IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 March 2024, accepted 9 May 2024, date of publication 13 May 2024, date of current version 20 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3400295

RESEARCH ARTICLE

Toward Intuitive 3D Interactions in Virtual Reality: A Deep Learning-Based Dual-Hand Gesture Recognition Approach

TRUDI DI QI[1], (Member, IEEE), FRANCELI L. CIBRIAN[1], MEGHNA RASWAN[1], TYLER KAY[1], HECTOR M. CAMARILLO-ABAD[2], AND YUXIN WEN[1], (Member, IEEE)

[1]Dale E. and Sarah Ann Fowler School of Engineering, Chapman University, Orange, CA 92866, USA
[2]Schmid College of Science and Technology, Chapman University, Orange, CA 92866, USA

Corresponding authors: Trudi Di Qi (dqi@chapman.edu) and Yuxin Wen (yuwen@chapman.edu)

The work of Trudi Di Qi was supported by the Research Seed Fund at Chapman University. The work of Franceli L. Cibrian and Yuxin Wen was supported by the Research Startup Fund at Chapman University. The work of Meghna Raswan was supported by the Robert A. Day Undergraduate Research Grant at Chapman University. The work of Hector M. Camarillo-Abad was supported by the Grand Challenges Initiative Postdoctoral Fellowship at Chapman University.

This work involved human subjects in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board (IRB) at Chapman University under IRB No. IRB-22-250.

ABSTRACT Dual-hand gesture recognition is crucial for intuitive 3D interactions in virtual reality (VR), allowing the user to interact with virtual objects naturally through gestures using both handheld controllers. While deep learning and sensor-based technology have proven effective in recognizing single-hand gestures for 3D interactions, research on dual-hand gesture recognition for VR interactions is still underexplored. In this work, we introduce CWT-CNN-TCN, a novel deep learning model that combines a 2D Convolution Neural Network (CNN) with Continuous Wavelet Transformation (CWT) and a Temporal Convolution Network (TCN). This model can simultaneously extract features from the time-frequency domain and capture long-term dependencies using 3D position and orientation data from handheld controllers for gesture classification. To evaluate the performance of the proposed model, we designed 13 dual-hand gestures representing fundamental 3D interaction tasks: translation, rotation, scaling, and selection, and then collected data from 26 participants using a VR system. The model's performance was rigorously tested under various hand-tracking scenarios, including dual-hand versus single-hand inputs and complete versus partial motion features. Benchmarking against four state-of-the-art neural networks revealed that CWT-CNN-TCN reliably detects dual-hand gestures with limited tracking data and outperforms the benchmarks. This result paves the way for a dual-hand gesture-based interface that enriches intuitive 3D interactions in VR.

INDEX TERMS Deep learning, dual-hand gestures, hand gesture recognition, natural interface, three-dimensional interactions, virtual reality.

I. INTRODUCTION

In virtual reality (VR), 3D interactions, such as rotating or scaling virtual objects, are facilitated through various user interfaces. Traditional buttons or keys on a handheld controller offer straightforward control but can be less efficient for 3D interactions in VR. Graphical user interfaces (GUI) that are displayed within VR environments provide a visual guide for interactions, but they can clutter the VR space and detract from the immersive experience [1]. Hand

The associate editor coordinating the review of this manuscript and approving it for publication was M. Sabarimalai Manikandan.

motion gestures offer a more natural and intuitive alternative, allowing users to interact with virtual content similarly to real-world actions [2]. For instance, users can intuitively scale objects in VR by simply moving their hands apart or together [3] rather than navigating through cumbersome graphical menus. Previous approaches for recognizing hand gestures in 3D interactions often rely on specialized hardware like data gloves [4], computer vision systems like Leap Motion [5] for hand skeleton tracking [6], mobile devices for hand movement detection [7], [8], or wearable sensors for wrist or arm motion tracking [9]. However, affordable VR systems such as Meta Quest 2 [10] and HTC Vive [11] now come with two handheld controllers equipped with optical and inertial sensors, providing accurate head and hand movement tracking in 3D. Therefore, developing gesture-based interaction techniques that work directly with handheld controllers would be more appealing to VR end-users and enlarge accessibility [2], [3].

While gesture-based interfaces with handheld controllers have been implemented in specific VR games [2], [12], [13] and other 3D applications [14], these gestures are often tailored for specific tasks like tennis swing and may lack generalizability. Most importantly, these interfaces typically only involve single-hand gestures. However, bimanual interactions hold significant promise in numerous VR applications [3], especially for tasks demanding coordinated use of both hands, like in virtual surgery systems [15], [16]. A recent study [3] highlighted various motion patterns in dual-hand interactions, varying from two hands performing simultaneous movements to asymmetric roles where one hand acts predominantly and the other provides a reference point. This underscores the need for advanced hand gesture recognition methods to discern these varied motion patterns and accurately classify dual-hand gestures for comprehensive 3D interactions.

Deep learning models, particularly Recurrent Neural Networks (RNN) [17], [18] and Convolutional Neural Networks (CNN) [19], [20], have been extensively utilized in sensor-based human activity recognition (HAR) [21]. Temporal Convolutional Networks (TCN) [22], [23] that use a hierarchy of temporal convolutions have outperformed other temporal models like Long-Short Term Memory (LSTM) and RNN [24] due to their efficiency in handling long-range temporal dependencies. 2D CNNs are recognized for their strong spatial data processing and accuracy in HAR [25], and the recent integration of Continuous Wavelet Transformation (CWT) with CNN has further boosted classification accuracy by capturing features across both frequency and time domains [26], [27], [28]. While some studies have investigated 1D CNN for single-hand VR gestures [2], [14], research into deep learning approaches for dual-hand gesture recognition in VR is still underexplored.

To address the research gap, we proposed CWT-CNN-TCN, a novel neural network designed to recognize dual-hand gestures in VR. CWT-CNN-TCN is a unified structure of two neural networks, CWT-CNN and TCN, and can simultaneously extract features from the time-frequency domain and capture long-term dependencies for gesture classification using the 3D position and orientation data from both handheld controllers. To evaluate the proposed model, we designed 13 dual-hand motion gestures selected from four fundamental forms of 3D interactions [3]: translation, rotation, scaling, and selection. These gestures encompass various motion patterns, including symmetric vs. asymmetric movements and circular vs. linear motions [3]. We then collected VR gesture data from 26 participants using a VR system from our prior study [29] and assessed CWT-CNN-TCN's performance in recognizing dual-hand gestures with various tracking inputs. This involved comparisons between dual-hand and single-hand inputs and between complete (3D position and 3D orientation) versus partial (3D position or 3D orientation) motion features. Additionally, the network's performance was benchmarked against four state-of-the-art neural networks.

Our goal in this work is to devise a neural network that can effectively recognize dual-hand gestures, even with limited hand-tracking input. This work presents the first step toward a dual-hand gesture-based interface that facilitates intuitive 3D interactions in VR. The main **contributions** of this paper are as follows:

1) We introduce CWT-CNN-TCN, a novel deep-learning neural network capable of accurately recognizing 13 dual-hand gestures using VR controllers, even with limited hand-tracking inputs.
2) We assess the performance of the state-of-the-art neural networks in recognizing dual-hand motion gestures captured by VR, an area previously unexplored in the literature.

II. RELATED WORK

This section explores literature relevant to this paper in two key areas: 1) 3D interactions in VR, discussing various user interface methods supporting these interactions, and 2) human motion gesture recognition, where we discuss the state-of-the-art machine learning and deep learning methods, especially for recognizing hand gestures, underscoring the novelty of our proposed method.

A. 3D INTERACTIONS IN VR

To support effective hand gesture interactions in VR, various hand-tracking devices for 3D user interfaces (UI) have been studied [1]. These devices are crucial for capturing the user's hand position and orientation in 3D space. Generally, hand-input devices fall into three categories: data gloves, vision-based systems, and motion sensor-based systems [1]. Data gloves offer precise hand gesture representation, facilitating direct manipulation of 3D VR objects [4]. However, their high cost and the fragility of glove sensors often impede their widespread adoption and durability [2]. Vision-based tracking systems, such as Leap Motion, have been investigated to support device-free, bare-hand 3D interactions in both VR [6] and augmented reality (AR) [30] settings.

Despite their potential, these vision-based systems may suffer from a limited field of view and noticeable latency issues. Inaccuracies can arise if the user's hands move too rapidly or beyond the system's tracking area, potentially restricting the usability of these devices in diverse VR experiences [2], [31]. On the other hand, wearable sensors provide a robust and cost-effective manner for tracking hand motions [21]. Given their relevance to this work, our review focuses on sensor-based hand input devices.

With the advent of the Internet of Things (IoT) and mobile computing, inertial sensors (accelerometers, magnetometers, gyroscopes) in wearable or mobile devices have become increasingly popular [21]. These devices, offering derivative measurements of hand position and orientation, such as linear acceleration, are being utilized as input mechanisms for virtual object interaction [7], [8], [9]. In these works, users interact with VR either by holding a mobile phone or by wearing a wearable device on the wrist. Katzakis and Hori [7] designed a mobile user interface that allows users to rotate a 3D object in virtual space, leveraging accelerometer data from a mobile phone. Liang et al. [8] developed a mobile phone-based input device tailored for single-hand navigation in VR, enabling users to turn left and right within the VR space by tilting their phones. Fugini et al. [9] introduced a wearable control device capable of interpreting hand gesture commands for actions like moving forward or drawing a circle in VR. Other studies [12], [13], [32], [33] have investigated single-hand gesture-based 3D gaming input (e.g., swing), utilizing game motion controllers (e.g., Nintendo WiiRemote) equipped with inertial sensors.

