Multi-Contact Grasp Interaction for Virtual Environments

Daniel Holz  Sebastian Ullrich  Marc Wolter  Torsten Kuhlen

Virtual Reality Group, RWTH Aachen University

Abstract: The grasping of virtual objects has been an active research field for several years. Solutions providing realistic grasping rely on special hardware or require time-consuming parameterizations. Therefore, we introduce a flexible grasping algorithm enabling grasping without computational complex physics. Objects can be grasped and manipulated with multiple fingers. In addition, multiple objects can be manipulated simultaneously with our approach. Through the usage of contact sensors the technique is easily configurable and versatile enough to be used in different scenarios.

Keywords: Grasping, Interaction, Virtual Reality, Multi-Finger Manipulation

1 Introduction

Grasping is one of the most frequent actions in everyday life. Therefore, interactive grasping in virtual environments has been investigated in different areas, ranging from fitting simulations, engineering and construction processes to psychological studies. Especially in psychological studies, the duplication of real world behavior is often favored over faster interaction methods to investigate human behavior or motor control.

Early grasping methods in virtual environments relied on gesture-based grasping: Objects are automatically selected when the user carries out a specific hand gesture. Grasp detection based on grasp taxonomy and grasp models consider human behavior for a more physiologically realistic grasping of objects. By contrast, physical-based manipulation techniques focus on the physics of grasping rather than the human motor activity. Another field of related techniques deals with automatic grasping, for instance in robotics. These techniques examine and apply motor planning for the imitation of prehension motions.

In this work, we confine ourselves to viewing grasping as a natural interaction technique to manipulate virtual objects. While we use grasping in everyday life as a matter of course, computer-based grasping is very complex. The missing information about the friction of the surface or the weight of the object, as well as the possible “sinking” of the real hand into virtual objects constrict intuitive grasp control. Modern grasp systems for virtual environments integrate collision detection, physics, and haptics to deal with these problems. Yet, the configuration of input devices (e.g., data gloves or force-feedback gloves) for the user and the definition of the virtual objects for a realistic grasping are time-consuming processes.

Therefore, we present a very intuitive grasp-based interaction technique which trades precision and physical correctness for the sake of an easy and expandable application to
different problems. The method is intended for use in different projects. Therefore the main objectives are flexibility and easy usability.

To achieve this goal, we apply a simple grasp condition in combination with an arbitrary number of contact sensors. Manipulation of grasped objects is simulated for each object independently, which allows multi-finger manipulation as well as multi-object manipulation. Some special cases, like grasping objects with rough surfaces or sharp edges, are supported. Transformation of each grasped object is evaluated for the visually best result based on the contact information of all colliding sensors to provide visually feasible manipulation. Since we do not require a specific number or positioning of the contact sensors, the method can be used independently of both the applied hand model and the input device used to control the virtual hand geometry or another virtual gripping device.

First, we give a short overview of related work in section 2. Section 3 describes the proposed approach in detail, including the grasp condition and the simulation used for object manipulation. We present and discuss the achieved results in section 4. An outlook on future work concludes the paper.

2 Related Work

We will briefly discuss several works in the field of interactive grasping. The works were selected to show the variety of research efforts in this area. However, neither gesture-based interaction nor grasp planning or automatic grasping will be discussed here.

Kahlesz et al. proposed a calibration technique aimed at high visual fidelity instead of accuracy [KZK04]. They explicitly addressed the problem of cross-coupling sensors, especially abduction sensors, of modern data gloves. Their calibration method can be carried out without additional hardware but achieves visually correct results. As in this work, the authors also favored visual correctness over realism.

Boulic et al. [BRT96] introduced a virtual contact model to describe interactions of hand sensors and objects. To avoid penetration of the hand into the virtual object, an unfolding routine was applied. It corrects the hand posture one joint at a time until all penetrations are removed. In [KH96], a kinematic method based on representative spherical planes for manual object manipulation without force-feedback was presented. Using this method, manipulation with either two or three fingertips was realized. The authors combined this manipulation method with a simple dynamics method for non-grasp object manipulation.

