2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

High-Fidelity Grasping in Virtual Reality using a Glove-based System

Hangxin Liu[1⋆] Zhenliang Zhang[1,2⋆] Xu Xie[1,3] Yixin Zhu[1,3] Yue Liu[2] Yongtian Wang[2] Song-Chun Zhu[1,3]

Abstract— **This paper presents a design that jointly provides hand pose sensing, hand localization, and haptic feedback to facilitate real-time stable grasps in Virtual Reality (VR). The design is based on an easy-to-replicate glove-based system that can reliably perform (i) a high-fidelity hand pose sensing in real time through a network of 15 IMUs, and (ii) the hand localization using a Vive Tracker. The supported physics-based simulation in VR is capable of detecting collisions and contact points for virtual object manipulation, which drives the collision event to trigger the physical vibration motors on the glove to signal the user, providing a better realism inside virtual environments. A caging-based approach using collision geometry is integrated to determine whether a grasp is stable. In the experiment, we showcase successful grasps of virtual objects with large geometry variations. Comparing to the popular LeapMotion sensor, we demonstrate the proposed glove-based design yields a higher success rate in various tasks in VR. We hope such a glove-based system can simplify the data collection of human manipulations with VR.**

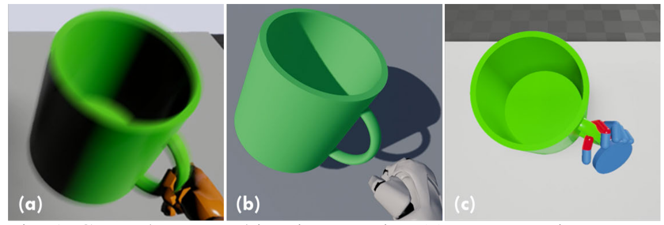


Fig. 1: Grasp the same object in VR using (a) a LeapMotion sensor, (b) an Oculus Touch controller, and (c) the proposed glove system. The grasping in (a) is unstable, reflected by the motion blur, due to the noisy vision-based hand pose sensing approach. While (b) allows a stable grasp, the grasp is unnatural and not even contacts the object, as the object simply attaches to the hand once the user triggers the grasp event. The proposed design in (c) yields the most stable and natural grasp among three. The red areas of the hand model indicate the forces exerted during manipulations.

I. Introduction

Grasping is arguably one of the most critical interactions inside VR environments. Serving as an effective means to train and test subjects in various environments and events [1], [2], [3], [4], [5], [6], [7], VR has been receiving increasing interests in both industrial and academic applications. In addition, researchers in Artificial Intelligence (AI) also identify VR as a compelling platform to collect human data for training AI agents or robots [8], [9], [10].

Some hardware setups (*e.g.*, [11]) have been developed to improve the user interactions inside virtual environments as the user experience of virtual manipulations becomes more and more crucial, demanding for higher fidelity solutions to enhance such a virtual interaction experience. However, fine-grained human grasp sensing in VR is still hindered by the lagging development of a low-cost but high-fidelity solution. We argue that a successful solution should address:

1) **Reliable hand pose sensing**. The commercial hand pose sensing products (*e.g.*, [12], [13]) for VR use vision-based gesture recognition approaches. Although these products are low-cost and can be easily set up, it is difficult to have a stable grasp due to occlusions and sensor noises (see Figure 1(a)). Additionally, the vision-based approach usually mounts the sensor on the head-mounted display, which makes it impossible to track the hand pose when the hand is out of the field-of-view of the head-mounted display. This paper presents a network of IMUs on a data glove to overcome challenges of the hand pose sensing.
2) **Real-time natural grasp**. The virtual grasp has to be stable in order to manipulate virtual objects properly. The existing solution used by the Vive and Touch controller is essentially an attachment-based approach; *i.e.*, the virtual object will automatically attach to the virtual controller/hand once the user triggers the grasp event (see Figure 1(b)). Such a grasp has very limited realism, providing unsatisfactory user experiences. The proposed design introduces a caging-based grasp to balance the realism of grasp and real-time performance.
3) **Fine-grained haptic feedback**. When manipulating physical objects in daily life, we actively sense the contact of the object on-the-fly and adjust the hand pose to manipulate the physical object accordingly. Such vital haptic feedback during the contact or collision is still largely missing in the previous designs. Although some commercial controllers (*e.g.*, Vive or Touch controller) provide vibration feedback, such feedback is very crude as the entire controller will vibrate. The proposed design advances the haptic feedback to each finger (see Figure 1(c)), enabling finer-level feedback.

In this paper, we propose a design that jointly addresses these three challenges by combining a glove-based hardware system and a caging-based grasp approach in VR. The glove-based system offers the merit of a more accurate hand pose sensing and hand localization; it also supports the implementation of haptic feedback devices. In addition to recovering the hand pose, a physics-based simulation is adopted for collision detection between virtual hands and objects. A real-time caging-based approach is integrated to determine whether the grasp is stable inside VR.

