Multi-Contact Grasp Interaction for Virtual Environments

Daniel Holz , Sebastian Ullrich , Marc Wolter , Torsten Kuhlen

Virtual Reality Group
RWTH Aachen University
Seffenter Weg 23, 52074 Aachen, Germany
www: www.rz.rwth-aachen.de/vr

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 a wide variety of projects. Hence,

First presented at the 4th Workshop "Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR", extended and revised for JVRB

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.

In [US00], Ullmann and Sauer presented a method for one-hand and two-hand grips with focus on the realization of realistic grasping gestures. In the process a set of so called collision boxes were used which had roles assigned to them (palm, thumb, finger). Several gestures, based on combinations of the above mentioned roles, were supported which aimed on the natural grasping behavior of humans.

The method described in [ZR01] focuses on grasping for virtual assembly tasks. An implementation of a fast collision detection was used to detect a set of pre-defined grasps aiming on the most common interactions in assembly scenarios. The authors proposed a method to prevent object penetrations while still allowing the object to move in a physically plausible way.

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. Also Ullmann et al. make restrictions concerning the supported types of grasps, which can be used exclusively with a model of the human hand. The method of Zachmann et al. requires information regarding the hierarchical structure of the virtual hand, and not all kinds of grasps, such as the cigarette grasp[1], are supported. 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

[1] A precision grasp performed with two adjacent fingers.

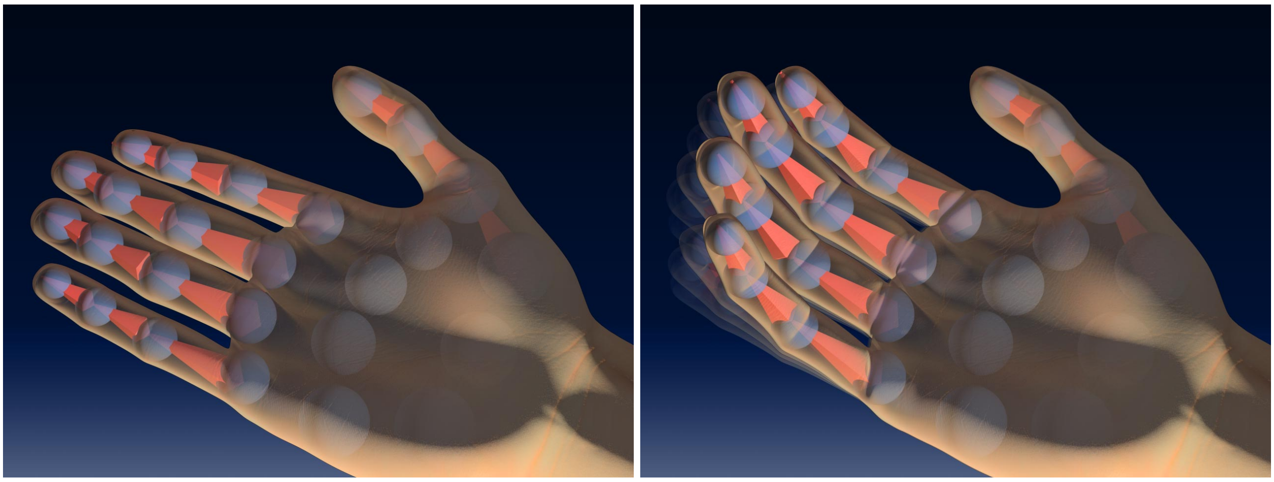

Figure 1: Example of a *sphere sensors / gripping device* configuration. The sensors are distributed over an articulated hand model and attached to a hierarchy of bones which controls the hand deformation.

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 2. 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 ar-

```
Update virtual gripping device
Update contact sensors
for each graspable object do
    G = object's grasp pairs
    do
        Simulate transformation (G)
        valid = Evaluate grasp (G)
        if (not valid) then
            Determine weakest grasp pair g
            G = G\{g}
            Reset transformation
        end
    until valid
```

Figure 2: Pseudocode of the grasping algorithm. Each object is treated independently. The simulation consists of several iterations until all grasp pairs in G are valid.

bitrary number of these sensors, variable in size, can be attached to the device, so that the device-specific 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.

