Trajectory-Based Grasp Interaction for Virtual Environments

Zhenhua Zhu, Shuming Gao, Huagen Wan, and Wenzhen Yang

State Key Lab of CAD&CG, Zhejiang University
Hangzhou 310027, P.R. China
{cocoon, smgao, hgwan, ywenz}@cad.zju.edu.cn

Abstract. Natural grasp interaction plays an important role in enhancing users' immersion experience in virtual environments. However, visually distracting artifacts such as the interpenetration of the hand and the grasped objects are always accompanied during grasp interaction due to a simplified whole-hand collision model, discrete control data used for detecting collisions and the interference of device noises. In addition, complicated distribution of forces from multi-finger contacts makes the natural grasp and manipulation of a virtual object difficult. In order to solve these problems, this paper presents a novel approach for grasp interaction in virtual environments. Based on the research in Neurophysiology, we first construct finger's grasp trajectories and detect collisions between the objects and the trajectories instead of the whole-hand collision model, then deduce the grasp configuration using collision detection results, and finally compute feedback forces according to grasp identification conditions. Our approach has been verified in a CAVE-based virtual environment.

1 Introduction

Virtual environments (VE) provide a platform for users to experience and work with three-dimensional computer generated scenes just like in real environments. Yet, after many years of research and development virtual environments are still used mainly as a visualization tool with some simple or specialized interaction techniques. Although some finger tracking devices such as instrumented gloves have made natural grasp interaction possible at least at hardware level, the interaction using a virtual hand as an avatar has come true only to a very limited extent. It is probably because the use of multi-finger grasp interaction with a haptic device presents a number of new challenges over that of single point interaction.

These challenges include: 1) how to simplify collision models in order to get high update rate for haptic rendering; 2) how to control visually distracting artifacts, such as interpenetration of a hand and grasped objects, mainly aroused by the interference of device noise and the discrete control data used for collision detection acquired from interaction devices; 3) how to model friction and resolve the distribution of finger force from multiple finger contact points.

Aiming to solve these problems, this paper proposes a simple and novel approach for users to naturally grasp and manipulate objects via a dexterous hand as an avatar in a virtual environment.

The rest of the paper is organized as follows. Related works are briefly reviewed in the next section. Section 3 overviews the approach. In Section 4, the construction of grasp trajectories is presented, while issues related to multi-finger grasp interaction using grasp trajectories is described in Section 5. Experiment results are shown in Section 6. Some conclusions and future works are discussed in Section 7.

2 Related Works

In virtual environments, real-time collision detection between a dexterous hand and objects is premier. Although the technology of continuous collision detection ([1] and [2]) has improved greatly, the requirement of high update rate makes it still expensive in virtual environments with haptic rendering. So, some discrete collision detection methods, such as VOXMAP-PONTSHELL [3], Bounding Sphere Tree [4], Axis Aligned Bounding Box Tree [5], Oriented Bounding Box Tree [6] and Convex Hull Tree [7] are still preferred. But those discrete methods inevitably arouse interpenetration.

In order to alleviate the unrealistic vision of interpenetration, Rezzonico et al. correct hand posture by unfolding the closet proximal joint (wrist side) until the corresponding sensor is tangent to the object or the joint reaches its limit [8]. Zachmann et al. convert the problem of natural grasping into a minimization problem for a joint vector under the constraint that finger-joints (and palm) must not penetrate the object [9].But their iterative adjustment is time-consuming.

Recently, physical-based dynamic simulation is employed for multi-finger grasp and manipulation of a virtual object. Based on point collision response forces, Hirota et al develop a manipulation system [10]. Melder et al. present an approach to allow users to manipulate a virtual object through multiple PHANToM devices by using friction cone ([11], [12] and [13]). Borst et al. develop a system to support natural whole-hand interactions in a desktop-sized workspace [14]. Yet, all of these methods are not very stable and sometimes face difficulties when grasping or manipulating objects. Therefore, approaches dependent on heuristic analysis of grasp stability or user intent are still applicable. Iwata et al. consider whether an object is captured by testing 16 points on a hand model [15]. Maekawa et al provide two finger grasp conditions to manipulate objects in virtual environments [16]. Piater et al. determine grasp configurations by using visual features [17]. Tzafestas et al. take into account the unilateral nature of the contacts and the limitations due to static friction to identify whether grasp is still maintained [18].

