GestOnHMD: Enabling Gesture-based Interaction on Low-cost VR Head-Mounted Display

Taizhou Chen,* Lantian Xu,† Xianshan Xu,‡ and Kening Zhu§

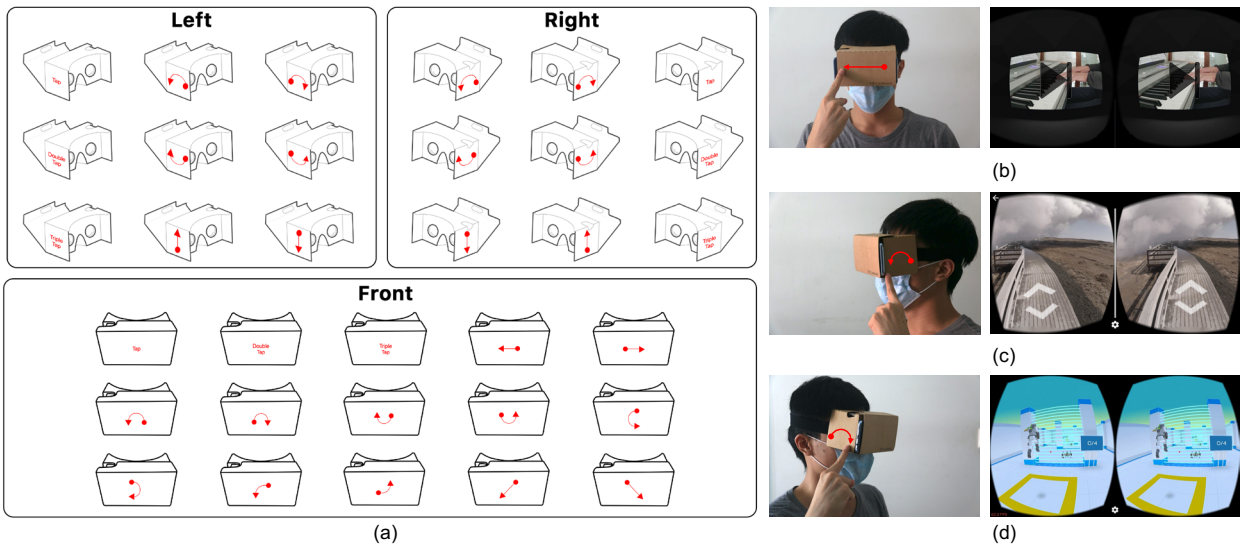

Fig. 1: (a) The GestOnHMD-enabled gesture sets for the left, the right, and the front surfaces respectively. (b) - (d) Application scenarios of GestOnHMD-enabled gesture interaction for mobile VR: (b) Next Video for Video Playback, (c) Move Forward for Street-view Navigation, (d) Jump for Mobile Gaming.

Abstract— Low-cost virtual-reality (VR) head-mounted displays (HMDs) with the integration of smartphones have brought the immersive VR to the masses, and increased the ubiquity of VR. However, these systems are often limited by their poor interactivity. In this paper, we present GestOnHMD, a gesture-based interaction technique and a gesture-classification pipeline that leverages the stereo microphones in a commodity smartphone to detect the tapping and the scratching gestures on the front, the left, and the right surfaces on a mobile VR headset. Taking the Google Cardboard as our focused headset, we first conducted a gesture-elicitation study to generate 150 user-defined gestures with 50 on each surface. We then selected 15, 9, and 9 gestures for the front, the left, and the right surfaces respectively based on user preferences and signal detectability. We constructed a data set containing the acoustic signals of 18 users performing these on-surface gestures, and trained the deep-learning classification pipeline for gesture detection and recognition. Lastly, with the real-time demonstration of GestOnHMD, we conducted a series of online participatory-design sessions to collect a set of user-defined gesture-referent mappings that could potentially benefit from GestOnHMD.

Index Terms—Virtual Reality, Smartphone, Mobile VR, Google Cardboard, Gesture

1 Introduction

In recent years, we are witnessing a great commercial success for immersive VR. Besides the high-end VR platforms (e.g., HTC Vive and Oculus), the smartphone-based VR HMDs have also become increasingly popular since the introduction of Google Cardboard [11] in 2014. With more than 3 billion smartphone users in the world, the mobile VR platforms have brought VR to the masses [54]. On the other hand, although a smartphone today could provide powerful computational and rendering capabilities, the interactivity of a low-cost mobile VR HMD is still limited. A common input method for these HMDs is to leverage the phone's built-in motion sensor and map the user's head rotation/orientation to the looking direction in VR. Besides, the first-generation Google Cardboard allows the user to slide an attached magnet on the side, which can be sensed by the in-phone magnetic sensor, to trigger an input event. In the second-generation Google Cardboard, a small lever button in contact with the phone's touchscreen is integrated to support the button-like input.

Researchers have proposed various solutions to enhance the interactivity of low-cost Cardboard-style VR HMDs, such as installing the touch panels and other interactive widgets on the HMD surfaces [13, 56], enabling voice-based interaction [12], enabling magnet-based gestures [27, 34, 50], tapping detection based on the built-in motion sensor's signal [63], mid-air tracking using the phone's rear camera [1, 23, 33, 62] and microphone [3], and eye-tracking using electrooculography sensors [47] and front camera [16]. While these solutions could enrich the interactivity of low-cost smartphone-based headsets, most of them require the installation of extra hardware (e.g., biometric sensors [47], touch panels [13], and earbuds [3]) or external passive accessories (e.g., magnets [27,

*taizhchen2-c@my.cityu.edu.hk
†lantianxu2-c@my.cityu.edu.hk
‡xianshaxu2-c@my.cityu.edu.hk
§Corresponding author: keninzhu@cityu.edu.hk

- *The authors are with School of Creative Media, City University of Hong Kong.*
- *Kening Zhu is also with City University of Hong Kong Shenzhen Research Institute, Shenzhen, P. R. China.*
- *Lantian Xu and Xianshan Xu are the co-second authors with the equivalent contributions to the paper.*

Manuscript received 9 Sept. 2020; revised 15 Dec. 2020; accepted 8 Jan. 2021. Date of publication 22 Mar. 2021; date of current version 7 Apr. 2021. Digital Object Identifier no. 10.1109/TVCG.2021.3067689

34,50], mirrors [1], and reflective markers [62]). While it was possible to detect on-HMD tapping based on the built-in motion-sensor data [63], its low sampling rate limited the capability of recognizing complex gestures. Voice-based interaction [12] may yield users' concern on privacy and social acceptance. Though mid-air gestures could be recognized by the rear camera, mid-air interaction may suffer from the fatigue problem due to the lack of physical anchor [20]. Besides the aforementioned techniques, the acoustic signal has been widely adopted for inferring human activities (please see Section 2.2). The sound induced by a surface gesture could be captured at a high sampling rate, without the need of extra external hardware. In this paper, we present GestOnHMD, a gesture-based interaction technique and a deep-learning-based pipeline that leverages the acoustic signal from the built-in stereo microphones in commodity smartphones for gesture recognition on the front, the left, and the right surfaces on the paper-based mobile VR headset. Here, we took Google Cardboard as the focus. With a three-step pipeline of deep-learning models, GestOnHMD classified the acoustic signal induced by the user's finger moving on the surface of Google Cardboard. We first conducted a gesture-elicitation study to generate 150 user-defined on-surface gestures, 50 for each surface. We then narrowed down the gesture sets to 15, 9, and 9 gestures for the front, the left, and the right surfaces respectively (Fig. 1) based on user preferences and signal detectability. We collected a data set containing the acoustic signals of 18 users performing these gestures (Data set available at: `https://github.com/taizhouchen/GestOnHMD`). We then trained a set of deep-learning classification models for gesture detection and recognition. According to the on-PC experiments, the GestOnHMD pipeline achieved an overall accuracy of 98.2% for both gesture detection and surface recognition, and 97.7% for gesture classification. We further conducted a series of online participatory design studies to generate the mapping between the GestOnHMD-enabled gestures and the commands in common mobile VR applications (e.g., web browsing, video play, gaming, online shopping, and so on).

The contributions of this paper are threefold:

- We present GestOnHMD, a gesture-based interaction technique for mobile VR using the built-in stereo microphones. We trained a deep-learning-based three-step gesture-recognition pipeline, and implemented a real-time prototype of GestOnHMD.
- We proposed a set of user-defined gestures on different surfaces of the Google Cardboard, with the consideration on user preferences and signal detectability.
- Through online participatory design sessions, we derived a set of gesture-referents mappings for a wide range of mobile VR applications.

2 Related Work

The presented work of GestOnHMD is largely inspired by the existing works on enriching the interactivity for mobile VR and audio-based gesture/activity recognition.

