GraspIt!: A Versatile Simulator for Grasp Analysis

Andrew T. Miller   Peter K. Allen
Department of Computer Science
Columbia University
New York, NY 10027 *

ABSTRACT

We have created a unique tool for grasp simulation, visualization, and analysis that allows a user to create and analyze grasps of a given 3D object model with a given articulated hand model. The grasps can be performed either automatically, where the system closes the fingers around the object at preset velocities, or manually through direct manipulation of the joints. As collisions occur between the links of the fingers and the object, the system locates the contacts and analyzes the evolving grasp on the fly. Each time the grasp changes, the system updates two numeric measures of quality and recomputes 3D projections of the grasp wrench space which are useful when visualizing a grasp's capabilities. We provide examples of the system being used with four different articulated robotic hand models, each grasping different object models. We feel the system is very useful for hand designers who prototype different hand models in simulation and determine how design decisions affect a hand's grasping ability. It is also useful for researchers in grasp planning or for simulation and virtual reality designers wishing to perform realistic grasping in a virtual setting.

1 INTRODUCTION

Designing an articulated robotic hand is a difficult task with many considerations such as task requirements, mechanism complexity, and physical size and weight. Often a physical prototype is necessary to truly test a hand's ability to perform tasks, but this can be quite costly and design changes are not easy to make. What is needed is a system that would allow a user to load a hand design, to interact with it and perform grasps of objects, and to visualize and evaluate the space of forces and torques that it can apply.

*This work was supported in part by an ONR/DARPA MURI award ONR N00014-95-1-0601, DARPA AASERT awards DAAHO4-93-G-0245 and DAAH04-95-1-0492, and NSF grants CDA-96-25374 and IRI-93-11877.

The increasing power of the currently available tools for geometric modeling and computational geometry have made such a system possible. An earlier paper [12] described our initial version of a system which when given 3D models of a robotic hand and an object, as well as a user supplied hand configuration, would run offline and analyze the grasp. Upon completion it presents two numeric measures of the grasp quality as well useful 3D projections of the 6D grasp wrench space. We have since taken the core analysis routines from that system and built a real-time, interactive simulator that allows the user to manipulate the hand, perform grasps, and see the analysis results on the fly. The intent of this paper is to demonstrate the versatility and effectiveness of our latest system, which we have dubbed "GraspIt!", as well as to describe its software components.

This system allows us to test a hand's ability to grasp different kinds of objects. This enables comparison of hand designs to find the one best suited to particular kinds of tasks. Until now, hand designs have emulated the human hand because of its proven ability to perform complex grasping and manipulation tasks, but a system with these capabilities would be an indispensable tool for designing and testing alternate possibilities. Another use of this system is planning grasping tasks for an existing robotic hand. Future work will allow a user to synthesize locally optimal grasps and will consider more of the task requirements and reachability constraints. Finally, GraspIt! can easily become part of a larger simulation or virtual reality project where realistic grasping of objects is required.

In summary we feel our system has the following advantages:

- We have a large and growing library of hand and object models.
- By employing a general method of specifying kinematics and geometry it has the flexibility to allow a wide variety of new hand designs.
- Real-time collision detection and grasp evaluation algorithms allow direct interaction with the hand and results are produced on the fly.
- Grasps can be performed automatically, saving time over individ-

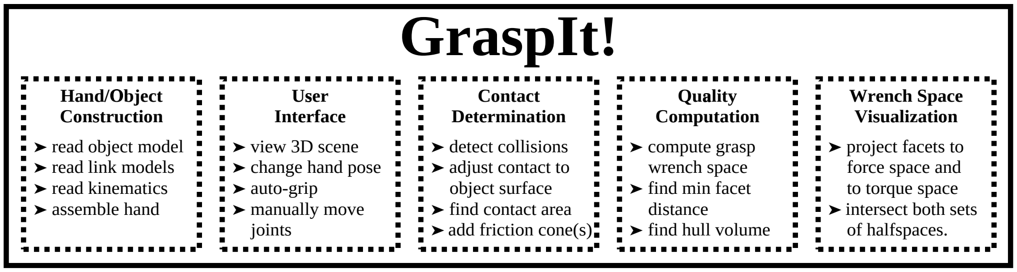


Figure 1: The internal components of GraspIt! and their functions.

ually manipulating each degree of freedom.

- Novel visualization methods aid the user in understanding a grasp's strong and weak points.

The remainder of this paper is laid out as follows. Section 2 briefly reviews some of the previous research. Section 3 describes the software architecture of GraspIt!, and describes how the various components work together to evaluate grasps. Section 4 demonstrates the system in action by presenting the analysis of example grasps performed by a parallel jaw gripper and the Barrett, DLR, and Robonaut robotic hands. Finally, Section 5 summarizes our contributions and details our proposed research directions.