3 Overview of the Approach

Before outlining our approach, we narrow down the discussion scope of this paper within pinch, based on users' common operations performed in virtual environments and at the same time for the sake of reducing interaction complexities between a hand and virtual objects. Therefore, consistent and independent fingertip motion for reach-to-grasp movements is fully utilized here. According to the research result of Neurophysiology "For reach-to-grasp movements to a variety of objects, fingertip motion was quite similar. The movement tended to follow a particular curved path." [19], our approach first constructs each fingertip's grasp trajectory and then detects collisions between objects and trajectories. Subsequently, the fingers' automatic contact conditions are estimated and the grasp configurations of the relevant fingers are deduced according to the collision detection results. Finally the finger feedback forces when grasping or manipulating an object is computed based on grasp identification conditions. Overall, the approach is composed of the pre-processing stage and the running stage, whose schematic overview is shown in Fig.1.

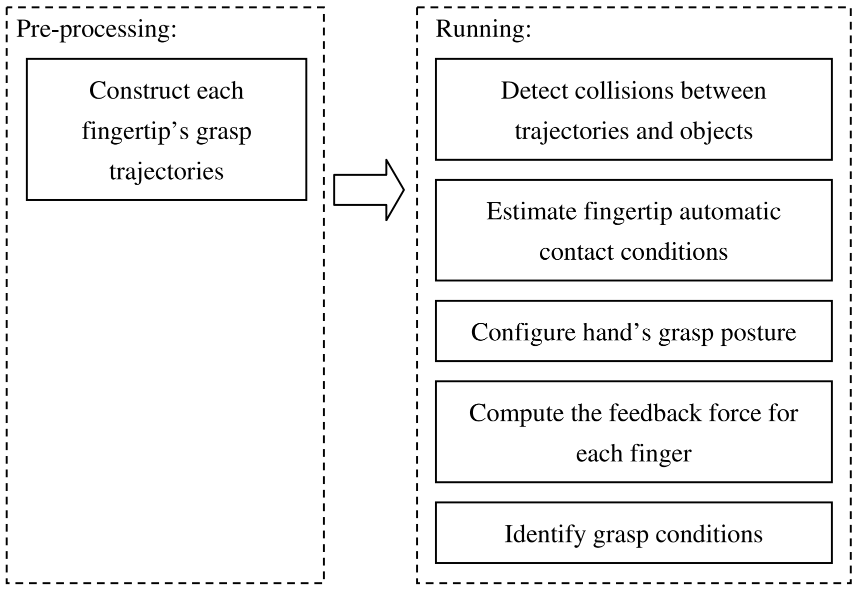

Pre-processing:
Construct each fingertip's grasp trajectories
Running:
Detect collisions between trajectories and objects
Estimate fingertip automatic contact conditions
Configure hand's grasp posture
Compute the feedback force for each finger
Identify grasp conditions

Fig. 1. Schematic overview of the approach

4 Construction of Grasp Trajectories

To gain real-time performance in virtual environments, most traditional methods have to maintain two kinds of hand models, one for display and the other, much simpler, for detecting collisions. Although two kinds of hand models could speed up collision detection, the simplified collision model, discrete control data used for collision detection and the interference of device noise will more or less lead to interpenetration of the hand into grasped objects.

The research result of Neurophysiology mentioned in Section 3 provides us a new promising way to solve this problem. In this section, we will put our emphasis on the construction of a fingertip's grasp trajectory, while leave the issue of how to use trajectories to control the visually distracting artifacts to the next section.