Modern VR systems like Meta Quest 2 and HTC Vive integrate both optical and inertial sensors to accurately track the position and orientation of the user's head and hand movements in 3D, enabling a wide range of tasks to be performed using either one hand or both hands. Han et al. [2] detected the user's game actions based on motion gestures presented by the 3D acceleration and orientation of the VR controller in the dominant hand. Fennedy et al. [14] developed a 3D mid-air gesture interface that aids users in executing intended gestures based on 3D position and orientation data provided by a VR controller. A recent study [3] explored user-elicited designs for dual-hand gestures for 3D interactions with VR controllers. However, there still remains a gap in research exploring effective hand gesture recognition methods to facilitate dual-hand gesture-based interactions using these controllers within VR environments.

B. HUMAN MOTION GESTURE RECOGNITION

Human activity recognition (HAR) through motion-tracking data captured from computer vision systems or on-body sensors varies in approach. Sensor-based HAR involves identifying activities (e.g., walking, writing) by analyzing physical movement data from wearables or handheld sensors. We focus on machine learning techniques for recognizing human activities through sensors, including hand motion gestures, due to the relevance of this work. We refer the reader to [21] for a more comprehensive review.

Nowadays, hand gesture recognition employs traditional methods like linear classifiers [12], [32], Hidden Markov Models (HMM) [13], [34], and Dynamic Time Warping (DTW) [9], [35]. Linear classifiers [12], [32], [33] analyze gesture feature vectors but may struggle with complex gesture patterns with subtle differences [36]. HMMs process sequential gesture data as a series of observable events tied to hidden states but can be computationally intensive and less effective with spatial complex gestures [2]. DTW aligns data sequences temporally to identify matches [9] but faces scalability challenges with increasing activity classes [21].

Neural networks have gained popularity in HAR for their automatic feature extraction [21]. Recurrent Neural Networks (RNN), including Long Short-Term Memory (LSTM) variants, are well-suited for sequential HAR data. Rivera et al. [17] and Kim et al. [18] employed LSTM and Gated Recurrent Unit (GRU) networks, respectively, for HAR tasks using wrist-worn sensors. However, RNNs can be computationally demanding and may not fully capture long-range sequential dependencies [21], [24]. Recently, Temporal Convolutional Networks (TCN) [22], [23] that use a hierarchy of temporal convolutions offer efficient long-range dependency handling without information leakage from the future to the past. Nair et al. [24] explored the use of TCNs in HAR to detect daily activities like sitting and walking and obtained better performance over other temporal models such as LSTM and RNN.

Convolutional Neural Networks (CNN) models are popular in HAR due to their high classification accuracy [2], [28]. They treat each data dimension (e.g., x-axis position) as an image channel, merging convolutional outputs to identify activities. Ignatov [20] utilized a shallow 1D CNN with statistical features obtained from accelerometer data for real-time activity recognition. Ronao and Cho [19] enhanced the performance by augmenting accelerometer and gyroscope data with frequency domain information for HAR using CNNs. In VR applications, 1D CNNs have been employed for gesture detection, with Han et al. [2] recognizing single-hand VR gestures for game input and Zhao et al. [37] proposing a two-stream 1D CNN for recognizing body actions and head gestures in VR interactions. While 1D CNNs have yielded satisfactory results, 2D CNNs are often preferred for their enhanced ability to classify spatial data [25]. Wavelet transforms, particularly effective for analyzing non-stationary data [26], [27], are commonly used to convert 1D signal data into 2D images. Particularly, Continuous Wavelet Transform (CWT) excels at localizing signal features in both time and frequency domains, enabling 2D CNNs to classify activities based on these features. Recent studies [26], [27], [28] that leveraged CWT in conjunction with 2D CNNs significantly enhanced activity recognition accuracy.

While deep learning has obtained remarkable results in sensor-based HAR, its application to hand motion gestures for 3D VR interactions remains underexplored. Although previous work [2], [14] employed neural networks for single-hand gesture recognition for VR game input, no research has been done to investigate effective methods for dual-hand gesture recognition for more general 3D interactions. In response, this work proposes the CWT-CNN-TCN model, specifically designed for dual-hand gesture recognition in VR interactions. We also assess the performance of the state-of-the-art neural networks in HAR on dual-hand gesture recognition, comparing their effectiveness to our model using varied hand data inputs.

III. METHODOLOGY

This section begins with a presentation of our design for dual-hand gestures tailored for 3D interactions in VR using two handheld controllers, detailed in Section A. Section B describes the VR system utilized for collecting dual-hand gesture data, along with the procedure for data collection. In Section C, we outline the data pre-processing steps required to prepare the training and testing datasets for machine learning. Finally, Section D explains the specific design considerations of the proposed CWT-CNN-TCN neural network.

A. DEFINITION OF DUAL-HAND VR GESTURES

As illustrated in Fig.1, we defined 13 dual-hand motion gestures selected from four fundamental forms of interactions when manipulating virtual objects in 3D [3]: translation, rotation, scaling, and selection, including 1) Translate-Left (T-L), 2) Translate-Right (T-R), 3) Translate-Up (T-U), 4) Translate-Down (T-D), 5) Rotate-X-Forward (R-X-F), 6) Rotate-X-Backward (R-X-B), 7) Rotate-Y-Clockwise (R-Y-C), 8) Rotate-Y-Counterclockwise (R-Y-CC), 9) Rotate-Z-Left (R-Z-L), 10) Rotate-Z-Right (R-Z-R), 11) Scale-Up (S-U), 12) Scale-Down (S-D), and 13) Selection. **Note** that all the moving directions (e.g., left, right) are relative to the user's first-person view, i.e., the head coordinate system (see Fig.2), to ensure the consistent orientation of gestures in 3D space. For example, when performing the Translation-Right gesture, the user's hand always moves horizontally to the right, regardless of their facing direction in the 3D environment. It can be seen that all the rotation and scaling interactions are performed by both hands rotating along the X, Y, and Z-axis symmetrically, while the rest of the interactions (translations and selection) are performed by assigning the two hands asymmetric roles. The goal of this design is to capture different **motion patterns** (see below) that may occur during bimanual interactions [3], including:

- Symmetric vs. Asymmetric: two hands move simultaneously, performing similar actions (e.g., R-X-F) or being assigned asymmetric roles with one hand moving to the designated direction and the other hand staying stationary, acting as a reference point (e.g., T-R).
- Circular vs. Linear: hands move in a circular motion (e.g., R-Y-C) or in linear motion (e.g., S-U).

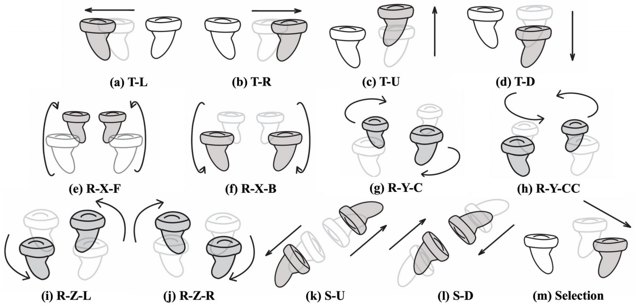


FIGURE 1. Definition of dual-hand motion gestures for 3D interactions in VR, including four fundamental forms of interactions: translation: (a-d), rotation: (e-j), scaling: (k-l), and selection (m). The controllers in motion are highlighted in the shade.

B. DATA COLLECTION STUDY AND VR SYSTEM

1) HARDWARE & SOFTWARE OF VR DATA COLLECTION SYSTEM

To gather dual-hand VR gesture data for 13 interactions, we developed a VR program using the Unity game engine. This program features visual prompts to guide participants: two virtual hand controllers demonstrate each interaction's gesture and a virtual cube displays the interaction outcome (e.g., enlarging a cube for Scale-Up) through animations, as described in previous work [29]. Our study utilized Meta Quest 2, working with our system for data collection, comprising a head-mounted display (HMD) and two handheld controllers. These devices offer six degrees of freedom (DOF), allowing the tracking of the participant's head and hand movements in all directions within the virtual environment, as illustrated in Fig.2. We recorded the HMD and controller tracking data, including position (p_x, p_y, p_z) and orientation in quaternion format (q_x, q_y, q_z, q_w), at a sampling frequency of 90 Hz. This data can be accessed via the XR plugin in Unity. Initially, each tracking device's position and orientation data were measured in the world coordinate system.

2) DATA COLLECTION

26 participants (Age:19-42, Female:8/Male:17/Other:1, Handedness: Right:24/Left:2) participated voluntarily in the Institutional Review Board (IRB) approved study. Each participant performed all 13 interaction gestures sequentially, and they were given the freedom to stand in any position and face any direction in the 3D world space. During each interaction session, an animated visual prompt guided participants, repeating as necessary until they understood the gesture. Participants then executed each gesture 5 times at their preferred pace. Following the completion of each gesture, researchers progressed to the next interaction. Throughout the data collection, visual prompts (similar to

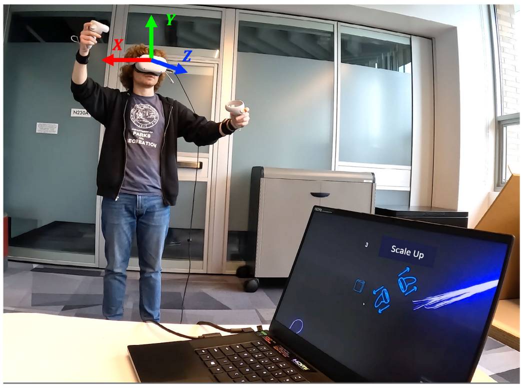


FIGURE 2. A participant was performing a dual-hand interaction gesture - Scale-Up using Meta Quest 2; the VR system for data collection showed the participant each interaction gesture using an animated visual prompt.

those in Fig.1) were displayed to both participants in VR and researchers in a display, see Fig.2. To distinguish gestures from other hand movements, we instructed participants to pull the hand controllers' triggers while performing each gesture and release them when the gesture was completed. This trigger action was recorded along with the motion-tracking data from the headset and both hand controllers. The data, captured frame by frame, was saved locally on the laptop running the VR gesture collection program.