Figure 1 shows a possible *sphere sensors / gripping device* configuration. A set of sphere sensors (illustrated in light blue) is distributed over an articulated hand model. The hierarchy of bones which controls the hand movement is depicted in red. Since being hooked into the hierarchy, the sensors are transformed according to the deformation of the hand model, as shown on the right side of figure 1. Here, one can see that by arbitrarily positioning the variable sized sensors relative to their parent bones an accurate geometric representation of the gripping device can be achieved.

One way to link the sphere sensors to a bone hierachy which controls a virtual gripping device is to use the scene graph, wherein the hierarchy is represented by a set of chained group nodes corresponding to bones or joints. In such a scenario, a sphere sensor could simply be attached to the group node of the bone (or joint) which is to control the sensor's transformation. Thus, transformation changes of the bone (or joint) nodes do not only affect the 3D model of the gripping device but also the sphere sensors, keeping the sensors and the gripping device synchronized.

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 transfer this physical model to our sensor approach by omitting the force constraints. We do 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* (cf. figure 3). We define this pair of sensors as a *grasp pair*. Figure 3 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. Formally, two sensors (s_i, s_j) form a grasp pair in frame k iff the conditions below are met:

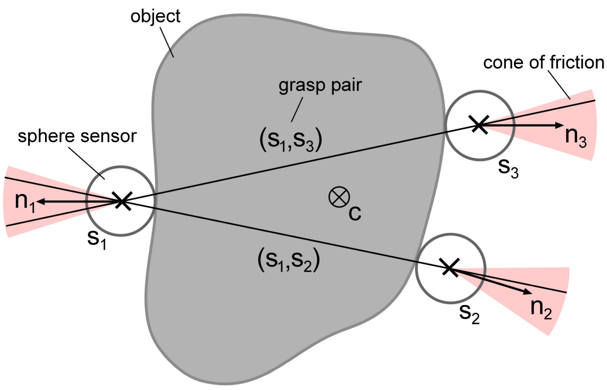

Figure 3: Contact model for three sensors (s_1,s_2,s_3) with an object. Position c denotes the center of the grasp. The vectors n_1,n_2,n_3 are the corresponding contact normals.

- Both s_i and s_j collide with the same object in frame k.
- In frame k the following two equations hold:

$$\angle(n_i, v_{ji}) \leq \alpha_{max} \tag{1}$$

$$\angle(n_j, v_{ij}) \leq \alpha_{max} \tag{2}$$

Here, v_{ij} is the vector from the center of sensor s_i to the center of sensor s_j, and α_{max} is the angle which defines the above mentioned cone of friction. In fact, the opening angle of the cone equals $2 \times \alpha_{max}$.

However, the presented 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 normals of the surrounding polygons to the computation as described above, would lead to erroneous results. The top of figure 4 shows such a situation. Here, two sensors are in contact with a sharp object. To test if s_1 and s_2 form a grasp pair we need to include a contact normal in the decision process for each sensor. For s_2 we take the surface normal n_2. Using either n_a or n_b as contact normal for s_1 would result in (s_1,s_2) not being identified as a grasp pair. The angle between the line connecting the sensors and either n_a or n_b simply exceeds the maximum value allowed (cf. cone of friction of s_1). As a solution to this problem we calculate the average direction vector of the normals of those polygons which surround the corner (here: n_a and n_b). This vector then serves as contact normal in the decision process (cf. n_1, top of figure 4). Consequently, the grasp of sharp-edged objects is possible without any difficulty.

Additionally, the cone of friction gives us the possibility to further adjust the algorithm according to the roughness of the object which is to be grasped. For very rough objects it is sufficient to enlarge the cone of friction as shown at the bottom of figure 4. 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). In general, 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.

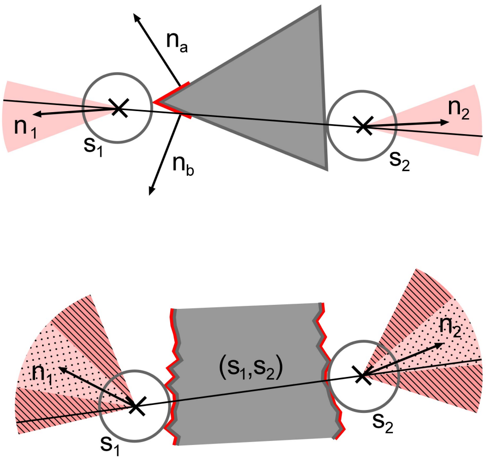

Figure 4: Treatment of special cases. Top: Grasping of sharp objects due to the average direction vector. Bottom: Enlarging the cone of friction for stable grasps of rough objects.

