# 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]

* Equal Contributors

1. Center for Vision, Cognition, Learning, and Autonomy, UCLA
2. School of Optics and Photonics, Beijing Institute of Technology
3. International Center for AI and Robot Autonomy (CARA)

## Introduction

Serving as an effective means to train and test subjects in various environments and events, Virtual Reality (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. 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.

Comparisons with previous method:

**(a)** The virtual grasp is **unstable**, reflected by the motion blur, due to the noisy vision-based hand pose sensing approach.

**(b)** Although a stable grasp, the grasp is **unnatural** and does not even contacts the object, as the object simply attaches to the hand once a user triggers the grasp event.

**(c)** The proposed design yields the most **stable** and **natural** grasp among three. The red areas of the hand model indicate the forces exerted during manipulations.

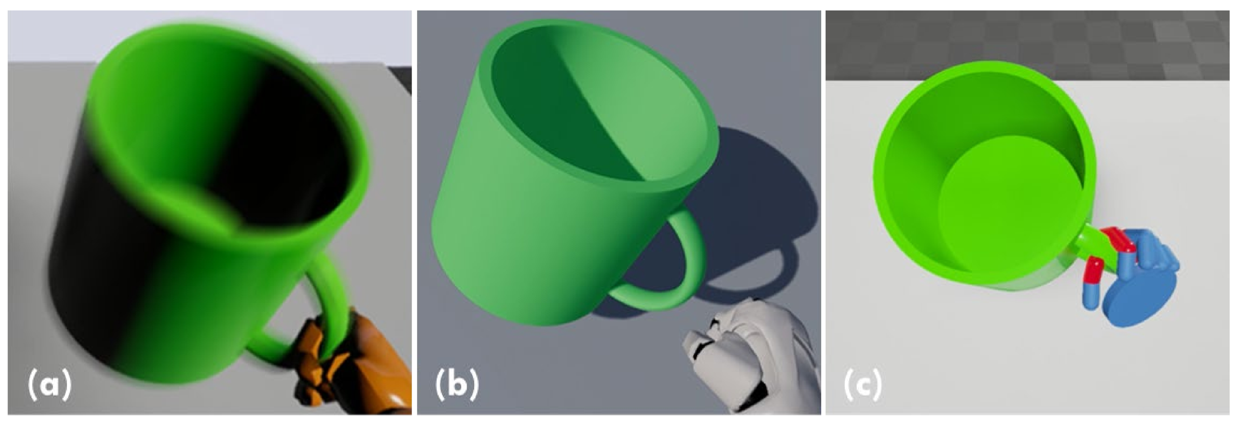


**Fig 1.** Grasp the same object in VR using (a) a LeapMotion sensor, (b) an Oculus Touch controller, and (c) the proposed glove system.

## System Architecture and Prototyping

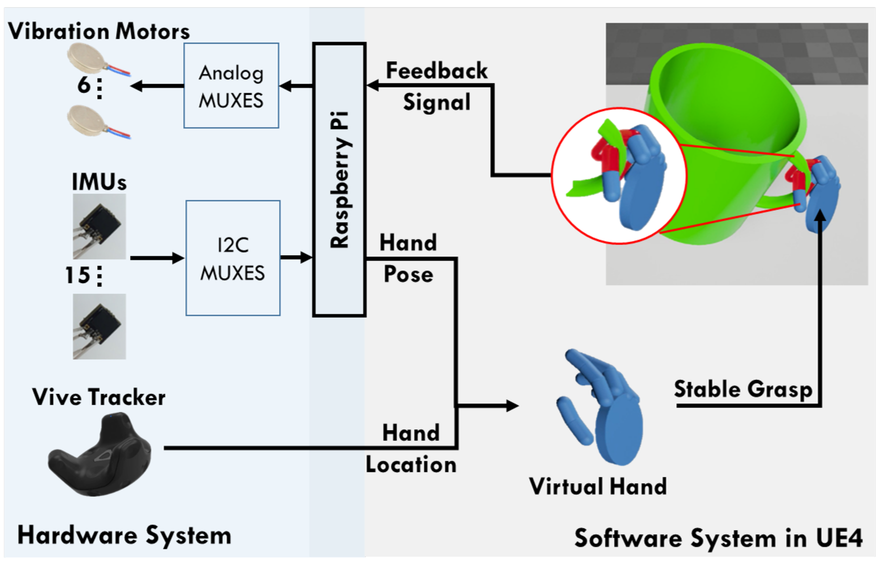



**Fig 2.** Schematics of the system architecture. When a stable grasp is formed based on the contact geometry, the virtual object follows the hand's movement.