C. DATA PROCESSING

We began by segmenting the data sequences based on trigger action, then processed each resulting gesture data sequence (referred to as a *gesture sample*) through **three steps** for neural network training: 1) orientation vector construction, 2) conversion to egocentric coordinates, and 3) normalization. Our dataset comprised a total of 2,367 gesture samples. The initial average length of these samples, denoting the number of points per gesture (with each point encompassing position and orientation data for both hands), was 58, with a minimum of 26 and a maximum of 151 points per sample. Table 1 details the number of gesture samples and their initial average length.

1) ORIENTATION VECTOR CONSTRUCTION

The orientation of the VR devices was originally captured as a quaternion with four components. Although quaternions are effective in representing 3D rotations and orientations, they can be less intuitive and demand more computational resources compared to 3D directional vectors [38]. To make data processing more efficient and reduce dimensionality, we transformed the quaternion representation into a 3D directional vector for both the headset and controllers. This vector clearly shows the direction the headset or controller is pointing, offering an easy-to-understand and geometrically intuitive orientation representation. For each quaternion, $Q = (q_x, q_y, q_z, q_w)$, in a gesture sample, we first converted it to a 3×3 rotation matrix $\boldsymbol{R}$, [38]. We then multiplied the matrix by a unit vector along the z-axis (0, 0, 1) — the initial pointing direction of the VR headset and hand controller in the world space — to derive a 3D direction vector, $\boldsymbol{d} = (d_x, d_y, d_z)$, as shown in (1) – (2).

$$\boldsymbol{R} = \begin{bmatrix} 1-2q_y^2-2q_z^2 & 2q_xq_y-2q_zq_w & 2q_xq_z+2q_yq_w \\ 2q_xq_y+2q_zq_w & 1-2q_x^2-2q_z^2 & 2q_yq_z-2q_xq_w \\ 2q_xq_z-2q_yq_w & 2q_yq_z+2q_xq_w & 1-2q_x^2-2q_y^2 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \boldsymbol{R} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2q_xq_z+2q_yq_w \\ 2q_yq_z-2q_xq_w \\ 1-2q_x^2-2q_y^2 \end{bmatrix} \quad (2)$$

Therefore, each point in a gesture sample can be represented by a 3D vector for position and a 3D vector for orientation for each tracking device (i.e., one HMD and two controllers). The rationale behind using a 3D directional vector over a quaternion is further elaborated in the discussion section.

2) EGOCENTRIC COORDINATE CONVERSION

Initially, the motion tracking data of the user's hand controllers are measured in the world coordinate system. To account for users performing the same gestures while standing and facing different directions, we converted the 3D position and orientation data from the world coordinate system to an egocentric coordinate system, or head space, which is a left-handed Cartesian coordinate system centered at the user's HMD, with the coordinate axes' basis vectors $\boldsymbol{X}$ pointing right, $\boldsymbol{Y}$ pointing up, and $\boldsymbol{Z}$ pointing forward (refer to Fig.2). Each gesture sample comprises motion sequences from three VR tracking devices (1 HMD and 2 hand controllers). In the world space, each point P in these sequences is represented by a 3D position (p_x, p_y, p_z) and orientation (d_x, d_y, d_z). To establish the head coordinate system for each gesture sample, we first calculate the average position $\boldsymbol{p^h_{world}} = (p^h_x, p^h_y, p^h_z)$ and orientation $\boldsymbol{d^h_{world}} = (d^h_x, d^h_y, d^h_z)$ of the head in the world space by averaging all the head positions and orientation vectors across the entire sequence. We then use cross products based on $\boldsymbol{d^h_{world}}$ and any vector $\boldsymbol{t}$, e.g., $\boldsymbol{t} = (0, 1, 0)$, that is not collinear with it to derive the head space's basis vectors $\boldsymbol{Z}$, $\boldsymbol{X}$, and $\boldsymbol{Y}$ using (3) – (5).

$$\boldsymbol{Z} = \frac{\boldsymbol{d}^h_{\text{world}}}{\|\boldsymbol{d}^h_{\text{world}}\|} \quad (3)$$

$$\boldsymbol{X} = \frac{\boldsymbol{t} \times \boldsymbol{Z}}{\|\boldsymbol{t} \times \boldsymbol{Z}\|} \quad (4)$$

$$\boldsymbol{Y} = \boldsymbol{Z} \times \boldsymbol{X} \quad (5)$$

Next, We transform each point in a controller's sequence in world space, including 3D position $\boldsymbol{p^c_{world}} = (p^c_x, p^c_y, p^c_z)$ and orientation $\boldsymbol{d^c_{world}} = (d^c_x, d^c_y, d^c_z)$, to the head space using (6) – (9), yielding position $\boldsymbol{p^c_{head}} = (p^c_X, p^c_Y, p^c_Z)$ and orientation $\boldsymbol{d^c_{head}} = (d^c_X, d^c_Y, d^c_Z)$ in the head space, where $\boldsymbol{t^c_{world}}$ is the relative position of each point in a controller's

TABLE 1. Statistics of gesture samples for each interaction for training and testing sets (8:2). The initial average length for each gesture is denoted in parentheses.

Interaction Gesture	Total #Samples	Training	Testing
T-L	184 (53)	138	46
T-R	183 (56)	151	32
T-U	180 (47)	139	41
T-D	181 (49)	138	43
R-X-F	190 (71)	152	38
R-X-B	185 (66)	145	40
R-Y-C	185 (62)	151	34
R-Y-CC	177 (61)	148	29
R-Z-L	182 (60)	147	35
R-Z-R	182 (60)	143	39
S-U	182 (57)	154	28
S-D	175 (50)	139	36
Selection	181 (59)	148	33
Total	**2367**	**1893**	**474**

sequence to the head position in the world space, and $\boldsymbol{R}_{w\rightarrow h}$ is the transformation matrix from world space to head space. We refer the reader to the chapter about coordinate system transformation in [39] for more information.

$$\boldsymbol{t}^{c}_{\text{world}} = \boldsymbol{p}^{c}_{\text{world}} - \boldsymbol{p}^{h}_{\text{world}} \tag{6}$$

$$\boldsymbol{R}_{w\rightarrow h} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{Y} & \boldsymbol{Z} \end{bmatrix}^{T} \tag{7}$$

$$\boldsymbol{p}^{c}_{\text{head}} = \boldsymbol{R}_{w\rightarrow h} \cdot \boldsymbol{t}^{c}_{\text{world}} \tag{8}$$

$$\boldsymbol{d}^{c}_{\text{head}} = \boldsymbol{R}_{w\rightarrow h} \cdot \boldsymbol{d}^{c}_{\text{world}} \tag{9}$$

This transformation from world space to the user's egocentric head coordinate system eliminates the need to include head tracking data in model training, reducing the input data dimensionality from 18 (6 each for HMD and hand controllers) to 12 (6 for each controller). This standardizes inputs to the neural network, ensuring more invariant and robust gesture recognition without compromising user experience in the VR environment.

3) NORMALIZATION

Each gesture sample in our dataset, representing simultaneous 3D positions and orientations for both hands, comprises 12 signals. We applied min-max normalization to each signal across the entire gesture dataset to standardize the data range. To ensure consistency in input size for the machine learning model, we interpolated each gesture sample to a uniform length of $T = 128$ time steps. This length was selected to balance the uniformity of input size with the retention of crucial information, preventing the potential loss of detail in longer gesture sequences. The interpolation used B-Spline with cubic interpolation, facilitated by the Scipy library [40].

Following normalization and interpolation, we partitioned the complete VR gesture dataset into training and testing sets, maintaining an 8:2 ratio. Table 1 displays the distribution of all interactions across these two sets, ensuring a balanced representation of gestures in both the training and testing phases.

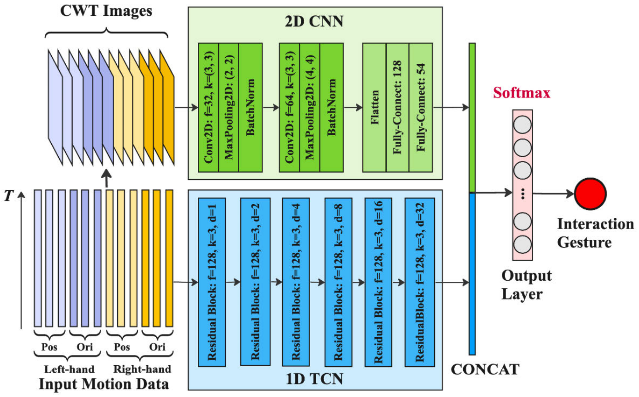


FIGURE 3. CWT-CNN-TCN architecture combines a CWT-CNN (top) - Continuous Wavelet Transform (CWT) in conjunction with a 2D CNN with a 1D Temporal Convolutional Network (TCN) consisting of 6 residual TCN blocks of dilated causal 1D convolution layers with dilation factors, $d = 1, 2, 4, 8, 16, 32$ respectively (bottom). The model takes 3D position and orientation data for both hands as input, and the outputs from these two networks are merged and passed into a fully connected layer with a size of 13 for dual-hand gesture classification.

