Evaluating Visual-Spatiotemporal Coregistration of a Physics-based Virtual Reality Haptic Interface

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

26-02-2022 / 03-03-2022

CITATION

Chowdhury, Suman; Mubarrat, Syed T.; Fernandes, Antonio; Binkley, Kieran (2022): Evaluating Visual-Spatiotemporal Coregistration of a Physics-based Virtual Reality Haptic Interface. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.19242951.v1>

DOI

[10.36227/techrxiv.19242951.v1](https://doi.org/10.36227/techrxiv.19242951.v1)

Evaluating Visual-Spatiotemporal Coregistration of a Physics-based Virtual Reality Haptic Interface

S. Chowdhury*, *Member, IEEE*, S. Mubarrat*, A. Fernandes, K. Binkley

Abstract— This study aimed to evaluate the visual-spatiotemporal coregistration of the real and virtual objects' movement dynamics in a low-cost physics-based virtual reality (VR) system that provides real cutaneous and kinesthetic haptic feedback of the objects instead of computer-generated haptic feedback. Twelve healthy participants performed three human-robot collaborative (HRC) sequential pick-and-place lifting tasks while both motion capture and VR systems respectively traced the movement kinematics of the real and virtual objects. We used an iterative closest point algorithm to transform the 3D spatial point clouds of the VR system into the motion capture system. We employed a novel algorithm and principal component analysis to respectively calculate visual and spatiotemporal coregistration precisions between virtual and real objects. Results showed a high correlation ($r > 0.96$) between real and virtual objects' movement dynamics and linear and angular coregistration errors of less than 5 cm and 8° , respectively. The trend also revealed a low temporal registration error of <12 ms and was only found along the vertical axis. The visual registration data indicated that the real cutaneous and kinesthetic haptics provided by the physical objects in the virtual environment enhanced proprioception and visuomotor functions of the users.

Index Terms— coregistration, high-fidelity, haptics technology, virtual reality, human-robot collaboration.

I. INTRODUCTION

THERE has been a considerable interest across various domains to the use of virtual reality (VR) technology to improve functional motor performance through training and performance feedback. The majority of studies that have evaluated the usability of VR systems in improving functional motor performance [1], [2], [3] and for rehabilitation [4], [5], [6], [7] were only based on auditory and visual (AV) feedback. The introduction of haptic interaction—the bidirectional transmission of cutaneous and kinesthetic sensations felt by a human when they interact with their environment—has been shown to enhance the experience, motor performance, and motor learning of the users in both industrial [8], [9], [10], [11] and rehabilitation applications [12], [13], improving motor skills performance and motor rehabilitation.

Over the past decade, the uses of various haptic devices were observed to simulate cutaneous and kinesthetic haptic interactions. For example, touchable pin-based shape-changing interfaces [14] and wearable tactile devices [15], [16] have been developed to provide cutaneous haptic interactions (or feedback) in VR systems. Wand-based controllers [17], [18],

haptic interface robots [19], and wearable devices with vibrotactile feedback [20] have been developed to provide kinesthetic haptic interaction in VR systems. However, all these aforementioned haptic devices are expensive and provide computer-generated (i.e., non-natural) limited force feedback. Consequently, they are not suitable for many applications (e.g., gross-motor tasks that require high force demands) due mainly to the complexity of simulating force-displacement relationships on the scale and precision of human force-sensing [21]. Furthermore, their non-natural haptic feedback fails to provide a high VR simulation fidelity [22]. In recent years, the development of VR trackers and the integration of VR and motion capture systems have aided researchers to develop high-fidelity virtual replications of real-world objects (i.e., physics-based modeling), in which users are provided with real cutaneous (object geometries) and kinesthetic (force/vibration) feedback of the virtual objects (i.e., physics-based haptics) [22], [23], [24]. This method uses different hardware (such as VR trackers or motion capture markers) to track objects in the real world and simulate them in the virtual environment (VE). For example, Cooper et al. [22] simulated a wheel change task on a virtual car by using the integration of VR and a motion capture system. They used the physical prop of the wheel to provide the geometric and force feedback and two vibration motors attached to hand gloves to provide vibration feedback to the participants. Several other studies have also embarked in this emerging research area—creating physics-based VR system using physical props—to increase fidelity of the VE [23], [24].

The fidelity and training effectiveness of the aforementioned VR systems depends not only on the spatial and temporal coregistration of physical objects and their virtual counterparts but also on the visual registration of physical objects and their virtual counterparts [25]. Therefore, it is important that the position and orientation tracking (i.e., surface registration) of the physical props are of sufficient accuracy (low offsets between real and virtual objects' position and orientation in the tracking space) and the latency between the physical movement of the props and the corresponding update in the VE display (i.e., temporal registration) is low in order to prevent the VE from becoming unstable and causing motion sickness [25]. In addition, as the acquisition of visuomotor skills is an integral part of motor skills learning [26], [27], [28] the visual registration of physical and virtual objects in such systems is crucial for effective training.

A few previous studies have evaluated spatial and temporal

coregistration errors of VR systems [25], [29], [30]. For example, one study [25] assessed position and orientation tracking errors and latency of the HTC VIVE VR system by placing the head-mounted display (HMD) of the system at several equally-spaced points in a 2D grid at a constant height (i.e., a constant value of the coordinate in the inferior-superior direction) and comparing the positional data of the VR system with the true grid values. The authors found that the tracking precision was high and the system latency was low (22 ms). Another study [29] evaluated the positional and orientational tracking errors of the HTC VIVE VR system with multiple VIVE trackers in static and dynamic conditions. The authors found a significant increase in errors in dynamic conditions compared to static conditions. Moreover, a real-time full-body human tracking system was developed using the HMD and multiple VIVE trackers of the HTC VIVE VR system in [30]. This system was subjectively and objectively evaluated in terms of user experience and end-to-latency, respectively. However, these aforementioned studies employed a single tracked object and expensive third party software (such as Vizard 5.5 in [25]) and hardware (e.g., for implementing the absolute pose estimator algorithm for calibration and tracking in [29]) for calibrating and assessing the spatiotemporal registration of their respective VR systems. To our knowledge, no previous studies have reported any empirical evidence of spatial and temporal coregistration and visual registration of a physics-based VR system simulating tasks characterized by perceptual and/or motor components (e.g., human-robot collaborative or HRC tasks) using multiple tracked objects.

Therefore, this study aimed to (1) develop a low-cost physics-based VR system and (2) implement a novel methodological approach to evaluate the effectiveness of the developed system in terms of surface and temporal coregistrations and visual registration of multiple real and virtual objects. We simulated an industrial HRC pick-and-place task using physical props. The main motivation was that such a VR system could be used in military training, industrial training, surgical training, and as assistive technology for both physically and cognitively disabled population and can provide users with not only the first-hand experience of the real-world task but also to retain relevant skills, safety-awareness, and functional work capacity. We hypothesize that there would be no significant differences between VE and real environment (RE) in our developed VR system in terms of surface and temporal coregistration and visual registration errors.