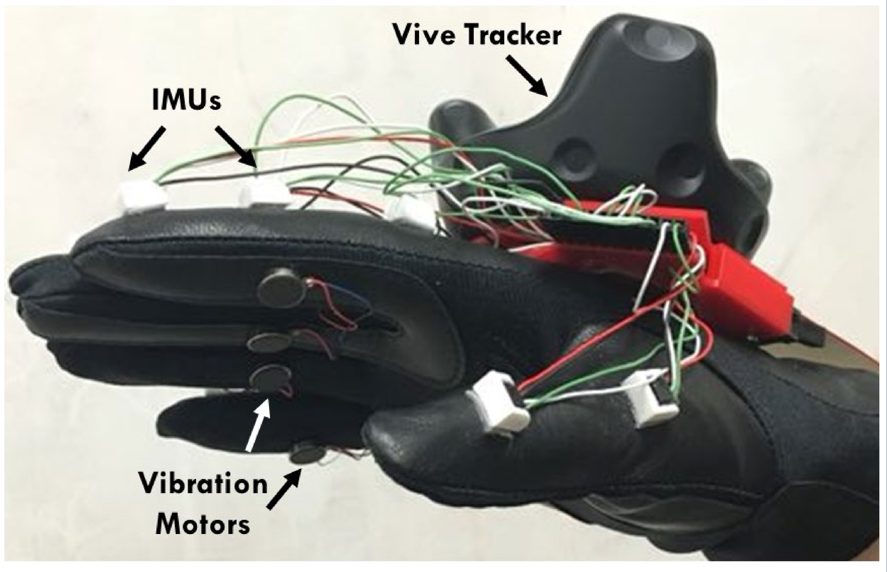



**Fig 3.** The prototype consists of a network of 15 IMUs, a network of 6 vibration motors, and a Vive Tracker.

**Pose Sensing:** 15 Bosch BNO055 9DoF IMUs over a pair of I2C multiplexers. 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.

**Hand Localization:** 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 using the HTC Lighthouse, a laser sweeping device.

**Haptic Feedback:** 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.

## Virtual Hand Construction

With these constraints, the 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:

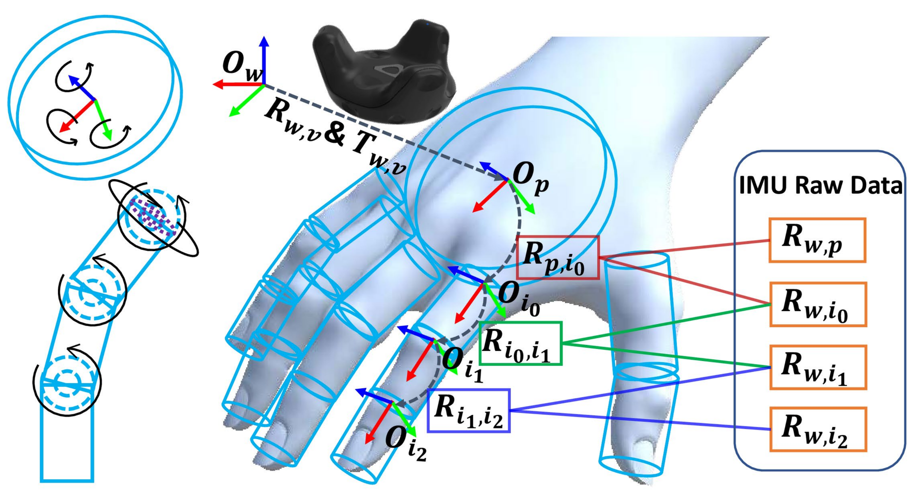



**Fig 4.** The structure of the virtual hand model, in which every phalanx is modeled as a small cylinder. The pose of each phalanx can be computed using IMUs, and a Vive Tracker directly tracks the pose of the palm.