2 RELATED WORK

There is a great deal of previous research in the field of grasp analysis and synthesis. However, due to space constraints we can only highlight a few of the most relevant papers. See reviews by Mishra and Silver [13] and by Troccaz [15] for references of some of the older work. Salisbury [17] classified types of contacts, with and without friction, between two bodies and provided an approach that accounts for uni-sense contact wrenches in determining whether a grasp completely restrains an object which he calls *form-closure*. Pollard [16] developed a parallel system to compute high quality grasps using prototype grasps as input, and Fischer and Hirzinger [5] created a system that repeatedly chooses 3 contact points on an object using a heuristic and checks whether these points can be realized by the hand. Recent work by Maekawa and Hollerbach [11] will allow a user to feel the interaction forces involved in grasping or manipulating an object with the use of a human hand model and a haptic display device.

Given the goal of minimizing the sum magnitude of the contact forces, Kirkpatrick et al. [7] proposed a general measure of quality for an n-contact grasp, defining it as the radius of the largest wrench space ball which just fits within the unit grasp wrench space. Ferrari and Canny [4] developed this measure further and proposed another measure minimizing the maximum contact force. Earlier, Li and Sastry [8] noted that similar measures are not invariant to the choice of torque origin, and proposed using the volume of the grasp wrench space as an invariant quality measure. They also developed a quality measure using task ellipsoids to better model the space of wrenches required for a task, whereas the previous authors all assumed the task wrench space is unknown and therefore defined the space as a ball centered at the origin of the wrench space. Our implementation of the grasp quality computations is derived from the the presentation of the L_1 unit grasp wrench space by Ferrari and Canny.

3 THE GRASPING SIMULATOR

While GraspIt! is quite easy to use, it relies on several components to run smoothly and efficiently. Some of the operations it must perform include: assembling the geometric link models of the chosen hand, locating contacts, evaluating grasp quality, and producing wrench space projections (see figure 1 for an overview). We describe the workings of each of these pieces in the following sections.

3.1 Hand Construction

We have designed the simulator to be as general as possible so that it is useful for a variety of hands and objects. However, it was necessary to create standard formats for the definitions of a hand and an object, so that they can be understood by the system. An object configuration file is fairly simple, and consists only of a pointer to a CAD model file and a material specification. Currently the surface of the object is assumed to be composed entirely of one type of material to ease the static friction computations.

Our hand description file similarly allows us to describe the link geometries as complete 3D entities, each with an associated material specification. Furthermore, it fully describes the kinematics of the hand using standard Denavit-Hartenberg parameters with additional information regarding joint limits and finger base transforms. Each joint, which is either revolute or prismatic, has its free parameter connected to one of the hand's degree of freedom variables either directly or through a linear function. This allows for coupled joints like the passively controlled distal joint found in many anthropomorphic finger designs. We have found our description method is flexible enough to construct hands as simple as a parallel jaw gripper or as complicated as many of the fully articulated hands in use today.

After reading the description file, GraspIt! reads each of the link model files and sets the initial transforms based on the kinematics. Once the hand construction is completed, it turns control over to the user.

3.2 Collision Detection and Contact Location

Throughout the simulation the object remains fixed with its center of gravity on the WCS origin, and the user controls only the configuration of the hand. The hand can be translated in three dimensions and rotated in three dimensions about any arbitrary point. In addition, the user can individually manipulate each of the hand's internal degrees of freedom or start an auto-grasp where several joints close simultaneously at velocities specified in the hand description file.

The auto-grasp is a new feature in our system that allows the operator to quickly form a grasp at the touch of a key, making the manual positioning of the links a less time consuming process required only for small refinements of the initial grasp. As the grasp closes, any contact between the object and a link causes all of the joints (and their related DOF) that are previous to the link in the finger chain to stop, while joints further in the chain continue to close until another contact is found or the joints reach their limits.

To detect these contacts and prevent the links from passing through the object or each other, we have incorporated the V-Collide collision detection package [6]. During initialization, this package arranges the triangles that make up each distinct object into a hierarchy of oriented bounding boxes. A fast recursive algorithm descends these hierarchies and can determine in real time if two objects are colliding. We modified the algorithm so that in the event that two triangles are colliding, it determines whether all the vertices of the triangles have moved less than some small epsilon. Using this as a stopping condition, we implemented a binary search to refine the hand DOF parameters until the link is separated from the object by a distance less than the tolerance of the solid modeler (currently set to 10^{-3} mm).

At this point GraspIt! calls the ACIS solid modeling engine to perform an intersection between the link and the object resulting in a common surface patch. If the patch consists only of a point, then we classify it as a point contact. If the patch has two vertices then the contact is classified as a line contact, and for a patch with greater than two vertices, we consider it a plane contact. Nguyen [14] points out that these complex contacts (linear and planar) can be represented as the convex sum of proper point contacts at each of the vertices of the surface patch.

