HotGestures: Complementing Command Selection and Use with Delimiter-Free Gesture-Based Shortcuts in Virtual Reality

Zhaomou Song (iD), John J. Dudley (iD), Per Ola Kristensson (iD),

Fig. 1: Illustration of conventional menu-based tool selection and HotGestures. Left) The user selects the sphere primitive tool from the menu, moves their hand to where they want to place the sphere, and then performs a pinch gesture to confirm its placement. Middle) The user moves their hand to where they want to place a sphere primitive, and then performs the sphere HotGesture to place it in a single gesture articulation. Right) The user can modulate the state of the HotGesture to control tool triggering, such as to designate the start and end of a planar cut.

Abstract—Conventional desktop applications provide users with hotkeys as shortcuts for triggering different functionality. In this paper we consider what constitutes an effective parallel to hotkeys in a 3D interaction space where the input modality is no longer limited to the use of a keyboard. We propose HotGestures: a gesture-based interaction system for rapid tool selection and usage. Hand gestures are frequently used during human communication to convey information and provide natural associations with meaning. HotGestures provide shortcuts for users to seamlessly activate and use virtual tools by performing hand gestures. This approach naturally complements conventional menu interactions. We evaluate the potential of HotGestures in a set of two user studies and observe that our gesture-based technique provides fast and effective shortcuts for tool selection and usage. Participants found HotGestures to be distinctive, fast, and easy to use while also complementing conventional menu-based interaction.

Index Terms—Gesture interaction, neural networks, gesture recognition, virtual reality

◆

1 INTRODUCTION

Our hands are powerful tools that allow us to complete a wide array of manual tasks and augment our communication with others. Hand interaction therefore offers an intuitive and natural way for engaging with interactive devices. Modern Virtual Reality (VR) headsets provide integrated hand tracking and embed users in a virtual world supporting fully embodied interaction. In VR, users can interact with objects and space in ways that are unrestricted by the limits of the physical world. To enable this vision of VR and apply it within daily routine tasks [7], prior work has sought to enhance and streamline the interaction experience [4, 8, 47]. Hand gestures offer an expressive and efficient form of interaction and have therefore received increasing research attention.

Enabling gesture interaction in VR requires robust and real time processing of hand motion to predict the user's intent. Thanks to recent advances in vision-based hand tracking technology, free-hand interaction in VR has emerged as a viable interaction method. Modern commercial VR headsets, such as the Oculus Quest, natively support vision-based hand tracking. Currently, these headsets provide recognition of basic gestures, such as pinching to select, but do not support a more extensive range of gestures for user interaction. Fast and accurate recognition of complex, multi-frame, dynamic gestures is now possible thanks to modern machine learning techniques but its integration into standard VR applications remains limited. In recognition of this gap, we examine the following research questions (RQs):

RQ1 Can a more extensive range of gestures be recognised continuously to enhance user interaction?

RQ2 Can hand gestures effectively complement conventional mid-air menu-based interaction in VR?

The metaphoric nature of hand gestures helps human beings to communicate and convey messages. Karam and Schraefel [24] presented a gesture taxonomy based on how gestures are used to interact with systems, and Aigner et al. [1] conducted a gesture elicitation study expanding on this concept. Our central hypothesis is that gestures with metaphorical meanings can be associated with system functions to provide shortcuts during free-hand interaction. The meaning attached to hand gestures can serve as a memory-aid that facilitates recollection of shortcuts [16, 23, 31, 34]. Furthermore, such gesture shortcuts can unify the sub-tasks of selection and use. That is, a hand gesture shortcut not only activates a function but also controls its use.

As an example, consider a user seeking to add a sphere to the scene as part of a 3D modeling task. A conventional approach might be to select a sphere primitive from a menu and then place (or *use*) it in the scene by pressing the controller's trigger button or by performing a pinch gesture. We propose an alternative and complementary approach where the user can perform a gesture with fingers clenched to form a ball which both triggers the function to insert a sphere primitive but also places it at the location of the hand. These two alternatives approaches are illustrated in Fig. 1.

We refer to our conceptualization of hand gestures as shortcuts as *HotGestures*. HotGestures enable the user to seamlessly switch between different tools by performing different gestures during a task, without having to pause their work to browse a menu or to press a button on the controller. The benefits of combining command selection and control have been previously shown in touchpad [17] and mouse input [43]. We hypothesize that this unification of selection and usage with hand

- *Zhaomou Song is with Cambridge University. E-mail: zs323@cam.ac.uk*
- *John Dudley is with Cambridge University. E-mail: jjd50@cam.ac.uk*
- *Per Ola Kristensson is with Cambridge University. E-mail: pok21@cam.ac.uk.*

gestures will enhance the interaction experience, and help the user stay focused on the current task without breaking the flow of thought. As a vehicle for demonstrating the concept of HotGestures, we evaluated the use of this technique within the context of a 3D modeling task. There are several commercial 3D modeling applications designed for use in VR, including Adobe Medium[1], Gravity Sketch[2] and ShapeXR[3]. However, they all rely on controller and menu interactions.

To enable HotGestures, we built a neural network gesture recognition system that is able to recognize online gestures by performing predictions on an incoming hand joint data stream. To support online recognition without explicit or implicit delimitation means that the system must robustly ignore typical hand gestures and motions that may be similar to but are not HotGestures. We also employ a multi-task structure to not only classify gestures, but also to classify intermediate states of dynamic gestures. This structure is particularly useful when the sub-states of a gesture may carry particular meaning. For example, this allows the system to not only recognize a 'scissors' (cut) gesture but also to classify whether the 'scissors' are open or closed. We refer to such gestures as *Multi-State Gestures* and illustrate this concept in Fig. 1. To our knowledge, we are the first to formalize dynamic 'stateful' gestures in this way and are also the first to offer a neural network structure that supports robust recognition of such gestures.

In summary, this paper makes the following three main contributions:

1. We present HotGestures: a novel framing and implementation of gesture-based shortcuts for command selection and use in VR.

2. We present a multi-task neural network-based recognition system with adaptations that underpins HotGestures. This system delivers real-time online recognition of multi-state gestures—that is, it not only distinguishes between different gestures but also classifies distinct states within gestures.

3. We present an evaluation of the performance and usability of HotGestures and demonstrate its ability to complement standard graphical menus in free-hand interactions in VR.

2 RELATED WORK

2.1 Gesture Interaction in Mixed Reality

Human hands play a significant role to complete physical tasks and communicate with other users in the real world, and Mixed Reality (MR) devices offer an extension of this experience by replacing or enhancing the real world with additional 3D information and interaction, supporting activities such as bare-handed 3D drawing [15]. As hand-tracking technology becomes more accessible and robust, VR research has started to focus on hand gestures as a more natural and intuitive way of interaction [29], opening up possibilities for new ways of efficient interaction, such as the selection of special characters by gesture switching [55]. Masurovsky et al. [33] compared free-hand interaction with controller interaction in a simple grab-and-place task. Surale et al. [57] and Park et al. [40] investigated static hand gestures as a method for mode switching in VR. Hand gesture are also explored in combination with other interaction techniques. Marquardt et al. [32] and De Araùjo et al. [12] proposed continuous interactive spaces that extend touchscreen interaction into gesture and touch on and above the surface. Surale et al. [56] explored the design space of VR interaction with a multi-touch tablet, supporting both screen touch and mid-air gestures. Ruiz et al. [44] conducted a guessability study to elicit user defined gestures for mobile interaction, and one interesting implication was that although some gestures have high agreement scores, they are indistinguishable from meaningless motions requiring some form of explicit delimitation to use them in practice.

Human beings often use gestures to convey information as an enhancement to spoken language, which means gestures often carry specific meanings. Prior work has looked at the metaphorical aspects of gestures, and how they can be used to issue commands during interaction. Arora et al. [3] conducted a gesture elicitation study to identify

user-defined gestures for animation generation in VR and implemented them into an animation design system. They then evaluated the concept using handheld controllers due to limitations in recognition technology. Seol and Kim [48] performed a gesture elicitation study to design hand gestures to mimic the usage of physical tools in VR. Song et al. [53] used a handlebar metaphor for object manipulation, and Hayatpur et al. [20] used gestures to aid precise object manipulation in 3D using a controller. Yan et al. introduced *VirtualGrasp* [67], a technique that allows users to retrieve virtual objects by 'grasping' them in VR. Pei et al. later proposed a similar concept, called *HandInterface* [42], that uses gestures to either imitate the tool shapes or the hand motion of using the tool to achieve object retrieval. They then used template matching to recognize a set of static gestures, and evaluated this technique through two qualitative user studies. *VirtualGrasp* and *HandInterface* share some conceptual similarities with HotGestures but the focus of this prior work was primarily on the design exploration of suitable gestures and the ability of users to recall them. The evaluations presented in *VirtualGrasp* and *HandInterface* relied on relatively simplistic implementations of gesture recognition sufficient for this purpose. However, as a consequence, these prior implementations are ultimately unsuitable for integration in productive free-hand applications. In this work, we aim to examine the potential usability of gesture shortcuts in a more complete and practical setup. The recognition system is online and delimiter-free, allowing us to fully explore the fluidity of hand gesture-based shortcuts. To the best of our knowledge, we are the first to evaluate the usability of this gesture-based interaction technique in a real use case with modern VR hand-tracking capabilities and using a novel robust recognition system.

