Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

A Fast and Robust Grasp Planner for Arbitrary 3D Objects

Ch. Borst, M. Fischer and G. Hirzinger

German Aerospace Research Establishment (DLR)
Institute of Robotics and System Dynamics
PO Box 1116, D-82230 Wessling, Germany
Christoph.Borst@dlr.de, Max.Fischer@dlr.de

Abstract

In the near future, more and more robots will be used for servicing tasks, tasks in hazardous environments or space applications. Dextrous hands are a powerful and flexible tool to interact with real world environments that are not specially tailored for robots. In order to grasp and manipulate real world objects, grasp planning systems are required. To be integrated in online planning systems for robots, they have to be very fast. This paper presents a method to compute a desirable grasp quality measure very fast and accurate – both aims haven't been reached simultaneously until now. Based on this measure an heuristic approach towards fast planning of precision grasps for arbitrarily shaped 3D objects is described. A number of feasible grasp candidates are generated heuristically. These grasp candidates are qualified using the described grasp quality measure and the best candidate is chosen. The planned grasps are robust in respect of grasp placement. It is shown that only a relatively small number of grasp candidates has to be generated in order to obtain a good – although not optimal – grasp.

1 Introduction

The theory of grasping is already well understood. Several measures to qualify grasps have been proposed: Grasp dexterity measures (ability to manipulate an object), grasp stability measures (ability to resist external forces and torques) or robustness (in respect to grasp point placement). A good overview can be found in [7, 9].

Efficient methods to construct force-closure grasps for polyhedral objects are proposed in [6, 10]. In [8] a method to generate power grasps is proposed, showing the complexity of grasp planning and the necessity of heuristics to improve planning time.

Shimoga [9] stated, that "none of the synthesis algorithms have been implemented in real time so far...The probable reasons for this situation appear to be (1) prohibitive computational complexities of the synthesis algorithms ...".

The purpose of this work is to generate precision (or force-closure) grasps for 3D objects of which a (at least partial) geometric model exists. The latter may be obtained by a CAD system or through sensors, e.g. video or laser cameras. The objects to be grasped can be arbitrarily shaped. In order to integrate a grasp planner in a task-level programming system, planning times should be in the order of a few seconds. Our approach is guided by the assumption that there exist many grasps fulfilling some given quality requirements quite well. So we can use some heuristics to find a good – although not optimal – grasp in a very short amount of time. It extends our work in [4] in those aspects:

Robust contact placement. An approach towards grasp point placement not close to edges is described.

Four finger grasps can be planned now.

Fast and accurate grasp quality determination. A major break-through in this field has been achieved: While in [4] we had to admit approximation errors of up to 30 % (!) and only were able to underestimate a desirable grasp quality measure by an easier computable one, we can determine the former now with high accuracy.

2 The DLR Hand

At our institute a dextrous hand has been developed (fig. 1) [2]. The four fingers have 3 DoF each. Due to the high degree of mechatronic integration –

12 actuators and more than 100 sensors are integrated in the hand itself – only a small box housing the joint controllers is needed externally. Therefore the hand can easily be mounted on a robot which makes it a flexible and powerful manipulation tool for service robots. With this hand several successful telemanipulation demonstrations were performed.