II. METHODS

A. PhyVirtual (Physics-based Multimodal Virtual Reality) System Design

Apparatus. We used the HTC Vive Pro Eye Virtual Reality System (HTC Corporation, New Taipei, Taiwan and Valve Corporation, Bellevue, Washington, USA)—consisting of a head-mounted display (HMD), two base stations, two controllers, a link box, wireless add-on, and multiple VIVE trackers—to develop the PhyVirtual (Physics-based Multimodal Virtual Reality) System. The system had dual

Fig. 1. Flowchart of the PhyVirtual system design process along with the initial and destination locations of the boxes. Both 'Real objects' and 'Tracked virtual objects' existed in both real and virtual environments.

AMOLED 3.5 in diagonal HMD of 2880×1600 pixels resolution with a 90 Hz refresh rate, embedded eye-tracking sensors with 110° field of view with accuracy of $0.5^\circ - 1.1^\circ$. Furthermore, we used the Unity3D game engine (Unity Technologies, San Francisco, USA) with the SteamVR plugin to design and display the virtual environment on a participant's HMD. The HTC VIVE Pro Eye VR system utilized SteamVR tracking to track the HMD and the VIVE trackers in real-time.

Virtual environment design. We designed a virtual warehouse environment simulating its interior containing workers, forklift, pallet racks, pallets, boxes, and four industrial collaborative robots (or cobots) (Fig. 1). We simulated cutaneous and kinesthetic haptic feedback inside the virtual environment through the use of physical props. Three real-world pallets (one in the front, and one each on the left and the right side of the participant; each with $91.4 \text{ cm} \times 91.4 \text{ cm} \times 15.2 \text{ cm}$ dimensions), four boxes (weighing 9 kgs with $42.2 \text{ cm} \times 28.4 \text{ cm} \times 23.6 \text{ cm}$ dimensions), and the participants' hands (real objects in Fig. 1) were registered and connected to the virtual environment by placing one HTC VIVE tracker (HTC Corporation, New Taipei, Taiwan and Valve Corporation, Bellevue, Washington, USA) on each of them in the real world. The position, orientation, and movement of these trackers were tracked by the two base stations of the VR system within the experimental setup area. Using these trackers, we simulated the position and movement of the real-world pallets, boxes and the participants' hands in the virtual environment (tracked virtual objects in Fig. 1). To enhance the display fidelity of the VE, we added forklifts, workers, pallet racks, and other interior components (untracked virtual objects in Fig. 1) to the virtual warehouse. In addition, we designed four virtual industrial cobots and programmed them to interact with the users in the VE (Fig. 1) to simulate a novel virtual HRC scenario. We also provided auditory feedback by simulating a varied level of environmental noise (85-90 dB), including sound effects of the robots' movements. The virtual environment was rendered on the HMD with a frame rate of 30 fps (frames per second).

Physics-based system mapping. As the Unity3D software used a metric measurement system, we measured the length, width, and height of the real-world boxes and pallets in meters. For their virtual counterpart, we created seven 1-cubic meter ($1 \text{ m} \times 1 \text{ m} \times 1 \text{ m}$) game objects in Unity3D and scaled them according to the measurement of the real boxes and pallets. For simulating the hands, we imported a generic glove model from the Unity3D asset store and scaled them according to generic hand size. Before the start of the design process, we calibrated the VR system by placing the two base stations on opposite sides of the experimental area facing each other. Then, we paired the VIVE controllers and VIVE trackers to the host PC through the SteamVR plugin. After pairing, the HMD, each controller, and each tracker obtained a unique ID inside the Unity3D design environment. Also, the location at which the HMD was placed in the experimental setup was registered by the VR system as the origin of the world space. Finally, we defined the calibrated space by using the HTC VIVE controllers

Fig. 2. The progression of an experimental task in the real world and the subject's point of view in the virtual environment along with the HMD, box vertices, and gaze direction coordinates and the vectors used for visual registration analysis.

to trace a rectangular area inside the experimental setup. Consequently, the position and orientation of the HMD and the VIVE trackers inside the calibrated space were calculated by the VR system with respect to the origin in the world space. Also, a new game object appeared in the Unity3D design environment with a bounding box according to the calibrated space in the real world.

For placing the VIVE trackers on the boxes in the real world, we measured the center points of one side of each box in the real world and placed one VIVE tracker at that location of each box (Fig. 1). Inside the Unity3D design environment, we placed an empty game object at the exact same location on each virtual box and made each of these empty game objects the parent of their respective virtual boxes. Similarly, we placed one VIVE tracker on one edge of each pallet, measured the position of the VIVE tracker on the real pallets, and placed an empty game object at the exact same location on each virtual pallet. Then, we made each virtual pallet a child of their respective empty game object. Finally, we made all virtual objects the children of the calibrated space in the Unity3D design environment and allocated the ID of the real-world VIVE trackers to their respective empty game objects in the virtual environment.

Thus, the Unity3D game engine obtained the position and orientation data of the boxes, pallets, participants' hands in the real world from the SteamVR tracking system and displayed them in the participants' HMD according to their respective tracker ID.

B. Experiment

Participants. Twelve healthy participants (all males; mass: $72.96 \pm 6.90 \text{ kg}$, height: $1.74 \pm 0.05 \text{ m}$, age: 25.92 ± 3.55 years; free from any musculoskeletal discomfort or abnormality at the time of participation) were recruited to participate in this study. All participants provided written

informed consent, approved by the Institutional Review Board of Texas Tech University.

Experimental design. In this study, participants performed three sequential HRC pick-and-place tasks in the VE and three simple pick-and-place tasks in the RE. The participants and four cobots interacted with each other in the VE to perform sequential pick-and-place tasks on the same workpieces (i.e., the boxes). Each cobot deposited a box at the participants' vicinity, which they had to move from one side pallet to the front pallet place by twisting their upper torso in a squat lifting manner (Box 1, 2, 3, and 4 from 30° right to the mid-sagittal plane to 10° left to the mid-sagittal plane, from right mid-frontal plane to 10° right to mid-sagittal plane, from 30° left to the mid-sagittal plane to 10° right to mid-sagittal plane, and from left mid-frontal plane to 10° left to mid-sagittal plane, respectively) (Fig. 1). A new cobot was activated only when the participant had completed moving the box deposited by the preceding cobot. The pace of the lifting was controlled—8 seconds—using a virtual metronome (through visual and auditory feedback) in the VE. Also, a 30-second rest time was provided using a virtual timer (through visual and auditory feedback) between each box lifting exertion in the VE. After the first two boxes were moved from the side pallet to the front pallet, the experimenters removed the boxes from the experimental setup while the participant was resting. When all four boxes (from both sides) were moved to the empty pallet, a VE task (VET) was completed, signifying one VET trial. Similar to VETs, participants performed the same pick-and-place sequential tasks in the real environment using the same background noise level (85 – 90 dB), metronome pace (8 seconds), and timer to guide both sequential tasks and rest period. The real environment task (RET) did not include cobots and warehouse interiors since this study was conducted in a controlled laboratory setting. We randomized the order of the six trials, with at least 3 minutes of rest in between tasks. Moreover, the box deposition order was randomly selected, and a unique order was assigned to the individual participants throughout the session for both VETs and RETs.