2.2 Gesture Interaction in Other Platforms

Prior work has investigated gesture interaction applications outside the MR domain [6, 25, 41, 62]. They are mostly based on accelerometer and gyroscope input, or body skeleton tracking, which have many lower degrees of freedom than hand skeletons. Hespanhol et al. [21] conducted a gesture elicitation study and compared different mid-air gestures in a card flipping exercise, yet the work did not include actual gesture recognition. Alanwar et al. [2] used the pointing nature of gestures to combine device selection and control, but did so with inputs using a smartwatch. Xu et al. [64] built a gesture recognition system on smart watches that also allowed users to customize gestures with a touchscreen based application. Schmitz et al. [46] investigated using pinch as a continuous input modality based on finger span.

2.3 Gesture Recognition

Research interest in hand gesture recognition has increased over the past few years [38] and there are recent systematic reviews on the topic [38, 68]. Previous work has explored an extensive range of techniques, including hidden Markov models [27, 35], decision trees [37], support vector machines [11, 45], and deep neural networks (DNN). Among these methods, DNNs have recently attracted more attention because of their superior performance over traditional approaches and have become the dominant approach for action and hand gesture recognition. Prior work has explored various model architectures as well as different input features. For example, Molchanov et al. [36] and Köpüklü et al. [26] used a convolutional neural network (CNN) as the classifier and images of gestures as input, whereas Devineau et al. [13] used a CNN but with hand skeleton data, that is, the 3D coordinates of hand joints. Liu et al. [30] separated hand skeletons into two data streams to distinguish between posture change and hand movement and then passed the data to a CNN to classify a set of dynamic gestures. Both Du et al. [14] and Wang [63] used a recurrent neural network (RNN) and hand skeleton data for prediction. Other derivatives of conventional networks, such as Graph CNN [28, 66] and Long-Short Term Memory (LSTM) [54, 58], have also been used for gesture and human action prediction. In addition, inspired by the recent success of the transformer architecture [61] for sequence-to-sequence tasks, self-attention-based models have attracted significant recent attention in the field of gesture recognition. Both Chen et al. [10] and Shi et

---

[1] https://www.adobe.com/uk/products/medium.html
[2] https://www.gravitysketch.com/
[3] https://www.shapesxr.com/

al. [52] applied self-attention models to benchmark dynamic gesture sets and achieved state-of-the-art results.

However, while the research on gesture recognition is extensive, few prior works acknowledge the intrinsic difference between offline and online real-time recognition and address the challenges of the latter. Molchanov et al. [36] investigated early detection of gestures in unsegmented video streams. Shen et al. [51] developed a toolkit for fast prototyping of recognition models for online hand gestures based on variations of CNN and LSTM architectures. Prior work has looked at multi-task prediction using neural networks for additional outputs [59,60] and a way to facilitate transfer learning [39,64,65,69]. Another area of interest has been generative adversarial networks (GANs) for gesture data augmentation (e.g. [49, 50]).

3 HotGestures: Delimiter-Free Gesture shortcuts

HotGestures provides a rapid and fluid method for triggering and applying application functions by performing hand gestures. We draw an intentional parallel with hotkeys which deliver quick shortcuts for frequently used functions in desktop applications. Indeed, we consider our proposed technique as an equivalent to hotkeys for embodied 3D free-hand interaction.

The design of HotGestures was motivated by two key hypothesized benefits of using hand gestures as a shortcut for command selection and use. First, hand gestures inherently serve as an aide-mémoire for associated function thanks to their ability to carry and convey meaning, whether this be a metaphorical or symbolic association. Many keyboard shortcuts exploit a related method to aid recall by utilizing the first character in the associated function as a hotkey. Second, hand gestures can be used to unify the usually distinct sub-tasks of function selection and use. By this we mean that gestures can be performed at a particular location in space to indicate the place at which the corresponding function should be applied. We hypothesize that these two key benefits will allow users to both quickly recall gestures and also more quickly execute application functions.

3.1 Evaluation Context

The concept of HotGestures described above has broad potential applications in VR. We offer a concrete demonstration of this technique by applying HotGestures in a 3D modeling use case. Our choice of 3D modeling as the context for our evaluations was based on two key factors. First, we see 3D modeling as a challenging interactive task that allows us to assess our proposed gesture shortcuts from a variety of angles, such as speed, accuracy, enjoyment, and immersion, etc. Second, we believe 3D modeling is a use case where delimiter-free gesture shortcuts can potentially improve the user experience by allowing users to freely and fluidly switch between tools.

We implemented a simplified 3D modeling application which provided 10 basic tools. The choice of these tools was, in part, inspired by the functions available in Adobe Medium, a VR-based 3D modeling application with handheld controllers. The application allows the selection of tools by either a traditional menu or by using HotGestures. The 10 tools and their corresponding gestures are illustrated in Fig. 2. Among these gestures, five are static gestures: *Cube*, *Cylinder*, *Sphere*, *Palette* and *Duplicate*, and the rest are dynamic gestures. *Spray* and *Cut* are Multi-State Gestures with four sub-states. We define static gestures as gestures that are performed with a static hand, and can be represented in a single frame. Dynamic gestures are gestures that are performed by hand motion, and are completed across multiple frames. Multi-State Gestures are dynamic gestures with state labels that indicate minor variations of the underlying gesture. To realize and demonstrate the HotGestures technique, a robust hand-gesture recognition model is required. We explain our design decisions regarding the recognition model in the following section.

4 Recognition Model

In this section, we describe the design of the gesture recognition system that underpins HotGestures. There are three main components of this recognition system: the gesture dataset with associated collection
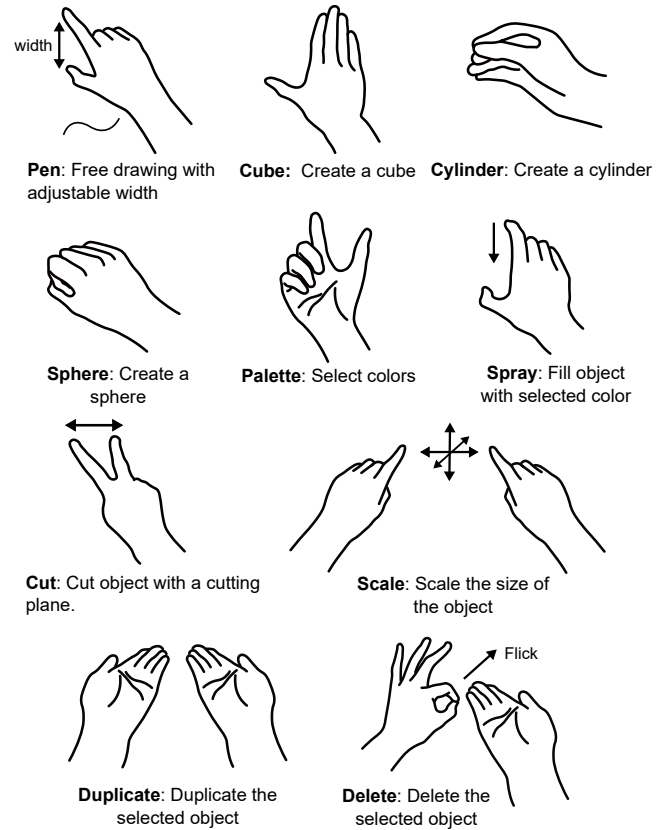

Fig. 2: The tool and gesture set for the 3D modeling application. The arrows indicate the movement direction of a dynamic gesture. (e.g., the arrow on the *Delete* gesture indicates an outward flick of the left index finger.)

procedure, the model architecture, and the training and recognition pipeline. Each of these components is described in detail below.