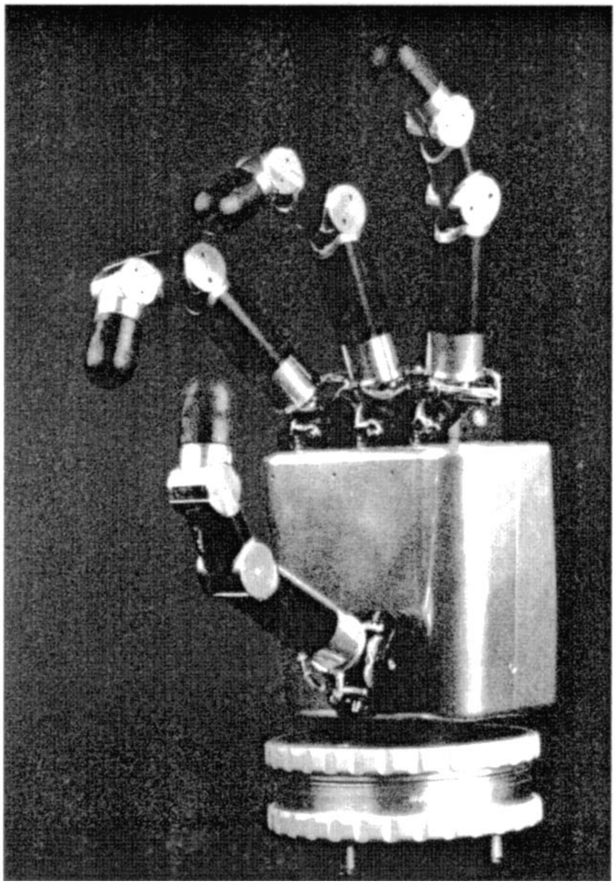

Figure 1: The four finger DLR Hand (12 DoF)

3 Grasp Quality Measure

The grasp quality measure we use in this work is a 'classical' static grasp stability measure [3], which is also used in [8, 5]. It measures to what extent a grasp can resist external wrenches that are exerted on the object to be grasped without fingers starting to slide at their contact points. Additionally, in our approach the robustness of a grasp in respect to grasp point placement is also taken into consideration (see ch. 7).

The method proposed in [3] realizes such a stability measure by determining the set of external wrenches (*grasp wrench space*, **GWS**) that can be resisted by a grasp if either

Q1 one unit force is distributed over all grasp points or

Q2 at each of the grasp points not more than a unit force is applied.

and all forces applied lie within the respective friction cone. Examining the **GWS**, the stability measure describes the weakest point of a grasp in general (d_1 in fig. 2) or in respect to a *task wrench space (***TWS***)*, a given set of wrenches that a grasp should resist ($\frac{d'_2}{d_2}$ in fig. 2).

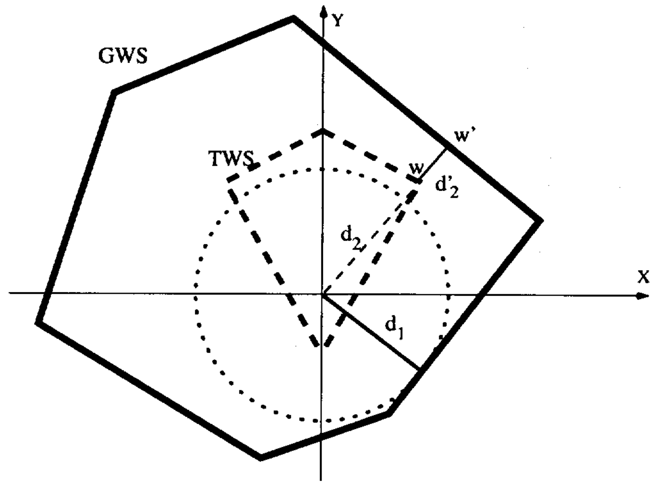

Figure 2: Quality measure with (dashed line) and without a given task wrench space (dotted)

In [3] is shown that the quality criterion **Q1** can be expressed by a L_1 ($\|\mathbf{g}\| = (g_1 + \ldots + g_n)$) and **Q2** by a L_∞ metric ($\|\mathbf{g}\| = \max(g_1, \cdots, g_n)$), respectively, of the *generalized force vector* $g = (\ f_1^T, \ldots, f_n^T\ )^T$. The forces f_i act at the contact points r_i of a given grasp and $g_i = |f_i|$.