2.1 Enriching Interaction for Mobile VR

The common interaction method of mobile VR is using the head rotation sensing by the built-in motion sensors of a smartphone. Researchers proposed various ways of enhancing head-based interaction in mobile VR, such as tilting-based spatial navigation [54], head-movement-based gestures [15, 64], head-based text entry [66], and so on. To support the eye/gaze-based interaction in Google Cardboard, Shimizu and Chernyshov [47] embedded two Electrooculography sensors at the nose position of the Cardboard, to detect the eye-based gestures, such as blinking and up/down eye movement. Ahuja et al. developed EyeSpyVR [2], an eye-tracking technique using the phone's front-facing camera while being placed in the mobile VR headset. However, Qian and Teather [33] showed that head/eye-based interaction in VR may induce neck fatigue.

To support the hand-based interaction in mobile VR, Li et al. developed ScratchVR [27] which used an irregular circular track in the inner cardboard layer and provide rich haptic feedback while a user is moving the magnet. Al Zayer et al. developed PAWdio [3], a 1-degree-of-freedom (DOF) hand input technique that uses acoustic sensing to track the relative position of an earbud that the users hold in his/her hand from a VR headset. Using the back-facing camera on the phone, Ishii et al. developed FistPointer [23], detecting the gestures of thumb pointing and clicking. Similarly, Luo and Teather tracked the finger-pointing direction for target selection in mobile VR [33]. To further leverage the capability of the back camera on the phone, Ahuja et al. developed MeCap [1] by installing a pair of hemi-spherical mirrors in front of the mobile VR headset. It estimates the user's 3D body pose, hand pose, and facial expression in real time by processing the mirror-reflection image. Jan et al. developed FaceTouch [13] with a touch-sensitive surface on the front surface of the VR headset, to support multitouch in VR. Extending the concept of FaceTouch, Tseng et al. developed FaceWidgets [56] by integrated various types of electronic input components on the VR headset. For mobile VR settings, Takada et al. developed ExtensionClip [53] which used conductive materials to extend the phone's touch-sensitive area to the Cardboard.

Researchers also explored the interaction technique of using motion sensors. Yan et al. developed CardboardSense [63] which detects the user's finger tapping at different locations (i.e., left, right, top, and bottom) of the Cardboard device according to the built-in motion sensors' data. Similarly, on processing the data of the built-in motion sensors, Tregillus and Folmer developed VR-STEP [55] to support the walking-in-place locomotion in mobile VR. More recently, Gu et al. introduced Alohomora [12], a motion-based voice-command detection method for mobile VR. It detects the headset motion induced by the user's mouth movement while speaking the keywords.

Smartwatch could be another add-on component for enriching the interactivity of mobile VR. Nascimento et al. investigated the use of a smartwatch to detect eye-free touch gestures for text input in mobile VR [37]. Hirzle et al. proposed WatchVR [21] to support target selection in mobile VR using a smartwatch. Their user studies showed that the pointing gesture induced by the watch's motion significantly reduced the selection time and error rate. Wong et al. further proposed the application of bezel-initiated swipe on a circular watch for typing and menu selection in mobile VR [59].

While the existing works discussed above offered various valid options for enriching the interactivity in mobile VR, most of them require the installation of external hardware or accessories. Voice-based interaction with the built-in microphones may yield concerns on privacy and social acceptance [9], and mid-air gestures may cause fatigue [20]. On the other hand, there is little studys focusing on on-surface gesture interaction for mobile VR. In GestOnHMD, we investigated the user-defined gesture on the surfaces of Google Cardboard for mobile VR, and developed a three-step deep-learning pipeline for real-time acoustic-based gesture detection and recognition.

2.2 Acoustic-based Activity Recognition

Audio signal has been adopted by many researchers to capture the information of a user's activity and context. As an early work of sound-based gesture recognition, Harrison and Hudson proposed Scratch Input, an acoustic-based gestural input technique that relies on the unique sound profile produced by a fingernail being dragged over the textured surface [17]. The introduction of Scratch Input has inspired many following works on gesture recognition using the technique of passive acoustic sensing [6,8,18,30,32,42,46,60,68]. Most recently, Xu et al. developed EarBuddy, a real-time system leverages the microphone in commercial wireless earbuds to detect tapping and sliding gestures near the face and ears [61]. Their DenseNet-based deep-learning model can recognize 8 gestures based on the MFCC (Mel-frequency cepstral coefficients) profiles of the gesture-induced sounds with an average accuracy over 95%.

The acoustic signal can also infer the user's activity and context. Lu et al. presented SoundSense [31], using the classic machine-learning techniques to classify ambient sound, music, and speech with an overall accuracy above 90%. Stork et al. [52] processed the MFCC features with non-Markovian ensemble voting to recognize 22 human activities within bathrooms and kitchens. Yatani and Truong presented BodyScope [65], using the Support-Vector-Machine model to classify 12 human activities, and achieved an accuracy of 79.5%. Similarly, Rahman et al. developed BodyBeat [65], classifying 8 human activities with the Linear Discriminant Classifier. Savage et al. introduced Lamello [45], a set of 3D-printed tangible props that can generate unique acoustic profiles while the user moving the embedded passive parts. Laput et al. [25, 26] developed custom hardware to distinguish 38 environmental events by processing MFCCs with a pre-trained neural network.

GestOnHMD builds on the idea of gesture recognition based on passive acoustic sensing, with the focus on enabling on-surface gestures for mobile VR headsets. We designed a three-step pipeline of deep-learning neural networks to classify 33 gestures on the surfaces of Google Cardboard.

3 Study 1: On-Surface Gesture Design

Before implementing the gesture classifiers for GestOnHMD, we investigated the on-surface gestures that may be preferred by users for common mobile VR applications. Existing research showed that the user-defined gestures could improve the learnability and the usability for gestural user interfaces [44], and can infer users' general mental model towards a particular interaction contexts [58, 69]. There have been gesture-elicitation studies focusing on various types of human-computer interaction (e.g., surface computing [58], mobile interaction [44, 48, 69], and augmented reality [41], etc.), to generate the user-defined gestures. To this end, we conducted a gesture-elicitation study to derive a set of user-defined on-surface gestures for a mobile VR headset, here Google Cardboard, and to be classified in the later technical implementation.

3.1 Referents

In a gesture-elicitation study, a user is usually shown to a set of referents or effects of actions (e.g., the operations in text editing [69], multimedia browsing [28], gaming [57], etc.). The user will then define his/her desired gestures accordingly. In our study, we selected video playback and web browsing, due to their popularity in mobile VR [39]. Referring to the previous related research [28], we selected 10 referents (Table 1), covering both action and navigation [44], for each of these two applications.

Category	Sub-Category	Task Name
Action	Video Playback	Play/Pause
		Stop
		Mute/Unmute
		Add to Play List
	Web Browsing	New Tab
		Close Tab
		Open Link
		Add Bookmark
Navigation	Video Playback	Next Video
		Previous Video
		Volume Up
		Volume Down
		Forward
		Backward
	Web Browsing	Next Tab
		Previous Tab
		Next Page
		Previous Page
		Scroll Up
		Scroll Down

Table 1: List of referents presented to the participants

3.2 Participants

Twelve participants (4 females and 8 males) were recruited for this study. The average age was 24.6 years old (SD = 4.17). One was left-handed. Six were from the professions in engineering and science, four were from art and design, and two were from business and management. Ten participants mentioned that they have used Google Cardboard before, and the applications they used in Google Cardboard include video playback (6), web browsing (3), and game (1).

3.3 Apparatus

The participant was provided with a Google Cardboard headset without a smartphone integrated inside. The referents were displayed on a 33" LCD monitor in front of the participant with the animation playing the effects of actions.

3.4 Procedure

Upon the arrival of a participant, the facilitator introduced the study purpose and asked the participant to fill the pre-study questionnaire for his/her anonymous biographic information, and sign the consent form voluntarily. The facilitator then explained the flow of the experiment, introduced the two selected applications and the referent sets. The participant was asked to design the gestures for the referents to be performed on three surfaces (i.e., front, left, and right) of the Google Cardboard. This resulted in 2 applications $\times$ 3 surfaces = 6 conditions presented in the Latin-square-based counterbalanced order, leading to 6 design blocks for each participant. In each block, the participant was asked to design two gestures for each referent. The participant was told not to use the same gesture for different referents under the same application but allowed to reuse gestures across different surfaces and applications. The study for each participant took around one hour.

3.5 Selection of User-defined On-surface Gestures for Mobile VR

With 12 participants, 3 surfaces, 20 referents, and 2 design for each referent on each surface, a total of $12\times3\times20\times2$ = 1440 gestures were collected. We adopted the open-coding protocol to group these gestures according to their shapes so that each group held one identical representative gesture that was clustered across all the participants. This resulted in a set of 50 gestures for each surface as shown in Fig. 2.

With these user-defined on-surface gestures, we would like to further narrow them down to an optimal subset that can be easily learned, naturally performed, and reliably classified. Therefore, we conducted an online user-preference survey and a series of acoustic-analysis experiments to identify a subset of the most preferable gestures.

3.5.1 User Preference

We first conducted an online questionnaire survey on the user preference towards the 50 user-defined gestures on the three different Cardboard surface. This resulted in 50 gestures $\times$ 3 surfaces = 150 sets of questions. Each set of questions was presented with the gesture images in a random order in the online questionnaire. There were three items for each gesture, for a 7-point Likert-scale rating (1: strongly disagree to 7: strongly agree):

- *Ease to perform*: "It is easy to perform this gesture precisely."
- *Social acceptance*: "The gesture can be performed without social concern."
- *Fatigue*: "The gesture makes me tired."

