2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Adaptive Precision-Enhancing Hand Rendering for Wearable Fingertip Tracking Devices

Hyojoon Park[1] and Jung-Min Park[2]

***Abstract*— We introduce a 3D hand rendering framework to reconstruct a visually realistic hand from a set of fingertip positions. One of the key limitations of wearable fingertip tracking devices used in VR/AR applications is the lack of detailed measurements and tracking of the hand, making the hand rendering difficult. The motivation for this paper is to develop a general framework to render a visually plausible hand given only the fingertip positions. In addition, our framework adjusts the size of a virtual hand based on the fingertip positions and device's structure, and reduces a mismatch between the pose of the rendered and user's hand by retargeting virtual finger motions. Moreover, we impose a new hinge constraint on the finger model to employ a real-time inverse kinematic solver. We show our framework is helpful for performing virtual grasping tasks more efficiently when only the measurements of fingertip positions are available.**

I. INTRODUCTION

Despite the recent advances by academia and industry in wearable hand motion-capturing devices featuring a high number sensors, in many cases tracking only the fingertips is sufficient since they are the ones most often used for grasping, manipulation, and probing the environment [1]. For similar reasons, many wearable devices focus only on tracking the fingertips, some with haptic feedback ([2]–[6]).

However, rendering a virtual hand becomes non-trivial for fingertip tracking devices due to their lack of detailed measurements. The apparent solution is to utilize more accurate and larger number of sensors (e.g., using IMUs [7], soft sensors [8], and IMUs/soft sensors [9]), yet with more sophistication and usually higher manufacturing costs. More practically, some researchers resort to third party vision-based tracking solutions ([10], [11]) or rather compromised yet practical rendering solutions using primitive shapes (e.g., spheres only at the fingertips [12], [13], and cylinders with a sphere to represent only the distal and intermediate phalanges [14]) while focusing more on their core research area such as cutaneous haptic feedback, etc.

In addition, fingertip tracking devices relying on forward kinematics for computing the fingertip positions may accumulate measurement and parametric errors towards the fingertips [15]. Even if the error is negligible, some devices may exhibit a mismatch between the intended pose and the actual pose due to physical constraints of the device [16]. These factors contribute to making the hand rendering more challenging.

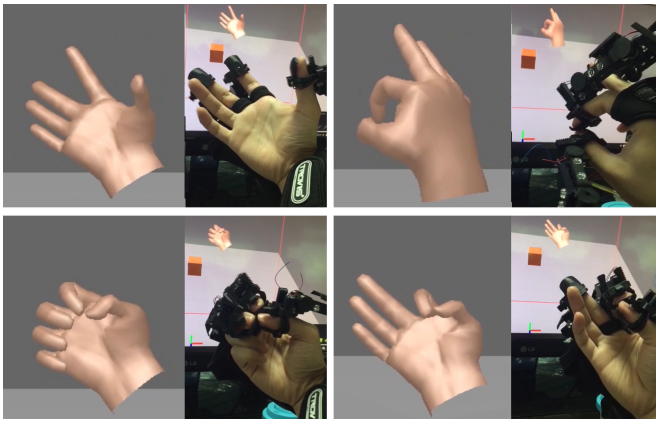


Fig. 1: Hand poses rendered using the fingertip positions from a wearable fingertip tracking device.

Nevertheless, only few studies focus on the fingertip positions based hand rendering. [17] approximates the hand shape in grasping tasks using postural synergies but assumes the fingertip positions measured are ideally accurate, which may not be true in practice. Similarly, [15] estimates the hand pose from fingertip positions using an approximated relationship between the angles of finger joints, but this relationship breaks when the finger is displaced by an external force. [18] estimates the hand pose by measuring MCP flexion/extension motions in addition to the fingertip positions, but they are not easily measurable for most fingertip tracking devices and introduces additional computations. Similarly, [19] measures additional hand motions using IMU and Leap Motion Camera [20] to complement the fingertip position measurements.

In this paper, we focus on rendering the virtual hand given only the fingertip positions to replace the rather compromised rendering solutions using primitive shapes [12]–[14]. Moreover, we assume practical cases where the given fingertip position inputs may not be ideally accurate due to practical measurement/parametric errors and/or device's physical constraints. As a result, a mismatch between the rendered hand and user's hand may exist if the given inputs are directly used for rendering. Our framework can deduce the hand pose with a reasonable visual plausibility, and we show that users benefit from it during virtual grasping tasks.