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 intended in certain scenarios. 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. This means that 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 which is used. In contrast, the method presented in [US00] exclusively applies to a model of the human hand by assigning the roles “thumb”, “palm” and “finger” to the sensors. Furthermore, only combinations of “thumb” and “finger” or “palm” and “finger” sensors can lead to a stable grasp performed with one hand. The avoidance of the use of such constraints is one of the main objectives in our approach.

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 s_i, s_j induces a vector that lies between the centers of those sensors. We call this vector *grasp axis* v_{ij}^k, where k corresponds to the current frame. The grasp axis 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_{ij}^k onto the axis v_{ij}^{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 (cf. figure 5). For translation, we apply the vector that points from the center of the grasp pair in frame k to the corresponding

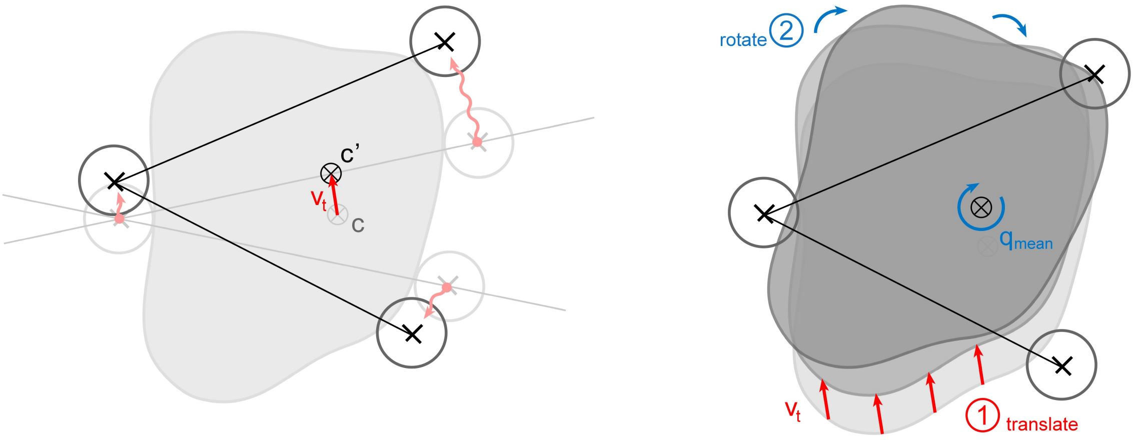

Figure 5: Object manipulation due to sensor transformation. On the update of the virtual gripping device, the sensors move accordingly from frame k to a new position in frame $k+1$ (left). This yields the object's translation v_t and the object's rotation q_{mean} (right).

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_{ij}^k and v_{ij}^{k+1} have the same direction. Hence, the method above would result in q_r being 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_{ij}^{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'.

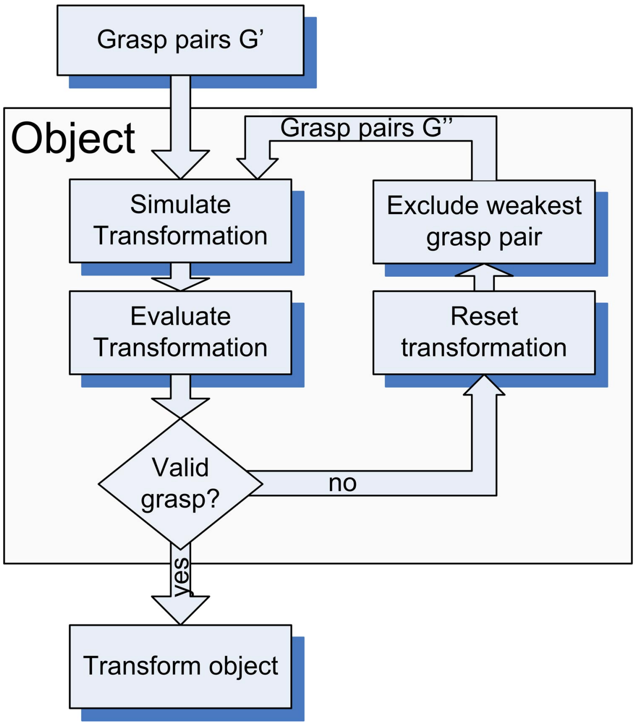

Figure 6: Loop of the simulation for one object.

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.

