Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

A Grasp Planner Based On Inertial Properties

Efrain Lopez-Damian[†], Daniel Sidobre[†‡], and Rachid Alami[†]
edamian@laas.fr, daniel@laas.fr, rachid@laas.fr

[†] *LAAS-CNRS*
7, avenue du Colonel Roche, 31077 Toulouse, France

[‡] *Université Paul Sabatier*
118, route de Narbonne, 31062 Toulouse, France

***Abstract*— The future personal robot must be capable to work and take decisions in a dynamic human environment. One important capability of such a robot is manipulation. Grasp, as the beginning of any manipulation task is a key point.**

In this paper we present a grasp planner for manipulating real objects modeled by polyhedra. The algorithm is based on a random generation of grasp oriented by mass and inertial properties of the object. Recent techniques to filter and evaluate the quality of grasp are discussed.

***Index Terms*— Grasp Planning, Force Closure, Manipulation.**

I. INTRODUCTION

Service or personal robots are one of the principal objectives in robotics today. Such robots systems must have the capabilities to make decisions autonomously depending on the circumstances imposed by a dynamic human environment and the constraints of the tasks to be accomplished. To reach this objective it is necessary to develop a series of functionalities that must be integrated in the machine. Among all the functions, the manipulation of objects has a decisive role to allow the robot to interact and modify its environment.

Grasping an arbitrary object is one of the simple tasks that a robot must be able to accomplish. All manipulation tasks begin by a grasp: pick and place, filling a glass, bringing back an object and giving it to a human, using a tool, turning a crank handle... Polyhedral models of objects and environment can be now easily obtained [1] and geometric algorithms for planning path are efficient [17]. The important point to solve is the automatic grasp of objects. In this paper we discuss a general method to compute a good grasp position given an object and a gripper, and we present simulation results for some objects computed by our grasp planner.

The quality of a grasp is difficult to define because the forces that the grasp has to resist are mainly unknown. The force closure concept is a necessary condition for the grasp planning in order to indicate that a grasp can resist to any perturbation force although some grip tool does not define force closure when the load is known. The comparison of two grasps must take into account not only the possibility to resist perturbation with the minimal contact force but also how easy it is to perform the grasp while avoiding collisions and bad positioning of contact points. More general criterion influence the ability to accomplish the task like the accessibility to the active zone of a tool or the avoidance of collision for placing.

As the computation of all grip positions and optimization algorithms are computationally expensive we propose a random generation of grasps with a heuristic based on inertial properties of the object which is supposed to be of uniform density.

Our grasp planner algorithm is implemented and integrated in the Move3d platform [17], a generic environment for robot motion planning.

Section II discusses related work on manipulation and introduces the grasp planning algorithm presented in section III. Section IV presents the force closure and collision filters. Section V describes the determination of the quality of grasps. Some simulation results are presented in section VI and limitations and future work are presented in conclusion.

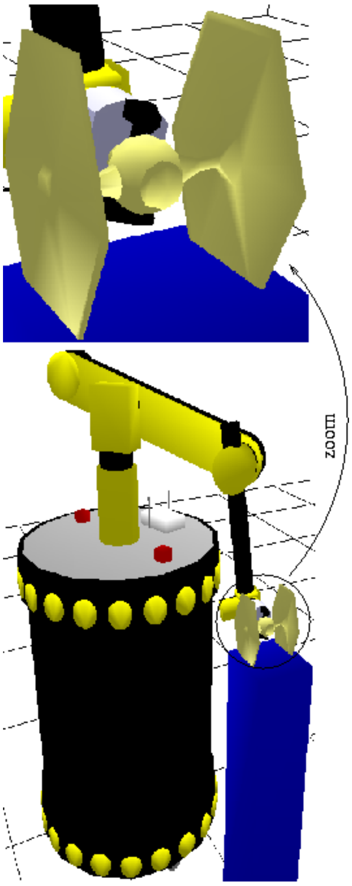

Fig. 1. Grasp for a Spaceship Model in Move3D

0-7803-8914-X/05/$20.00 ©2005 IEEE.

