FaceSight: Enabling Hand-to-Face Gesture Interaction on AR Glasses with a Downward-Facing Camera Vision

Yueting Weng
Department of Computer Science and Technology, Tsinghua University
Key Laboratory of Pervasive Computing, Ministry of Education
Beijing, China
wengyt19@mails.tsinghua.edu.cn

Chun Yu†
Department of Computer Science and Technology, Tsinghua University
Key Laboratory of Pervasive Computing, Ministry of Education
Beijing, China
chunyu@tsinghua.edu.cn

Yingtian Shi
Department of Computer Science and Technology, Tsinghua University
Key Laboratory of Pervasive Computing, Ministry of Education
Beijing, China
shiyt20@mails.tsinghua.edu.cn

Yuhang Zhao
Department of Computer Sciences, University of Wisconsin-Madison
Madison, U.S.A
yuhang.zhao@cs.wisc.edu

Yukang Yan
Department of Computer Science and Technology, Tsinghua University
Key Laboratory of Pervasive Computing, Ministry of Education
Beijing, China
yyk@mail.tsinghua.edu.cn

Yuanchun Shi
Department of Computer Science and Technology, Tsinghua University
Key Laboratory of Pervasive Computing, Ministry of Education
Beijing, China
shiyc@tsinghua.edu.cn

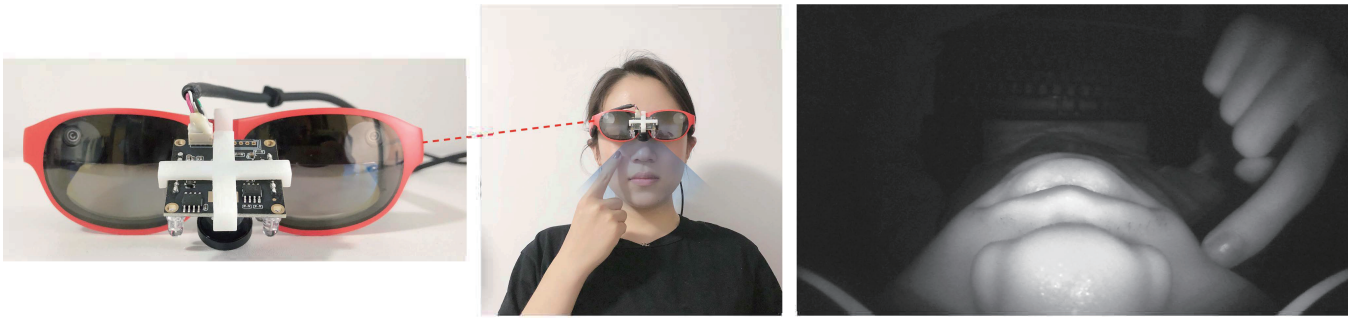

Figure 1: (a) An infrared camera fixed on the nose bridge of an AR glasses looking downward. (b) The space around the user's lower face can be captured by the camera in high resolution. (c) An image captured by the camera.

ABSTRACT

We present FaceSight, a computer vision-based hand-to-face gesture sensing technique for AR glasses. FaceSight fixes an infrared camera onto the bridge of AR glasses to provide extra sensing capability of the lower face and hand behaviors. We obtained 21 hand-to-face gestures and demonstrated the potential interaction benefits through five AR applications. We designed and implemented an algorithm pipeline that segments facial regions, detects hand-face contact (f1 score: 98.36%), and trains convolutional neural network (CNN) models to classify the hand-to-face gestures. The input features include gesture recognition, nose deformation estimation, and continuous fingertip movement. Our algorithm achieves classification accuracy of all gestures at 83.06%, proved by the data of 10 users. Due to the compact form factor and rich gestures, we recognize FaceSight as a practical solution to augment input capability of AR glasses in the future.

CCS CONCEPTS

• **Human-centered computing** → **Gestural input**.

KEYWORDS

Hand-to-Face Gestures; AR Glasses; Computer Vision

ACM Reference Format:
Yueting Weng, Chun Yu, Yingtian Shi, Yuhang Zhao, Yukang Yang, Yuanchun Shi. 2021. FaceSight: Enabling Hand-to-Face Gesture Interaction on AR

† indicates the corresponding author.

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445484

Glasses with a Downward-Facing Camera Vision. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3411764.3445484

1 INTRODUCTION

Hand-to-face gesture (e.g., tapping the cheek) interaction inherits the benefits of always-available, haptic, proprioceptive on-body input [9, 11, 13], which are also intuitive and easy to learn because of the semantic connections between facial parts to interaction tasks. [27, 50]. Face provides not only large space for interaction that benefits users in transferring touchscreen interaction to hand-to-face gesture (e.g., panning and zooming [41]), but also tactile feedback that can facilitate eyes-free interaction [54]. In this research, we focus on hand-to-face interaction on AR glasses, since they are head-worn devices that involve contact with face. Moreover, AR glasses have embedded cameras and other sensors, which have incomparable potential to sense hand gestures performed on the face. To our knowledge, prior research mainly explored electrical or audio signals to sense hand-to-face gestures [19, 29, 48–50]. However, the gestures that can be sensed via these sensing technologies are limited to simple and discrete gestures. More diverse and continuous gestures are needed to support the rich AR glasses interaction.