⋆ H. Liu, Z. Zhang contributed equally to this work.
1 UCLA Center for Vision, Cognition, Learning, and Autonomy (VCLA) at Statistics Department. Emails: {hx.liu, zlz, xiexu, yixin.zhu}@ucla.edu, sczhu@stat.ucla.edu.
2 School of Optics and Photonics, Beijing Institute of Technology. Emails: {liuyue, wyt}@bit.edu.cn.
3 International Center for AI and Robot Autonomy (CARA)

A. Related Work

There are three main streams of hardware setups for **hand pose sensing**. *Exoskeleton* [14], [15] usually relies on complex mechanical structures to sense hand pose and is able to provide haptic feedback. It is, however, usually bulky for natural hand motions and difficult to replicate. *Pneumatic* glove designs are powerful in actuating hand motions, but it requires a large and heavy control box to pump liquid/air to the pneumatic actuators. This characteristic makes it unsuitable for VR applications. The glove designs utilizing *Inertial* sensors, mainly IMUs [16], [17], provide a more elegant solution as its setup is usually light enough that would afford natural movements. Efforts have also been made to improve pose sensing accuracy using filtering [18] and estimation algorithms (*e.g.*, extended Kalman filter [19], [20]). Note that the prior work can only refer the sensed hand pose to a local frame; for instance, to palm or wrist frame. In contrast, in the proposed design, we incorporate a network of 15 IMUs (one for each phalange) for hand pose sensing with a Vive Tracker to provide global hand positioning.

The current dominating method in the **virtual grasp** is the symbolic grasp defined by [21]—the virtual object is attached to the virtual hand when a grasp happens or is triggered. Symbolic grasps can achieve real-time computation; however, it limits natural manipulations, and user experiences as the poses of object attachment are pre-defined, causing finger penetrations or non-contacting grasp. In order to facilitate more accurate grasp, the collisions between the hand and the object have been taken into account to determine a stable grasp [22], [23]. Meanwhile, physics-based approaches (*e.g.*, [24]) in grasp and manipulation of virtual objects can help address the problem of the visual interpenetration of hand and object models, but often requires sophisticated simulations which have difficulties to run in real-time. A grasped object usually would "stick" to the hand when releasing the object, resulting in less satisfactory user experience. In this paper, we propose a caging-based approach to address the stable grasp issue that balances between the real-time requirement and the natural interactions in VR.

There is a recent trend in providing hardware-based **haptic feedback** to users in VR. Choi *et al.* [25] designs a motor and braking system to constrain the grasp pose. Kim *et al.* [26] provides tactile, pseudo-force, and vibrotactile feedback, but only for the thumb. Choi *et al.* [27] simulates grasping weight, and Jain *et al.* [28] introduces the Peltier module to simulate temperature change. Although such work provides very detailed rendering for a particular type of feedback, these designs are complicated to be implemented to provide generic haptic feedback or too complex to be compatible with other hardware systems. We instead use a network of vibration motors to provide haptic feedback in a fine-grained scale (*i.e.*, in each finger) to increase the realism in VR.

B. Contribution

This paper makes the following contributions:

1) We propose a design of a glove-based hardware system that incorporates a network of IMUs for hand pose sensing, a Vive Tracker for hand localization, and a network of vibration motors to provide vibrational haptic feedback. A light-weight Raspberry Pi serves as the control unit of the system that collects and transmits the data between the physical glove and the virtual environment.
2) We introduce a real-time method to determine stable grasps based on the collision geometry between the hand and object during manipulations. The geometry center of all the colliding parts of the hand is computed, and the object pose follows the hand pose if the geometry center is overlapping with any part of the object.
3) We provide a Data Logging system that records detailed manipulative data involved during the hand-object interaction in addition to the hand pose afforded by the glove-based system, such as object pose, contact points, *etc*..

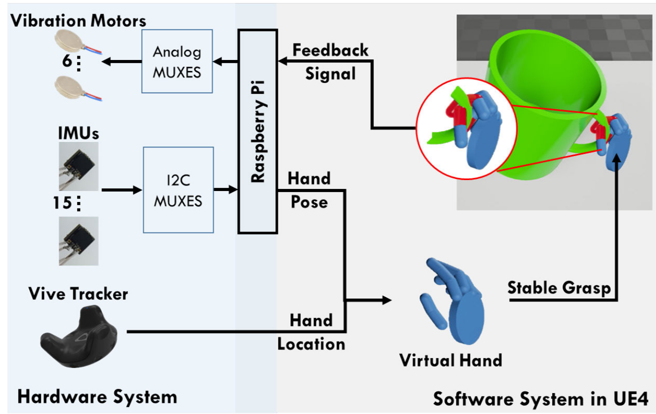


Fig. 2: Schematics of the system architecture. A Vive Tracker and a network of IMUs provide comprehensive hand pose and location in the VR simulator Unreal Engine 4 (UE4). When a stable grasp is formed based on the contact geometry, the virtual object follows the hand's movement, and the vibration motors at the contact area are triggered to provide the user with vibrational feedback.

C. Overview

The remaining of this paper is organized as follows. In Section II, we introduce the proposed overall design and hardware implementations of the prototype. The virtual grasp algorithm is described in Section III. The performance of the proposed design is evaluated and discussed in Section IV via a series of experiments. We conclude our work in Section V.