With $w_i(f) = \binom{f}{\lambda \cdot (r_i \times f)}$ (see ch. 8 for choosing λ), let $\hat{w}(g) = \sum_{i=1}^{n} w_i(f_i)$ be the wrench that results from a given generalized force vector and a given set of n contact points. Then the *grasp wrench space* corresponding to one of the above criteria is

$$\mathbf{GWS_L} = \{\hat{w}(g) \mid \|g\|_L \leq 1 \wedge \forall i : f_i \text{ is in friction cone } i\}$$

The set of forces within the friction cones at contact point i can be approximated by a linear combination of a finite set of n unit force vectors $f_{i,j}$ at the friction cone boundaries

$$f_i = \sum_{j=1}^{n} \alpha_{i,j} \cdot f_{i,j}, \qquad \alpha_{i,j} \geq 0, \qquad \sum_{i=1}^{n}\sum_{j=1}^{k} \alpha_{i,j} \leq 1$$

Thus the resulting approximation for the set of wrenches resulting from friction cone forces applied at contact i can be written as

$$w_i = \sum_{j=1}^{n} \alpha_{i,j} \cdot w_{i,j}, \quad w_{i,j} = \begin{pmatrix} f_{i,j} \\ \lambda \cdot (r_i \times f_{i,j}) \end{pmatrix}$$

Following [3], the *grasp wrench spaces* according to the criteria **Q1** and **Q2** using approximated friction cones can be calculated geometrically ($\bigoplus$: Minkowski sum):

$$GWS^0_{L_1} = ConvexHull\ (\bigcup_{i=1}^{n} \{w_{i,1}, \ldots, w_{i,m}\})$$

$$GWS^0_{L_\infty} = ConvexHull\ (\bigoplus_{i=1}^{n} \{w_{i,1}, \ldots, w_{i,m}\})$$

4 Computational Aspects

In theory, the measure family mentioned above is a very good and simple solution for grasp planning. Unfortunately, in order to realize a practical grasp planner, things get a little bit unfriendlier. We had the aim of generating and testing about 100 grasp candidates in less than 10 s. This means that for one grasp candidate there are only about 100 ms left. This puts hard constraints especially on the grasp measure calculation. There were two main problems we had to focus our interest on:

Problem 1: Friction Cone Approximation. Approximating the friction cone with n force vectors leads into a conflict: On the one hand, a rough approximation leads to bigger errors. On the other hand, more force vectors for a friction cone makes the convex hull calculation costs explode. In [4] we used 4 vectors, which yields a theoretical worst case error of underestimating the real friction by factor $k = \frac{1}{\sqrt{2}}$, an error of almost 30 %. Rotating the friction cone approximation pyramids about their center axes we sampled the space of all possible pyramid rotation combinations and thus determined the effect of the approximation error. The variation of the resulting quality is shown in fig 4. One solution would be to increase the number of approximation vectors. A number of $n = 8$ would reduce the error to bearable 8 %. However, the costs for calculating the convex hull rise by a factor of aprox. 3.6, which is also very undesirable.

Problem 2: Calculation of Q2 Measure. As mentioned above, the more desirable grasp measure is the Q2 measure, as it takes more the finger force limitations than the cumulative force costs into consideration. Unfortunately – like sometimes in life – the more desirable things are harder to get: The calculation of the Minkowski sum for 4 fingers and 4 friction cone approximation vectors means the calculation of a convex hull for 256 6D vectors, for 8 friction cone approximation vectors, we have 4096 vectors. Both cases move us far away from online applications.

The Solution: Incremental Convex Hull Calculation. Regarding above considerations it became clear that in order to calculate the convex hull using a given number of friction cone vectors can't satisfy accuracy and time requirements at the same time. A way out of this conflict is an incremental construction of the convex hull: We construct the convex hull with a rough approximation of the friction cones with only three vectors (e.g. $w2\_0, w2\_1, w2\_2$ in fig. 3) and retrieve the face representing the quality measure, F_{weak} in fig. 3 (nearest to origin or smallest **GWS/TWS** ratio, see fig. 2). We then try to compensate for the approximation error by incrementally adding additional wrenches to the wrench set spanning the convex hull: To all friction cones participating in spanning F_{weak} (friction cones $FC1$, $FC2$ in fig. 3, spanning wrenches $w1\_1, w2\_1, w2\_2$) we add the wrenches at which the largest approximation error in respect to F_{weak} occurs. These are the wrenches with the largest positive distances to F_{weak} (in fig. 3: $w1^{F_{weak}}_{max}$ with distance $d1$ and $w2^{F_{weak}}_{max}$ with $d2$, resp.). Those wrenches reduce the approximation error at the weakest face in the next iteration (i.e. calculation of the weakest face). Now we just have to calculate the $wi^{F_{weak}}_{max}$:

The original, not approximated grasp wrench spaces are spanned by circular one-dimensional manifolds $FCi_0(\alpha)$ in the 6D wrench space. We can represent them analytically ($FC2_0(\alpha)$ in fig. 3) as wrenches resulting from an initial force vector ($w2\_0$ in fig. 3) on a friction cone rotating by an angle $\alpha \in [0, 2\pi]$ about the friction cone center axis. $FCi_0(\alpha)$ either lie in parallel to F_{weak} or there exists a wrench $FCi_0(\alpha_{max})$ maximizing the positive distance to F_{weak}. With $dist^{FCi_0}_{F_{weak}}(\alpha)$ being the distance of the wrench $FCi_0(\alpha)$ from F_{weak} we get α_{max} straightforward by solving $dist'^{FCi_0}_{F_{weak}}(\alpha) = 0$ and choosing the solution with $dist''^{FCi_0}_{F_{weak}}(\alpha) < 0$. We continue adding new wrenches and recalculating the weakest face until the quality measure doesn't improve anymore or only marginally.

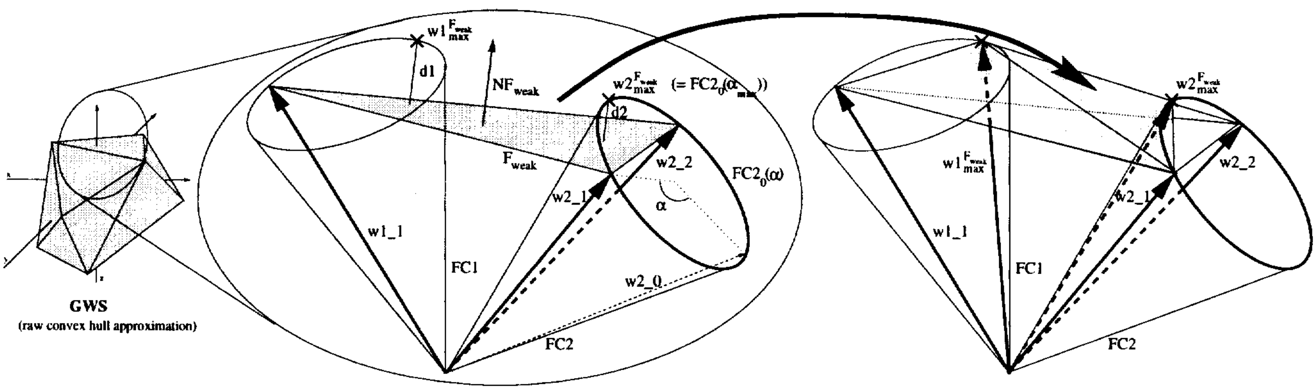

Figure 3: Improving friction cone approximation at the weakest convex hull face, 3D analogon of 6D case

More formally, we have the iteration

$$
\begin{aligned}
W_{L_1}^{i+1} &= ConvexHull\ (\ W_{L_1}^{i}\ \cup \\
&\cup\ \ \{w1_{max}^{F_{weak}^i}, \ldots, wn_{max}^{F_{weak}^i}\}\ )
\end{aligned}
$$

with F_{weak}^i being the weakest face of the **GWS** approximation W_L^i.