To fill this gap, we propose FaceSight, a computer vision-based sensing technique leveraging a downward-looking infrared camera fixed on the bridge of a pair of AR glasses to capture a user's lower face (Figure 1). The camera has six infrared lights around the camera lens as active light source. This novel placement and configuration of the camera brings three benefits. First, the user's face and the hand can be captured in high resolution images so that we can detect rich and subtle hand-to-face gestures. Second, by adjusting the luminous intensity of the infrared light source, we can illuminate only the foreground (the nose, cheeks, and the hands) with the background almost dark. This not only simplifies the computer vision process but also mitigates the privacy concerns of capturing the surrounding environment. Third, attaching a camera to the bridge of the AR glasses promises a compact form factor, which is crucial for the design of wearable devices and practical use.

In this paper, our contributions are three-folds:

First, we designed FaceSight, a novel camera-based sensing technique to enable hand-to-face interaction on AR glasses. FaceSight can sense a rich set of hand-to-face gestures with one single camera in a compact and social acceptable form factor.

Second, we presented a rich set of twenty-one hand-to-face gestures, with ten gestures being novel and not depicted in the prior literature. We also developed five example AR applications to demonstrate potential interaction techniques, which were highly accepted by users.

Third, we designed and implemented an algorithm pipeline to detect hand-face contact and recognize hand-to-face gestures.

2 RELATED WORK

The rapid advances in sensing technology and the current strong computational power in small and mobile devices have led to the emergence of on-body interaction [12], a new interaction modality where the human body was used as the input surface to better support always-available interaction for mobile computing [40]. Many researchers have explored this rich interaction space by designing and recognizing hand gestures on and around different body parts, such as palm [4, 9–11, 36, 44], arms [13, 15, 17, 33, 42], leg [23], face [29, 41, 49], nose [19, 35, 56], ear [16, 22, 30, 45], and even the hair [6]. Among the various on-body interaction techniques, hand-to-face interaction is one important interaction modality for AR glasses control [41]. In this section, we introduce prior works on sensing technology for hand-to-face interaction, as well as the different camera deployment methods on HMDs.

2.1 Hand-to-Face Interaction

2.1.1 Studies on Hand-to-Face Interaction Design. Prior research has explored the design space of hand-to-face interaction. Serrano et al. [41] first demonstrated the potential of hand-to-face interaction to support everyday mobile tasks on head-mounted displays (HMDs). They conducted a guessability study to elicit suitable hand-to-face gestures from 14 users. They found the cheeks and forehead to be good interaction surfaces. Lee et al. [18] also conducted an elicitation study to derive design guidelines for social acceptable hand-to-face input. They suggested that the input areas away from the center of the face (e.g., ear, neck) may be appropriate for hand-to-face interaction designs, and small and discrete gestures (e.g., tap) were preferred than big movements with all five fingers.

2.1.2 Sensing Technology for Hand-to-Face Interaction. Researchers have created different sensing technology to support hand-to-face interaction. One approach was to detect the deformation of a user's face to recognize gestures. For example, Itchy Nose [19] used the electrooculography (EOG) sensors on J!ns Meme glasses to recognize five different hand-to-nose gestures, such as pushing and rubbing nose. EOG sensors were embedded in the left and right nose pads, so that the system could identify the hand-to-nose gestures by recognizing the different signal patterns caused by the nose deformation. Some researchers also used optical sensors to recognize face deformation. For example, CheekInput [49] attached several photo-reflective sensors on the HMD to recognize face-pulling gestures based on how the facial skin was deformed. FaceRubbing [29] used similar sensors on smartglasses, allowing a user to rub her face at different locations to generate different input. All these technologies only recognized gestures that generated obvious face deformation, which could raise fatigue and social acceptance issues [41]. More subtle gestures, such as swiping on the face, could not be recognized with these technologies.

Some prior work also used sound-based sensing techniques to recognize hand-to-face gestures. PrivateTalk [50] detected a whisper gesture (i.e., a person covers her mouth with her hand) by comparing voice data received by two earphones when the user is speaking. This gesture was interpreted as a wake-up command to automatically activate voice input. Moreover, EarBuddy [48] used an earphone to capture the sound when a user's finger touched her face, thus recognizing different hand-to-face gestures, including tapping, double-tapping, and swiping. However, the number and type of gestures that can be recognized by sound-based sensing are still limited. For example, they only recognized discrete gestures, such as tap and double-tap. More complex gestures that involved continuous control (e.g., slider) were not supported.

As opposed to most prior sensing technology, advances in camera devices and computer vision technology had presented a unique opportunity to recognize more complex hand-to-face gestures to support richer interaction. However, little work has used camera-based technology to conduct the recognition. Only recently, Loorak et al. [24] created a technology that recognized a user's touch gestures on her face with the front-facing camera on a smartphone. However, the gestures were only limited to tapping on different locations on the user's face and were only designed to support smartphone interaction. To our knowledge, our research was the first to build camera-based technology to recognize complex hand-to-face gestures to support AR glasses interaction. The hand-to-face gesture set we could recognize was the biggest among all existing hand-to-face interaction research.