To find the space of forces that may be applied at a particular contact, we first find the local contact normal which is defined as the inward pointing normal of the tangent plane of the contact. If a contact tangent plane cannot be defined, the contact is considered unstable and is disregarded for the remainder of the process. If we assume a Coulomb friction model, then the total force, $\mathbf{f}$, acting on the object at a contact point must lie within a cone that has an apex at the contact point, an axis along the contact normal, and a half angle of $\tan^{-1}\mu_s$, where μ_s is the coefficient of static friction. In order to find the total grasp wrench space when we compute the grasp quality, we will need a finite basis set of vectors, so it is necessary to approximate this cone with a proper pyramid (see figure 2). The coefficient μ_s is defined between pairs of materials and can only be found empirically. However, since it can be different for each pair of materials, we have allowed the materials of the links and object to be individually specified, thus increasing the flexibility and realism of the system. Table 1 is a matrix containing the coefficients we have assumed for the materials currently defined in our simulation. These values were obtained from the ranges provided in the CRC Handbook of Chemistry and Physics [9].

Each time a contact occurs we follow this procedure, and if there are at least three point contacts, GraspIt! computes the grasp quality.

$$\mathbf{f} \approx \sum_{j=1}^{m} \alpha_j \mathbf{f}_j$$

$$\alpha_j \geq 0,\ \textstyle\sum_{j=1}^{m} \alpha_j = 1$$

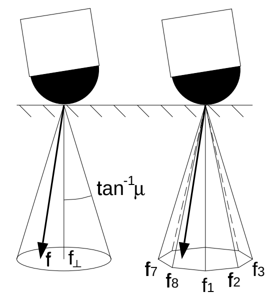


Figure 2: We have approximated the friction cone with an 8 sided pyramid. The contact force, $\mathbf{f}$, is represented as a convex combination of 8 force vectors around the boundary of the cone.

material	metal	glass	plastic	wood	rubber
metal	0.2	0.2	0.2	0.3	1.0
glass	0.2	0.2	0.2	0.3	1.0
plastic	0.2	0.2	0.3	0.4	1.0
wood	0.3	0.3	0.4	0.4	1.0
rubber	1.0	1.0	1.0	1.0	2.0

Table 1: The values of the coefficient of friction for materials defined within GraspIt!.

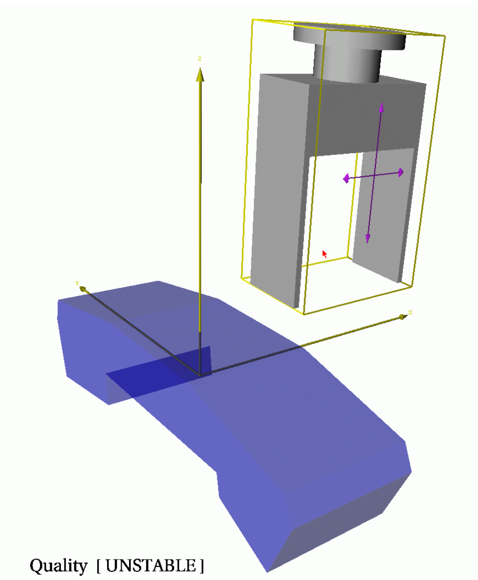
3.3 Computing Grasp Quality

First, the system must analyze each contact and determine its contribution to the overall stability of the grasp. Then we compute a convex hull of the contact wrenches to determine the overall space of wrenches that can be applied by this grasp. Using the hull, the system can compute two commonly used measures of grasp quality. Although we only present a summary of these steps, a more in depth description can be found in our earlier work [12].

Using the object's center of gravity as reference point, we start by finding the 6D-wrench that is associated with each force acting at a contact point. Then we must compute the unit grasp wrench space, and Ferrari and Canny [4] describe two different methods. One limits the maximum magnitude of the contact normal forces to 1, and the other limits their sum magnitude to 1. We have implemented the second option due to its ease of computation. Under this constraint, the set of wrenches that can be exerted on the object is:

$$W = ConvexHull(\bigcup_{i=1}^{n}\{\mathbf{w}_{i,1}, \ldots, \mathbf{w}_{i,m}\}) \quad (1)$$

If this convex hull contains the wrench space origin then the grasp is stable. One quality measure that is often proposed is the radius, ϵ, of the largest 6D ball, centered at the origin, that can be enclosed with the hull. The vector from the wrench space origin to the point where the ball contacts the boundary of the hull is the smallest maximum wrench that can be applied by the grasp. In this worst case the sum magnitude of the contact wrenches would have to be $\frac{1}{\epsilon}$ times the magnitude of the disturbance wrench. The closer ϵ is to 1 the more efficient the grasp is. However, this measure is not invariant to the choice of torque origin as Li and Sastry point out, so the volume v of the convex hull can be used as an invariant average case quality measure for the grasp.