II. RELATED WORK

In the field of service robotics we can find manipulation projects developed by researchers of different universities or laboratories. The Robutler [8] is an experimental service robot with an anthropomorphic hand integrated with a stereo vision system for the recognition of a-priori known objects. The system has a modular grasp planner based on a grasp random generator [2].

In [19], a vision system approximates the shape of the object by a box primitive and the grasp planner generates grasps for this boxes. The approximation of the object by a box is used by Petersson [14], he uses a database containing the object label, size, grasping configuration and necessary way points. The robot is tested in a fetch and carry operation.

In the literature on grasp planning, we can identify two general approaches. The first one consists in finding the optimal grasp that satisfies the force closure property [5], [18], [16]; such methods work fine but are complicated to implement and have a no negligible time of processing.

The second approach assumes that we can generate a sufficient number of candidate grasps and choose the best among them. Reference [3] shows that it is not necessary to generate optimal grasps. An average quality grasp is an acceptable good grasp. The way to quickly find a grasp on the object is to use heuristics. Borst [2], [7] uses a random generation strategy. From one arbitrary point and frame inside the object, they launch a first ray in direction of the frame's X-axis. The same for the others rays but rotated about the Z-axis of 120 degrees. Toth [20], tries to find a Y-shape grasp star. The penetration points on the object surface give the contact points.

As our final intention is the integration of the grasp planner as a module of a personal robot, we must be capable to plan grasps on-line. Consequently, we have chosen to solve the problem using the second approach which has a more acceptable processing time. Instead of generating arbitrary points in any direction and orientation, we propose a different strategy for this purpose.

The object is mainly subject to external force of gravity and acceleration that act on the center of mass. It is a good idea to grasp the object around this point. The forces and torques that we have to apply to the object for keeping it stable will be lower if our grasp is closer to the center of mass. This will allow us to obtain a set of grasps with a good quality and it would not be necessary to produce a big number of grasps. In the same idea, good directions candidates for grasps are given by the principal inertial directions of an object.

III. GRASP PLANNING

1) Grasp Planner Algorithm: Here we introduce the basic algorithm for a random generator of grasps for polyhedral objects. We define a grasp as the contact points on the object surface and the frame associated with them that we have called the grasp frame.

Basic Grasp Planner Algorithm

- ⋆ Random Generation
- ⋆ Filter
- ⋆ Quality

A) Random Generation Step: Generation of contact points.

B) Filter Step: As quality determination is computationally expensive, we introduce the filter step to reject as soon as possible unfeasible grasps. Some constraints are imposed by the system itself, the grasp must be kinematically reachable by the robot and free of collision with the possible obstacles in the environment. To guarantee that the object is firmly held with no slippage, we use a force closure test. The collision and force closure filters are explained in the next sections.

C) Quality Measure Step: Several grasps can be produced after the first two steps are executed. The final step is the assignment of a quality measure to the grasps. Various measures have been proposed based in wrench space. A more detailed explanation of these measures will be presented in section V.

In the sequel of the section, we present the generation of grasps. We first recall how the axes of inertia and the center of mass can be computed.

2) Axes of Inertia: The calculation of the axes of inertia and center of mass is taken from a 3D model of the object based on stereo vision or laser range sensors. These parameters depend only on object geometry with a constant density ρ. The object model is composed of facets and vertices. The location of the center of mass and the inertia tensor can be computed by the conversion of the integrals of mass into the volume integrals. We suppose the polyhedron (P) has a mass m and a uniform density ρ, we can relate the volume as $m = \rho V$, we compute

$$V = \int_P dV \tag{1}$$

The volume integrals can be reduced into surface integrals by the divergence theorem.

$$\int_V \nabla \cdot F dV = \int_S F \cdot N dS \tag{2}$$

for any vector field F defined on V. Where V is the region bounded by the surface S (union of triangular faces) and N is the vector of the exterior unit normal of V along its boundary. We use the algorithm developed by Mirtich [12]. The complexity of the algorithm is linear depending in the number of vertices and faces of the object.