II. System Architecture and Prototyping

This section presents the overall hardware system schematic. A network of 15 IMUs following the design of [16] is configured and deployed. Six vibration motors are used to provide vibrational haptic feedback for each finger. Each vibration motor would be triggered when the corresponding finger collides with virtual objects. Hand location is tracked by a Vive Tracker [29] with a Lighthouse.

A. Overall Design

Figure 2 shows the schematic of the integrated system. The hardware includes a data glove and a processing unit. The data glove supports a network of 15 IMUs for hand pose sensing. The pose information is collected and processed by a Raspberry Pi. With additional hand locations captured by

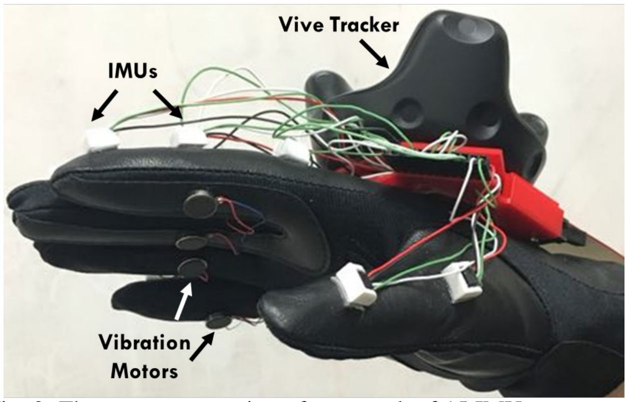


Fig. 3: The prototype consists of a network of 15 IMUs, a network of 6 vibration motors, and a Vive Tracker.

a Vive Tracker, the grasp pose and the hand movement are reconstructed in the world coordinate system as a virtual hand in VR. When a stable grasp is detected (see Section III), the phalanges that collide with the virtual object would trigger a network of 6 vibration motors, controlled by the Raspberry Pi, to provide vibrational haptic feedback to users.

B. Hardware Implementation

Pose Sensing: The pose sensing module consists of 15 Bosch BNO055 9DoF IMUs and a pair of TCA9548A I^2C multiplexers. One IMU is mounted to the palm, 2 IMUs are to the 2 phalanges of the thumb, and the rest of 12 IMUs are to the 3 phalanges for each of the other four fingers. The IMUs yield high-fidelity orientation in the form of a quaternion for each phalanx of the hand through their built-in proprietary sensor fusion algorithm.

A customized 6.35×6.35 mm^2 breakout PCB is designed to mount IMUs, making it easier to attach to the glove fabric with minimum constraints to the user's natural hand motion. The IMUs are connected to the I^2C bus interfaces available on a single Raspberry Pi 2 Model B—acting as the master controller of the system—through the multiplexers. The layout for our pose-sensing pipeline is based on [16], in which the experiments quantify the characteristics of such an arrangement and the IMU performance.

Haptic Feedback: Visual information alone is often insufficient to afford a natural and realistic hand-object interaction in VR, requesting for additional haptic feedback. Although a haptic device is itself a complex system, this paper implements a network of shaftless vibration motors to provide vibrational feedback at each finger, which will be triggered when the phalanx collides with the virtual object.

The vibration motors used in the system is small (10×2 mm^2) and light-weight (0.8 $gram$), providing $14,500$ Rotation per Minute at $3V$ input voltage. The motors are connected via a 74HC4051 analog multiplexer, which is controlled by the Raspberry Pi's GPIO. Once the hand forms a stable grasp in the virtual environment, the vibration motors located at the corresponding contact points are powered up for a short duration. This feedback signals the user to maintain current hand pose to keep a stable grasp in VR.

Hand Localization: Although the IMU-based hand pose sensing solution can reconstruct the hand pose within a local frame (*e.g.*, the wrist frame), the hand location relative to the global frame is still needed to recover a global hand movement inside the VR. Localizing hand by vision-based methods often require a skin-mask as the first step to segment the hands from the background. Wearing data gloves disallows such methods, requiring a hardware solution.

Naturally compatible with the VR, the Vive Trackers allow precise localization in the environment using the HTC Lighthouse, a laser sweeping device. Minimal two Lighthouses is set up to emit lasers that sweep alternatively in vertical and horizontal directions. The location and the orientation of a Vive Tracker in the 3D space is calculated by the time-difference-of-arrival of the laser impulse detected on the photodiode inside the Vive Tracker.

In the proposed design, a Vive Tracker is attached to the back of the palm. The position and orientation of the virtual hand are computed using the Vive tracker and the IMU on the glove's palm, respectively.

Communication: A real-time ROS-UE4 bridge is built to communicate between the Raspberry Pi mounted on the glove and the UE4 engine running on a remote desktop. Specifically, the local hand poses obtained from IMUs are collected by the Raspberry Pi and wirelessly transmitted to the UE4 simulator. The Vive Tracker that provides global hand poses is also wirelessly connected to the UE4 environment. Using such information, the hand pose is reconstructed in UE4. A caging-based algorithm detects the collisions to determine whether a stable grasp occurs. The detected collisions in UE4 will consequently trigger the vibration motors wirelessly on the glove through the ROS-UE4 bridge.