2.2 Camera Deployment Methods on HMDs

Camera has always been an important input sensor for mobile devices, such as smartphone [21, 55], watch [53] and glasses [7]. More and more smartglasses have embedded cameras to not only capture user's current activity [3] or emotion states [32] but also recognize hand gestures as input (e.g., Microsoft HoloLens). Besides the built-in cameras on HMDs, recent

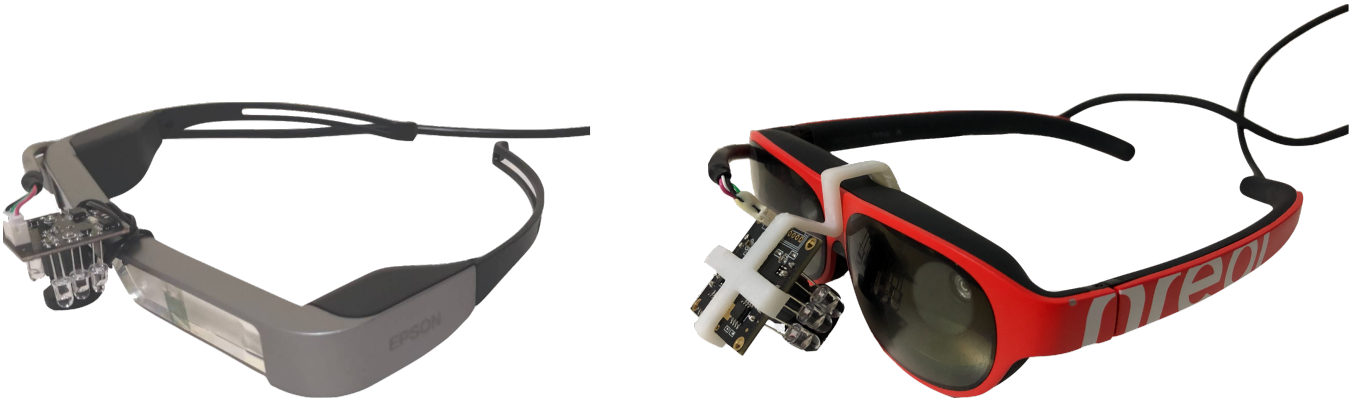

Figure 2: A wide-angle infrared camera is mounted on the frame of AR glasses, such as (a) Epson Moverio BT-300 and (b) Nreal Light AR glasses, with different camera placement to capture a user's face.

research has added extra cameras to HMDs to capture more information from the user. For example, Mo^2cap^2 [47] installed a fisheye camera on a hat, looking down to capture a user's body poses. Rhodin et al. [38] built EgoCap, a motion-capture system that had two fisheye cameras extending from a virtual reality (VR) HMD to the front of the user's face. The cameras faced down to capture the user's body. xR-EgoPose [43] also added a down-forward looking fisheye camera to an HMD to capture a user's body. However, none of the prior work captured information from the user's face. Mecap [1] was a system that could capture both the user's body gestures and mouth states. It attached a pair of hemi-spherical mirrors to the front of a Google Cardboard and used the smartphone's rare-facing camera to capture the user's body and facial information reflected in the mirrors. While the system could recognize users' mouth states, the detailed facial information it captured was still limited because the mirrors had to be put at a relatively far distance from the user's face to capture the user's full body.

Besides capturing a user's full body, some research has focused on adding cameras to only capture the user's face. For example, EgoFace [7] attached an RGB camera to the user's eyeglass frame to capture her right lateral face, so that the system could recognize the user's mouth movements and facial expressions. Li et al. [20] extended an RGB-D camera from a VR HMD to the front of the user's face to capture her mouth region. Moreover, Olszewski et al. [34] attached a monocular to the bottom of a VR HMD to capture the user's mouth movements and used an IR camera inside of the HMD to capture the user's eye movements. With this camera setup and computer vision algorithms, the system captured subtle details of the user's facial expressions to support compelling speech animation in VR.

Most prior work deployed the camera in an intrusive way by extending the camera to the front of the user's face. While these camera deployment methods can be used for VR applications, they are not suitable for AR glasses that are designed to be used on-the-go in public places. Unlike prior research, our work presented a practical approach to camera deployment by attaching a downward-looking, wide-angle infrared camera to the AR glasses. With this setup, close-up and high resolution images of users' lower face (nose to chin area) can be captured to recognize diverse and accurate hand-to-face gestures to support rich AR glasses interaction.

3 FACESIGHT HARDWARE

3.1 Proof-of-Concept Prototype

The idea of FaceSight is to mount a video camera on the nose bridge of AR glasses, as shown in figure 2. The key requirement of the add-on camera is that it should promise the compactness of the form factor, which is critical for wearable devices in practical scenarios. In this paper, we use Nreal Light

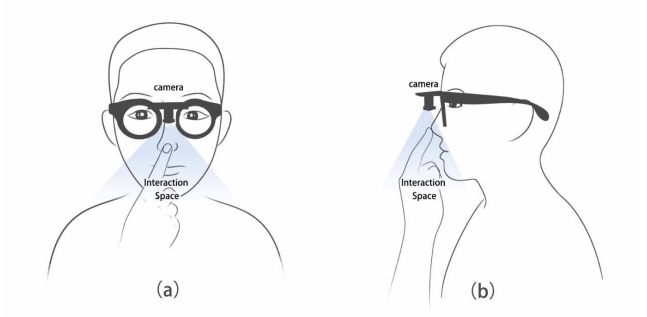

Figure 3: The field of view of the camera and the interaction space.