Online Respondents. The questionnaire was published online and available to the public for one week. Through the word of mouth and the advertisement on the social network, we received in total the responses of 30 persons (14 males and 16 females). The average age was 27.5 years old (SD = 3.53). Three were left-handed. Twenty-two respondents stated that they have at least 6-month experience of using VR, while eight never used VR before.

Results. A multi-factorial repeated-measures ANOVA was performed on the ratings of ease to perform, social acceptance, and fatigue. The results showed a significant effect of the gesture type on the ratings of ease to perform ($F(49,1421) = 15.22$, $p < 0.005$, $\eta_p^2 = 0.344$), social acceptance ($F(49,1421) = 6.40$, $p < 0.005$, $\eta_p^2 = 0.181$), and fatigue ($F(49,1421) = 8.87$, $p < 0.005$, $\eta_p^2 = 0.234$), while there was no significant effect of the surface on these ratings. Therefore, we first averaged the three ratings on the three surfaces for each gesture. Fig. 3 shows the descriptive results of the average ratings for each gesture.

Following the previous practice of preference-based gesture selection [61], we then selected the gestures whose three ratings were all above 4. Therefore, This process removed 22 gestures with at least one rating below 4 were eliminated. We further considered the design consistency of the gestures. Previous research on gesture elicitation showed that users tended to include mirrored/reversible gestures in their gesture sets, especially for dichotomous referents [58]. To this end, we removed the mirrored and reversible gestures of those that are eliminated due to the low ratings, resulting in 22 gestures (highlighted with the green background in Fig. 2 and Fig. 3) remaining for each surface after this process.

3.5.2 Signal Detectability

After considering the user preference as the first factor, we examined the remaining gestures according to the signal detectability. We collected the acoustic signals of three persons (the co-authors) performing the 22 remaining gestures. Each gesture was performed within 1.5s for 20 repetitions on each surface. In addition, all three persons performed the gestures in the same lab environment where there was a constant background noise of the air conditioner and the fan. We recorded 10 acoustic signals of pure background noise, and used the average signal for signal-to-noise ratio (SNR) calculation. The average noise level was

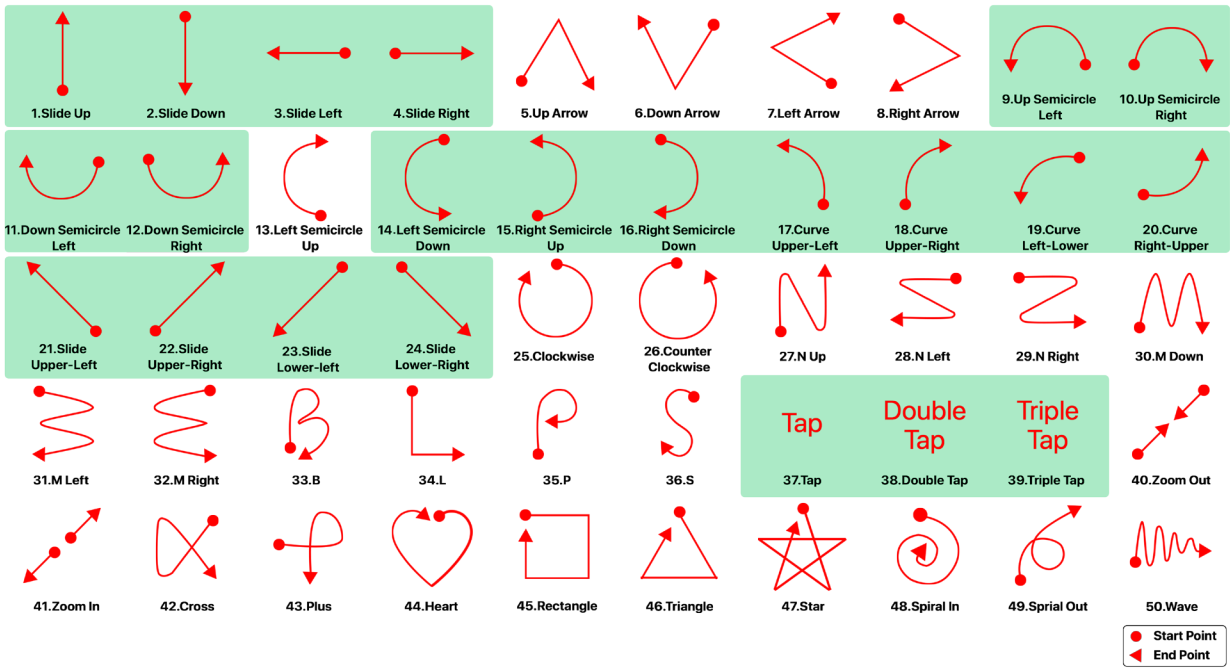

Fig. 2: User-defined on-surface gesture set for each surface. The green background highlights the remaining gestures after the user-preference-based filtering.

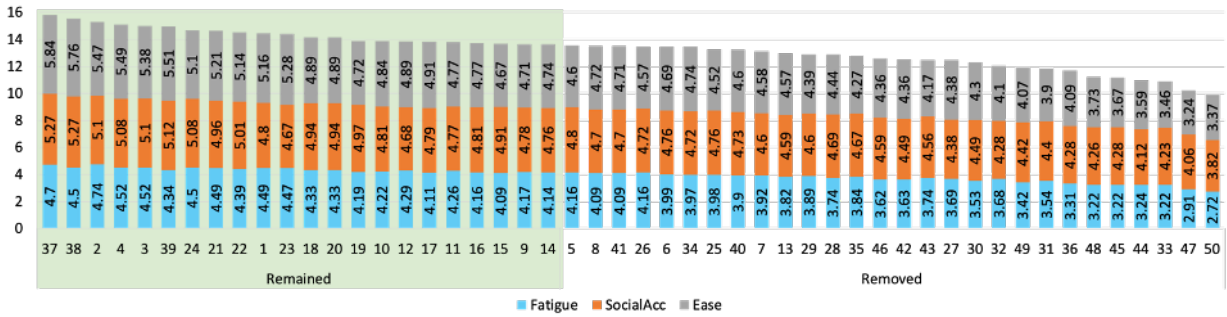

Fig. 3: Average ratings of the 50 gestures in terms of simplicity, social acceptability, and fatigue (reversed). The green background indicates the remaining gestures after the user-preference-based filtering.

around 50 db. This resulted in 3 persons $\times$ 22 gestures $\times$ 3 surfaces $\times$ 20 repetitions = 3960 acoustic signals, with 60 signals for each gesture on each surface. Please refer to the Additional File-2 for the sample audio clips and the sample visualizations for all the gestures in the final set.

Signal-to-Noise Ratio Analysis. We calculated the SNR based on the MFCC images for each sample on the front, the left, and the right surface respectively. We then removed the gestures with an average SNR lower than 5 dB which is a common criteria for signal detection [7, 43]. With the consideration of gesture mirroring, this process deleted no gesture for the front surface, 8 gestures for the left surface (i.e., gesture#3, #4, #15, #16, #17, #18, #20, #23 in Fig. 2), and 8 gestures for the right surface (i.e., gesture#3, #4, #14, #16, #18, #19, #22, #23 in Fig. 2). Noted that there was no gesture deleted for the front surface in this step. This could be due to the gestures generating an evenly distributed acoustic signal for the stereo microphones, leading to a considerable level of SNR. For the gestures on the left or right surface, the acoustic signal was likely to be biased to the channel on the corresponding side.

Signal Similarity Analysis. We used dynamic time warping (DTW) [5] on the average signal for each gesture, to calculate signal similarity between pairs of gestures within each surface. For each surface, we calculated the distance matrix where each cell was the DTW distance across all possible pairs of the corresponding gestures. We then summed each row to calculate the similarity between each gesture and all others within the same surface. Gestures with total distances lower than the 25th percentile were removed, as they are most likely to be confused in the classification [61]. Doing so with the additional consideration of gesture mirroring removed 7 gestures, 5 gestures, and 5 gestures for the front, the left, and the right surface respectively.

The procedure of gesture selection resulted in 15 gestures for the front surface, 9 gestures for the left surface, and 9 gestures for the right surface, as shown in Fig. 1.

4 Study 2: On-Surface Gesture Recognition

After finalizing the gesture set for each surface, we experimented with the feasibility of GestOnHMD, through 1) constructing a data set with a variety of instances of the selected gestures, and 2) training the machine-learning models for real-time gesture detection and recognition in the GestOnHMD pipeline.

4.1 Data Collection

In this section, we will present the process of constructing the data set for gesture detection and recognition in GestOnHMD.