C. Prototyping

Figure 3 shows a prototype of the proposed design. We choose a compression glove, aiming to reduce the skin-motion artifact. Each of the 15 IMUs is placed inside a 3D-printed housing that is sewed into the fabric of the glove. The physical connections from the IMUs to the multiplexer use a high-flexibility, silicone-coated 29-gauge stranded-core wire. Such a layout of the IMUs and the wire connection do not interfere with hand motions.

A total of 6 vibration motors are attached to the surface of the glove: one on the middle of the palm, one on the distal phalanx of the thumb, and one on the middle phalanx of each other finger. A Vive Tracker is placed on the back of the palm to maximize its exposure to the Lighthouse.

III. Virtual Grasp

In this section, we present the construction, calibration, and grasp detection of the virtual hand to realize a stable grasp in UE4, a popular VR environment.

A. Virtual Hand Construction

The human hand has approximately 22 degree-of-freedom (DoF) regarding rotation (see Figure 4): 2 DoFs for metacarpophalangeal (MCP) joints, 1 DoF for proximal interphalangeal (PIP) joints, 1 DoF for distal interphalangeal (DIP) joints, and 3 DoFs for the palm. With these constraints, the

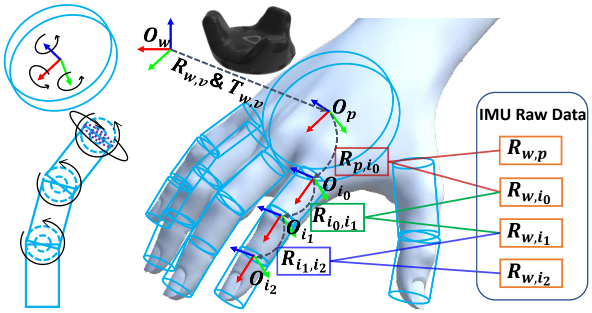


Fig. 4: The structure of the virtual hand model, in which every phalanx is modeled as a small cylinder. The pose of each phalanx can be computed using IMUs, and a Vive Tracker directly tracks the pose of the palm.

thumb finger can be modeled as a 3 DoFs kinematic chain, and each of the other four fingers can be modeled as a 4 DoFs kinematic chain, where the palm is the base of all the fingers. The pose of each finger phalanx is determined by the forward kinematic using homogeneous transformation

$$ {}^{i-1}_{\ \ i}H = \begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix}, $$

where R is a 3×3 rotation matrix and T is a 3×1 translation matrix *w.r.t.* the parent frame $i-1$ to the child frame i.

Let O_w denotes the origin of world coordinate system, the Vive Tracker provides the rotation from *world (w)* to *palm (p)* $R_{w,p}$, and the translation $T_{w,p}$ to indicate the palm coordinate within O_w. Each IMU produces a raw rotation data in quaternion q_i *w.r.t.* its local frame O_i. Using the index finger as illustration, the IMUs placed on palm, proximal, middle, and distal phalanx produce quaternion q_p, q_1, q_2, and q_3, respectively. The corresponding rotations for the MCP, PIP, and DIP joints are

$$ R_{MCP} = f(q_1/q_p), R_{PIP} = f(q_2/q_1), R_{DIP} = f(q_3/q_2), $$

where $f(\cdot)$ denotes the conversion function from a quaternion to a rotation matrix, and $/$ denotes the quaternion division.

Translation matrix $T_{(\cdot)}$ is defined according to the dimension of the hand. Finger joints motion limits are adapted based on the previous study [30].

B. Hand Calibration

Drifting is a common issue with the inertial sensors like IMUs, which introduce accumulated errors in grasp sensing. To eliminate the effect, we design a hand calibration routine, during which the user wears the glove and holds the hand flat in a canonical pose on a flat surface (see Figure 5). The orientation of a single IMU is measured in the local coordinate system, *i.e.*, the glove frame. Once the calibration event is triggered, the relative pose between the glove frame and the world frame is recorded, and the hand poses obtained later is multiplied by the inverse of the relative pose to cancel out the differences.

C. Caging-based Stable Grasp in UE4

There are two main streams to realize a virtual grasp inside VR; both have their advantages and disadvantages. One stream is to simulate soft tissues of the hand and the material of the virtual object using a physics-based simulation with collision detection. This method provides a very realistic manipulation but often requires significant computing resources, making it difficult to run in real-time.