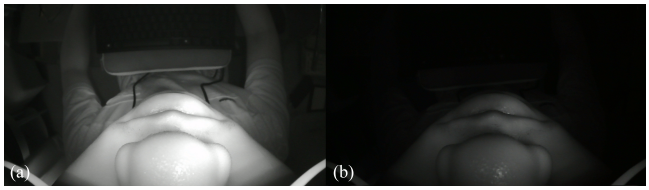

Figure 4: The high contrast between face and background, and the effects of setting (a) higher or (b) lower lighting power to the camera.

AR glasses [8] as our proof-of-concept prototype. It adopts an advanced optical module, has high rendering quality and wide FoV, which is beneficial to user experience. The Nreal Light AR glasses adopts a distributed system that is configured to distribute projection and processing functionalities among the eyewear glasses and computing unit. The eyewear glasses has a field of view of approximately 52° diagonally, and the weight is 88 grams. The operating system of the computing unit is Android 8.0, on which we can build customized applications. The two devices are connected through a USB-C cable.

We mounted a wide-angle camera (resolution: 1920 × 1080, FoV: 150° diagonally, streaming video at 30 FPS through USB cable) on the AR glasses, which looked downward to capture users' lower face. Figure 3 depicts the interaction space of FaceSight. The vertical FoV of camera is 63°, while the horizontal FoV is 125°. Such a wide FoV enables the camera to capture the users' nose, cheek, chin, mouth, and all hand behaviors on and around the

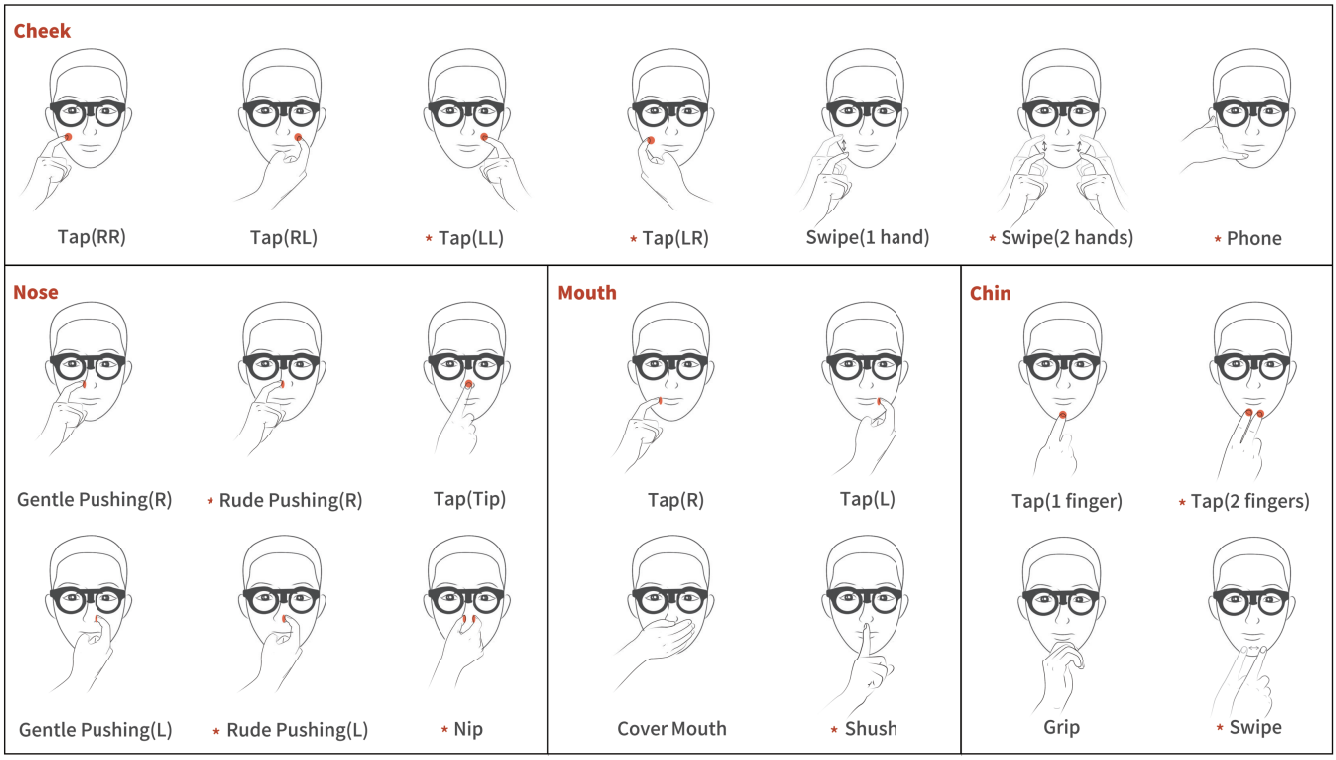

Figure 5: The 21 hand-to-face gestures proposed in this paper. Asterisks mark novel hand-to-face gestures that have not been described in prior works about hand-to-face gestural input.

face. We 3D printed a mount to hold the camera module, with a hook to fix on the device. The camera and 3D printed mount together increased around 25 grams to the AR glasses. We prepared three 3D printed mounts with different parameters to treat various users. These mounts were able to capture the user's face appropriately in a pilot study (N=15).

