An Automatic Grasp Planning System for Service Robots*

Zhixing Xue, Alexander Kasper, J. Marius Zoellner and Ruediger Dillmann
Forschungszentrum Informatik
Haid-und-Neu-Str. 10-14, 76137 Karlsruhe, Germany
xue@fzi.de

***Abstract*— Service robots need many different informations about the objects, they want to grasp and manipulate. Besides the physical information such as geometry and weight, semantic information about the objects is also needed. To model both of these types of information, we have constructed a multimodal object modeling center. It enables the modeling of physical properties of the object, such as the textures and the 3D geometry, with a digitizer and a pair of movable stereo cameras. Other properties of the objects relevant for grasping can also be automatically computed. Furthermore, a human teacher can communicate with the system through multimodal techniques to introduce the semantic information relevant grasping to the system. We have implemented a grasp planning system based on the grasp simulator "GraspIt!" to plan high quality grasps. The semantic information is represented as shape primitives, which are treated by the grasp planning as obstacles or must-touch regions of the object to influence the resulting grasps. The modeled physical, semantic and automatically computed information, together with the computed grasps are saved in a database, which provides the service robot the needed knowledge to grasp and manipulate various household objects.**

***Index Terms*— grasp planning, manipulation**

Fig. 1. A Service robot with two KUKA LWR Arms and two Schunk Anthromorph Hands which can grasp typical objects in household.

I. Introduction

Grasping and manipulation are the key functions of service robots to help people with their household tasks. To grasp and manipulate real world objects, detailed information about the objects needs to be made available to the robot system, so that the robot can autonomously detect, localize and manipulate them. In this paper, we present an automatic grasp planning system with an object modeling system, which can model the objects in a household environment and compute matching grasps for these objects.

To grasp and manipulate objects, a service robot needs a variety of information about the real world. It needs the position and the orientation of the object to be grasped, which can be computed by object localization algorithms using color cameras. These algorithms need further appearance information like the textures of the object. The geometric model of the object should be available as well, to plan the region where the robot can touch and grasp the object. The material properties are very important to assume the friction coefficient between the robotic finger and the object. The weight of the object should also be considered during the manipulation. To perform the grasp safely, the robot needs also information about the surroundings of the object.

The information listed above can be gathered with suitable sensors such as laser scanners or color cameras. There is however other functional, user-specific or semantic knowledge, which is difficult to obtain without human help. For example, how can a robot know, that a cup should be kept upright during the transport, if it is filled with water? Humans are able to interact with their environment in a very successful way because they have a very detailed model of all the things surrounding them. While growing up we acquire a vast set of different informations describing many aspects of items we see and use, ranging from appearance over haptic sensations, texture and smell, to such abstract terms like containers. To enable a robotic system to use such information for manipulation tasks, it is important to find ways how a human can transfer his/her knowledge about the world to a robot in an efficient and easy way. Semantic object modeling is a method to achieve this for a part of the available knowledge.

*This work was funded as part of the research project DESIRE by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01E. The research leading to these results has been supported by the DEXMART Large-scale integrating project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

In the described scenario this means that we focus on object knowledge that is crucial to grasping and manipulating.

We have constructed an object modeling center to model these physical properties of the objects in the household for service robots [1]. A digitizer scans the 3D geometry model of the object, which is placed on a rotational turntable. A movable stereo camera system is used to take images of the object from different viewpoints. A human teacher can communicate with the system in an intuitive way to model the semantic information. Microphones are used for speech recognition. A magnetic field tracker allows the user to demonstrate manipulation operations to the system. All of these informations about the objects are saved in an object database. Other informations of the object such as stable grasps for specific robotic hands will be further computed automatically and also saved into the object database. The goal is to create a knowledge base for the service robots, so that they can grasp and manipulate a large variety of known household objects.