The inertia tensor T is composed by the moments and products of inertia about the center of mass CM.

$$\mathbf{T} = \begin{bmatrix} Ixx & -Ixy & -Ixz \\ -Iyx & Iyy & -Iyz \\ -Izx & -Izy & Izz \end{bmatrix} \mathbf{CM} = \begin{bmatrix} Cx \\ Cy \\ Cz \end{bmatrix} \tag{3}$$

The principal axes of inertia are the principal directions of T.

3) Grasp Generation Algorithm: Following the basic algorithm, the main goal of our grasp planner is to produce a set of contact points on the object surface and the grasp frame. We have decomposed the whole algorithm in two sub-algorithms for better understanding.

Sub-algorithm1: Grasp Generation

Input: Geometric Model of the object and environment. Geometric Model and Kinematics of robot and gripper.
Output: Grasps

Step1: Compute the center of mass and the principal axes of inertia (*AOI*). The initial grasp frame is given by the axes of inertia and the center of mass.

Step2: For each axis of inertia compute several orientations of the grasp frame by an angle θ about the X and Y axes of the frame. This allows us to find grasps in case the gripper is in collision with the object. Compute several grasp frame positions along the axes of inertia when it is not possible to obtain a grasp about the center of mass because an obstacle is placed near the object.

Step3: Compute contact points (see sub-algorithm2) for each position and orientation of the grasp frame defined in step 2.

In sub-algorithm2 we present the generation of contact points. The generator takes into account the structure of the gripper. Our gripper is composed of two-fingers with three-contacts (see Fig. 2). The current stage has to be implemented manually for each gripper. General solutions can be proposed like generate a closing path for the fingers using Move3D. The fingers close until a collision with object is detected and the contact points are found. Unfortunately this kind of solutions are computationally too expensive to be used extensively.

sub-algorithm2: Contact Points

Input: Grasp Frame
Output: Contact points

Step1: A grasp plane *Gp* is formed with the X-axis and Y-axis of the grasp frame, Z-axis is one of the principal axis of inertia (see Fig. 2).

Step2: To find the first point (P_1) of contact, we draw up a ray in the direction of the X-axis of the grasp plane. We find the intersection point (P_1) between the ray and the object. As the fingers have a spherical fingertips, we define P_1' as the translation of P_1 in the direction of the object surface normal by the radius (Rf) of the fingertip.

Step3: For the second point of contact (P_2), we find the intersection between the plane Gp and planes formed by the facets of the object translated by a distance Rf in the facet normal direction. The line that results of the intersection of planes must intersects as well a circle with center at P_1' of radius (*R*), where *R* is equal to the distance between the two-contact finger of the gripper. The intersection point is P_2'. P_2 is the contact point of the object that satisfies

$$\overrightarrow{P_1P_2} \cdot Y_{gp} > 0 \tag{4}$$

Step4: Relocation of the grasp frame according to P_1 and P_2

$$Y_{gp} = \frac{P_1' - P_2'}{\|P_1' - P_2'\|} \tag{5}$$

$$X_{gp} = Y_{gp} \times Z_{gp} \tag{6}$$

Step5: We emit a second ray with the same direction of axis X_{gp} of grasp plane from the middle point of $P_1'P_2'$. P_3' is the intersection between the last ray and the object facets translated by Rf. P_3 is the contact point associated.

Step6: The new origin of the grasp frame is the middle point of the line formed between P_3' and the middle point of $P_1'P_2'$. This point will be near of the initial point.

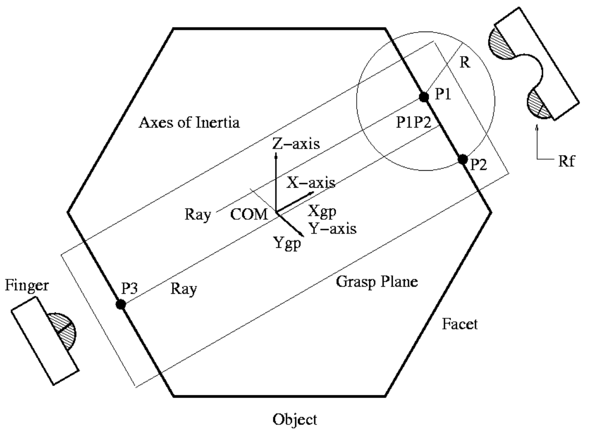

Fig. 2. Grasp Generation from Axes of Inertia and Center of Mass

IV. GRASP FILTERS

A. Force Closure Property

One of the most important properties for a grasp is the notion of force closure. A grasp is force closure if it can balance any external forces exerted on the object. Several works have been made in the analysis and synthesis of force closure. Nguyen [13] proposes an algorithm for constructing 2D force closure grasps based on the geometry of the object, Ponce [16] computes grasps of polygonal objects using a projection algorithm based on linear programming.