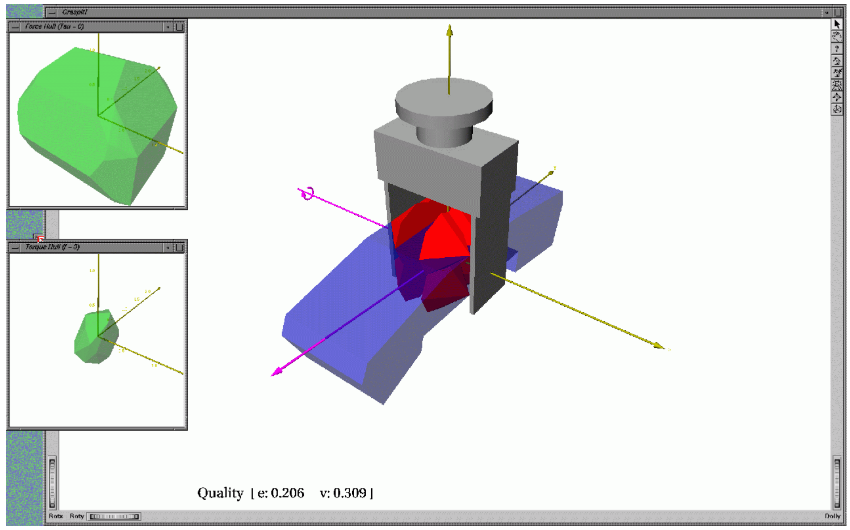


Figure 3: Left: Translating the parallel jaw gripper into position over the phone handset. Right: A completed stable grasp of the phone.

We have implemented this portion of the analysis with the Qhull program [2]. It produces a list of facets of the convex hull, and each is described by a normal vector and a signed offset from the origin. This makes it easy to determine whether the origin is contained in the hull, indicating a stable grasp, and which facet is closest to the origin, indicating the most difficult wrench for the grasp to apply. Clearly, the wrench in the opposite direction is the most difficult external wrench for the grasp to resist. The minimum offset value gives us ϵ and indicates how efficient the grasp is at handling this worst case. Qhull can also be queried for the volume of the computed hull which gives us v.

3.4 Wrench Space Visualization

One reason 3D examples are avoided in previous papers is due to the difficulty of visualizing the results. A 2D planar object will have a 3D wrench space, but a 3D object will have a 6D wrench space, and to display it, we must project it into 3D space by fixing three of the wrench coordinates. We have chosen four projections of the grasp wrench space that can help us understand some of the characteristics of a particular grasp. If we fix the torque coordinates of the wrench space to 0, the resulting projection shows the space of forces that can be applied by the grasp without imparting a net torque to it, and likewise, if we fix the force coordinates to 0, we can visualize the space of torques that can be applied to the object without a net force acting on the body. By setting the torque coordinates to their values at the point on the hull boundary that is closest to the origin, $\mathbf{w}^{\tau}_{min}$, we can display the space of forces that can be applied if the worst case torque needs to be applied. Similarly we can display the space of torques that can be applied if the worst case force, $\mathbf{w}^{f}_{min}$, must be applied. In general, we start with the collection of 6D halfspaces, $H_{n\times 6}\mathbf{x} \leq \mathbf{b}$, but once we have chosen values for three coordinates of $\mathbf{x}$, we arrive at a new set of 3D halfspaces, $H'_{n\times 3}\mathbf{x} \leq \mathbf{b}'$. From there we use Qhull again to compute their intersection which gives us the hull bodies that are presented in the next section.

4 EXAMPLES

In this section, we present examples of different grasps of various objects performed using models of a parallel jaw gripper and two articulated robotic hands. Each section begins with a description of the hand used and continues with a demonstration of the process described in the previous section. These examples are intended to demonstrate the versatility of the system and its effectiveness as a tool for visualizing a grasp's strengths and weaknesses.

4.1 A Parallel-Jaw Gripper

Our gripper model has the approximate dimensions of an actual Lord gripper. For ease of use, we changed the kinematics of our gripper so that each plate can be independently controlled. When only one coupled DOF is used to grip a non-moving object, the gripper must be centered over the object exactly for both plates to come in contact. Besides the kinematics, we also specified the surface material of the palm as metal and the plates as rubber.

Figure 3 shows two screen shots from GraspIt!. The first shot is a portion of the main window which contains our model of the gripper and a model of a telephone handset with a coordinate frame attached to its center of gravity. The user is translating the gripper to a position above the phone's center at which point the gripper can be lowered and its plates closed. Since there is no contact between the gripper and the object, the grasp is still unstable, and that is indicated in the quality readout at the bottom of the window. The second shot, this time of the full-screen, shows that when the jaws are closed, the result is a stable grasp, and the two side windows display selected 3D projections of the computed grasp wrench space. Currently, the top side window shows the 3D space of forces that can be applied to the phone with no net torque, and the bottom side window shows the space of torques that can be applied with no net force, although these projection choices can be

