked, the length of the end-effector's motion in workspace is additionally measured.
6) *Clearance:* The average distance of the robot to obstacles during path execution.

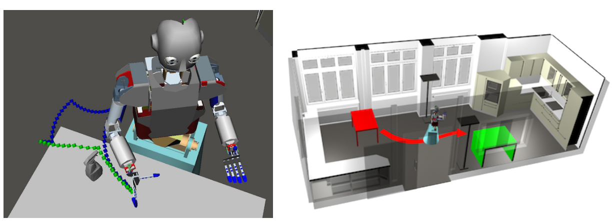
In order to offer well-defined scenes for differing grasping and manipulation tasks, several setups are initially provided by the framework which can be extended in further releases. All scenes are related to the humanoid robot *ARMAR-III* operating in a kitchen environment as described earlier in Sec. II-E:

1) *Standard motion planning:* In this scene, the start and goal configuration of the robot are predefined, so that standard motion planning algorithms (e.g. RRT or PRM) can be evaluated. The joints used for planning

cover the hip yaw joint and all seven DoF of the right arm of ARMAR-III. The setup together with exemplary solution paths can be seen in Fig. 6(a).

2) *IK-based motion planning:* The goal is not defined as a single point in configuration space, but as a grasping pose in workspace related to an object. Hence, two tasks have to be considered: solving the inverse kinematics (IK) problem and finding a collision-free motion towards the IK-solution. The position of the object is randomly sampled in front of the robot, so that no pre-calculated IK-solutions can be used. To solve this scene, either a two step algorithm (at first solve the IK, then plan the motion) or an integrated approach as the IK-RRT [51] can be used.

3) *Clean the Table:* This is the most challenging setup, where multiple objects are located on a table and the goal is to clean the table by transporting the objects to another area on the sideboard (see Fig. 6(b)). To plan the sequence of actions, multiple sub-tasks have to be solved, such as deciding which object should be grasped, finding IK-solutions, and planning collision-free motions for grasping and placing the objects. The considered configuration space covers an arm, the hip and the position and orientation of the robot's base.



(a) (b)

Fig. 6. (a) The setup of the first planning scene. An planned and a post-processed solution are depitced in blue and green. (b) Exemplary showcase of the *Clean the Table* scene, that will be offered by the motion planning benchmark. The goal of this benchmark is to transport all objects that are randomly located on the red table to the green sideboard without any collisions.

Currently, the setups are specified as described in this section and interface methods are provided within the benchmarking framework. A reference implementation of the first scene has been evaluated according to the proposed metrics (see Tab. III). It can be seen, that the sampling step size of 0.05 produces several incorrect results after post-processing. This is caused by the shortcut algorithm, which tends to generate motions that come close to obstacles and hence the probability of an undetected collision increases due to the applied discrete collision detection (DCD) methods. Further information about the paramter setup can be found online[2]. In the near future we will upload more evaluations based on Simox, and thus serving reference benchmarks as a basis for comparison of different approaches for motion planning.

[2]Address: `http://wwwiaim.ira.uka.de/GraspBenchmark`

TABLE III
EVALUATION OF THE FIRST MOTION PLANNING SCENE.

Average time		Solution	Planned	Postprocessed
Setup	1.44s	Valid	98.1 %	76.92 %
Planning	0.41s	Length	6.73	2.85
Postprocessing	1.17s	Clearance	111.12 mm	44.29 mm

V. Conclusion and Future Work

In this work, the development of a new environment for benchmarking was presented which is integrated in the OpenGRASP toolkit. It will provide a complete tool chain for the integration of various grasp and grasp-related manipulation algorithms, including many robot models and a tool to design new ones, a database containing standardized graspable items, an real-life kitchen environment featuring a complete humanoid robot with anthropomorphic hands. Its individual components were described in detail. The benchmarks are configured and launched using a control program with a graphical user interface that connects to a web service that administers the available benchmarks, scenarios, models and a benchmarking "high-score" that records the participants' results.

OpenGRASP Benchmark is intended as a first step in the direction of an open platform for comparative analysis of algorithmic technologies related to grasping and grasp manipulation. The project will be heavily depending on the robotics community, and further development of test cases and scenarios will rely on suggestions and input of fellow robot scientist. This is why, at the time of writing, there still exist only the two presented benchmarks: *grasp planning* and *motion planning*. Further advances will probably occur in the form of the design and definition of additional benchmarks and scenarios. Areas likely to be covered are hereby *grasping and re-grasping*, *tactile exploration*, *pre-/post-grasp manipulations*, *pick and place actions* and of cource the design of many more daily-life scenarios.