D. CWT-CNN-TCN MODEL FOR DUAL-HAND MOTION GESTURE RECOGNITION

In this paper, we propose the design of a deep learning model named CWT-CNN-TCN for recognizing dual-hand VR motion gestures. As illustrated in Fig3, the CWT-CNN-TCN model combines a 2D Convolutional Neural Network (CNN), which takes the Continuous Wavelet Transform (CWT) of the hand motion data, with a 1D Temporal Convolutional Network (TCN) that receives the hand motion data directly. Through this structure, the proposed model can simultaneously extract meaningful features from the time-frequency domain through CWT-CNN and capture long-term dependencies in the input VR motion tracking data via TCN. Specifically, this model processes 3D position and orientation data for both hands as input, and the outputs from these networks are merged and passed into a fully connected layer with a size of 13, corresponding to the number of interaction gestures to be classified. The model identifies the final dual-hand gesture by choosing the gesture with the highest softmax-activated prediction. For this multi-class classification, we applied cross-entropy as the loss function and used the Adam optimizer. We discuss the design considerations for each of the models in the following section.

1) CWT-CNN

We use Continuous Wavelet Transform (CWT) to convert hand motion tracking signals into the time-frequency domain, represented by 2D images suitable for processing by a 2D CNN [28]. Each of the 12 signals in a sample is transformed into a scalogram using Morlet mother wavelets, with scale values from 12 to 75 - identified as optimal in our experiments. This results in 12 scalograms of size 64 × 128, stacked to form a 12-channel image (64, 128, 12). Fig.4 illustrates CWT images for gestures including Selection,

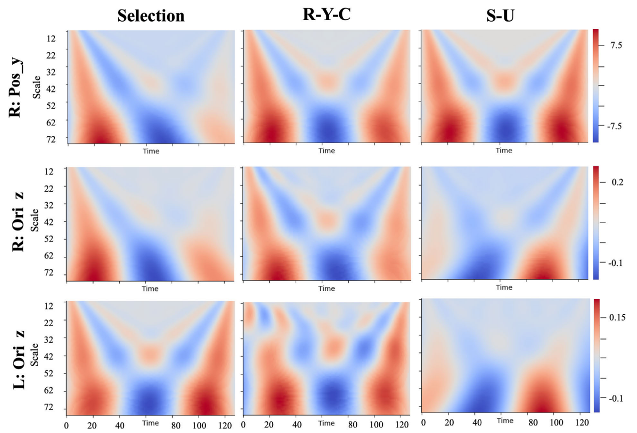


FIGURE 4. Continuous Wavelet Transform (CWT) images of three selected hand-tracking signals for three distinct interaction gestures: Selection (left column), Rotate-Y-Clockwise (R-Y-C, middle column), and Scale-Up (S-U, right column). The top row illustrates the right hand's position along the y-axis ($R : Pos_Y$), the middle row depicts the right hand's orientation along the z-axis ($R : Ori_Z$), and the bottom row shows the left hand's orientation along the z-axis ($L : Ori_Z$).

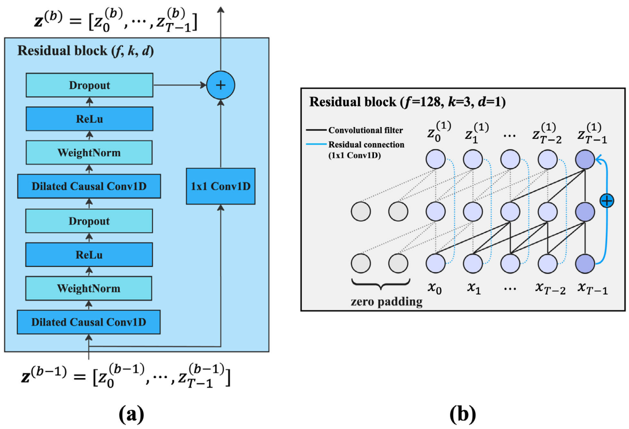


FIGURE 5. (a) The b-th TCN residual block (f, k, d) consists of two 1D convolution layers of the filter size f, kernel size k, and dilation factor d, followed by weight normalization, ReLU activation, and a dropout layer. A 11 convolution is added to handle input and output of different lengths. (b) The first residual block ($f = 128$, $k = 3$, $d = 1$), where the dilated causal convolution operations are indicated in black lines, while residual connections are shown in blue lines.

Rotate-Y-Clockwise (R-Y-C), and Scale-Up (S-U), highlighting that orientation data typically produces more distinct scalograms than position data from the right hand. For improved computational efficiency and memory management, we downsample the scalograms by a factor of 2 along the time axis, yielding images of shape (64, 64, 12). These processed CWT images are input into a 2D CNN for feature extraction.

Considering the images have small dimensions and low complexity, we employed a simple 2D CNN architecture. As depicted at the top of Fig. 3, it comprises 2 convolutional layers – the first with 32 filters of size 3×3 and a 2×2 max-pooling layer, and the second with 64 filters of size 3×3 and a 4×4 max-pooling layer. Each max-pooling layer is followed by batch normalization. A flattening layer precedes two fully connected layers with 128 and 54 units, respectively, both using rectified linear unit (ReLU) activation. The output from the CWT-CNN is a 1D array of length 54, ready to be combined with the output from the 1D-TCN, detailed subsequently.

2) TCN

A TCN excels in processing sequential data by utilizing two techniques: causal convolution, which ensures predictions are based solely on past information, and dilated convolution, allowing the network to learn from a more distant past. In this work, we employ a dilated TCN consisting of 1D dilated causal convolution layers, where filters skip over inputs at increasing intervals, allowing the network to cover a larger range of input with fewer layers. We further add residual connections to the causal dilated convolution layers to preserve the information from the initial layers and improve the performance of TCN, as informed by [23].

As illustrated at the bottom of Fig.3, our 1D TCN consists of 6 residual blocks with dilation factors $d = 1, 2, 4, 8, 16, 32$, respectively. Each residual block (denoted as b, where $b = 1, 2, \ldots, 6$) contains two layers of 1D dilated causal convolution (filter size $f = 128$ and kernel size $k = 3$), followed by weight normalization, ReLU activation, and a dropout layer, as shown in Fig.5(a). The input of the $b - th$ residual block, $z^{(b-1)} = [z_0^{(b-1)}, \ldots, z_{T-1}^{(b-1)}]$, is combined with the output of the two dilated causal convolutional layers to form the input for the next residual block. When the lengths of the residual input and output differ, a 1×1 convolution adjusts this discrepancy. Fig.5(b) shows the first residual block ($f = 128$, $k = 3$, $d = 1$), demonstrating how the input sequence of length $T = 128$, $x = [x_0, x_1, \ldots, x_{T-1}]$, is processed. The diagram highlights the interactions of dilated causal convolution operations (black lines) and residual connections (blue lines) influencing the last step in the output, $z_{T-1}^{(1)}$. As the dilation factor increases in successive residual blocks, the longer input history of the motion is taken into account. The final output of the 1D TCN, a 1D array of length 128, is obtained from the last time step of the last residual block and is then concatenated with the output from the CWT-CNN before proceeding to the final output layer.

IV. EXPERIMENTS AND RESULTS

The proposed CWT-CNN-TCN model was implemented using TensorFlow 2.13 GPU version with Python 3.8. In our experiment, we evaluate the performance of the CWT-CNN-TCN model in recognizing dual-hand gestures using a testing set comprising 474 gesture samples evenly distributed across all the 13 interaction gesture categories, as described in Section III-C. To assess the model's effectiveness across various input conditions and understand its performance with limited motion tracking information, we investigated four input hand conditions: 1) Dual hands with *complete motion features* (i.e., both 3D position and 3D orientation

TABLE 2. Summary of the overall dual-hand gesture recognition performance (accuracy, %) of the proposed CWT-CNN-TCN (Ours) and four benchmark models: CWT-CNN, TCN, 1D-CNN, and LSTM, averaged across all 13 interaction gestures. Each neural network is evaluated under 9 input configurations. The highest classification accuracy are shown in bold.

Input Config.		Ours	CWT-CNN	TCN	1D-CNN	LSTM
Dual	Pos.+Ori.	**98.73**	97.26	96.62	96.84	92.62
	Pos.	96.41	**97.26**	95.36	95.36	88.61
	Ori.	**96.20**	95.36	95.57	93.25	84.60
Right	Pos.+Ori.	**95.57**	**95.57**	93.88	92.41	85.02
	Pos.	**91.77**	90.93	86.08	81.22	67.72
	Ori.	**93.04**	91.35	91.35	84.18	75.74
Left	Pos.+Ori.	**80.80**	77.43	74.05	74.68	61.81
	Pos.	**78.90**	71.73	69.62	62.87	53.59
	Ori.	**70.46**	65.40	64.77	63.50	47.47

data), 2) Dual hands with *partial motion features* (i.e., either 3D position or 3D orientation data); 3) Single hand (left or right) with complete motion features; 4) Single hand (left or right) with partial motion features. These conditions create **9 input configurations** (see the first two columns of Table 2) that were used to evaluate our model. To clearly understand the CWT-CNN-TCN's performance in recognizing individual interaction gestures under different input configurations, we further compare the results obtained from those 9 input configurations in **three evaluation scenarios**:

1) Dual-Hand: Complete vs. Partial Motion Features
2) Dual-hand vs. Single-hand, both with Complete Motion Features
3) Single-hand: Complete vs. Partial Motion Features

Additionally, we selected four deep learning models commonly used in human activity recognition to benchmark our study. The evaluation of the CWT-CNN-TCN model's performance in dual-hand gesture recognition across different hand input configurations is presented in Section IV-A. Subsequently, in Section IV-B, we introduce the benchmark deep learning neural networks and compare their performance with our model under the same input configurations. In both sections, we focus on analyzing the results for the three evaluation scenarios mentioned earlier to understand how our model and the benchmarks perform with limited hand motion input and their effectiveness in recognizing specific interaction gestures of different motion patterns across various input setups.