3.2 Camera Illumination and Sensing

The camera module is based on 850-nanometer infrared illumination and sensing. It has six infrared light bulbs around the camera lens, which provides active illumination source to enable FaceSight to work properly even in a completely dark environment. More importantly, the active infrared illumination warrants stable image quality, by adjusting the lighting power of the light bulbs and exposure value of the camera via commercial webcam software or specific computer vision API, we can illuminate only the face area while keeping the background (such as the chest and objects in the surrounding environment) almost dark. Figure 4 illustrates the image effects of setting different lighting powers to the camera. This illumination adjustment not only eases the computer-vision problem, making the recognition algorithm efficient and robust, but also reduces potential privacy concerns of capturing the users' surrounding environment.

4 HAND-TO-FACE GESTURE INTERACTION

In this section, we explore the design space of hand-to-face gesture interaction that can be supported by FaceSight. We analyze the potential feature dimensions in the design space and build a gesture set leveraging the findings. The gesture set contains twenty-one hand-to-face gestures, including seven hand-to-cheek gestures, six hand-to-nose gestures, four hand-to-mouth gestures, and four hand-to-chin gestures. Figure 5 asterisk marks the novel gestures that have not been described in any prior research about the hand-to-face gestural input. Such a large hand-to-face gesture set not only enriches input methods to AR glasses, but also has the potential to improve the interaction efficiency and user experience. We discuss the detailed feature dimensions and their potential usages below.

4.1 Touch Location

Distinguishing the touch points on different locations or landmarks on the face is a first feature dimension of the design space for hand-to-face gesture interaction. Locating the touch point can be achieved by computer vision methods [14, 46]. Thanks to the unique camera placement, FaceSight can capture most areas of the lower face, including the cheek, nose, chin, and mouth. Some facial parts also have several sub-areas: the cheek has the left and right sides, the nose has the nose tip, and the left and right nose wings, while the mouth has the left and right mouth corner, and the middle point. Compared with the other areas of face (e.g., forehead), these facial areas are where people touch more frequently, as well as the most preferred to use [27]. One restriction is that area on or around the ears are unavailable for interaction because of limited camera FoV.

4.2 Tapping and Swiping

Tapping and swiping are the most common input methods on the modern touchscreen [5], thus transferring them to hand-to-face gestures requires little learning effort of users. Besides, considering different types of tapping including single click, double click, and long touch (i.e., stay for several seconds) [39] can further enlarge the gesture set.

The smooth surface of the cheek and chin is well-suited for fingertip swiping to complete panning or zooming tasks. We finally formed three

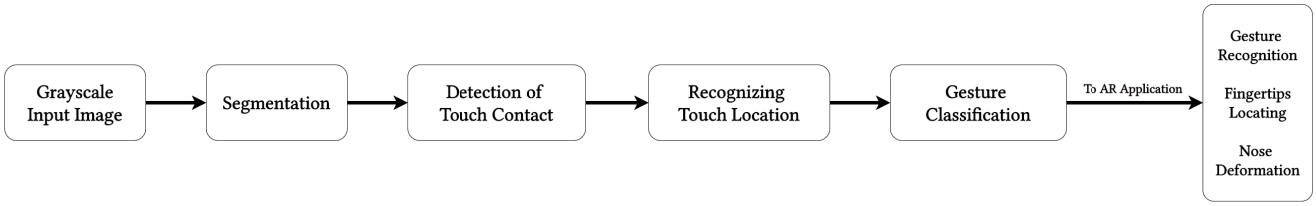

Figure 6: The steps of FaceSight for recognizing hand-to-face gestures.

swiping gestures. The first one is hand swiping vertically on one side of cheek. Two hands with two cheek sides enable the control of different scroll bars in this manner, such as a volume bar and a brightness bar on two sides. The second one is both hands swiping vertically on two sides of the cheek simultaneously, and the third one is one hand swiping horizontally on the chin.

4.3 Symbolic Hand Gesture

Camera sensing techniques also have the potential to recognize symbolic gestures. Associated with the face location, some gestures possess specific semantic implication that leads to high intuitiveness and learnability [25–27]. For example, the gesture of "making a phone call" is used to automatically launch the contacts applications; the gesture of "covering the mouth" can be used to activate voice input [50]; the "shush" gesture can mute an application or the device. This can lessen the time consuming of finding target applications or accessing a specific element.

4.4 Nose Deformation as Input

As shown in figure 4, the video camera is immediately above the nose, so that a user's nose can be captured by the camera in high resolution. It is possible to detect slight deformation of the nose when it is pushed or nipped by fingers. The deformation can be used as continuous input signal to enable novel nose-based input techniques. For example, one can control a scroll-bar by pushing the nose or using different levels of pushing pressure to trigger different functions. We proposed two different pressures of nose-pushing gestures, one is a gentle pushing with rare deformation, and the other one is a rude pushing that will deform the nose obviously.

4.5 Hand Identity

Thanks to the symmetrical camera view, FaceSight also has the capability to determine the identity of touching hands, which can further increase the interaction expressivity and also provide more solutions for the design of the gesture set. For example, one can use the right hand to manipulate virtual objects, and use the left hand to switch tools or modes. In figure 5, all gestures were described from the right-handed perspective, we added two left-handed tapping gestures on the cheek to illustrate this concept.

4.6 Number of Touching Fingers

Multi-finger interaction is also a widely used interaction technique on the modern touchscreen, including such gestures into the design of hand-to-face gestures set is an intuitive choice in our exploration. we showed examples of using two fingers (index finger and middle finger) tapping on the chin, and using thumb and index finger to nip the nose.