4.1.1 Participants and Apparatus

We recruited 18 participants (6 female and 12 male) from a local university. The average age was 24.7 years old (SD = 2.54). Two of them were left-handed, while the rest were right-handed. The data-collection process was done in a sound studio where the average noise level is lower than 30dB. Participants were provided with a Google Cardboard headset (with the elastic head strap) with a Galaxy S9 integrated inside. We

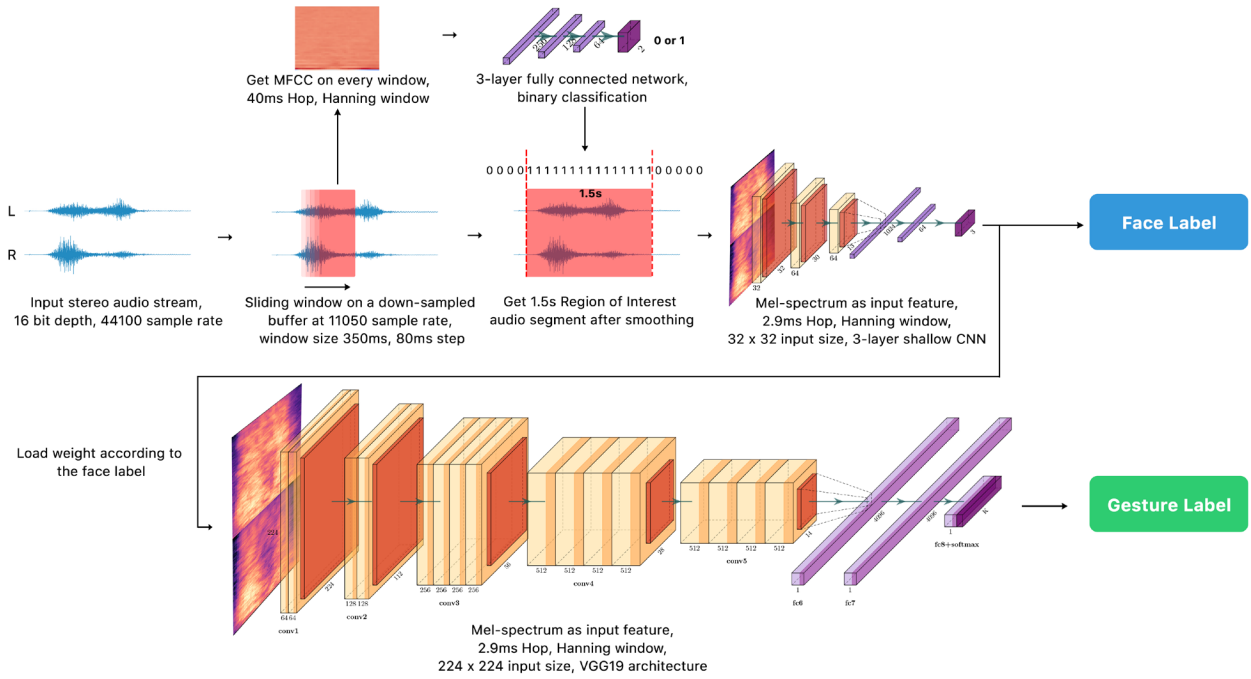

Fig. 4: System pipeline of GestOnHMD. Note that we trained 3 different VGG19 models for gesture classification for the front, the left, and the right surfaces. The corresponding weight parameters of the VGG19 network will be loaded according to the face label.

ran a customized mobile VR program on the smartphone for collecting acoustic data. The program was controlled by a Java server that runs on the facilitator's laptop through TCP/IP protocol. Each gesture was recorded as a 1.5-second-length stereo audio clip in 16 bit at the sample rate of 44100Hz. The participants sit in front of a 45-inch monitor which shows a slide of the to-record gesture's illustration and demo video.

4.1.2 Procedure

There were one facilitator and one participant in each session. The experiment facilitator first asked the participant to fill the pre-study questionnaire for his/her biographic information and sign the consent form voluntarily. The participant was then told to follow the gesture illustration and the demo video to perform the gestures on the cardboard surface. Before each recording, we first asked the participant to practice the gesture for several times until he/she was comfortable doing so. During the recording of each gesture, the participant first saw a 3-second count-down in the Cardboard, followed by a 1.5-second decreasing circular progress bar. The participant was asked to start performing gestures right after the count-down and finish before the progress bar ended. For each gesture, the participant needed to repeat 20 times. We also asked him/her to take off the headset and take it on again after every 10 times of recording, to increase the data variance. There was a mandatory 5-minutes break after the recording of all the gestures on each surface, and the participant can request for a short break at any time during the session. The order of the surfaces was counterbalanced across all the participants, while the gesture order within each surface was randomly shuffled for every participant. The experiment took about one and a half hours to complete. As a result, we collected 3239 valid audio clips, containing 413,694,118 audio frames in total for 33 gestures.

4.2 Gesture Detection and Classification

With the collected data set of acoustic gestural signal, we proposed GestOnHMD with a three-step deep-learning-based pipeline for on-surface gesture detection and recognition on Google Cardboard. As shown in Fig. 4, the GestOnHMD pipeline first detected whether the user is performing a gesture on the surface of the headset. Once a gesture is detected, the pipeline classified the surface where the gesture is being performed, and then classified what gesture is being performed using the gesture-classification model corresponding to the predicted surface.

4.2.1 Gesture Detection

The GestOnHMD pipeline first detects whether a gesture has been performed by the user on the surface. To simulate the real-time gesture detection, we adopted the sliding-window algorithm on the recorded audio clip. More specifically, we applied a 350ms sliding window with 80 steps on a down-sampled audio sample with a sample rate of 11050Hz. For each window, we extracted 20 MFCC features using a *Hanning* window in the hop size of 40ms. For each extracted MFCC feature, we calculated its mean and standard deviation to form a 40-dimensional feature vector and passed the feature vector a 3-layer fully connected network with a binary classifier for gesture detection. We randomly chose 3 users' acoustic data of gesture performing (Label: 1), along with the soundtracks of office and street noise (Label: 0) to train the classifier. The resulted gesture-detection classifier achieved an overall accuracy of 98.2%. During detection, we apply a smoothing algorithm as [61] did. More specifically, we treated adjacent sequences of continuous positive detection which lasted more than 1.5 seconds as a valid detection. To reduce the noisy shifting, we also tolerated if a long consecutive positive sequence was separated by one or two negative detections, and treated the whole sequence as a valid detection. As a result, we formed a 1.5s segment of audio signal with a sequence of positive detection as a candidate audio segment for future classification.

4.2.2 Surface Recognition

For each audio segment from the step of gesture detection, the pipeline performs the process of surface recognition before the gesture classification, to classify on which surface the gesture is performed. We converted the audio segment to a 32×32 mel-spectrogram image, before feeding into a shallow convolutional neural network (CNN) with 3 convolutional layers and one fully connected layer for surface classification. Since we recorded the acoustic signal in the stereo format, it may encode special features that could be useful for surface recognition. For example, if the gesture was performed on the right surface, the right channel will possibly contain more energy than the left channel, and vice versa. Thus, we extracted the mel-spectrograms for both the left and the right channels separately, and concatenated them vertically into one image, and lastly reshaped the image into 32×32. We trained the model of surface recognition using all data from 18 users by 8-2 train-test split. The overall accuracy is 98.2%.

4.2.3 Gesture Classification

The audio signal from the step of gesture detection was converted to the format of mel-spectrogram, and used for gesture classification by the pre-trained deep convolution neural network according to its surface label.

Data Augmentation

To increase the model's generalizability and avoid overfitting during training, we adopted the following data-augmentation schemes. Each of these schemes was independently applied to the input data during training with a probability of 0.5.

Noise Augmentation. As we collected the raw acoustic data of the gestures in a quiet studio, it is necessary to simulate the real-world scenario with various background noises. To this end, we randomly mixed the noisy signal from the soundtracks of two common scenarios, office noise [1] and street noise [2] to the raw audio data with a signal-to-noise mixing rate of 0.25 before converting them to the format of mel-spectrogram.

Time Warping. Although we set the recording duration as 1.5 seconds to cover all the selected gestures, the gesture performing speed across different users may vary, which may also lead to overfitting. To this end, we apply the strategy of time warping [40] to augment data along the time domain. More specifically, for each mel-spectrogram image with τ time step where the time axis is horizontal and the frequency axis is vertical, we picked a random point α from the time axis within the time step $(W, \tau - W)$. We then warped all points along the horizontal axis with a time step α to the left or right by a distance ω chosen from a uniform distribution from 0 to the time-warping parameter $W = 80$.

Frequency Mask. For each mel-spectrogram image, we also applied the technique of frequency masking [40] so that f consecutive mel-frequency channels $[f_0, f_0 + f)$ are masked by their average. f is chosen from a uniform distribution from 0 to the frequency-masking parameter $F = 27$, and f_0 is chosen from $[0, \nu - f)$ where ν is the number of mel-frequency channels. We generated two masks for each mel-spectrogram image.

Classification

Training. We treated the gesture-classification tasks on the three surfaces as three different classification problems. Through a pilot experiment using three users' data, we found that treating the recorded audio clip as a mono-channel acoustic signal could effectively increase the classification performance on both the left and the right surfaces, while treating the recorded audio clip as a multi-channel/stereo acoustic signal lead to a better performance for classifying the data on the front surface. Therefore, for the left and the right surfaces, we averaged the data of the left and the right channels for all the recorded audio clips before converting them to the 224×224 mel-spectrogram images. For the front surface, we generated two mel-spectrogram images for the data of the left and the right channels respectively, and concatenated them vertically, and lastly reshaped them into 224×224.