A. CWT-CNN-TCN EVALUATION RESULTS

We trained the CWT-CNN-TCN network using a batch size of 128, a learning rate of 0.001, and a training epoch of 60. This training was conducted under 9 input configurations concerning hands and motion features, and each trained model was evaluated using the testing set corresponding to the respective configuration. Table 2 presents our model's average classification accuracy (third column) for each input configuration (first and second columns). It can be observed that when the model is trained with both position and orientation data from dual hands, it archives the highest overall classification accuracy of 98.73%. The confusion matrix in Fig.6 indicates that the model reached 100% accuracy for interaction gestures involving linear motions, such as selection, scaling (up and down), and translations; the only exception was Translate-Down, which had an accuracy of 95.35%. For rotation gestures involving circular motions, R-Y-C, R-X-F, and R-Z-L each achieved 100% accuracy, R-Z-R reached 94.87%, while the remaining gestures reached an accuracy of 97.14%. Moreover, our model obtained 100% accuracy for all the asymmetric gestures (one hand moves and the other hand stays stationary), except for Translation-Down (95.35%). For symmetric gestures where both hands move at the same time, the model reached 100% accuracy on scalings (linear motion) and an average accuracy of 98.24% on rotations. We further discuss the results in three evaluation scenarios as follows.

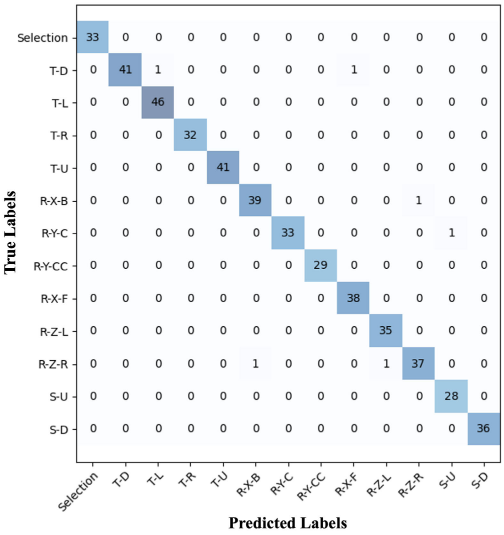


FIGURE 6. Confusion matrix for the CWT-CNN-TCN model's classification results on the testing set. This matrix reflects the performance of the network when trained with both position and orientation data from dual hands.

1) EVALUATION SCENARIO 1: DUAL-HAND: COMPLETE VS. PARTIAL MOTION FEATURES

We compared the CWT-CNN-TCN model's performance in recognizing dual-hand gestures using complete and partial motion features from both hands. The overall recognition accuracy using position only (96.41%) and orientation only (96.20%) were slightly lower than both presented (98.73%), as shown in Fig.7. Upon analyzing individual interaction gestures, the network demonstrated similar performance for most gestures, whether using position or orientation data alone. For instance, the recognition accuracy for gestures T-R, S-U, and S-D remained at 100%, regardless of motion features being used, indicating the network's ability to identify gestures accurately with either position or orientation

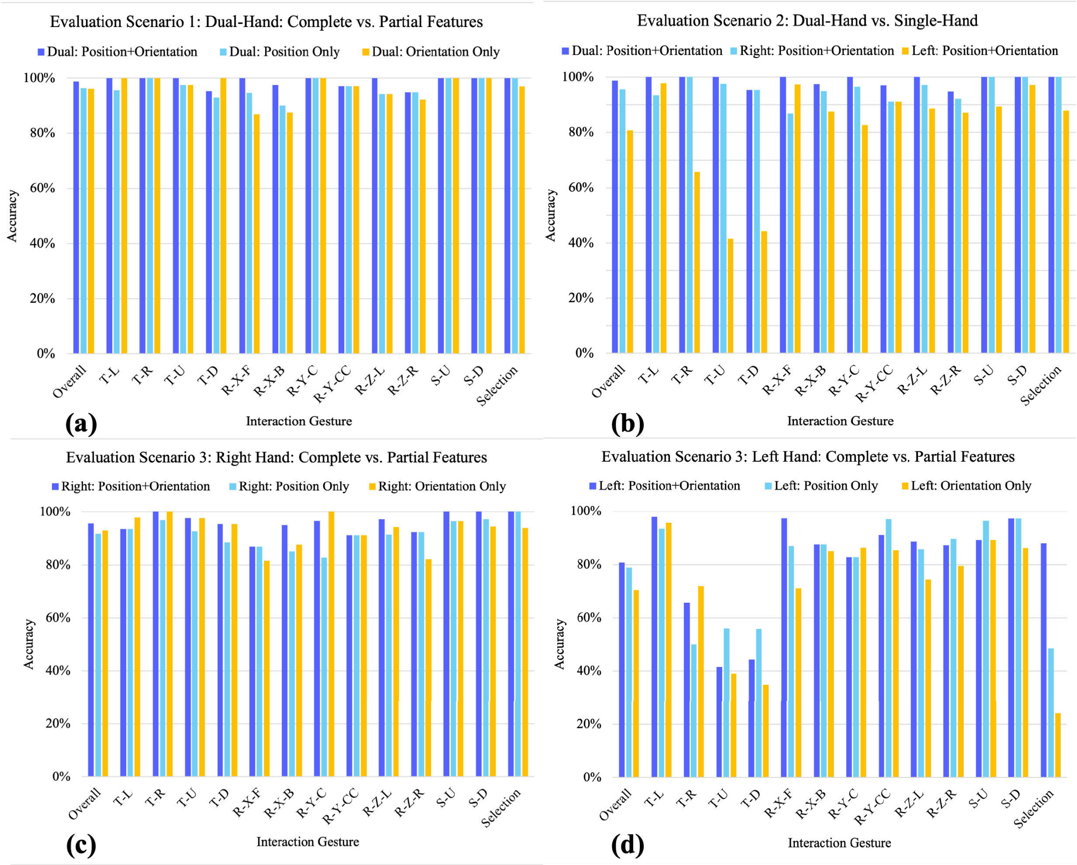


FIGURE 7. CWT-CNN-TCN performance results in (a) Evaluation Scenario 1: Dual-hand: Complete vs. Partial Motion Features, (b) Evaluation Scenario 2: Dual-hand vs. Single-hand, with complete motion features, (c) Evaluation Scenario 3: Right-hand: Complete vs. Partial Motion Features, and (d) Evaluation Scenario 3: Left-hand: Complete vs. Partial Motion Features, for all 13 interaction gestures.

information from both hands. However, performance varied between position ("P") and orientation ("O") data for certain gestures. Specifically, the model trained with position data was more effective in gestures involving two hands moving symmetrically in circular movements. For example, in R-X-F (P: 94.74% vs. O: 86.84%) and R-X-B (P: 90.00% vs. O: 87.50%) - gestures that involve rotating along the x-axis - the model performed better with position data since these gestures typically have both handheld controllers oriented in nearly the same direction. On the other hand, the model trained with orientation excelled in dual-hand gestures primarily involving linear hand movements. For example, in gestures like T-L (O: 100% vs. P: 95.65%) and T-D (O: 100% vs. P: 93.02%), the orientation-based model outperformed the position-based one.

2) EVALUATION SCENARIO 2: DUAL-HAND VS. SINGLE-HAND

To evaluate the CWT-CNN-TCN model's ability to recognize dual-hand gestures using data from a single hand, we trained two separate models, one with right-hand data ("R") and the other with left-hand data ("L"), both incorporating complete motion features. Fig.7(b) illustrates that our model performed better with dual-hand data (98.73%) than with single-hand data for all gestures. Moreover, the model trained using right-hand data achieved a higher overall accuracy (95.57%) compared to the left-hand data model (80.80%). Notably, the right-hand data model exhibited consistently high performance across various gestures and was comparable to both hands presented. Specifically, it achieved 100% accuracy in recognizing L-R, S-U, S-D, and Selection gestures. Conversely, the left-hand data model showed a more varied performance across different gestures. It matched comparable performance for most gestures, particularly those involving symmetric hand motions, such as rotation and scaling gestures. Interestingly, for gestures involving significant left-hand movements, such as T-L (L: 97.83% vs. R: 93.48%) and R-X-F (L: 97.37% vs. R: 86.84%), the left-hand data model outperformed the right-hand data model; however, it underperformed in asymmetric gestures where the right hand actively moved, such as T-R, T-U, and T-D. This outcome is understandable, as the model trained with left-hand data may struggle to extract relevant features from stationary left-hand movements to distinguish these gestures.

3) EVALUATION SCENARIO 3: SINGLE-HAND: COMPLETE VS. PARTIAL MOTION FEATURES

To further investigate the CWT-CNN-TCN network's ability to estimate dual-hand gestures using partial motion features from a single hand, we trained the network on both complete and partial motion inputs derived from either the right or left hand, examining each hand's contributions individually. Fig.7 (c) shows that training with right-hand data, regardless of the motion feature type, yielded similar outcomes, with a slight advantage for combining position and orientation (95.57%) over using only position (91.77%) or only orientation (93.04%). Specifically, the network showed superior performance with the orientation data from the right hand compared to the position data. When trained with the right hand's orientation data, the model excelled in recognizing translations with an accuracy of over 95%. This aligns with our observations from Evaluation Scenario 1, where the model demonstrated better performance in differentiating gestures involving linear motions when trained with orientation data.