Data acquisition system. A 10-camera *Motion Analysis motion capture system* (Motion Analysis Corporation, Rohnert Park, California, USA) was used to record the positions and movement of the boxes. Eight surface markers were placed on each box on the eight vertices and five markers were placed on the HMD during the VR tasks. The marker data was collected continuously at a sampling frequency of 100 Hz for the entire duration of each task. The *Cortex software* (Motion Analysis Corporation, Rohnert Park, California, USA) was used for recording and processing the marker data. A Bertec force plate (Bertec Corporation, Columbus, Ohio, USA) was integrated with the motion capture to collect ground force data during each task. Eye-tracking data (gaze origin and gaze direction coordinates in the VE coordinate system) were recorded using the built-in eye tracker on the HMD of the VR system during the VR tasks from the Unity3D interface. The Unity3D interface also recorded timestamp, frame rate, frame number, and the coordinates of the HMD, the VIVE trackers, and box vertices in the VE coordinate system.

C. Data Analysis

Data preprocessing. The motion capture data recorded using Cortex for the complete task duration were separated into four sections (one for each box lifting exertion). The marker data were then processed, exported out of Cortex into separate text files, and resampled into 2000 samples using a custom-built Matlab script in MATLAB (R2019b, The Math Works Inc.) program. We then calculated coordinates of the centroid (X_c, Y_c, Z_c) of each box by averaging coordinates of motion capture markers placed near to eight vertices of each box.

The data from the Unity3D interface were separated into four sections and exported out into separate text files. The two sets of data were synced using the timestamps on Cortex and Unity3D. The eye gaze origin and direction, HMD, and vertices coordinates were then resampled to the same frames as their counterparts from the motion capture data.

Coordinate transformation. The VE coordinate system was a head-fixed coordinate system (anterior-posterior direction as the Z-axis, the medial-lateral direction as the X-axis, and the inferior-superior direction as the Y-axis) [30] whereas the real objects were placed in a right-handed coordinate system (anterior-posterior direction as the X-axis, the medial-lateral direction as the Y-axis, and the inferior-superior direction as the Z-axis). Furthermore, as we did not use any third-party software (such as Vizard 5.5 used in [25]) to extract the raw position and orientation data provided by the HTC VIVE VR system, the origin of the VE coordinate system was also different from that of the motion capture coordinate system. As a result, the eye gaze origin and direction, HMD, and vertices coordinates in the VE coordinate system had to be transformed into the real-world coordinate system.

Assuming $\mathbf{X}$ and $\mathbf{X}'$ are the real-world and VE coordinate systems, respectively, we can denote the coordinates of the centroid of the box as $\mathbf{A} = (A_x, A_y, A_z)$ and $\mathbf{A}' = (A'_x, A'_y, A'_z)$ for $\mathbf{X}$ and $\mathbf{X}'$ coordinate systems, respectively. Thus, the transformation of $\mathbf{A}'$ from $\mathbf{X}'$ to $\mathbf{X}$ can be written in matrix form as in (1) and in equation form in (2).

$$\mathbf{A} = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} \cos \theta_{xx} & \cos \theta_{xy} & \cos \theta_{xz} \\ \cos \theta_{yx} & \cos \theta_{yy} & \cos \theta_{yz} \\ \cos \theta_{zx} & \cos \theta_{zy} & \cos \theta_{zz} \end{bmatrix} \begin{bmatrix} A'_x \\ A'_y \\ A'_z \end{bmatrix} + \begin{bmatrix} O_x \\ O_y \\ O_z \end{bmatrix} \quad (1)$$

$$\mathbf{A} = \mathbf{R}\mathbf{A}' + \mathbf{T} \quad (2)$$

where, $\mathbf{R}$ is the rotation matrix; $\mathbf{T}$ is the translation matrix; θ_{ij} are the angles between the axes of the two coordinate systems ($i = x, y, z; j = x, y, z$); and O_i are the coordinates of the origin of the VE coordinate system with respect to the origin of the real-world coordinate system. Using same translation and rotation matrices, the eye gaze origin and direction and HMD coordinates can also be transformed into the real-world coordinate system.

To obtain the rotation and translation matrices, we used the *iterative closest point* (ICP) algorithm [31]. This algorithm iteratively revised the translation and rotation required to minimize an error metric between two point clouds. In this method, one point cloud was kept fixed (the reference point cloud), whereas the other one (the source point cloud) was transformed to best match the reference point cloud. After the iterative process, the transformed point cloud, as well as the

translation and the rotation matrices were obtained.

For our application, the set of coordinates of the centroid of a box in the real-world coordinate system for each frame of the task duration was considered as the reference point cloud, while its counterpart in the VE coordinate system was the source point cloud. We chose the root mean squared error (RMSE) as the error metric. The ICP algorithm was implemented using a custom-built Matlab script in MATLAB (R2019b, The Math Works Inc.) program. Using translation and rotation matrices obtained from the ICP algorithm, the eye gaze origin and direction, HMD, and vertices coordinates were transformed into the real-world coordinate system using (2).

Spatial coregistration. In static condition (when the box was stationary), assuming (X_R, Y_R, Z_R) and (X_{VR}, Y_{VR}, Z_{VR}) as coordinates of the centroid of a box obtained from the motion capture system and their transformed counterparts from the VE, respectively, we calculated the RMSE between them using (3) - (5), which served as the linear offset of each box.

$$\text{RMSE}_X = \sqrt{\frac{\sum_{i=1}^N (X_{VRi} - X_{Ri})^2}{N}} \quad (3)$$

$$\text{RMSE}_Y = \sqrt{\frac{\sum_{i=1}^N (Y_{VRi} - Y_{Ri})^2}{N}} \quad (4)$$

$$\text{RMSE}_Z = \sqrt{\frac{\sum_{i=1}^N (Z_{VRi} - Z_{Ri})^2}{N}} \quad (5)$$

where $N = 20$ (i.e., the first 20 frames in one task duration); i denotes the i -th frame during that task duration ($i = 1, 2, 3, \dots, N$); RMSE_X , RMSE_Y , and RMSE_Z are the RMSE values for X, Y, and Z axes, respectively.