References

[1] F. Amigoni, S. Gasparini, and M. L. Gini, "Good experimental methodologies for robotic mapping: A proposal," in *ICRA*, 2007, pp. 4176–4181.

[2] W. Nowak, A. Zakharov, S. Blumenthal, and E. Prassler, "Benchmarks for mobile manipulation and robust obstacle avoidance and navigation," in *BRICS Deliverable D3.1*, 2010.

[3] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The Great Robot Race*, 1st ed. Springer Publishing Company, Incorporated, 2007.

[4] H. Kitano, M. Asada, I. Noda, and H. Matsubara, "RoboCup: robot world cup," *Robotics & Automation Magazine, IEEE*, vol. 5, no. 3, pp. 30–36, 1998.

[5] T. Wisspeintner, T. van der Zant, L. Iocchi, and S. Schiffer, "Robocup@home: Results in benchmarking domestic service robots," in *RoboCup Symposium 2009*, vol. 5949, Graz, Austria, 2009, inproceedings, pp. 390–401.

[6] J. Baltes, "A benchmark suite for mobile robots," in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 2, 2000, pp. 1101 –1106.

[7] Z. Tuza, J. Rudan, and G. Szederke? andnyi, "Developing an integrated software environment for mobile robot navigation and control," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, Sept. 2010, pp. 1 –6.

[8] S. Egerton and V. Callaghan, "A benchmark for measuring mobile robot environment modelling performance," in *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, vol. 1, Dec. 2004, pp. 407 – 412 vol.1.

[9] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm Intelligence*, vol. 2, no. 2, Dec. 2008.

[10] A. E. Howe and E. Dahlman, "A critical assessment of benchmark comparison in planning," *Journal of Artificial Intelligence Research*, vol. 17, pp. 1–33, 2002.

[11] M. Ralph and M. A. Moussa, "An integrated system for user-adaptive robotic grasping," *Trans. Rob.*, vol. 26, pp. 698–709, August 2010.

[12] I. Iossifidis, G. Lawitzky, S. Knoop, and R. Zllner, *Towards Benchmarking of Domestic Robotic Assistants*. Springer Press, 2004, vol. 14/2004, pp. 403–414.

[13] O. Michel, F. Rohrer, and Y. Bourquin, "Rat's Life: A Cognitive Robotics Benchmark," in *European Robotics Symposium 2008*, ser. Springer Tracts in Advanced Robotics, H. Bruyninckx, L. Přeučil, and M. Kulich, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 44, ch. 23, pp. 223–232.

[14] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Proc. IEEE-RAS International Conference on Humanoid Robots-06*, Dec. 2006, pp. 169–175.

[15] G. Sandini, G. Metta, and D. Vernon, "The iCub cognitive humanoid robot: an open-system research platform for enactive cognition," in *50 years of artificial intelligence*, M. Lungarella, R. Pfeifer, F. Iida, and J. Bongard, Eds. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 358–369.

[16] Shadow Robot Company, "Design of a dextrous hand for advanced CLAWAR applications," in *Climbing and Walking Robots and the Supporting Technologies for Mobile Machines*, 2003, pp. 691–698. [Online]. Available: http://www.shadowrobot.com/

[17] I. Gaiser, S. Schulz, A. Kargov, H. Klosek, A. Bierbaum, C. Pylatiuk, R. O. T. Werner, T. Asfour, G. Bretthauer, and R. Dillmann, "A new anthropomorphic robotic hand," in *Proc. IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2008, pp. 418–422.

[18] M. Grebenstein, M. Chalon, G. Hirzinger, and R. Siegwart, "Antagonistically driven finger design for the anthropomorphic DLR hand arm system," in *HUMANOIDS 2010*, Dezember 2010.

[19] K. Matheus and A. M. Dollar, "Benchmarking grasping and manipulation: Properties of the objects of daily living," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, 2010.

[20] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ser. ICRA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3343–3349.

[21] A. Kasper, R. Becher, P. Steinhaus, and R. Dillmann, "Developing and analyzing intuitive modes for interactive object modeling," in *Proceedings of the Ninth International Conference on Multimodal Interfaces (ICMI 07)*, 2007.