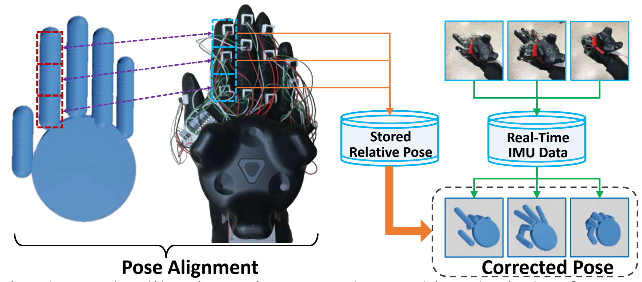
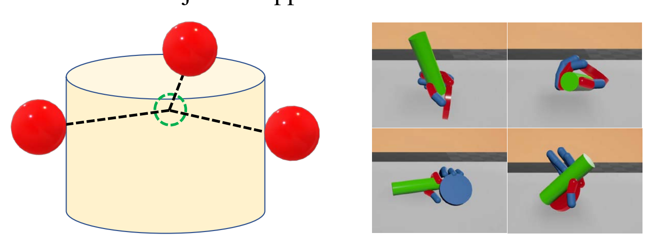


Fig. 5: Hand calibration using IMU data. Taking the index finger as an example, the user first holds the finger flat, the same pose as the index finger of the virtual hand model. Once the calibration event is triggered, the relative pose between the virtual hand model and the IMU coordinate system is recorded. The IMU data obtained later is then corrected using the inverse of the recorded relative pose.

Conversely, another stream is the symbolic-based or rule-based grasp. Using a set of pre-defined rules, a grasp or a release is triggered once specific rules are satisfied. This approach is usually straightforward to compute but provides very limited user experiences in terms of realism.

In this paper, we seek to find a reasonable balance, *i.e.*, to provide a more natural manipulation than the symbolic-based approaches, but require less computational demands than physics-based simulations. The proposed caging-based stable grasp consists of the following four steps.

1) Starts with a full simulation for both virtual hands and all the virtual objects so that the collision can be simulated as realistically as possible.
2) Detect all parts of hands that are in collision with other virtual objects (see red areas in Figure 6 (b)).
3) Compute the geometry center of all the collisions points of the hand, and check whether the center is overlapped with any virtual objects (see Figure 6 (a)). If there is an overlapped object with the computed geometry center, we consider this overlapped object is caged, *i.e.*, grasped in the virtual environment. The physics property of this object is turned off so that the object will move together with the hand. In this process, we guarantee that the grasp only starts when a caging is formed, improving the user experience of natural manipulations.
4) Release the grasped object when (i) the computed geometry center is no longer overlapped with the grasped object, or (ii) the collision between the virtual hand and the virtual object disappears.



(a) (b)

Fig. 6: Collision detection of a grasp. (a) Illustration of the collision detection method. When the geometry center (green ball) of all the collision points (red balls) is overlapped with an object (yellow cylinder), the overlapped object will be grasped and attached to the hand. (b) Examples of grasping small cylinders in various ways.

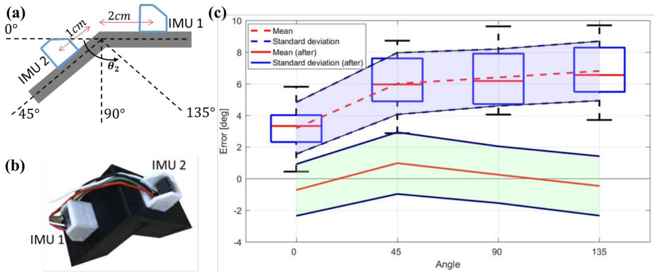


Fig. 7: (a) Schematic of the experimental setup for evaluating the angle reconstruction using two articulated IMUs. (b) An exemplary setup of the physical device. (c) Performance of the angle reconstruction. Box plot: red horizontal lines indicate the median error, and the bottom and top edges of the blue boxes indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points that were not considered outliers. Blue strip: the mean and standard deviation of the reconstructed angles using articulated IMUs. Green strip: the mean errors reduce after applying for least squared error compensation.

IV. Experiment

A. IMU Evaluation

The pose sensing building upon the performance of the IMUs is critical to the fidelity of grasp and user experience in the VR. We experiment to evaluate the accuracy in estimating a joint angle using two adjacent IMUs and model the IMU's performance in terms of its bias and variance. In particular, we manufacture four rigid bends with fixed angles of $0°$, $45°$, $90°$, and $135°$ to simulate a finger's bending. The experimental schematic is shown in Figure 7(a). Figure 7(b) is an exemplary setup of the physical device using $90°$ joint angle: two IMUs are placed $2cm$ away behind the bend and $1cm$ ahead of the bend, aiming to recover the joint angle of the bend. This setup relies on two reasonable assumptions for evaluations: finger joints are revolute, and IMUs are at the same radii relative to joint.

Figure 7(c) shows the errors of estimated joint angles for four settings with twenty repeated trails. The errors increase from $4°$ to approximately $6°$ when the bending angles increase from $0°$ to $135°$. We can also see that the articulated IMUs under-perform as the bending angle increases, but the error range reveals a consistent pattern, hinting at a possible compensation for such error. We apply a least squared fit $y = 0.0249x + 3.9068$, where y is the compensation amount, and x is the current sensed angle. Such error compensation procedure reduces the sensing error to about ± 1 degree.

B. User Experience

We further demonstrate the performance of the proposed design qualitatively by grasping a variety of objects with a large variation of the geometry, including a mug, a tennis racket, a bowl, and a goose toy; see Figure 8. These four everyday objects are chosen for two reasons: (i) the four objects are everyday objects in daily life with different shapes and complexities, providing effective assessments of the performance measure of the virtual grasp, and (ii) these objects have different functions, requiring various grasp types [31].