4.1 Gesture Dataset

To train a recognition model for this work, we collected a customized gesture set of 10 gestures as well as hand movements representative of a *Null* class (see Section 4.1.1). We used the integrated hand tracking functionality of the Oculus Quest 2 VR HMD to obtain the hand skeleton data. We recruited 8 participants and introduced them to the 3D modeling application that provides the application context for the gesture set. Participants were then shown a video demonstrating how to perform each gesture. We logged participants' hand skeleton data as they performed the 10 gestures. Each gesture class was repeated 20 times. For the static gestures, each gesture clip has a fixed length of 2 seconds. For the dynamic gestures, the video was manually clipped to record from the start to the end of the gesture, and each clip lasts approximately 2–5 seconds. We ran the data collection at the native frame rate of the Quest 2, which is approximately 72 Hz. We introduced Multi-State Gestures for *Cut* and *Spray*. These gestures are associated not only with a single gesture class label, but also a series of sub-states (0–3) with a frame-to-frame correspondence (a gesture clip of length T has sub-state labels of length T). The participants were asked to perform the gesture steadily following a progression bar from start to finish. This allowed the sub-states to be automatically labeled according to the progression. These state labels essentially divide a dynamic gesture into discrete segments, allowing more specialized applications for gesture recognition than a single classification output. For *Cut* and *Spray* gestures, the state labels that are smaller essentially represent *cutting down* and *pressing down*, and state labels that are larger represent *not cutting* and *not pressing*. All other gestures in the
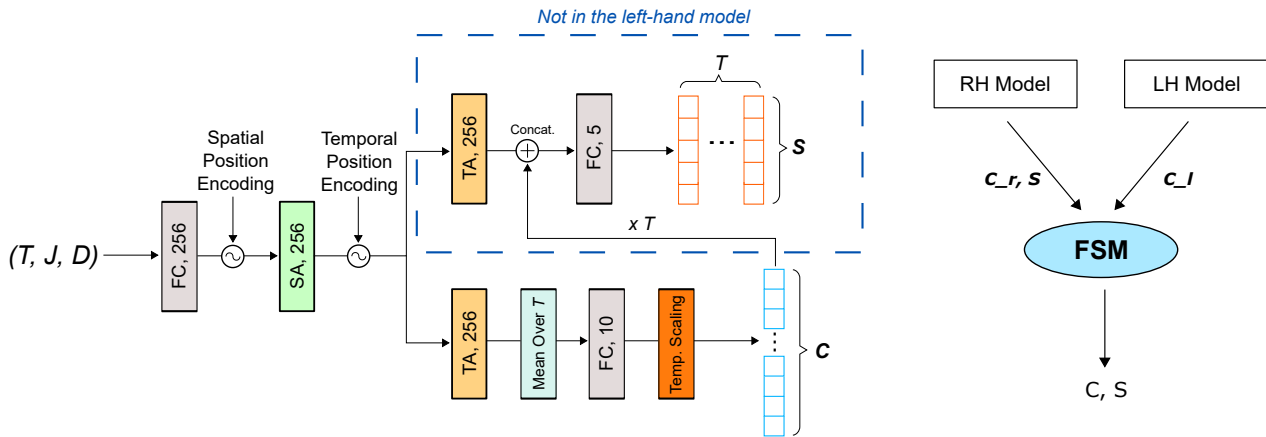
Fig. 3: The recognition network and the framework overview. Left) SA is the masked Spatial Attention block. TA is the masked Temporal Attention block. FC is a linear transformation layer with ReLU activation function and layer-normalization. The number on each block indicates the output dimension. The left-hand model has a similar structure, except that it does not have the sub-state branch, and the output dimension **C** is 4. Right) Both model outputs are jointly considered and passed to the finite state machine (FSM) to obtain the final prediction.

set are labeled with the sub-state '4'. We designed our Multi-State Gestures to have four sub-states and trained the model using all state labels. However, during the actual experiment we collapsed the four sub-states into two, and treated *Cut* and *Spray* as binary-state gestures, that is, open/closed and on/off. This is a design choice we make to better suit the tools. One can make use of all the states by, for example, using different *Spray* states for different color intensity. We built the finer granularity state recognition capability into our system to offer greater extensibility than binary state recognition alone and plan to validate this functionality in future work.

4.1.1 Null Gestures

A common strategy to reduce false activation is to include a *Null* gesture class which captures common hand movements (e.g., hands at rest/grabbing at objects) the user may perform in between intentionally performing gestures. We designed four mini-tasks for the participant to complete during which their hand motion was recorded to collect random gestural behaviors for the *Null* gesture class. In the first two tasks the participants were asked to rest their hands on the laps, and then move their hands up and down. In the third task the participants were asked to touch objects that appeared at random locations, and in the fourth task the participants grabbed objects and moved them from one location to another. In each task, approximately 10 seconds of hand movement was recorded and labeled as *Null* for training. Given that the recognition is delimiter-free, the ability to ignore *Null* gestures is essential to prevent false activation when the user is not intending to perform a meaningful gesture.

4.2 Model Architecture

To recognize both uni-manual and bi-manual gestures, we ran two recognition models in parallel for each hand. The right-hand recognition model is a multi-task self-attention based neural network, which takes fixed-length, hand skeleton graphs of dimension (T, J, D) as input, and predicts the gesture class probability **C** and the sub-state probability of the predicted gesture **S**. A diagram of the model architecture is shown in Fig. 3.

The input dimension T is the number of frames of each skeleton graph, also known as the window size of the gesture. The dimension J is the number of hand joints, and D is the feature dimension of each joint. The network consists of two types of multi-head self-attention blocks: the Spatial Attention block and the Temporal Attention block, similar to Chen et al. [10] and Shi et al. [52].

We extended the model from Chen et al. [10] using a two-head architecture to adapt our concept of Multi-State Gestures. One head

outputs the class probability distribution and the other outputs the sub-state probability of the current gesture. The classification output **C** was expanded T times and concatenated to the output from the second temporal block, and then passed to a final linear layer. The concatenation of class label helps the model to correctly predict sub-states only when a multi-state gesture is classified. The state prediction **S** is a sequence of 5-dimensional vectors with the same length as the input gesture graph, therefore providing a frame-to-frame sub-state estimation of the input gesture. It was found that this separation of Temporal blocks can help learn separate attention weights for both the coarse temporal association of the entire gesture and also the fine relation between frames of multi-state gestures. The final classification output was also passed through a temperature scaling layer [18] before the final prediction.

The left-hand model has the same structure as the right-hand model, except that it does not have the sub-state branch, and the output dimension is **C** = 4 for the three bi-manual gestures and the *Null* class.

4.3 Training and Recognition Pipeline

There is a trade-off between the ability to learn long temporal features with large window sizes and fast inference during online recognition with short window sizes. The increase in window size also scales up the model size at a quadratic rate. We found $T = 20$ yields good performance with short inference delay. We used $J = 11$, including the wrist root, all finger tips, and one joint below each finger tip for prediction as these joints are considered most relevant when performing dynamic gestures. In order for the predictions to be spatially invariant, we used the relative 3-dimensional position and 4-dimensional rotation (quaternion) of the 10 joints with respect to the wrist root, and thus $D = 7$. For the wrist root joint, we padded the position with zeros and kept the absolute rotation for prediction. We separated the data set in Section 4.1 into training and test sets by randomly selecting two participants to be the test subjects, and used the remaining six participants as the training subjects. We cross-validated all possible combinations and chose the model with the best performance on the test set for the user studies. We augmented the training set by adding uniformly distributed noise to 5 random joints in 20% of the data. The model was trained using a NVIDIA GTX 1070 GPU using a standard Adam optimizer and learning rate of 10^{-3}. The weights were updated by a combination of two Cross-Entropy (CE) loss functions, one for the gesture classification task and one for the sub-state prediction task, as shown in Equation 1. $\mathcal{L}_{class}$ represents the CE loss between the class prediction **C** and the class labels. $\mathcal{L}_{state}$ represents the CE loss between the state prediction **S** and the state labels. The parameters $\alpha = 0.8$ and
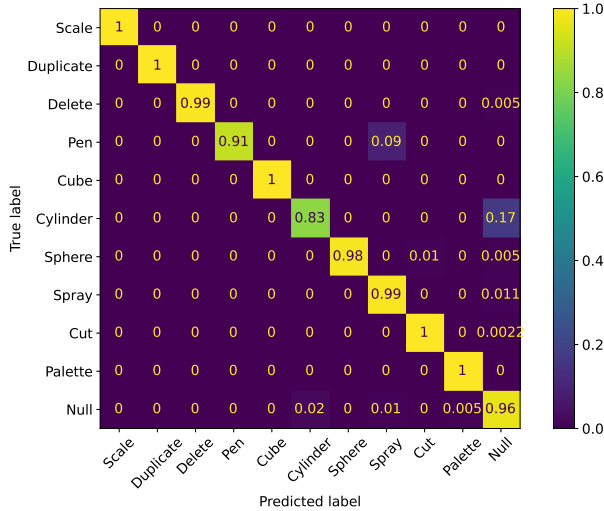
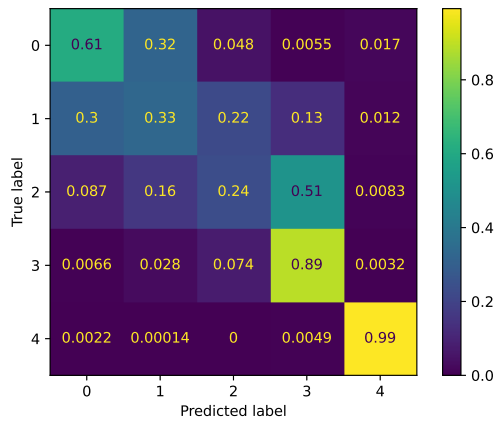Fig. 4: Class Prediction Confusion Matrix.


