2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Direct Hand Manipulation of Constrained Virtual Objects

Jun-Sik Kim and Jung-Min Park

Abstract— **We propose a direct object manipulation system that a user can manipulate objects constrained to each other by his or her hands. Because there is no haptic feedback to a user, manipulation of constrained objects is much harder than that of a simple rigid body. In addition, many physics simulator usually suffers from instability in handling the constrained objects in a virtual system. We solve the problem by adding another micro simulator which enforce the constraint between the objects, and guarantees that the rendered image of an object and its physics counterpart always satisfy the given constraint. The experiment shows that the proposed method makes a user manipulate constrained objects with no difficulties in the way that he or she manipulates such objects in the real world.**

I. INTRODUCTION

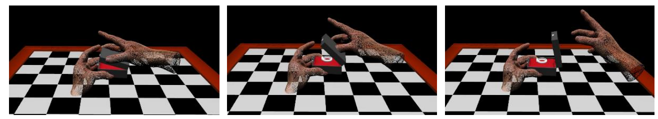
Virtual reality and augmented reality have been emerging for the past few years, and nowadays their markets start to open. It is an important issue to provide a more realistic user experience for advancing the VR applications. However, most of the state-of-the-art technologies focus on making a better visual experience to a user, while realistic interaction or manipulation of virtual objects by users is another big issue to have a better user experience.

Currently, interactions to the virtual objects use traditional methodologies: using hand-held devices to select/manipulate objects by an indirect raycasting method, or using gestures which are pre-defined. Though many "natural" user interaction methods have been proposed, most of them just use a remote motion sensing device and recognize the user's gesture, which is not the way where we manipulate objects in the real world.

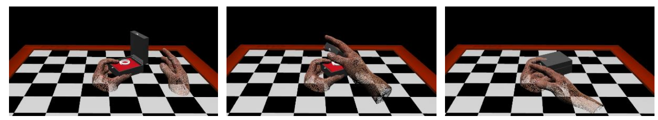
It is hard to develop a "really natural" user interaction to virtual objects, by which a user can manipulate the objects as if they would actually exist in the real world. First, there are numerous ways to handle objects depending on the object properties like shape, pose and texture. It is not possible to define and to implement all the possible ways of interaction for every object. Second, there is no device which can provide an accurate and realistic haptic feedback to a user's hand. Because a user's hand is not blocked by accurate reaction force, a hand and fingers can penetrate the surface of virtual objects, which causes a severe mismatching between virtual and real spaces.

This work was supported by the Global Frontier R&D Program on "Human-centered Interaction for Coexistence" funded by the National Research Foundation of Korea grant funded by the Korean Government(MSIP)(2011-0031425)

Jun-Sik Kim and Jung-Min Park are with the Center for Robotics Research, Korea Institute of Science and Technology, Hwarangro 14 gil 5, Seongbuk-Gu, Seoul 02792, Republic of Korea. junsik.kim@kist.re.kr.pjm@kist.re.kr



(a) Open by grasping



(b) Close by pushing

(c) Manipulation by multi-fingers

(d) Handover

Fig. 1: Various types of two-hand interaction with a 1-DOF constrained virtual object - a hinge-constrained object. Free-hand interactions appears in a various way.

Hand manipulation has been investigated in the robotics community for planning a grasping of a robotic gripper. One possible and traditional way is to use a prior knowledge of the objects in a form of predefined rules. [1], [2] This track of research is successful for planning to grasp an object which is simple in terms of its shape by a robotic manipulator. However, it is not applicable to free-hand manipulation which requires to solve interactions between a much more complicated hand model and unknown objects. Another track is to generate grasping poses automatically by data-driven methods. By collecting huge data of human grasping for various objects in various poses, it aims to generate possible grasping poses. [3], [4], [5], [6] Especially, Zhao et al. [6] proposed a method to generate the data-driven grasping sequence of a human hand which is captured by an RGBD sensor. Though this track of research is flexible and applicable to a human hand interaction, it is limited to generate grasping poses and hard to apply complicated interactions.