Additionally, we test different ways to interact and manipulate virtual objects; *e.g.*, we can pick up a mug not only

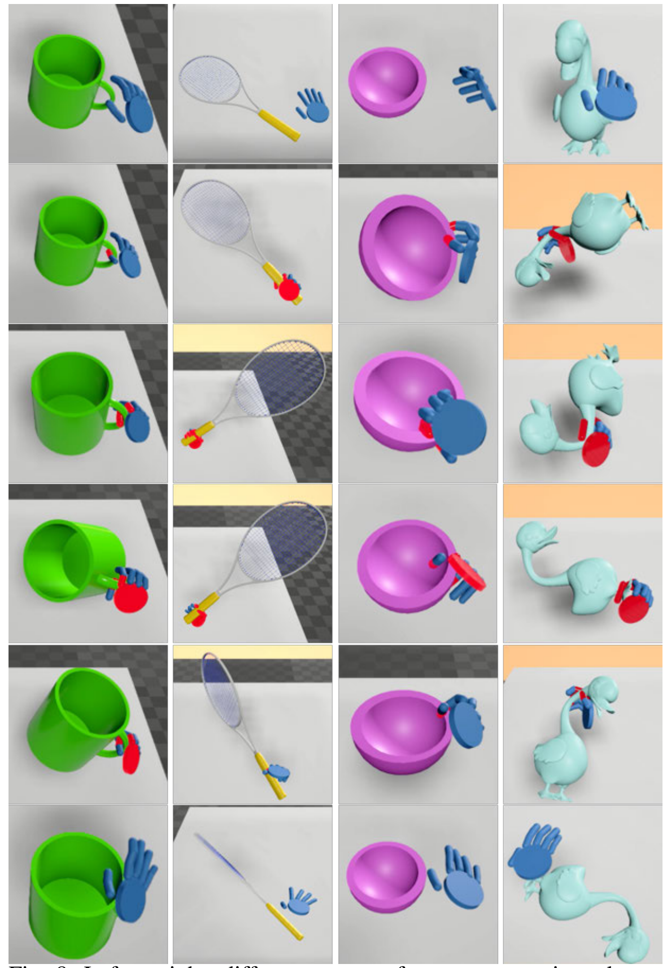


Fig. 8: Left to right: different grasps of a mug, a tennis racket, a bowl, and a goose toy. Top of each column: approaching the target object. Bottom of each column: release the target object.

by grasping the handle but also by grasping the rim. Such diverse interactions and manipulations afford a natural interaction experience by integrating unconstrained fine-grained gestures, difficult for other existing devices (*e.g.*, LeapMotion) to achieve. With the proposed caging-based grasp approach, the proposed glove-based device has achieved a balanced trade-off between the natural interactions and the stable grasp, providing a better realism in the virtual world that is close to the manipulation in the physical world.

Note that one significant advantage of the proposed design compared to the popular LeapMotion sensor setup is: the

TABLE I: Success rates of grasping and moving four different objects. The proposed glove-based system (G) significantly surpass the LeapMotion sensor (L) in both cases.

Objects	Setup	Grasp (%)	Move (%)
Mug	L	80	33
	G	100	100
Racket	L	13	7
	G	100	93
Bowl	L	27	0
	G	100	93
Toy	L	67	47
	G	93	87

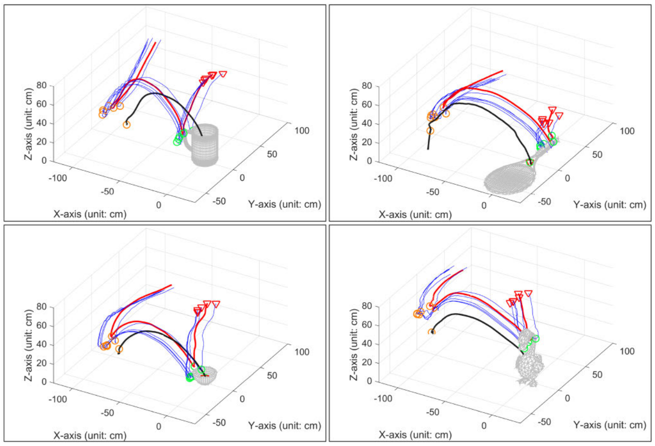


Fig. 9: Various hand and object trajectories collected using the proposed design in VR. Starting from the red triangles, the red line indicates the hand movement, and the blue lines are the trajectories of fingertips. After an object is grasped, determined by the contact points (green circles), the object moves along with the hand following the black lines and is released at the orange circles.

proposed design allows tracking the hand pose and maintaining the grasp outside the user's FoV. We compare the performance of the proposed design with the LeapMotion quantitatively in two settings: (i) grasp an object, and (ii) move an object during which the hand would briefly be outside of the FoV; Table I shows the success rates. The deployed caging-based grasp significantly surpasses the performance provided by the LeapMotion sensor in grasping various objects. The success rate retains high and consistently when moving objects using the proposed design, whereas the success rate drops significantly using the LeapMotion sensor.