Fig. 5: State Prediction Confusion Matrix.

$\beta = 0.2$ were the loss weights for back-propagation. We allow training to continue for 5 epochs with no accuracy improvement on validation set, and the average convergence time is approximately 10 minutes.

$$\mathcal{L}_{total} = \alpha \times \mathcal{L}_{class} + \beta \times \mathcal{L}_{state} \qquad (1)$$

4.3.1 Offline Recognition Accuracy

The offline recognition accuracy of the model was calculated by directly comparing the window-level predictions **C** and **S** of both hands with the ground truths. Figure 4 and 5 show the confusion matrices for the window-level predictions of gesture classes and sub-states on the validation set. The model achieves an accuracy above 90% for all gestures except for *Cylinder*. The lower accuracy of the *Cylinder* class results from its confusion with the *Null* gestures. Some of the *Cylinder* gestures will be falsely recognized as *Null*, resulting in a less accurate prediction when making a *Cylinder*. Note the *Null* class has accuracy of 0.96, providing reasonable prediction when hands are inactive.

For the state prediction, the state '4', which represents non-Multi-State gestures, has the highest accuracy, while states '0-2' have low accuracy. This is expected because the state loss weight β is lower than the class loss weight α, forcing the model to learn better at class prediction and be less sensitive about sub-states. Eight out of the 10 gestures are single-state gestures and this imbalance distribution of gesture type will also cause the model to be biased towards less accurate prediction for Multi-State Gestures.
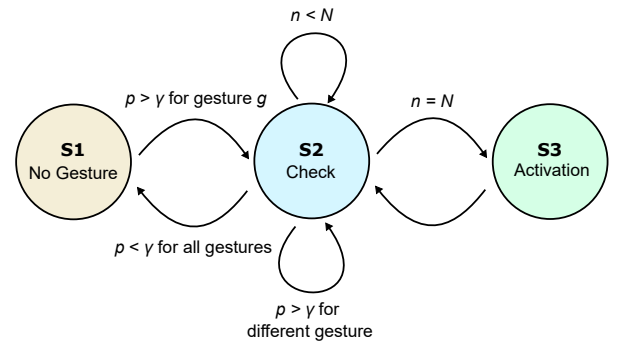
Fig. 6: Finite-State Machine (FSM) for online prediction. *n* represents the counter at S2, and is reset to zero if a different gesture is detected.

4.3.2 Online Recognition

The difference in online recognition is that gesture inputs no longer arrive in segmented, fixed-length windows, but instead in a continuous data stream. We use a data buffer to always select the latest T frames from the data stream as input to predict the gesture class **C** and state sequence **S**. We post-process the outputs **C** and **S** for robust online recognition. The classification output **C** is passed through a softmax function to obtain a probability distribution of the gesture classes, and a gesture is only considered to be detected if its output probability is above a threshold γ. A finite-state machine (FSM), as shown in Fig. 6, is used to process the prediction. When a gesture class that is not *Null* is detected, the machine will move to S2. If the machine receives N gestures of the same class consecutively at S2, the machine will move to S3, fire a gesture output and return to S2. If no gesture probability exceeds the threshold γ during S2, the machine will return to S1. This significantly reduces the oscillation of prediction. A 'majority voting' process is used to pick the most frequent sub-state from the sub-state sequence **S** to be the final state of the current gesture.

We tuned the parameters N and γ using the unsegmented recordings of the 8 participants in Section 4.1 performing all the gestures one after another. During the recording, the participants were asked to return back to rest position before they start the next gesture. We fed the long sequence of gesture data to the model to perform sliding-window recognition with step size of 1 to obtain a sequence of gesture classes **c**, and compared this with the ground truth **y** using the Levenshtein distance between the two sequences. We experimented with different values for the parameters to produce the shortest distance, and found $N = 30$ and $\gamma = 0.9$ to be the optimal choice. We tested our final model on these sequences and found the average percentage Levenshtein distance to be 5.7%, which can be interpreted as a 94.3% online recognition accuracy on our test data stream. We have additionally created a confusion matrix in Fig. 7 for online prediction. Considering that the predicted sequences might have varying lengths compared to the ground truth labels, we inserted null labels to the ground truth where necessary to match the lengths of the predictions. The *Pen* gesture has the lowest accuracy, with 14% being misclassified as *Spray*, illustrating that gestures that have similar form would cause degradation of performance.

We evaluate our proposed interaction technique along with our novel recognition system in a series of two user studies, as explained in the following sections.

5 EVALUATION 1: COMPARING HOTGESTURES AND MENUS FOR COMMAND SELECTION AND USE

We conducted a series of three user studies to evaluate our proposed method of gesture shortcuts in the context of 3D modeling in VR. We implemented our experiments using the Unity Engine and ran them on a Quest 2 HMD via Oculus Link. Our recognition model was implemented using PyTorch, and the sampling rate during the experiments was the same as for the data collection in Section 4.1. In
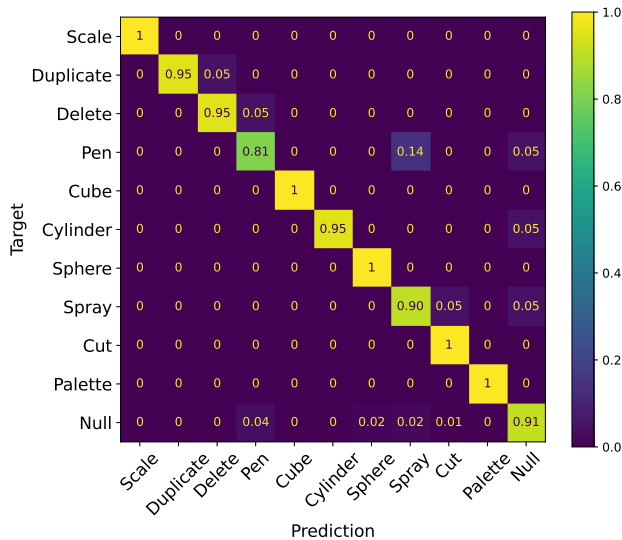
Fig. 7: Online Class Prediction Confusion Matrix.

this first evaluation, we examine HotGestures by comparing it with a baseline representing a standard menu-based tool selection method. Participants completed the study in both experimental conditions.

5.1 Participants

We recruited 16 participants via convenience sampling for the evaluation (average age = 24.4 $\pm$ 3.9, min = 20, max = 34; 9 male, 7 female; all right-handed). No one from the data collection described in Section 4.1 participated in the evaluation. Four participants were undergraduate students while the other participants were postgraduate students or holding a postgraduate degree. Their educational backgrounds were as follows: engineering (13), education (1), philosophy (1), and linguistics (1). In a pre-experiment questionnaire, we asked the participants to rate their level of prior VR experience. Three participants had never experienced VR before, five participants had used VR once or twice before, three participants used VR once a year, and five participants used VR every month or more. We also asked participants to rate their 3D modeling skills on a scale from 1–5 (1 = inexperienced, 5 = very experienced). The mean result was 2.5 $\pm$ 1.2.

5.2 Baseline Choice

To examine how our proposed gesture shortcuts compare with existing methods for tool selection and usage, we chose a baseline representing a well-established and conventional method for selecting and using tools. This baseline involved menu-based tool selections and a pinch gesture to use a selected tool. The menu design was based on demo samples from the Microsoft Mixed Reality Toolkit (MRTK)[4]. The participant can activate a hand-attached, hierarchical menu by raising their left hand and then press buttons on this menu with their right hand to select desired tools. The user can use any selected tool with an index-thumb pinch gesture as the triggering mechanism. For interactions that require more than a trigger (e.g., width of *Pen*, *ScaleX*, *ScaleY*, and *ScaleZ*), a touch slider is provided. The aesthetics of the menu interface strictly followed the default design of the sample MRTK menus. We grouped the 10 tools into 4 categories to make the menu as understandable and accessible as possible. We understand that a menu hierarchy artificially introduces extra steps for interaction, however we suggest it is a reasonable choice for a menu with 10 tools. We chose this method as the baseline because we believe it is the most widely adopted hand interaction technique and one that most users can readily familiarize themselves with.