5 THE RECOGNITION ALGORITHM

In this section, we describe our algorithm pipeline to recognize the set of hand-to-face gestures in FaceSight, we then evaluate the detection accuracy and computing efficiency.

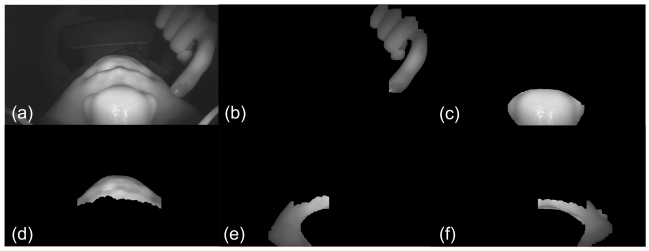

Figure 7: Example results of our segmentation approach while a user touch the right cheek. (a) input image, (b) hand, (c) nose, (d) mouth, (e) left cheek, and (f) right cheek.

5.1 The Algorithm Pipeline

Figure 6 illustrates our recognition pipeline step by step. Given a gray-scale input image captured from the infrared camera, we first segment the hand, nose, mouth, left and right cheeks by applying a number of brightness features. Then, we conduct a four-stage algorithm to detect hand-to-face gestures: 1) Detection of touch contact. 2) Recognizing the location (nose, mouth, chin, left cheek, or right cheek) if we detect a touch in stage 1. 3) Using a convolutional neural network (CNN) model to classify hand posture performed on that location. The CNN model is trained for each location separately. 4) If a classified gesture belongs to the categories of *nose pushing* and *cheek or chin tapping*, we further conduct stage 4, running corresponding algorithms to determine the required interaction parameters, like estimating the nose deformation or locating fingertips for swiping inputs. We describe each step in the pipeline below.

5.1.1 Facial Region (and Hand) Segmentation. Thanks to FaceSight's unique camera and illumination setup, there is a high contrast between the background (the chest or further objects) and foreground (the face and touching hands) (Fig. 4). We firstly apply brightness thresholds to eliminate the background accurately. To extract the hand, we take advantage of the fact that the facial region is static in the image, while a hand is moving that will cause noticeable brightness changes. Specifically, we use the current frame to subtract a background frame (i.e., an image contains only the face) to achieve it. To segment nose and mouth, we leverage the features and empirical parameters: 1) The directional change of intensity between targets and surrounding pixels; 2) Nose is always located in the middle of the image as well as connects to the bottom (from the camera view); 3) Mouth must be above the nose and below the top of the face (from the camera view); 4) Nose possesses higher brightness due to its proximity to the camera and lighting bulbs. Finally, we determine the left and right cheeks by computing the remaining pixels (input image without background, hand, nose, and mouth). Using these features, we could effectively segment the user's hand, nose, mouth, left cheek, and right cheek, as shown in figure 7.

5.1.2 Stage 1: Detection of Touch Contact. For hand-to-face gesture interaction, it is essential to determine when the hand/finger contacts the face. In

a 2D image, a straightforward method is to determine whether two objects (hand, face) are overlapped. However, if one's target is the nose, a touch will falsely recall as the fingertip must first contact the face before getting to the nose. To address this issue, we combine two features to detect a touch: 1) A contact occurs when the fingertip is on or within the face region; 2) A touch event is reported if the fingertip movement has a sudden change in two consecutive frames. Specially, We determine a "sudden change" by first calculating the distance between nose centroid and fingertip, and judging whether this parameter becomes greater in two consecutive frames; The second statement above leads to another frame of delay, but it can effectively reduce unintentional activation and only report the exact touch moment.

5.1.3 Stage 2: Recognizing Touch Location. If a hand as well as its contact with the face is detected in the image, we further determine the location where the touch occurs (five categories: nose, mouth, chin, left cheek, and right cheek). To do this, we find the four segmented facial regions' keypoints, such as the leftmost of the nose regions, topmost of the mouth regions (i.e., chin), and so on. We determine the landmark to which fingertip is closest. The distance between the fingertip and contour is also useful as it reflected whether the fingertip is in that region.

5.1.4 Stage 3: Gesture Classification with CNN. Given a touch location, we further classify what hand posture of a user performing on that location. We use a convolutional neural network (CNN) model, a well-known deep learning architecture, proven to be very powerful in image classification tasks. We train a CNN model for each location separately (i.e., nose, mouth, chin, left cheek, and right cheek). The model contains two convolutional layers, a 2×2 maximum pooling layer, and a fully connected layer. We adjust the parameters of the convolutional layers to improve accuracy, and finally determine the following parameters: first layer 11×11 for kernel size, 5 for stride step and 3 for padding, the second layer 5×5 for kernel size, 1 for stride step and 2 for padding. We use softmax and cross-entropy as the loss function, and use accuracy rate and false recognition rate as the accuracy index. We use Adam optimizer to train all models. The learning rate coefficient is set to 3×10^{-3}. The model's input is the hand regions with downsampling to 200×200, while the output is a label corresponding to a specific gesture.

5.1.5 Stage 4.1: Locating the Touching Fingertip for Continuous Input. Based on the segmented hand region(s), we recognized the points along the contour that achieved local minimum according to the distance to the face region as the fingertip candidates, and we finally located the fingertip location as the lowest one. We calculate the average of the location found in the instant frame and two previous frames for smoothing. We complete fingertip locating before running the detection of touch contact step (Stage 1), due to it requires the fingertip's parameters.