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

Rotation matrices $R$ are obtained from the IMUs placed on palm, proximal, middle, and distal phalanx produce quaternion $q_p, q_1, q_2$, and $q_3$:

$$ R_{MCP} = f\left(\frac{q_1}{q_p}\right),\ R_{PIP} = f\left(\frac{q_2}{q_1}\right),\ R_{DIP} = f\left(\frac{q_3}{q_2}\right) $$

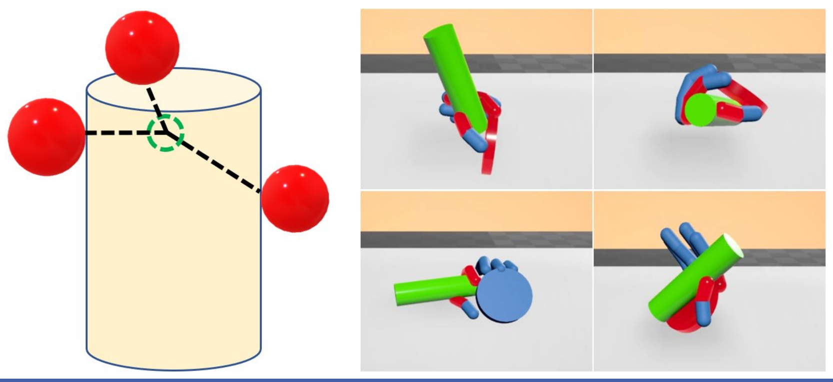
Translation matrices $T$ is defined according to the dimension of the hand.

We design a hand calibration routine to eliminate IMU drifting by recording the relative pose between the glove frame and the world frame, and multiplying the inverse of the relative pose to cancel out the differences.

## Caging-based Stable Grasp in UE4

1. Start 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 5).
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 5). 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 physical property of this object is turned off so that the object will move together with the hand. Through 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.

**Fig 5.** Collision detection of a grasp. (left) 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. (right) Examples of grasping small cylinders in various ways.



## IMU Evaluation

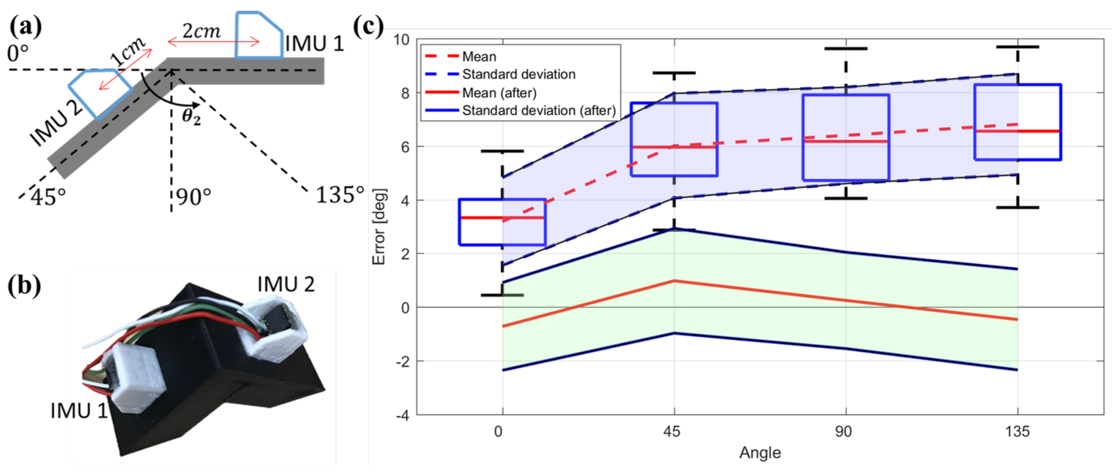



**Fig 6.** (a) The experimental schematic. (b) An exemplary setup of the physical device using 90° joint angle. (c) The errors of estimated joint angles under four settings.

Using four rigid bends with fixed angles of $0^\circ, 45^\circ, 90^\circ$, and $135^\circ$ to simulate a finger's bending. Two IMUs are placed 2cm away behind the bend and 1cm ahead of the bend to recover the joint angle of the bend. The results under-perform as the bending angle increases.