Conversely, for models trained with left-hand data, those utilizing complete motion features (80.80%) outperformed those trained on partial features, with position-only data (78.90%) doing slightly better than orientation-only data (70.46%). It was observed from Fig.7 (d) that for gestures requiring active left-hand movement (e.g., T-L) or both hands moving symmetrically, like in rotation and scaling, both position-only and orientation-only data provided similar results, matching closely with the combined position and orientation performance. However, in cases where the left hand was stationary and the right hand was active, such as T-U, T-D, and Selection gestures, the model more effectively utilized the left hand's position data. This suggests that position data more clearly captures the stationary status of the hand compared to orientation data.

B. BENCHMARK EVALUATION AND COMPARISON WITH CWT-CNN-TCN

1) BENCHMARK DEEP LEARNING MODELS

Human activity recognition through motion sensors typically involves classifying spatial-temporal data. Common approaches include using 1D-CNN [20] or LSTM [17], and more recent techniques like CWT-CNN [28] and TCN [24]. However, none of these networks have been applied for recognizing motion gestures made by two hands, and their performance using collected VR tracking data remains unclear. To address this, we developed benchmark models based on 1D-CNN and LSTM, respectively, specifically tailored for dual-hand gesture recognition under various input configurations. Additionally, we implemented CWT-CNN and TCN models separately to assess their individual efficacy in this task. All benchmarks, including their hyperparameters, were optimized for the best performance to facilitate a fair comparison with our model. The 1D-CNN comprises two convolutional layers (32 and 128 filters, respectively, both with a kernel size of 3), each followed by a max-pooling layer (pool size of 2) and a dropout layer (rate of 0.2) to mitigate overfitting. It also includes two fully connected layers (128 and 54 neurons). The LSTM consists of two hidden layers (50 neurons each) with a dropout rate of 0.3 and two fully connected layers (128 and 54 neurons). The configurations for the CWT-CNN and TCN models are identical to those in the CWT-CNN-TCN network. All four models conclude with a 13-neuron output layer using softmax activation for gesture prediction. We implemented and tested these benchmark models in the same computational environment as our model.

Table 2 presents the overall accuracy of each benchmark model for every input configuration. It can be seen that the CWT-CNN-TCN network generally outperformed all benchmark networks in recognizing dual-hand motion gestures. Among benchmarks, CWT-CNN exhibited the best performance, closely followed by TCN. 1D-CNN showed comparable performance to TCN when using data from both hands. For instance, with position data from both hands, 1D-CNN and TCN both achieved a gesture recognition accuracy of 95.36%. However, TCN performed better than 1D-CNN with single-hand data; see the comparison for the right hand with position only in Table 2, where TCN achieved 86.08% and 1D-CNN 81.22%. LSTM's performance was not on par with other networks, particularly with single-hand data using partial motion features (position or orientation).

CWT-CNN-TCN is built by combining CWT-CNN and TCN, both of which outperformed 1D-CNN and LSTM in most input configurations. Therefore, we focus on benchmarking with CWT-CNN and TCN in those three evaluation scenarios mentioned earlier. Moreover, to simplify the comparison between different models on various interaction gestures, we assign individual gestures with similar motion patterns (see details in Section III-A) into **three Gesture Groups**:

- **Translations + Selection**: All translation gestures (left, right, up, and down) are grouped with selection, as they both involve one hand moving linearly while the other hand remains stationary (**asymmetric, linear motion**);
- **Rotations**: All rotation gestures involving both hands moving simultaneously in circular motion are grouped (**symmetric, circular motion**);
- **Scalings**: Scaling up and scaling down, which involve linear motion of both hands, are grouped. (**symmetric, linear motion**).

2) EVALUATION SCENARIO 1: DUAL-HAND: COMPLETE VS. PARTIAL MOTION FEATURES

As shown in Table 3, CWT-CNN-TCN demonstrated higher overall performance with complete motion features (98.73%) and with orientation data only (96.20%) from both hands across all gestures; CWT-CNN (97.26%) performed slightly

TABLE 3. Comparison result between our model CWT-CNN-TCN (Ours) and benchmarks in Evaluation Scenario 1: Dual-hand: Complete vs. Partial Motion Features. The row "Overall" indicates each model's classification accuracy (%) under different input configurations across all interaction gestures. The following rows indicate the average accuracy (%) for each gesture group.

Gesture Group	Dual Hands: Position+Orientation			Dual Hands: Position			Dual Hands: Orientation		
	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN
Overall	**98.73**	97.26	96.62	96.41	**97.26**	95.36	**96.20**	95.36	95.57
Translation+Selection	**99.07**	97.98	97.10	97.25	**98.15**	96.48	98.91	97.98	**98.93**
Rotations	**98.24**	96.01	95.58	95.16	**96.05**	93.50	93.00	**96.01**	91.53
Scalings	100	100	100	100	100	100	100	100	100

TABLE 4. Comparison result between our model CWT-CNN-TCN and benchmarks in Evaluation Scenario 2: Dual-hand vs. Single-hand, with complete motion features. The row "Overall" indicates each model's classification accuracy (%) under different input configurations across all interaction gestures. The following rows indicate the average accuracy (%) for each gesture group.

Gesture Group	Dual Hands			Right Hand			Left Hand		
	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN
Overall	**98.73**	97.26	96.62	**95.57**	**95.57**	93.88	**80.80**	77.43	74.05
Translation+Selection	**99.07**	97.98	97.10	**97.28**	96.14	96.15	**67.40**	58.35	53.30
Rotations	**98.24**	96.01	95.58	93.17	**93.45**	92.08	**89.09**	88.63	88.09
Scalings	100	100	100	**100**	99.32	95.04	93.26	**96.83**	77.38

better than ours (96.41%) with position data only. All models demonstrated enhanced accuracy when complete motion features versus partial ones. Specifically, for scaling gestures involving symmetric linear motions of both hands, each model reached 100% accuracy, regardless of the motion features utilized. Performance varied across the other gesture groups based on the motion features used. In the Translation + Selection group, characterized by asymmetric linear movements, our model excelled with complete motion features at 99.07%, while CWT-CNN led with position data at 98.15%, and TCN slightly outdid our model with orientation data at 98.93% versus our 98.91%. For rotations - gestures involving symmetric circular movements, CWT-CNN consistently outperformed other models with both position (96.05%) and orientation (96.01%) data, although our model still led with complete motion features at 98.24%. TCN slightly underperformed on rotations, particularly with orientation data alone. This outcome was interesting since TCN showed remarkable performance with orientation data in linear motion scenarios. Overall, CWT-CNN and TCN demonstrated performances comparable to CWT-CNN-TCN when using dual-hand data.

3) EVALUATION SCENARIO 2: DUAL-HAND VS. SINGLE-HAND

As illustrated in Table 4, CWT-CNN-TCN consistently outperformed CWT-CNN and TCN, demonstrating higher accuracy with dual-hand (98.73%) and single-hand inputs (right: 95.57%, left: 80.80%) across all gesture groups. All the models achieved improved results with dual-hand inputs over single-hand inputs for each gesture group, underscoring the advantage of using both hand inputs for dual-hand gesture recognition. Particularly when analyzing right-hand data, the CWT-CNN-TCN model maintained the best performance, excelling in Translation + Selection (97.28%) and Scalings (100%), while CWT-CNN marginally outperformed our model in Rotations (93.45% compared to 93.17%). A significant performance decrease was observed across all models when only left-hand data was used, especially for Translation + Selection, due to the lack of sufficient information from the stationary left hand, which occurred for most gestures within that group. Despite the challenges, CWT-CNN-TCN demonstrated superior overall performance (80.80%) in recognizing dual-hand gestures compared to CWT-CNN (77.43%) and TCN (74.05%) using left-hand data alone, leading in Translation + Selection (67.40%) and Rotations (89.09%), with CWT-CNN performing better in Scalings (96.83%). Overall, CWT-CNN-TCN proved more effective than the benchmarks in dual-hand gesture recognition with single-hand data.

4) EVALUATION SCENARIO 3: SINGLE-HAND: COMPLETE VS. PARTIAL MOTION FEATURES

Overall, the CWT-CNN-TCN model demonstrated superior performance in recognizing dual-hand gestures with either right or left-hand data over CWT-CNN and TCN. The results are detailed for each hand below. Table 5 indicates that with right-hand data, CWT-CNN-TCN had higher accuracy in recognizing dual-hand gestures, especially with partial motion features such as position only (91.77%) or orientation only (93.04%). It matched CWT-CNN's accuracy (95.57%) when utilizing complete motion features. Furthermore, models trained with complete motion features consistently outperformed those trained with only position or orientation, with orientation data generally leading to higher accuracy across all gesture groups. Specifically, in the Translation

TABLE 5. Comparison result between our model CWT-CNN-TCN and benchmarks in Evaluation Scenario 3: Right-hand: Complete vs. Partial Motion Features. The row "Overall" indicates each model's classification accuracy (%) under different input configurations across all interaction gestures. The following rows indicate the average accuracy (%) for each gesture group.

Gesture Group	Right: Position + Orientation			Right: Position			Right: Orientation		
	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN
Overall	**95.57**	**95.57**	93.88	**91.77**	90.93	86.08	**93.04**	91.35	91.35
Translation+Selection	**97.28**	96.14	96.15	94.28	**95.72**	89.05	96.94	96.40	**98.12**
Rotations	93.17	**93.45**	92.08	**88.25**	87.65	81.38	**89.43**	85.85	85.18
Scalings	**100**	99.32	95.04	**96.83**	91.92	90.08	95.44	95.44	95.44

TABLE 6. Comparison result between our model CWT-CNN-TCN and benchmarks in Evaluation Scenario 3: Left-hand: Complete vs. Partial Motion Features. The row "Overall" indicates each model's classification accuracy (%) under different input configurations across all interaction gestures. The following rows indicate the average accuracy (%) for each gesture group.