[HH03] applied physics to manipulate objects, but instead of using a restricted set of sensors, they used a large number of interaction points densely distributed over the hand. In order to efficiently compute the object interaction with all interaction points, an original collision detection and stable manipulation implementation was integrated. Borst and Indugula [BI05] recently introduced a physical-based approach to grasping and manipulation of virtual objects. They applied freely available toolkits for physics and collision detection to realize a grasping technique independent of any grasp model. According to the authors, their approach can be used together with a haptic rendering system.
All these approaches show promising results. Yet, there are always restrictions. Kijima et al. restrict themselves to fingertips only, Boulic et al. allow only grasps with one finger being the thumb and disallow global movement of the hand. Furthermore, the interpretation of the hand model as a state machine does not allow the grasping of multiple objects. The technique of Borst et al. relies on a spring-mass model which must be fine-tuned for each different sized/shaped hand. Hirota and Hirose use a large set of approximately 1200 interface points, which requires an expensive computation. Also, their technique has problems handling objects that have sharp edges or that are wedge-shaped.

The approach presented in this paper tries to resolve some of the above-mentioned limitations. In contrast to most mentioned works, we do not intend to simulate completely realistic grasping. Our method aims at easy and fast usability for the developer and user.

3 Method

In the design process, we tried to widen the range of application by avoiding constraints whenever possible. Since the main objective was to allow the usage of the system in existent projects, we wanted to ensure, first of all, that arbitrary shaped rigid objects can be grasped. Second of all, the grasping system needed to be independent of both the geometric properties of the virtual gripping device and its hierarchical structure. The virtual gripping device could be an articulated human hand or a robotic grasping tool. Hence, we decided to use primitive spheres as sensors which can easily be integrated into any kind of structure. This also allows us to provide multi-finger manipulation, which enhances interaction in virtual environments.

The method’s workflow is as follows: every movement of the user’s hand is applied to a virtual hand model via some hardware device (e.g., a data glove). The sensors are attached to the skeletal hierarchy of the virtual hand and are transformed accordingly. The method is not restricted to this application, since the sensors can be distributed freely throughout all kinds of geometric structures, like a pincer, chopsticks or robotic grasping tools. For each grasped object, a simulation is computed. The simulated results are evaluated for a visually optimal transformation, which is then applied to the corresponding grasped object.

A pseudocode of the complete grasping algorithm is depicted in figure 1. We decided to treat each grasped object independently throughout the calculation of the object manipulation. This allows even grasping and manipulation of multiple targets at the same time.

All stages, except for the simulation, which is presented in section 3.3, are rather straightforward and need no further attention.

3.1 Sensor model

Our method is based on the concept of using a set of sensor elements for grasping decision rather than integrating the often complex geometry of the articulated gripping device into the calculation process. An arbitrary number of these sensors, variable in size, can be attached to the device, so that the device’s special grasping capabilities can be either almost perfectly
simulated (by using a lot of sensors) or optimized for the application purpose (by using sensors at selected positions). Consequently, the sensor model allows a high precision without the burden of long calculation times in the collision detection, which plays a significant role in our method. As stated in [BRT96], sphere sensors are very efficient for grasping problems - a result which motivated us to use primitive spheres as sensor elements in our approach as well.

3.2 Grasp pairs

We define the grasp of a virtual object by a condition which is based on the description of a stable grasp in [MI94]. A stable grasp of an object is accomplished if the forces applied to the object from two directions are equal in magnitude and if the forces’ direction vectors are collinear. Furthermore, the angle between each direction vector and the surface normal of the object in the corresponding area of contact may not exceed a particular value. This value is determined by the friction, which is induced both by the roughness of the gripping device and the roughness of the object. We transferred this physical model to our sensor approach by omitting the force constraints. We did not consider any input or output of force, since this would yield a hardware dependency of the technique. Thus, in the case of any two sensors being in contact with the same object, we assume the forces being equal in magnitude and collinear in direction as claimed above. In other words, we assume those sensors to be antagonistic regarding force and force direction. We then verify the contact angles in respect of the above-mentioned condition. In case of a collision between a sensor sphere and the surface of an object we derive the normal of the object’s polygon which is in contact with the sensor. We call this vector contact normal $n$. Then two sensors induce a stable grasp if for each sensor $s_i$ the angle between the line that connects the sensors and the contact normal $n_i$ does not exceed a particular value, i.e., the angle is inside the so called cone of friction (figure 2). We define this pair of sensors as a grasp pair. Figure 2 shows a situation in which both $(s_1, s_3)$ and $(s_1, s_2)$ form a grasp pair and hence induce a stable grasp of the object.