The modeled information is used by an automatic grasp planning system to plan high quality and executable grasps. The grasp planning system deals with mainly four problems: *grasp planning* tries to find the configuration of the robotic hand in which the hand grasps the object with contact points between the fingers and the object surface. These contact points are evaluated by *grasp analysis* to check if the object can be held firmly and save, also in the presence of disturbances acting on it. In this phase, the approximation of the possible forces that can be applied by the hand onto the object is commonly used to rate the grasp. To get the discrete forces which act "optimal" on the object is the problem of *grasp force optimization*. After the optimal forces are computed, they are performed by *grasp execution*. After the computation, the planned grasps are also saved into the object database, so that the robot can (re-)use this knowledge at run time to grasp the object, as shown in Fig. 1.

Main contributions of this paper are:

1) The concept and the setup of a semantic object modeling system, which can gather not only the physical properties of the objects with suitable sensors, but also the semantic information by exploiting human knowledge.
2) An automatic grasp planning system uses this information to plan high quality grasps for robotic hands.
3) Objects in a household environment can be grasped after the introduced modeling and computation process.

The paper is laid out as follows: the next section introduces our grasp planning system. The used hardware in the modeling center and the semantic object modeling process is introduced in Sec. III. Experimental results are presented in Sec. IV. The work is summarized in Sec. V.

II. Grasp Planning System

A. State of the art

Grasp planning deals with a high dimensional space. Besides the internal degrees of freedom of the robotic hand, the relative position and orientation between hand and object are also to be considered. Grasp planning tries to find the connection between a configuration in this high dimensional space and the contact points, which will be used to evaluate the grasp quality. The grasp planning problem can be solved in either forward or backward direction. The forward solution involves the finger forward kinematic to close the fingers and uses the collision detection technique to detect the finger joint positions at collision, such as the grasp planning simulator "GraspIt!" [2]. Miller et al. [3] have used hand preshapes before grasping to shrink the hand configuration space. The relative position and orientation between hand and object are reduced to grasp starting positions and directions for the robotic hand in the simulation, which are generated using shape primitives decomposed from the object's geometric model. This decomposition from arbitrary object geometry to shape primitives, such as spheres, boxes, cylinders or cones, can not be performed in an automatic way. From the point clouds of the object, superquadrics [4] and minimum volume bounding boxes [5] were introduced to generate the approach directions, which do not need manual decomposition of the object and can be done automatically. The backward method is object centered. Contact points are randomly [6] or analytically located on the object surface to evaluate the grasp quality, without considerations about the hand kinematic. If the grasp quality is high, an inverse kinematic algorithm for the finger is used to find the corresponding feasible finger joint position. The collision between the fingers should be further checked to avoid self collision [7]. A main drawback is here, however, the inverse kinematic algorithm needs an unambiguous position of the contact point. This limits the grasps found to be only with fingertips, whereas a grasp with more than one contact points by one finger can be found by the forward method so that the object can be grasped more firmly. Because the finger joint positions are determined without consideration of collision, the backward method can not find collision free feasible grasps in the presence of obstacles.

After a grasp is found, its grasp quality is rated with some given criteria using the contact points found in the grasp planning step. To check if the object can be grasped firmly in the hand, both the forces and the torques are to be considered. With a fixed reference point to the object, the torque acting onto the object by the modeled force can also be computed, which together with the force forms a 6D vector *wrench*. The force closure and form closure properties have been intensively studied in the past. A force closure grasp can apply a wrench required to resist any external disturbances. If and only if a grasp with frictionless point contact model achieves force closure, it is also form closure, where the grasp completely immobilizes the grasped object [8]. Another often used grasp quality is measured considering the largest perturbation wrench that the grasp can resist with independence of the perturbation direction (largest sphere in the grasp wrench space) using point contact with friction (PCWF) as contact model [9]. By grasp quality computation, the possible forces acting at the contact points are used to quantify a grasp, the exact direction and magnitude of the forces are to be optimized. Based on the observation by Buss, Hashimoto and Moore [10] that the nonlinear friction cone constraints can be transformed into positive definiteness constraints imposed on certain symmetric matrices, Han, Trinkle and Li [11] have further written the positive definiteness as a linear combination of matrices. The contact force optimization problem can then be solved using the determinant maximization as a linear matrix inequalities problem. These two works, solving as linear matrix inequalities and gradient method, were combined very well by Liu and Li [12] with a solution for the initialization. These optimized forces can be applied onto the object by joint torque impedance control [13].