For calculating the angular orientation of the boxes, we used the principal component analysis (PCA) method. Assuming $[X_i \ Y_i \ Z_i]$ were the coordinates of eight vertices (i.e., motion capture markers) of a box for one frame of data calculated from the motion capture data ($i = 1, 2, 3, \dots, 8$), we represented the data using a matrix $\mathbf{A}$, where

$$\mathbf{A} = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_8 & Y_8 & Z_8 \end{bmatrix} \quad (6)$$

The mean of matrix $\mathbf{A}$ would be

$$\bar{\mathbf{A}} = [\bar{X} \ \bar{Y} \ \bar{Z}] \quad (7)$$

Then, we calculated the covariance matrix $\text{cov}\mathbf{A}$ by computing the covariance of three components: X, Y, and Z in matrix $\mathbf{A}$.

$$\text{cov}\mathbf{A} = \begin{bmatrix} \text{cov}_{xx} & \text{cov}_{xy} & \text{cov}_{xz} \\ \text{cov}_{yx} & \text{cov}_{yy} & \text{cov}_{yz} \\ \text{cov}_{zx} & \text{cov}_{zy} & \text{cov}_{zz} \end{bmatrix} \quad (8)$$

Next, we calculated the eigenvalues $\lambda = \lambda_1, \lambda_2, \lambda_3$ of $\text{cov}\mathbf{A}$ by solving (9), which can be simplified as (10).

$$\det(\text{cov}\mathbf{A} - \lambda \mathbf{I}) = 0 \quad (9)$$

$$\det \begin{pmatrix} \text{cov}_{xx} - \lambda & \text{cov}_{xy} & \text{cov}_{xz} \\ \text{cov}_{yx} & \text{cov}_{yy} - \lambda & \text{cov}_{yz} \\ \text{cov}_{zx} & \text{cov}_{zy} & \text{cov}_{zz} - \lambda \end{pmatrix} = 0 \quad (10)$$

where, $\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ was the identity matrix. Then, we calculated eigenvectors: $\mathbf{EV}_1$, $\mathbf{EV}_2$, and $\mathbf{EV}_3$ corresponding to

eigenvalues: λ_1 , λ_2 , and λ_3 by solving equations (11)-(13).

$$\begin{bmatrix} \text{cov}_{xx} - \lambda_1 & \text{cov}_{xy} & \text{cov}_{xz} \\ \text{cov}_{yx} & \text{cov}_{yy} - \lambda_1 & \text{cov}_{yz} \\ \text{cov}_{zx} & \text{cov}_{zy} & \text{cov}_{zz} - \lambda_1 \end{bmatrix} \begin{bmatrix} \mathbf{EV}_{1x} \\ \mathbf{EV}_{1y} \\ \mathbf{EV}_{1z} \end{bmatrix} = \mathbf{0} \quad (11)$$

$$\begin{bmatrix} \text{cov}_{xx} - \lambda_2 & \text{cov}_{xy} & \text{cov}_{xz} \\ \text{cov}_{yx} & \text{cov}_{yy} - \lambda_2 & \text{cov}_{yz} \\ \text{cov}_{zx} & \text{cov}_{zy} & \text{cov}_{zz} - \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{EV}_{2x} \\ \mathbf{EV}_{2y} \\ \mathbf{EV}_{2z} \end{bmatrix} = \mathbf{0} \quad (12)$$

$$\begin{bmatrix} \text{cov}_{xx} - \lambda_3 & \text{cov}_{xy} & \text{cov}_{xz} \\ \text{cov}_{yx} & \text{cov}_{yy} - \lambda_3 & \text{cov}_{yz} \\ \text{cov}_{zx} & \text{cov}_{zy} & \text{cov}_{zz} - \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{EV}_{3x} \\ \mathbf{EV}_{3y} \\ \mathbf{EV}_{3z} \end{bmatrix} = \mathbf{0} \quad (13)$$

$$\text{where, } \mathbf{EV}_1 = \begin{bmatrix} \mathbf{EV}_{1x} \\ \mathbf{EV}_{1y} \\ \mathbf{EV}_{1z} \end{bmatrix}, \mathbf{EV}_2 = \begin{bmatrix} \mathbf{EV}_{2x} \\ \mathbf{EV}_{2y} \\ \mathbf{EV}_{2z} \end{bmatrix}, \text{ and } \mathbf{EV}_3 = \begin{bmatrix} \mathbf{EV}_{3x} \\ \mathbf{EV}_{3y} \\ \mathbf{EV}_{3z} \end{bmatrix}.$$

$\mathbf{EV}_1$, $\mathbf{EV}_2$, and $\mathbf{EV}_3$ were respectively the principal component axes $\mathbf{X}_R$, $\mathbf{Y}_R$, and $\mathbf{Z}_R$ of the real box. We repeated the same steps to calculate the principal component axes ($\mathbf{X}_V$, $\mathbf{Y}_V$, and $\mathbf{Z}_V$) of the virtual box. Finally, we calculated the angles between $\mathbf{X}_R$ and $\mathbf{X}_V$, $\mathbf{Y}_R$ and $\mathbf{Y}_V$, and $\mathbf{Z}_R$ and $\mathbf{Z}_V$ using (14) - (16).

$$\theta_X = \cos^{-1} \left| \frac{\mathbf{X}_R \cdot \mathbf{X}_V}{|\mathbf{X}_R| |\mathbf{X}_V|} \right| \quad (14)$$

$$\theta_Y = \cos^{-1} \left| \frac{\mathbf{Y}_R \cdot \mathbf{Y}_V}{|\mathbf{Y}_R| |\mathbf{Y}_V|} \right| \quad (15)$$

$$\theta_Z = \cos^{-1} \left| \frac{\mathbf{Z}_R \cdot \mathbf{Z}_V}{|\mathbf{Z}_R| |\mathbf{Z}_V|} \right| \quad (16)$$

We then repeated these steps for the first 20 frames in each task duration, averaged the values, and obtained the angular orientation errors (i.e., the angular offset) between virtual and real boxes in static conditions. Similarly, we used (3) - (16) to calculate linear and angular offsets during the dynamic condition (i.e., for the entire task duration of 2000 frames). Considering the errors in static conditions as baseline errors, we subtracted them from the errors calculated in dynamic conditions to obtain the dynamic spatial registration errors.

Temporal coregistration. We considered the X coordinates of the centroid of a box obtained from the motion capture system (X_R) and their transformed counterparts from the VE (X_{VR}) as two separate discrete signals and measured the time delay between them by computing the cross-correlation function [32], which was defined using (17) and normalized using (18) [33].

$$\hat{R}_{XR_{XVR}}(m) = \begin{cases} \sum_{n=0}^{N-m-1} X_R(n+m) X_{VR}^*(n), & m \geq 0 \\ \hat{R}_X^*(-m), & m < 0 \end{cases} \quad (17)$$

$$\hat{R}_{XR_{XVR_norm}}(m) = \frac{\hat{R}_{XR_{XVR}}(m)}{\sqrt{\hat{R}_{XR_{XR}}(0) \hat{R}_{XVR_{XVR}}(0)}} \quad (18)$$

where, N was the total number of frames in one task duration, $m = -(N-1), -(N-2), \dots, 0, 1, 2, \dots, (N-2), (N-1)$ were the lag indices between two samples, and the asterisk denoted complex conjugation. Thus, for the values of m in the range $2N-1$, we get the cross-correlation vector in (19).

$$c(m) = \hat{R}_{XR_{XVR_norm}}(m - N) \quad (19)$$

Next, the sample delay between the two signals, sd_X , was

computed by taking the argmax of the cross-correlation vector, as in (20), which denoted the lag for which the normalized cross-correlation had the largest absolute value.

$$sd_x = \text{argmax}[c(m)] \quad (20)$$

Finally, the time delay between the two signals, td_x in seconds was computed by dividing the sample delay (sd_x), by the sampling rate of the signals (SR). We then repeated the same steps for calculating time delays along Y (medial-lateral direction) and Z (inferior-superior direction) axes.