This experiment also indicates that the proposed design is more robust in terms of the grasp types. According to the grasp taxonomy presented in [31], grasping a mug from the handle and a goose toy from the neck are classified as *Power Grip with thumb abducted*, the racket as *Cylindrical Grasp*, and the bowl from the rim as *Extension Grip*. Although the LeapMotion sensor does reasonably well for *Power Grip with thumb abducted*, it performs poorly in the other two types of grasps. In contrast, the grasp using the proposed design performs consistently across all types.

C. Data Logging

The glove-based system is capable of logging various grasp-related data automatically, providing a unique advantage compared to traditional hardware. This subsection demonstrates two types of data that can be collected effectively through the proposed glove-based system.

Hand/Object Trajectory is a useful knowledge in the field of robot Learning from Demonstration, which is usually collected using visual tracking methods. The proposed Vive Tracker and Lighthouse setup using the glove is a low cost and off-the-shelf solution to provide a reliable hand tracking. Diverse object models can be placed in the VR without to prepare physical objects to ensure natural hand trajectory. The quality of the recorded data is shown in Figure 9, demonstrating the reliability of the proposed design.

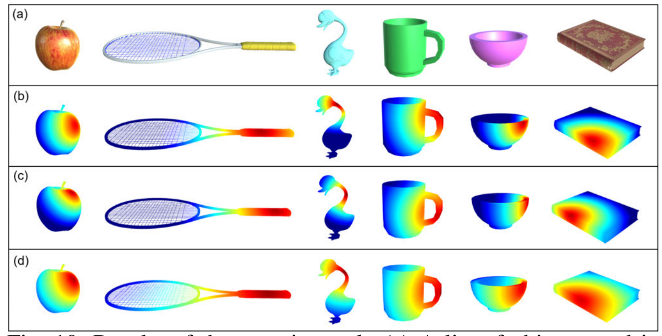


Fig. 10: Results of the grasping task. (a) A list of objects used in the study. (b)(c) Some examples of the grasp performed by different human subjects. (d) The average grasp calculated from multiple human subjects.

Grasping Points are extremely challenging to obtain. Conventional solutions addressing this challenge use computer-vision based method [32], which heavily rely on training data and highly vulnerable to the occlusion between hand and objects. By taking the intersection of the virtual objects and the virtual hand, such data can be elegantly logged using the proposed system. The grasp data shown in Figure 10 contains meaningful information about how human tend to grasp certain objects. This grasping data collected could be used for different domains of research ranging from studying human utilities [33] or robotics grasp planning [34].

V. Conclusion

This paper presents a design to enable a caging-based natural grasp in the VR. The hardware design is based on an easy-to-replicate glove that senses hand pose through a network of 15 IMUs, localizes hand using a Vive Tracker, and provides haptic feedback through 6 vibration motors. We have demonstrated that the design captures fine-grained hand motions. According to the collisions geometry between the virtual hand and virtual objects, a caging-based approach is integrated to determine a stable grasp. The experiment exhibits the effectiveness of the proposed design in grasping various objects, resulting in a significantly higher success rate in grasping and moving objects in VR, compared to the popular LeapMotion sensor. The data logging module also enables the potential of using the proposed glove-based system to collect ground truth grasp data during human manipulations, such as hand trajectories and grasping points, which may facilitate in AI and robotics related research.

Ongoing work includes enriching the forms of haptic feedback and acquiring large-scale grasp data of manipulating various objects, which can be used for training virtual AI agents for robot grasping tasks.

Acknowledgement: The work reported herein is supported by ONR MURI grant N00014-16-1-2007, ARO grant W911NF-18-1-0296, and an NVIDIA GPU donation grant. Zhenliang Zhang is a visiting student at UCLA during the project period, and his advisors Yue Liu and Yongtian Wang at BIT acknowledge the support from a program in China (No. 2017YFB1002504) and an NSF China fund (No. 61731003). The authors thank Eric Peltola and Professor Veronica Santos of Biomechatronics Lab at UCLA MAE Department for useful discussions and assistance with the PCB design.

References

[1] C. Li, W. Liang, C. Quigley, Y. Zhao, and L.-F. Yu, "Earthquake safety training through virtual drills," *IEEE Transactions on Visualization and Computer Graph (TVCG)*, vol. 23, no. 4, pp. 1275–1284, 2017.

[2] J. Lin, Y. Zhu, J. Kubricht, S.-C. Zhu, and H. Lu, "Visuomotor adaptation and sensory recalibration in reversed hand movement task," in *Annual Meeting of the Cognitive Science Society (CogSci)*, 2017.

[3] T. Ye, S. Qi, J. Kubricht, Y. Zhu, H. Lu, and S.-C. Zhu, "The martian: Examining human physical judgments across virtual gravity fields," *IEEE Transactions on Visualization and Computer Graph (TVCG)*, vol. 23, no. 4, pp. 1399–1408, 2017.

[4] C. Aravena, M. Vo, T. Gao, T. Shiratori, and L.-F. Yu, "Perception meets examination: Studying deceptive behaviors in vr," in *Annual Meeting of the Cognitive Science Society (CogSci)*, 2017.