Computing 3D force closure grasps has been treated in [18], they find that the four finger grasps fall in three categories and developed new necessary and sufficient conditions for it. However the algorithm that we implement is based on the work done by Li [10], due to its simplicity and its small processing time. He presents new conditions for computing the 3D force closure grasps in a geometric way for a robot hand with three hard fingers and contact point with friction.

1) Necessary notions:

we briefly recall here some notions.

Wrench: the combination of the force *f* and the torque τ form the wrench $w_i = [fi \;\; \tau_i]^T$

Equilibrium: a set of n wrenches achieves equilibrium when the Convex Hull of the points $w_1, ..., w_n$ contains the origin.

Coulomb friction model: how much force a contact can apply in the tangent directions to a surface as a function of the applied normal force. This implies that $|f^t| \leq \mu f^n$, where $\mu \geq 0$ is the static coefficient of friction.

Hard-finger model: thanks to friction between the fingertip and the object, forces can be exerted in any direction within the friction cone.

The algorithm is divided in two sub-algorithms: the first one computes force closure grasps in 2D, this sub-algorithm is used by the second one to finally compute the 3D force closure grasps.

2) 2D Grasps Force Closure Algorithm:

The algorithm considers a hard finger model, contact points (C_1, C_2 and C_3) and the normals at these contact points (pointing inside of the object). The friction cones are bounded in pairs by the unit vectors $n_{i,1}$ and $n_{i,2}$. External forces and torques act in a point of the object, the center of mass frequently.

Proposition: Three non parallel contact forces in the friction cones different to zero achieves equilibrium if they positively span the plane and its lines of action intersect at some point.

Li [10], propose the substitution of the unknown forces in the proposition by the boundary vectors of the friction cones and state a new proposition to compute equilibrium grasps.

The algorithm starts with the elimination of the regions in the friction cone that do not contribute to equilibrium, this is called disposition H. The new proposition states that the three finger grasp is equilibrium if the intersection of the three friction cones is not empty after the operation of disposition H.

Next step is the calculation of the points of intersection by the two boundary lines of each friction cone. The grasp will be force closure if at minimum there is one point due to the intersection of two different boundary lines of friction cones that is inside of the third friction cone. The complexity of the algorithm is minimum, only a few operations are needed [10].

3) 3D Grasps Force Closure Algorithm:

A 3D three finger grasps are force closure if two conditions are fulfilled:

1) There exists a contact plane *Sp* and contact unit vectors $n_{i,1}$ and $n_{i,2}$. Three contact points define a plane *Sp* if they don't lie on the same line. The intersection of the friction cones with the plane *Sp* can be in three ways: at a point, on a line or on the plane. In the last case as the apex of the friction cones lies on the contact plane, the intersection gives two lines defined by a pair of contact unit vectors $n_{i,1}$ and $n_{i,2}$.
2) The contact unit vectors form a 2D force closure grasp in the contact plane.

B. Collision Checking

The second type of filter that we use is the collision checker. We verify that there is no intersection between the geometric models of the robot, gripper, object and the obstacles. We use the collision checker implemented in the motion planning tool Move3D [17]. When a grasp is found, we place the gripper by calculating the inverse kinematics of the arm and we test that the execution is free of collision of any kind. Taking into account the whole robot and the environment, we avoid the risk of collision between arm-object, arm-obstacles, gripper-obstacles (see Fig.1), and reduce the need of backtrack at high level task planning.

V. MEASURE OF QUALITY

A. Quality based on Wrench Space

The planning of a good grasp is important when the robot has to take an object firmly, for this a quality criterion has been developed in [6]. The criterion tries to quantify the notion of a good grasp for a force closure grasp.