To ensure realistic interaction, it has been also investigated to utilize the physics engine actively. Kumar and Todorov [7] introduced a virtual hand interaction system based on their own physics engine *MuJoCo* specialized to a hand

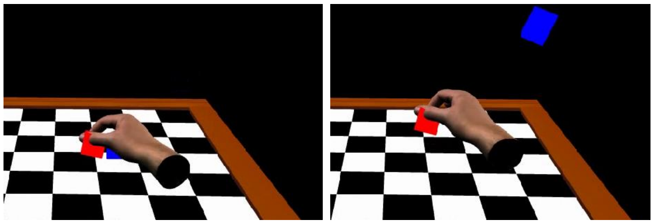
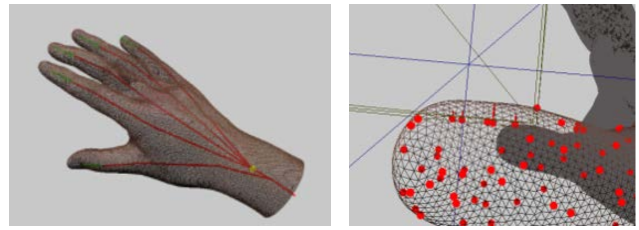


Fig. 2: Rubber band effect. The red and blue blocks are constrained by a hinge, but by grasping red block, the blue one is detached from the red one.



(a) Global deformation (b) Local deformation

Fig. 3: Deformable hand model and physics particles. [9]

interaction. Holodesk by Microsoft captures the postures fo human hands by an RGBD camera. From the given depth image, it generates physics particles which interact with virtual objects. [8] Kim and Park proposed a human hand deformation model with controllable physics particles which can interact with virtual objects. [9], [10] Although these physics-based methods work well, all of them is for manipulating a single rigid body.

Once multiple objects are constrained to each other, the number of possible manipulations increases, which implies simultaneous interaction by multiple hands. Fig. 1 shows some examples of manipulations of a simple constrained model, which has two objects constrained by a single hinge. These objects usually exist in the real world, but representing its hand interaction is not trivial. Its simulation complexity becomes bigger, and the instability becomes higher as well.

Modern physics engines have features to describe those constraints between objects. The constraint is usually solved by iterative solvers like a sequential impulse solver or a projected Gauss Seidel method. [11], [12] These *approximated* methods are working well when the given forces are not so abrupt, but grasping with a virtual hand model usually induces a big and abrupt contact force, which makes the approximation broken. Consequently, the pose of an object constrained to another object goes wrong or takes a lot of time to be converged, which is called a *rubber-band effect*, as shown in Fig. 2. Mathematically, the rubber-band effect is caused by the soft optimization, which means that the number of unknowns are larger than required, while the constraint is represented in the cost function to be minimized. To reduce the effect, careful determination of simulation time step and physics parameters is required. However, determination of time step and the mass and inertia for proper simulation is usually hand-crafted and scene dependent.

In this paper, we propose a geometry-based optimization of a hard-constraints to guarantee the given constraint satisfied. Objects constrained to each other is represented as a group object and its overall degree-of-freedom becomes minimal. Each object including a group object collects the contact information, and simulates its own pose update by optimization, which we call *micro-simulator*. The minimal parameterization of object poses is used in the "hard" optimization, which enforces the constraints so that a group object does not suffer from breaking constraints or the rubber band effect. Experiments shows that the proposed optimization induces real-like direct user interactions.

II. Hand Interaction Model and Grasping Control

Kim and Park introduced virtual object manipulation techniques based on a virtual hand model with physics particles [9] and object centered grasping control [10], and we extend the work so as to handling constrained objects. We briefly review the virtual hand model in Section II-A, and grasping control in Section II-B for further extension.

A. Virtual Hand Model

The used hand model has two deformation models: one for global deformation and the other for local deformation. Fig. 3 simply illustrates the concept of the two deformation models. A hand sensor such as a motion capture system provides a hand skeleton along with the hand pose, the global deformation process reshapes the hand mesh and calculates the positions of vertices. The position of each vertex $\mathbf{v}_i$ is calculated as

$$\mathbf{v}_i^{t+1} = \sum_{j=1..n} w_{ij}\mathbf{M}_{ij}\mathbf{T}_j\mathbf{v}_i^t \tag{1}$$

using multiple bone transformations $\mathbf{T}_j$, bone-to-vertex transformations $\mathbf{M}_{ij}$ and predetermined weighting factors w_{ij}. Physics particles are small spheres located at the selected vertices, and these are registered in the physics world to induce physics interactions. In our system, 1,182 vertices are randomly selected for physics simulation among the 39,225 vertices for rendering.