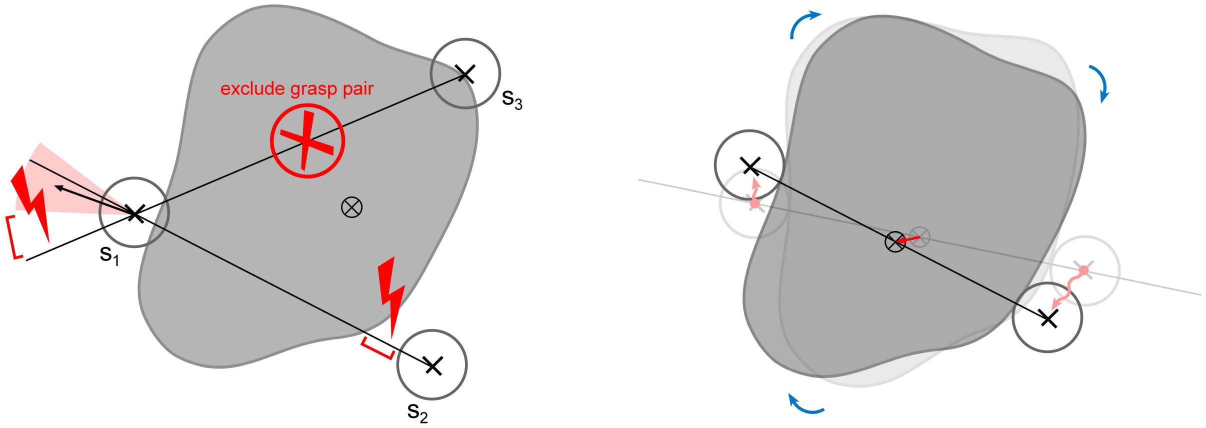

Figure 7: Schematic view of one simulation step. Left: A preliminary object transformation leads to an invalid grasp setup. The weakest grasp pair is excluded from the simulation. Right: After exclusion, the new transformation is computed, which is a valid grasp situation.

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* ϕ 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. The *weakest grasp pair* is defined as the grasp pair which has the lowest force coefficient. 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 force coefficient ϕ of a given grasp pair (s_i, s_j) can be computed by using the following equation:

$$\phi = 1 - \frac{\angle(n_i, v_{ji}) + \angle(n_j, v_{ij})}{2 \times \alpha_{max}} \tag{3}$$

As before, the vector n_i corresponds to the contact normal of sensor s_i, and v_{ij} is the vector from the center of sensor s_i to the center of sensor s_j. From the equation above and from the equations 1 and 2 presented in section 3.2 it follows that $0 \leq \phi \leq 1$, where 1 corresponds to the maximum and 0 to the minimum force induced by a grasp pair.

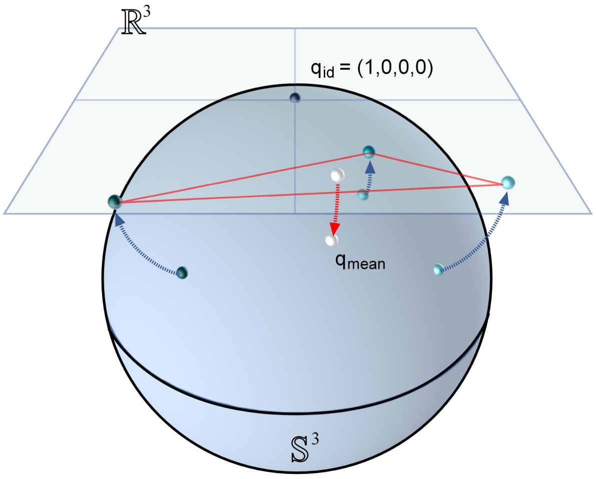

Figure 8: Determination of the mean rotation by quaternion mapping.

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 (cf. c in figure 3), 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