This work was supported in part by the Global Frontier R&D Program on "Human-centered Interaction for Coexistence" funded by National Research Foundation of Korea grant funded by the Korean Government(MSIP) (2011-0031425) and by the Korea Institute of Science and Technology (KIST) Institutional Program under Project 2E30280.

[1]Hyojoon Park is an intern researcher at Korea Institute of Science and Technology (KIST), Seoul, Korea hjoonpark.us@gmail.com

[2]Jung-Min Park is a Senior Research Scientist at the Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology (KIST), Seoul, Korea pjm@kist.re.kr

II. System Overview

Our rendering framework after the two calibration steps (discussed in Sec. II-B-II-C) is illustrated in Fig. 2. Here, we illustrate an example case where a user's hand is in grasping pose in which the fingertips are touching each other (Fig. 2e), but the measured fingertips are further apart (Fig. 2a) due to possible inaccurate sensor measurements/calibrations or physical constraints from the device. We assume such mismatch is common in practice which needs to be compensated to enhance the grasping precision.

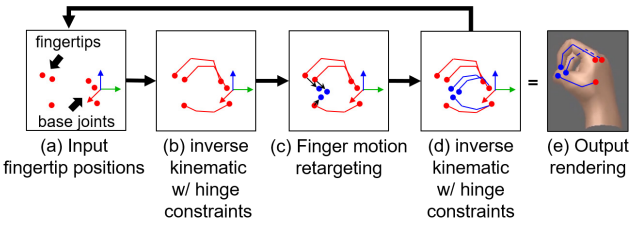


Fig. 2: Hand rendering pipeline after the two calibrations.

The inputs are the fingertip positions w.r.t. a local coordinate system (Fig. 2a). Next, inverse kinematic (IK) with our new hinge constraints is solved to obtain the rest of the joint poses (Fig. 2b). Then, the joint angles of each finger are retargeted (Sec. II-C), and the new fingertip positions are computed which match the user's intended pose better (Fig. 2c). Lastly, IK with hinge constraints is solved again from the new retargeted fingertip positions (Fig. 2d-e). This process runs in real-time ($>$ 30Hz) and is repeated each frame.

A. Device and hand model for implementation

We briefly introduce the fingertip tracking device and hand model used throughout this paper.

1) Device: For implementation, we use a wearable 12 DoF 3-finger exoskeleton device CHICAP [6]. The device is attached to a user's wrist via its wristband and clipped to each fingertip (Fig. 3), and computes the fingertip positions of thumb, index, and middle fingers w.r.t. the wrist via forward kinematics. The global position of the wrist is tracked using HTC VIVE Tracker [21].

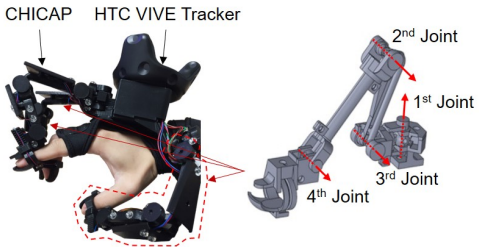


Fig. 3: CHICAP, a 12 DoF 3-finger exoskeleton hand motion-capturing device.

2) Hand model: For rendering, we use the generic hand model provided by 3Gear Systems [22] comprised of 16 joints for its simple structure and usage. The model takes the positions of the 16 joints as inputs and outputs a skinned mesh. Each finger is modeled as a 4 DoF chain of three serially-connected links (Fig. 4). The base joint (CMC for thumb, MCP for index/middle fingers) of fingers is modeled as a 2 DoF ball joint with a fixed axial rotation, and the rest of the joints are modeled as 1 DoF hinge joints. The wrist is modeled using a 3 DoF ball joint, and the ring and pinky fingers simply copy the motion of the middle finger.

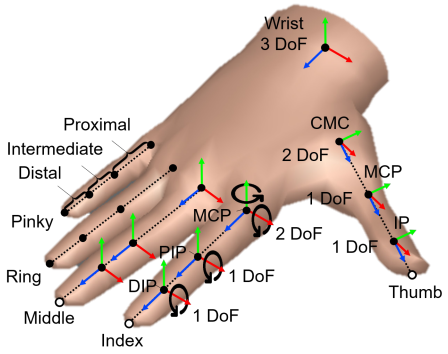
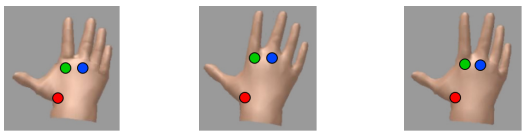


Fig. 4: The generic hand model from 3Gear System.