[5] Y. Lang, L. Wei, F. Xu, Y. Zhao, and L.-F. Yu, "Synthesizing personalized training programs for improving driving habits via virtual reality," in *Virtual Reality and 3D User Interfaces (VR)*, 2018.

[6] D. Wang, J. Kubrlcht, Y. Zhu, W. Lianq, S.-C. Zhu, C. Jiang, and H. Lu, "Spatially perturbed collision sounds attenuate perceived causality in 3d launching events," in *Virtual Reality and 3D User Interfaces (VR)*, 2018.

[7] B. Xie, Y. Zhang, H. Huang, E. Ogawa, T. You, and L.-F. Yu, "Exercise intensity-driven level design," *IEEE Transactions on Visualization and Computer Graph (TVCG)*, vol. 24, no. 4, pp. 1661–1670, 2018.

[8] E. G. Kaigom and J. Roßmann, "Developing virtual testbeds for intelligent robot manipulators-an erobotics approach," in *International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[9] T. Bates, K. Ramirez-Amaro, T. Inamura, and G. Cheng, "On-line simultaneous learning and recognition of everyday activities from virtual reality performances," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[10] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[11] J. Lin, X. Guo, J. Shao, C. Jiang, Y. Zhu, and S.-C. Zhu, "A virtual reality platform for dynamic human-scene interaction," in *SIGGRAPH ASIA 2016 Virtual Reality meets Physical Reality: Modelling and Simulating Virtual Humans and Environments*, 2016.

[12] "Leap motion controller." Accessed: 2018-08-31.

[13] "Intel realsense." Accessed: 2018-08-31.

[14] M. Zhou and P. Ben-Tzvi, "Rml glovean exoskeleton glove mechanism with haptics feedback," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 641–652, 2015.

[15] X. Gu, Y. Zhang, W. Sun, Y. Bian, D. Zhou, and P. O. Kristensson, "Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in vr," in *CHI Conference on Human Factors in Computing Systems*, 2016.

[16] H. Liu, X. Xie, M. Millar, M. Edmonds, F. Gao, Y. Zhu, V. J. Santos, B. Rothrock, and S.-C. Zhu, "A glove-based system for studying hand-object manipulation via joint pose and force sensing," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[17] T. Taylor, S. Ko, C. Mastrangelo, and S. J. M. Bamberg, "Forward kinematics using imu on-body sensor network for mobile analysis of human kinematics," in *Engineering in Medicine and Biology Society*, 2013.

[18] G. Santaera, E. Luberto, A. Serio, M. Gabiccini, and A. Bicchi, "Low-cost, fast and accurate reconstruction of robotic and human postures via imu measurements," in *International Conference on Robotics and Automation (ICRA)*, 2015.

[19] G. Ligorio and A. M. Sabatini, "Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: comparative analysis and performance evaluation," *Sensors*, vol. 13, no. 2, pp. 1919–1941, 2013.

[20] H. G. Kortier, J. Antonsson, H. M. Schepers, F. Gustafsson, and P. H. Veltink, "Hand pose estimation by fusion of inertial and magnetic sensing aided by a permanent magnet," *Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 5, pp. 796–806, 2015.

[21] R. Boulic, S. Rezzonico, and D. Thalmann, "Multi-finger manipulation of virtual objects," in *VRST*, 1996.

[22] T. Ullmann and J. Sauer, "Intuitive virtual grasping for non haptic environments," in *Computer Graphics and Applications*, 2000.

[23] M. Prachyabrued and C. W. Borst, "Visual feedback for virtual grasping," in *3DUI*, 2014.

[24] C. W. Borst and A. P. Indugula, "A spring model for whole-hand virtual grasping," *Presence: Teleoperators & Virtual Environments*, vol. 15, no. 1, pp. 47–61, 2006.

[25] I. Choi, E. W. Hawkes, D. L. Christensen, C. J. Ploch, and S. Follmer, "Wolverine: A wearable haptic interface for grasping in virtual reality," in *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[26] H. Kim, M. Kim, and W. Lee, "Hapthimble: A wearable haptic device towards usable virtual touch screen," in *CHI Conference on Human Factors in Computing Systems*, 2016.

[27] I. Choi, H. Culbertson, M. R. Miller, A. Olwal, and S. Follmer, "Grabity: A wearable haptic interface for simulating weight and grasping in virtual reality," in *ACM Symposium on User Interface Software and Technology (UIST)*, 2017.

[28] D. Jain, M. Sra, J. Guo, R. Marques, R. Wu, J. Chiu, and C. Schmandt, "Immersive terrestrial scuba diving using virtual reality," in *CHI Extended Abstracts on Human Factors in Computing Systems*, 2016.

[29] "Vive tracker." Accessed: 2018-08-31.

[30] J. Lin, Y. Wu, and T. S. Huang, "Modeling the constraints of human hand motion," in *Workshop on Human Motion*, 2000.

[31] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, 2016.

[32] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[33] Y. Zhu, Y. Zhao, and S.-C. Zhu, "Understanding tools: Task-oriented object modeling, learning and recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *International Conference on Robotics and Automation (ICRA)*, 2003.