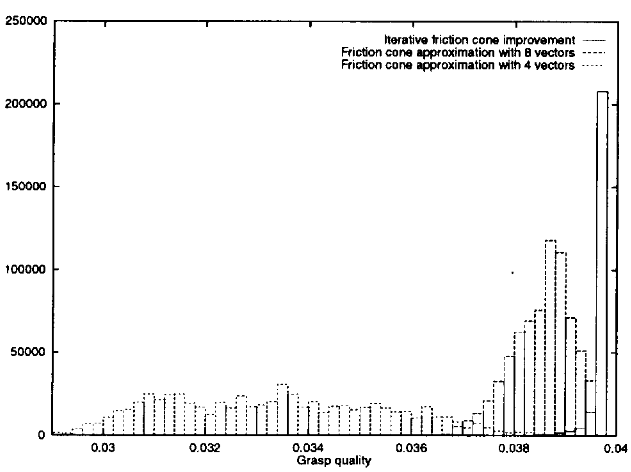

Figure 4: Approx. error, 4 and 8 friction cone vectors (scaled for clearer representation), new method (line)

To compare the new method with the former one, we calculated the measure for a 4 and 8 vector friction cone approximation and the new measure while rotating the friction cone approximations by small angles (resulting quality distribution for one particular grasp: fig. 4). We did this for different grasps and found that for the new measure (using 4 iterations) the probability for a residual error of more than 1 % turned out to be 4.5 %, in only 0.04 % of the samples we obtained an error of more than 5 %. To clarify the costs for convex hull calculation: We start with a convex hull of 4 (fingers) x 3 (Vectors per cone) = 12 wrenches and add in the case of 4 iterations $4 \times 4 = 16$ additional wrenches. We get computation times of about 80 ms on a SGI Indigo2 R10000, which we regard an excellent value compared with the very high accuracy we obtain!

But the **Q1** measure initially was not the one we were heading for: The more interesting one still is the **Q2** measure. As mentioned in section 4 it seemed to be clear for a long time that calculating the Q2 measure would not only yield a bad, but an inacceptable ratio of computation costs and accuracy. We seemed to have to put up with underestimating **Q2** by **Q1**, which yields a theoretical ratio from 1 : 2 up to 1 : n (n being the number of contacts)! But the good results for the Q1 measure gave us hope to be able to get around computational complexity. We also start with the (very rough) initial approximation $\mathbf{GWS}_{L_1}^0$ and use the following iteration:

$$
\begin{aligned}
W_{L_\infty}^{i+1} &= ConvexHull\ (\ W_{L_\infty}^{i}\ \cup \{w_{best}\}) \\
w_{best} &\in \bigoplus \{w1_{max}^{F_{weak}^i}), \ldots, wn_{max}^{F_{weak}^i}\}
\end{aligned}
$$

with w_{best} the wrench from the Minkowski sum of all $wi_{max}^{F_{weak}^i}$ maximizing the positive distance to F_{weak}^i. In only about 30 iterations ($\sim$ 140 ms) the **G2** measure doesn't improve anymore. So now there exists a method of calculating the desirable **Q2** measure, which seemed to be too complex to be calculated effectively, with high precision in a very short time by improving an approximation locally and incrementally!

5 Grasp Generation

The overall strategy to plan a grasp is to generate a given number of candidate grasps and chose the best among them. This heuristic is based on the assumption that there exist many sufficiently good grasps, which is shown to be true in section 8.

There are two constraints in order to generate candidate grasps: For a four finger grasp, contact placement on the vertices of a tetrahedron with the surface normals pointing from the tetrahedron center would be most stable. However, a given hand kinematics puts constraints on the grasp point reachability. To find a compromise between these requirements, we chose a point on the object surface as the first contact point (**1** in fig. 5). We calculate a ray direction deviating from the negative surface normal by an arbitrary angle (**2**). We use the friction cone angle as standard distribution for this angle. A penetration point (**3**) of this ray is chosen as the second grasp point. In case that there is more than one such point, we chose one at which the surface is penetrated from inside to outside. Next we calculate the center point of the two already existing grasp points (**4**) and emit two rays (**5a, 5b**) in directions which are arbitrarily chosen with a distribution between: (1) All three rays lie in a plane, 120° between **2** and **5a**, **5b**, respectively (optimal for gripper kinematics). (2) **2** and **5a**, **5b**, respectively, are perpendicular, 120° between **5a** and **5b** (optimal for tetrahedral grasp point distribution). The penetration points **6a, 6b** become contact points 3 and 4.

