RimSense: Enabling Touch-based Interaction on Eyeglass Rim Using Piezoelectric Sensors

WENTAO XIE, Hong Kong Univ. of Science and Technology and Southern Univ. of Science and Technology
HUANGXUN CHEN, Hong Kong University of Science and Technology (Guangzhou)
JING WEI, University of Melbourne
JIN ZHANG, Southern University of Science and Technology
QIAN ZHANG, Hong Kong University of Science and Technology

Smart eyewear's interaction mode has attracted significant research attention. While most commercial devices have adopted touch panels situated on the temple front of eyeglasses for interaction, this paper identifies a drawback stemming from the unparalleled plane between the touch panel and the display, which disrupts the direct mapping between gestures and the manipulated objects on display. Therefore, this paper proposes RimSense, a proof-of-concept design for smart eyewear, to introduce an alternative realm for interaction - touch gestures on eyewear rim. RimSense leverages piezoelectric (PZT) transducers to convert the eyeglass rim into a touch-sensitive surface. When users touch the rim, the alteration in the eyeglass's structural signal manifests its effect into a channel frequency response (CFR). This allows RimSense to recognize the executed touch gestures based on the collected CFR patterns. Technically, we employ a buffered chirp as the probe signal to fulfil the sensing granularity and noise resistance requirements. Additionally, we present a deep learning-based gesture recognition framework tailored for fine-grained time sequence prediction and further integrated with a Finite-State Machine (FSM) algorithm for event-level prediction to suit the interaction experience for gestures of varying durations. We implement a functional eyewear prototype with two commercial PZT transducers. RimSense can recognize eight touch gestures on the eyeglass rim and estimate gesture durations simultaneously, allowing gestures of varying lengths to serve as distinct inputs. We evaluate the performance of RimSense on 30 subjects and show that it can sense eight gestures and an additional negative class with an F1-score of 0.95 and a relative duration estimation error of 11%. We further make the system work in real-time and conduct a user study on 14 subjects to assess the practicability of RimSense through interactions with two demo applications. The user study demonstrates RimSense's good performance, high usability, learnability and enjoyability. Additionally, we conduct interviews with the subjects, and their comments provide valuable insight for future eyewear design.

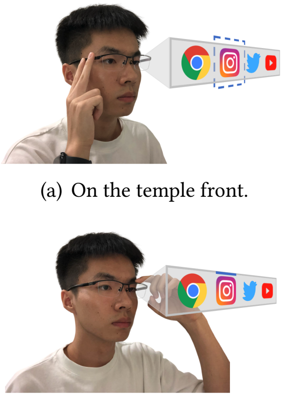
CCS Concepts: • **Human-centered computing → Gestural input**; **Mobile devices**.

Additional Key Words and Phrases: eyewear, interaction, touch gesture, piezoelectric sensor

ACM Reference Format:
Wentao Xie, Huangxun Chen, Jing Wei, Jin Zhang, and Qian Zhang. 2023. RimSense: Enabling Touch-based Interaction on Eyeglass Rim Using Piezoelectric Sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 4, Article 191 (December 2023), 24 pages. https://doi.org/10.1145/3631456

Authors' addresses: Wentao Xie, wxieaj@cse.ust.hk, CSE, Hong Kong Univ. of Science and Technology, Hong Kong and Southern Univ. of Science and Technology, Shenzhen, China; Huangxun Chen, huangxunchen@hkust-gz.edu.cn, IoT Thrust, Information Hub, Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China; Jing Wei, jing.wei@student.unimelb.edu.au, CIS, University of Melbourne, Melbourne, Australia; Jin Zhang, zhang.j4@sustech.edu.cn, Shenzhen Key Laboratory of Safety and Security for Next Generation of Industrial Internet, CSE, Southern University of Science and Technology, Shenzhen, China; Qian Zhang, qianzh@cse.ust.hk, CSE, Hong Kong University of Science and Technology, Hong Kong.

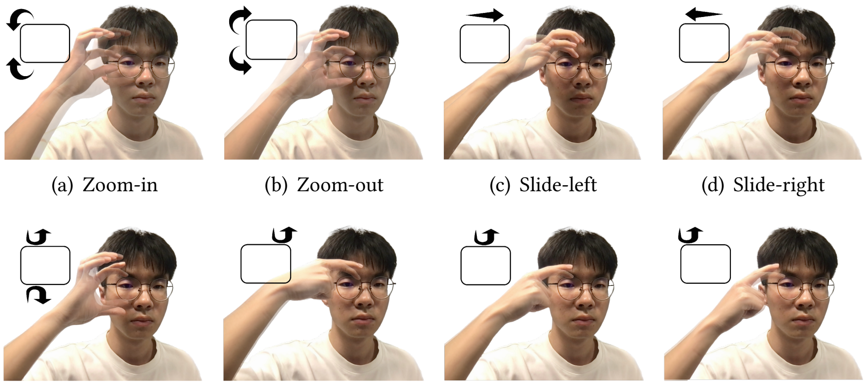
2474-9567/2023/12-ART191 $15.00
https://doi.org/10.1145/3631456



(a) On the temple front.

(b) On the rim.

Fig. 1. Eyewear interaction.



(a) Zoom-in

(b) Zoom-out

(c) Slide-left

(d) Slide-right

(e) Press

(f) Tap-left

(g) Tap-mid

(h) Tap-right

Fig. 2. RimSense supports eight input gestures that are performed on the rim.

1 INTRODUCTION

Smart eyewear, commonly referred to as AR glasses, represents a novel form of smart wearable technology that employs advanced waveguide technology to project graphical information onto the eyeglass. According to a recent market report, the public demand for AR glasses is projected to increase considerably in the coming years, with an anticipated market value of up to 883.4 million US dollars by 2025 [1].

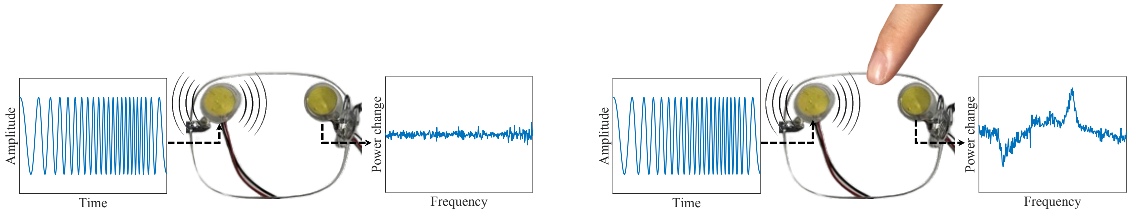
Interactions with smart eyewear have undergone comprehensive research, including gaze gestures [12, 25, 31, 37, 38, 47, 65], facial expressions [28, 29, 46, 49, 60, 71], head movements [22, 53, 75, 77], mid-air gestures [15, 17–20, 27, 34, 42, 76], touch gestures [2, 4, 7–10, 55, 66, 70] and etc. Among these approaches, touch panels positioned at the temple region of the eyeglass frame, as illustrated in Figure 1(a), have become the predominant mode of interaction in most commercial AR glasses [2, 4, 7–10]. However, despite the touch panel's recognition reliability, it is noticed that the eyeglass temple is not aligned with the same plane as the display. Consequently, to manipulate virtual objects on display, a user slides the eyeglass temple *forward and backward* to make a cursor shuffle between the *left and right* sides and then confirms a selection by clicking on the panel. That says the slide direction is perpendicular to the display. This discrepancy disrupts the inherent mapping between gestures and manipulated objects and goes against the previously established user habits through the use of touch screens on smartphones and tablets.

This motivates us to introduce an alternative realm of interaction - eyewear rim to address the previously mentioned discrepancy. As shown in Figure 1(b), the user can navigate the screen between *left and right* by executing a *left or right* slide gesture on the upper rim. Furthermore, suppose there are four app icons positioned on the glass, the user can touch the corresponding locations on the rim to open the target app. This design of interaction introduces a natural mapping, a principle well-regarded in interaction design practices [52], connecting the gestures employed for control to the display itself, as the eyewear rim exists within the same plane as the display.

In this study, our goal is to propose and evaluate a proof-of-concept design that makes the rim of an eyeglass interactable. A potential hardware configuration for this design is to incorporate a touch panel, *e.g.*, utilizing thin thread [36] in the shape of an eyeglass rim. However, fabricating slender touch-sensitive material to accommodate diverse rim shapes while simultaneously upholding stable recognition accuracy is challenging. Furthermore, the thin structure could be susceptible to wear and tear from prolonged usage, thereby potentially leading to degraded recognition performance. Therefore, we propose to convert the eyeglass rim into a touch-sensitive surface via an external stimulus source rather than add extra material directly onto the already slender rim. Drawing inspiration from previous research [54], we choose piezoelectric (PZT) transducers as the facilitators for our concept. The sheet-like shape of PZT transducers is well-suited for seamless integration with eyeglasses. Moreover, recent advancements in material science have showcased the feasibility of transparent PZT sensors [58, 59], thereby mitigating potential vision obstruction issues. As shown in Figure 3, our prototype incorporates two compact PZT transducers affixed to the eyeglass. One transducer is responsible for generating imperceptible vibrations, while the other is tasked with reception to detect touch gestures. The underlying principle is that the two sensors, in conjunction with the eyeglass, form a vibration system with a structural signature - channel frequency response (CFR) shown in Figure 3 - during vibration. A user's touch on the rim results in a modification in this structural signature, the pattern of which depends on the touch manner. Therefore, the touch gesture can be recognized by detecting and classifying this changing pattern.

To realize such a system, we encounter two major challenges: waveform design for stimulus source, *i.e.*, probe signal, and algorithm design tailored for real-time gesture detection. In the first aspect, our goal is to have a waveform with both noise resistance and high sensing granularity. However, in existing relevant works [54, 67], these two objectives generally contradict each other. That says, it is common to use a damping window (such as the Hanning window) on the probe signal to reduce the impact of impulse noise caused by sudden frequency jumps between successive probes. Unfortunately, this approach inevitably limits the effective sensing bandwidth and reduces sensing granularity. Drawing inspiration from waveform design in the communication domain [21, 40], we adopt a buffered chirp design to incorporate gradual frequency changes between successive probes instead of applying a damping window. This approach allows us to maintain a wide sensing bandwidth while effectively reducing impulse noise. In the second aspect, we first design a set of gestures to fulfill interaction requirements on the eyeglass rim, as illustrated in Figure 2. Compared with the detection of static grasping gestures in [54], we notice that interaction gestures on the eyeglass rim inherently have varying durations. For instance, in our real-world experiments, tapping actions (Figure 2(f)-(h)) might only take 0.5 seconds, whereas zooming actions (Figure 2(a)-(b)) could last for up to 3 seconds. Moreover, different users might exhibit diverse tendencies in terms of gesture speed. An intuitive approach for detection is to utilize classification techniques, such as random forest, alongside a sliding detection window. However, producing a single prediction within a fixed detection window does not yield a satisfactory real-time interaction experience for gestures with varying durations. This is because using a long window sacrifices detection latency for short gestures, while a short window sacrifices detection accuracy for long gestures. To overcome this dilemma, we propose to enable fine-grained timestep-level prediction that avoids gesture prediction on a window basis. The proposed scheme predicts the gesture for every timestep and subsequently derives a reliable event-level prediction for each gesture event. This approach makes it more adaptable to gestures of arbitrary durations. For long gestures, we can aggregate features from relevant timeslots to achieve high detection accuracy. For short gestures, we can detect the concluding timeslot promptly to provide an instantaneous interactive response. Furthermore, the fine-grained timestep-level prediction greatly supports the measurement of each gesture's duration. This broadens the design scope of the gesture set, as the same gesture with different durations can function as distinct gesture inputs. For instance, a long tap and a short tap can be programmed to execute different control tasks.