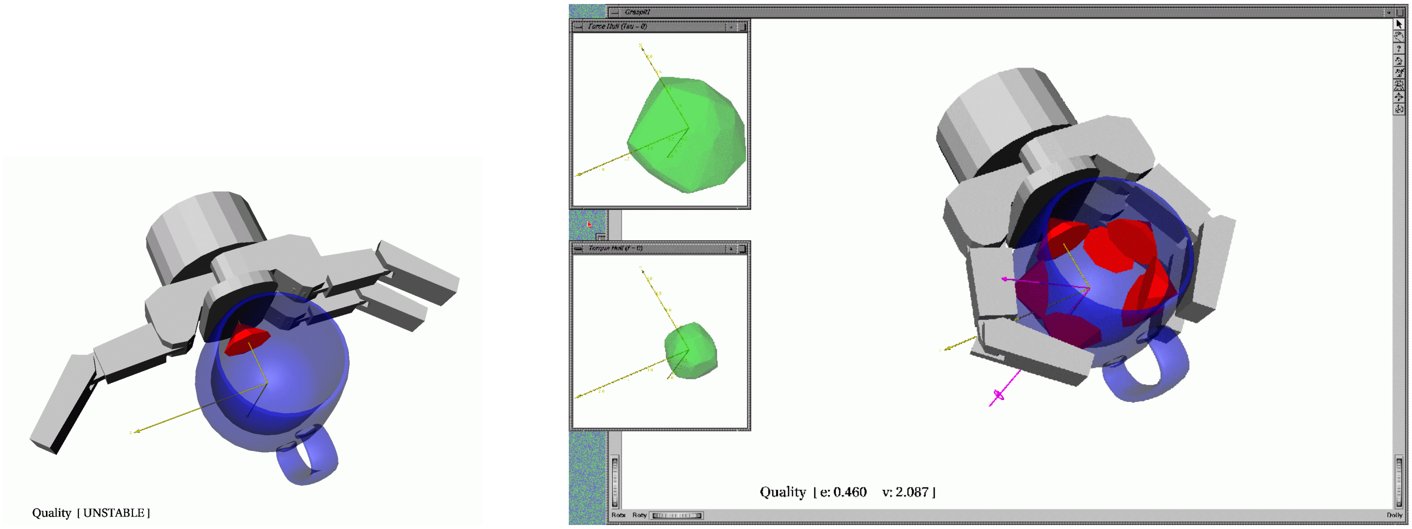


Figure 4: Left: The Barrett hand has been translated until the palm contacted the mug surface. Right: A completed stable grasp of the mug.

changed. The red cones[1] seen on the inside surface of the phone are the friction cones placed at the vertices of the contact regions. Finally, in addition to the coordinate axes present before, two new indicators have appeared in the main window. They show the directions of the force-torque combination that is most difficult for this grasp to resist, in this case a twist parallel to the jaw axis and a simultaneous force pushing tangentially along the the plates. This sort of disturbance wrench can only be resisted with the frictional forces since there are no contacts normal to the twist axis and no contacts opposing the force vector.

4.2 The Barrett Hand

The dextrous robot hand used in this example is the Barrett Hand [1]. It is an eight-axis, three-fingered mechanical hand with each finger having two joints. One finger is stationary and the other two can spread synchronously up to 180 degrees about the palm (finger 3 is stationary and fingers 1 and 2 rotate about the palm). Although there are eight axes, the hand is controlled by four motors. Each of the three fingers has one actuated proximal link, and a coupled distal link that moves at a fixed rate with the proximal link. A novel clutch mechanism allows the distal link to continue to move if the proximal link's motion is obstructed (referred to as *breakaway*). An additional motor controls the synchronous spread of the two fingers about the palm.

The Barrett hand has 10 degrees of freedom: 6 for the pose of the wrist, 1 for the spread angle of the fingers, and 3 for the angles of the proximal links. Our current model of the hand uses slightly simplified link geometries, but the kinematics of the model match the real hand exactly. Because the actual hand in our lab is outfitted with tactile sensors on the inner pad of each link, including the palm, we have set the link material of each link in the model to rubber. Our object to be grasped is a mug, and we have set the material as glass in the object configuration file.

We began this grasp by translating the hand until the flat palm contacted the rounded outer surface of the mug. At this point we have GraspIt! perform an auto-grasp where the three finger motors have the same relative velocity and the spread motor has its velocity set to 0. Each finger has a contact on its inner link which results in the outer link breaking away and continuing to close until another contact occurs at the finger endpoints. The resulting wrapping grasp is very stable due to the number of highly frictional contacts spread out around the mug's center of gravity (figure 4).

4.3 The DLR Hand

The DLR hand [3], developed at the German Aerospace Center, is a four-fingered articulated robotic hand, and it has an anthropomorphic design. The fingers are identical, and each consists of three links with two joints at the base, one joint between the proximal and medial links, and one joint between the medial and distal links. This last joint is coupled in a fixed ratio to the previous joint in the chain just as it is in the human hand. The DLR hand has a total of 18 degrees of freedom: 6 specify the pose of the wrist, and there are 3 independently controllable joints in each of the 4 fingers. In the description file for the DLR hand, we specify its kinematics, including joint limits, and its link materials (metal for the palm, proximal, and medial links, rubber for distal links). Also specified are the link models, which are accurate replicas of the actual hand, except we have omitted unnecessary details at the joints. The object to be grasped in this example is a glass Erlenmeyer flask.