Similar to the method presented in [19], the construction of finger's grasp trajectories is described as follows:

1) Manually select a point in each fingertip surface as a seed point for that finger. The point should be located in the center of the fingertip's surface, where we think the first contact generally happens when pinching an object (Fig.2).

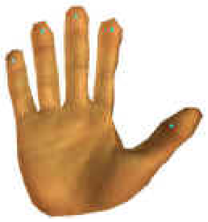


Fig. 2. Seed points on five fingertip

2) Ask a user to participate in a grasping task and his finger joint angles acquired from CyberGlove® are recorded at approximately 50 Hz. The process is repeated several times to eliminate the side effect of some accidental factors such as device noises as far as possible.
3) Simulate the user's grasping process and generate the motion of the seed point for each finger with those recorded joint angles. The seed point's motion trajectory is regarded as a grasp trajectory for that finger.
4) Approximate the grasp trajectory of each finger with a series of line segments. We pick up a number of recorded finger's joint angles as critical joint angles and the positions of seed point corresponding to those joint angles are computed to form critical points on the grasp trajectory. All of these critical points form a series of line segments to approximate the grasp trajectory tightly (Fig.3).

We have to admit that the approximation will result in some inaccuracies both in detecting collision and determining the grasp configuration of the finger. Fortunately, we can control these inaccuracies by determining the number of these critical points on the trajectory. Moreover the speed of collision detection between an object and a series of line segments is obviously far faster than that of collision detection between an object and a curve.

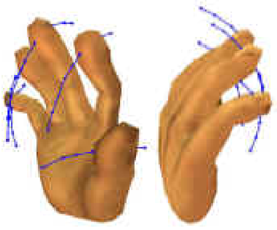
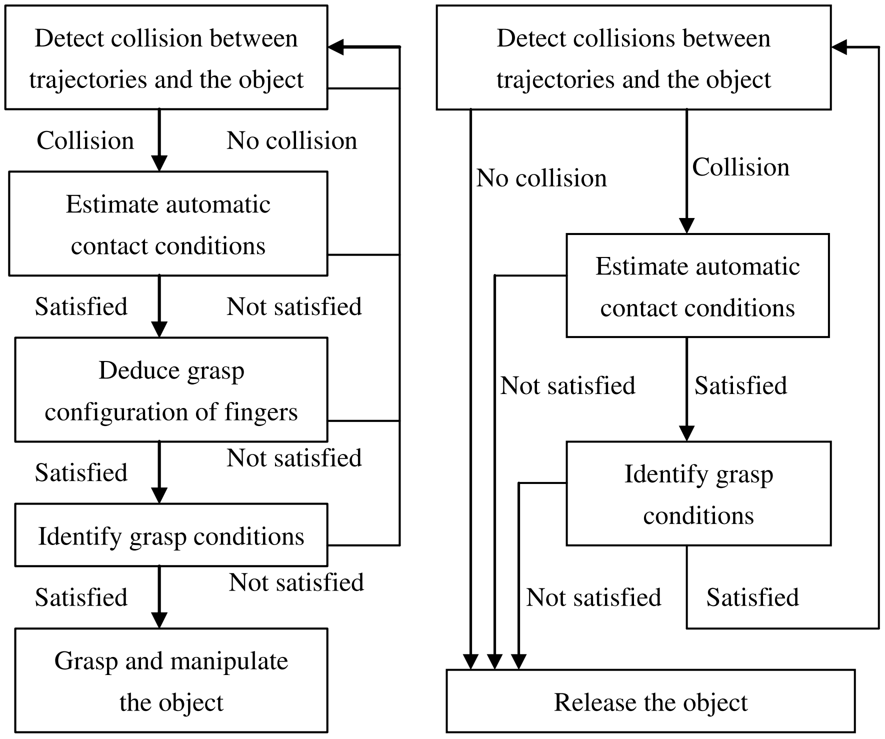


Fig. 3. Line segments to approximate grasp trajectories

5 Grasp Interaction

Fig.4 presents the process of a dexterous hand interacting with an object we are conceiving, and more details will be described below.