Visual registration. We considered the visual registration as a user-centric performance measure to evaluate the developed PhyVirtual system. We used the coordinates of the vertices of the top surface of the box from the real-world coordinate system and the transformed coordinates of the HMD and the gaze direction from the VE coordinate system to calculate the metrics required to validate the visual registration of a box. Firstly, we resampled all data into 1000 frames. Assuming (X_G, Y_G, Z_G) , (X_H, Y_H, Z_H) , and $(X_H, Y_H, 0)$ were the coordinates of the gaze direction, the HMD and the ground, respectively, we computed the normalized gaze vectors, $\mathbf{E}$ and the normalized ground vector, $\mathbf{F}$ using (25) - (26) (Fig. 2).

$$\mathbf{E} = \frac{\mathbf{G}-\mathbf{H}}{\text{norm}(\mathbf{G}-\mathbf{H})} \quad (21)$$

$$\mathbf{F} = \frac{\mathbf{C}-\mathbf{H}}{\text{norm}(\mathbf{C}-\mathbf{H})} \quad (22)$$

$$\text{where, } \mathbf{G} = \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}, \mathbf{H} = \begin{bmatrix} X_H \\ Y_H \\ Z_H \end{bmatrix}, \text{ and } \mathbf{C} = \begin{bmatrix} X_H \\ Y_H \\ 0 \end{bmatrix}$$

Similarly, we used (27) to calculate the normalized vectors connecting the HMD to the box vertices of the top surface (Fig. 2).

$$\mathbf{B}_i = \frac{\mathbf{V}_i-\mathbf{H}}{\text{norm}(\mathbf{V}_i-\mathbf{H})} \quad (23)$$

where, $\mathbf{B}_i$ was the normalized vector for the i -th vertex ($i = 1,$

$$2, 3, 4); \mathbf{V}_i = \begin{bmatrix} V_{ix} \\ V_{iy} \\ V_{iz} \end{bmatrix} \text{ were the coordinates of the } i\text{-th vertex } (i =$$

$$1, 2, 3, 4); \text{ and } \mathbf{H} = \begin{bmatrix} X_H \\ Y_H \\ Z_H \end{bmatrix} \text{ (Fig. 2).}$$

We then projected all vectors into the XZ plane (frontal plane) to obtain $\mathbf{E}_{xz}$, $\mathbf{F}_{xz}$, and $\mathbf{B}_{i,xz}$ vectors (Fig. 2). Assuming $\angle \mathbf{E}_{xz}$ and $\angle \mathbf{B}_{i,xz}$ denoted the angles (in degrees) that $\mathbf{E}_{xz}$ and $\mathbf{B}_{i,xz}$ ($i = 1, 2, 3, 4$) vectors made with the XY plane (transverse plane), we calculated them using (28) - (29).

$$\angle \mathbf{E}_{xz} = 90 - \theta_{\mathbf{E}_{xz}\mathbf{F}_{xz}} \quad (28)$$

$$\angle \mathbf{B}_{i,xz} = 90 - \theta_{\mathbf{B}_{i,xz}\mathbf{F}_{xz}} \quad (29)$$

where, $\theta_{\mathbf{E}_{xz}\mathbf{F}_{xz}}$ denoted the angle between $\mathbf{E}_{xz}$ and $\mathbf{F}_{xz}$ (Fig. 2); and $\theta_{\mathbf{B}_{i,xz}\mathbf{F}_{xz}}$ denoted the angles between $\mathbf{B}_{i,xz}$ and $\mathbf{F}_{xz}$ ($i = 1, 2, 3, 4$) (Fig. 2).

Projecting the vectors on the YZ plane (sagittal plane), we repeated the same steps to obtain $\angle \mathbf{E}_{yz}$ and $\angle \mathbf{B}_{i,yz}$. Using the values of $\angle \mathbf{E}_{xz}$, $\angle \mathbf{E}_{yz}$, $\angle \mathbf{B}_{i,xz}$, and $\angle \mathbf{B}_{i,yz}$, we developed a program in MATLAB (R2019b, The Math Works Inc.) to determine if the gaze fell on the top surface of the box (returning a value of '1' if it did and '0' otherwise).

Next, we calculated the percentage of gaze falling on the box across all subjects and all trials for each frame of the task duration, generating a percentage distribution. Furthermore, we separated the task duration into five events—event 1, 2, 3, 4, and 5 denoting 0-10% (moving toward the box), 20-30% (lifting the box), 55-65% (moving the box), 80-90% (depositing the box), and 90-100% (moving away from the box) of the task duration, respectively. We calculated the percentage of gaze fixated on the box for all frames in each event, which served as the gaze fixation percentage for that event. In addition, we calculated the percentage of subjects that looked at the box at least once during each event across all subjects and all trials. Finally, using a moving average window of 10 frames, we smoothed and resampled the percentage distribution of gaze fixation for each box into 100 frames, with each frame denoting one percentile of task duration.

Statistical analysis. We calculated the descriptive statistics (mean and standard error) of the spatial registration errors in static and dynamic conditions separately and the time delay (temporal coregistration errors) by averaging across all subjects and all trials. We also estimated the mean and standard error of the spatial registration errors along X, Y, and Z axes separately by averaging across all boxes, subjects, and trials. Then, we performed Shapiro-Wilk's test to check if the normal distribution assumption was met in the individual datasets. We employed pairwise T-tests (confidence interval at 95%, i.e., $\alpha = 0.05$) for normally distributed datasets to investigate the difference between the spatial registration errors in static and dynamic conditions. The conditions (static or dynamic) were the grouping variables, whereas the linear and angular offsets along X, Y, and Z axes were the dependent variables at their respective inferential statistical tests. In addition, we performed similar pairwise T-tests to examine the differences of the spatial coregistration errors between different axes where the axis directions (X, Y, or Z) were grouping variables. If the normality assumption was violated, we employed non-parametric Wilcoxon signed-rank tests for pairwise sample comparison.

We subtracted the dynamic spatial coregistration error values from the static spatial coregistration error values to determine the actual spatial coregistration errors during the dynamic condition. To compare the movement trajectories of the boxes in VE and RE, we averaged the coordinates of the X, Y, and Z axes separately for each frame of the time-series (0~100% of task progression) data across all subjects and all trials, with each 20 frames denoting one percentile of the task progression. We then computed the Pearson's Correlation Coefficient (r) as a similarity measure between two discrete time-series data of centroids (i.e., movement trajectory) in VE and RE for each box.

III. RESULTS

A. Temporal Co-registration

The time delay data showed no delay (i.e., no temporal coregistration error) along Y-axis and X-axis, but the average time delays (i.e., temporal coregistration error) along the Z-axis were 11.94 ± 5.21 ms, 10.45 ± 4.50 ms, 10.00 ± 5.95 ms, and

8.44 ± 4.81 ms for Box 1, 2, 3, and 4, respectively.