Once the hand mesh occludes other object as shown in Fig. 3(b), the physics particles move back to the object surfaces. This is a simple model of the skin deformation, which provides a stability in a physics simulation by reducing the contact forces.

B. Object-Centered grasping control

Though the physics particle based interaction method is basically using the physics engine to induce motions and manipulation, grasping an object is hard to be described only in the physics world. The grasping in full simulation requires strong friction forces between the hand and the target object, and reduces the contact forces to avoid the unexpected reaction forces. The lack of proper haptic feedback makes this harder because there is no way to block the finger motion penetrating into the object.

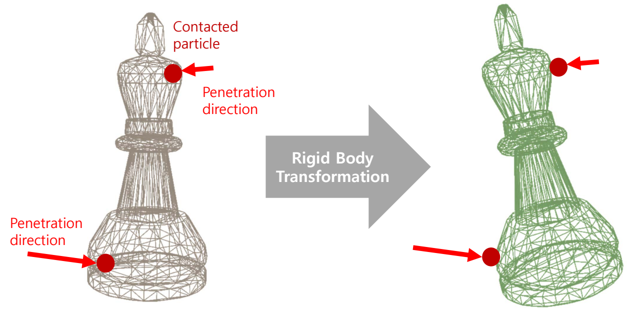


Fig. 4: Object pose update under multiple contacts. [10]

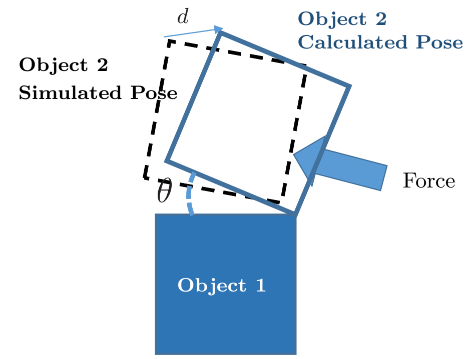


Fig. 5: Representation of a hard constraint. The relative pose of the object 2 with respect to the object 1 is determined by only the angle *theta*.

Rather than solving everything in the physics simulator, the object-centered interaction representation is used. In the object-centered representation, each object collects the collided objects including the hand physics particles. As shown in Fig. 4, the object determines its new pose considering all the physical contacts. This process is expressed as a simple pose optimization as

$$\arg\min_{\mathbf{R},\mathbf{t}} \sum_i |\mathbf{p}_i - \mathbf{p}'_i(\mathbf{R},\mathbf{t})| \tag{2}$$

where i is an index of the penetrating particle, $\mathbf{p}_i$ is the position of the penetrating particle i, and $\mathbf{p}'_i(\mathbf{R},\mathbf{t})$ is the transformed object surface point which corresponds to the $\mathbf{p}_i$. By solving the object poses continuously, the object can move in a natural way. The constraint will also be in the object-centered representation, which we will discuss in Section III.

The grasping is determined by the directions of the penetrating particles as

$$\mathbf{d}_i^\top \mathbf{d}_j < \gamma \tag{3}$$

where i and j are the indices of the contact particles and $\mathbf{d}_i$ and $\mathbf{d}_j$ are the corresponding penetrating directions. The grasping parameter γ is determined as 0.7 empirically. Once an object is grasped, the relative pose between the object and the grasping hand is fixed until the object is released.

III. Hand Manipulation of a Constrained Object

As we discussed earlier, solving dynamics of constrained objects is not trivial, and causes instability of the determined poses. It is mainly because the solver "over-parameterizes" the problem in a form of soft constraints. By representing an object or an object group in a minimal parameterization which enforces hard-constraint, instability of the constrained poses should be reduced.

We extend the idea of the object-centered manipulation representation in order to solve the problem. The key idea is that the pose of the constrained object is recalculated after all the simulation and hand interactions are done so that the poses of the objects can satisfy the given constraint. This can be regarded as a system-wide formulation of a hard constrained optimization, by approximating the hard-constrained poses in each iteration.

In this paper, we make an example of a two-body model which is constrained by a hinge, though the idea itself is not limited to a hinge joint.