Translating the hand towards the flask results in only one point contact on the palm, but by rotating around that contact point, as seen in the left portion of figure 5, another contact will occur. After performing an auto-grasp where the motor velocities are set so the fingers curl at equal speeds, the result is a somewhat weak but stable grasp shown in the right portion of that figure. However, by manually adjusting the finger joints so that contacts occur on the proximal finger links and so the thumb contacts the bottom surface of the flask, we have found a higher quality grasp (figure 6).

[1] A color version of this paper is available at http://www.cs.columbia.edu/~amiller

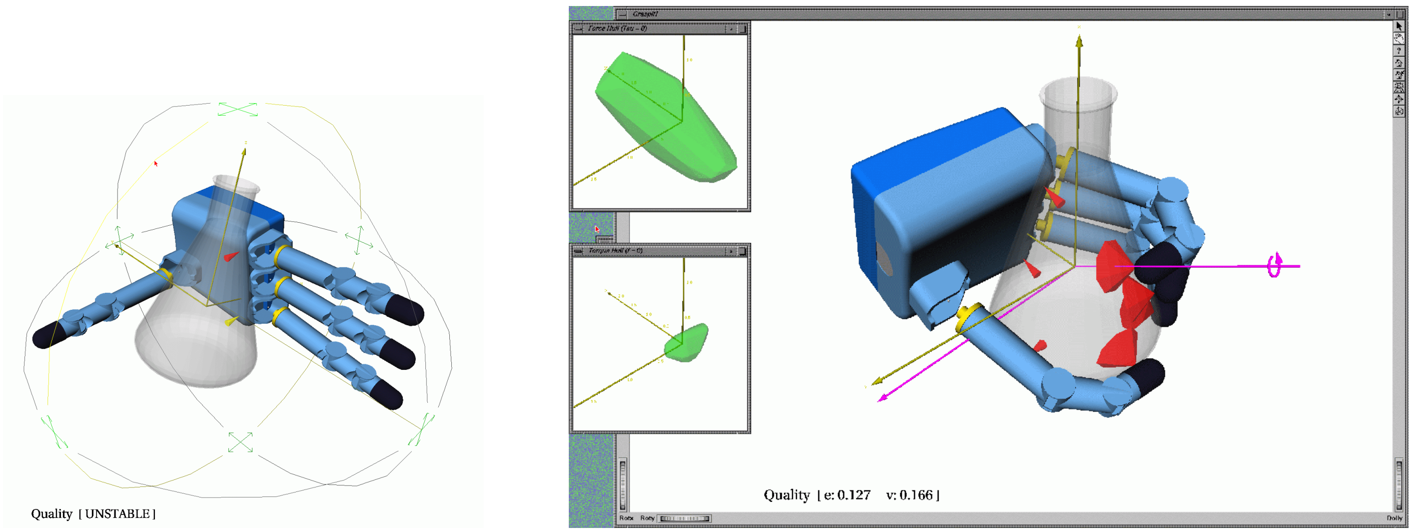


Figure 5: Left: The DLR hand being rotated about a contact point until the second point of contact between the palm and flask occurs. Right: After the auto-grasp has finished the result is a stable grasp of the flask.

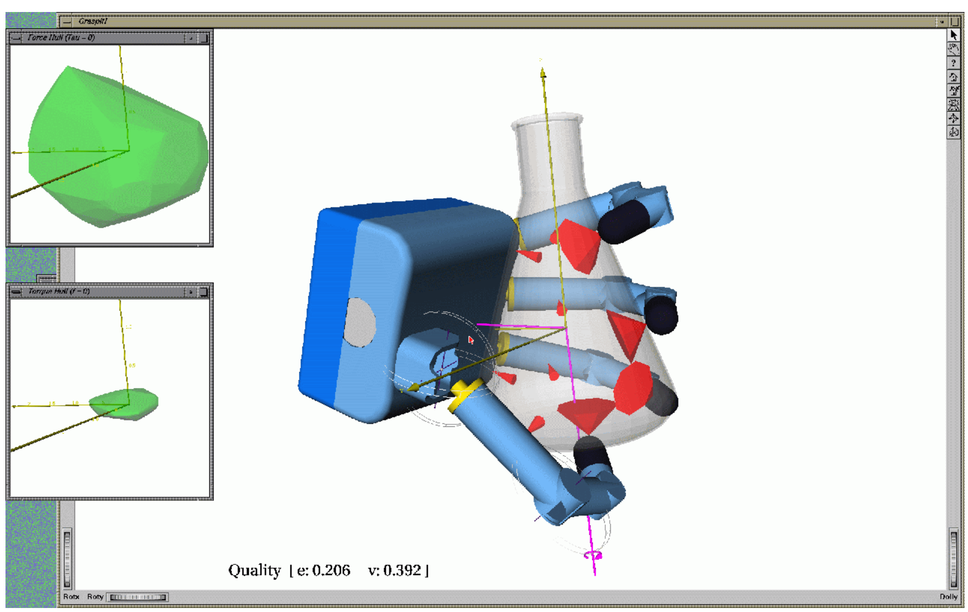


Figure 6: Adjusting the DOF manually can improve upon the auto-grasp. Note the proximal links of the fingers are now in contact with the flask.