B. Spatial Co-registration

The spatial registration errors for the static condition (presented in Table 1) were treated as the baseline error for the dynamic condition (shown in Table 2) to under-stand the effect of movement dynamics on the co-registration error. The difference of angular spatial registration errors between static and dynamic conditions was found to be statistically insignificant ($p>0.05$) for all conditions except along X-axis for Box 1 and along Z-axis for Box 4. On the contrary, the linear spatial registration errors were found to be significantly different between static and dynamic conditions ($p<0.05$) for most cases, except along Y-axis for Box 1 and Box 3, and along X- and Y- axes for Box 4. Across all axes, the linear and angular coregistration errors in dynamic conditions for Box 1 (linear: 4.61 ± 1.05 cm and angular: $7.46 \pm 1.58^\circ$) and Box 3 (linear: 6.98 ± 2.30 cm and angular: $6.24 \pm 1.40^\circ$) were comparatively higher than Box 2 (linear: 3.16 ± 0.53 cm and

TABLE 1

SPATIAL COREGISTRATION ERRORS AT STATIC CONDITIONS

	Linear offset (cm) ^a		
	Anterior-posterior (Y-axis)	Medial-lateral (X-axis)	Inferior-superior (Z-axis)
Box1	2.56±0.84	5.78±0.58	2.97±0.67
Box2	1.70±0.22	1.54±0.17	0.75±0.16
Box3	1.27±0.37	5.07±1.30	4.20±0.77
Box4	1.77±0.24	1.36±0.28	0.37±0.06
	Angular offset (°) ^a		
	Anterior-posterior (Y-axis)	Medial-lateral (X-axis)	Inferior-superior (Z-axis)
Box1	17.77±2.30	19.63±1.85	7.84±1.66
Box2	7.08±1.37	7.69±1.07	3.20±0.54
Box3	13.10±2.06	12.59±1.98	6.89±1.48
Box4	7.82±1.65	7.74±1.63	2.08±0.41

TABLE 2

SPATIAL COREGISTRATION ERRORS AT DYNAMIC CONDITIONS

	Linear offset (cm) ^a		
	Anterior-posterior (Y-axis)	Medial-lateral (X-axis)	Inferior-superior (Z-axis)
Box1	2.35±0.73	0.97±0.21	3.68±0.90
Box2	2.86±0.80	1.72±0.42	3.29±0.59
Box3	4.86±0.82	3.28±1.13	2.89±0.64
Box4	1.48±0.60	1.92±0.39	3.29±0.88
	Angular offset (°) ^a		
	Anterior-posterior (Y-axis)	Medial-lateral (X-axis)	Inferior-superior (Z-axis)
Box1	7.10±1.20	6.96±1.85	4.02±1.02
Box2	2.93±0.68	3.78±0.84	1.01±0.17
Box3	6.10±1.43	5.17±0.97	3.72±0.89
Box4	2.82±0.61	2.14±0.48	1.68±0.46

^aMean ± standard error, averaged across entire task duration.

angular: $2.57 \pm 0.40^\circ$) and Box 4 (linear: 3.53 ± 1.11 cm and angular: $9.44 \pm 2.35^\circ$). Across all boxes, the linear error along the Z-axis was significantly higher than those along the Y-axis

Fig. 3. Box centroid coordinate comparisons between virtual and real environments throughout the progression of the task cycle for four different box positions. VE, RE, and r denote virtual environment, real environment, and Pearson's correlation coefficient, respectively.

(78.88% higher; $p<<0.001$) and X-axis (75.99% higher; $p=0.002$). No such trend was observed for the angular errors. Moreover, high values of r (>0.96) between the movement trajectories along all three axes of the individual boxes in virtual and real environments exhibited a high precision of spatial coregistration during the dynamic condition (Fig. 3).

C. Visual Registration

The percentage of gaze fixated on the box for each event exhibited a similar trend for all box positions (Fig. 4), with the highest percentage occurring at the lifting instance (Event 2) for all box positions. The highest percentage of gaze was found for Event 2 and Event 4 at 24-th percentile ($88.94 \pm 1.63\%$) and 89-th percentile ($57.93 \pm 5.18\%$) of task duration, respectively. The lowest percentage of gaze was found at 60-th percentile ($27.32 \pm 5.25\%$) of task duration at Event 3 (moving the box), indicating a gaze distraction of the participants due to high-fidelity moving objects inside the virtual warehouse interior environment. Similar trends were observed for the percentage of subjects that looked at the box at least once during each event (Fig. 5), with the percentage decreasing from Event 2 to Event 3 and increasing again from Event 4.

IV. DISCUSSION

This study developed a novel physics-based VR system simulating an industrial HRC pick-and-place task that provided both physical and cognitive interactions through the use of physical props. Evaluation results showed that linear and angular spatial registration errors between real and virtual objects in dynamic conditions were lower than 5 cm and 8° , respectively. However, an interesting trend observed was that although the linear spatial registration errors in static condition were lower along Z axis compared to X and Y axes, they were

Fig. 4. Percentage of gaze fixated on the box during each event of the box lifting exertion for four different box positions. E1, E2, E3, E4, and E5 denote moving towards the box, lifting the box, moving the box, depositing the box, and moving away from the box, respectively.

found to be significantly higher in dynamic condition (Table 1 and Table 2). This can be explained by the finding of a previous study [25] that the Z-axis of the HTC VIVE system is slightly tilted compared to that of the real-world experimental setup. The authors also found that even when the misalignment was corrected, the HTC VIVE's axis definition changed again after the loss of tracking (e.g., if the base stations lost track of any VIVE tracker in dynamic conditions). This may have led to higher spatial registration error along the Z-axis in dynamic conditions (Table 1 and Table 2). Moreover, as it has been found that the level and direction of misalignment of the HTC VIVE's Z-axis was randomized each time the system was initialized [25], [34], which could be the cause of the inter-subject variability regarding the linear and angular spatial registration errors. Future studies should focus on implementing novel algorithms to correct this misalignment, as

Fig. 5. The percentage of subjects that looked at the box at least once during each event for four different box positions. Event 1, Event 2, Event 3, Event 4, and Event 5 denote moving towards the box, lifting the box, moving the box, depositing the box, and moving away from the box, respectively.

proposed in [29] and [34].

Nevertheless, the spatial registration errors observed in our study can be comparable to previous studies that have evaluated the position and orientation tracking errors of the HTC VIVE VR system [25], [29], [34]. For example, although the HMD of the VR system was kept at a constant height (i.e., a constant value of the coordinate along Z-axis) at several equally-spaced points in a 2D grid in two different studies [25], [34], the authors found that the height of the HMD recorded by the VR system varied across different grid points, which is consistent with our finding that the spatial registration errors were higher along Z axis. However, the authors in both studies only focused on analyzing the tracking of the HMD and did not provide any information about the VIVE trackers. Another study [29] found that although the accuracy of the tracking of the HTC VIVE VR system was in the submillimeter range in static conditions, it was very unstable in dynamic conditions (ranging from 0.10 cm to 80.20 cm). For their study, the authors considered only one VIVE tracker and did not investigate the effect of the presence of multiple trackers on the spatial registration errors. On the other hand, our system consisted of multiple trackers (four for the boxes, three for the pallets, and two for the hands) being tracked by the base stations in real-time. Our findings suggest that although the errors in static conditions were higher than those reported in [29], the accuracy was much more stable in dynamic conditions. Higher spatial coregistration errors in static condition was mainly due to the human error associated with the marker placement on the different locations of multiple tracked objects (i.e., on the vertices of the boxes) in the RE.