A. Microsimulator: Enforcing hard constraints

Fig. 5 shows the idea of preserving hard constraint of a hinge joint, for example. The object 1 and 2 are constrained by a hinge and the only unknown is θ between the object. After the simulation, the object 2 is moved to "Simulated Pose" by the force in the figure, which breaks the hinge constraint. Our idea is to approximate θ of the calculated pose of the object 2 which describes the hardly constrained motion between the object.

As in the object centered interaction, the approximation of the angle θ is formulated as

$$\arg\min_{\theta} \sum_i |\mathbf{p}_i - \mathbf{p}'_i(\mathbf{R}(\theta),\mathbf{t}(\theta))| \tag{4}$$

where i is the index of a vertex $\mathbf{p}_i$ in the model object in the "Simulated Pose", and $\mathbf{p}'_i$ is the position of the vertex i in the "Calculated Pose" determined by the pose $\mathbf{R}(\theta)$ and $\mathbf{t}(\theta)$. The pose $\mathbf{R}(\theta)$ and $\mathbf{t}(\theta)$ is calculated by the given constraint. For example, if the hinge axis is the x-axis in the coordinate of the object 1 and the pivot point on the object 2 is $\mathbf{p}_2$ in the object 2 local coordinate and $\mathbf{p}_g$ in the coordinate of the object 1,

$$\mathbf{R}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}, \quad \mathbf{t}(\theta) = \mathbf{p}_g - \mathbf{R}(\theta)\mathbf{p}_2.$$

Eq.(4) is usually highly nonlinear, and thus, to solve the optimization, an iterative method by linearization should be used. In our implementation, Levenberg-Marquardt optimization is used with iteration limitation of 10.

This additional optimization is a form of object centered motion representation of a set of objects constrained to each other. By forcedly reducing the unknowns of the object poses, the "motion" is extended to a constraint between objects. Conceptually, this regards the two objects as a single object which has two parts, and the motion parameters are the pose of the combined object and the single hinge angle θ,

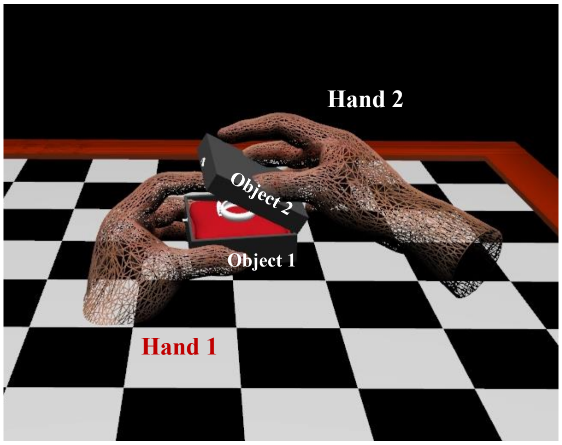


Fig. 6: Two-hand manipulation of a hinge-constrained object.

Algorithm 1: Multiple Hand Interaction with Constrained Objects

---

Data: Hand Skeleton Data
Result: Pose update of virtual objects
Deform the hand mesh;
Update positions of physics particles;
Detect hand collision and contact particles;
for *each object* **do**
  Collect contacted particles of the object;
  Determine if one or more hands grasp the object;
  if *one hand grasps it* **then**
    Set the object kinematic;
    Store the hand index and its relative pose as a primary hand;
    Update the object pose by the primary hand pose data;
  else
    if *multiple hands grasp it* **then**
      Store the hand indices and their relative poses of non-primary hands;
      Update the object pose by the primary hand pose data;
    else
      Set the object rigid;
Update poses for rigid objects;
Do physics simulation to update non-contacted objects.;
Update constrained parameters poses for constrained objects (4);
Update hand to object transformation for second hand.

---

which is minimal. As for the object-centered representation, this process can be regarded as a micro-simulator of the object group which reserves the minimal parameterization.

B. Object manipulation by hands

As we discussed in Section II-B, the grasped object is *attached* to the grasping hand. Once the constrained objects are manipulated by two hands as shown in Fig. 6, the pose of the object 2 is fully determined by the hand 2.