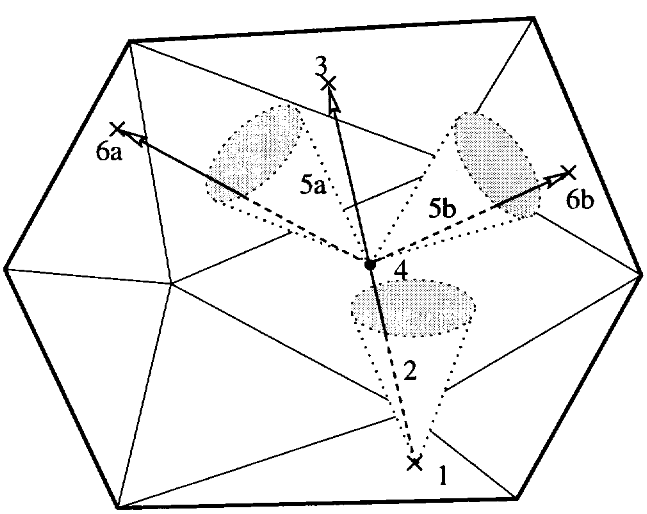

Figure 5: Generation of a grasp candidate

For non-convex objects we may get more than one penetration point penetrating the object surface from inside. In this case we select one of those penetration points randomly. If all three rays yield one penetration point (which may not be the case if the frame doesn't lie within the object), we accept this set of points as a grasp candidate.

6 Filter Hierarchy for Generated Grasps

Determining the grasp quality of a generated grasp (or grasp candidate) is quite time consuming. Thus we first perform some simpler (and faster) tests for a grasp candidate in order to exclude grasp candidates which cannot lead to feasible grasps as early as possible. In our current implementation [1] we use the following filter hierarchy:

Grasp Point Distance. If the distance between two grasp points of a grasp are further apart than the corresponding dextrous hand can spread it's fingers, this grasp can't be executed and thus it is discarded at this early stage.

Grasp Point Close to a Supporting Surface. In most cases the object to be grasped is supported by a (piece of a) surface. If a grasp point is situated on or close to this surface, a collision free grasp may not be found. It may be better to generate a grasp with grasp points not too close to the supporting surface. Thus grasp candidates with grasp points close to the supporting surface are discarded.

Colliding Grasps In order to execute the planned grasp the dextrous hand mustn't collide with the object to be grasped and the environment. We determine a gripper pose which fulfils the requirement that the distances from a finger base to the corresponding grasp point should approximately be a given constant **D**.

With the hand frame matching a grasp frame and the finger tips on the respective grasp points a collision check between the hand and the environment (including the object to be grasped) is performed. Colliding grasps are discarded.

Of course this is a very crude heuristic, as one might use the redundance in the transformation between object frame and hand frame to avoid collisions. However, a modified path planning strategy which is able to escape from a colliding configuration to a non colliding one will need a lot of time. It seems to be more efficient to generate more grasp candidates and check them using the heuristics mentioned above. Of course we have to face the drawback that there are situations in which no collision free grasp can be found using this simple heuristic. In this case our planner fails. Therefore we will experiment with path planning strategies to escape from colliding configurations in the future.

7 Robust Contact Point Placement

Sometimes, the planner yields grasps with contact points close to object edges. In reality, grasps of this kind are low quality grasps. Uncertainties in the object model or in contact placement during grasp execution as well as the wish to manipulate a grasped object make those grasps fail easily. Therefore we developed a method to measure the robustness of contact placement and move the contact to a more robust – i.e. smooth – region.