5.3 Procedure

Participants were shown a video demonstration of each of the gestures introduced in Section 4.1. They were then given as long as they wished to practice these gestures, with feedback provided by the gesture recognition system. Participants were asked to complete the experiment using both conditions (MENU and GESTURE). The order of the conditions was counterbalanced across participants. The experiment is a instruction-based task where participants follow instructions to select and use specified 3D modeling tools.

5.3.1 Familiarization phases

At the beginning of the experiment, the participants were introduced to the 3D modeling tools. Then they were shown video clips of the 10 gestures associated with each tool and were asked to memorize them. Before the start of the experiment, they were given the opportunity to practice with the tools and their first experimental condition. After the first condition was completed, they were given the opportunity to familiarize with the second condition before continuing with the experiment.

5.3.2 Tasks

In this experiment, participants received a sequence of instructions to select and use one of the 10 tools summarized in Fig. 2 (e.g. 'Please Use Cube', 'Please Select this Color'). For each condition, participants needed to complete 6 groups of instructions, with 12 instructions within each group. The order of instructions was randomized within each group. The first instruction group was considered to be a practice group, and the next 5 groups were the actual task. Using a Scale X task as an example, the participants were first shown the task objective and a target object. To indicate they were ready to commence the task, participants placed their hands inside a virtual prism roughly aligned with the surface of a desk. This initialization ensured a consistent starting position. Once their hands left the prism, the timer would start and they could perform the task. Participants were told to scale their object according to the randomized X scale of the target. As soon as the participants scaled the object to the correct size, the task was treated as being complete. We recorded the completion time and selection accuracy of each individual task. An opportunity to take a break was provided before each group. At the end of the study, participants were asked to complete a post-experiment questionnaire capturing the perceived speed and accuracy of each condition.

5.4 Results

5.4.1 Completion Time

The distribution of mean completion time for participants using each tool is summarised in Fig. 8a. Repeated measure analysis of variance on the overall completion time across all tools with a significance level of $\alpha = 0.05$ revealed a significant main effect due to the interaction technique (GESTURE or MENU) ($F_{1,15} = 109.9$, $\eta_p^2 = 0.365$, $p < .001$). This result suggests that the tasks can be completed significantly faster in the GESTURE condition than in the MENU condition. This is consistent with our hypothesis that gesture shortcuts can provide a faster way to select and use tools by avoiding the additional steps involved in menu-based tool selection and use.

Non-parametric Wilcoxon signed-rank test on the completion time for each individual tool, with an initial significance level of $\alpha = 0.05$ and applying Holm-Bonferroni correction for multiple comparisons [22], revealed that the GESTURE method was significantly faster than the MENU for *Pen*, *Cube*, *Sphere*, and *Duplicate*.

The 2-level menu used in this study is consistent with MRTK design guidelines[5] but does introduce one additional step during selection compared to an equivalent 1-level menu. To assess the impact of this particular design decision, we performed a post hoc analysis by removing the time from the first button press to the last button press when computing completion time for MENU. This is based on the assumption that the participant only takes one 'press' to access the

---

[4]https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05

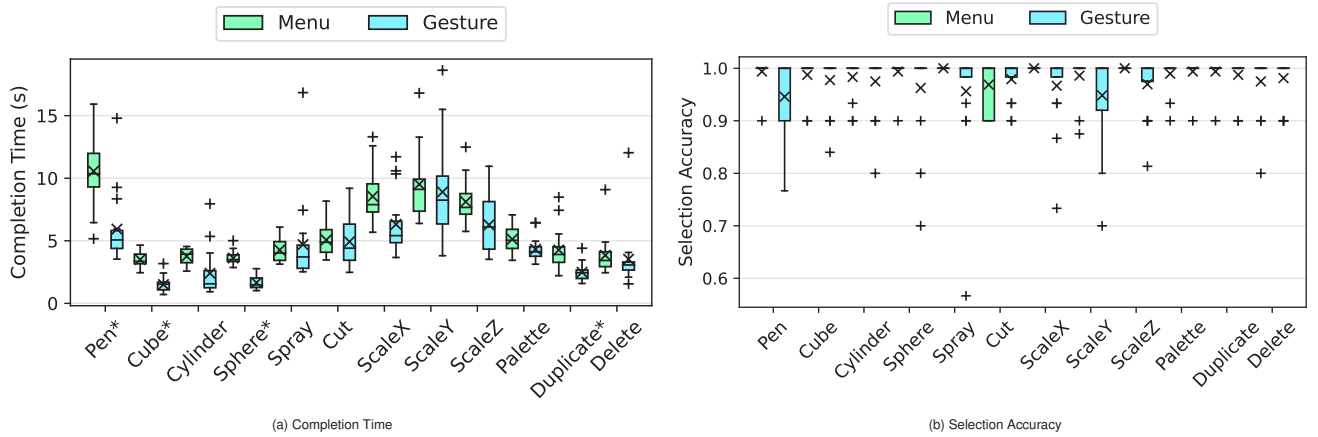[5]https://learn.microsoft.com/en-us/windows/mixed-reality/design/hand-menu

Fig. 8: Distribution of completion time and selection accuracy of Evaluation 1 tasks for each individual tool. The boxplots show the median (the horizontal line), the mean ('x'), the first and third quartile (the box) and the minimum and maximum (the whiskers). The plus signs ('+') indicate outliers. The asterisks ('∗') on x-axis indicate the sample groups that have a statistically significant difference.

desired tool and is always correct on their first try. Note that there is additional task time associated with actually using the tool, which remains unchanged. This analysis revealed that for 7 out of the 12 tasks, GESTURE was still faster, and among those *Pen*, *Cube* and *Sphere* were significantly faster. No GESTURE was significantly slower than the MENU. These results show that our proposed gesture shortcuts indeed provide users with a faster way of selecting and using a majority of tools compared to a conventional menu-based interaction.

5.4.2   Selection Accuracy

A non-parametric Wilcoxon signed-rank test on the overall accuracy with a significance level of $\alpha = 0.05$ revealed that there was a significant difference in overall accuracy between GESTURE and MENU ($z = -3.2$, $p = .001$). We summarize the selection accuracy of each tool during Evaluation 1 in boxplots as shown in Fig. 8b. The average accuracy of selection when using the MENU condition is higher for all tools than the GESTURE condition except for *Cut*, *Palette*, and *Delete*. This is expected as the MENU condition provides more certainty than the GESTURE condition during selection, and it was less likely for participants to make mistakes when using the menu. For the GESTURE condition, the selection accuracy is above 0.95 for all task types. The *Pen* had low accuracy because some participants confused the gesture with *Spray* due to their similarity. Some participants also found the scaling gesture unnatural to use, as they needed to move their hands in a particular way to scale precisely. This could cause discomfort and lead to inaccurate gestural forms.

A Holm-Bonferroni corrected non-parametric Wilcoxon signed-rank test, with an initial significance level of $\alpha = 0.05$, revealed that the difference in selection accuracy for individual tools is not significant for all tools. These results suggest that gesture shortcuts are not significantly less accurate than menu-based selection.

5.4.3   Participant Feedback

At the end of Evaluation 1, participants were asked to complete the perceived workload (NASA-TLX) [19] questionnaire and answer five questions about their perceived speed and accuracy of each condition as well as their overall preference. Figure 9 shows that during Evaluation 1 the participants perceived the GESTURE condition to be more mentally demanding, required more effort, and had performed worse. Although the actual speed and accuracy did not suggest a worse performance, one reason can be that memorizing and recalling different gestures during the task causes more mental burden. Further, since the concept of using gestures for 3D modelling is likely unfamiliar to most participants, it may lead to more hesitation and uncertainty compared to use of a more conventional menu. However, a non-parametric Wilcoxon test did not reveal any statistical significance in the difference in ratings.
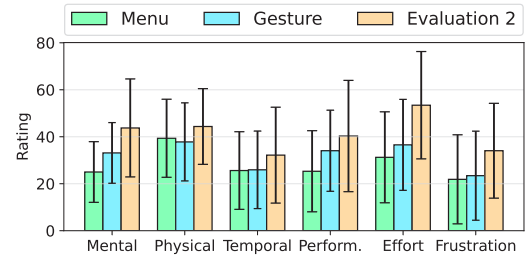

Fig. 9: The perceived workload ratings for Evaluation 1 (MENU and GESTURE) and 2. The error bars show ± 1 standard deviation. A lower 'Performance' rating indicates 'better' perceived performance. Evaluation 1 and 2 concerned different tasks and are therefore not directly comparable.

Table 1 lists the question statements and the participant responses. Participants rated the GESTURE condition higher for 'select and use tools quickly', while the MENU condition was rated higher in 'select and use tools accurately'. This agrees with our previous findings about the completion time and selection accuracy. Overall, 13 participants preferred the GESTURE conditions for this task. Based on these results, it can be concluded that while gesture shortcuts provide additional benefits to the user experience in terms of faster access to desired functionality, they have shortcomings in terms of selection accuracy and additional effort to learn.