When the minimal parameterization proposed in Section III-A, the pose discrepancy between the object and the grasping hand becomes bigger as time goes by, and the optimization in (4) becomes harder. To avoid this problem, we reset the relative pose between the object and its grasping hand at every moment. Once the object is repositioned by the optimization (4), the pose of its physics counterpart is also updated. After the update, the relative pose $\mathbf{T}'_{rel}$ of the object with respect to the grasping hand is recalculated as

$$\mathbf{T}'_{rel} = \mathbf{T}'_{obj}\mathbf{T}^{-1}_{hand} \tag{5}$$

where $\mathbf{T}'_{obj}$ is a new object pose which is updated by the optimization, and $\mathbf{T}_{hand}$ is the pose of the grasping hand.

This process is a model of a physical interaction. When you recall the situation of opening lids which is constrained by a hinge as in Fig. 6, the grasping hands usually slip rotationally and the relative pose between the object and the hand changes. The updating of the relative poses is only applied to the second grasping hand in terms of the object group, which means the pose of the object group which consists of two constrained objects is not changing.

The overall interaction algorithm is shown in Algorithm 1. Though the collision detection and physics simulation are made twice, it is not so problematic. The required timing for our application is only 30 fps which is video rate, and the additional process does not hurt the overall speed.

IV. Experiments

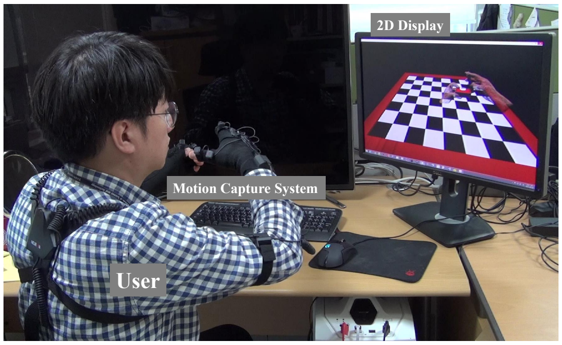
To test the performance of the proposed interaction system, we made a user interaction system using an IMU based motion capture system for hand and finger motions. For this test, we evaluated the effect of manipulation tasks, opening or closing an lid of a box which is constrained by a hinge. In addition, the effect of the visualization to a user's perception has been tested by comparing the result using a flat panel 2D display and that using a head-mounted display providing 3D sensation.

A. System Setup

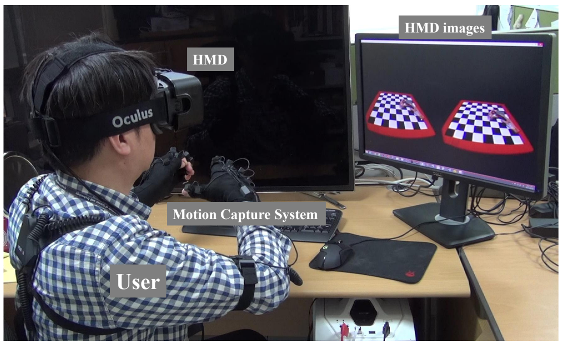
Fig. 7 shows the test setups for two visualization effects. One is using a conventional 2D display, and the other is using an Oculus Rift DK2 HMD for 3D visual effect. [13] A user's head motion is compensated by a built-in IMU sensor in the HMD, so that the user can perceive the object location and environment easily.

For hand and finger motion, we used the Perception Neuron [14], which has 26 IMUs for upper body and finger motion capture. Finger poses which is provided by the sensor is only 21 DOF including the 6 DOF hand pose for each hand, and thus, the finger motion is not so accurately represented. The user pose was determined manually in order to simplify the registration of the user with respect to the virtual space.

The used system is with Intel Core i7-3770 (3.5GHz) and 16 GB memory. All the system has been implemented on

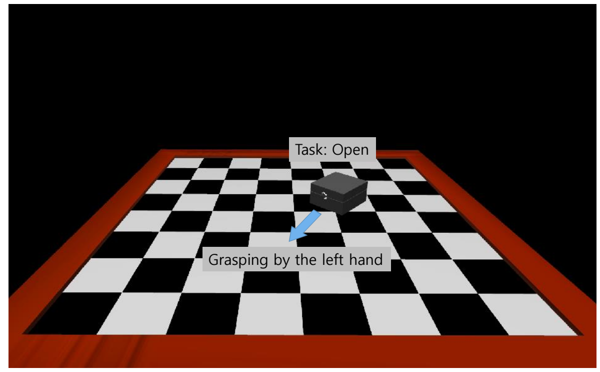


(a) 2D Visualization using a 2D monitor



(b) 3D Visualization using an HMD

Fig. 7: Experimental setup. Effects of two perception methods have been tested. (a) 2D perception using a 2D flat display, and (b) 3D perception using an HMD. For motion capture of hands and fingers, a motion capture system based on inertial sensors has been used.