Suppose the current hand posture recognized by stage 3 is a cheek tapping (Tap RR, Tap RL, Tap LL, Tap LR, or using both hands together) or chin tapping (Tap (1 finger)), we start to track the fingertip(s) for continuous input by calculating its(their) displacement between each frame.

5.1.6 Stage 4.2: Estimating the Degree of Nose Deformation. The action of pushing the nose can cause deformation or shifting of the nose region. We calculate the change of area of nose region (area decreased since the nose wing is squeezed by fingertip) and the offset of the nose centroid and the nose wing keypoint between sequential image frames. Precisely, we empirically determine the weights of 0.00005 for the area change, 0.02 for the offset of nose centroid, and 0.04 for the offset of nose wing keypoint. These offset threshold units are pixel. We add them up and recognize a rude pushing if the sum greater than 1.0. This approach would only activate when the recognition of stage 3 is a nose-pushing gesture.

5.2 Data Collection

We collected hand-to-face gesture samples used to evaluate both the contact detection and classification accuracy of FaceSight.

5.2.1 Participant. In this study, we recruited 10 participants (2 females, 8 males), their ages ranged from 18 to 55 (mean age = 27.8). The goal was to collect data from users with various facial forms. All of them were right-handed and usually wore eyeglasses in their daily life.

5.2.2 Design and Procedure. We expand each swiping gesture (i.e., *Swipe(1 hand), Swipe(2 hands)* on cheek, and *Swipe* on chin) to two different gestures with opposite movements used to evaluate the fingertip locating performance, such as swipe up/down, and swipe left/right. In total, our gesture set has 24 gestures (18 + 3 swiping gestures × 2 movements). Participants were required to perform each of 24 gestures for 60 times at random order. During the data collection, we asked participants to move their hand(s) away from the face between each contact to simulate the process of raising a hand to touch face. Participants were seated to perform gestures and had a one-minute break for each round. We video-recorded the data.

5.2.3 Data. After data collection, we collected a total of 14440 hand-to-face gesture samples (10 participants × 24 gestures × 60 times). These samples were all used to evaluate our touch contact detection and touch location recognition methods. To acquire data for training the neural network models, we obtained the hand region of each frame from the recorded video using our segmentation approach, and finally received 198572 images. We manually examined those images and excluded inappropriate ones, such as a hand being out of the camera's FoV or a gesture being wrongly performed by participant. After filtering, we remained in 194204 images (97.8%).

We created five datasets associated with the facial parts: nose, mouth, chin, left cheek, and right cheek. CNN models were trained for each of the five datasets, as shown in table 3. The gesture of *Swipe (2 hands)* required the delimiter (i.e., know when to activate the mode), so we replaced it with *Tap (2 hands)* gesture to classify. On the other hand, some gestures involved contact with multiple facial parts, such as the *Cover Mouth* gesture and the *Tap (2 hands)* gesture, the data of the two gestures were placed into multiple datasets they involved. Overall, there were 67553 images in the nose dataset, 30368 images in the mouth dataset, 33582 images in the chin dataset, 51869 images in the left cheek dataset, and 57747 images in the right cheek dataset.

We added two additional datasets to evaluate the accuracy of fingertip swiping and nose deformation estimation. The first dataset contained six different swiping gestures on the cheek and chin. The second dataset had four classes, including the nose-pushing gestures of gentle or rude pressure.

5.3 Algorithm Evaluation

We evaluated the recognition accuracy and computational efficiency of each stage in our algorithm pipeline. Note that each stage was evaluated individually, not depending on the results of the previous stage.

5.3.1 Recognition Accuracy of Touch Contact and Location. Through offline calculation, our contact detection algorithm successfully recalled 14097 / 14440 touches (recall rate: 97.90%). Most of the false negatives came from that the touch was too close to the border of the camera view, making the fingertip invisible or being excessively dark that could be cut after segmentation. We detected 168 false positives (precision: 98.82%), and the F1-Score was 98.36%.

Over those recalled touches, the average accuracy of location recognition was 94.69% (Nose: 92.67%, Mouth: 94.43%, Chin: 95.12%, Left Cheek: 95.06%, Right Cheek: 94.22%). Table 2 was the confusion matrix of touch location recognition. Most of the misidentifications occurred between: 1) Nose and cheek; 2) Mouth and chin, particularly at performing shush gesture or gripping the chin; 3) Mouth and cheek.

Table 1. Evaluation of hand-face contact detection. (Stage 1)

	Sample	Recall	Precision	F1-Score
Detection of Touch Contact	14400	97.90%	98.82%	98.36%

Table 2. Confusion matrix of touch location recognition. (Stage 2)

	Predicted (%)				
Truth	Nose	Mouth	Chin	Left Cheek	Right Cheek
Nose	92.67	1.58	0.05	2.15	3.55
Mouth	0.22	94.43	3.96	0.11	1.28
Chin	0.00	4.84	95.12	0.04	0.00
Left Cheek	1.21	3.63	0.10	95.06	0.00
Right Cheek	1.44	4.28	0.06	0.00	94.22

Table 3. The classification accuracy of CNN models proved by leave-one-out validation. (Stage 3)