Table 1: Median and interquartile range (IQR) of the responses to the five questions in Evaluation 1. Responses for Q1 to Q4 were recorded on a five-point Likert scale from 1—strongly disagree to 5—strongly agree. Q5 lists the number of participants who preferred each method.

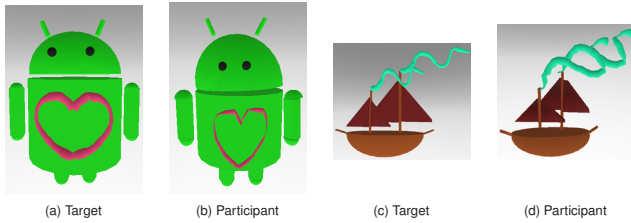|		MENU		GESTURE	
The technique made it easy to...		MEDIAN	IQR	MEDIAN	IQR
Q1	Select tools quickly.	3.5	2	5	1
Q2	Select tools accurately.	4	1.25	3	1.25
Q3	Use tools quickly.	2	1	5	1
Q4	Use tools accurately.	4	1	3	1.25
Q5	Preference for this task?		3		13

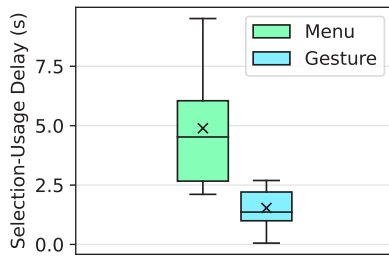Fig. 10: The target 3D models used for Evaluation 2 and illustrative outcomes from participants.

Fig. 11: Distribution of selection-usage delay during Evaluation 2.

Fig. 12: Proportion of usage of each technique during Evaluation 2.

Fig. 13: Participant response on technique preference for each tool.

6 EVALUATION 2: COMPLEMENTING MENU USE WITH HOT-GESTURES IN AN APPLICATION

6.1 Participants

The same group of participants from Section 5.1 took part in Evaluation 2 immediately after they completed Evaluation 1. They were given a short break before the experiment, and were given the opportunity to revise both MENU and GESTURE conditions before they started.

6.2 Procedure

In the second study, participants were asked to reproduce two 'stimulus' 3D models (see Fig. 10) one after the other with balanced order. We designed the targets such that they require all 10 tools to be fully replicated. In this study, however, participants were provided with both the MENU and GESTURE methods, and were able to interchange freely between the two during the study. The aim was to investigate whether this hybrid mode can combine the benefits of both techniques. The participants pressed a start button when they were ready to begin, and the target model would appear on the left-hand side of the workplace and remained visible throughout the study for reference. Participants were told there was no time limit and were asked to continue with the modeling task until they were satisfied with their results. After they completed both models, the participants were asked to complete a series of questionnaires: System Usability Scale (SUS) [9], perceived workload (NASA-TLX), and a customized questionnaire capturing more qualitative feedback. The aims of this evaluation are to examine user behavior when provided with both MENU and GESTURE under a realistic 3D modeling task. Unlike Evaluation 1 where participants were allowed to only use one technique, this experiment encouraged free technique switching at their will.

6.3 Results

6.3.1 Selection-Usage Delay

We computed the participants' average time delay between selecting a tool and successfully using the tool during the modeling task and plot this in Fig. 11. The average time delay between selection and usage is approximately 1.5s for GESTURE, whereas the average time delay for MENU is 4.9s. A repeated measure analysis of variance with significance level of $\alpha = 0.05$ on the log-delays revealed that difference in log-delays is significant ($F_{1,15} = 38.6$, $\eta_P^2 = 0.734$, $p < .001$).

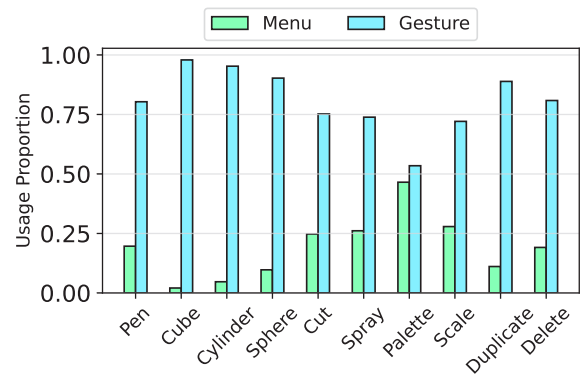This result confirms the central hypothesis underpinning the design of HotGestures: it allows rapid tool selection and usage at almost no delay by executing both actions in one gesture. Unlike the MENU condition which forces participants to shift their focus away from the current task in order to make a new selection, and then shift back to their work to use the tool, HotGestures reduces the time taken for participants to recover from the selection process, allowing them to switch their attention back to the actual task quickly. This demonstrates that HotGestures increases the speed of interaction and potentially helps the user concentrates more on the task.
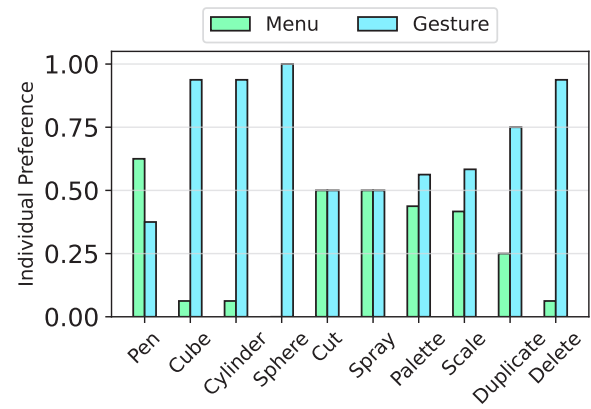
6.3.2 Individual Preferences

We computed the proportion of valid tool selections performed by participants during the task with either GESTURE or MENU (see Fig. 12). All tools had higher proportion of use with the gesture condition, consistent with the overall preference from the participants. The three geometry primitive tools and *Duplicate* were mostly used by gestures only, whereas *Cut*, *Spray*, and *Palette* and *Scale* had higher proportion of menu use at 25% or higher.

Participants were asked to indicate which technique they ultimately preferred to use for each tool, as summarized in Fig. 13. The individual preferences mostly agree with the proportion of selection during Evaluation 2 (see Fig. 12). Most participants preferred gesture shortcuts for the geometry primitives and *Duplicate* and *Delete*, and preferences were approximately equal for other tools except for *Pen*. More participants preferred pen on the menu despite its relatively low proportion of menu use during the task. It was observed that for the menu-preferred tools, participants often attempted to use the gesture first, had unsatisfactory outcomes, and then decided to use the menu. Because such tools require more precise operation, participants were tempted to choose the more reliable menu approach rather than the faster but potentially more error-prone gesture shortcut.

6.3.3 Participant Feedback

Participants completed the questionnaires for the System Usability Scale (SUS) at the end of Evaluation 2. The average SUS score for this hybrid system was 70.6 ± 11.1. According to the results from Bangor et al. [5], a SUS score of 70 means the system usability is at an acceptable level. This suggests that the participants found that the combined provision of a conventional menu with HotGestures as delimiter-free shortcuts delivered good usability. The NASA-TLX results from Evaluation 2 are shown in Fig. 9.

When participants were asked about what they thought made an effective gesture shortcut, most participants mentioned a gesture that is 'easy to use', 'quick', and can be recognized reliably. A more distinctive design was also mentioned as the reason why the gestures were preferred because they were 'easy to remember'. On the other hand, the menu technique requires 'more steps' than the gestures. This agrees with our hypothesis that HotGestures combines tool selection and usage into one action, thereby reducing the time taken to complete tasks. When asked in what circumstances they would prefer to use the menu, two participants said the menu was preferred when the gesture failed to be recognized reliably, making them 'less confident' about using gesture shortcuts. Eleven participants mentioned 'hard to control' or 'accurate operation' when answering why the menu was preferred for tools like *Pen*, *Cut* and *Scale*. These participants thought gesture shortcuts could not offer the same level of control and precision as a menu method because of occlusion of hand-tracking and shaking on the fingertips. When asked about whether they prefer MENU, GESTURE, or a hybrid system, 14 out of the 16 participants preferred the mixture of menu interaction with complementary gesture shortcuts, and two participants preferred the gesture shortcut only approach.

7 DISCUSSION

Evaluation 1 compares HotGestures directly with a representative menu baseline, and the results reveal that our proposed gesture shortcuts sacrifice selection accuracy for fast selection and usage. This is the typical trade-off between speed and accuracy that is commonly observed in Human-Computer Interaction (HCI). The same trade-off is exhibited by conventional shortcut techniques used in other interaction settings. Note that the reduction in accuracy was minor, and some gestures had higher selection accuracy, demonstrating that HotGestures are a complementary alternative to the conventional menu. There are multiple reasons for the lower accuracy of HotGestures, including misrecognitions by the model, the articulation of an incorrect gesture by the user, and inaccurate hand tracking by the headset. Nonetheless the majority of participants still preferred HotGestures to complete the tasks.