Detect collision between trajectories and the object
Collision
No collision
Estimate automatic contact conditions
Satisfied
Not satisfied
Deduce grasp configuration of fingers
Satisfied
Not satisfied
Identify grasp conditions
Satisfied
Not satisfied
Grasp and manipulate the object
Detect collisions between trajectories and the object
No collision
Collision
Estimate automatic contact conditions
Not satisfied
Satisfied
Identify grasp conditions
Not satisfied
Satisfied
Release the object

(a) The process of grasping an obect (b) The process of releasing an object

Fig. 4. Grasp Interaction between a hand and an object

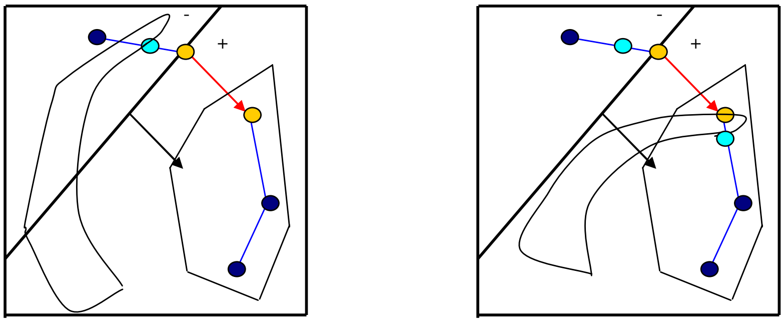
5.1 Collision Detection Between Trajectories and Objects

The collision detection is performed between objects and the grasp trajectories instead of the virtual hand. As the grasp trajectory of each fingertip is approximated by a series of line segments, the problem of collision detection can be therefore converted to perform an intersection test between the objects and the line segments.

An intersection test between a line segment and an object could be implemented by a general ray-tracing algorithm [20]. But to accelerate the intersection computation, an OBB-tree for each object is created. Ray-Box intersections are firstly tested and if a line passes through all Ray-Box intersection tests, a Ray-Triangle intersection is performed (An object is represented by mesh in this work). For Ray-Triangle intersection, the algorithm presented by Möller [21] is applied, while for Ray-Box intersection, the Mahovsky's algorithm [22] is employed. The algorithm makes use of Plücker coordinates and tests the ray against the edges comprising the silhouette of the box instead of testing against individual faces so that the technique's performance is up to 93% faster.

5.2 Automatic Contact Estimation

When pinching, an automatic contact condition is provided to estimate whether fingers contact an object, since it is difficult to predicate when the fingers will contact the object. At first, the line segment intersecting with an object is found during collision detection and then an auxiliary plane passing the start point of the intersecting line segment and perpendicular to the line segment is created to divide the space into the positive and negative subspaces respectively. If the seed point on the fingertip resides in the positive subspace, an automatic contact condition is thought to be satisfied and the corresponding finger will be automatically moved to contact the object. Otherwise, the corresponding finger is still allowed to move along its trajectory. The condition is illustrated in Fig.5.



(a) Seed point in negative sub-space (b) Seed point in positive sub-space

Fig. 5. Illustration of automatic contact estimation

5.3 Grasp Configuration Deduction

After it has been estimated that one finger should be automatically moved to contact a virtual object, it is necessary to determine the grasp configuration of that finger. Without iteratively adjusting finger's posture, our method is able to deduce the grasp configuration of the finger immediately. During our construction of finger's trajectory, the flexion/extension of the distal inter-phalangeal (DIP), proximal inter-phalangeal (PIP), and metacarpal-phalangeal (MCP) joints as well as its corresponding abduction, such as Ring-Middle abduction, are recorded and represented as a set of angles (θ_{n1}, θ_{n2}, θ_{n3}, θ_{n4}) for each critical point P_n. As illustrated in Fig.6, the grasp configuration of the finger could be deduced based on these data as follows:

1) For the finger whose grasp trajectory intersects with an object on the point C_n, determine a variable t that makes

$$C_n = P_{n-1} + t\times(P_n - P_{n-1}) \quad (1)$$

2) Get corresponding joint angles (θ_{n-11}, θ_{n-12}, θ_{n-13}, θ_{n-14}) about P_{n-1} and (θ_{n1}, θ_{n2}, θ_{n3}, θ_{n4}) about P_n;

3) Compute approximate joint angles (θ_{c1}, θ_{c2}, θ_{c3}, θc_4) about C_n by using linear interpolation:

$$\theta_{cx} = \theta_{n-1x} + t\times(\theta_{nx} - \theta_{n-1x}) \ (x = 1,2,3,4) \tag{2}$$

4) Apply (θ_{c1}, θ_{c2}, θ_{c3}, $\theta_{c4)}$ to formulate the grasp configuration of the finger, i.e. the flexion or extension of DIP, PIP and MCP and the abduction.

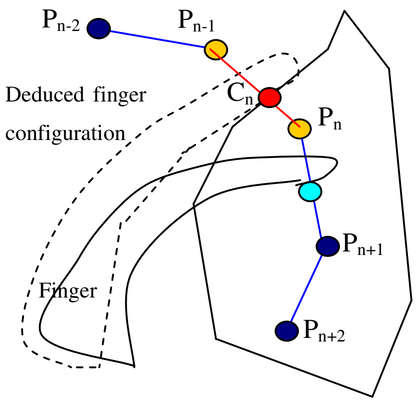


Fig. 6. Illustration of deducing finger's grasp configuration

5.4 Finger Force Computation and Feedback

The finger force responding to collision and being fed back to a user is generated, when collision happens. Realistic finger feedback forces are very important to enhance users' immersion experience in virtual environments. In this paper, the feedback force for each finger is computed according to the relationship between the dexterous hand and the object collided with and two kinds of computation models are presented. One is grasping force computation model, which is utilized when the dexterous hand contacts the object but does not grasp and manipulate it. The other is manipulating force computation model, which is used when the dexterous hand grasps and manipulates the object. No matter which one is used, the feedback forces will finally be mapped to the user's fingertips through CyberGrasp, a haptic feedback device developed by Immersion Corp using its application programmer interface (API): vhtCyberGrasp->setforce() [23].

5.4.1 Grasping Force Computation

For force calculation at a single finger, we refer to the penalty-based force computation model. The force generated on each finger is calculated by utilizing the Hooke's law:

$$\vec{F} = (K_f \times d)\vec{N} \tag{3}$$

Where d is the penalty depth, namely the distance from the collision point to the seed point on finger's surface along the vector $\vec{N}$, and K_f is the stiffness coefficient of the object collided with.

5.4.2 Manipulation Force Computation

Most often, besides the grasping force, object's gravity also has an effect on finger force distribution when an object is manipulated. Considering force effects alone, we observe that the effect of the object's gravity on a finger is relevant to the angle between the grasping force of the finger and the gravity of the grasped object. The bigger the angle, the larger the force received by the finger. So, we allocate the gravity of the object to fingers using dynamic weights, described as follows, and then feed them back to the user plus the grasping forces calculated before.

1) For the i-th contact finger, the angle α_i between the grasping force $\vec{F}$ and the gravity force $\vec{G}$ is calculated.
2) For the i-th contact finger, its new feedback force, $\vec{F}'$, is calculated as:

$$\vec{F}' = \vec{F} + (\alpha_i/\Sigma\alpha_i)\,\vec{G} \tag{4}$$

5.5 Multi-finger Grasp Identification

In order to judge which finger force computation model should be used, some conditions must be provided. Here, both elementary and advanced grasp identification conditions are given. Actually, these conditions could also be used to identify the state of the hand. More details refer to [24].

5.5.1 Elementary Grasp Identification

Elementary grasp identification condition is employed to preliminarily determine whether the hand is able to grasp an object, when collision occurs between them. The condition is that object must be contacted by the thumb and any other finger of the hand. Obviously, if a user wants to grasp an object later, the elementary grasp identification condition must be first satisfied.