B. Implemented grasp planning system

The grasp simulator "GraspIt!" [2] is used to plan high quality grasps. In this simulation, the modeled robotic hand is set to a preshape with finger postures at a starting position and along an approach direction towards the object. Hand preshapes with different finger postures for the Schunk Anthropomorphic Hand (SAHand) [14] have been defined. Superquadrics are computed automatically from the object's scanned point clouds by a split-and-merge algorithm. The surface function of the generated superquadrics is sampled to compute the approach directions and the starting positions. The approach direction is defined as the opposite direction of the normal at the sampled point, towards the object. The starting position for the hand is defined at the sampled point a small distance away from the object, so that the hand does not collide with the object initially, see Fig. 2.. Compared to the method introduced in [3], this process does not need manual decomposition of the object and can be done automatically.

During hand moving and finger closing, it is desired to find the first contact point between the robotic hand and the object using the collision detection technique. This is a problem of *continuous collision detection*, which does not only check the collision between two static objects, but also takes into account the relative motion between them and finds the first time of contact (TOC). Compared to discrete collision detection, continuous collision detection does not miss collisions between too fast moving objects or between very thin objects. In [2], a Newton-Raphson method was used to find the contact point with the complexity of $O(\log n)$. We have integrated a continuous collision detection library

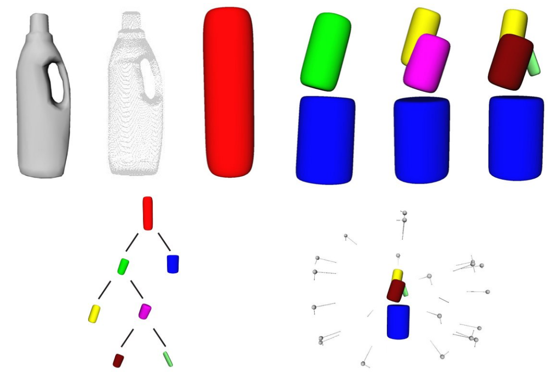

Fig. 2. Hierarchical decomposition of the object in multiple superquadrics, which are used to generate the approach direction for grasp planning as introduced in [4].

based on conservative advancement [15], [16] into "GraspIt!" to find all the contact points. Conservative advancement [17] is an efficient method to approximate two objects to each other. It uses an upper bound of the relative motion and the distances between the two objects in each time step to approximate them without collision. In the next iteration step, the moving object is placed at the computed position, until the distance between them is smaller than a predefined distance threshold, with which the objects are treated as colliding and the contact point between them is found. Zhang et al. [18] uses conservative advancement to compute contact points and extends the supported objects from convex to non-convex polyhedrons. The work is further extended to articulated models [16] with Taylor models and temporal culling. The collision between one finger link and one object can be computed within only a few iterations. Its complexity is reduced to a constant value, $O(1) \approx 2.1$ in our tests, with $0.1mm$ as the distance threshold. With the efficient continuous collision detection technique, each grasp candidate can be computed within only $50msecs$.

The closing algorithm works as follows:

1) Move robot hand with the preshape from the starting position along the approach direction to the object until it collides with the object. Only one TOC query is needed.
2) Try to close each finger separately by one TOC query. If a link collides the object, move the following links beneath the colliding link.
3) The algorithm stops, if all fingers are colliding with the object or reach their maximal joint positions.