We estimate the radius **r** of a circle through given grasp uncertainties, contact circle and finger tip rolling movements resulting from manipulations. We then measure the distance from several equidistant points **P1...Pn** from a circle with radius **r** around the surface normal at contact point **CP** with its center point moved away from the surface by a distance **h**. Starting from a certain distance from the surface we can measure also concave regions by simply emitting rays in one direction (fig. 6). We then take the maximum of the penetration point distances (**d1..d3**x in fig. 6) to the tangent plane in **CP** as a measure of the smoothness. The direction from the point where the maximum distance was measured to the center point can be used to move the contact point in order to try to improve it. We then move the grasp point **CP** about the distance **r** in this direction and accept the new contact point if its robustness increased.

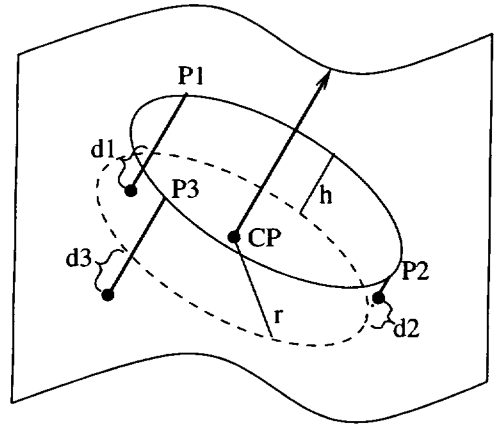

Figure 6: Measuring the smoothness of a grasp point's environment

8 Results

To validate our approach we experimented with geometric models of different objects. The results for different objects (e.g a banana, fig. 8, 7) are shown below. The most time costly steps in the algorithm above are the generation of a grasp candidate ($\sim$ 0.7 ms), collision check ($\sim$ 1 ms) and grasp measure calculation ($\sim$ 100 ms).

To determine the optimal number of grasp candidates to be generated in order to obtain a good grasp, we planned 100 grasps (i.e. the best grasp of n grasp candidates). Of these 100 trials, we plotted the best and the worst grasp quality. The number of grasp candidates generated to plan one grasp is plotted on the x-axis (fig. 7). We see that increasing the number n of grasp candidates to be generated to plan a grasp doesn't improve the worst case result as well if we exceed approximately $n = 100$. In this case the worst grasp we obtained had 65 - 75 % of the best grasps' quality. This means that we had to apply 1.33 to 1.54 times the forces at the contact points for the worst case grasp than for the best grasp. We see that only a relatively small number of approx. 100 grasp candidates is necessary in order to obtain reliable, good quality results.

Using this result, time measurements yielded the following planning times:

Object	time for 100 candidate grasps	8 CPU's
Sphere	14 s	2 s
Cube	13 s	2 s
Phone	68 s	9s
Banana	29 s	4 s

The experiments were performed on a *SGI INDIGO^2 Maximum Impact R10000.* The planning time can be varied very easily by choosing n, but as can be seen in fig. 7 increasing n doesn't improve the average grasp quality proportionally, while e.g. selecting $n = 50$ still yields good (62 - 69 % of best case) grasps in the worst case, but only needs about 3 - 10 s of planning time.

As the structure of the planner – testing a given number of independent grasp candidates – is well-suited for parallelisation in order to improve planning performance, we also implemented a parallel version of it. Due to the low communication costs, we get an almost linear (in the number of CPUs) speedup.

9 Future Work

One research topic at our institute is sensor based world model update. Using different sensors, we can obtain a partial, polygonal model of an unknown object. As the grasp candidate generation step of our algorithm is based on a ray cast onto a geometric face set only, it can be extended to generate grasps for partially known objects easily. This is important for integrating the grasp planning algorithm in a robotic system which is able to sense and grasp unknown objects. Getting a better understanding on force/torque scaling effects on the grasp measure (currently we scale torques with the size of the largest torque that a unit force might generate when acting on the given object) and incorporating the effects of gravity into the grasp measure also are current research topics.