5.5.2 Advanced Grasp Identification

Elementary grasp identification condition only preliminary differentiate whether the hand is able to grasp an object. It is advanced grasp identification conditions that determine whether the user could manipulate an object via the dexterous virtual hand from physical aspects.

Considering that the forces exerted on the grasped object include not only press but also friction, which is variable, before describing the advanced conditions, the following two suppositions are introduced:

1) If the forces exerted by the dexterous hand on the object can counteract its gravity force in Z-Axis and the forces' directions can balance in X or Y axis, then the object can be manipulated.
2) The direction of the i-th finger's friction is identical with the negative direction of the projection of the gravity vector of the object on the contact plane (For each contact point, we define a plane passing the point and perpendicular to the direction of the grasping force as the contact plane) and its magnitude ranges from 0 to f_{imax} ($f_{imax} = \mu F_i$, where F_i is the grasping force of the i-th finger to the object, μ is the static friction coefficient, and f_{imax} means the i-th finger's maximum static friction.).

According to the above suppositions, the advanced conditions are given as follows:

1) The inequation below should be satisfied.

$$\sum_{i=1}^{k}(F_{iz} + f_{imaxz}) + G_z \geq 0 \tag{5}$$

   Where k is the number of contact fingers, F_{iz} (f_{imaxz}) is the z-component of F_i (f_{imax}). The physical meaning behind this condition is that the forces the virtual hand exerts on the virtual object could counteract its gravity.

2) If one finger has a force component along the positive direction of X-axis (or Y-axis), there must be another finger which has a force component along the negative direction of X-axis (or Y-axis) and vice versa. The physical meaning behind this condition is that the virtual object could resist any impulse from X (or Y) direction.

6 Experiment Results

We have implemented the approach in a CAVE-based virtual environment using the CAVELib™. The CAVELib™ is a powerful API that provides the cornerstone for creating robust interactive three-dimensional (i3D) environments [25]. An Ascension 6 degrees of freedom (DOF) tracking sensor is used for tracking user's hand motion. The CyberGlove® and the CyberGrasp® [23] are used to capture finger motions and

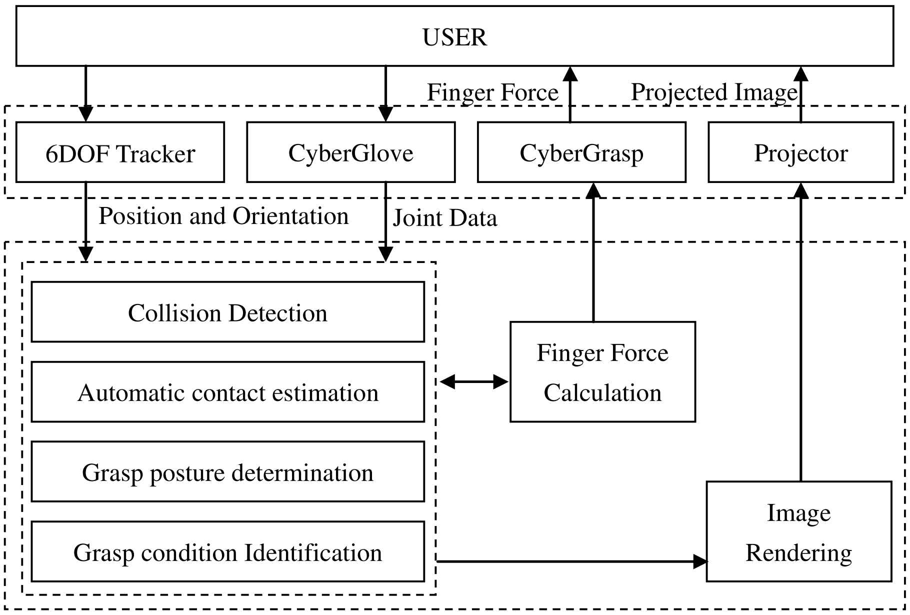


Fig. 7. Architecture of implementation

provide force feedback respectively. The program runs on an SGI Onxy2 (with 4 CPUs and 2 IR4 graphic pipelines). High-resolution stereo images are projected onto four imaging surfaces of the CAVE by four projectors. The overall architecture of the implementation is illustrated in Fig.7.