Evaluation 2 reveals the underpinning benefit of HotGestures, which is that they reduce the delay between tool selection and use by combining the two operations into one fluid execution within a single continuous gesture articulation. HotGestures had a significantly larger proportion of use during the task overall as the participants thought they were faster and easier to use. There were several drawbacks, however, including the inaccuracy in recognition and difficulty in control for tools that require precision. The *Pen* tool was more preferred with the menu due to these two reasons. It is worth noting that one participant encountered difficulty in selecting the *Spray* tool, even though they performed the 'correct' gesture from a human's perspective. These observations are consistent with the lower online accuracy in Fig. 7 for *Pen* and *Spray*. In contrast, the menu method provided users with more accurate tool selection and usage and a more consistent interaction with better control. Overall one approach forms a suitable counterpart for the other, and a hybrid interaction was preferred by almost all participants over either technique in isolation. It shows that HotGestures form a suitable complementary technique that can benefit conventional interaction by offering optional shortcuts for frequently used system functionality in the system.

From a system design perspective we found that users tend to perform the same gesture differently, even though they were shown the same demonstration video. Therefore a more distinctive gesture design is helpful for both memorability and model prediction. Caution is required when using gestures that are easily mistaken for another gesture.

This also applies for gestures that are more difficult to predict, such as dynamic gestures. Moreover, even if the gesture set is distinctive according to human judgment, uncertainty in model prediction may still remain. The system needs to handle erroneous input appropriately. Gestures with lower accuracy should be assigned a less critical function to avoid catastrophic consequences from false recognition. It was also found that the natural shaking of participants hands made precise operation with gestures difficult at times. This proved to be one of the most common reasons for participants to switch to the menu technique. Therefore gesture shortcuts are most suitable for tasks that require speed more than precision.

8 LIMITATIONS AND FUTURE WORK

The system only considered ten gestures. As the gestures in the system increase, gesture-only interaction becomes problematic as larger gesture sets are more difficult to memorize, recognize and require more training data. We suggest future work explores the scalability of gesture interaction systems further.

The recognition model was trained and calibrated using a 8-subject data set. A larger and more diverse data set would likely improve the models ability to generalize to unseen users. Including a new gesture requires collecting data for that gesture and retraining the model.

The data set was collected using Oculus Quest 2's integrated hand tracking system, which may be subject to occlusion. The participants were only allowed to perform gestures in certain postures and orientations to avoid occlusion, and this limits the variety of the gestures. To collect a dataset with more diversity and higher precision, wearable device based hand-tracking could be a better choice. Although we believe most of the findings in this work are transferable to other platforms, we would like to see whether more reliable hand-tracking can further improve HotGestures.

While we believe 3D modeling is a suitable application for demonstrating the utility of HotGestures, there are certainly other applications where HotGestures can be beneficial as a complementary technique, such as data visualization, video editing, text entry, etc., as the recognition model is not bound to a specific gesture set. We anticipate that gesture shortcuts will be most useful in tasks that value a fast interaction process and a continuous workflow. We plan to explore the effectiveness of gesture shortcuts in other contexts as future work.

9 CONCLUSIONS

In this work we present HotGestures, a gesture-based shortcut system for command selection and use in VR. We introduce a multi-task recognition framework developed to realize this system and trained on a customized gesture set with both uni-manual and bi-manual gestures. We evaluate this system with two user evaluations in the context of a 3D modeling application. In the first evaluation we directly compared HotGestures with a conventional menu baseline and found that HotGestures allow faster tool selection with only a small reduction in accuracy, and were preferred by the majority of participants. There was no significant difference between HotGestures and the menu baseline in terms of perceived workload (NASA-TLX). In Evaluation 2 we combined HotGestures with the menu for a 3D modeling task. The task revealed that HotGestures combine both selection and use into a single gesture and support user tool-based interaction with less delay than the menu. We suggest this served to lower the barrier between user intention and execution. During the task, HotGestures had a larger usage proportion than the menu across all tools, and half of the HotGestures were overwhelmingly preferred by the participants. The System Usability Scale (SUS) score was above 70, indicating that HotGestures complementing a conventional menu had acceptable usability. Participants also commented on the shortcomings of the gesture shortcuts, including inaccuracy, difficulty to control, and lack of precision. The majority of participants therefore appreciated the hybrid approach of allowing the user to choose whether to use the menu or a HotGesture depending on the task and context. Finally, we conclude that HotGestures are a highly promising complementary interaction technique for VR applications since they are distinctive, fast, and easy to use.

REFERENCES

[1] R. Aigner, D. Wigdor, H. Benko, M. Haller, D. Lindbauer, A. Ion, S. Zhao, and J. T. K. V. Koh. Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci. Technical Report MSR-TR-2012-111, November 2012. 1

[2] A. Alanwar, M. Alzantot, B.-J. Ho, P. Martin, and M. Srivastava. Selecon: Scalable iot device selection and control using hand gestures. vol. 2017, pp. 47–58, 04 2017. doi: 10.1145/3054977.3054981 2

[3] R. Arora, R. H. Kazi, D. M. Kaufman, W. Li, and K. Singh. Magicalhands: Mid-air hand gestures for animating in vr. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 463–477. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347942 2

[4] R. Ban, K. Matsumoto, T. Narumi, and H. Kuzuoka. Wormholes in vr: Teleporting hands for flexible passive haptics. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 748–757, 2022. doi: 10.1109/ISMAR55827.2022.00093 1

[5] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Studies*, 4(3):114–123, may 2009. 9

[6] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. vol. 5394, pp. 73–84, 02 2009. doi: 10.1007/978-3-642-12553-9_7 2

[7] V. Biener, S. Kalamkar, N. Nouri, E. Ofek, M. Pahud, J. J. Dudley, J. Hu, P. O. Kristensson, M. Weerasinghe, K. u. Pucihar, M. Kljun, S. Streuber, and J. Grubert. Quantifying the effects of working in vr for one week. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3810–3820, 2022. doi: 10.1109/TVCG.2022.3203103 1

[8] V. Biener, D. Schneider, T. Gesslein, A. Otte, B. Kuth, P. O. Kristensson, E. Ofek, M. Pahud, and J. Grubert. Breaking the screen: Interaction across touchscreen boundaries in virtual reality for mobile knowledge workers. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3490–3502, 2020. doi: 10.1109/TVCG.2020.3023567 1

[9] J. Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995. 8

[10] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas. Construct Dynamic Graphs for Hand Gesture Recognition via Spatial-Temporal Attention. In *30th British Machine Vision Conference*, 2019. doi: 10.48550/ARXIV.1907.08871 2, 4

[11] E. Cippitelli, S. Gasparrini, E. Gambi, and S. Spinsante. A human activity recognition system using skeleton data from rgbd sensors. *Computational intelligence and neuroscience*, 2016, 2016. 2

[12] B. R. De Araùjo, G. Casiez, and J. A. Jorge. Mockup builder: Direct 3d modeling on and above the surface in a continuous interaction space. In *Proceedings of Graphics Interface 2012*, GI '12, p. 173–180. Canadian Information Processing Society, CAN, 2012. 2

[13] G. Devineau, F. Moutarde, W. Xi, and J. Yang. Deep learning for hand gesture recognition on skeletal data. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 106–113, 2018. doi: 10.1109/FG.2018.00025 2

[14] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[15] J. J. Dudley, H. Schuff, and P. O. Kristensson. Bare-handed 3d drawing in augmented reality. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 241–252, 2018. 2

[16] P. Feyereisen. Further investigation on the mnemonic effect of gestures: Their meaning matters. *European Journal of Cognitive Psychology*, 18:185–205, 03 2006. doi: 10.1080/09541440540000158 1

[17] F. Guimbretière, A. Martin, and T. Winograd. Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact.*, 12(3):460–476, sep 2005. doi: 10.1145/1096737.1096742 1

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks, 2017. doi: 10.48550/ARXIV.1706.04599 4

[19] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In P. A. Hancock

and N. Meshkati, eds., *Human Mental Workload*, vol. 52 of *Advances in Psychology*, pp. 139–183. North-Holland, 1988. doi: 10.1016/S0166-4115(08)62386-9 7

[20] D. Hayatpur, S. Heo, H. Xia, W. Stuerzlinger, and D. Wigdor. Plane, ray, and point: Enabling precise spatial manipulations with shape constraints. pp. 1185–1195, 10 2019. doi: 10.1145/3332165.3347916 2