Our resulting system, named RimSense and depicted in Figure 4, has three modules. In the probe signal generation module, we utilize an FMCW signal with a buffered chirp design as the probe signal. This signal is applied to one PZT transducer, transforming the eyeglass rim into a touch-sensitive surface. In the CFR extraction module, we apply signal processing to the received signal from the other PZT transducer. This processing is aimed at extracting CFR features relevant to gestures. In the real-time gesture recognition module, we design a deep learning-based classifier to generate predictions at the timestep level using the extracted CFR features. In contrast to conventional classifiers like random forest and SVM, deep neural networks offer greater flexibility



(a) No gesture.

(b) With gesture.

Fig. 3. Sensing principle.

for adapting to the requirements of timestep-level prediction. In particular, RimSense's classifier (Figure 10) incorporates CNN blocks to capture features from CFR amplitude and phase and GRU blocks to generate features for each timestep, facilitating timestep-level prediction. In addition, self-attention blocks are used to relate the CNN blocks and GRU blocks. Furthermore, data augmentation techniques are implemented to enhance classifier performance and ease the demand for extensive training dataset collection. Building upon the timestep-level predictions, we develop a Finite-State Machine (FSM) algorithm (Figure 12) to achieve event-level prediction. This approach ensures real-time gesture recognition that delivers a satisfactory user interaction experience.

Our contribution can be summarized as follows. The main contribution is that, to the best of our knowledge, this study is the first to showcase the feasibility of utilizing the eyeglass rim as a new interaction space for smart eyewear. We present a proof-of-concept interaction system based on PZT sensors that transform the slim rim into a touch-sensitive surface, enabling the recognition of eight distinct touch gestures on the eyeglass rim. These gestures, including zoom-in, zoom-out, slide-left, slide-right, press, tap-left, tap-mid, and tap-right, as depicted in Figure 2, are selected to align with the common experiences when interacting with a smartphone[1]. Additionally, we adopt the probe signal with a buffered chirp to fulfil the need of both sensing granularity and noise resistance (Section 3.1). Furthermore, we propose transitioning from window-level to timestep-level prediction granularity to suit the interaction experience for gestures with varying durations. We devise a deep learning-based gesture recognition framework for timestep-level prediction, integrated with an FSM algorithm to enable event-level prediction to support real-time gesture inference (Section 3.3). Lastly, we implement a functional eyewear prototype that integrates two commercial PZT transducers into eyeglasses with real-time operation. We conduct two studies to evaluate RimSense's performance: an offline study involving 30 participants to assess gesture recognition accuracy using leave-one-subject-out validation (Section 5), and a user study involving 14 participants to evaluate usability across two demo applications (Section 6). The results demonstrate RimSense's high accuracy (F1-score of 0.95 and 11% relative error in gesture duration estimation), high usability, learnability, and enjoyability. In addition, our interviews with the subjects provide valuable insight into the future development of smart eyewear.

2 SENSING PRINCIPLES

When subjected to external probe signals, an object exhibits different damping factors across different probing frequencies, depending on its shape and structure. In other words, each object generates its own signature response given a probe source. This distinct signature response is regarded as its channel frequency response (CFR). Conversely, if we can detect an object's CFR, we may be able to infer its structural characteristics [57]. RimSense leverages this principle to sense touch gestures. As illustrated in Figure 3, when human fingers touch

[1] In this research, we restrict the tap and slide gestures to be performed only on the upper side of the rim.

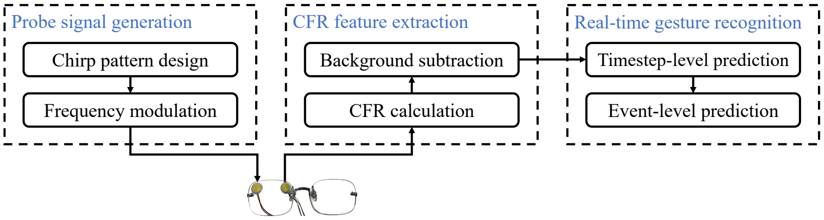


Fig. 4. RimSense's system overview

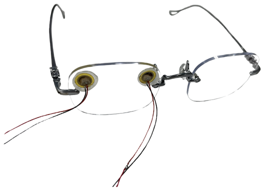


Fig. 5. RimSense prototype.

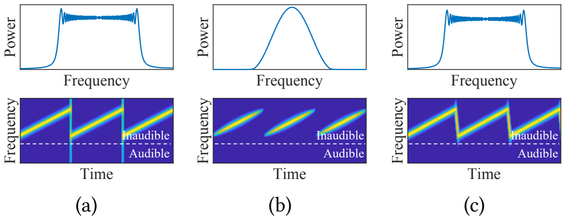


Fig. 6. Chirp designs. (a) Original chirp. (b) Chirp with Hanning window. (c) Chirp with frequency buffer. (Color represents power in 2nd-row figures)

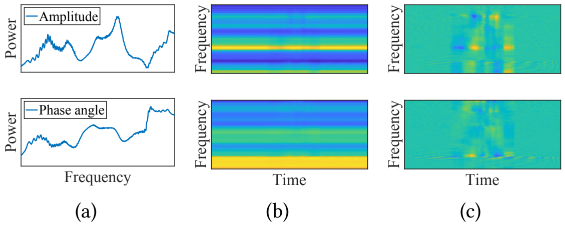


Fig. 7. CFR feature extraction. (a) A frame of CFR. (b) 3s of CFRs. (c) 3s of CFRs after background subtraction. (1st row: amplitude, 2nd row: phase angle)

the eyeglass, the eyeglass's structure undergoes alteration, thereby causing changes in its CFR. Thus, RimSense can infer the touch gesture by recognizing the CFR changing pattern. As mentioned in Section 1, an eyeglass is equipped with two PZT sensors. PZT sensors can transduce between the vibration signal and the electrical signal. One of these sensors employs probe signals, represented as x, to induce specific vibrations in the eyeglass, while the other sensor collects the resultant response signal, denoted as y. The CFR can be computed as $H = \frac{Y}{X}$, where X and Y correspond to the Fourier transforms of x and y, respectively. The CFR H encapsulates information about the touch gesture.

3 SYSTEM DESIGN

The system overview of RimSense is depicted in Figure 4. RimSense contains three modules. The first is the probe signal generation module. This module generates a specially-designed frequency-buffered chirp signal tailored for CFR measurement. The second is the CFR feature extraction module. This module takes the measured probe signal as the input and computes the CFR patterns caused by the user's input gesture. Based on the computed CFR pattern, the third module, the real-time gesture recognition module uses a deep learning model for tracking the input gesture at the timestep level and further integrates with a finite state machine to achieve real-time event-level gesture prediction. The following sections elaborate on each individual module respectively.

3.1 Probe Signal Generation

As discussed in Section 2, RimSense detects a touch gesture by monitoring the CFR pattern. The common way to measure the CFR of an object is to use chirp signals whose frequency linearly sweeps over a predefined range [54, 61, 62, 67]. However, for vibration-based designs [54, 67], to reduce the impulse noise caused by sudden frequency jump between successive chirps (Figure 6(a)), a Hanning window is often used to damp the starting

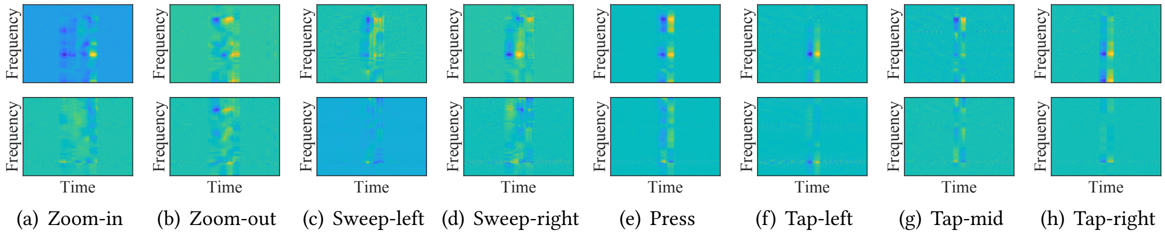


Fig. 8. The CFR patterns of a subject performing the eight touch gestures. (Color represents amplitude)

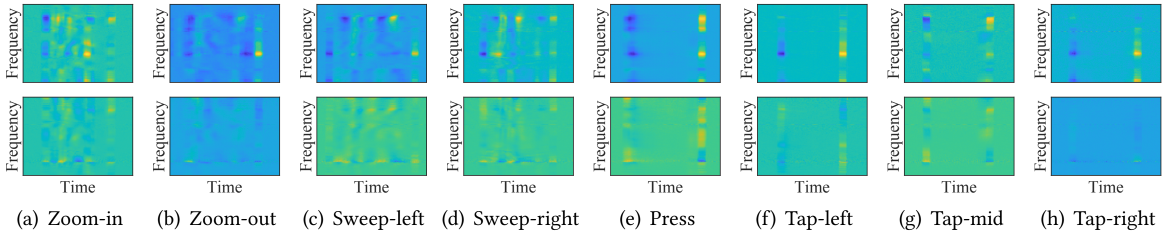


Fig. 9. The CFR patterns of another subject performing the eight touch gestures.

and ending portions of the chirp signal. One problem with this design is that it limits the usable bandwidth for CFR measurement, as shown in Figure 6(b).

In RimSense, we leverage a frequency-buffered chirp signal to maintain the maximal usable bandwidth while not causing any impulse noises. The rationale of the design is described as follows. Without loss of generality, we assume a chirp signal is an up-chirp that sweeps from the lowest designated frequency to the highest. After the frequency reaches the highest, we append a short down-chirp signal as a buffer to let the frequency drops continuously from the highest to the lowest. In this way, the frequency flows smoothly from the current chirp to the next chirp, and no impulse noise will be generated. The frequency-buffered chirp design is shown in Figure 6(c). Note that in the subsequent processes, the buffering down-chirp is discarded, and only the up-chirp is kept for further processing.

In practice, we use frequency modulation to generate the probe signals. We first compute the frequency pattern as a function of time. This frequency function is denoted as $f(t)$. Then, the probe signal is derived as $x(t) = cos(2\pi \int_0^t f(t)dt)$. In RimSense, the designated frequency $f(t)$ vibrates linearly and periodically from $20kHz$ to $40kHz$. Also, the durations for the up-chirp and down-chirp are $19ms$ and $1ms$. Therefore, within each cycle of $f(t)$, it goes from the lowest to the highest for the first $19ms$ and from the highest to the lowest for the last $1ms$. The designed probe pattern is fed to one of the PZT sensors on the glass to vibrate the eyewear system. It is worth noting that although the eyewear is vibrated by a PZT sensor, the vibration is imperceivable because of the extremely low power and high frequency. According to our user study described in Section 6, none of our participants noticed any vibration or noise generated by the eyewear.