B. Model adjustment: the first calibration

Each of the two calibration steps introduced in this paper is performed instantly upon a key press using only the simple algebraic computations. The *first calibration* is performed on the user's hand in an open pose and adjusts the size of the hand model to a user's hand. Specifically, once a user wears CHICAP with the hand flat open, (1) the base joint positions of the hand model and (2) the lengths of the finger segments are calibrated.

1) Base joint positions: The base joint positions are required for IK. While the fingertip positions are provided by the device, the base joint positions are not and vary for every users. Nevertheless, they play a dominant role in determining the size of the hand model, and their incorrect positions can render the hand unnatural (Fig. 5).

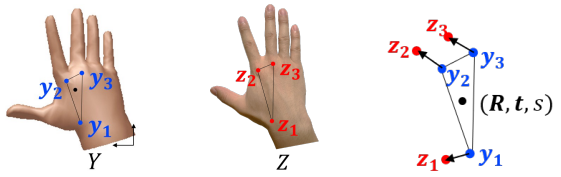


(a) (b) (c)

Fig. 5: The base joints dominantly determines the palm size and its natural look: (a) too small, (b) too large, or (c) natural.

To determine the optimal base joint positions of the hand model, we rotate $\boldsymbol{R} \in SO(3)$, translate $\boldsymbol{t} \in E(3)$, and scale $s \in \mathbb{R}$ the generic hand model's set of base joints at its rest pose and align it with the user's hand (Fig. 6). Then, the optimal rigid transformation $(\boldsymbol{R}^*, \boldsymbol{t}^*, s^*)$ is obtained by solving the least-squares problem involving rigid motion with scale using singular value decomposition (SVD) [23].

Concretely, let $Y = \{\boldsymbol{y}_i\}_{i=1}^3$ be a set of base joint positions of the hand model in rest pose w.r.t. its local coordinates system (Fig. 6a), and $Z = \{\boldsymbol{z}_i\}_{i=1}^3$ a set of target



(a) (b) (c)

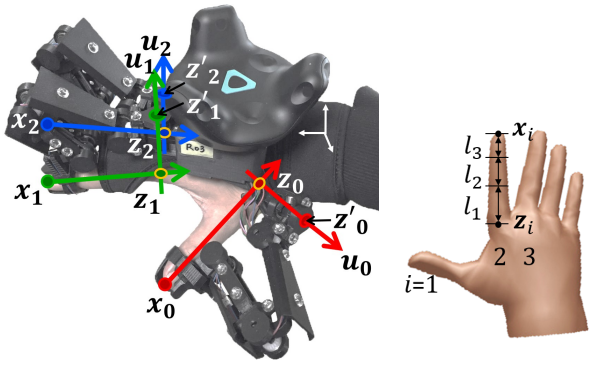
Fig. 6: We align Y (base joints of hand model) with Z (estimated base joints of user's hand) optimally using a rigid transformation with scaling.

base joint positions defined using the geometric relation between a user's fingertip positions and device's structure (Fig. 6b), discussed later in this section. The objective is to find $(\boldsymbol{R}^*, \boldsymbol{t}^*, s^*)$ which optimally aligns Y with Z in the least squares sense (Fig. 6c):

$$(\boldsymbol{R}^*, \boldsymbol{t}^*, s^*) = \underset{\substack{\boldsymbol{R}\in SO(3),\\ \boldsymbol{t}\in E(3), s\in\mathbb{R}}}{\arg\min} \sum_{i=1}^{3} w_i || \left(s\boldsymbol{R}\boldsymbol{y}_i + \boldsymbol{t}\right) - \boldsymbol{z}_i ||^2, \quad (1)$$

where $w_i \in \mathbb{R}$ is a weight indicating the amount influence of $\boldsymbol{z}_i$ on $\boldsymbol{y}_i$.

The target positions Z required for (1) are the estimated base joint positions of the user's hand. Z is estimated at runtime using the geometric relationship between the fingertip positions and the device structure. For CHICAP specifically, we define $\boldsymbol{z}_i$ to be at a point of intersection between the up-vector $\boldsymbol{u}_i$ passing through the device's base position $\boldsymbol{z}'_i$, and a line perpendicular to $\boldsymbol{u}_i$ passing through the user's fingertip position $\boldsymbol{x}_i$ as in Fig. 7a. Here, $\{\boldsymbol{z}'_i\}_{i=1}^3$ and $\{\boldsymbol{u}_i\}_{i=1}^3$ are geometrically measurable and constant for CHICAP. Then, the closed-form solution to (1) is obtained by solving a rather well-established *absolute orientation problem* [24].