After the object is grasped, the contact points between the hand and the object are collected to evaluate the grasp quality. "Largest sphere in grasp wrench space" is used as the grasp quality measurement [9]. The algorithm by Han et al. [11] is further used to compute the optimal contact force as a linear matrix inequality problem. At execution time, such

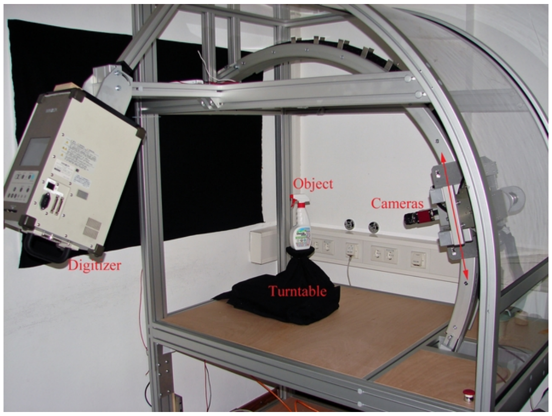

Fig. 3. Object modeling center for object digitization

computed optimal contact force is applied onto the object by joint torque based impedance control of the fingers [13]. The torque values τ_{act} at the finger joints θ are computed from the resulting optimal contact wrenches ω of the force optimization:

$$J^T(\theta)\omega + g(\theta) = \tau_{act} \tag{1}$$

with $J^T(\theta)$ being the transpose of the hand Jacobian and external load $g(\theta)$, such as gravity. To let the fingers apply τ_{act}, we use embedded joint torque based impedance control, based on the following well-known control scheme:

$$\tau = M\ddot{\theta} + D\dot{\theta} + K(\theta_{ref} - \theta) \tag{2}$$

with positive definite matrices M, D and K representing the virtual inertia, damping and stiffness of the system and θ_{ref} being the commanded joint position. After the object is grasped in the hand, the fingers are in steady state, such that $\ddot{\theta} = \dot{\theta} = 0$, $\tau = \tau_{act}$, and we receive the reference joint positions for the impedance control:

$$\theta_{ref} = \theta + K^{-1}\tau_{act} \tag{3}$$

It is to notice that with the same virtual inertia, damping and stiffness parameters, the computed reference joint positions of the fingers do not change. The found contact point information, the finger joint positions with which the hand and the object collide and the reference joint positions for the impedance control are all saved in the database.

III. SEMANTIC OBJECT MODELING

A. 3D geometry as a modeling basis

Since humans in general heavily rely on their vision to gain information about the environment, a visual representation during the whole modeling process is very beneficial. To exploit this we need a visual representation of the objects in question to display to the user during modeling to enable direct feedback. Geometry is also crucial for grasp planning and evaluation, which means that it would be very beneficial

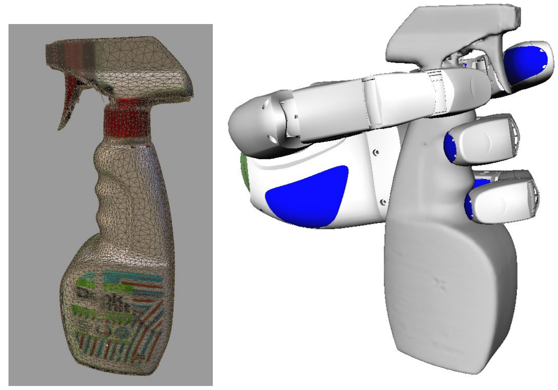

Fig. 4. Digitized object with texture and planned grasp by the grasp planning with human help. The informations that the object has a rotational part and the rotational axis of it are interactive modeled.

to be able to quickly generate a geometric representation of a real world object.