Figure 1: Pseudocode of the grasping algorithm. Each object is treated independently. The simulation consists of several iterations until all grasp pairs in $G$ are valid.
Figure 2: Contact model for three sensors \((s_1, s_2, s_3)\) with an object. Position \(c\) denotes the center of the grasp. The vectors \(n_1, n_2, n_3\) are the corresponding contact normals.

Figure 3: Treatment of special cases. Top: Grasping of sharp objects due to the average direction vector. Bottom: Enlarging the cone of friction for stable grasps of rough objects.

It is obvious that this technique is not sufficient when it comes to objects with rough surfaces. In the case of a sensor being in contact with a sharp corner, bringing any of the surface normals to the computation would lead to erroneous results (cf. \(n_a\) and \(n_b\), top of figure 3). We took this into account by calculating the average direction vector of the surrounding surfaces’ normals. This vector serves as contact normal in the calculation (cf. \(n_1\), top of figure 3). Consequently, the grasp of sharp-edged objects is possible without any difficulty. Additionally, the cone of friction gives us the possibility to adjust the algorithm according to the roughness of the object that is to be grasped. For very rough objects it is sufficient to enlarge the cone of friction as shown in figure 3 (bottom). Here, the original cone (dotted area) is not large enough to allow the detection of the grasp pair \((s_1, s_2)\) in contrast to the enlarged one (striped area). It makes sense to adapt the cone of friction to the roughness of the object which is to be grasped, since rougher objects induce more friction when being in contact with the gripping device. Generally, we tried to assure that every kind of geometry is manipulable to allow intuitive interaction with the scene. The adjustment of the cone of friction is a powerful tool in this matter. We simply enlarge it for objects which are hard to grab or if we want to simplify interactions with objects in a physically unrealistic way, which is often intended by some kinds of applications. Later in section 4, we present results for grasping both rough and sharp objects.

It is of particular importance that we do not assign specific roles to sensors. A grasp pair can emerge from any pair of sensors and all of them are handled in exactly the same way throughout the procedure. Hence, there is no restriction whatsoever concerning the kind of gripping device that is used. This is one of the main objectives in our approach.
Figure 4: Object manipulation due to sensor transformation. On the update of the hand model, the sensors move accordingly from time $k$ to a new position in time $k+1$ (left). This yields the object’s translation $v_t$ and the object’s rotation $q_{mean}$ (right).

Figure 5: Schematic view of one simulation step. Left: A preliminary object transformation leads to an invalid grasp setup. The weakest grasp pair is excluded from the simulation. Right: After exclusion, the new transformation is computed, which is a valid grasp situation.

### 3.3 Grasp simulation

Once grasped, an object has to be transformed according to the user’s manipulation. Let $n$ be the number of grasp pairs in contact with an object. In the case of $n = 1$, i.e. there is exactly one grasp pair, the manipulation is straightforward. A grasp pair consisting of two sensors induces a vector that lies between the centers of those sensors. We call this vector grasp axis $v$. The grasp axis $v_k$, where $k$ corresponds to the current frame, changes over time according to the motion of the gripping device which changes the positions of the sensors.

The quaternion $q_r$, which rotates the axis $v_k$ onto the axis $v_{k+1}$, corresponds to the rotation of the grasped object from frame $k$ to $k+1$. Furthermore, the repositioning of the sensors yields the object translation over this period of time. For translation, we apply the vector that points from the center of the grasp pair in frame $k$ to the appropriate center in frame $k+1$. The performed manipulation results from the mentioned translation followed by the rotation $q_r$ around the pair’s center in frame $k+1$. 