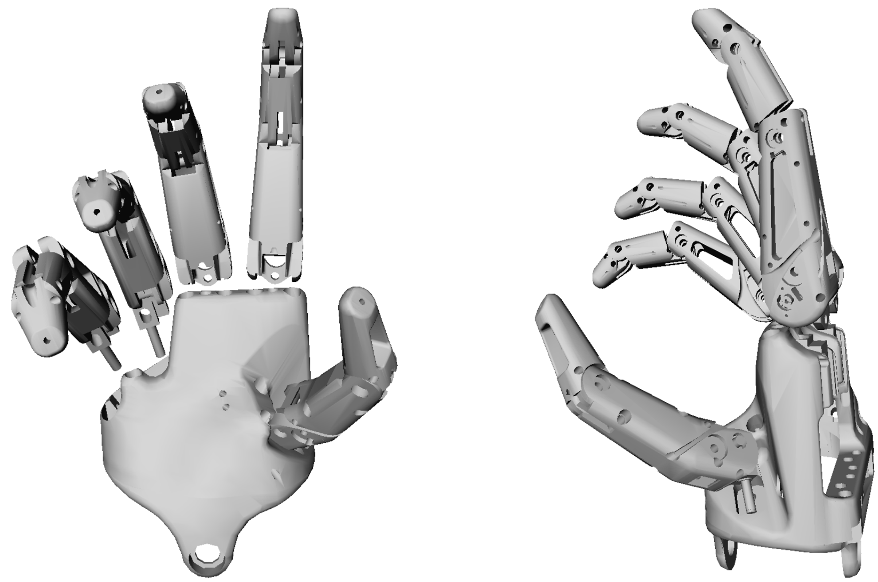


Figure 7: The Robonaut hand model.

4.4 The Robonaut Hand

The final hand we examined was the Robonaut hand developed at NASA's Johnson Space Center [10]. It is the most anthropomorphic and complex of the hand designs with 5 fingers and 14 internal degrees of freedom (see figure 7). The index and middle fingers along with the thumb are considered the primary manipulation fingers and are capable of abduction and adduction. The ring and pinkie are mounted on squeezing palm joint which make them good grasping fingers capable of wrapping around a tool or other object. Although the link geometry only specifies the metal structure of the hand, we assumed rubber contact surfaces to facilitate stronger grasps. In this example, we chose the rather difficult star model to see if the hand's many degrees of freedom could allow it to grasp the star stably.

After moving the palm into contact with the top surface of the star, we rotated the two primary fingers to the left away from the secondary grasping fingers. This pregrasp shape is shown on the left side of figure 8. Then we performed an auto-grasp which closed the fingers around the sides of the star. A small bit of adjustment was necessary to bring two of the finger tips in contact with the bottom surface of the star. While we were able to perform a relatively stable grasp of the star, we note that the grasp is still weak at resisting forces pushing away from the palm. This object model may be at the large end of the range of what this hand can grasp given its kinematic structure. Further research will determine the size and shapes of objects that are easiest for this hand to grasp.

5 CONCLUSION AND FUTURE WORK

We have demonstrated a system that, given models of a hand and an object, allows a user to manipulate the hand to perform grasps of the object which the system analyzes. The real-time analysis results in two numeric measures of grasp quality as well as indicators of the grasp's weak point, and a partial visualization of the space of forces and torques that can be applied by the grasp. The system has been designed to be flexible so it is easy to add in new hand or object models, and thanks to the fast collision detection algorithm, we are able to analyze how complex models interact with real link geometries, rather than restricting ourselves to contacts on the ends of rounded fingertips.

Future work will include adding more quality measures to the simulator, including a measure of a grasp's manipulability, and adding a variety of hand and object models. Such a grasping library will allow us to perform many comparative tests and help answer the question of how the kinematic and geometric design of a hand affects grasp stability and grasp manipulability. We are also interested in the area of grasp synthesis, and while the total space of hand configurations is prohibitively large, we hope to use heuristics and a limited number of hand preshapes to limit the total space we actually look at. We would like to have GraspIt! at least locally optimize a grasp performed by the user. Finally, we would like to add a workspace and an arm attachment so we can consider reachability constraints and possibly test these grasps with a real hand.

Acknowledgment: We would like to thank Professor Gerd Hirzinger and Dr. Max Fischer from the German Aerospace Center (DLR) for providing us with models of their robotic hand. We would also like to thank Dr. Myron Diftler of NASA's Johnson Space Center for the Robonaut hand model.

References

[1] P. K. Allen, A. T. Miller, P. Y. Oh, and B. S. Leibowitz. Using tactile and visual sensing with a robotic hand. In *Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation*, April 1997.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.