To generate these representations an object modeling center (see Fig. 3) was built which employs a Minolta Vi-900 digitizer, a turntable and a pair of Marlin cameras to scan real world objects and recreate their geometry and texture. The objects are put on the turntable and are rotated for different view points. First, the geometry of the object is acquired using the Minolta digitizer. It has a resolution of 640×480 with less than 0.2 mm error distance. Registration and further editing of the point clouds as well as triangulation is done with a software based on the commercial library "Rapidform.dll" [19]. When the geometric representation is complete, the color stereo camera images with a high resolution of 1392×1038 pixels are taken from different angles. In this case, both the object can be rotated, and the camera system is mounted to a rotating bracket which can turn up and down (between -25 and 75 degrees, compared to a level angle). This allows us to get views e.g. into a cup from above. With high precision calibration between the turntable, the rotating bracket, the stereo cameras and the digitizer, these images can then be mapped to the geometry as a texture. This setup can create highly detailed 3D models of many kinds of objects (see e.g. Fig. 4). The 3D models generated this way are the basis for the following modeling process and grasp planning. The generated textured models are also published in a publicly accessible web database [20]. Anyone interested can download textured meshes and camera images modeled by this system.

B. Object knowledge for grasping

After the object geometry is modeled, some geometric properties and representations of the object can be directly computed. The stable planes of the object and its middle planes are used for the grasp planning.

If an object is placed on a horizontal table, the face

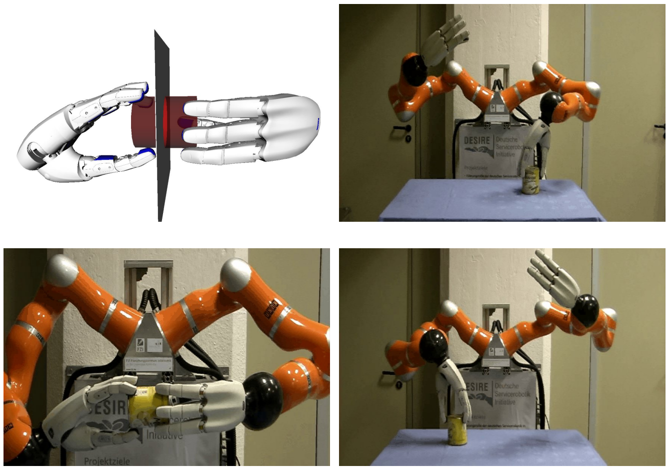

Fig. 5. Top left: planing bimanual grasps using middle planes. Top right: real execution, the object is grasped by left hand. Lower left: the object is delivered from left to the right hand. Lower right: the object is placed by the right hand.

Fig. 6. Possible 3D representations of semantic object properties. From left to right: movable part with rotation axis; opening for pouring; decomposition to separable parts.

of its convex hull in contact with the table is a stable plane. The stable plane can be interactively modeled [21], or automatically computed by projecting the center of mass of the object onto faces of its convex hull [22]. During grasp planning, one of the stable planes is placed in the grasp simulation as an obstacle to plan grasps that do not collide with such a stable plane. This way, feasible grasps for all possible poses of the object on a table are found. Also, the combination of two stable planes are now placed together in the simulation to find other collision free grasps. These grasps can be used to plan regrasp operations to change the object pose by only one hand [22].

The bounding box of the object can be easily computed from its geometric data. The three middle planes of the object represent a simple decomposition of the object. If the object is too big for the robotic hand, this decomposition can be used to plan bimanual grasps. A plane is placed at the middle of the object and treated as an obstacle. Grasps for the left and right hand without collision with the plane can be computed separately. If force closure grasps of the two hands can be found, they can be combined arbitrarily because the two hands do not collide with each other, due to the decomposition by the middle plane. This kind of bimanual grasps are used to hand over the object from one hand to the other, as depicted in Fig. 5.