A rotation around the grasp axis itself is not detected by this technique. Such a situation occurs, for instance, if there is no translation of the two sensors of a grasp pair from frame $k$ to $k + 1$, but a rotation of the sensors around the grasp axis. In this case the vectors $v_k$ and $v_{k+1}$ have the same direction. Hence, the method above would result in the identity quaternion which yields no rotation at all. In the general case, we solve this problem by determining the sensors’ rotation angles around the grasp axis and creating a quaternion $q'$ which corresponds to a rotation of the mean value of these angles around the axis $v_{k+1}$. We use the fact that quaternions allow the concatenation of rotations by left-sided multiplication to add the rotation $q'$ to $q_r$. Consequently, the product $q'q_r$ matches the complete rotation and replaces $q_r$ in the transformation procedure.

In the case of $n > 1$, where more than one grasp pair fulfills the grasp condition for a single object, a more complex heuristic is necessary. Let’s assume a grasp is induced by the set of grasp pairs $G = \{g_1, \ldots, g_n\}$ in frame $k$. After a repositioning of the sensors, i.e., a movement of the hand, it is not clear which sensors are actually involved in the object manipulation the user has performed.

Obviously, particular grasp pairs from the $k$-th frame are not valid in the $k + 1$-th frame anymore, which means that they definitely do not take part in the object manipulation during this time. This certainly holds in the cases of those grasp pairs where the distance between the sensors increases in such a way that a stable grasp is obviously not possible anymore in the $k + 1$-th frame. Those grasp pairs are to be excluded from the remaining process. We define the set of remaining grasp pairs as $G'$.

Next, we simulate a manipulation based on the assumption that the object is intentionally manipulated by the entire set $G'$. This assumption holds as long as all the grasp pairs move (or are moved) as a whole. In other words, the assumption is incorrect if there are at least
two grasp pairs that are counteracting, i.e. being rotated or translated in opposed directions. This case occurs, for instance, if some grasp pair which took part in a manipulation up to frame $k$ does not take part in the manipulation anymore when frame $k+1$ is reached. For this purpose we assign to each grasp pair a value of the force which it exerts on the grasped object. This force coefficient gives higher priority to particular grasp pairs concerning the object manipulation and thus allows the system to decide which grasp pairs induce the correct one. The less force is applied to an object by a grasp pair the lower is its influence on the object manipulation. Consequently, the system gradually excludes those pairs that apply the weakest force in order to find the set of grasp pairs which induces the correct manipulation. We determine the force coefficient with regard to the areas of contact between both sensors of a grasp pair and the grasped object. The smaller the contact angles are (cf. ‘cone of friction’ in section 3.2), the higher is the force applied to the object and so is the force coefficient.

The workflow of the algorithm for each graspable object is depicted in figure 6. In the simulation we first calculate the transformation of the object, which is induced by the alteration of the positions and orientations of all the grasp pairs $g \in G'$. In order to achieve this, we obtain the translation of the object similar to the case of $n = 1$. We know that the object is manipulated according to the center of the grasp, i.e. the center of the cloud of those sensors which belong to some grasp pair in $G'$. As before, we derive the desired translation $v_t$ by simply comparing the positions of the cloud’s center in the current and the previous frame ($c'$ and $c$ respectively in figure 4). For the computation of the object rotation we apply quaternion mathematics for the description of rotations and orientations in the 3-dimensional Euclidean space. The aim is to obtain the mean rotation $q_{mean}$ induced by all the individual rotations of the grasp pairs in $G'$, each represented by a quaternion as described above. As depicted in figure 7, we use a mapping from the 4-dimensional hypersphere of unit quaternions onto the tangent plane at the identity quaternion as presented in [Cho06] and thus receive one point on the plane for each quaternion. The quaternion of the mean rotation corresponds to the center of the obtained points mapped back to the hypersphere. Now we transform the object in the collision detection as we did in the case $n = 1$ and check whether all the grasp pairs are still valid. A grasp pair is valid if there is an appropriate collision for each of the sensors and the grasp pair still fulfills the grasp condition. If so, the simulated manipulation is assumed to be the correct one and we apply the results to the real scene. In the case that at least one grasp pair does not fulfill the required grasp condition anymore, as shown on the left side of figure 5, we assume that the grasp pair $g'$, which exerts the smallest force to the object, i.e. the one with the smallest force coefficient, is of no influence for the object manipulation. Hence, we exclude it from the simulation and define $G'' := G'\{g'\}$. The simulation sequence is repeated with the new set of grasp pairs $G''$ until a valid manipulation is obtained (cf. right side of figure 5).
4 Results and Discussion