Discrepancies in the spatial registration between real and virtual objects may also be due to the sudden loss of tracking and measurements as the objects move around the tracking space in dynamic conditions, as suggested by [25] and [29]. The HTC VIVE VR system utilizes inertial measurements from the HMD and the trackers as the primary inputs in estimating the current tracked position and orientation of the HMD and the trackers [25]. For this estimation, dead reckoning (or path integration)—a method of estimating the current position of a

moving object by using a previously determined position and then incorporating estimations of speed, heading direction, and course over elapsed time—is used. However, dead reckoning based on inertial measurements is susceptible to accumulation of positional and orientational errors with time [35], [36], particularly for consumer-grade inertial measurement units such as the ones on the HMD and the trackers [29]. As a result, the base stations are used to estimate the absolute pose of the HMD and the trackers and limit the build-up of error in dead reckoning [25]. Nonetheless, there were possibilities of the laser beams of the base stations being blocked off by the subject during the dynamic condition, leading to the loss of tracking. As loss of tracking has been found to cause higher position and orientation offsets in dynamic conditions than static conditions [29], this could have caused some noticeable differences in the spatial coregistration errors between static and dynamic conditions. Additionally, Box 1 and Box 3 were highly susceptible to loss of tracking due to their positioning (blocked by the human participant) in our experimental setup area (Fig. 1), which could be the leading cause of the higher spatial coregistration errors observed for these two boxes. Nevertheless, a strong correlation between the trends of the coordinates of the centroid of real and virtual boxes (Fig. 3) for all box positions indicated that our system can simulate a high-fidelity VR environment with a minimal amount of coregistration errors. In addition, the subjective questionnaire data, reported in a separate article [37], exhibited a high interaction fidelity (i.e., haptic realism) during both static and dynamic conditions (a score of 5.00 out of 7.00 in the Likert Scale), revealing that the coregistration error values were unnoticeable and thus did not influence the motor performance of the subjects.

The time delay (viz., the temporal coregistration error) between the physical movement of real objects and the corresponding update in the VE display (i.e., the latency) was found to be lower than 12 ms for all box positions. The latency values obtained in our study fall between the findings of previous studies where the system latency of the HTC VIVE system was found to be 22 ms [25] and the total time delay of a full-body human tracking system using the HTC VIVE system was found to be 6.71 ± 0.80 ms [30]. Although we set a constant frame rate of 30 fps in the VE, the experimental frame rate was found to be variable (28.30 ± 1.03 fps), which might have also caused some additional time delay for the boxes in the Z-axis.

Although the spatial registration errors in dynamic conditions for Box 1 and Box 3 were higher than Box 2 and Box 4 (Table 1), the trends in the percentage of gaze fixated on the boxes throughout the task duration were similar for all box positions (Fig. 4). We expected the percentage of gaze fixation to be the highest during picking and depositing the boxes and the lowest while moving the boxes. As expected, the percentage of gaze fixated on the box was found to be the highest for Event 2 (lifting the box) and lowest for Event 3 (moving the box) for all four box positions. This suggests a high coregistration precision between virtual and real objects. In addition, the

cutaneous and kinesthetic haptic feedback provided by the real objects in the VE mitigated the discrepancies between real and virtual objects' position and orientation (i.e., erroneous spatial registration) to a greater extent by enhancing the proprioception and visuomotor functions of the users. Nevertheless, the percentage of subjects that looked at the box at least once during each event was found to be less than 100% while lifting the box (Event 2) and depositing the box (Event 4) for three box positions (Fig. 5). This could be due to the inter-subject variability in lifting dynamics, where some subjects may have caused the occurrence of a specific event in the lifting exertion sooner or later than other subjects. Furthermore, the presence of multiple untracked virtual objects (e.g., warehouse interiors, virtual human, cobots, forklifts, etc.) in the VE might have acted as distraction factors, causing the decrease in the percentage of gaze fixated on the box during each event.

There are some limitations of this study that need to be acknowledged. Firstly, we used the HTC VIVE VR system's default calibration and tracking algorithms. Secondly, the base stations used in our experimental setup were not affixed to a rigid surface, which may have caused some inter-subject variability regarding the spatial and temporal coregistration errors whenever the base stations were moved for a new subject.

V. CONCLUSION

In this study, we designed a novel low-cost physics-based VR system and evaluated it in terms of the visual-spatiotemporal coregistration of real and virtual objects. The minimal spatial coregistration errors (linear <5 cm and angular $<8^\circ$) and latency (<12 ms) showed promise regarding the effectiveness of our physics-based VR system in simulating a high-fidelity VE for sensorimotor training, motor rehabilitation, and motor learning applications without the requirement of expensive software and hardware. The developed system can especially be useful for industrial scenarios that require high real word staging investments, involve safety risks, or specific geographic settings. However, as loss of tracking is a leading cause of coregistration errors, future studies should focus on implementing optimal placement of the base stations such that they can easily track multiple trackers without the possibility of occlusion. Moreover, the results warrant potential future studies to develop novel algorithms or methodologies for calibration and object tracking in a physics-based VR system to further reduce (or eliminate) spatiotemporal coregistration errors without the need of expensive software and hardware.

ACKNOWLEDGEMENT

The authors wish to thank study participants and Manoj Srinivasan, Thomas Chen, and Oluwatosin Opafunso for assistance with data collection and processing.

REFERENCES

- [1] D. C. Kamaraj, B. E. Dicianno, H. P. Mahajan, A. M. Buhari, and R. A. Cooper, "Stability and workload of the virtual reality-based