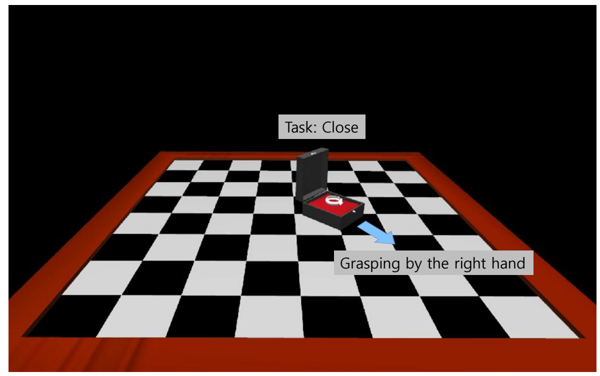


Fig. 8: Task specification for experiments. The subject is asked to pick the object up by the fronting hand and open/close the hinge object with the other hand.

Openframeworks in C++ on OpenGL, and Bullet Physics 2.82 for physics simulation and contact detection. [15] The whole process is running only on the CPU multithreaded by Boost 1.57 and OpenMP. The overall frame rate including rendering and the physics simulation is more than 60 Hz, which is good for visualization in video rate.

B. Two-hand Manipulation Test

In Fig. 8, experimental tasks are illustrated. In the beginning, a jewelry box is dropped at random position in front of a subject, and its lid is open or close. Once the lid is open, the subject needs to close the lid after picking up the box in one hand. If the lid is close at the first time, the subject is asked to open it after picking the box up as well. The hinge joint was limited from zero to 90 degrees. One can notice the small blue arrows in Fig. 8, and it denotes the grasping hand. The box should be picked up with the user's right hand if the box faces rightward, and vice versa.

Success or failure of the task is determined by a human test manager, with the criterion if the object moves in a strange way, or it moves in a way different to the subject's task. The experiment system automatically measures the time for manipulation, starting from the grasping of the first hand until the given task is accomplished.

1) Effect of Task: Table I shows the number of trials and its success rate. Overall success rate is about or over the

TABLE I: Number of trials and success rate in handover test

Task and Condition	Trials	Success Rate (%)
Open 2D	221	88.2
Close 2D	302	99.0
Open 3D	238	90.3
Close 3D	263	98.9

90% in any task under any perception condition. The success rate of the closing manipulation is significantly higher than that of the opening one, and this is because the closing is much easier than the opening. The lid opening requires the continuous hand grasping of the lid object, and once the hand motion is faster than that, the grasping status is released, which causes manipulation failure. Closing the lid is possible by only pushing the object in a single direction, which is much simpler. This effect is clearer in the execution time analysis.

Table II shows the execution time for each task under the visualization condition for successful trials. As one can simply notice, the lid-opening task takes significantly longer time than the closing tasks in both visualization conditions. Fig.9 shows the distribution of the task execution time, which is estimated using a kernel density estimation. In this figure, the mode of the distribution is significantly separated, and this implies the closing manipulation is much easier.

The impact is much more dramatic when we check the hinge operation time. The hinge manipulation time is the time between the second hand touches the lid object and the tasks is ended, which is the actual time only for hinge

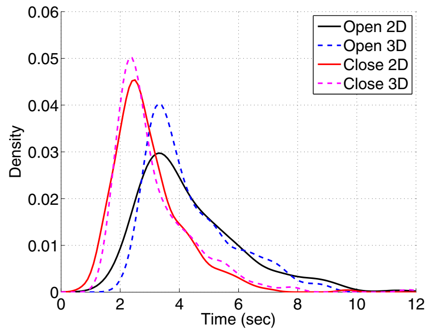


Fig. 9: Effect of tasks and visualization methods.