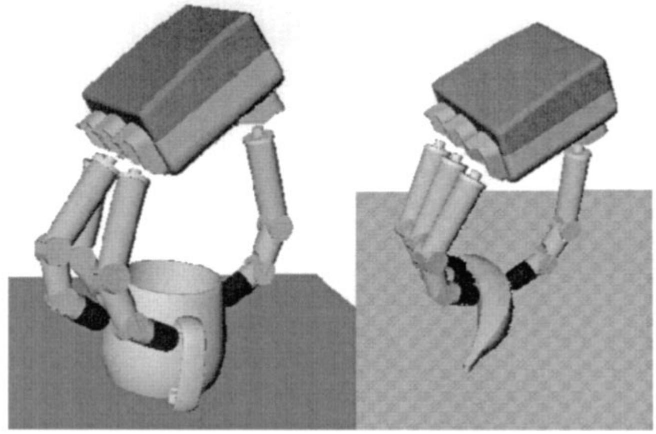

Figure 8: Planned grasp for Mug and Banana

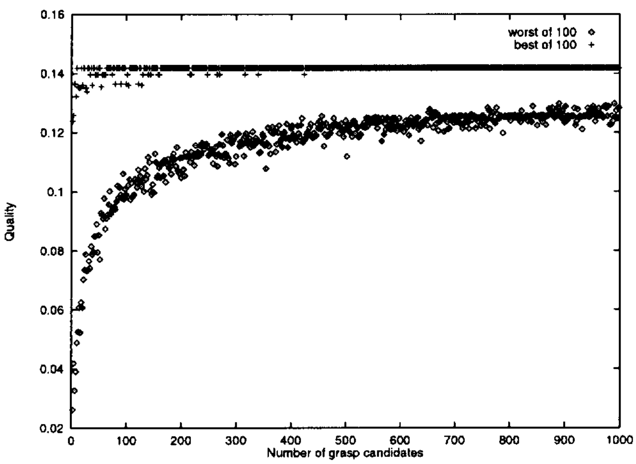

Figure 7: Quality results for Banana

References

[1] Ch. Borst. *Untersuchung von Stabilitäts - und Robustheitsmaßen zur Planung von Fingerspitzengriffen.* Diploma Thesis, Institute of Robotics and System Dynamics, DLR Oberpfaffenhofen, 1998.

[2] J. Butterfass, G.Hirzinger, S. Knoch, and H. Liu. DLR's Multisensory Articulated Hand. part I: Hard- and software architecture. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 2081 - 2086, Leuven, 1998. IEEE.

[3] C. Ferrari and J. Canny. Planning optimal grasps. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2290 - 2295, Nice, France, May 1992. IEEE.

[4] M. Fischer and G. Hirzinger. Fast planning of precision grasps for 3D objects. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 1:120–126, Grenoble, 1997.

[5] B. Mirtich and J. Canny. Easily computable optimum grasps in 2-d and 3-d. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 739 – 747, San Diego, California, May 1994. IEEE.

[6] V.-D. Nguyen. Constructing force-closure grasps. *Int. Journal of Robotics Research*, 7(3):3 – 16, June 1988.

[7] J. Pertin-Troccaz. Grasping: A state of the art. In O. et al. Khatib, editor, *Robotics Review*, volume 1, pages 71 – 89. MIT Press, Cambridge, MA, 1989.

[8] N. Pollard. Synthesizing grasps from generalized prototypes. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 2124 –2130, Minneapolis, Minnesota, April 1996. IEEE.

[9] K. B. Shimoga. Robot grasp synthesis: A survey. *Int. Journal of Robotics Research*, 15(3):230 – 266, June 1996.

[10] A. Sudsang and J. Ponce. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, pages 1355 – 1360, Nagoya, Japan, May 1995. IEEE.