We trained three gesture-classification models, one for each of the three surfaces respectively, on a Desktop PC with one GTX 1080 Ti NVIDIA GPU, 32GB RAM, and one Intel i7-8700 CPU. We used an Adam optimiser [24] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with the learning rate of $1e-5$ for model optimization. The batch size was set to 16. During training, we applied the dropout technique [51] with the dropout rate of 0.5 to avoid overfitting.

Performance. For training the gesture-classification model for each surface, we first calculated the average signal-to-noise ratio for each user's data. We removed the data of three users whose SNR values are three lowest. We then shuffled the recorded acoustic data from the remaining 15 participants, and separated them into an 8-2 train-validation split. We experimented with the gesture classification on the three surfaces with six different structures of convolution neural networks as shown in Table 2, each being trained for 20 epoch. Table 2 shows their performance of gesture classification.

Our experiments showed that VGG19 achieved the highest accuracy of gesture classification across the three surfaces. The overall accuracy of VGG19 are 97.9%, 99.0%, and 96.4% for the left, the right, and the front surface respectively. Fig. 5 and Fig. 6 illustrate the confusion matrices for the gesture classification on the three surfaces with VGG19.

[1] https://www.youtube.com/watch?v=D7ZZp8XuUTE

[2] https://www.youtube.com/watch?v=8s5H76F3SIs

Model	Face	Accuracy	Precision	Recall
VGG16 [49]	F	0.9304	0.9409	0.9227
	R	0.9741	0.9740	0.9698
	L	0.9677	0.9717	0.9612
VGG19 [49]	F	**0.9639**	**0.9755**	**0.9579**
	R	**0.9905**	**0.9943**	**0.9905**
	L	**0.9792**	**0.9790**	**0.9735**
DenseNet169 [22]	F	0.7589	0.9043	0.5904
	R	0.7936	0.9182	0.6591
	L	0.7462	0.8727	0.4545
DenseNet201 [22]	F	0.8449	0.9218	0.7634
	R	0.9034	0.9272	0.8920
	L	0.9015	0.9407	0.8712
ResNet50 [19]	F	0.7366	0.8435	0.6317
	R	0.7083	0.8788	0.4943
	L	0.6307	0.7807	0.5057
ResNet101 [19]	F	0.6261	0.7992	0.4397
	L	0.6231	0.7746	0.4621
	R	0.6212	0.8087	0.4242

Table 2: The performance of GestOnHMD on different models. The accuracy, precision, and recall are weighted across all gestures.

As shown in Fig. 5, for the right and the left surface, three tapping-based gestures perform the best (averagely over 99.0% for both left and right), followed by four semicircle-based gestures (averagely 98.0% on the right surface, and 96.0% on the left surface). Two sliding-based gestures yielded the lowest accuracy (Right: 97.5%, Left: 95.0%). For the front surface, three tapping-based gestures achieved the highest accuracy of 100.0%. Six semicircle-based gestures and two left-right slide gestures yielded the same average accuracy of 97.0%, followed by two curved-based gestures (96.0%). Slide lower-left and slide lower-right yielded the lowest average accuracy (93.0%).

Leave-Three-User-Out Experiments. For the performance experiments of different classification models, we eliminated the user data of the top three lowest SNR, and used the data of the remaining 15 users for training and testing. However, the eliminated data of the three users may represent a specific range of on-surface gesture patterns. To investigate the generalizability of the trained gesture-classification model, we tested it using the data of the left-out users. This revealed an overall accuracy of 76.3%, 87.4%, and 93.7% for the front, the left, and the right surface respectively. There was an average drop of 12.0% from the within-user test, with a large drop around 20% for the gestures on the front surface.

In a real-world scenario, many applications often ask a new user to perform and practice each gesture for a few times before the actual usage. The recorded gestures can be used for transfer learning on a pre-trained model. To this end, we experimented with the transfer-learning process on the trained VGG19 models with a small amount of data from the three left-out users. Fig. 7 shows how the amount of training data included from the left-out users could improve the gesture-classification performance on the three surfaces. With a minimum amount of five samples for each gesture from each user, the overall performance improved to 96.7% averagely.

Real-time Performance. To evaluate the real-time efficiency of GestOnHMD, we implemented the three-step pipeline using Python 3.5 with TensorFlow 2.2 framework on a desktop PC with the same specification of the computer used for model training. The pipeline received the real-time audio stream from the smartphone through a TCP/IP protocol. The total inference time for the three-step pipeline was 1.96s, indicating an acceptable response speed [36] of GestOnHMD.

5 Study 3: Investigating the Mappings between On-Surface Gestures and Mobile VR Applications

With the considerable performance of gesture recognition in GestOnHMD, we further investigated how the gesture set enabled by GestOnHMD could be used for mobile VR applications. We conducted a series of online participatory sessions, by inviting mobile VR users to create the mapping between the GestOnHMD-enabled gestures and the operations in mobile VR.

(a) Left (b) Right

Fig. 5: Confusion matrices of the gesture classification on the left and the right surfaces.

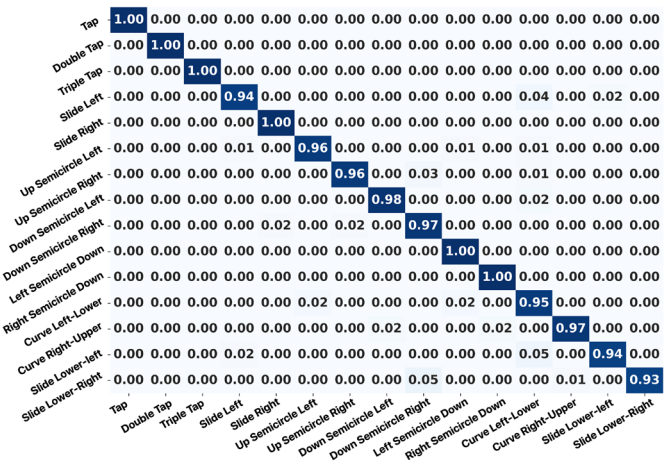

Fig. 6: Confusion matrix of the gesture classification on the front surface.

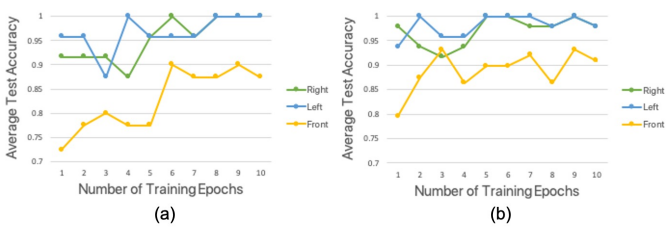

Fig. 7: Gesture-classification performance after transfer learning with the three left-out users' data: (a) 5 training samples and (b) 10 training samples for each gesture from each left-out user.

5.1 Participants

Through the word of mouth and the advertisement on social network, we recruited 19 participants (6 females and 13 males). The average age was 25.2 years old (SD = 3.52). Two were left-handed. Nine stated that they have at least 6-month experience of using VR, and five stated they have <6-month experience of using VR, while six never used VR before. All the participants were in the same region as the authors.

5.2 Study Setup & Apparatus

The participatory design sessions were conducted in the format of online video conferencing using Zoom. Before the scheduled session, a set of Google Cardboard with the elastic head strap was sent to the participant through the local post. The participant was told to use the Google Cardboard during the design session. On the side of the experiment facilitator, the prototype of GestOnHMD was set up for demonstration.

We used the 20 referents from the two mobile VR applications used in Study 1 (i.e. video playback and web browsing). For each referent, the participant was asked to assign one GestOnHMD-enabled gesture that he/she felt the most suitable. The gesture cannot be reused for different referents within the same application, but can be reused across different applications. The applications were presented in a Latin-squared-based counterbalanced order across all the participants, and the order of the referents under the same application was randomized.

5.3 Procedure

There were one facilitator and one participant in each session. The facilitator first guided the participant to fill the pre-study questionnaire for his/her anonymous biographic information, and sign the consent form voluntarily. The facilitator then presented the flow of the study, and introduced the think-aloud protocol to encourage the participant to verbally describe his/her thinking process. In addition, the facilitator instructed the participant to try Google Cardboard by installing the common applications, such as Youtube and VR web browser, on the participant's phone. The facilitator then demonstrated GestOnHMD by randomly selecting three gestures from each surface for demonstration. After the demonstration, the participant started the process of gesture-referent mapping. There were two design blocks, one for each application, in each participant session. In each block, the facilitator shared the screen of a mapping questionnaire. The participant verbally described the mappings, and the facilitator dragged and dropped the gestures image to the corresponding referents for confirmation. The participant can modify the mappings freely until he/she was satisfied. After the two design blocks, the facilitator instructed the participant to propose at least three pairs of mappings between the GestOnHMD-enabled gestures and the referents from other VR applications. The whole session took around 1 hour, and was video recorded with the prior consent of the participant.