Fig.8 shows a virtual one-cylinder motor assembly scene we created to test the presented method. The virtual assembly scene is comprised of some common mechanical components such as nuts, bolts. A dexterous virtual hand as well as the proposed grasp interaction is used to perform assembly tasks, and four grasp actions are displayed in Fig.9. Fig.10 gives a snapshot of a user's hand when performing virtual assembly tasks with the CyberGrasp® in our CAVE-based environment.

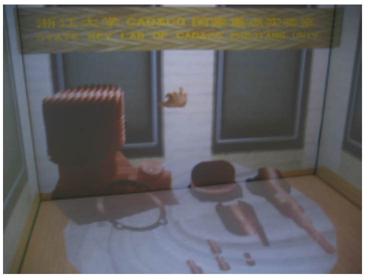


Fig. 8. One-cylinder motor assembly scene

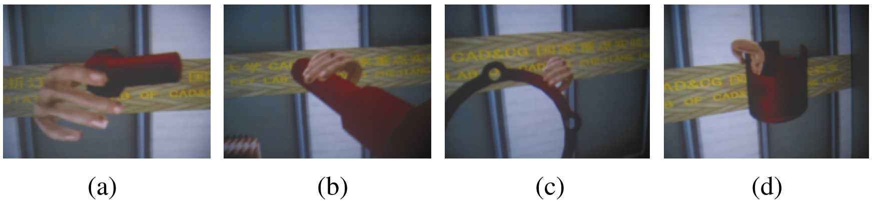


Fig. 9. Grasp interaction with different mechanical components: *(a)* a nut; *(b)* a crank; *(c)* a gasket and *(d)* a piston

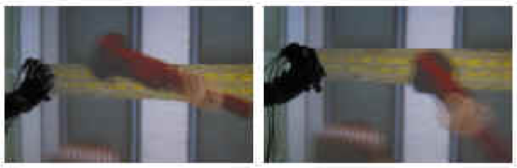


Fig. 10. A snapshot of performing a virtual assembly task using CyberGrasp®

7 Conclusion and Future Work

Natural grasp interaction and its realistic vision play important roles in enhancing users' immersion experience in virtual environments.

In this paper, a trajectory-based approach to grasp interaction is presented. The approach is based on the research result of Neurophysiology and enables a user to grasp and manipulate an object naturally with a dexterous virtual hand. Unlike some traditional methods which totally rely on computer performance to alleviate visually distracting artifacts, our approach can control artifacts by determining the proper number of sampling points on the grasp trajectories. Moreover, automatic contact estimation conditions, grasp identification conditions, and the grasp configuration of the virtual hand can be determined rapidly utilizing the grasp trajectories, which will inevitably save time for more realistic finger feedback force computation, given that the requirement of update rate in haptic rendering is up to 1 KHz.

Future works will include: 1) to propose more reasonable conditions for grasp identification and 2) to provide more realistic force computation model.