A hard finger contact model and Coulomb friction are assumed between the object and the fingertips. The forces applied by the finger f_i must remains in the friction cone to avoid the slippage. We must discretize the cone of friction to represent it by a finite set of *m* vectors.

For the quality measure, Ferrari [6] consider that the sum of the magnitude of the forces applied by the gripper at the *n* contact point is 1, then f_i can be written as:

$$f_i = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{i,j} f_{i,j} \tag{7}$$

with $\alpha_{i,j} \geq 0$. Similarly, the total wrench applied on the object is expressed by:

$$w = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{i,j} w_{i,j} \tag{8}$$

and the set of all wrenches is:

$$W = CHULL(\bigcup_{i=1}^{n} w_{i,1}, ..., w_{i,m}) \tag{9}$$

The quality measure is the distance of the nearest facet of the Convex Hull from the origin.

In reference [4], Borst proposes that instead of using the wrench space, an alternative approach is the definition of the Task Wrench Space (TWS) to calculate the quality of a grasp. The TWS are the wrenches expected to occur for a given task. The approach adopted is the combination of the approximation of the TWS by a task ellipsoid and the incorporation of the object geometry. The latter one is introduced in [15].

B. Stable Grasps

It may happen that grasps may be generated near the borders of the object. Such grasps are not desirable because they can be unstable at the moment when the gripper grasps the object.

We define a stable grasp as the grasp where the contact points are away from the borders of the object. In the case where the facets that describe the object are big enough, a grasp will be more stable if the contact points are in the middle of the facets.

VI. RESULTS

We have tested the algorithm with the geometric model of several objects. Several results are shown in the series of figures. The model of horse (Fig. 4) was taken from the Large Geometric Models Archive of Georgia Institute of Technology. The couch and doll models (Fig. 5) come from the imagery of Ohio State University and finally the spaceship model (Fig 1), was obtained from the 3D cafe web page.

In Fig. 3 and Fig. 4, we show the solutions for a cube and a horse model. We can see the grasps generated and executed by our grasp algorithm. In the cases when there are no obstacles in the manipulation space, the grasps are near the center of mass of the objects. When obstacles are placed near the objects we observe how the planner is capable to find other good grasps. The last two models in Fig. 5 are models of real objects acquired by a range scanner.

The experiments were performed using a 500 MHz Solaris SunBlade. The time that the algorithm requires to generate the final grasp is determined by the complexity of the object model. In Table 1 we can see the number of force closure grasps generated by the algorithm for each object, the quality of the best grasp and the total processing time of the whole process.

TABLE I

RESULTS FOR SEVERAL OBJECTS

Object	Total Time	Grasps	Quality	Facets
Horse	7.41 s	17	0.66	600
Couch	10.77 s	26	0.512	1000
Spaceship	8.67 s	21	0.87	617
Doll	5.45 s	20	0.44	500
Cube	2.45 s	30	0.13	12

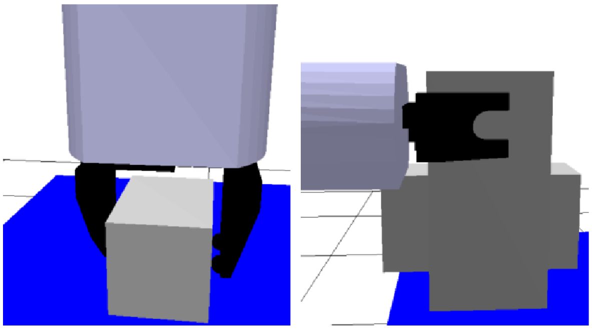

Fig. 3. Cube Model. For the same model, two different situations gives very different grasps.

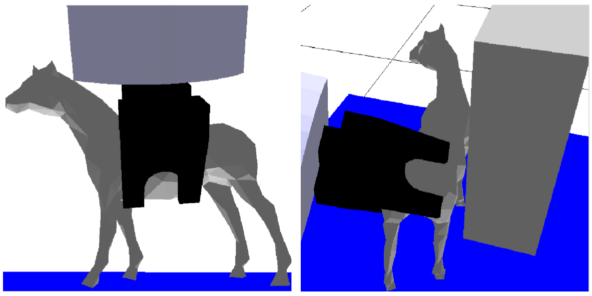

Fig. 4. Horse Model with and without obstacles. In first figure we see a grasp near the center of mass. Due to the obstacle the planner finds another different grasp avoiding collision.