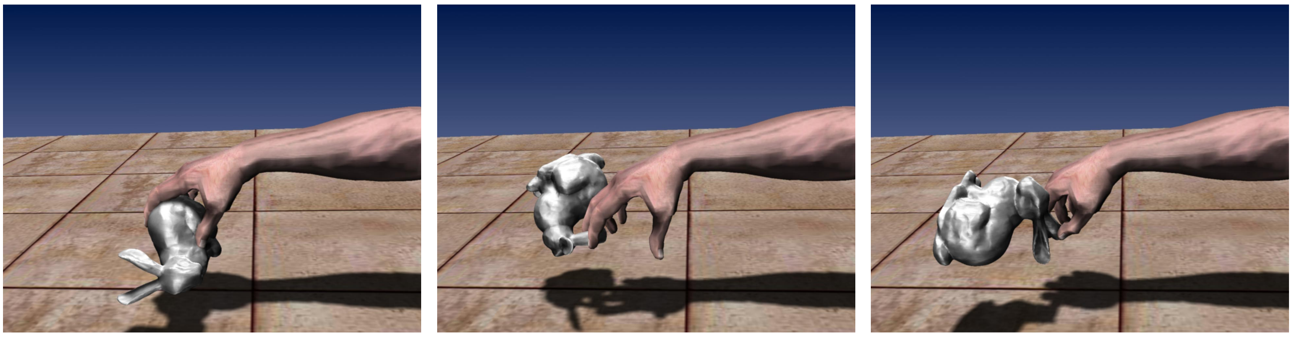

Figure 9: Different grasps and manipulations of the Stanford Bunny.

current and the previous frame (c' and c respectively in figure 5). 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 8, 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 7, we assume that the weakest grasp pair g', i.e. the grasp pair which exerts the smallest force to the object, is of no influence for the object manipulation. Hence, we exclude it from the simulation and define $G'' := G' \backslash \{g'\}$. The simulation sequence is repeated with the new set of grasp pairs G'' until a valid manipulation is obtained (cf. right side of figure 7). Due to the way the object rotation is computed (see above: mean rotation, defined by q_r) the manipulation of an object is performed in consideration of the whole set of grasp pairs. This means that, for example in the case of 3 touching sensors forming 2 grasp pairs (which could be a 3-finger manipulation), the movement of each of these sensors plays a role in the object transformation, a fact which allows for precise multi-contact manipulation of the respective object.

4 Results and Discussion

In our implementation we used an H-ANIM 200x compliant VRML-model [Gro05] 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 IMMERSION 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. Figures 9 and 11 show some grasps and manipulations of the objects mentioned above, whereas figure 12 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

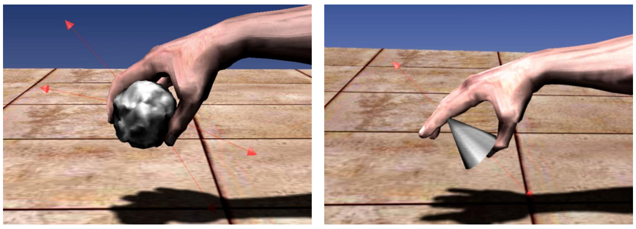

Figure 10: Examples for grasping an object with rough surface (left) and a sharp-edged object (right). The small arrows indicate the contact normals of the grasp pairs.

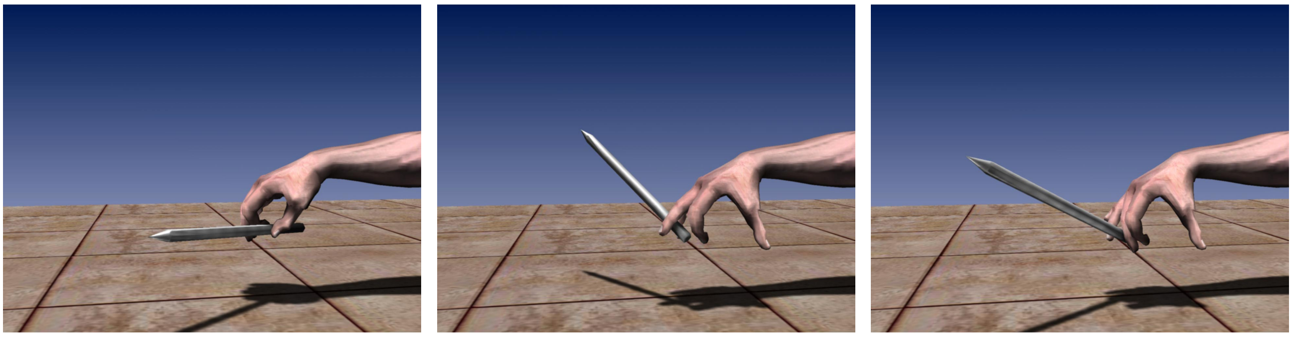

Figure 11: Different grasps and manipulations of a pencil.