3.2 CFR Feature Extraction

For each frame of the transmitted and received signals, we denote x as the transmitted up-chirp signal and y as the corresponding received up-chirp. Then, the CFR is computed as $H = \frac{Y}{X}$ as introduced in Section 2. Note that here, H is a complex sequence. We decompose H into amplitude and phase as $H = Ae^{j\theta}$, and we keep the

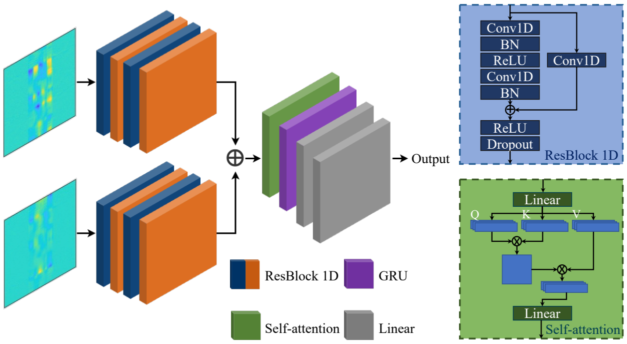


Fig. 10. Model architecture.

Table 1. Model parameters.

Layer	Parameters
ResBlock1D 1	Kernel #: 128, kernel size: 3, stride: 2
ResBlock1D 2	Kernel #: 64, kernel size: 3, stride: 2
ResBlock1D 3	Kernel #: 32, kernel size: 3, stride: 1
ResBlock1D 4	Kernel #: 16, kernel size: 3, stride: 1
Self-attention	Input: 32, embedding: 64, head #: 8
GRU	Input: 64, embedding: 64, bi-directional
Linear - 1	Input: 128, output: 64, dropout rate: 0.5
Linear - 2	Input: 64, output: 9

amplitude A and phase angle θ to characterize the CFR of the current frame. Figure 7(a) depicts a frame of CFR. Note that we discard the signal outside of the 20-40kHz range.

RimSense transmits a frame of probe signal every 20ms, so the frame rate for CFR is 50Hz. Therefore, we can stack multiple CFR frames to see how a touch gesture influences CFR. Figure 7(b) shows 150 CIR frames (3 seconds) of a subject performing a sweep-left on the glass. However, We can only observe mild disturbance patterns in the CFR amplitude and there is almost no change in the CFR phase angle. This is because of the strong static CIR components that do not vary over time. Therefore, we can apply a background subtraction process to mitigate the effect of the static components. We achieve this by subtracting a frame by the frame 0.02s ahead of time. Figure 7(c) shows the CIR amplitude and phase angle after background subtraction, and we can see clear patterns that represent a touch gesture. For the rest of this paper, we use the term "CIR patterns" to refer to the stack of a series of CFR amplitude and phase angle signals that have undergone background subtraction.

Figure 8 and Figure 9 depict the CIR patterns of two different subjects performing the eight touch gestures. We can observe that the CFR patterns of the two subjects share some similarities for the same gesture, especially for press and tap gestures. However, the duration of the same gesture between these two subjects varies a lot. For example, the first subject performs a tap-mid within 0.5s, but it takes around 2.5s to perform the same gesture for the second subject. This variation in gesture duration brings a major challenge for the design of RimSense which will be discussed in the next section.

3.3 Real-time Gesture Recognition

Next, we discuss how RimSense recognizes the performed gestures based on the CFR patterns derived from the previous module. Due to the varied gesture duration between different gestures and subjects, achieving real-time prediction is challenging. This is because it is hard to balance the length of the gesture detection window and the delay caused by the window. When a large detection window (*e.g.*, a 3-second window) is used, the prediction lag will be large, especially for the short gestures. This is because it would take a long time for the gesture (patterns of high amplitude in Figure 8 and Figure 9) to move out of the detection window. Instead, If a shorter detection window is used, it may not cover the long gestures so the gesture may not be correctly predicted, particularly for gestures with blank CFR patterns in the middle (Figure 9(e)-Figure 9(h)).

In this module, we design a two-step approach to resolve the above challenge. First, we use a deep-learning model to predict the performed gesture for each time step. This facilitates the prediction of the exact start and end time of a gesture. Next, we use a finite-state machine to decide when is the correct time to give a gesture prediction.

3.3.1 Timestep-level Prediction. The deep learning model architecture is depicted in Figure 10. It contains four stages, the convolutional stage, the attention stage, the recurrent stage, and the output stage. The convolutional stage takes a pair of CFR patterns as the input and passes them through a series of convolutional layers. Specifically, the convolutional stage has two branches, one for processing the CFR amplitude and the other for the CFR phase angle. Each of these two branches stacks four classic residual blocks [26]. After the convolutional stage, the processed CFR amplitude and phase angle features are concatenated along the frequency dimension and fed to the attention stage where one layer of multi-head self-attention [68] is applied. In the recurrent stage, a Gated Recurrent Unit (GRU) is applied to further process the feature and produce an embedding for each timestep. Finally, at the output stage, a fully-connected layer is applied to each timestep embedding to give the prediction at that timestep. The detailed network parameters are summarized in Table 1. Note that the CFR frame rate is $50Hz$, and it will be down-sampled to $12.5Hz$ by the convolutional stage. Therefore, we resample the ground truth label to $12.5Hz$ to make sure every timestep has a label.

To increase the robustness of the model, we apply a few data augmentation techniques to enhance the training dataset. The data collection process is described in Section 5.1. For each original sample, we apply three augmentation techniques [30] to generate a new sample, and the augmentation pipeline is applied to each original sample five times to enlarge the dataset by five times. The techniques are: (i) shifting: to lag or move forward a sample by a time randomly selected from -1.5 to 1.5 seconds for simulating the randomness that a gesture can appear anywhere in the input CFR patterns; (ii) accelerating: to accelerate a sample by a factor chosen from 1.0 to 1.5 for simulating different gesture speeds; (iii) scaling: to scaling a sample with a scaling factor $s \sim \mathcal{N}(1,\ 0.2^2)$ for simulating different gesture strengths. In our evaluation, we have collected 3586 samples, and the dataset size is 15238 samples after data augmentation. An important thing to note here is that although we have balanced the number of samples for each gesture in our data collection process (Section 5.1), since our model is to predict the gesture at each timestep, the total number of timesteps for each gesture is unbalanced. Therefore, we compute the total number of timesteps for each gesture and compensate for the unbalance through class weights.

Examples of timestep-level gesture prediction are shown in Figure 11. These two examples depict the prediction between the time when the gesture just arrives in the detection window and when the gesture is about to leave the detection window. The time gap between adjacent figures is 1 second.

3.3.2 Event-level Prediction. From examples in Figure 11, we can observe that a gesture will remain in the detection window for several seconds. However, in real-time usage, we want the gesture command to be issued only once as soon as the gesture is performed. Therefore, we propose a finite-state machine (FSM)-based algorithm to decide the exact time when a gesture is over, and the gesture prediction is given only at this particular time point. The FSM is shown in Figure 12.

We first define the prediction front (PF) as the last 5 samples ($0.4s$) of the model prediction. If the majority of the predictions in the prediction front are not null, we say this is a positive prediction front, otherwise, the prediction front is negative. There are two states in the FSM, the wait-for-gesture state and the wait-for-prediction state. The following walk through one cycle of the FSM to show how a gesture event is predicted. (1) The system starts with the wait-for-gesture state. If the PF is negative, then the system remains in this state. Instead, if the PF is positive, it means the detection window is about to welcome a gesture (Figure 11(a) and Figure 11(e)). Then, the system moves to the wait-for-prediction state. It means the system is expected to give a prediction soon. (2) At the wait-for-prediction state, if the PF gives a positive reading again, then the system will do nothing because it means the gesture is not finished yet. An example of this case is shown in Figure 11(b). Once the PF gives a negative at this state, the system gives a gesture prediction because it means the user has just finished a gesture input. Examples of this state are Figure 11(c) and Figure 11(f). After that, the system returns to the wait-for-gesture state and waits for another round of gesture prediction. Note that when predicting a gesture

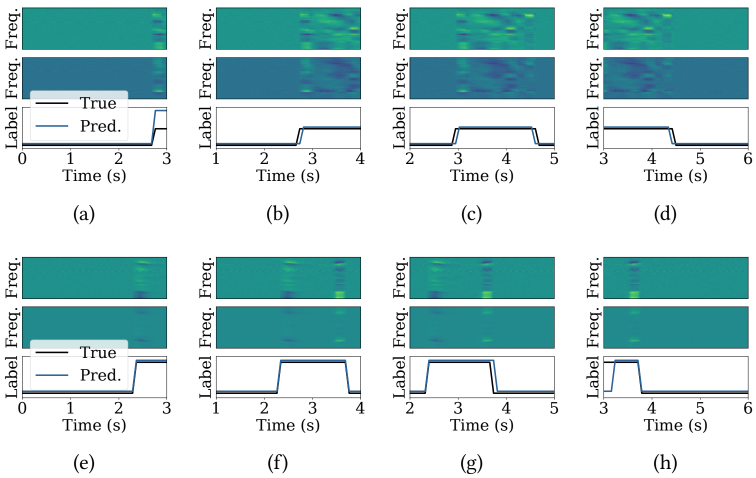


Fig. 11. Timestep-level gesture prediction. (a-d) Zoom-in. (e-h) Tap-right. (1st row: CFR amplitude, 2nd row: CFR phase angle, 3rd row: prediction and ground truth)

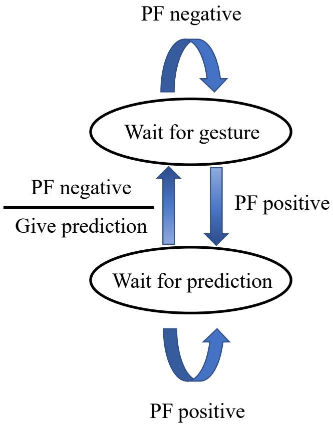


Fig. 12. The FSM for event-level prediction.

event, the system summarizes all the timestep-level predictions in the current detection window and outputs the gesture class (except for the null class) with the longest duration. Also, the gesture duration is output as well.

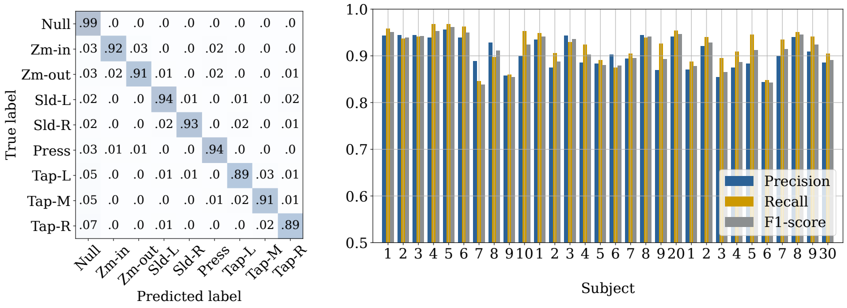
In real-time usage, we configure RimSense so that it gives a prediction (either a gesture or a null) every 0.2 seconds, *i.e.*, $5Hz$ refresh rate. It is worth noting that in the real world, there might be cases when one gesture is immediately followed by other gestures. In other words, there might be two or more different gestures in one detection window. To mitigate the influence that one gesture may have on the prediction of the following gestures, we use a feature masking technique to mask the CFR patterns of previous gestures if they have been output by RimSense. Specifically, once a gesture event is recognized by RimSense, the system replaces the CFR patterns that belong to this gesture with random noise $n \sim \mathcal{N}(0,\ 0.01^2)$.