One can see that it is not necessary to find a great number of grasps to find one with a good quality.

VII. CONCLUSION AND FUTURE WORK

In this paper we describe a grasp planner for 3D objects using a geometric model. In the case of objects with non uniform density, the grasp chosen may be inadequate. By measuring the position of the center of mass at the moment of the grasp, a new grasp will be computed.

We can see that all the grasp planners tackle the problem of generating a set of contact points assuming that the object is small enough for the robot to grasp it. One make a partition of the object into smaller pieces using an approximate convex decomposition process [11]. Convex decomposition of three-dimensional polyhedra is done by iteratively removing the non-convexity of the polyhedron until all components are convex. We can launch the grasp planner for each convex piece and try to grasp that part of the object.

Other extensions that we will implement in our grasp planner is take into account a series of constraints imposed by the task that the robot has to accomplish. These constraints will be reflected in the grasp selection. In this case the grasp with the best quality measure could not be the adequate one for a specific task. For example Jones and Lozano-Perez [9] take into account constraints imposed by the classical pick and place problem to plan grasps.

A third quality criterion can be implemented by taking into consideration the distance between the gripper and the environment. We can assign a lower quality value for the grasps that are located near to the obstacles around the object.

We are now working on the implementation of the extensions and the integration of the grasp planner module in a new physical robotic platform composed of a mobile Neobotix platform and a Mitsubishi PA10 arm. Making the extensions mentioned above we think that the grasp planner will be more general and robust for treating a variety of manipulation tasks for a service or personal robot.

ACKNOWLEDGMENT

The work described in this paper was partially conducted within the EU Integrated Project COGNIRON (”The Cognitive Companion”) and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020. The authors acknowledge the support of the National Council of Science and Technology of Mexico (CONACyT).

REFERENCES

[1] J.M. Badcock and R.A. Jarvis. Wire-frame modelling of polyhedral objects from rangefinder data. *Robotica*, 12:65–75, 1994.

[2] Ch. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3d objects. In *In Proc. IEEE conf. on Robotics and Automation*, pages 1890–1896, Detroit, Michigan, May 1999.

[3] Ch. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. In *In Proc. IEEE/RSJ conf. on Intelligent Robots and Systems*, pages 3692–3697, Las Vegas, Nevada, October 2003.

[4] Ch. Borst, M. Fischer, and G. Hirzinger. Grasp planning: How to choose a suitable task wrench space. In *In Proc. IEEE conf. on Robotics and Automation*, pages 319–325, New Orleans, 2004.

[5] D. Ding, Y.H.Liu, and S. Wang. Computing 3-d optimal form-closure grasps. In *In Proc. IEEE conf. on Robotics and Automation*, pages 3573–3578, San Francisco, April 2000.

[6] C. Ferrari and J. Canny. Planning optimal grasps. In *In Proc. IEEE conf. on Robotics and Automation*, pages 2290–2295, Nice, France, May 1991.

[7] M. Fischer and G. Hirzinger. Fast planning of precision grasps for 3d objects. In *In Proc. IEEE/RSJ conf. on Intelligent Robots and Systems*, pages 120–126, Grenoble, 1997.

[8] U. Hillenbrand, B. Brunner, Ch. Borst, and G. Hirzinger. The robutler: a vision-controlle hand-arm system for manipulating bottles and glasses. In *In Proc.35th Int. Symposium on Robotics*, Washington D.C., 2004.

[9] J. Jones and T. Lozano-Perez. Planning two-fingered grasps for pick-and-place operations on polyhedra. In *In Proc. IEEE conf. on Robotics and Automation*, pages 683–688, Cincinnati, Ohio, May 1990.

[10] J.W. Li, H. Liu, and H.G. Cai. On computing three-finger force-closure grasps of 2-d and 3-d objects. *In IEEE Transactions on Robotics and Automation*, 19(1):155–161, February 2003.