Gesture Group	Left: Position + Orientation			Left: Position			Left: Orientation		
	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN	Ours	CWT-CNN	TCN
Overall	**80.80**	77.43	74.05	**78.90**	71.73	69.62	**70.46**	65.40	64.77
Translation+Selection	**67.40**	58.35	53.30	**60.77**	46.95	47.67	**53.13**	44.39	41.59
Rotations	**89.09**	88.63	88.09	**88.27**	86.91	85.77	**80.22**	74.40	76.05
Scalings	93.26	**96.83**	77.38	**96.83**	92.66	79.17	87.70	91.87	**95.84**

+ Selection group, CWT-CNN-TCN achieved the highest performance (97.28%) with both position and orientation. CWT-CNN was most effective with position only (95.72%), and TCN with orientation only (98.12%). In the Rotations, CWT-CNN-TCN showed the best results with either position (88.25%) or orientation (89.43%), although CWT-CNN marginally surpassed our model (93.45% vs. 93.17%) when combining position and orientation. For Scalings, CWT-CNN-TCN excelled with both position and orientation (100%) and with position only (96.83%), while all networks achieved similar accuracy (95.44%) with orientation only.

As shown in Table 6, when analyzing left-hand data, CWT-CNN-TCN significantly surpassed benchmark models in overall performance, whether using complete motion features (80.80%), position only (78.90%), or orientation only (70.46%). All models trained with complete motion features consistently showed higher accuracy than those trained with only position or orientation. Interestingly, with left-hand data, position data led to higher accuracy than orientation data across all gesture groups, contrasting with the right-hand data outcomes. Furthermore, we noted that gestures involving symmetric hand movements, such as Rotations and Scalings, achieved higher accuracy across all models compared to Translation + Selection, which involves asymmetric hand movements. Specifically, in the Translation + Selection category, CWT-CNN-TCN achieved the highest accuracy with both complete and partial motion features. For Rotations, our model showed strong performance with both position and orientation (89.09%), position only (88.27%), and orientation only (80.22%). In Scalings, CWT-CNN-TCN excelled with position only (96.83%), closely matched by CWT-CNN with both position and orientation (96.83%), and followed by TCN with orientation only (95.84%).

V. DISCUSSION AND FUTURE WORK

Dual-hand interactions have become increasingly prevalent in VR systems, with two handheld controllers now widely accessible [3]. Accurately recognizing dual-hand gestures using VR controllers is essential, yet this area has received limited attention in previous research, which mainly focused on single-hand gestures [2], [14]. Addressing this gap, this paper introduces the CWT-CNN-TCN neural network, designed for recognizing dual-hand gestures with VR controllers. Since our goal is to create a hand gesture-based user interface for intuitive 3D VR interactions, the network is expected to recognize bimanual hand motion gestures representing various motion patterns. We thus designed 13 dual-hand interaction gestures based on four fundamental forms of 3D interactions [3], including moving both hands in a symmetric fashion (rotation and scaling gestures) or moving two hands asymmetrically, with one hand actively moving and the other remaining stationary (translation and selection gestures). We collected 13 types of dual-hand interaction gestures from participants and evaluated the CWT-CNN-TCN's ability to classify these gestures. The performance was assessed using both dual-hand and single-hand data, incorporating complete or partial motion features. Additionally, we compared the effectiveness of various state-of-the-art neural networks often used for human activity recognition (1D-CNN, LSTM, CWT-CNN, and TCN) in recognizing dual-hand gestures under different input conditions.

The results show that CWT-CNN-TCN can accurately recognize dual-hand gestures, utilizing either position, orientation, or both types of motion features from the hands. Its proficiency in extracting features from both the time-frequency domain and long-term dependencies in VR motion tracking data enables it to outperform benchmark models, especially with single-hand data. Notably, the model's accuracy with right-hand data is comparable to that of dual-hand data inputs, demonstrating its capability to discern essential features from limited motion-tracking information. This finding could potentially facilitate the development of hand gesture-based user interfaces compatible with various motion-tracking devices, accommodating scenarios where users may use only one VR controller or have access to only certain types of motion features (e.g., orientations only) from the motion trackers. Next, we will discuss the limitations of the current work and explore future directions to enhance the CWT-CNN-TCN network, aiming to advance hand gesture-based user interfaces for VR interactions.

1) DUAL-HAND MOTION GESTURE DESIGN

The performance of our proposed network was notably robust for rotation and scaling gestures, irrespective of whether the model was trained with data from the left or right hand. This can be attributed to both hands moving symmetrically in these gestures. However, the network was less effective in recognizing asymmetric gestures (such as Translate-Right, Translate-Up, Translate-Down, and Selection) when trained solely with left-hand data. These gestures are typically performed with active right-hand movement and a stationary left hand, leading to a lack of informative left-hand data for accurate prediction. This limitation seems more related to our gesture design rather than the network's capability. To address this issue and be more inclusive for the users who prefer using their left hand more, our future work will explore diverse methods for performing the same gestures. For instance, to perform Translate-Up, a left-handed user could use their left hand, while a right-handed user would use their right hand, as per the current design. This approach, however, might introduce more complex gestures with greater similarities between different classes. We are keen to assess how the CWT-CNN-TCN network performs under these conditions and explore ways to enhance its ability to recognize such complex dual-hand gestures.

2) HAND ORIENTATION REPRESENTATION

In this study, we chose to represent the orientation of VR headsets and hand controllers using 3D directional vectors for dual-hand gesture recognition instead of the original quaternion representation. This decision was driven by the need to reduce feature dimensionality, given our small training dataset. Although quaternions provide more comprehensive information about 3D rotations [38], our gestures did not involve complex rotational movements like rolling. We thus considered 3D vectors adequate for this context, offering computational efficiency and simpler data interpretation. As we plan to include more complex hand movements in VR gesture recognition and expand our data collection, our future work will focus on refining the neural network. This will involve optimizing it to effectively recognize intricate gestures while maintaining computational efficiency and real-time performance, potentially by incorporating more detailed tracking information, such as quaternion-based orientation data.

3) REAL-TIME DUAL-HAND GESTURE RECOGNITION IN VR

This paper presents a neural network designed to recognize dual-hand motion gestures with two VR controllers trained using various hand-tracking inputs. The current network was trained with complete gesture sequences for classification. However, real-time applications often require continuous gesture recognition during performance. In future work, we aim to train the model using a sliding window technique, applying a shorter window that moves along each gesture sequence. This approach will enable more frequent gesture predictions. Additionally, we plan to implement a CWT-CNN-TCN-based dual-hand gesture recognition model within a virtual reality environment to assess its real-time recognition accuracy and speed, particularly in relation to the user's gesture performance speed. Another key aspect of our future work involves the model's adaptability to different input configurations. Currently, separate CWT-CNN-TCN models are trained and evaluated for each input configuration. Moving forward, we aim to refine the network to automatically accommodate various input types, enabling it to make predictions about whether the input comes from a single hand or both hands, with either complete or partial motion features.

VI. CONCLUSION

In this paper, we presented a new neural network, CWT-CNN-TCN, for recognizing dual-hand gestures made with VR handheld controllers. We demonstrated its effectiveness in detecting dual-hand gestures using various experiments and compared it with the state-of-the-art deep-learning neural networks that are often used in human activity recognition. Results showed that CWT-CNN-TCN can accurately predict dual-hand gestures even with limited hand tracking information (e.g., feeding only a single hand data or partial motion features) and outperformed all the benchmark models. To achieve our ultimate goal of devising a dual-hand gesture-based user interface for 3D interactions in VR, the next step is to integrate the CWT-CNN-TCN-based dual-hand gesture recognition model into a VR environment and evaluate gesture recognition accuracy and speed in a real-time setting.

REFERENCES

[1] J. LaViola, E. Kruijff, R. McMahan, D. Bowman, and I. Poupyrev, *3D User Interfaces: Theory and Practice*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2017.

[2] D. H. Han, C. W. Lee, and H. Y. Kang, "Gravity control-based data augmentation technique for improving VR user activity recognition," *Symmetry*, vol. 13, no. 5, p. 845, May 2021.

[3] V. Nanjappan, H.-N. Liang, F. Lu, K. Papangelis, Y. Yue, and K. L. Man, "User-elicited dual-hand interactions for manipulating 3D objects in virtual reality environments," *Hum.-Centric Comput. Inf. Sci.*, vol. 8, no. 1, p. 31, Dec. 2018.

[4] M. Huenerfauth and P. Lu, "Accurate and accessible motion-capture glove calibration for sign language data collection," *ACM Trans. Accessible Comput.*, vol. 3, no. 1, pp. 1–32, Sep. 2010.

[5] (2024). *Leap Motion Controller, Ultraleap, Inc*. Accessed: May 9, 2024. [Online]. Available: https://www.ultraleap.com/

[6] A. Vaitkevič ius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliūnas, and M. Woźniak, "Recognition of American sign language gestures in a virtual reality using leap motion," *Appl. Sci.*, vol. 9, no. 3, p. 445, Jan. 2019.

[7] N. Katzakis and M. Hori, "Mobile phones as 3-DOF controllers: A comparative study," in *Proc. 8th IEEE Int. Conf. Dependable, Autonomic Secur. Comput.*, Dec. 2009, pp. 345–349.

[8] H.-N. Liang, Y. Shi, F. Lu, J. Yang, and K. Papangelis, "VRMController: An input device for navigation activities in virtual reality environments," in *Proc. 15th ACM SIGGRAPH Conf. Virtual-Reality Continuum Its Appl. Ind.*, Dec. 2016, pp. 455–460.