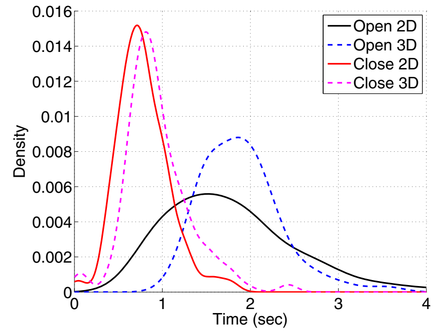


Fig. 10: Execution time only for hinge manipulation.

TABLE II: Statistics of execution time for successful handovers

Task and Condition	mean (second)	standard deviation
Open 2D	4.34	1.71
Close 2D	2.96	1.17
Open 3D	4.31	1.44
Close 3D	3.12	1.34

manipulation. Fig. 10 shows the distribution of the hinge manipulation time. For closing, the manipulation takes less than a second by pushing the lid in a right direction. Opening the lid is much harder and takes more time which is longer than two seconds in some cases, depending on the hand direction.

2) Effect of 3D perception: Previously, Kim and Park reported that the 3D visual experience provides a big impact to a user's interaction [10] when a haptic feedback is not sufficient. In the hinge manipulation test, the difference between the two is not significant in both the success rate and the execution time. Although the hinge manipulation experiments do not have any haptic feedback as well, it would be because the tasks required is more difficult than the simple rigid-body manipulation, and the grasping time is almost the same in both tasks.

The success rate is also similar in both visualization methods. It implies that the important factor in manipulating constrained objects is the constraint control rather than the grasping itself.

V. Conclusion

We have proposed a direct hand manipulation system for constrained virtual objects. Physics simulation of constrained object is usually formulated as an optimization of soft constraints, which have more unknowns than actually required. The over-parameterization makes the simulation instable and causes the rubber band effect, especially in the human hand interaction due to the strong and abrupt contact forces. We add another small optimization for hardly enforcing the minimal parameterization, and the optimization can be regarded as a microsimulator of the object group which consists of constrained objects. This is a direct extension of the object-centered motion representation and successfully solve the instability issues in a simple addition to the existing system. Combining the hand-based direct interaction system, any natural and direct user interaction in virtual reality for the constrained objects can be achieved.

Experiments show that the proposed methods provides a good enough for users to manipulate the complicated object as they do in the real world overall. Two aspects have been tested, one for tasks, and the other for 3D visualization effect. Though there is no haptic feedback at all, users can successfully do the given tasks by using a 2D display and a 3D HMD.

References

[1] H. Rijpkema and M. Girard, "Computer animation of knowledge-based human grasping," in *ACM Siggraph Computer Graphics*, vol. 25, no. 4. ACM, 1991, pp. 339–348.

[2] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1824–1829.

[3] G. ElKoura and K. Singh, "Handrix: animating the human hand," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003.

[4] L. Ying, J. L. Fu, and N. S. Pollard, "Data-driven grasp synthesis using shape matching and task-based pruning," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 4, pp. 732–747, 2007.

[5] N. S. Pollard and V. B. Zordan, "Physically based grasping control from example," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2005.

[6] W. Zhao, J. Zhang, J. Min, and J. Chai, "Robust realtime physics-based motion control for human grasping," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 207, 2013.

[7] V. Kumar and E. Todorov, "Mujoco haptix: A virtual reality system for hand manipulation," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 657–663.

[8] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson, "Holodesk: direct 3d interactions with a situated see-through display," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 2421–2430.

[9] J.-S. Kim and J.-M. Park, "Physics-based hand interaction with virtual objects," in *Robotics and Automation, 2015. Proceedings. ICRA'15. IEEE International Conference on*. IEEE, 2015.

[10] J.-S. Kim and J.-M. Park, "Direct and realistic handover of a virtual object," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[11] E. Catto, "Modeling and solving constraints," in *Game Developers Conference*, 2009, p. 16.

[12] S. Rabin, *Introduction to game development*. Nelson Education, 2010.

[13] "Oculus rift," https://www.oculus.com/en-us/rift/, accessed: 2017-02-22.

[14] "Perception neuron by noitom," https://neuronmocap.com/, accessed: 2017-02-22.

[15] E. Coumans, "Bullet physics simulation," in *ACM SIGGRAPH 2015 Courses*, ser. SIGGRAPH '15. New York, NY, USA: ACM, 2015. [Online]. Available: http://doi.acm.org/10.1145/2776880.2792704