In our implementation we used an H-Anim 200x compliant VRML-model [H-A05] of a human hand and integrated sites as markings for the sensors’ positions. Four sensors were distributed over each finger and six sensors were attached to the palm. Currently, the system supports two types of data gloves: an Immersion CyberGlove with 18 sensors and a 14 sensors 5DT DataGlove for fMRI studies. As the 5DT glove exhibits strong cross-sensor conflicts, we decided to do all tests with the CyberGlove device. For collision detection, we integrated the SOLID collision detection library.

We evaluated the proposed method with different objects and present results for a small representative selection, namely a primitive sphere, the well-known Stanford Bunny, and a pencil. Figure 8 shows some grasps and manipulations of the objects mentioned above, whereas figure 9 shows the manipulation of multiple objects at the same time. In order to demonstrate the system’s ability to grab even objects that have a rather rough surface, we added the model of a crumpled sheet of paper and initialized it with a larger cone of

Figure 8: Different grasps and manipulations of the Stanford Bunny and a pencil.

Figure 9: Manipulation of multiple grasped objects with a single hand.
friction (see figure 10, left). Furthermore, we present the method’s ability to enable grasps of sharp objects like a cone (see figure 10, right). In several trials we experienced the grasping behavior as being equally good for all models. Consequently, there was no need for any kind of additional tuning apart from adjusting the cone of friction.

Concerning runtime, we measured the simulation time while grasping one object. We used a PC with a 3 GHz Pentium 4 processor and an NVIDIA GeForce 6600 GT graphics card. The results for the cone object are depicted in figure 11. A total runtime of approximately 60 seconds (which corresponds to 4400 frames with an average framerate of 73 fps) was recorded. Figure 11 shows the simulation time and the transformation time, which is the simulation time without collision detection, in milliseconds and the corresponding number of necessary simulation steps to achieve a valid grasp. When no contact sensors touched the object, the simulation time was approximately zero. The segments marked as A and C belong to a multi-finger grasp (three to five fingers manipulating the object which corresponds to a maximum of 20 sensors in contact with the object). In the segment B the test person grasped the cone with two fingers only. The average simulation time is below 4 ms, with an average of two simulation steps necessary. One can see that the main part of the simulation time is spent on the collision detection.

Compared to the related work our technique is less complex and easier to implement. The combination of independent sensors and arbitrary grasp pairs allows intuitive interactions and multiple-object manipulation in addition to that. One issue that has not been addressed is the handling of visual interpenetration of hand model and grasped object. Further improvement has been achieved by the ability to grasp sharp-edged objects and rough surfaces.

5 Future Work and Conclusion

Feedback on the current grasp is crucial for manual interaction with objects. Even though we do not support haptic rendering, possible solutions are correction of the finger joints
as described in [BRT96] or corrected grasp postures with visual feedback [US00]. In addition, the integration of a physics system would allow finger manipulation that goes beyond grasping, such as pushing objects.

We showed that multiple objects are manipulable at the same time. Anyway, it is not yet possible to manipulate several objects which are in contact with each other as a whole. An idea to address this problem is to generate a virtual sensor for each of the contact points between the objects and to simply include those in the calculation process.

We proposed a new method to enable the visually feasible grasping of multiple objects in virtual environments. The method focuses on flexibility and is therefore applicable as a general interaction technique in existing projects. This is achieved by an arbitrary configuration of contact sensors and by being hardware-independent, together with a large class of graspable objects. The only changeable parameter is the applied cone of friction. We evaluated the proposed methods with several objects ranging from simple over complex to sharp and rough objects.
References