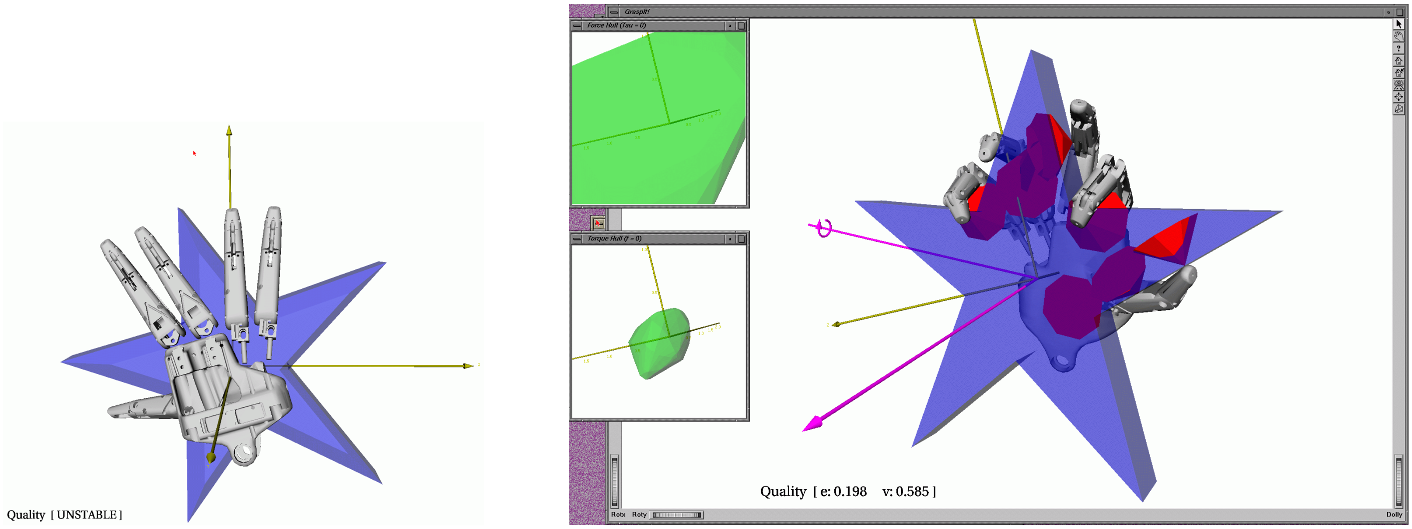


Figure 8: Left: The Robonaut hand has a palm contact on the top surface of the star and the fingers have been abducted to form the pregrasp shape. Right: The star is held stably after minor adjustments to the result of the auto-grasp.

[3] J. Butterfass, G. Hirzinger, S. Knoch, and H. Liu. DLR's multisensory articulated hand, part I: Hard- and software architecture. In *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, pages 2081–2086, 1998.

[4] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. of the 1992 IEEE Int. Conf, on Robotics and Automation*, pages 2290–2295, 1992.

[5] M. Fischer and G. Hirzinger. Fast planning of precision grasps for 3D objects. In *Proc. of the 1997 IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 120–126, 1997.

[6] T. Hudson, M. Lin, J. Cohen, S. Gottschalk, and D. Manocha. V-Collide: accelerated collision detection for VRML. In *Proc. of VRML'97*, 1997.

[7] D. Kirkpatrick, B. Mishra, and C. K. Yap. Quantitative Steinitz's theorems with applications to multifingered grasping. In *Proc. of the 20th ACM Symp. on Theory of Computing*, pages 341–351, 1990.

[8] Z. Li and S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, RA-4(1):32–44, February 1988.

[9] D. R. Lide, editor. *CRC Handbook of Chemistry and Physics*, pages 15–40 to 42. CRC Press, seventy-fifth edition, 1995.

[10] C. S. Lovchik and M. A. Diftler. The Robonaut hand: A dextrous robot hand for space. In *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, pages 907–912, 1998.

[11] H. Maekawa and J. M. Hollerbach. Haptic display for object grasping and manipulating in virtual environment. In *Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation*, pages 2566–2573, 1998.

[12] A. T. Miller and P. K. Allen. Examples of 3D grasp quality computations. In *Proc. of the 1999 IEEE Int. Conf. on Robotics and Automation*, pages 1240–1246, 1999.

[13] B. Mishra and N. Silver. Some discussion of static gripping and its stability. *IEEE Journal on Systems, Man, and Cybernetics*, 19(4):783–796, August 1989.

[14] V.-D. Nguyen. Constructing force-closure grasps. In *Proc. of the 1986 IEEE Int. Conf. on Robotics and Automation*, pages 2290–2295, 1986.

[15] J. Pertin-Troccaz. Grasping: A state of the art. In O. Khatib, J. J. Craig, and T. Lozano-Pérez, editors, *The Robotic Review*, volume 1, pages 71–98. MIT Press, Cambridge, MA, 1989.

[16] N. S. Pollard. *Parallel Methods for Synthesizing Whole-Hand Grasps from Generalized Prototypes*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 1994.

[17] J. K. Salisbury. *Kinematic and Force Analysis of Articulated Hands*. PhD thesis, Department of Mechanical Engineering, Stanford University, 1982.