5.4 Results

We collected in total of 360 pairs of gesture-referent mappings from all the participants. For each GestOnHMD gesture, we calculated its number of being used in each referent (i.e. appearance frequency). As one gesture could be mapped to different referents in the same application, for each gesture, we selected the referent for which the gesture achieved the highest appearance frequency as the first step of deriving the final mappings. However, the conflict may still exist as multiple gestures

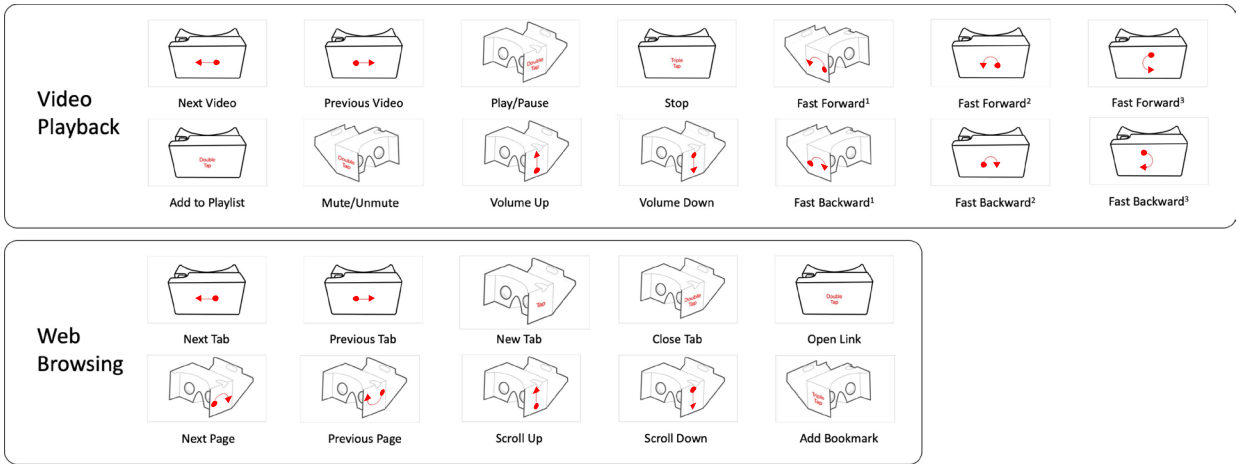

Fig. 8: The gesture-referent mapping generated in Study 3 for video play and web browsing in mobile VR.

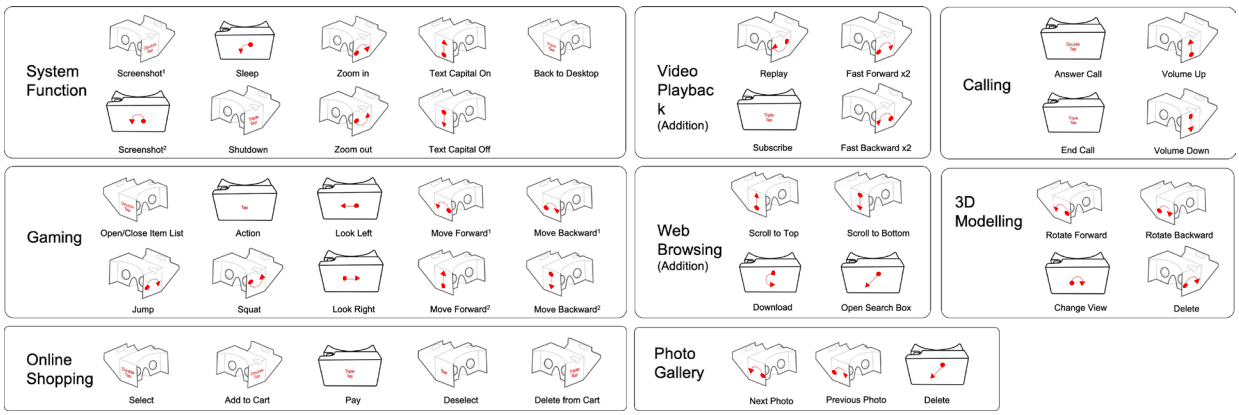

Fig. 9: The gesture-referent mapping generated for other referents.

could be mapped to one referent by different participants. To resolve this, the gesture with the highest appearance frequency within one referent won. Lastly, it is possible that multiple gestures achieved the same appearance frequency which was the highest within one referent. In this case, we kept all the gestures. We also considered the design consistency of gesture mirroring for dichotomous referents for mapping selection.

Our resulting gesture-referent map (Fig. 8) is conflict-free and covers 59.4% of all the gestures proposed and enabled by GestOnHMD. The gestures on the front and the right surfaces were more frequently used than those on the left surface. This could be because most of the participants were right-handed, with only two left-handed. Noted that there are three pairs of candidature gestures for fast forward/backward. We considered all of them as reasonable mappings, with the same appearance frequency.

Beside the gesture-referent mappings for the two selected mobile VR applications, we collected in total 74 pairs of gesture-referent mappings for other referents in video playback (8), web browsing (5), and other applications (e.g., system functions: 11, calling: 4, photo gallery: 5, 3D modeling: 11, online shopping: 20, and gaming: 11). For these sets of gesture-referent mappings, we adopted the same conflict-resolving solution as we did for the 20 selected referents above. We also considered the mappings proposed for system functions higher priority, to solve the conflict between system functions and applications. This resulted in the mappings shown in Fig. 9, covering 87.9% of the proposed gesture set enabled by GestOnHMD.

6 Discussion

The above results showed that the GestOnHMD pipeline could recognize 33 on-surface gestures for Google Cardboard, and support a wide range of applications. In this section, we discuss the insights and observations on signal quality, robustness improvement, user-specific optimization, gesture design, device generalization, and possible future improvement.

6.1 Signal Quality

During the gesture-selection process, the SNR level for each gesture was calculated using the recorded background noise which can be considered as a moderate level, similar to the environments with light traffic [10]. For the model training, the acoustic gesture signals recorded in the quiet sound studio were mixed with the noise from an office environment. The SNR levels for the gestures may vary in other common noisy environments, and this may affect the model performance of gesture detection and recognition. During the experiments, we also observed that the SNR level may vary when users perform the on-surface gestures with different parts of their fingers. Generally speaking, the fingernail could generate the highest SNR level, while the finger pad tends to result in a softer acoustic signal. When collecting our current data set, we encouraged the participants to use their fingernails as much as possible, to ensure the strengths of the signals.

The system's robustness towards noise can be improved with a larger data sets covering a wider range of background noise. In addition, various noise-reducing approaches have been proposed for speech recognition [38] and sound-based activity recognition [29]. It is worth to investigate the feasibility of these approaches being adopted in GestOnHMD. On the other hand, there have been commercial products of background-noise-canceling microphones integrated into the headsets in the market [4]. It is reasonable to envision that such hardware can be minimized in shape and integrated into the smartphone in the future, which could potentially improve the quality of the acoustic-based mobile interaction.

6.2 Error Handling

One potential issue with any prediction technique is addressing or correcting the errors. One type of error that may likely occur is the ambient sound generating as the false positives for gesture detection. Our empirical experiments showed that GestOnHMD can perform

robustly against the sound of random hand actions (e.g., clapping and snapping, see the Additional File-1 (i.e. the supplementary video) 00:55 - 01:02 for reference). However, more in-depth investigation on noise reducing is needed as discussed in the previous subsection.

Another potential solution for error handling could be introducing explicit correction/confirmation operation. For instance, with the presentation of the predicted gesture, the system can prompt an interface for the user to confirm or reject the prediction. One example for confirmation/rejection is through head nodding/shaking which could be detected based on the motion-sensor data [15,64]. Yet, a high accuracy of gesture classification could reduce the need for explicit error correction.

6.3 Lower Accuracy on the Front Surface

We observed a lower gesture-classification accuracy in general for the front surface (96.4%) than the other two surfaces (Left: 97.9%; Right: 99.0%). One possible reason is that there were more classes of gestures on the front surface. In addition, four participants in the data-collection study commented that it was less smooth/comfortable to perform the gestures on the front surface, as they needed to twist their wrists and arms inward. This may affect the shapes of the gestures performed, and further influence the acoustic signals. However, our online user-preference survey didn't reflect on this issue, with no significant difference among the three surfaces in terms of the ease to perform. This could be because the user-preference survey was conducted online. With the online survey, we intended to reduce the face-to-face contact under the situation of global health incident. Though the questionnaire description encouraged the respondents to try performing the gestures before rating, it could be possible that the respondents didn't have sufficient hands-on experience on the gestures, which may affect their ratings on the ease of performance.

6.4 On-Surface Gesture Design for Mobile VR Headset

We derived a few insights from the user-preference survey. In general, users preferred simple gestures (e.g., tapping and short sliding). Across the three surfaces, tap, double taps, and triple taps were the top three rated in terms of ease to perform and social acceptance. The gesture of sliding down yielded the lowest rating of fatigue, as it was the lower the better for fatigue. Tap and double taps were rated within the top 5 for low fatigue.