4 IMPLEMENTATION

The hardware prototype of RimSense is shown in Figure 5. We use a standard pair of glasses for building the prototype, and we affix two commercial PZT sensors on the upper-left and upper-right corners of the right glass using a glue gun. The PZT sensors, of type PUI Audio AB1070B-LW100-R [5], are $10mm$ in diameter and $0.12mm$ in thickness. The PZT sensors are connected to an audio interface card of type Focusrite Scarlett 2i2 [6] through a 3.5mm audio interface. The audio interface card is connected to a 13-inch MacBook Pro 2019 laptop with a 1.4 GHz Quad-Core Intel Core i5 CPU for software processing. The software of RimSense is implemented using Python 3.8. The deep learning model is developed with PyTorch 1.10.0 and trained on a server with an NVIDIA V100 GPU. The model's training involves an Adam optimizer [35], a batch size of 512 samples, and a learning rate of 0.001. The training process lasts 200 epochs.

5 EVALUATION

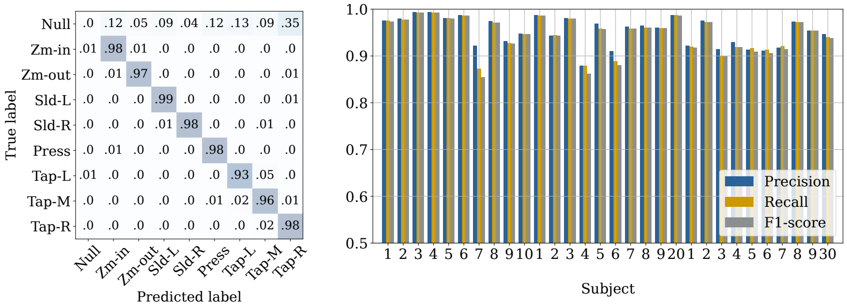
In this section, we evaluate RimSense's technical performance in touch gesture recognition in an offline manner. The collected data in this section are stored on a hard disk and processed offline by the software. We report



(a) Confusion matrix.

(b) Performance on each subject.

Fig. 13. Timestep-level performance. (Zm: zoom; Sld: slide; L: left; M: mid; R: right)



(a) Confusion matrix.

(b) Performance on each subject.

Fig. 14. Event-level performance. (Zm: zoom; Sld: slide; L: left; M: mid; R: right)

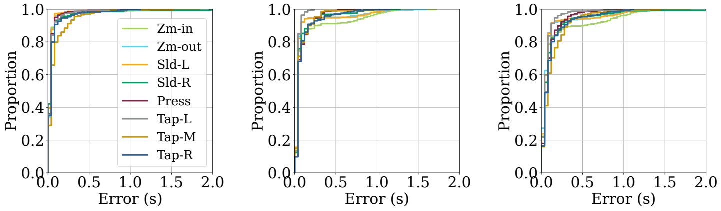
RimSense's performance in terms of timestep-level gesture prediction accuracy, event-level gesture prediction accuracy, gesture duration estimation error, and latency.

5.1 Data Collection

This study recruited 30 participants (15 females and 15 males). Most of the participants are either students or staff members from our campus, with an average age of 27.6 years. Most subjects are between 18 and 35 years old, with two exceptions - a 62-year-old female and a 65-year-old male. Among the 30 participants, seven do not wear glasses daily. Each participant completed one or two data collection sessions depending on their own will, with each session lasting approximately 15 minutes. Participants received a compensation of 25 HKD[2] after completing each data collection session. The experiments were conducted in a lab room where normal lab activities, such as people walking and discussion, occurred during the experiments. Participants were allowed to move their bodies and talk freely during the experiment. The data collection setup is similar to that of Figure 16.

Each data collection session begins with an introduction to the study, followed by a tutorial on how to perform the eight target touch gestures: zoom-in, zoom-out, slide-left, slide-right, press, tap-left, tap-mid, and tap-right (Figure 2). The participants are given time to learn and get familiar with the gestures. We divide a data collection

[2] Around 3.20 USD (1 HKD ≈ 0.13 USD).



(a) Duration (b) Starting time (c) End time

Fig. 15. Gesture duration estimation. (ZM: zoom; Sld: slide; L: left; M: mid; R: right)

Table 2. The latency of RimSense. (BG: background)

CFR feature extraction		Real-time gesture recognition		Total
4.02ms		13.69ms		17.71ms
CFR calculation	BG subtraction	Timestep level	Event level	
3.57ms	0.45ms	13.60ms	0.09ms	

session into four sub-sessions in which similar gestures are grouped together. The first sub-session contains zoom-in and zoom-out, while the second sub-session contains slide-left and slide-right. The third sub-session focuses solely on the press gesture, while the last sub-session includes tap-left, tap-mid, and tap-right. During each sub-session, the participant wears the RimSense prototype and sits comfortably in front of a computer screen. The computer screen displays a visual instruction video to guide the participant in performing each touch gesture. For each gesture, the video first displays the gesture name and a suggested gesture speed, followed by a three-second timer to let the participant complete the gesture in three seconds, and then the video proceeds to the next gesture. The speed suggestion is either fast, regular, or slow, and participants themselves manage the actual gesture speed. The speed suggestion is used to increase the variability of the dataset. Each gesture is repeated nine times within each sub-session. This study has been approved by the Institutional Review Board (IRB) of our institution[3].

5.2 Gesture Recognition Accuracy

As discussed in Section 3.3, RimSense utilizes a two-stage approach to detect touch gestures. The first stage involves predicting the gesture for each timestep, while the second stage involves predicting a gesture event by analyzing a series of timestep-level predictions. In this section, we evaluate RimSense's performance in recognizing gestures at both these levels. To present our results, we utilize leave-one-subject-out (LOSO) validation.

5.2.1 Timestep-level Accuracy. To assess the timestep-level accuracy, we segment the data from each data collection session into segments of 3-second duration and with a step size of 0.2 seconds. This is done to mimic RimSense's 5Hz output rate (discussed in Section 3.3.2). These segments are then fed into our model, and the model's predictions are compared with the ground truth.

Figure 13(a) illustrates the confusion matrix and the precision, recall, and F1-score for the predicted gesture. The confusion matrix shows that there is some confusion among the tap gestures, which is expected given their similarity. Nonetheless, the eight gestures are generally well classified. It is worth noting that all gestures have misclassified samples that are assigned to the null class. This is understandable since it might be confusing at the

[3] Hong Kong University of Science and Technology HREP-2023-0198.

start and end of each gesture, as shown in Figure 11. Figure 13(b) presents RimSense's performance across all 30 subjects, with an average precision, recall, and F1-score of 0.92, 0.91, and 0.91, respectively.

5.2.2 Event-level Accuracy. We also present an evaluation of RimSense's gesture recognition performance at the event level. As described in Section 3.3.2, we employ an FSM-based algorithm that reports a gesture prediction only once for a gesture event. Following the setup described in the previous section, we segment the data from each data collection session into 3-second segments with a step size of 0.2 seconds to simulate real-time prediction. We then apply the FSM algorithm to each session's data to obtain the event-level predictions of that session. For each targeted gesture G in the event-level evaluation, we define its true positive, false positive, and false negative samples as follows.

- True positive (TP). The predicted gesture interval is overlapped with the true gesture interval, and the predicted gesture is correct, *i.e.*, gesture G.
- False positive (FP). There is a gesture prediction for gesture G but the ground truth shows there is no gesture overlapped with the prediction, or there is a gesture event overlapped with the prediction but the actual gesture is not gesture G.
- False negative (FN). The ground truth shows that there is a gesture G at a certain time period but there is no predicted gesture that is overlapped with the ground truth gesture G, or there is a gesture event overlapped with the ground truth, but the prediction is wrong.

It's worth noting that at the event level, RimSense only outputs gestures. As a result, there is no true null sample in this evaluation. Figure 14(a) shows the confusion matrix of each gesture. Figure 14(b) depicts the performance of each subject. The average precision, recall, and F1 score among all subjects are 0.95, 0.95, and 0.95, respectively. The event-level performance is significantly improved compared to the timestep-level results. This is because the timestep-level predictions are aggregated to provide a single prediction at this level. As a result, minor errors, such as those at the beginning and end of each sample, are tolerated. Notably, the performance of some participants is worse than that of others, where the errors mainly come from the confusion among the three tapping gestures. This is possibly because these participants tend to perform the three tapping gestures with less spatial distinction.

5.3 Gesture Duration Estimation Error

In addition to recognizing gestures, RimSense estimates the duration of each gesture. In this section, we evaluate the accuracy of RimSense's gesture duration estimation. We calculate the start time, end time, and duration estimation errors of all true positive samples defined in the previous section. Figure 15 shows the cumulative distribution function (CDF) plots of these errors. Overall, the mean errors for start time, end time, and gesture duration are 52.5ms, 83.8ms, and 109.1ms, respectively. Notably, the 109.1ms duration error is equivalent to an 11% percentage error relative to the actual gesture duration.

5.4 Latency

We also examine the latency of RimSense's software system in this section. We report the average time consumed by two of the three modules involved in gesture recognition, namely, CFR feature extraction (Section 3.2) and real-time gesture recognition (Section 3.3). It is important to note that, in the current implementation, the software runs on a laptop. Hence, the latency needs to be further assessed when the system is actually built on a wearable device with limited computation resources in the future. We ran RimSense for five minutes, which resulted in 1500 gesture predictions, and computed the average execution time for each component. Table 2 presents the result. In general, it takes 17.71ms for RimSense to give a gesture prediction.

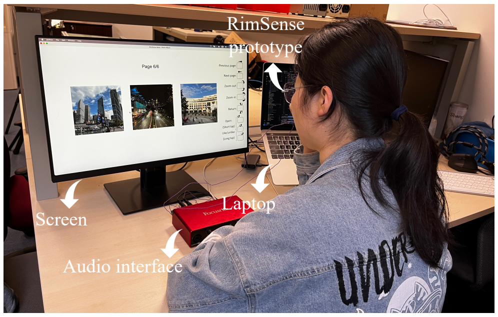


Fig. 16. The setup for the user study.

6 USER STUDY

In addition to the offline performance evaluation, we conduct a user study to evaluate the usability of RimSense. We implement a real-time version of RimSense and deploy two applications that users can interact with through RimSense. We evaluate the real-time and real-world gesture recognition performance and most importantly, we evaluate the usability of RimSense through user feedback. Note that different from real AR glasses where the virtual content is projected to the glass, in this study, we simulate the projection through a combination of normal glasses and a computer monitor. Therefore, when using the apps, a user needs to look at the monitor as a simulation of the experience with real AR glasses.

6.1 App Design

We develop a photo album app and an ebook reader app to test out RimSense. In the photo album app, the user can navigate photos that are stored in this app. The photo gallery is displayed with three photos in a row on each page, and users can open one of the three photos with a short tap ($<1s$). A tap-left opens the left photo, a tap-mid opens the middle one, and a tap-right opens the right photo. Alternatively, the user can use a long tap ($>1s$) to give a like to a photo (a heart icon will show below the photo) or unlike a photo (the heart icon will disappear). Once a photo is opened with a short tap, the user can zoom in or out the photo using the gestures zoom-in and zoom-out, and the user can return to the gallery page with a press gesture. To navigate to the next or previous page of the photo album, users can perform a slide-left or slide-right gesture. A long slide ($>1s$) results in navigating one page forward or backward, while a short slide ($<1s$) results in navigating two pages.

The ebook reader app features two modes: gallery mode and reader mode. In gallery mode, users can access all ebooks and select one to read in reader mode. The gallery mode here works similarly to that of the photo album app, including a long or short slide gesture to turn one or two pages, a short tap gesture to open a book, and a long tap gesture to give or cancel a like. Once a book is opened, the app enters reader mode where the content of the book is shown on the app. In this mode, users can turn to the next page using a tap-right or slide-left gesture, or turn to the previous page using a tap-left or slide-right gesture. Additionally, users can increase or decrease the font size using a zoom-in or zoom-out gesture. Finally, a press gesture can be used to return to gallery mode.