A possible compensation for such error. We apply a least squared fit $y = 0.0249x + 3.9068$ can reduce the sensing error to about 1 degree.

## User Experience

**Data Logging:**

- By taking the intersection of the virtual objects and the virtual hand, grasping points can be logged elegantly.
- The system is a low cost and off-the-shelf solution to provide a reliable hand and objects tracking.

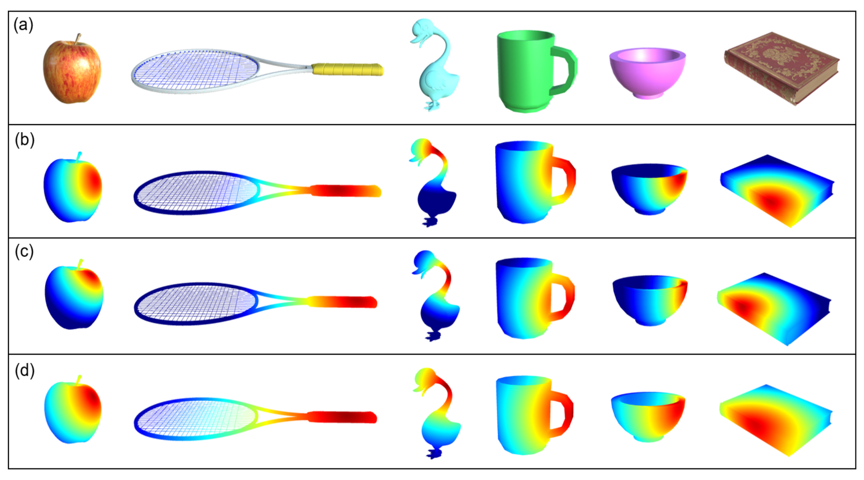


**Fig 7.** Results of the collected grasping areas.

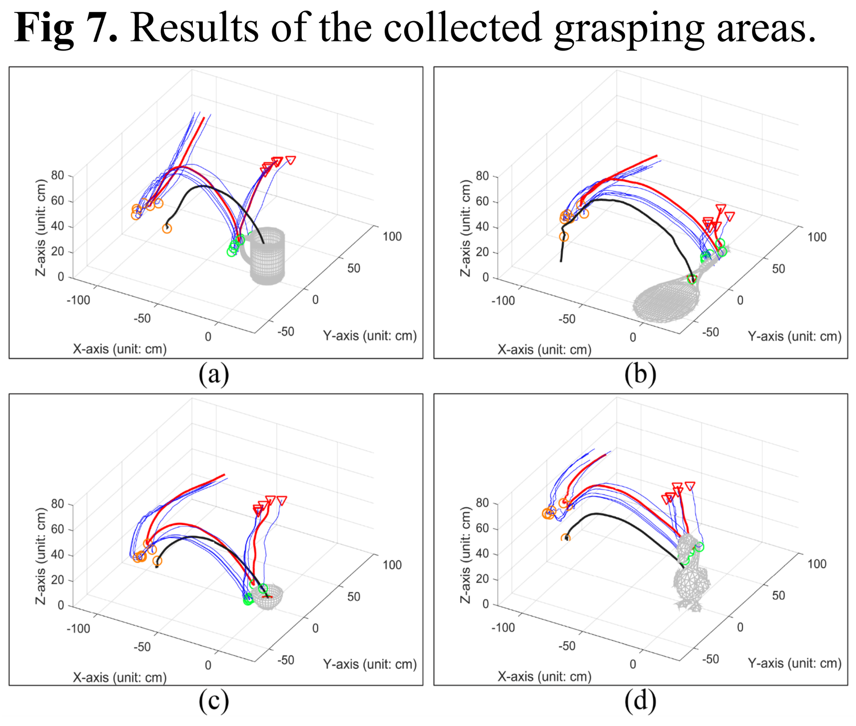


**Fig 8.** Various hand and object trajectories collected using the proposed design in VR. Red line: hand movement; blue lines: fingertips' trajectories; green circles: contact points; black line: object movement.

Grasping a mug from the handle and a goose toy from the neck are classified as *Power Grip*, the racket as *Cylindrical Grasp*, and the bowl from the rim as *Extension Grip*.