Looking at the gestures being removed due to low user preference, most of them involve either >1 directions (e.g., arrows), long-distance (e.g., circles), or complex shapes (e.g., star, wave, X, $+$, etc.). In previous work on user-defined gesture for surface computing [58] and mobile devices [44, 69], users tended to the gestures with shape drawing on the surface or in the mid-air, such as drawing circles, letters, and symbols. This was different from our observation in the user-defined gesture for GestOnHMD. One possible reason is that users performed the gestures on the Cardboard surfaces in an eyes-free manner, which may affect their confidence in correctly performing the gestures. Therefore, simple gestures were more preferred.

For the signal strength, it was observed that tapping yielded louder sounds than sliding did, as tapping is usually quick and short. Gestures on the front surface generated stronger acoustic signals than those on the two sides. This phenomenon could be mainly due to the distance from the gesture surface to the stereo microphones. As discussed above, the participants may find it less smooth to perform the gestures on the front surface, so we see a trade-off between the signal strength and the actual ease to perform. To this end, both user preferences and signal strength should be taken into account for on-surface gesture design.

6.5 Limitations and Future Work

During the studies, we identified a few limitations of our work for future improvement. Firstly, due to the tightened policy of face-to-face meetings in the latter part of our studies, it was challenging to recruit participants for the on-site usability study. As the alternative approach to evaluate how GestOnHMD may facilitate mobile VR interaction, we conducted the online participatory design sessions, with the participants proposing the gesture-referent mappings. The resulted mappings showed that the gesture sets enabled by GestOnHMD could provide a sufficient number of gesture options for mobile VR interaction. Previous research showed that gesture-based mobile interaction outperformed graphical-user-interface-based (GUI-based) interaction by yielding shorter task-completion time and lower workload [61]. We hypothesize that GestOnHMD may yield a similar better performance over GUI-based interaction for mobile VR, and plan to conduct a thorough usability experiment in the near future.

Secondly, we only experimented with the acoustic gestural signals which were collected on the surfaces of Google Cardboard (2nd Generation). The surfaces of Google Cardboard are usually rough and thick, which may enhance the signal quality. As the design of Google Cardboard is open source, there are many design variations of paper-based mobile VR headsets in the market. Some are with glossy surfaces, and some are with tactile patterns. These may result in softer acoustic gestural signals. As one important future work to improve the technique generalization, we will collect, analyze, and classify the acoustic gestural signals from different processed surfaces for GestOnHMD.

Thirdly, compared to the voice-based commands, GestOnHMD also processed the audio signal from the phone's built-in microphone. As the voice commands could be eavesdropped by the devices close by and lead to the privacy concerns [9], it could be possible for the acoustic signal of an on-HMD swiping/tapping gesture to be inferred/spied by other devices which are not in the HMD. However, the gestural acoustic profile captured by the phone in the HMD could be different from those captured by the devices in other locations, making it potentially difficult to directly use the GestOnHMD classification pipeline for eavesdropping on the side. To this end, we will experiment and improve GestOnHMD's capability of anti-eavesdropping in the future.

Last but not the least, the current prototype of GestOnHMD was run on a desktop PC as a proof of concept. While the current accuracy of on-PC gesture classification was above 95% with a reasonable inference time, the performance could be negatively affected while directly running the deep-learning models on the phone due to the model complexity and the computational constraint on the smartphone. While this issue could be solved with the future hardware advancing in the smartphone, one potential solution that can be feasible in a near future is leveraging the advantage of modern high-speed mobile networks (e.g., 5G cellular network) [35, 67]. As suggested by Guo, the modern deep-learning-based mobile applications could benefit from the hybrid approach of combining the on-device and the cloud-based classification [14]. More specifically for GestOnHMD, we could run the light-weight gesture-detection process in real-time locally on the phone, since this process requires less computational resources than the other two following processes. The acoustic signal could be simultaneously sent to the cloud server for the face and the gesture recognition through the cellular network (e.g., 5G) in low latency. In addition, we will develop and open-source the GestOnHMD package and resource which will be compatible with the popular VR development platforms (e.g., Unity and Unreal engines), to contribute to the design and development community for mobile VR.

7 Conclusion

In this paper, we propose GestOnHMD, a deep-learning-based gesture-recognition framework to support gestural interaction for mobile VR headsets. The gesture-elicitation studies resulted in a set of in total of 150 user-defined gestures for the front, the left, and the right surfaces for the Google-Cardboard headset. We further narrowed the gesture sets down with the consideration of user preference and signal detectability, resulting in 15 gestures, 9 gestures, and 9 gestures for the front, the left, and the right surfaces respectively. We then collected the acoustic signals of 18 users performing the selected gestures on the surfaces of Google Cardboard, and trained a three-step framework of deep neural networks for gesture detection and recognition. The on-PC experiments showed that the GestOnHMD framework achieved an overall accuracy of 98.2% for both gesture detection and surface recognition, and 97.7% for gesture classification. In addition, our online participatory design studies showed that the GestOnHMD-enabled gesture sets provide a sufficient number of design options for a wide range of mobile VR applications. With GestOnHMD, we demonstrate the feasibility of enriching the interactivity for mobile VR with surface-gesture-based interaction, and we hope to create a new interaction paradigm for mobile VR.

Acknowledgments

This research was partially supported by the Young Scientists Scheme of the National Natural Science Foundation of China (Project No. 61907037), the Applied Research Grant (Project No. 9667189), and the Centre for Applied Computing and Interactive Media (ACIM) in School of Creative Media, City University of Hong Kong.

REFERENCES

[1] K. Ahuja, C. Harrison, M. Goel, and R. Xiao. Mecap: Whole-body digitization for low-cost vr/ar headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 453–462. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347889

[2] K. Ahuja, R. Islam, V. Parashar, K. Dey, C. Harrison, and M. Goel. EyeSpyVR: Interactive Eye Sensing Using Off-the-Shelf, Smartphone-Based VR Headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):1–10, 2018. doi: 10.1145/3214260

[3] M. Al Zayer, S. Tregillus, and E. Folmer. PAWdio: Hand input for mobile VR using acoustic sensing. In *CHI PLAY 2016 - Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pp. 154–158, 2016. doi: 10.1145/2967934.2968079

[4] ASUS. Ai noise cancelling microphone., 2020.

[5] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, vol. 10, pp. 359–370. Seattle, WA, USA:, 1994.

[6] A. Braun, S. Krepp, and A. Kuijper. In *ACM International Conference Proceeding Series*. doi: 10.1145/2790044.2790052

[7] J. T. Bushberg and J. M. Boone. *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2011.

[8] M. Chen, P. Yang, J. Xiong, M. Zhang, Y. Lee, C. Xiang, and C. Tian. Your Table Can Be an Input Panel. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):1–21, 2019. doi: 10.1145/3314390

[9] A. Easwara Moorthy and K.-P. L. Vu. Privacy concerns for use of voice activated personal assistant in the public space. *International Journal of Human-Computer Interaction*, 31(4):307–335, 2015.

[10] C. for Disease Control and Prevention. What noises cause hearing loss?, 2019.

[11] Google. Google cardboard, 2014.

[12] J. Gu, Z. Yu, and K. Shen. Alohomora: Motion-Based Hotword Detection in Head-Mounted Displays. *IEEE Internet of Things Journal*, 7(1):611–620, 2020. doi: 10.1109/JIOT.2019.2946593

[13] J. Gugenheimer, D. Dobbelstein, C. Winkler, G. Haas, and E. Rukzio. FaceTouch: Touch interaction for mobile virtual reality. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 07-12-May-, pp. 3679–3682, 2016. doi: 10.1145/2851581.2890242

[14] T. Guo. Cloud-based or on-device: An empirical study of mobile deep inference. In *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 184–190. IEEE, 2018.

[15] T. Hachaj and M. Piekarczyk. Evaluation of pattern recognition methods for head gesture-based interface of a virtual reality helmet equipped with a single IMU sensor. *Sensors (Switzerland)*, 19(24):1–19, 2019. doi: 10.3390/s19245408

[16] H. Hakoda, W. Yamada, and H. Manabe. Eye tracking using built-in camera for smartphone-based HMD. In *UIST 2017 Adjunct - Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 15–16, 2017. doi: 10.1145/3131785.3131809

[17] C. Harrison and S. E. Hudson. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In *UIST 2008 - Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, pp. 205–208, 2008. doi: 10.1145/1449715.1449747

[18] C. Harrison, J. Schwarz, and S. E. Hudson. TapSense: Enhancing finger interaction on touch surfaces. In *UIST'11 - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pp. 627–634, 2011. doi: 10.1145/2047196.2047279

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[20] J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1063–1072, 2014.