Screenshots of the two apps are shown in Figure 17 and Figure 18, and all gesture instructions and explanations are summarized in Table 3.

6.2 Study Procedure

14 participants are invited to participate in the user study, three of whom are female. The average age of the participants is 25 years old. The experiment environment is consistent with that of Section 5. Following an

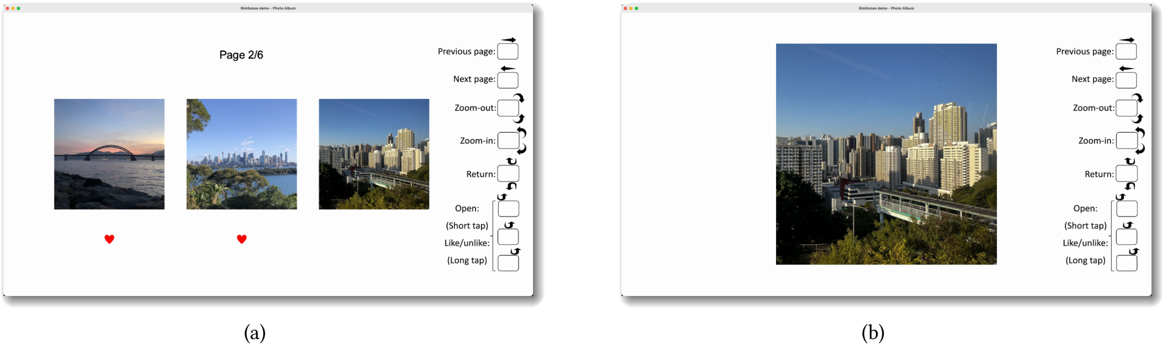
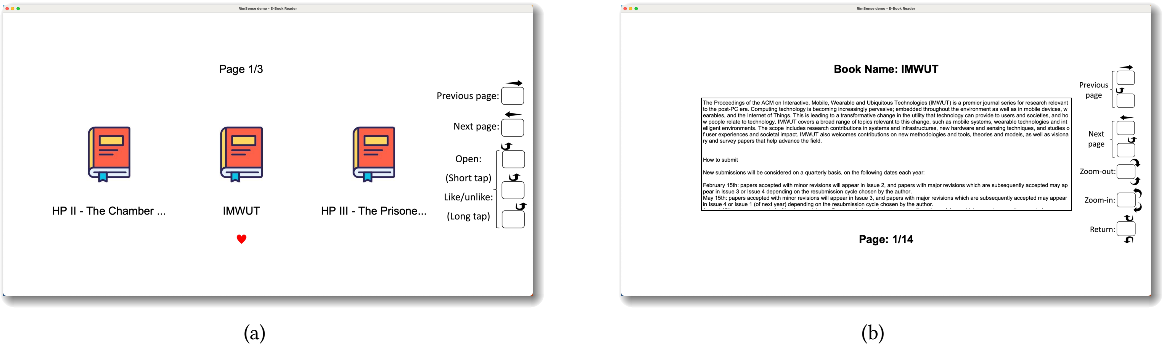


Fig. 17. Screenshots of the photo album app. (a) The gallery page. (b) The photo page.



Book Name: IMWUT

The Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) is a premier journal series for research relevant to the post-PC era. Computing technology is becoming increasingly pervasive; embedded throughout the environment as well as in mobile devices, wearables, and the Internet of Things. This is leading to a transformative change in the utility that technology can provide to users and societies, and how people relate to technology. IMWUT covers a broad range of topics relevant to this change, such as mobile systems, wearable technologies and intelligent environments. The scope includes research contributions in systems and infrastructures, new hardware and sensing techniques, and studies of user experiences and societal impact. IMWUT also welcomes contributions on new methodologies and tools, theories and models, as well as visionary and survey papers that help advance the field.

How to submit

New submissions will be considered on a quarterly basis, on the following dates each year:

February 15th: papers accepted with minor revisions will appear in Issue 2, and papers with major revisions which are subsequently accepted may appear in Issue 3 or Issue 4 depending on the resubmission cycle chosen by the author.
May 15th: papers accepted with minor revisions will appear in Issue 3, and papers with major revisions which are subsequently accepted may appear in Issue 4 or Issue 1 (of next year) depending on the resubmission cycle chosen by the author.

Page: 1/14

Fig. 18. Screenshots of the ebook reader app. (a) The gallery page. (b) The reader page.

introduction to the study, participants are instructed on how to use the two apps described above, with specific emphasis placed on the gesture instructions outlined in Table 3. Subjects were given the opportunity to practice the gestures on the prototype to become familiar with the interaction method. The introduction and practice phase typically lasted approximately five minutes. The setup for the user study is shown in Figure 16.

The experiment starts once the participant feels fully prepared. For each app, the participant is verbally asked by the researcher to complete three tasks before they can navigate the app freely. The tasks are designed to ensure the participant is engaged in the study and can use all of the gestures. An example task is: "Please turn to page four and open the second photo on that page. Take a closer look at it and give it a like if you like the photo". Each app navigation requires approximately five minutes to complete. Throughout the experiment, a camera was used to record the subject's face, providing ground truth data to evaluate RimSense's performance in real app usage.

Upon the completion of the experiment session, participants are asked to fill out the System Usability Scale (SUS) questionnaire [16] to provide feedback on the usability of RimSense, as well as a customized questionnaire to address other user concerns. Additionally, we conduct an exit interview with each participant to learn about their user experience and subjective comments about RimSense. The exit interview specifically includes questions

Table 3. The gesture instructions and explanations.

		Zm-in	Zm-out	Sld-left		Sld-right		Prs	Tp-left		Tp-mid		Tp-right	
				Short	Long	Short	Long		Short	Long	Short	Long	Short	Long
Photo Album	**Gallery**	/	/	Nxt pg	Nxt pg ×2	Prv pg	Prv pg ×2	/	Open photo (1st)	Like/ unlike (1st)	Open photo (2nd)	Like/ unlike (2nd)	Open photo (3rd)	Like/ unlike (3rd)
	Photo	Zoom in	Zoom out	/	/	/	/	Rtn	/	/	/	/	/	/
Ebook Reader	**Gallery**	/	/	Nxt pg	Nxt pg ×2	Prv pg	Prv pg ×2	/	Open book (1st)	Like/ unlike (1st)	Open book (2nd)	Like/ unlike (2nd)	Open book (3rd)	Like/ unlike (3rd)
	Reader	↑ font size	↓ font size	Nxt pg	Nxt pg ×2	Prv pg	Prv pg ×2	Rtn	Previous page		/		Next page	

Zm: zoom; Sld: slide; Prs: press; Tp: tap; Nxt: next; Prv:previous; Pg: page; Rtn: return.

about the participants' overall impression of RimSense, perceived benefits and drawbacks, and suggestions for future development of the system.

The deep learning model utilized in this user study is trained with data collected in the data collection sessions discussed in Section 5.2, which occurred approximately one month prior to the user study. Notably, some participants in the user study have also participated in data collection[4], and for those individuals, their own LOSO models are utilized. For the remaining participants, the model trained using all data in the dataset is employed. Participants are encouraged to ask the researcher any questions they may have throughout the study, which lasts approximately 20 minutes for each subject.

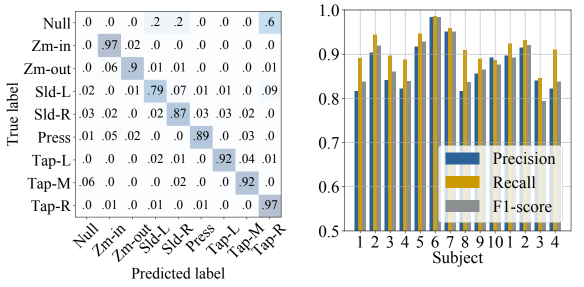
6.3 Gesture Recognition Accuracy

In this section, we present an evaluation of RimSense's gesture recognition performance in real-world app usage, which complements the previous evaluation in Section 5.2. While the previous evaluation provides a controlled environment for testing where a single gesture is repeated multiple times and similar gestures are grouped together (see Section 5.1), this evaluation aims to assess RimSense's performance in a more naturalistic setting where users perform gestures randomly. As discussed above, we use a camera to record the gesture ground truth during the user study. We compare RimSense's predictions with the ground truth and evaluate the gesture recognition accuracy. The true-positive samples, the false-positive samples, and the false-negative samples are defined in Section 5.2.2.

We have collected 1716 samples in total, among which the numbers of samples for each gesture are 219, 283, 407, 262, 256, 97, 84, and 103, respectively, and one participant contributes to around 120 samples. Figure 19(a) depicts the confusion matrix of each gesture. The average precision, recall, and F1 score for the eight gestures are 0.89, 0.88, and 0.88. Figure 19(b) shows the performance on each subject. The average precision, recall, and F1-score among all the subjects are 0.88, 0.92, and 0.88, respectively.

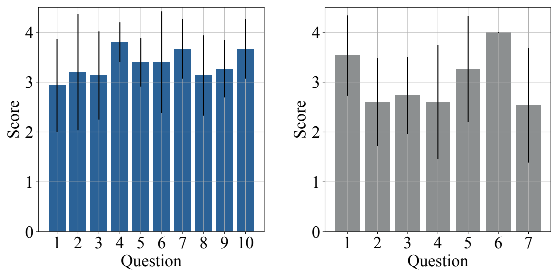
Comparing this result to the result in Section 5.2.2, we see a 7% performance degradation in terms of average F1 score, which we attribute to users being more focused on app interaction than on performing the gestures correctly. We discuss potential solutions to this issue in Section 7, including model adaptation and few-shot learning techniques that can be employed to customize the model for individual users and improve accuracy.

[4] In Section 6, the participants P1, P2, P3, P4, P5, P6, P7, P8, P9 are the same as participants P7, P6, P20, P19, P9, P12, P13, P11 and P14 in Section 5



(a) Confusion matrix. (b) Performance across subjects

Fig. 19. RimSense performance in the user study. (Zm: zoom; Sld: slide; L: left; M: mid; R: right)



(a) SUS questionnaire. (b) Customized questionnaire.

Fig. 20. The questionnaire results (4 is the highest score)

.

6.4 User Experience Study

We use two questionnaires to collect the user's feedback on their experience of using RimSense. One questionnaire is the SUS questionnaire which is widely used to evaluate the system's usability (Appendix A). The result of the questionnaire can be converted to a score ranging from 0 to 100, representing the overall usability [16]. Apart from the standard SUS questionnaire, we are also interested in other issues that are specific to the user experience of RimSense. We ask the following questions in our second questionnaire. The design of these questions is based on previous works [33, 63, 64, 69, 79].

1. Enjoyment. "I feel it is interesting to interact with apps using RimSense."
2. Accuracy. "I believe RimSense is accurate in reading my gesture input."
3. Efficiency. "I think RimSense is an efficient way to interact with apps."
4. Fatigue. "I don't think using RimSense makes me tired."
5. Social concerns. "I don't think using RimSense in public spaces would raise my social concern."
6. Distraction. "I don't think RimSense is producing any noise or vibrations."
7. Vision blockage. "I don't think the sensors on the glass blocked my vision when I was using RimSense."

All the above questions are answered by a 5-point scale score for their agreement with the above statements where the lowest score means strongly disagree and the highest score means strongly agree. The 5-point scale is used to make the second questionnaire consistent with the SUS questionnaire.