References

1. S. Redon, A. Kheddar and S. Coquillart. CONTACT: arbitrary in-between motions for continuous collision detection. In Proceedings of IEEE ROMAN'2001, Sep. 2001.
2. S. Redon, A. Kheddar and S. Coquillart. Fast Continuous Collision Detection between Rigid Bodies. In proceedings of Eurographics 2002. September 2002
3. McNeely W, Puterbaugh K, Troy J. Six Degree-of-freedom Haptic Rendering Using Voxel Sampling. In Proceedings of Siggraph 1999, LosAngeles, CA
4. Palmer I, Grimsdale R. Real-time collision detection for animation using Sphere-Trees. Computer Graphics Forum 1995, 14(2):105-116
5. Zachmann G. Optimizing the Collision Detection Pipeline, In Proceedings of the First International Game Technology Conference(GTEC), Hong Kong, 18-21 January 2001
6. Gottschalk S, Lin M, Manocha D. OBB-Tree: A Hierarchical Structure for Rapid Interference Detection, the Proceedings of ACM SIGGRAPH'96, 1996:171-180
7. Ehmann S, Lin MC. Accurate and Fast Proximity Queries between Polyhedra using Convex Surface Decomposition. In Proceedings of the Eurographics Conference, Manchester, 2001:500-510
8. S. Rezzonico, Z. Huang, R. Boulic, N. Magnenat Thalmann, D. Thalmann, Consistent Grasping Interactions with Virtual Actors Based on the Multi-sensor Hand Model, Proc. 2nd Eurographics workshop on Virtual Environments, Monte Carlo.
9. G.Zachmann and A. Rettig, "Natural and Robust Interaction in Virtual Assembly Simulation," presented at Eighth ISPE International Conference on Concurrent Engineering: Research and Application, Anaheim, 2001
10. K. Hirota and M. Hirose, Dexterous Object Manipulation Based On Collision Response. Presented at IEEE Virtual Reality, Los Angeles, 2003
11. W.S.Harwin and N. Melder. Improved Haptic Rendering for Mult-Finger Manipulation Using Friction Cone based God-Objects. Proceedings of Eurohaptics Conference, 2002.
12. N.Melder, W.S.Harwin, P.M.Sharkey. Translation and Rotation of Multi-Point Contacted Virtual Objects. Proceedings of Eurohaptics 2003 pp 218-227
13. N.Melder and W.S.Harwin. Extending the Friction Cone Algorithm for Arbitrary Polygon Based Haptic Objects. Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, Chicago, 2004.
14. Christoph W. Borst and Arun P. Indugula. Realistic Virtual Grasping. Presened at IEEE Virtual Reality 2005 pp. 91-98.

15. H. Iwata. Artificial Reality with Force-Feedback: Development of Desktop Virtual Space with Compact Master Manipulator. Computer Graphics, vol. 24, pp. 165-170, 1990.
16. Maekawa, H. and JM Hollerbach. Haptic Display for Object Grasping and Manipulating in Virtual Environment. Proc. of Int. Conf. on Robotics and Automation, pp. 2566-2573, 1998
17. Justus H. Piater. Learning Visual Features to Recommend Grasp Configurations. CMPSCI Technical Report 2000-40. July 2000
18. Costas S. Tzafestas. Whole-Hand Kinesthetic Feedback and Haptic Perception in Dexterous Virtual Manipulation. IEEE Trans. on Sys. Man and Cybernatics, 33(1):100-113, January 2003
19. D. G. Kamper, E. G. Cruz and M. P. Siegel. Stereotypical Fingertip Trajectories During Grasp. Journal of Neurophysiology 90: 3702-3710, 2003
20. T.Whitted, An improved Illumination Model for Shaded Display, Comm ACM Vol.32, No. 6, 1980
21. Möller T. Trumbore B. Fast, Minumum Storage Ray-Triangle Intersection Journal of Graphics Tools, 1997, 2(1):21-28
22. Jeffrey Mahovsky, Brian Wyvill, Fast ray-axis aligned bounding box overlap tests with Plücker coordinates. Jour-nal of Graphics Tools: JGT, 2004, 9(1):35-46
23. Hand SDK, last visit: Aug 20, 2005, http://www.immersion.com
24. Z. Zhu, S. Gao, H. Wan, Y. Luo and W. Yang. Grasp Identification and Multi-Finger Haptic Feedback for Virtual Assembly. In Proc. Of DETC'04 Salt Lake City, Utah USA, 2004
25. CAVELib Manual, last visit: Aug 20, 2005, http://www.vrco.com/CAVE_USER