C. Semantic knowledge for grasping

Besides the object properties analyzed above, several other object properties can have a big impact on how an object is grasped best. If the object possesses an opening e.g. like a cup and one would like to pour some milk into the cup while holding it, grasping the cup in such a way that the hand is covering the opening would be very impractical [23]. Or maybe if a hot pan should be moved from one place to another, the hot part should not be grasped but rather the handle should be used for that action. How can a robot system incorporate that knowledge into its grasp planning to avoid these pitfalls and use reasonable grasps for all the different objects? Again our paradigm is that a human already knows how to do that and he can share that knowledge with the robot. Since grasp planning is commonly done on a 3D geometrical representation of an object it seems only natural to incorporate that kind of information into that representation so it can easily be extracted to influence grasp planning in a positive way. To achieve that we propose the creation of several special 3D shape primitives that correspond to different object properties affecting grasping. A human "teacher" can then use and adapt these primitives to a given object to ensure proper description of the object's uniqueness. The exact number and shape of the primitives is still subject of research, but a set of possible primitives can be seen in Fig. 6. The idea is to create a modeling environment in which a human user can use the real object to demonstrate to the system the different properties in a natural way by using pointing and other gestures. The semantic information is represented by such primitives for the grasp planning. The primitives are treated either as obstacles or as parts of the object. With the primitives as obstacles, the covered region of the object, such as the opening part of a cup, will not be touched by the hand. If the primitives are treated as part of the grasp, after a feasible grasp is found, only the grasps that collide with the primitive are desired ones. This is to make the hand to grasp only in the selected region of the object.

To manage and use the modeled semantic information is beyond the scope of this paper. In the current system, the modeled semantic information is used to influence the automatically planned grasps and save it into the object database. During the real execution, the robot can use the semantic and context information to access and use the saved grasps for specific tasks.

IV. Experimental Results

Currently, more than 40 objects have been modeled at the modeling center. Grasps for all of the objects can be automatically computed and are saved into the database. Some of the system-known objects are depicted in Fig. 1. The grasps for the objects in the grasp database contain the grasps for left and right SAHand. The number of the saved grasps ranges from 200 to 600. If the object is placed on the table and well reachable by the arm, it can be grasped, no matter what orientation it has. By the use of stable planes, such grasps are computed and saved in the grasp database.

We have tested the introduced system together with object localization, grasping and environment modeling [24]. After the object is localized based on its SIFT features, a grasp for the object is searched in the grasp database, which can be performed by the robotic arm without any collision with the environment. For placing, the reverse process of grasping, it is also checked, if the placing operation can be executed collision free. After a feasible grasp is found, where the fingers of the robotic hand only collide with the object at desired contact points and without any other collision between the robot and the environment, a probabilistic collision free path planner [25] is used to bring the robotic arm to the starting position. After the object is grasped, it is also treated as part of the kinematic chain for the following collision free path planning to avoid collisions between the grasped object in the hand and the obstacles.

V. Conclusion And Future Work

In this paper, an automatic grasp planning system with object modeling center was presented. It provides a service robot knowledge about the objects in a household, such as a 3D model of the object, how the object can be grasped by a robotic hand and the appearance of the object for visual localization. Other semantic, user-specific and context knowledge can also be modeled interactively by the user at the object modeling center. To improve the efficiency of the grasp simulation, continuous collision detection was used to quickly find contact points between the object and the robotic hand. If a new object is brought to the modeling center, after a few hours of modeling and computation, it can be grasped by the service robot.

References

[1] R. Becher, P. Steinhaus, and R. Dillmann, “Interactive object modelling for a humanoid service robot,” in *IEEE International Conference on Humanoid Robots*, 2003.

[2] A. Miller and P. Allen, “Graspit! a versatile simulator for robotic grasping,” *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[3] A. Miller, S. Knoop, H. Christensen, and P. Allen, “Automatic grasp planning using shape primitives,” in *IEEE International Conference on Robotics and Automation*, vol. 2, 14-19 Sept. 2003, pp. 1824–1829.

[4] C. Goldfeder, P. Allen, C. Lackner, and R. Pelossof, “Grasp planning via decomposition trees,” in *Robotics and Automation, 2007 IEEE International Conference on*, 10-14 April 2007, pp. 4679–4684.

[5] K. Huebner, S. Ruthotto, and D. Kragic, “Minimum volume bounding box decomposition for shape approximation in robot grasping,” in *IEEE International Conference on Robotics and Automation*, May 19-23 2008, pp. 1628–1633.

[6] C. Borst, M. Fischer, and G. Hirzinger, “Grasping the dice by dicing the grasp,” in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, 27-31 Oct. 2003, pp. 3692–3697.