6.4.1 Questionnaire Results. The average scores of each question in the SUS questionnaire are shown in Figure 20(a). Following the score calculation method introduced in [30], we obtain the overall system usability (SU) value of 84.00 ±10.09 out of 100. This result indicates the high usability of RimSense. Specifically, in the SUS questionnaire, Q4 and Q10 are for learnability and the rest are for usability. By considering these two categories separately, we obtain the learnability score of 93.3 (±8.98 and the usability score of 81.67 (±12.80). This result implies that RimSense is generally easy to use and learn.

For the second questionnaire, the average score for each category is depicted in Figure 20(b). For the question of enjoyment, 10 out of 14 participants gave the highest score in this category indicating the participants generally feel RimSense is fun to use. For the accuracy category, while 9 participants agreed that the system is accurate, 4 participants gave a neutral score and 2 participants disagreed. This means that although the average accuracy is high according to the evaluation above, the performance of RimSense is not perceived as good enough for some participants. As the evaluation in Section 6.3 and users' subjective feedback in Section 6.4.2 suggest, the reason might be that the accuracy of the slide gestures is not optimized. As discussed Section 7, future work can leverage transfer learning and few-shot learning techniques to adapt the gesture recognition model to each user for boosting performance. As for the question of efficiency, 13 out of 14 participants agreed that RimSense is an efficient way to interact with the apps. We believe this should be the benefit of the direct manipulation user experience enabled by RimSense. For the question of fatigue, 2 participants reported feeling fatigued after using RimSense (strongly disagree or disagree) and 5 participants stayed neutral. This is probably because of the nature of smart eyewear - a user needs to raise an arm in order to reach the eyeglass. It is also possible that our designed apps require frequent input commands (turn pages). Therefore, future works and future app designs based on RimSense should consider this and functionalities that require intense gesture input are not recommended. As for the question of social concern, although most participants would feel comfortable using RimSense in public, 2 participants disagreed. We believe this is an inherent problem of smart eyewear's form factor, and it may take some time for the public to fully accept this form of interaction. For the question of distraction, all participants reported no noise or vibration (strongly agree or agree) produced by RimSense. This result proves the effectiveness of our probe signal design discussed in Section 3.1. Regarding the question of vision blockage, 3 participants reported that the sensors on the eyeglass block the vision (strongly disagree or disagree), and 3 participants remained neutral on this question. This result indicates that some users may feel that the sensor placement on the glass will influence their eyesight. We will discuss this issue in Section 7, and future work can explore using advanced transparent PZT sensors [58, 59] or integrating the sensors into fashion designs to ease the perceptibility of the sensors.

6.4.2 Subjective Feedback. In addition to the questionnaires, we ask the participants to provide comments about RimSense. We summarize their comments as follows.

Enjoyment. Some participants explicitly commented on their excitement about using the system. *"I felt excited when using the eyewear. I think it is absolutely interesting and fun to play with."* (P2). *"This is a new way of interacting with wearable devices. It is interesting and I have never seen it before. I would recommend my colleagues and friends to try it out."* (P7). *"I think this interaction is new, novel, and I find it very interesting."* (P10)

Reasonableness. Some participants acknowledged the usefulness and reasonableness of this design. *"Opening a photo by clicking on the corresponding location of the glass is really convenient. I think it is a reasonable design."* (P9). *"I think it makes a lot of sense to interact with your eyewear like that. I would picture myself lying on my couch watching TV, and switching channels by just touching my eyeglass without looking for my remote control everywhere."* (P1) .*"The gestures are easy to understand and make sense to me. Therefore, I think it is easy for me to learn all the gestures." (P14)*

Intuitiveness. The participants felt that most input gestures are intuitive and easy to remember. *"It is intuitive to perform a zoom-in or zoom-out gesture on the glass and see the content on the screen is also zoomed in or out"* (P8). *"It felt very smooth to turn pages by sliding on the glass."* (P4). *"I think the slide, tap and zoom gestures are quite easy to remember because they are natural.* (P7). However, the press gesture (which means a return) is not very intuitive to some participants. *"I always forgot a press gesture means a return."* (P6) *"Sliding is the standard way of returning to the previous page in Android and iOS smartphones, so I think the press-to-return design is somehow contrary to common practice."* (P2).

Robustness. The participants generally felt RimSense is accurate and robust. *"I am satisfied with the system's accuracy. Also, the robustness is good. I particularly tried gestures with different speeds, strengths and different touching locations, and most gestures are accurately recognized."* (P9). *"I think the system is smooth to use, and almost all my gestures are recognized correctly."* (P7). *"I'm satisfied with the general accuracy of the system, especially for the tap gestures, they are very accurate."* (P11). P14 also gave very similar feedback. However, some participants raised concerns about the robustness of some gestures. *"The recognition accuracy for the sliding gestures is not as* *good as the others."* (P8). P4 also has similar comments: *"Sometimes a slide-left gesture is recognized as tap-right, so sometimes I was intended to turn to the previous page but the photo or book on the right is opened instead."*

Latency. Two of the participants feel the latency of the system (currently $5Hz$ refresh rate) needs to be improved. *"There is a perceivable delay before the input gesture takes effect, so there is a gap between RimSense's refresh rate and the 90 or* $120Hz$ *refresh rate that we currently enjoy with modern smartphones."* (P4) *"I felt the system is a little laggy to me if I'm going to use it daily."* (P9).

Some participants also gave us suggestions about what they expect to see in the future generation of RimSense. These comments could shed light on the future directions of this study.

Micro gestures. P1, P3, and P9 commented that they would like to see micro gestures to be enabled in the future design. Specifically, P9 said: *"The zoom gestures are large gestures in a sense that it takes me around one second to finish. I would recommend changing it to a single-finger gesture that gently slides on the upper-right corner."*

Finger tracking. P2 and P4 suggested that finger touch tracking on the rim would be a nice feature to have in the future product. *"In addition to touch gesture recognition on the rim, I would recommend finger tracking because it would enable gestures with finer granularity."* (P4). *"Finger tracking would be a nice feature to have since it would introduce gesture path as another dimension to a gesture in addition to gesture duration. For example, a full-slide gesture that covers the entire upper edge of the glass and a half-slide gesture that goes from the upper-left or right corner to the middle can be different gestures."* (P2).

7 DISCUSSION

In this section, we discuss the limitations of RimSense and point out the future directions of this study.

Unintended touch. This study aims to detect and recognize touch gestures on the eyeglass rim. However, sometimes users can touch the rim unintentionally. For example, users may touch the eyeglass to adjust the eyeglass position, and this may lead to a false positive gesture prediction. This case is not considered in this study. Future studies can consider collecting unintentional touch samples and labelling them as null samples. Also, data augmentation techniques introduced in [73] can be used to increase the number of this kind of null samples. In addition, a specially designed gesture (*e.g.*, double tap) can serve as the keyword to initiate an interaction session, where only gestures detected within this session would be recognized and others ignored.

Personalization. As discussed in the user study (Section 6), RimSense's performance on some subjects is not as good as the others. This can be solved by introducing a personalization approach where a user needs to provide a few samples of gestures, and the samples are used to adapt the model to the user via transfer learning [71]. This initialization step is also common in commercial products‘ gesture input systems [3]. Also, since we only build one prototype in this study, leveraging personalization techniques can help to adapt the deep learning model to new eyewear models with PZT sensors attached.

Form factor. The current implementation of RimSense relies on an audio interface card and a laptop to run the software. Future research needs to focus on how to integrate the entire system into a single pair of eyeglasses, which would introduce new implementation challenges because of the limited computing resources and power constraints. This can potentially be solved by the recent research progress on TinyML [41]. Also, as we discuss in Section 6, some users find it annoying to have two sensors attached to the glass blocking their eyesight. Therefore, future implementation could consider adopting smaller or even transparent PZT sensors [58, 59] with a more compatible shape and fashionable design to the eyeglasses to help the users ignore the existence of the sensors. Additionally, considering VR headsets are the most popular form of smart eyewear, future research could explore how RimSense can improve the interaction experience for VR headsets, say, making the exterior of the VR headset interactable using the techniques behind RimSense.

Robustness. As discussed in Section 5.1 and Section 6.2, the majority of participants recruited in our evaluation are young adults. However, it is also important to evaluate RimSense on other age groups, such as children and the elderly. Also, all the current evaluations are conducted in a lab setting where the participants are seated. Future work should also consider the situation when the users are in motion.

Comparison with de facto method. RimSense serves as a proof-of-concept interaction design to showcase the possibilities and potential of utilizing the eyeglass rim for interaction purposes. Future work should optimize the hardware and software design and, most importantly, comprehensively compare the performance and practicability with the de facto interaction method - using a touch panel on the temple front of the eyeglass.

8 RELATED WORK

In this section, we give an overview of the related work of this study. We group the related works into two categories: eyewear interaction systems and PZT-based sensing systems.

8.1 Eyewear Interaction Systems

We separately discuss the surveyed works by the designed interfaces including the gaze-eyewear interface, the face-eyewear interface, the head-eyewear interface, the hand-face interface, and the hand-eyewear interface.

8.1.1 Gaze-Eyewear Interface. Gaze gestures are suitable for eyewear control for the eye's close proximity to the eyeglass. Jalaliniya and Mardanbegi [31] propose to interact with scrolling content by detecting eye movement patterns using an eye-tracking camera. Ahn and Lee [12] propose a text entry method for smart eyewear that involves eye-tracking techniques and a touchpad. In [25, 37, 38, 47, 65], the authors propose to use eye expressions as input gestures to interact with smart eyewear. One common issue for the above gaze-based interaction systems is that most of the designs require an eye-facing camera which will raise privacy concerns. In addition, this method requires complete visual attention of the user, impeding immersive experiences. Also, since these systems require the user to adjust gaze direction in specific ways, using these systems may raise social concerns.

8.1.2 Face-Eyewear Interface. Researchers are also interested in developing interaction methods that use facial expressions for input gestures. In [28, 46, 49, 60, 71], the authors propose to use facial actions (mostly the actions around the eye area) for interacting with eyewear. In addition, Yuya *et al.* [29] sense silent speech so that users can say key works silently to control the eyewear. Compared with the gaze-eyewear interface, using facial actions for interaction does not require a camera. However, performing specific facial expressions in public may still raise social concerns. Furthermore, extensive use of these facial actions for interaction might be exhausting.

8.1.3 Head-Eyewear Interface. Head motions are also explored to control smart eyewear. HeadTurn [53] proposes to detect the head orientation as the input for eyewear. Authors in [75, 77] propose to use the eyewear-embedded IMU to sense head gestures as the eyewear input. SmoothMoves [22] designs a target selection method powered by IMU-measured head orientation patterns. Similar to gaze-eyewear and face-eyewear interaction, using head gestures to control eyewear is not well-accepted by the public. Therefore, the head-eyewear interface faces the challenge of social concerns. Also, the interference of unintended body motion might be another limitation of the head-eyewear interface.

8.1.4 Hand-Face Interface. The facial area acts like a natural control pad for the eyewear. Lee *et al.* [39] propose an eyewear system called Ithcy Nose that senses finger gestures on the nose. Papers [48, 69, 74] explore to detect finger-to-face gestures as input vocabulary for eyewear. Compared with the gaze-eyewear and head-eyewear interfaces, one advantage of the hand-face interface is its tangibility. However, since the hand-face interface involves touching or even pinching facial skin, the social acceptability of this interface needs further study. Also, there might be hygiene concerns regarding this interaction modality.