[21] T. Hirzle, J. Gugenheimer, J. Rixen, and E. Rukzio. Watchvr: Exploring the usage of a smartwatch for interaction in mobile virtual reality. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2018-April, pp. 1–6, 2018. doi: 10.1145/3170427.3188629

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[23] A. Ishii, T. Adachi, K. Shima, S. Nakamae, B. Shizuki, and S. Takahashi. FistPointer: Target Selection Technique Using Mid-air Interaction for Mobile VR Environment. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, p. 474, 2017. doi: 10.1145/3027063.3049795

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[25] G. Laput, K. Ahuja, M. Goel, C. Harrison, and F. Ave. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*, pp. 213–224, 2018. doi: 10.1145/3242587.3242609

[26] G. Laput, Y. Zhang, and C. Harrison. Synthetic sensors: Towards general-purpose sensing. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2017-May, pp. 3986–3999, 2017. doi: 10.1145/3025453.3025773

[27] R. Li, V. Chen, G. Reyes, and T. Starner. ScratchVR: Low-cost, calibration-free sensing for tactile input on mobile virtual reality enclosures. In *Proceedings - International Symposium on Wearable Computers, ISWC*, pp. 176–179, 2018. doi: 10.1145/3267242.3267260

[28] W. H. A. Li, K. Zhu, and H. Fu. Exploring the design space of bezel-initiated gestures for mobile interaction. *International Journal of Mobile Human Computer Interaction (IJMHCI)*, 9(1):16–29, 2017.

[29] K. Lopatka, J. Kotus, and A. Czyzewski. Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations. *Multimedia Tools and Applications*, 75(17):10407–10439, 2016.

[30] P. Lopes, R. Jota, and J. A. Jorge. Augmenting touch interaction through acoustic sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS'11*, pp. 53–56, 2011. doi: 10.1145/2076354.2076364

[31] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. SoundSense: Scalable sound sensing for people-centric applications on mobile phones. In *MobiSys'09 - Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services*, pp. 165–178, 2009. doi: 10.1145/1555816.1555834

[32] G. Luo, P. Yang, M. Chen, and P. Li. HCI on the Table: Robust Gesture Recognition Using Acoustic Sensing in Your Hand. *IEEE Access*, 8:31481–31498, 2020. doi: 10.1109/ACCESS.2020.2973305

[33] S. Luo and R. J. Teather. Camera-based selection with cardboard HMDs. In *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, pp. 1066–1067, 2019. doi: 10.1109/VR.2019.8797772

[34] K. Lyons. 2D input for virtual reality enclosures with magnetic field sensing. In *International Symposium on Wearable Computers, Digest of Papers*, vol. 12-16-Sept, pp. 176–183, 2016. doi: 10.1145/2971763.2971787

[35] M. McClellan, C. Cervelló-Pastor, and S. Sallent. Deep learning at the mobile edge: Opportunities for 5g networks. *Applied Sciences*, 10(14):4735, 2020.

[36] R. B. Miller. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 267–277, 1968.

[37] T. H. Nascimento, F. A. A. De Melo Nunes Soares, D. V. Oliveira, R. L. Salvini, R. M. Da Costa, and C. Goncalves. Method for text input with google cardboard: an approach using smartwatches and continuous gesture recognition. In *Proceedings - 19th Symposium on Virtual and Augmented Reality, SVR 2017*, vol. 2017-Novem, pp. 223–226, 2017. doi: 10.1109/SVR.2017.36

[38] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143–19165, 2019.

[39] S. Neelakantam and T. Pant. *Learning web-based virtual reality: build and deploy web-based virtual reality technology*. Apress, 2017.

[40] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

[41] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn. User-Defined Gestures for Augmented Reality. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2013-April, pp. 955–960, 2013. doi: 10.1145/2468356.2468527

[42] S. Robinson, N. Rajput, M. Jones, A. Jain, S. Sahay, and A. Nanavati. TapBack: Towards richer mobile interfaces in impoverished contexts. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 2733–2736, 2011. doi: 10.1145/1978942.1979345

[43] A. Rose. *Vision: human and electronic*. Springer Science & Business Media, 2013.

[44] J. Ruiz, Y. Li, and E. Lank. User-defined motion gestures for mobile interaction. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 197–206, 2011. doi: 10.1145/1978942.1978971

[45] V. Savage, A. Heady, B. Hartmanny, D. B. Goldman, G. Mysore, and W. Li. Lamello: Passive acoustic sensing for tangible input components. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2015-April, pp. 1277–1280, 2015. doi: 10.1145/2702123.2702207

[46] L. Shi, M. Ashoori, Y. Zhang, and S. Azenkot. Knock knock, What's there: Converting passive objects into customizable smart controllers. In *MobileHCI 2018 - Beyond Mobile: The Next 20 Years - 20th International Conference on Human-Computer Interaction with Mobile Devices and Services, Conference Proceedings*, 2018. doi: 10.1145/3229434.3229453

[47] J. Shimizu and G. Chernyshov. Eye movement interactions in google cardboard using a low cost EOG setup. In *UbiComp 2016 Adjunct - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1773–1776, 2016. doi: 10.1145/2968219.2968274

[48] S. S. A. Shimon, C. Lutton, Z. Xu, S. Morrison-Smith, C. Boucher, and J. Ruiz. Exploring non-touchscreen gestures for smartwatches. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 3822–3833, 2016. doi: 10.1145/2858036.2858385

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[50] B. Smus and C. Riederer. Magnetic input for mobile virtual reality. In *ISWC 2015 - Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp. 43–44, 2015. doi: 10.1145/2802083.2808395

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[52] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras. Audio-based human activity recognition using Non-Markovian Ensemble Voting. In *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, pp. 509–514, 2012. doi: 10.1109/ROMAN.2012.6343802

[53] R. Takada, H. Manabe, T. Isomoto, B. Shizuki, and W. Yamada. ExtensionClip: Touch point transfer device linking both sides of a smartphone for mobile VR environments. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2018-April, pp. 1–6, 2018. doi: 10.1145/3170427.3188644

[54] S. Tregillus, M. Al Zayer, and E. Folmer. Handsfree omnidirectional VR navigation using head tilt. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2017-May, pp. 4063–4068, 2017. doi: 10.1145/3025453.3025521

[55] S. Tregillus and E. Folmer. VR-STEP: Walking-in-place using inertial sensing for hands free navigation in mobile VR environments. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1250–1255, 2016. doi: 10.1145/2858036.2858084

[56] W. J. Tseng, L. Y. Wang, and L. Chan. FaceWidgets: Exploring tangible interaction on face with head-mounted displays. In *UIST 2019 - Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 417–427, 2019. doi: 10.1145/3332165.3347946

[57] Y.-C. Tung, C.-Y. Hsu, H.-Y. Wang, S. Chyou, J.-W. Lin, P.-J. Wu, A. Valstar, and M. Y. Chen. User-defined game input for smart glasses in public space. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3327–3336, 2015.

[58] J. O. Wobbrock, M. R. Morris, and A. D. Wilson. User-defined gestures for surface computing. *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1083–1092, 2009. doi: 10.1145/1518701.1518866

[59] P. C. Wong, K. Zhu, X.-D. Yang, and H. Fu. Exploring eyes-free bezel-initiated swipe on round smartwatches. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–11. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376393

[60] R. Xiao, G. Lew, J. Marsanico, D. Hariharan, S. E. Hudson, and C. Harrison. Toffee: Enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. In *MobileHCI 2014 - Proceedings of the 16th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 67–76, 2014. doi: 10.1145/2628363.2628383

[61] X. Xu, H. Shi, X. Yi, W. Liu, Y. Yan, Y. Shi, A. Mariakakis, J. Mankoff, and A. K. Dey. Earbuddy: Enabling on-face interaction via wireless earbuds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2020.

[62] D. Yamaji, H. Hakoda, W. Yamada, and H. Manabe. A low-cost tracking technique using retro-reflective marker for smartphone based HMD. In *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2018-April, pp. 3–8, 2018. doi: 10.1145/3170427.3188516

[63] X. Yan, C. W. Fu, P. Mohan, and W. B. Goh. Cardboardsense: Interacting with DIY cardboard VR headset by tapping. In *DIS 2016 - Proceedings of the 2016 ACM Conference on Designing Interactive Systems: Fuse*, pp. 229–233, 2016. doi: 10.1145/2901790.2901813

[64] Y. Yan, C. Yu, X. Yi, and Y. Shi. Headgesture: hands-free input approach leveraging head movements for hmd devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–23, 2018.

[65] K. Yatani and K. N. Truong. BodyScope: A wearable acoustic sensor for activity recognition. In *UbiComp'12 - Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 341–350, 2012. doi: 10.1145/2370216.2370269

[66] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi. Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4479–4488, 2017.

[67] C. Zhang, P. Patras, and H. Haddadi. Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(3):2224–2287, 2019.

[68] C. Zhang, A. Waghmare, P. Kundra, Y. Pu, S. Gilliland, T. Ploetz, T. Starner, O. Inan, and G. Abowd. FingerSound: Recognizing unistroke thumb gestures using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–19, 2017. doi: 10.1145/3130985

[69] K. Zhu, X. Ma, H. Chen, and M. Liang. Tripartite effects: Exploring users' mental model of mobile gestures under the influence of operation, handheld posture, and interaction space. *International Journal of Human–Computer Interaction*, 33(6):443–459, 2017. doi: 10.1080/10447318.2016.1275432