(a) (b)

Fig. 7: (a) The geometry of CHICAP is used for estimating Z. (b) The finger length is divided into three segments.

2) Finger lengths: For finger i, the length of a user's finger $||\boldsymbol{z}_i - \boldsymbol{x}_i||$ is divided into three segments by the predetermined segment ratio $l_1 : l_2 : l_3$ (Fig. 7b). We observe the ratio $1 : 1.3 : 2.2$ renders natural looking hand for various finger lengths, also similar to the average ratio obtained from 66 adults' hand in [25] (i.e., $1 : 1.1\text{-}1.5 : 2.1\text{-}2.6$).

C. Motion retargeting: the second calibration

1) Motion retargeting: An example of a mismatch is when a user makes a grasping pose with the fingertips touching each other, but the fingertips of a rendered hand do not (Fig. 8); or, a user makes a flat open pose, but the rendered hand is slightly curved. Such mismatch deteriorates the grasping precision, so we compensate for it by scaling and adding a bias to each joint angle similar to the motion warping technique introduced in [26]. Assuming the hand pose changes continuously between the open and grasping poses, the fingertips of the rendered hand should lie flat on the same plane when the user's hand is in open pose (Fig. 9a) and end up close to each other in grasping pose (Fig. 9b).

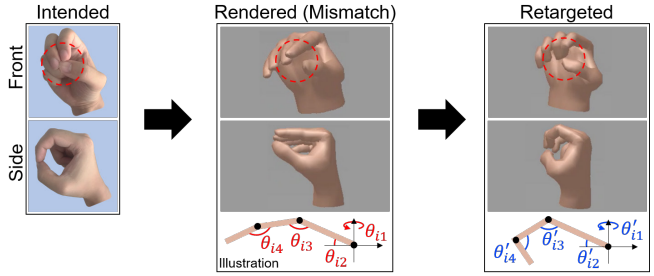
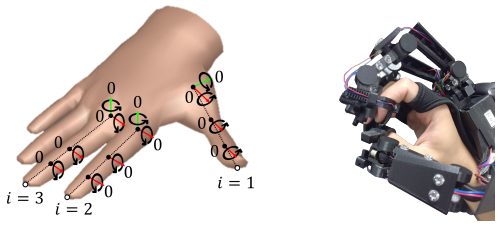


Fig. 8: A mismatch is reduced by retargeting the angles of the finger joints to better match a user's hand pose.

Concretely, let $\boldsymbol{\theta}_i \in \mathbb{R}^4$ be a vector of four joint angles of finger i computed from IK, and $\boldsymbol{\theta}'_i \in \mathbb{R}^4$ a vector of the corresponding retargeted angles (Fig. 8), computed by:

$$\boldsymbol{\theta}'_i = \boldsymbol{\alpha}_i \boldsymbol{\theta}_i + \boldsymbol{\beta}_i, \quad (2)$$

where $\boldsymbol{\alpha}_i = \text{diag}(\alpha_1, \alpha_2, \alpha_3, \alpha_3)$ and $\boldsymbol{\beta}_i \in \mathbb{R}^4$ are the retargeting parameters to be calibrated for each user.



(a) (b)

Fig. 9: (a) Hand model at zero configurations (open pose). (b) User's hand in a grasping pose

2) The second calibration: The *second calibration* is performed first on the open pose (Sec.II-C.2.a) then on the grasping pose (Sec. II-C.2.b), and the objective is to determine $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ (i.e., 8 unknowns) by solving (2) using the known two pairs of current $\boldsymbol{\theta}_{ij}$ and target joint angles $\boldsymbol{\theta}'_{ij}$: $(\boldsymbol{\theta}_{i1}, \boldsymbol{\theta}'_{i1})$ and $(\boldsymbol{\theta}_{i2}, \boldsymbol{\theta}'_{i2})$.

a) The first pair (open pose): For the first pair $(\boldsymbol{\theta}_{i1}, \boldsymbol{\theta}'_{i1})$, the values for $\boldsymbol{\theta}_{i1}$ are obtained from IK on the virtual hand rendered in an open pose, which may not look

perfectly flat open if a mismatch exists. On the other hand, the values for $\boldsymbol{\theta}'_{i1}$ are set to zeros which renders the hand ideally flat open (Fig. 9a). That is, if a user's hand is in a flat open pose, but the virtual hand appears to be slightly curved, we make it flat by retargeting the current joint angles $\boldsymbol{\theta}_{i1}$ of the fingers to the target angles $\boldsymbol{\theta}'_{i1}$ of zeros.

b) The second pair (grasping pose): We determine the values of $(\boldsymbol{\theta}_{i2}, \boldsymbol{\theta}'_{i2})$ from the user's hand in a grasping pose (Fig. 9b). Similar to the open pose case, $\boldsymbol{\theta}_{i2}$ is obtained from IK on the hand rendered in a grasping pose, in which the fingertips may not lie close to each other if a mismatch exists (Fig. 8). Then, the target joint angles $\boldsymbol{\theta}'_{i2}$ are set to the values that make the fingertips in grasping pose close to each other.