8.1.5 Hand-Eyewear Interface. Directly interacting with the eyewear itself is the de facto interaction method for commercial eyewear products [2, 4, 7–10]. These products leverage a touch panel at the temple front of the eyeglass for receiving gesture input. In addition to these commercial products, the research community also proposed new touch-based interaction methods for eyewear. FaceWidgets [66] proposes a panel of physical controls located at the back side of the VR headset to enable tangible interactions. StretchAR [55] presents a wearable stretch that is interactable with finger touching. Xie *et al.* [70] present an eyeglass prototype that can recognize five finger-touching locations for interaction. Different from the above products and research works, RimSense explores a new interaction space — the eyeglass rim — for interaction with smart glasses. In addition, our user study shows that interacting on the rim is natural and intuitive. In addition to touch-based gestures, mid-air gestures are also explored to facilitate eyewear interaction. Authors in [15, 17–19, 34, 42, 76] propose to use cameras to recognize mid-air hand gestures. Despite the convenience and intuitiveness of the use of mid-air gestures, the widely used camera of these systems may raise privacy and social concerns. Also, this modality lacks tangibility compared with touch-based systems such as RimSense.

8.2 PZT-based Sensing Systems

A PZT sensor is a device that can transduce between the vibration signal and electrical signal. It is widely used in civil engineering for structural health monitoring [24, 43, 50]. More recently, researchers started to design human-centric sensing systems using PZT sensors. Stane [51] is a PZT sensor-enabled tactile input device. Similarly, Fan *et al.* [23] develop a tactile surface for robots using PZT sensors for human-robot interaction. Ono *et al.* [54] propose to make daily objects interactable by attaching a pair of PZT sensors. V-Speech [45] presents an eyewear prototype with a PZT sensor attached to the nose pad for voice capturing. Kalantatian *et al.* [32] embed a PZT sensor into a necklace to sense the user's eating habits. In addition, there is a rich body of work that uses PZT sensors to detect human vital signs such as respiration rate and heart rate [11, 13, 14, 44, 56, 78].

9 CONCLUSION

This paper introduces RimSense, a proof-of-concept approach for interacting with smart eyewear by making the eyeglass rim interactive through piezoelectric (PZT) transducers. RimSense incorporates a frequency-buffered chirp signal to ensure optimal sensing granularity and minimal impulse noise. Additionally, RimSense integrates a timestep-level deep learning-based prediction model and an event-level FSM-based prediction algorithm, facilitating the real-time recognition of gestures with varying durations. Our implemented prototype demonstrates RimSense's ability to recognize eight distinct touch gestures executed on the eyeglass rim, including zoom-in, zoom-out, slide-left, slide-right, press, tap-left, tap-mid, and tap-right. The system undergoes comprehensive evaluation involving 30 subjects, achieving an F1-score of 0.95 for gesture recognition and an 11% relative gesture duration estimation error. A user study involving 14 subjects further validates RimSense's good performance, high usability, learnability and enjoyability. To the best of our knowledge, this study is the first to showcase the feasibility of utilizing the eyeglass rim as a new interaction space for smart eyewear.

ACKNOWLEDGMENTS

The authors would like to thank the Turing AI Computing Cloud (TACC) [72] and HKUST iSING Lab for the computation resources and platform. This research is supported in part by Hong Kong RGC under Contract CERG 16204820, 16206122, AoE/E-601/22-R, Contract R8015, and 3030_006, in part by Shenzhen Science and Technology Program with grant ZDSYS20210623092007023. Jin Zhang and Qian Zhang are the corresponding authors.

REFERENCES

[1] 2022. AR Glasses Market Size, Trends, Growth, Industry Analysis 2025. https://www.fairfieldmarketresearch.com/report/ar-glasses-market Accessed Mar 13, 2023.

[2] 2023. Augmented Reality and Mixed Reality | by MOVERIO | Epson.com | Epson US. https://epson.com/moverio-augmented-reality Accessed Mar 10, 2022.

[3] 2023. Buy Apple Watch Series 8. https://www.apple.com/shop/buy-watch/apple-watch Accessed Mar 10, 2022.

[4] 2023. Glass. https://www.google.com/glass/start/ Accessed Mar 13, 2022.

[5] 2023. Home | PUI Audio. https://puiaudio.com/ Accessed Mar 10, 2022.

[6] 2023. Homepage | Focusrite. https://focusrite.com/en Accessed Mar 10, 2022.

[7] 2023. Iristick - Smart glasses built for every industry. https://iristick.com/ Accessed Mar 10, 2022.

[8] 2023. Rokid Glass 2 | Everyday AR Glasses Build for Enterprises. https://rokid.ai/products/rokid-glass-2/ Accessed Mar 10, 2022.

[9] 2023. Smart Glasses by solos® | Personalize your Audio & Style with AirGo™ 2. https://solosglasses.com/ Accessed Mar 10, 2022.

[10] 2023. Vuzix | Heads-Up, Hands-Free AR Smart Glasses. https://www.vuzix.com/ Accessed Mar 10, 2022.

[11] Mahmoud Al Ahmad. 2016. Piezoelectric extraction of ECG signal. *Scientific Reports* 6 (Nov. 2016), 37093.

[12] Sunggeun Ahn and Geehyuk Lee. 2019. Gaze-Assisted Typing for Smart Glasses. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 857–869.

[13] Mahmoud Al Ahmad and Soha Ahmed. 2017. Heart-rate and pressure-rate determination using piezoelectric sensor from the neck. 1–5.

[14] Areen Allataifeh and Mahmoud Al Ahmad. 2020. Simultaneous piezoelectric noninvasive detection of multiple vital signs. *Scientific Reports* 10, 1 (Jan. 2020), 416.

[15] Huidong Bai, Gun Lee, and Mark Billinghurst. 2014. Using 3D hand gestures and touch input for wearable AR interaction. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. Association for Computing Machinery, New York, NY, USA, 1321–1326.

[16] Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An Empirical Evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction* 24, 6 (July 2008), 574–594.

[17] Mayra D. Barrera-Machuca, Alvaro Cassinelli, and Christian Sandor. 2020. Context-Based 3D Grids for Augmented Reality User Interfaces. In *Adjunct Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 73–76.

[18] Eugenie Brasier, Olivier Chapuis, Nicolas Ferey, Jeanne Vezien, and Caroline Appert. 2020. ARPads: Mid-air Indirect Input for Augmented Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 332–343. ISSN: 1554-7868.

[19] Han Joo Chae, Jeong-in Hwang, and Jinwook Seo. 2018. Wall-based Space Manipulation Technique for Efficient Placement of Distant Objects in Augmented Reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 45–52.

[20] Andrea Colaço, Ahmed Kirmani, Hye Soo Yang, Nan-Wei Gong, Chris Schmandt, and Vivek K. Goyal. 2013. Mime: compact, low power 3D gesture sensing for interaction with head mounted displays. In *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13)*. Association for Computing Machinery, New York, NY, USA, 227–236.

[21] Ramen Dutta, Andre B. J. Kokkeler, Ronan v. d. Zee, and Mark J. Bentum. 2011. Performance of chirped-FSK and chirped-PSK in the presence of partial-band interference. In *2011 18th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*. 1–6.

[22] Augusto Esteves, David Verweij, Liza Suraiya, Rasel Islam, Youryang Lee, and Ian Oakley. 2017. SmoothMoves: Smooth Pursuits Head Movements for Augmented Reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 167–178.

[23] Xiaoran Fan, Daewon Lee, Larry Jackel, Richard Howard, Daniel Lee, and Volkan Isler. 2022. Enabling Low-Cost Full Surface Tactile Skin for Human Robot Interaction. *IEEE Robotics and Automation Letters* 7, 2 (April 2022), 1800–1807. Conference Name: IEEE Robotics and Automation Letters.

[24] Ehsan Ghafari, Ying Yuan, Chen Wu, Tommy Nantung, and Na Lu. 2018. Evaluation the compressive strength of the cement paste blended with supplementary cementitious materials using a piezoelectric-based sensor. *Construction and Building Materials* 171 (May 2018), 504–510.

[25] Philip Graybill and Mehdi Kiani. 2019. Eyelid Drive System: An Assistive Technology Employing Inductive Sensing of Eyelid Movement. *IEEE Transactions on Biomedical Circuits and Systems* 13, 1 (Feb. 2019), 203–213.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[27] Gunther Heidemann, Ingo Bax, and Holger Bekel. 2004. Multimodal interaction in an augmented reality scenario. In *Proceedings of the 6th international conference on Multimodal interfaces (ICMI '04)*. Association for Computing Machinery, New York, NY, USA, 53–60.

[28] Steven Hickson, Nick Dufour, Avneesh Sud, Vivek Kwatra, and Irfan Essa. 2019. Eyemotion: Classifying Facial Expressions in VR Using Eye-Tracking Cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa Village, HI, USA, 1626–1635.

[29] Yuya Igarashi, Kyosuke Futami, and Kazuya Murao. 2022. Silent Speech Eyewear Interface: Silent Speech Recognition Method using Eyewear with Infrared Distance Sensors. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers (ISWC '22)*.

Association for Computing Machinery, New York, NY, USA, 33–38.

[30] Brian Kenji Iwana and Seiichi Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *PLOS ONE* 16, 7 (July 2021), e0254841. Publisher: Public Library of Science.

[31] Shahram Jalaliniya and Diako Mardanbegi. 2016. Seamless interaction with scrolling contents on eyewear computers using optokinetic nystagmus eye movements. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. ACM, Charleston South Carolina, 295–298.

[32] Haik Kalantarian, Nabil Alshurafa, Tuan Le, and Majid Sarrafzadeh. 2015. Monitoring eating habits using a piezoelectric sensor-based necklace. *Computers in Biology and Medicine* 58 (March 2015), 46–55.

[33] Daehwa Kim, Keunwoo Park, and Geehyuk Lee. 2021. AtaTouch: Robust Finger Pinch Detection for a VR Controller Using RF Return Loss. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–9.

[34] Myung Jin Kim and Andrea Bianchi. 2021. Exploring Pseudo Hand-Eye Interaction on the Head-Mounted Display. In *Proceedings of the Augmented Humans International Conference 2021 (AHs '21)*. Association for Computing Machinery, New York, NY, USA, 251–258.

[35] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[36] Pin-Sung Ku, Qijia Shao, Te-Yen Wu, Jun Gong, Ziyan Zhu, Xia Zhou, and Xing-Dong Yang. 2020. ThreadSense: Locating Touch on an Extremely Thin Interactive Thread. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12.

[37] Pin-Sung Ku, Te-Yan Wu, and Mike Y. Chen. 2017. EyeExpression: exploring the use of eye expressions as hands-free input for virtual and augmented reality devices. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. ACM, Gothenburg Sweden, 1–2.

[38] Pin-Sung Ku, Te-Yen Wu, and Mike Y. Chen. 2018. EyeExpress: Expanding Hands-free Input Vocabulary using Eye Expressions. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. ACM, Berlin Germany, 126–127.

[39] Juyoung Lee, Hui-Shyong Yeo, Murtaza Dhuliawala, Jedidiah Akano, Junichi Shimizu, Thad Starner, Aaron Quigley, Woontack Woo, and Kai Kunze. 2017. Itchy nose: discreet gesture interaction using EOG sensors in smart eyewear. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers (ISWC '17)*. Association for Computing Machinery, New York, NY, USA, 94–97.