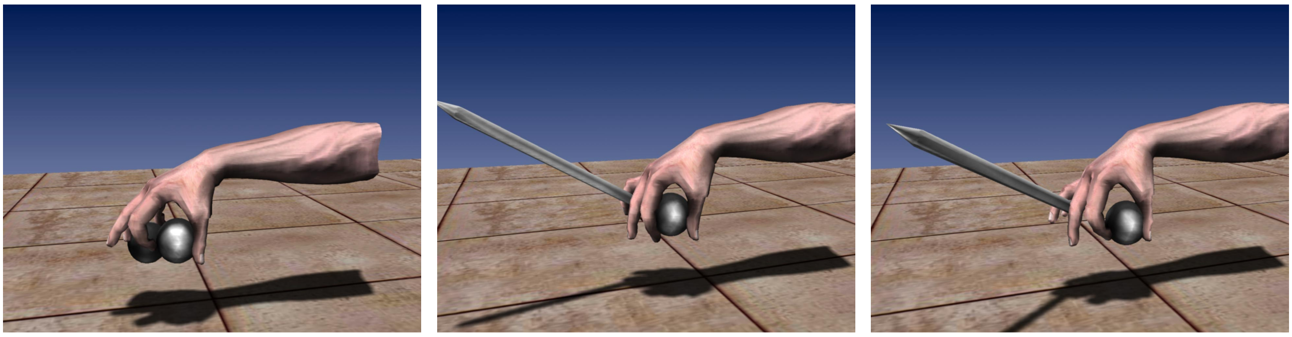

Figure 12: Manipulation of multiple grasped objects with a single hand.

of a crumpled sheet of paper and initialized it with a larger cone of 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, which corresponds to the time consumption of the algorithm shown in figure 6, 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 13. A total runtime of approximately 60 seconds (which corresponds to 4400 frames with an average framerate of 73 fps) was recorded. Figure 13 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, as expected. 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 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 was spent on the collision detection. In fact, the time consumption of the complete grasp heuristic (cf. transformation time in figure 13), which includes the determination of the grasp pairs, the computation of the object manipulation and its verification, is smaller than the time consumed by the collision detection by a factor of about 100 (cf. simulation time in figure 13). Consequently, improving the collision detection would have a great impact on the overall computation time.

Compared to the related work our technique is less complex and easier to implement. The implementation was done by an undergraduate student. As shown in figure 13 the overhead runtime cost to an existing collision detection is small. In contrast to physics-based approaches, as the one presented in [BI05], the cost of parameterizing is inconsiderable, since the only parameters of our technique are the sensor positions as well as a single cone of friction value for each graspable object. We do not restrict the positioning of the sensors in the proposed *spheres sensors / gripping device* configuration, which enables usage of any kind of articulated geometry with an arbitrary hierar-