[7] C. Borst, M. Fischer, and G. Hirzinger, “Efficient and precise grasp planning for real world objects,” in *Multi-point Interaction with Real and Virtual Objects*, ser. Springer Tracts in Advanced Robotics, F. Barbagli, D. Prattichizzo, and K. Salisbury, Eds. Springer, 2005, vol. 18, pp. 91–111.

[8] X. Zhu and J. Wang, “Synthesis of force-closure grasps on 3-d objects based on the q distance,” in *IEEE Transactions on Robotics and Automation*, 2003, pp. 669–679.

[9] C. Ferrari and J. Canny, “Planning optimal grasps,” in *IEEE International Conference on Robotics and Automation*, 12-14 May 1992, pp. 2290–2295.

[10] M. Buss, H. Hashimoto, and J. Moore, “Dextrous hand grasping force optimization,” *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 406 – 418, 1996.

[11] L. Han, J. Trinkle, and Z. Li, “Grasp analysis as linear matrix inequality problems,” *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 663 – 674, Dec 2000.

[12] G. Liu and Z. Li, “Real-time grasping-force optimization for multifingered manipulation: theory and experiments,” *IEEE/ASME Transactions on Mechatronics*, vol. 9, pp. 65 – 77, March 2004.

[13] Z. Xue, J. M. Zoellner, and R. Dillmann, “Automatic optimal grasp planning based on found contact points,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 02-05 Jul. 2008, pp. 1053–1058.

[14] H. Liu, P. Meusel, N. Seitz, B. Willberg, G. Hirzinger, M. Jin, Y. Liu, R. Wei, and Z. Xie, “The modular multisensory dlr-hit-hand,” *Mechanism and Machine Theory*, vol. 42, pp. 612–625, 2007.

[15] L. Zhang, G. Bao, Q. Yang, J. Ruan, and L. Qi, “Static model of flexible pneumatic bending joint,” in *Control, Automation, Robotics and Vision, 2006. ICARCV ’06. 9th International Conference on*, 5-8 Dec. 2006, pp. 1–5.

[16] X. Zhang, S. Redon, M. Lee, and Y. J. Kim, “Continuous collision detection for articulated models using taylor models and temporal culling,” *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, vol. 26, no. 3, p. 15, 2007.

[17] B. V. Mirtich, “Impulse-based dynamic simulation of rigid body systems,” Ph.D. dissertation, University of California, Berkeley, 1996.

[18] X. Zhang, M. Lee, and Y. J. Kim, “Interactive continuous collision detection for non-convex polyhedra,” in *The Visual computer (Proceddings of Pacific Graphics 2006)*, vol. 22, 9-11, 2006.

[19] “Rapidform.dll: Software development kit for 3rd party 3d scanning application developers,” http://www.rapidform.com, 2009.

[20] “Kit objectmodels web database,” http://wwwiaim.ira.uka.de/ObjectModels.

[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, J. M. Zoellner, and R. Dillmann, “Planning regrasp operations for a multifingered robotic hand,” in *IEEE Conference on Automation Science and Engineering (CASE)*, 23-26 Aug. 2008, pp. 778–783.

[23] T. Baier and J. Zhang, “Reusability-based semantics for grasp evaluation in context of service robotics,” in *IEEE International Conference on Robotics and Biomimetics*, Dec. 2006, pp. 703–708.

[24] J. Kuehnle, Z. Xue, T. Grundmann, A. Verl, S. Ruehl, R. Eidenberger, J. M. Zoellner, R. D. Zoellner, and R. Dillmann, “6d object localization and obstacle detection for collision-free manipulation with a mobile service robot,” in *14th International Conference on Advanced Robotics (ICAR)*, 22-26 Jun. 2009.

[25] M. Saha, G. Sanchez, and J. Latombe, “Planning multi-goal tours for robot arms,” in *IEEE International Conference on Robotics and Automation*, 2003.