[11] JM. Lien and N. Amato. Approximate convex decomposition. Technical Report TR03-001, PARASOL LAB, Texas A&M University, January 2003.

[12] B. Mirtich. Fast and accurate computation of polyhedral mass properties. *Journal of Graphics Tools*, 1(2):31–50, 1996.

[13] V. Nguyen. Constructing force-closure grasps. *International Journal Of Robotics Research*, 7(3):3–16, 1988.

[14] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic, and H.I. Christensen. Systems integration for real-world manipulation tasks. In *In Proc. IEEE conf. on Robotics and Automation*, pages 2500–2505, Washington D.C., 2002.

[15] N. Pollard. *Parallel Methods for Synhesizing Whole-Hand Grasps from Generalized Prototypes*. PhD thesis, Massachusetts Institute of Technology, 1994.

[16] J. Ponce and B. Faverjon. On computing three-finger force-closure grasps of polygonal objects. *In IEEE Transactions on Robotics and Automation*, 11(6):868–881, December 1995.

[17] T. Siméon, J-P. Laumond, and F. Lamiraux. Move3d: A generic platform for motion planning. In *Proc. 4th International Symposium on Assembly and Task Planning (ISATP'2001)*, 2001.

[18] A. Sudsang and Jean Ponce. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *In Proc. IEEE conf. on Robotics and Automation*, volume 1, pages 1355–1360, Nagoya, Japan, 1995.

[19] G. Taylor and L. Kleenman. Grasping unknown objects with a humanoid robot. In *In Proc. Australian Conference on Robotics and Automation*, pages 191–196, Aucland, November 2002.

[20] E. Toth. Stable object grasping with dextrous hand in three-dimension. *Periodica Polythechnica SER. EL. ENG.*, 43(3):207–214, 1999.

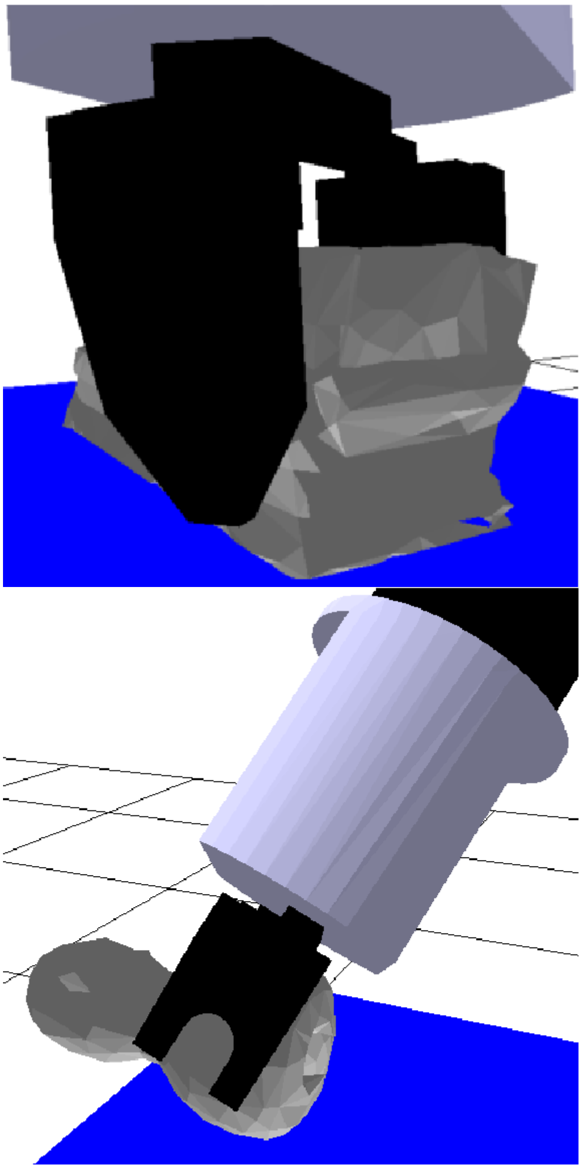

Fig. 5. Grasp planner tested in Real Couch and Doll Models