[21] L. Hespanhol, M. Tomitsch, K. Grace, A. Collins, and J. Kay. Investigating intuitiveness and effectiveness of gestures for free spatial interaction with large displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, PerDis '12. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2307798.2307804 2

[22] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. 6

[23] S. Kang and B. Tversky. From hands to minds: Gestures promote understanding. *Cognitive Research: Principles and Implications*, 1:4, 12 2016. doi: 10.1186/s41235-016-0004-9 1

[24] M. Karam and M. Schraefel. A taxonomy of gestures in human computer interactions. Project report, 2005. 1

[25] P. O. Kristensson, T. Nicholson, and A. Quigley. Continuous recognition of one-handed and two-handed gestures using 3d full-body motion tracking sensors. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pp. 89–92, 2012. 2

[26] O. Köpüklü, A. Gunduz, N. Kose, and G. Rigoll. Real-time hand gesture detection and classification using convolutional neural networks. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pp. 1–8, 2019. doi: 10.1109/FG.2019.8756576 2

[27] H.-K. Lee and J. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, 1999. doi: 10.1109/34.799904 2

[28] B. Li, X. Li, Z. Zhang, and F. Wu. Spatio-temporal graph routing for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8561–8568, Jul. 2019. doi: 10.1609/aaai.v33i01.33018561 2

[29] Y. Li, J. Huang, F. Tian, H. Wang, and G. Dai. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019. doi: 10.3724/SP.J.2096-5796.2018.0006 2

[30] J. Liu, Y. Liu, Y. Wang, V. Prinet, S. Xiang, and C. Pan. Decoupled representation learning for skeleton-based gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[31] C. Madan and A. Singhal. Using actions to enhance memory: effects of enactment, gestures, and exercise on human memory. *Frontiers in Psychology*, 3, 2012. doi: 10.3389/fpsyg.2012.00507 1

[32] N. Marquardt, R. Jota, S. Greenberg, and J. A. Jorge. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, eds., *Human-Computer Interaction – INTERACT 2011*, pp. 461–476. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 2

[33] A. Masurovsky, P. Chojecki, D. Runde, M. Lafci, D. Przewozny, and M. Gaebler. Controller-free hand tracking for grab-and-place tasks in immersive virtual reality: Design elements and their empirical study. *Multimodal Technologies and Interaction*, 4(4), 2020. doi: 10.3390/mti4040091 2

[34] D. Mcneill. Hand and mind: What gestures reveal about thought. *University of Chicago Press*, 27, 06 1994. doi: 10.2307/1576015 1

[35] G. B. Mo, J. J. Dudley, and P. O. Kristensson. Gesture knitter: A hand gesture design tool for head-mounted mixed reality applications. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445766 2

[36] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 3

[37] E.-J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pp. 889–894, 2004. doi: 10.1109/AFGR.2004.1301646 2

[38] M. Oudah, A. Al-Naji, and J. Chahl. Hand gesture recognition based on computer vision: A review of techniques. *Journal of Imaging*, 6(8), 2020. doi: 10.3390/jimaging6080073 2

[39] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions*

on Knowledge and Data Engineering, 22(10):1345–1359, 2010. doi: 10. 1109/TKDE.2009.191 3

[40] C. Park, H. Cho, S. Park, Y.-S. Yoon, and S.-U. Jung. Handposemenu: Hand posture-based virtual menus for changing interaction mode in 3d space. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, ISS '19, p. 361–366. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3343055 .3360752 2

[41] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song. E-gesture: A collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. pp. 359–360, 11 2011. doi: 10. 1145/1999995.2000034 2

[42] S. Pei, A. Chen, J. Lee, and Y. Zhang. Hand interfaces: Using hands to imitate objects in ar/vr for expressive interactions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3491102.3501898 2

[43] S. Pook, E. Lecolinet, G. Vaysseix, and E. Barillot. Control menus: Excecution and control in a single interactor. pp. 263–264, 01 2000. doi: 10.1145/633292.633446 1

[44] J. Ruiz, Y. Li, and E. Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, p. 197–206. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/1978942.1978971 2

[45] S. Sapienza, P. M. Ros, D. A. F. Guzman, F. Rossi, R. Terracciano, E. Cordedda, and D. Demarchi. On-line event-driven hand gesture recognition based on surface electromyographic signals. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018. doi: 10.1109/ISCAS.2018.8351065 2

[46] M. Schmitz, S. Günther, D. Schön, and F. Müller. Squeezy-feely: Investigating lateral thumb-index pinching as an input modality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3491102.3501981 2

[47] D. Schneider, A. Otte, T. Gesslein, P. Gagel, B. Kuth, M. S. Damlakhi, O. Dietz, E. Ofek, M. Pahud, P. O. Kristensson, J. Müller, and J. Grubert. Reconviguration: Reconfiguring physical keyboards in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 25(11):3190–3201, 2019. doi: 10.1109/TVCG.2019.2932239 1

[48] E. Seol and G. Kim. Handytool: Object manipulation through metaphorical hand/fingers-to-tool mapping. In C. Stephanidis, ed., *HCI International 2019 - Posters - 21st International Conference, HCII 2019, Proceedings*, Communications in Computer and Information Science, pp. 432–439. Springer Verlag, 2019. doi: 10.1007/978-3-030-23528-4_58 2

[49] J. Shen, J. Dudley, and P. O. Kristensson. The imaginative generative adversarial network: Automatic data augmentation for dynamic skeleton-based hand gesture and human action recognition. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pp. 1–8. IEEE, 2021. 3

[50] J. Shen, J. Dudley, and P. O. Kristensson. Simulating realistic human motion trajectories of mid-air gesture typing. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 393–402. IEEE, 2021. 3

[51] J. Shen, J. Dudley, G. Mo, and P. O. Kristensson. Gesture Spotter: A rapid prototyping tool for key gesture spotting in virtual and augmented reality applications. *IEEE transactions on visualization and computer graphics*, PP, September 2022. doi: 10.1109/tvcg.2022.3203004 3

[52] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In *ACCV*, 2020. 3, 4

[53] P. Song, W. B. Goh, W. Hutama, C.-W. Fu, and X. Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, p. 1297–1306. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2207676.2208585 2

[54] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi: 10.1609/aaai.v31i1.11212 2

[55] Z. Song, J. J. Dudley, and P. O. Kristensson. Efficient special character entry on a virtual keyboard by hand gesture-based mode switching. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 864–871. IEEE, 2022. 2

[56] H. B. Surale, A. Gupta, M. Hancock, and D. Vogel. Tabletinvr: Exploring the design space for using a multi-touch tablet in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300243 2

[57] H. B. Surale, F. Matulic, and D. Vogel. Experimental Analysis of Barehand Mid-Air Mode-Switching Techniques in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–14. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605. 3300426 2

[58] T.-M. Tai, Y.-J. Jhang, Z.-W. Liao, K.-C. Teng, and W.-J. Hwang. Sensor-based continuous hand gesture recognition by long short-term memory. *IEEE Sensors Letters*, 2(3):1–4, 2018. doi: 10.1109/LSENS.2018. 2864963 2

[59] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE Transactions on Medical Imaging*, 36(1):86–97, 2017. doi: 10.1109/TMI.2016.2593957 3

[60] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov. Multi-task recurrent neural network for surgical gesture recognition and progress prediction. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1380–1386, 2020. doi: 10.1109/ICRA40945.2020.9197301 3

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[62] T. H. Vu, A. Misra, Q. Roy, K. C. T. Wei, and Y. Lee. Smartwatch-based early gesture detection 8 trajectory tracking for interactive gesture-driven applications. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), mar 2018. doi: 10.1145/3191771 2

[63] H. Wang and L. Wang. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[64] X. Xu, J. Gong, C. Brum, L. Liang, B. Suh, S. K. Gupta, Y. Agarwal, L. Lindsey, R. Kang, B. Shahsavari, T. Nguyen, H. Nieto, S. E. Hudson, C. Maalouf, J. S. Mousavi, and G. Laput. Enabling hand gesture customization on wrist-worn devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10. 1145/3491102.3501904 2, 3

[65] X. Xu, T. M. Hospedales, and S. Gong. Multi-task zero-shot action recognition with prioritised data augmentation. In B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., *Computer Vision – ECCV 2016*, pp. 343–359. Springer International Publishing, Cham, 2016. 3

[66] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 2

[67] Y. Yan, C. Yu, X. Ma, X. Yi, K. Sun, and Y. Shi. Virtualgrasp: Leveraging experience of interacting with physical objects to facilitate digital object retrieval. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173652 2

[68] M. Yasen and S. Jusoh. A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science*, 5:e218, 2019. 2

[69] C. Yuan, W. Hu, G. Tian, S. Yang, and H. Wang. Multi-task sparse learning with beta process prior for action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 423–429, 2013. doi: 10. 1109/CVPR.2013.61 3