[9] M. Fugini, J. Finocchi, and G. Trasa, "Gesture recognition using dynamic time warping," in *Proc. IEEE 29th Int. Conf. Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*, Sep. 2020, pp. 279–282.

[10] (2024). *Meta Quest 2: Immersive All-in-one Vr Headset*. Accessed: May 9, 2024. [Online]. Available: https://www.meta.com/quest/products/quest-2/

[11] (2024). *HTC VIVE—VR, AR, and MR Headsets, Glasses, Experiences*. Accessed: May 9, 2024. [Online]. Available: https://www.vive.com/us/

[12] M. Hoffman, P. Varcholik, and J. J. LaViola, "Breaking the status quo: Improving 3D gesture recognition with spatially convenient input devices," in *Proc. IEEE Virtual Reality Conf. (VR)*, Mar. 2010, pp. 59–66.

[13] L. Kratz, M. Smith, and F. J. Lee, "Wiizards: 3D gesture recognition for game play input," in *Proc. Conf. Future Play Future Play*, 2007, p. 209.

[14] K. Fennedy, J. Hartmann, Q. Roy, S. T. Perrault, and D. Vogel, "OctoPocus in VR: Using a dynamic guide for 3D mid-air gestures in virtual reality," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 12, pp. 4425–4438, Dec. 2021.

[15] D. Escobar-Castillejos, J. Noguez, L. Neri, A. Magana, and B. Benes, "A review of simulators with haptic devices for medical training," *J. Med. Syst.*, vol. 40, no. 4, p. 104, Apr. 2016, doi: 10.1007/s10916-016-0459-8.

[16] D. Qi, K. Panneerselvam, W. Ahn, V. Arikatla, A. Enquobahrie, and S. De, "Virtual interactive suturing for the fundamentals of laparoscopic surgery (FLS)," *J. Biomed. Informat.*, vol. 75, pp. 48–62, Nov. 2017, doi: 10.1016/j.jbi.2017.09.010.

[17] P. Rivera, E. Valarezo, M.-T. Choi, and T.-S. Kim, "Recognition of human hand activities based on a single wrist IMU using recurrent neural networks," *Int. J. Pharma Med. Biol. Sci.*, vol. 6, no. 4, pp. 114–118, 2017.

[18] J.-H. Kim, G.-S. Hong, B.-G. Kim, and D. P. Dogra, "DeepGesture: Deep learning-based gesture recognition scheme using motion sensors," *Displays*, vol. 55, pp. 38–45, Dec. 2018.

[19] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.

[20] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, Jan. 2018.

[21] L. Minh Dang, K. Min, H. Wang, M. Jalil Piran, C. Hee Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107561.

[22] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1003–1012.

[23] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.012710*.

[24] N. Nair, C. Thomas, and D. B. Jayagopi, "Human activity recognition using temporal convolutional network," in *Proc. 5th Int. Workshop Sensor-Based Activity Recognit. Interact.*, Sep. 2018, pp. 1–8.

[25] O. Pavliuk and M. Mishchuk, "A novel deep-learning model for human activity recognition based on continuous wavelet transform," in *Proc. IDDM*, 2022, p. 236.

[26] G. Q. Ali and H. Al-Libawy, "Time-series deep-learning classifier for human activity recognition based on smartphone built-in sensors," *J. Phys. Conf. Ser.*, vol. 1973, no. 1, Aug. 2021, Art. no. 012127.

[27] A. Nedorubova, A. Kadyrova, and A. Khlyupin, "Human activity recognition using continuous wavelet transform and convolutional neural networks," 2021, *arXiv:2106.12666*.

[28] I. Trabelsi, J. Francoise, and Y. Bellik, "Sensor-based activity recognition using deep learning: A comparative study," in *Proc. 8th Int. Conf. Movement Comput.*, Jun. 2022, pp. 1–8.

[29] M. Raswan, T. Kay, H. M. Camarillo-Abad, F. L. Cibrian, and T. D. Qi, "Guess the gesture: Uncovering an intuitive gesture-based user interface for 3D content interaction in virtual reality," in *Creativity Cognition*. New York, NY, USA: ACM, Jun. 2023, pp. 361–364.

[30] J. Jia, G. Tu, X. Deng, C. Zhao, and W. Yi, "Real-time hand gestures system based on leap motion," *Concurrency Comput., Pract. Exper.*, vol. 31, no. 10, pp. 1–14, May 2019.

[31] N. Breslauer, I. Galic, M. Kukec, and I. Samardic, "Leap motion sensor for natural user interface," *Teh. vjesn.*, vol. 26, no. 2, pp. 1–11, Apr. 2019.

[32] M. Chen, G. AlRegib, and B.-H. Juang, "A new 6D motion gesture database and the benchmark results of feature-based statistical recognition," in *Proc. IEEE Int. Conf. Emerg. Signal Process. Appl.*, Jan. 2012, pp. 131–134.

[33] S. Cheema, M. Hoffman, and J. J. LaViola, "3D gesture classification with linear acceleration and angular velocity sensing devices for video games," *Entertainment Comput.*, vol. 4, no. 1, pp. 11–24, Feb. 2013.

[34] D. Arsenault and A. D. Whitehead, "Gesture recognition using Markov systems and wearable wireless inertial sensors," *IEEE Trans. Consum. Electron.*, vol. 61, no. 4, pp. 429–437, Nov. 2015.

[35] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, "A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors," *IEEE Sensors J.*, vol. 16, no. 16, pp. 6425–6432, Aug. 2016.

[36] G. Bastas, K. Kritsis, and V. Katsouros, "Air-writing recognition using deep convolutional and recurrent neural network architectures," in *Proc. 17th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Sep. 2020, pp. 7–12.

[37] J. Zhao, M. Shao, Y. Wang, and R. Xu, "Real-time recognition of in-place body actions and head gestures using only a head-mounted display," in *Proc. IEEE Conf. Virtual Reality 3D User Interface (VR)*, Mar. 2023, pp. 105–114.

[38] J. Diebel. (2006). *Representing Attitude: Euler Angles Unit Quaternions and Rotation Vectors Semantic Scholar*. [Online]. Available: https://api.semanticscholar.org/

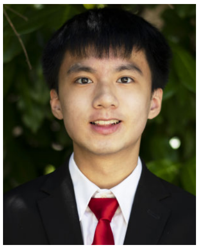
[39] P. Shirley, M. Ashikhmin, and S. Marschner, *Fundamentals of Computer Graphics*. Boca Raton, FL, USA: CRC Press, Jul. 2005.

[40] (2024). *Scipy: Foundamental Algorithms for Scientific Computing in Python*. Accessed: May 9, 2024. [Online]. Available: https://scipy.org/

TRUDI DI QI (Member, IEEE) received the Ph.D. degree from HKUST, with a focus on complex 3D modeling for computer-aided design systems. She was a Postdoctoral Researcher with the Rensselaer Polytechnic Institute, where she served as a Senior Researcher and led technical innovations for multiple National Institute of Health-funded projects of VR surgery systems. She is currently an Assistant Professor of electrical engineering and computer science with the Fowler School of Engineering, Chapman University. Her research interests include computer graphics, virtual reality (VR), and artificial intelligence (AI). She is particularly interested in integrating AI with visual computing technologies and connecting her research to human-centered endeavors, such as education and healthcare.

FRANCELI L. CIBRIAN received the master's and Ph.D. degrees in computer science from the CICESE Research Center, Ensenada, Mexico. She completed her postdoctoral training at UCI. She is currently an Assistant Professor with the Fowler School of Engineering, Chapman University. Her research interests include designing, developing, and evaluating digital health intervention and assessment to support people facing barriers in achieving healthcare, education, and well-being outcomes. She is a fellow of the Joan Ganz Cooney Center's Well-Being by Design Fellowship by Sesame Workshop. She belongs to the National System of Researchers in Mexico, given by CONAHCYT-Mexico.

MEGHNA RASWAN received the B.S. degree in computer science from Chapman University, in 2023, with a focus on game development and visual effects. Her passion lies in game development and leveraging visualization techniques as problem-solving tools. Her research interests include data analysis and machine learning, particularly in the realm of multidimensional visual analytics and human-centered data interaction.



TYLER KAY received the B.S. degree in computer science from Chapman University, in 2023. Throughout his time at Chapman University, he has explored and gained an interest in virtual reality (VR), software development, and data science. He was the Vice President of the Chapman's Data Analytics Association, leading discussions about some of the latest machine-learning discoveries and papers. His research interests include VR, machine learning, and human–computer interaction.

HECTOR M. CAMARILLO-ABAD received the B.S. degree in electrical engineering, the M.S. degree in computer science, and the Ph.D. degree in intelligent systems from Universidad de las Américas Puebla (UDLAP), in 2012, 2014, and 2020, respectively. In 2022, he was a Postdoctoral Researcher with the Schmid College of Science and Technology, Chapman University, where he is currently joining the Grand Challenges Initiative Program. His research interests include analyzing movements through technology, intersecting human–computer interaction, signal processing, and intelligent systems.

YUXIN WEN (Member, IEEE) received the B.S. degree in medical informatics and engineering from Sichuan University, Chengdu, China, in 2011, the M.S. degree in biomedical engineering from Zhejiang University, Hangzhou, China, in 2014, and the Ph.D. degree in electrical and computer engineering from The University of Texas at El Paso (UTEP), in 2020. She is currently an Assistant Professor with the Fowler School of Engineering, Chapman University. Her research interests include statistical modeling, health monitoring and prognostics, and reliability analysis.