The values for $\boldsymbol{\theta}'_{i2}$ are set to the joint angles of the virtual hand artificially made into an ideal grasping pose, generated from the steps illustrated in Fig. 10:

1) Compute the weighted centroid $\bar{\boldsymbol{X}}$ of the fingertips: $\bar{\boldsymbol{X}} = \sum_{i=1}^{3} m_i \boldsymbol{x}_i / \sum_{j=1}^{3} m_j$.
2) Compute the weighted centroid $\bar{\boldsymbol{Y}}$ of the base joints: $\bar{\boldsymbol{Y}} = \sum_{i=1}^{3} n_i \boldsymbol{y}_i / \sum_{j=1}^{3} n_j$.
3) Determine the new fingertip position $\boldsymbol{X}'$ (same value for all three fingers) using interpolation between $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{Y}}$ by $s \in [0, 1]$: $\boldsymbol{X}' = (1 - s)\bar{\boldsymbol{X}} + s\bar{\boldsymbol{Y}}$.
4) Solve IK between the new fingertip $\boldsymbol{X}'$ and the base joint $\boldsymbol{y}_i$ (Sec. II-D).
5) Finally, $\boldsymbol{\theta}'_{i2}$ are set to the joint angles of the retargeted hand (blue in Fig. 10). $m_i, n_i > 0$ are the predefined weights for the two centroids $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{Y}}$, respectively.

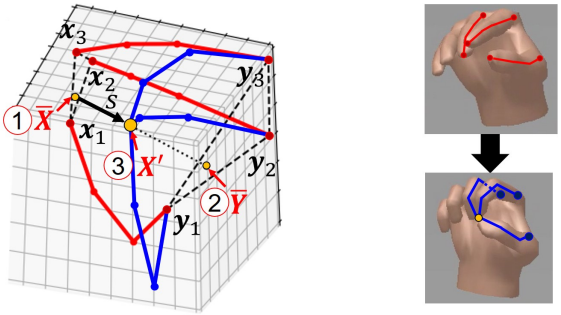


Fig. 10: The current fingertip positions (red) are retargeted to better match the user's intended grasping pose (blue).

After the calibrations, the joint angles of each finger are mapped to the retargeted angles in each frame by (2). As a result, the rendered hand follows the user's hand pose which continuously changes between the open and grasping poses, reducing a mismatch between the rendered and intended poses.

D. Inverse kinematics with hinge constraints

In general, IK problems can be solved using: analytical, Jacobian, Newton, statistical, data-driven, and heuristic methods [27]. For real-time applications, we employ one of the heuristic methods named Forward and Backward Reaching Inverse Kinematics (FABRIK) [28] for its simplicity and fast convergence. FABRIK is a real-time IK solver, which treats finding the joint locations as a problem of finding a point on a line using forward and backward iterations in the most efficient manner.

1) Limitations of unconstrained FABRIK: FABRIK assumes all links to be serially-connected with ball joints, which, however, is not the case for our finger model with hinge joints. Thus, using FABRIK without constraints may render the finger unnaturally bent downward or sideways if that is the most efficient direction for solving IK (Fig. 11).

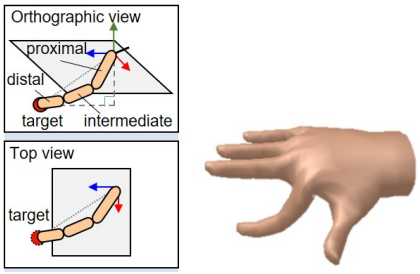


Fig. 11: Unconstrained FABRIK may render the finger unnaturally bent downward or sideways.

In fact, a hinge constraint for FABRIK is introduced in [28]; however, this method enforces the constraint to one joint at a time and treats its neighboring joints as ball joints. Thus, as one joint becomes a hinge joint its neighboring joints become unconstrained again, which can make a part of the finger bend sideways.