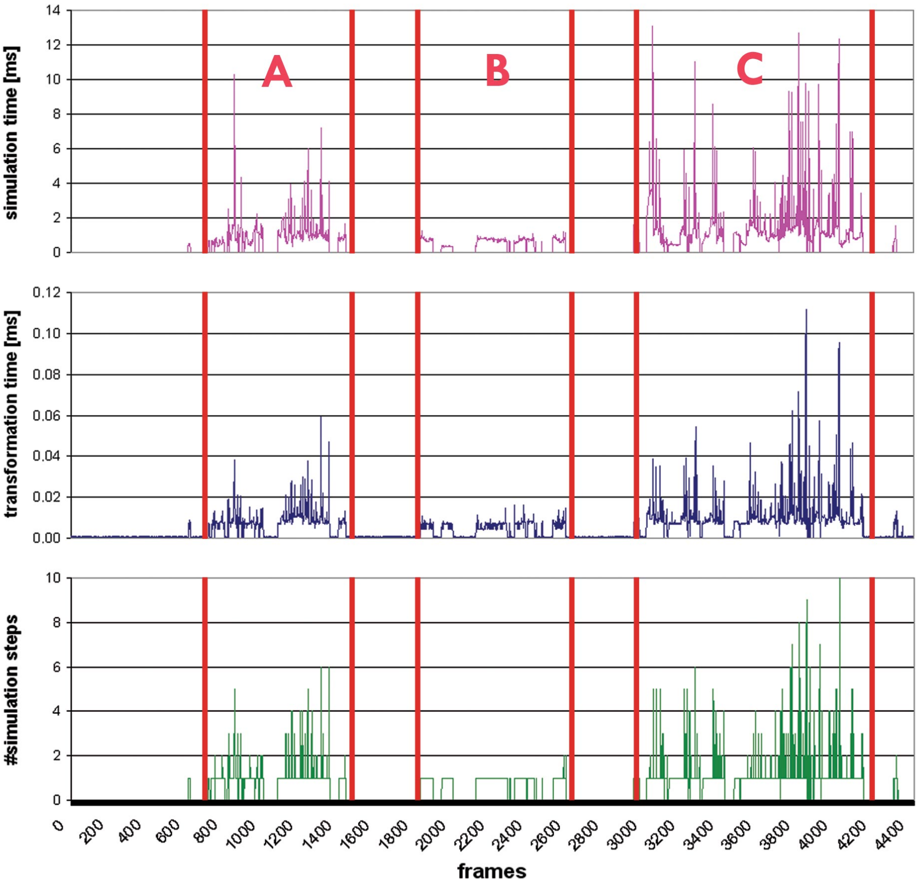

Figure 13: Simulation time, transformation time (which is simulation time excluding collision detection) and number of required simulation steps for different grasps of one object. The segments A and C are multi-finger grasps, the segment B is a two-finger grasp.

chical structure for grasping decision. Furthermore, the proposed method does not need a specifically designed collision detection algorithm, such as the one in [HH03] (cf. section 2). In fact, any modern collision detection library can be used.

The combination of independent sensors with the concept of grasp pairs allows intuitive interactions and multi-contact manipulation of multiple objects. 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 to address the issue of visual interpenetrations of the virtual gripping device and the grasped object 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. However, it is not yet possible to manipulate several objects which are in contact with each other as a whole, e.g., grasping two touching boxes with two sensors, each sensor being in contact with just one of the boxes. 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.

6 Acknowledgments

Parts of this work are supported by a grant from the Interdisciplinary Center for Clinical Research "BIOMAT." within the faculty of medicine at the RWTH Aachen University.

References

[BI05] Christoph W. Borst and Arun P. Indugula, *Realistic Virtual Grasping*, VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality, 2005, ISBN 0-7803-8929-8, pp. 91–98, 320.

[BRT96] R. Boulic, Serge Rezzonico, and Daniel Thalmann, *Multi Finger Manipulation of Virtual Objects*, Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST), 1996, ISBN 0-89791-825-8, pp. 67–74.

[Cho06] Su Bang Choe, *Statistical Analysis of Orientation Trajectories via Quaternions with Applications to Human Motion*, Ph.D. thesis, The University of Michigan, 2006.

[Gro05] H-Anim Working Group, *H-anim - humanoid animation 200x*, http://www.h-anim.org/Specifications/H-Anim200x, 2005, International Standard ISO/IEC FCD 19774:200x, last visited april 2007.

[HH03] Koichi Hirota and Michitaka Hirose, *Dexterous Object Manipulation Based on Collision Response*, VR '03: Proceedings of the IEEE Virtual Reality 2003, 2003, ISBN 0-7695-1882-6, pp. 232–239.

[KH96] R. Kijima and M. Hirose, *Representative Spherical Plane Method and Composition of Object Manipulation Methods*, Proceedings of the IEEE Virtual Reality Annual International Symposium '96, 1996, ISBN 0-8186-7296-X, pp. 195–202.

[KZK04] Ferenc Kahlesz, Gabriel Zachmann, and Reinhard Klein, *'Visual-Fidelity' Dataglove Calibration*, CGI '04: Proceedings of the Computer Graphics International (CGI'04), 2004, ISSN 1530-1052, pp. 403–410.

[MI94] Christine L. MacKenzie and Thea Iberall, *The Grasping Hand*, pp. 242–245, North Holland, 1994, ISBN 0444817468.

[US00] T. Ullmann and J. Sauer, *Intuitive Virtual Grasping for Non Haptic Environments*, PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications, 2000, ISBN 0-7695-0868-5.

[ZR01] Gabriel Zachmann and Alexander Rettig, *Natural and Robust Interaction in Virtual Assembly Simulation*, Eighth ISPE International Conference on Concurrent Engineering: Research and Applications (ISPE/CE2001), 2001.

Citation

Daniel Holz, Sebastian Ullrich, Marc Wolter, Torsten Kuhlen, *Multi-Contact Grasp Interaction for Virtual Environments*, Journal of Virtual Reality and Broadcasting, 5(2008), no. 7, July 2008, urn:nbn:de:0009-6-14909, ISSN 1860-2037.