[22] Z. Xue, A. Kasper, M. J. Zoellner, and R. Dillmann, "An automatic grasp planning system for service robots," in *14th International Conference on Advanced Robotics*, 2009.

[23] F. Röthling, R. Haschke, J. J. Steil, and H. J. Ritter, "Platform portable anthropomorphic grasping with the Bielefeld 20-DOF Shadow and 9-DOF TUM hand," in *Proc. Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE. San Diego, California, USA: IEEE, Oct 2007, pp. 2951–2956.

[24] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids07)*, Dec. 2007.

[25] A. P. del Pobil, "Benchmarks in robotics research," in *Lecture Notes for IROS 2006 Workshop II(WS-2)*, October 2006.

[26] R. Dillmann, "Ka 1.10 benchmarks for robotics research," 2004.

[27] E. R. R. Network, "Survey and inventory of current efforts in comparative robotics research – introduction: Benchmarks in robotics research."

[28] T. Niemueller, A. Ferrein, G. Eckel, D. Pirro, P. Podbregar, T. Kellner, C. Rath, and G. Steinbauer, "Providing Ground-truth Data for the Nao Robot Platform," in *Proc. of RoboCup Symposium 2010*, 2010.

[29] A. Shakhimardanov and E. Prassler, "Comparative evaluation of robotic software integration systems: A case study," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, October 2007, pp. 3031–3037.

[30] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[31] O. Michel, Y. Bourquin, and J.-C. Baillie, "RobotStadium: Online Humanoid Robot Soccer Simulation Competition," in *RoboCup 2008: Robot Soccer World Cup XII*, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5399, ch. 50, pp. 580–590.

[32] P. I. Corke, "A robotics toolbox for MATLAB," *Robotics & Automation Magazine, IEEE*, vol. 3, no. 1, pp. 24–32, 1996.

[33] J. Jackson, "Microsoft robotics studio: A technical introduction," *Robotics Automation Magazine, IEEE*, vol. 14, no. 4, pp. 82 –87, Dec. 2007.

[34] B. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *11th International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, Jun. 2003, pp. 317–323.

[35] R. Diankov and J. Kuffner, "OpenRAVE: A planning architecture for autonomous robotics," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34, July 2008.

[36] A. Miller and P. Allen, "GraspIt!: A versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[37] B. Len, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, and R. Dillmann, "OpenGRASP: A toolkit for robot grasping simulation," in *2nd International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Darmstadt, Germany, November 15-18 2010.

[38] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Simox: A simulation and motion planning toolbox for C++," Karlsruhe Institute of Technology (KIT), Tech. Rep., 2010, (Online) http://www.sourceforge.net/projects/simox.

[39] "Blender." [Online]. Available: \url{http://Blender.org}

[40] "Collada." [Online]. Available: http://collada.org

[41] "Pal (physics abstraction layer)." [Online]. Available: \url{http://www.adrianboeing.com/pal}

[42] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems," in *5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia (GRAPHITE'07)*, Perth, Australia, 2007, pp. 281–288.

[43] Weiss Robotics. [Online]. Available: \url{http://www.weiss-robotics.de/en.html}

[44] D. Gonzalez-Aguirre, T. Asfour, and R. Dillmann, "Towards stratified model-based environmental visual perception for humanoid robots," *Pattern Recognition Letters*, 2010.

[45] M. Przybylski, T. Asfour, and R. Dillmann, "Unions of balls for shape approximation in robot grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[46] ——, "Planning grasps for robotic hands using a novel object representation based on the medial axis transform," in *IROS*, 2011.

[47] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE Int'l Conf. on Robotics and Automation (ICRA'2000), San Francisco, CA*, 2000, pp. 995–1001.

[48] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, 1994.

[49] D. Abrahams and R. W. Grosse-Kunstleve, "Building hybrid systems with boost.python," 2003, Boost Consulting.

[50] M. Tang, Y. Kim, and D. Manocha, "Ccq: Efficient local planning using connection collision query," in *Algorithmic Foundations of Robotics IX*, ser. Springer Tracts in Advanced Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. Lin, Eds. Springer Berlin / Heidelberg, 2011, vol. 68, pp. 229–247.

[51] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *Intelligent Robots and Systems, IROS*, October 2009.