2) New hinge constraints for FABRIK: The key idea is to reorient a set of consecutive hinge joints in a straight line all together prior to executing FABRIK. This makes FABRIK problem 2D where all joints lie on a plane and prevents them from deviating from it (i.e., sideways). The method in [28] similarly reorients joints prior to FABRIK, but only maintains three joints on a plane at a time; thus, for our finger model with four joints, either the first or last joint can undesirably deviate from the plane.

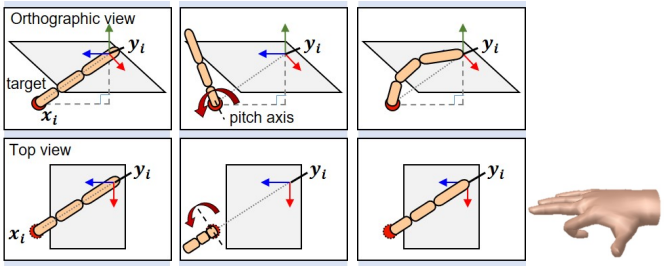


Fig. 12: FABRIK with the new hinge constraints.

The new hinge constraint are imposed as the following. (1) Place the fingertip of the hand model on a measured fingertip position $\boldsymbol{x}_i$, with the rest of the joints in line towards the base joint $\boldsymbol{y}_i$ (Fig. 12a). (2) Rotate the finger around its local pitch axis at the fingertip to ensures it curves only upward when FABRIK is solved (Fig. 12b). (3) FABRIK is solved as a 2D problem, and all four joints of a finger remain on the

same plane while bending naturally as in Fig. 12c-12d. This process is repeated for each finger. We observe setting the pitch rotation angle to 80-90° consistently renders the finger natural.

III. Experimental Results

Given only the fingertip positions as inputs, the easiest yet unrealistic rendering solution is to represent only the fingertips using spheres as in [12], [13]. On the other hand, a visually plausible-looking hand can be rendered without much sophistication using our framework. We conduct user experiments to validate that given the same inputs, the rendered hand using our framework is reasonably realistic to be more helpful and preferable for virtual grasping tasks.

A. Experiment Setup

1) Grasping task: We ask a group of users to perform a set of tasks, in which the objective is to grasp and relocate a virtual rigid ball in the virtual environment. The time duration of each task is recorded.

a) Simulator: The virtual environment is built using OpenGL/C++ with the minimum physics simulation (i.e., frictionless with only the gravity). The simulation time-step (0.33s) and mass/damping of each object are selected such that the numerical stability is conserved. Contact forces among the objects are modeled as penalty-based, but the physical interactions between the hand and objects are neglected. Instead, we use a geometrical grasping criteria; the ball is considered grasped if the bounding spheres of all three fingertips are on its surface within a small margin. When the ball is grasped, it follows a trajectory of the palm. When one of the fingertips leaves the surface, the ball is released from the hand and falls by gravity.

For implementation, we set the heuristic parameters $w_i = 1$, $m_i = n_i = 1$, and $s = 0.4$ by which we observe the hand is rendered robustly natural for various hand sizes.

b) Task Settings: We prepare two types of hand representations (Fig. 13a): (1) spheres at each fingertip and a palm, and (2) full hand rendering using our framework, and two sizes of virtual balls (Fig. 13b): (1) 100 and (2) 50 [mm] in diameter. Combining these gives four different settings.

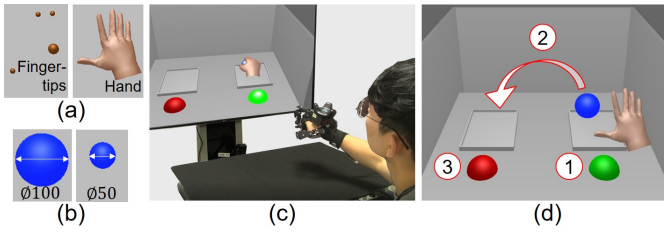


Fig. 13: (a) Two types of hand representations, (b) two sizes of virtual balls, and (c) the experiment setup. (d) The objective is to grasp and relocate the blue ball from the right to left tray.

c) Instructions: A user wears CHICAP on the right hand and sits in front of a flat screen where the virtual world is rendered (Fig. 13c). Then, the two calibration steps are performed. Prior to the experiment, each user gets familiarized with the device for 10 minutes.

Next, the user completes a set of 20 tasks, each with a setting randomly selected from the four settings. Each task is comprised of the following steps (Fig. 13d): (1) touch the green button to start the timer, (2) relocate the blue ball, and (3) touch the red button to stop the timer.