- simulator-2," *Archives of physical medicine and rehabilitation*, vol. 97, no. 7, pp. 1085-1092. e1, 2016.
- [2] Y. Wada *et al.*, "Effects of repeated snowboard exercise in virtual reality with time lags of visual scene behind body rotation on head stability and subjective slalom run performance in healthy young subjects," *Acta oto-laryngologica*, vol. 136, no. 11, pp. 1121-1124, 2016.
 - [3] V. Mouraviev *et al.*, "Urology residents experience comparable workload profiles when performing live porcine nephrectomies and robotic surgery virtual reality training modules," *Journal of robotic surgery*, vol. 10, no. 1, pp. 49-56, 2016.
 - [4] F. Alemanno *et al.*, "Efficacy of virtual reality to reduce chronic low back pain: Proof-of-concept of a non-pharmacological approach on pain, quality of life, neuropsychological and functional outcome," *PloS one*, vol. 14, no. 5, p. e0216858, 2019.
 - [5] P. S. Archambault, N. G. Norouzi, D. Kairy, J. M. Solomon, and M. F. Levin, "Towards establishing clinical guidelines for an arm rehabilitation virtual reality system," in *Replace, Repair, Restore, Relieve—Bridging Clinical and Engineering Solutions in Neurorehabilitation*: Springer, 2014, pp. 263-270.
 - [6] D. Malinvaud *et al.*, "Auditory and visual 3D virtual reality therapy as a new treatment for chronic subjective tinnitus: results of a randomized controlled trial," *Hearing research*, vol. 333, pp. 127-135, 2016.
 - [7] E. Seabrook *et al.*, "Understanding How Virtual Reality Can Support Mindfulness Practice: Mixed Methods Study," *Journal of Medical Internet Research*, vol. 22, no. 3, p. e16106, 2020.
 - [8] B. L. Grant, P. C. Yelder, T. A. Patrick, B. Kapralos, M. Williams-Bell, and B. A. Murphy, "Audiohaptic Feedback Enhances Motor Performance in a Low-Fidelity Simulated Drilling Task," *Brain sciences*, vol. 10, no. 1, p. 21, 2020.
 - [9] C. Harvey, E. Selmanović, J. O'Connor, and M. Chahin, "A comparison between expert and beginner learning for motor skill development in a virtual reality serious game," *The Visual Computer*, pp. 1-15, 2019.
 - [10] F. Škola, S. Tinková, and F. Liarokapis, "Progressive training for motor imagery brain-computer interfaces using gamification and virtual reality embodiment," *Frontiers in Human Neuroscience*, vol. 13, p. 329, 2019.
 - [11] E. Triantafyllidis, C. Mcgreavy, J. Gu, and Z. Li, "Study of Multimodal Interfaces and the Improvements on Teleoperation," *IEEE Access*, vol. 8, pp. 78213-78227, 2020.
 - [12] J. Broeren, M. Rydmark, and K. S. Sunnerhagen, "Virtual reality and haptics as a training device for movement rehabilitation after stroke: a single-case study," *Archives of physical medicine and rehabilitation*, vol. 85, no. 8, pp. 1247-1250, 2004.
 - [13] G. C. Burdea *et al.*, "Feasibility of integrative games and novel therapeutic game controller for telerehabilitation of individuals chronic post-stroke living in the community," *Topics in stroke rehabilitation*, vol. 27, no. 5, pp. 321-336, 2020.
 - [14] K. Nakagaki *et al.*, "Materiable: Rendering dynamic material properties in response to direct physical touch with shape changing interfaces," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 2764-2772.
 - [15] M. Bianchi, "A fabric-based approach for wearable haptics," *Electronics*, vol. 5, no. 3, p. 44, 2016.
 - [16] F. Sergi, D. Accoto, D. Campolo, and E. Guglielmelli, "Forearm orientation guidance with a vibrotactile feedback bracelet: On the directionality of tactile motor communication," in *2008 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2008: IEEE, pp. 433-438.
 - [17] E. Strasnick, C. Holz, E. Ofek, M. Sinclair, and H. Benko, "Haptic links: Bimanual haptics for virtual reality using variable stiffness actuation," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1-12.
 - [18] E. Whitmire, H. Benko, C. Holz, E. Ofek, and M. Sinclair, "Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1-12.
 - [19] T. Endo *et al.*, "Five-fingered haptic interface robot: HIRO III," *IEEE Transactions on Haptics*, vol. 4, no. 1, pp. 14-27, 2010.
 - [20] 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*, 2017, pp. 119-130.
 - [21] H. Culbertson, S. B. Schorr, and A. M. Okamura, "Haptics: The present and future of artificial touch sensation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 385-409, 2018.
 - [22] N. Cooper, F. Milella, C. Pinto, I. Cant, M. White, and G. Meyer, "The effects of substitute multisensory feedback on task performance and the sense of presence in a virtual reality environment," *PLOS ONE*, vol. 13, no. 2, p. e0191846, 2018.
 - [23] S. Hanus, M. Hoover, A. Lim, and J. Miller, "A Collaborative Virtual Reality Escape Room with Passive Haptics," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2019: IEEE, pp. 1413-1414.
 - [24] S. Hasanzadeh and M. Jesus, "Understanding Roofer's Risk Compensatory Behavior through Passive Haptics Mixed-Reality System," in *Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation*, Y. K. Cho, F. Leite, A. Behzadan, and C. Wang Eds.: American Society of Civil Engineers Reston, VA, 2019, pp. 137-145.
 - [25] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research," *i-Perception*, vol. 8, no. 3, p. 2041669517708205, 2017.
 - [26] K. Burris *et al.*, "Sensorimotor abilities predict on-field performance in professional baseball," *Scientific reports*, vol. 8, no. 1, pp. 1-9, 2018.
 - [27] D. R. Feldhacker *et al.*, "Efficacy of high-performance vision training on improving the reaction time of collegiate softball athletes: A randomized trial," *Journal of Sports Medicine and Allied Health Sciences: Official Journal of the Ohio Athletic Trainers Association*, vol. 4, no. 3, p. 6, 2019.
 - [28] D. Formenti *et al.*, "Perceptual vision training in non-sport-specific context: Effect on performance skills and cognition in young females," *Scientific reports*, vol. 9, no. 1, pp. 1-13, 2019.
 - [29] M. Borges, A. Symington, B. Coltin, T. Smith, and R. Ventura, "HTC Vive: analysis and accuracy improvement," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018: IEEE, pp. 2610-2615.
 - [30] P. Caserman, A. Garcia-Agundez, R. Konrad, S. Göbel, and R. Steinmetz, "Real-time body tracking in virtual reality using a Vive tracker," *Virtual Reality*, vol. 23, no. 2, pp. 155-168, 2019.
 - [31] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992, doi: 10.1109/34.121791.
 - [32] M. Azaria and D. Hertz, "Time delay estimation by generalized cross correlation methods," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 280-285, 1984.
 - [33] M. Rhudy, B. Bucci, J. Viperman, J. Allanach, and B. Abraham, "Microphone array analysis methods using cross-correlations," in *ASME 2009 International Mechanical Engineering Congress and Exposition*, 2009: American Society of Mechanical Engineers Digital Collection, pp. 281-288.
 - [34] A. Peer, P. Ullich, and K. Ponto, "Vive tracking alignment and correction made easy," in *2018 IEEE conference on virtual reality and 3D user interfaces (VR)*, 2018: IEEE, pp. 653-654.
 - [35] B.-S. Cho, W.-s. Moon, W.-J. Seo, and K.-R. Baek, "A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding," *Journal of mechanical science and technology*, vol. 25, no. 11, pp. 2907-2917, 2011.
 - [36] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket—an experimental study," in *2010 IEEE international conference on pervasive computing and communications (PerCom)*, 2010: IEEE, pp. 162-170.
 - [37] S. T. Mubarrat, A. Fernandes, and S. K. Chowdhury, "A Physics-based Virtual Reality System Design and Evaluation by Simulating Human-Robot Collaboration," *IEEE Transactions on Human-Machine Systems*, to be published.