LeapMotion sensor does well for the first type, but not the other two types of grasps. The proposed design performs consistently across all three types.

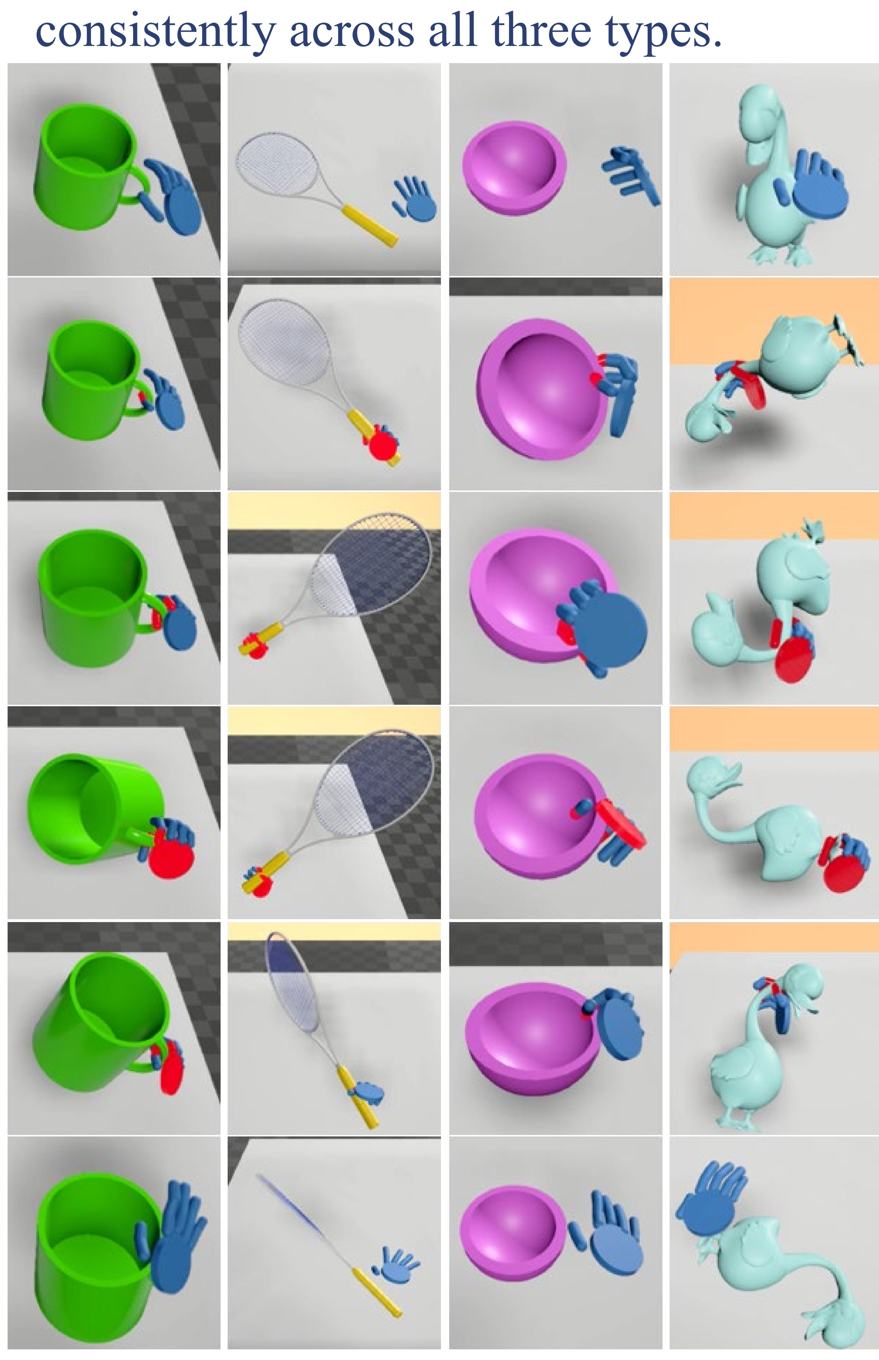


**Fig 9.** Different grasps of a mug, a tennis racket, a bowl, and a goose toy.

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).