2) User survey: Upon completion, the user answers two sets of USE (Usefulness, Satisfaction, and Ease of use) questionnaires [29] to give their subjective assessments on the sphere and hand renderings, respectively. USE questionnaire has total of 30 items under four categories: Usefulness, Ease of Use, Ease of Learning, and Satisfaction. The scores are averaged for each category.

B. Results and Discussion

Fourteen users (9 males and 5 females) with average age of 27.2±3.7 year-old, without any physical or mental difficulties, have participated.

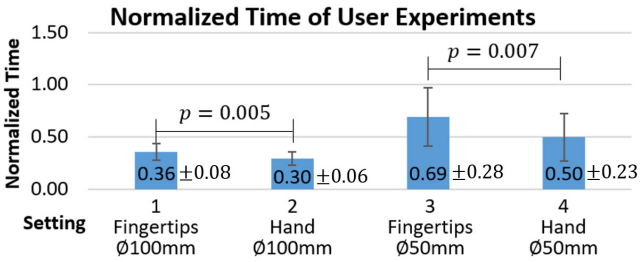


Fig. 14: The normalized mean task time and the standard deviation of the four different task settings.

1) Effectiveness: First, using our framework shortened the normalized mean task time by 17.08% for the large ball (100mm in diameter), and 28.14% for the small ball (50mm in diameter), compared to the sphere rendering (Fig. 14). This implies the rendered hand is visually realistic enough that users have benefited from it.

Second, the larger decrease in task time for the small ball (28.14%) compared to the larger ball (17.08%) presumably implies the motion retargeting improves the grasping precision and is especially helpful for small objects since the retargeting bends in the fingertips closer to each other during grasping (Fig. 8).

Lastly, the paired sample t-test at the $\alpha = 0.05$ level of significance validates the significance in the differences of normalized mean task times between the setting 1 and 2 ($p = 0.005$), and the setting 3 and 4 ($p = 0.007$), respectively.

2) Usability: The descriptive statistics of the scoring is summarized in Fig. 15. The mean scores of all four categories are higher for our rendering. This implies the rendered hand feels realistic enough to make the grasping tasks easier, and the users are satisfied using it.

	Fingertip rendering (given)				Hand rendering (ours)			
Category	Mean	SD	Min	Max	Mean	SD	Min	Max
Usability	4.28	1.59	2	7	4.79	1.50	2	7
Ease of Use	3.97	1.52	1	7	5.14	1.54	1	7
Ease of Learning	5.20	1.39	2	7	6.02	1.29	2	7
Satisfaction	4.07	1.47	1	7	5.04	1.35	3	7

Fig. 15: The descriptive statistics of the USE questionnaires.

Lastly, the paired sample t-test at the $\alpha = 0.05$ level of significance validates the significance in the differences of the mean scores between the hand and sphere rendering for every categories (Table I).

TABLE I: Paired samples test of USE questionnaires at the $\alpha = 0.05$ level of significance.

Category	Score	DoF	p-value
Usability	2.161	110	0.033
Ease of Use	6.373	152	0.000
Ease of Learning	3.624	55	0.001
Satisfaction	5.074	99	0.000

IV. Conclusion and future work

We observed the users can benefit from our framework presumably due to the rendered hand visually matching the size and pose of the user's hand. For future work, a thorough quantitative analysis on the reconstruction accuracy may be helpful. The quantitative accuracy is not investigated here since our primary focus is to render the hand that is sufficiently visual-realistic to replace the compromised abstract hand representations, given the same measurement inputs. Moreover, we expect to extend this work to vision-based fingertip trackers. The current work relies on the known geometry of the device to estimate the base joint positions of the user's hand, so a different approach would be needed for the same purpose.

Acknowledgments

This work was supported in part by the Global Frontier R&D Program on "Human-centered Interaction for Coexistence" funded by National Research Foundation of Korea grant funded by the Korean Government (MSIP) (2011-0031425) and by the Korea Institute of Science and Technology (KIST) Institutional Program under Project 2E30280.

References

[1] C. Pacchierotti, S. Sinclair, M. Solazzi, A. Frisoli, V. Hayward, and D. Prattichizzo, "Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives," *IEEE transactions on haptics*, vol. 10, no. 4, pp. 580–600, 2017.

[2] D. Gomez, G. Burdea, and N. Langrana, "Integration of the rutgers master ii in a virtual reality simulation," in *Proceedings Virtual Reality Annual International Symposium'95*. IEEE, 1995, pp. 198–202.