[40] Jansen C. Liando, Amalinda Gamage, Agustinus W. Tengourtius, and Mo Li. 2019. Known and Unknown Facts of LoRa: Experiences from a Large-Scale Measurement Study. *ACM Trans. Sen. Netw.* 15, 2, Article 16 (feb 2019), 35 pages.

[41] Ji Lin, Ligeng Zhu, Wei-Ming Chen, Wei-Chen Wang, Chuang Gan, and Song Han. 2022. On-Device Training Under 256KB Memory. arXiv:2206.15472 [cs.CV]

[42] Sikun Lin, Hao Fei Cheng, Weikai Li, Zhanpeng Huang, Pan Hui, and Christoph Peylo. 2017. Ubii: Physical World Interaction Through Augmented Reality. *IEEE Transactions on Mobile Computing* 16, 3 (March 2017), 872–885. Conference Name: IEEE Transactions on Mobile Computing.

[43] Peng Liu, Weilun Wang, Ying Chen, Xing Feng, and Lixin Miao. 2017. Concrete damage diagnosis using electromechanical impedance technique. *Construction and Building Materials* 136 (April 2017), 450–455.

[44] Ifana Mahbub, Salvatore Andrea Pullano, Hanfeng Wang, Syed Kamrul Islam, Antonino S. Fiorillo, Gary To, and M. R. Mahfouz. 2017. A Low-Power Wireless Piezoelectric Sensor-Based Respiration Monitoring System Realized in CMOS Process. *IEEE Sensors Journal* 17, 6 (March 2017), 1858–1864. Conference Name: IEEE Sensors Journal.

[45] Héctor A. Cordourier Maruri, Paulo Lopez-Meyer, Jonathan Huang, Willem Marco Beltman, Lama Nachman, and Hong Lu. 2018. V-Speech: Noise-Robust Speech Capturing Glasses Using Vibration Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (Dec. 2018), 180:1–180:23.

[46] Katsutoshi Masai, Kai Kunze, Daisuke Sakamoto, Yuta Sugiura, and Maki Sugimoto. 2020. Face Commands - User-Defined Facial Gestures for Smart Glasses. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 374–386. ISSN: 1554-7868.

[47] Katsutoshi Masai, Kai Kunze, and Maki Sugimoto. 2020. Eye-based Interaction Using Embedded Optical Sensors on an Eyewear Device for Facial Expression Recognition. In *Proceedings of the Augmented Humans International Conference (AHs '20)*. Association for Computing Machinery, New York, NY, USA, 1–10.

[48] Katsutoshi Masai, Yuta Sugiura, and Maki Sugimoto. 2018. FaceRubbing: Input Technique by Rubbing Face using Optical Sensors on Smart Eyewear for Facial Expression Recognition. In *Proceedings of the 9th Augmented Human International Conference (AH '18)*. Association for Computing Machinery, New York, NY, USA, 1–5.

[49] Denys J.C. Matthies, Alex Woodall, and Bodo Urban. 2021. Prototyping Smart Eyewear with Capacitive Sensing for Facial and Head Gesture Detection. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. ACM, Virtual USA, 476–480.

[50] Jiyoung Min, Seunghee Park, Chung-Bang Yun, Chang-Geun Lee, and Changgil Lee. 2012. Impedance-based structural health monitoring incorporating neural network technique for identification of damage type and severity. *Engineering Structures* 39 (June 2012), 210–220.

[51] Roderick Murray-Smith, John Williamson, Stephen Hughes, and Torben Quaade. 2008. Stane: synthesized surfaces for tactile input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1299–1302.

[52] Donald A. Norman. 2002. *The design of everyday things.* Basic Books, [New York].

[53] Tomi Nukarinen, Jari Kangas, Oleg Špakov, Poika Isokoski, Deepak Akkil, Jussi Rantala, and Roope Raisamo. 2016. Evaluation of HeadTurn: An Interaction Technique Using the Gaze and Head Turns. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16).* Association for Computing Machinery, New York, NY, USA, 1–8.

[54] Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. 2013. Touch &amp; activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13).* Association for Computing Machinery, New York, NY, USA, 31–40.

[55] Luis Paredes, Ananya Ipsita, Juan C. Mesa, Ramses V. Martinez Garrido, and Karthik Ramani. 2022. StretchAR: Exploiting Touch and Stretch as a Method of Interaction for Smart Glasses Using Wearable Straps. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (Sept. 2022), 134:1–134:26.

[56] Dae Yong Park, Daniel J. Joe, Dong Hyun Kim, Hyewon Park, Jae Hyun Han, Chang Kyu Jeong, Hyelim Park, Jung Gyu Park, Boyoung Joung, and Keon Jae Lee. 2017. Self-Powered Real-Time Arterial Pulse Monitoring Using Ultrathin Epidermal Piezoelectric Sensors. *Advanced Materials* 29, 37 (2017), 1702308.

[57] Gyuhae Park and Daniel J Inman. 2006. Structural health monitoring using piezoelectric impedance measurements. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365, 1851 (Dec. 2006), 373–392. Publisher: Royal Society.

[58] Chaorui Qiu, Bo Wang, Nan Zhang, Shujun Zhang, Jinfeng Liu, David Walker, Yu Wang, Hao Tian, Thomas R Shrout, Zhuo Xu, et al. 2020. Transparent ferroelectric crystals with ultrahigh piezoelectricity. *Nature* 577, 7790 (2020), 350–354.

[59] Tiago Rodrigues-Marinho, Nelson Pereira, Vitor Correia, Daniel Miranda, Senentxu Lanceros-Méndez, and Pedro Costa. 2022. Transparent Piezoelectric Polymer-Based Materials for Energy Harvesting and Multitouch Detection Devices. *ACS Applied Electronic Materials* 4, 1 (2022), 287–296.

[60] Soha Rostaminia, Alexander Lamson, Subhransu Maji, Tauhidur Rahman, and Deepak Ganesan. 2019. W!NCE: Unobtrusive Sensing of Upper Facial Action Units with EOG-based Eyewear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (March 2019), 1–26.

[61] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, Austin Texas USA, 483–492.

[62] Munehiko Sato, Rohan S. Puri, Alex Olwal, Yosuke Ushigome, Lukas Franciszkiewicz, Deepak Chandra, Ivan Poupyrev, and Ramesh Raskar. 2017. Zensei: Embedded, Multi-electrode Bioimpedance Sensing for Implicit, Ubiquitous User Recognition. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* ACM, Denver Colorado USA, 3972–3985.

[63] Chen-Hsuan (Iris) Shih, Naofumi Tomita, Yanick X. Lukic, Álvaro Hernández Reguera, Elgar Fleisch, and Tobias Kowatsch. 2019. Breeze: Smartphone-based Acoustic Real-time Detection of Breathing Phases for a Gamified Biofeedback Breathing Training. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (Dec. 2019), 1–30.

[64] Jaemin Shin, Seungjoo Lee, Taesik Gong, Hyungjun Yoon, Hyunchul Roh, Andrea Bianchi, and Sung-Ju Lee. 2022. MyDJ: Sensing Food Intakes with an Attachable on Your Eyeglass Frame. In *CHI Conference on Human Factors in Computing Systems.* ACM, New Orleans LA USA, 1–17.

[65] Asfand Tanwear, Xiangpeng Liang, Yuchi Liu, Aleksandra Vuckovic, Rami Ghannam, Tim Böhnert, Elvira Paz, Paulo P. Freitas, Ricardo Ferreira, and Hadi Heidari. 2020. Spintronic Sensors Based on Magnetic Tunnel Junctions for Wireless Eye Movement Gesture Control. *IEEE Transactions on Biomedical Circuits and Systems* 14, 6 (Dec. 2020), 1299–1310.

[66] Wen-Jie Tseng, Li-Yang Wang, and Liwei Chan. 2019. FaceWidgets: Exploring Tangible Interaction on Face with Head-Mounted Displays. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19).* Association for Computing Machinery, New York, NY, USA, 417–427.

[67] Yu-Chih Tung and Kang G. Shin. 2015. EchoTag: Accurate Infrastructure-Free Indoor Location Tagging with Smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking - MobiCom '15.* ACM Press, Paris, France, 525–536.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[69] Yueting Weng, Chun Yu, Yingtian Shi, Yuhang Zhao, Yukang Yan, and Yuanchun Shi. 2021. FaceSight: Enabling Hand-to-Face Gesture Interaction on AR Glasses with a Downward-Facing Camera Vision. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21).* Association for Computing Machinery, New York, NY, USA, 1–14.

[70] Wentao Xie, Jin Zhang, and Qian Zhang. 2022. Transforming eyeglass rim into touch panel using piezoelectric sensors. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking.* ACM, Sydney NSW Australia, 838–840.

[71] Wentao Xie, Qian Zhang, and Jin Zhang. 2021. Acoustic-based Upper Facial Action Recognition for Smart Eyewear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (June 2021), 1–28.

[72] Kaiqiang Xu, Xinchen Wan, Hao Wang, Zhenghang Ren, Xudong Liao, Decang Sun, Chaoliang Zeng, and Kai Chen. 2021. TACC: A Full-stack Cloud Computing Infrastructure for Machine Learning Tasks. *arXiv preprint arXiv:2110.01556* (2021).

[73] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. 2022. Enabling Hand Gesture Customization on Wrist-Worn Devices. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–19.

[74] Koki Yamashita, Takashi Kikuchi, Katsutoshi Masai, Maki Sugimoto, Bruce H. Thomas, and Yuta Sugiura. 2017. CheekInput: turning your cheek into an input surface by embedded optical sensors on a head-mounted display. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology (VRST '17)*. Association for Computing Machinery, New York, NY, USA, 1–8.

[75] Yukang Yan, Chun Yu, Xin Yi, and Yuanchun Shi. 2018. HeadGesture: Hands-Free Input Approach Leveraging Head Movements for HMD Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (Dec. 2018), 198:1–198:23.

[76] Hui-Shyong Yeo, Juyoung Lee, Woontack Woo, Hideki Koike, Aaron J Quigley, and Kai Kunze. 2021. JINSense: Repurposing Electrooculography Sensors on Smart Glass for Midair Gesture and Context Sensing. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, 1–6.

[77] Shanhe Yi, Zhengrui Qin, Ed Novak, Yafeng Yin, and Qun Li. 2016. GlassGesture: Exploring head gesture interface of smart glasses. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9.

[78] Zhiran Yi, Wenming Zhang, and Bin Yang. 2022. Piezoelectric approaches for wearable continuous blood pressure monitoring: a review. *Journal of Micromechanics and Microengineering* 32, 10 (Aug. 2022), 103003. Publisher: IOP Publishing.

[79] Yuzhou Zhuang, Yuntao Wang, Yukang Yan, Xuhai Xu, and Yuanchun Shi. 2021. ReflecTrack: Enabling 3D Acoustic Position Tracking Using Commodity Dual-Microphone Smartphones. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 1050–1062.

A SYSTEM USABILITY SCALE QUESTIONNAIRE

The questions in the System Usability Scale (SUS) questionnaire [16] are listed below and each question is answered using a 5-point scale ranging from strongly disagree to strongly agree.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The SUS questionnaire can be interpreted numerically. A score ranging from 0 to 4 can be assigned to each question. For questions 1, 3, 5, 7, and 9, the score is calculated as the scale position minus 1. For questions 2, 4, 6, 8, and 10, the score is calculated as 5 minus the scale position. Then the overall system usability (SU) is obtained by summing up the scores of all questions and multiplying it by 2.5. The SU has a range of 0 to 100 where 100 represents the highest usability.