[3] M. Zhou and P. Ben-Tzvi, "Rml glove—an exoskeleton glove mechanism with haptics feedback," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 641–652, 2014.

[4] P. Agarwal, R. R. Neptune, and A. D. Deshpande, "A simulation framework for virtual prototyping of robotic exoskeletons," *Journal of biomechanical engineering*, vol. 138, no. 6, 2016.

[5] 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 *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2016.

[6] Y. Lee, M. Kim, H. Kim, D. Lee, and B. You, "CHICAP: low-cost hand motion capture device using 3d magnetic sensors for manipulation of virtual objects," in *Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH, Emerging Technologies, Vancouver, BC, Canada*, 2018, pp. 4:1–4:2.

[7] "Avatar vr." [Online]. Available: https://avatarvr.es

[8] W. Park, K. Ro, S. Kim, and J. Bae, "A soft sensor-based three-dimensional (3-d) finger motion measurement system," *Sensors*, vol. 17, no. 2, p. 420, 2017.

[9] Y. Lee, M. Kim, Y. Lee, J. Kwon, Y.-L. Park, and D. Lee, "Wearable finger tracking and cutaneous haptic interface with soft sensors for multi-fingered virtual manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 67–77, 2018.

[10] L. Meli, S. Scheggi, C. Pacchierotti, and D. Prattichizzo, "Wearable haptics and hand tracking via an rgb-d camera for immersive tactile experiences," in *ACM SIGGRAPH 2014 Posters*, 2014, pp. 1–1.

[11] L. Meli, C. Pacchierotti, G. Salvietti, F. Chinello, M. Maisto, A. De Luca, and D. Prattichizzo, "Combining wearable finger haptics and augmented reality: User evaluation using an external camera and the microsoft hololens," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4297–4304, 2018.

[12] V. Yem, R. Okazaki, and H. Kajimoto, "Fingar: combination of electrical and mechanical stimulation for high-fidelity tactile presentation," in *ACM SIGGRAPH 2016 Emerging Technologies*, 2016, pp. 1–2.

[13] 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 *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2017.

[14] S. B. Schorr and A. M. Okamura, "Fingertip tactile devices for virtual object manipulation and exploration," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2017.

[15] G. Hillebrand, M. Bauer, K. Achatz, G. Klinker, and A. Oferl, "Inverse kinematic infrared optical finger tracking," in *9th International Conference on Humans and Computers*. Citeseer, 2006.

[16] C. Hansen, F. Gosselin, K. B. Mansour, P. Devos, and F. Marin, "Design-validation of a hand exoskeleton using musculoskeletal modeling," *Applied ergonomics*, vol. 68, pp. 283–288, 2018.

[17] S. Mulatto, A. Formaglio, M. Malvezzi, and D. Prattichizzo, "Animating a synergy-based deformable hand avatar for haptic grasping," in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 2010, pp. 203–210.

[18] M. Bianchi, P. Salaris, A. Turco, N. Carbonaro, and A. Bicchi, "On the use of postural synergies to improve human hand pose reconstruction," in *IEEE Haptics Symposium (HAPTICS)*. IEEE, 2012, pp. 91–98.

[19] L. C. Vu and B.-J. You, "Hand pose detection in hmd environments by sensor fusion using multi-layer perceptron," in *2019 IEEE International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019, pp. 218–223.

[20] S. Ameur, A. B. Khalifa, and M. S. Bouhlel, "A comprehensive leap motion database for hand gesture recognition," in *7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*. IEEE, 2016, pp. 514–519.

[21] "Htc vive tracker." [Online]. Available: https://www.vive.com

[22] "3gear systems." [Online]. Available: http://nimblevr.com

[23] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using svd," *Computing*, vol. 1, no. 1, 2017.

[24] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine vision and applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[25] B. Alexander and K. Viktor, "Proportions of hand segments," *International Journal of Morphology*, vol. 28, no. 3, pp. 755–758, 2010.

[26] A. Witkin and Z. Popovic, "Motion warping," in *SIGGRAPH*, vol. 95, no. 29, 1995, pp. 105–108.

[27] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," in *Computer Graphics Forum*, vol. 37, no. 6. Wiley misc Library, 2018.

[28] A. Aristidou and J. Lasenby, "Fabrik: A fast, iterative solver for the inverse kinematics problem," *Graphical Models*, vol. 73, no. 5, pp. 243–260, 2011.

[29] A. M. Lund, "Measuring usability with the use questionnaire12," *Usability interface*, vol. 8, no. 2, pp. 3–6, 2001.