Dataset	Gesture	Accuracy
Nose	Tap(Tip), Pushing(L), Pushing(R), Nip, Cover Mouth	96.18%
Mouth	Tap(L), Tap(R), Shush	99.53%
Chin	Tap(1 finger), Tap(2 fingers), Grip	94.00%
Left Cheek	Tap(LL), Tap(RL), Tap (2 hands), Cover Mouth	94.65%
Right Cheek	Tap(LR), Tap(RR), Tap (2 hands), Cover Mouth, Phone	97.73%

Table 4. The recognition accuracy of two different pressures of nose-pushing and swiping gestures. (Stage 4)

Dataset	Gesture	Accuracy
Swiping	Swipe up on cheek (1 hand), Swipe down on cheek (1 hand), Swipe up on cheek (2 hands), Swipe down on cheek (2 hands), Swipe left on chin, Swipe right on chin.	94.67%
Nose Pushing	Gentle Pushing(R), Rude Pushing (R), Gentle Pushing(L), Rude Pushing (L)	94.12%

5.3.2 Recognition Accuracy of Hand-to-Face Gestures. Considering participants could perform gestures in unique ways, and the camera views produced by different participants had a slight difference, we followed leave-one-out cross-validation approach to train and test our classification models, by training the models using all other users' data except for the wearer's data. All model was trained for ten epochs. We obtained the recognition accuracies as follows: nose 96.18% (5 classes), mouth 99.53% (3 classes), chin 94.00% (3 classes), left cheek 94.65% (4 classes), and right cheek 97.73% (5 classes). The average accuracy of the five classification models was 96.42%, as shown in Table 3.

5.3.3 Recognition Accuracy of Nose Pushing Estimation and Swiping. Results are shown in Table 4. Our method achieved 94.12% accuracy in recognizing two different nose-pushing pressures on both nose wings. Meanwhile, the recognition accuracy of six swiping gestures was 94.67% (Chin swiping: 97.5%. Cheek Swiping (1 hand): 94.17%. Cheek Swiping (2 hands): 92.18%).

5.3.4 Computing Efficiency. We ran our algorithm pipeline on a server with 1 GTX 1080 Ti NVIDIA GPU, 12GB memory and Intel(R) Xeon(R) CPU. We tested the average time each component took to process a single frame (Resolution: 960×540). The results were 35 ms for the segmentation algorithm, 13 ms for the CNN classification. The fingertip locating and contact detection approach both spent only 1ms.

5.3.5 Summary. To process the image data generated by the downward-facing camera, we leverage the high contrast properties for segmentation, geometric features to detect touch contact, and CNN models to recognize hand-to-face gestures. According to current results, if a user performs a *Gentle Pushing(R)* gesture, it would be correctly recognized at 82.51% accuracy (98.36% × 92.67% × 96.18% × 94.12%, of stage 1 to 4 respectively). For all 24 gestures, FaceSight achieves an overall classification accuracy of 83.06 % for classifying them simultaneously, which is validated by the data of ten users. We deem that the results are sufficient to prove the computation and recognition feasibility of using a camera sensing solution to identify a large space of hand-to-face interaction. In the future, a thorough dataset is required to build more robust and generalized models.

To test the applications and interaction techniques, we trained additional models using all data we gathered from the data collection study. For each application, it can register or disable interaction of a particular facial model, to only interact with the desired one. For example, if an application only requires hand-to-nose gesture interaction, it can directly conduct stage 3 (i.e., classify gesture using CNN model) once our algorithm recalls a touch contact at stage 1, as it does not need to recognize touch location anymore. In that case, the algorithm performance could significantly improve since it reduces the computation and recognition of stage 2.

6 APPLICATION AND INTERACTION DESIGN

Based on the gesture set proposed in this paper, we designed and developed five AR applications and twelve interaction techniques to demonstrate the potential uses of FaceSight. The five applications included *Home*, *Video Player*, *Photo Library*, *Contacts*, and *Voice Assistant*. We chose these applications since they were representative tasks on AR HMD devices. We adopted a client-server architecture to test our purpose, in which the detection pipeline was implemented on a server, and the applications were developed on the Nreal Light AR glasses. The server calculated locally and sent predictions to the AR glasses via Wi-Fi.

6.1 Home

Home is a very common application in commercial AR systems (e.g., HoloLens). It usually has a user interface that displays the icons of all other applications on the device and enables the user to access these applications via its corresponding icon. For example, in HoloLens, a user needs to conduct a bloom gesture to call the Home interface. With FaceSight, we allowed users to conduct touch gestures on their chin to interact with the Home application. Specifically, a user can tap her chin with both their index and middle fingers to call the home interface. Then she can gaze at a specific icon and tap her cheek or chin with one finger as selection to open the corresponding application. In our Home application, there were four icons on the user interface which represented the four other applications we describe in this section.

6.2 Video Player

We designed six interaction techniques to allow users to control the Video Player with FaceSight: pause/resume, fast forward, fast rewind, playing the next video, playing the last video, and mute. Specifically, a user can tap on the nose tip to pause/resume a video. and she can gently push her left (and right) nose wing for fast forward (and rewind), or rudely push her left (and right) nose wing for playing last (and next) video. She can also conduct a shush gesture to mute the video. Different from the conventional interaction techniques in current commercial AR systems (e.g., HoloLens) where a user had to stare at a button and conducted an air-tap gesture to trigger the function, FaceSight allowed the user to easily assign a command using a

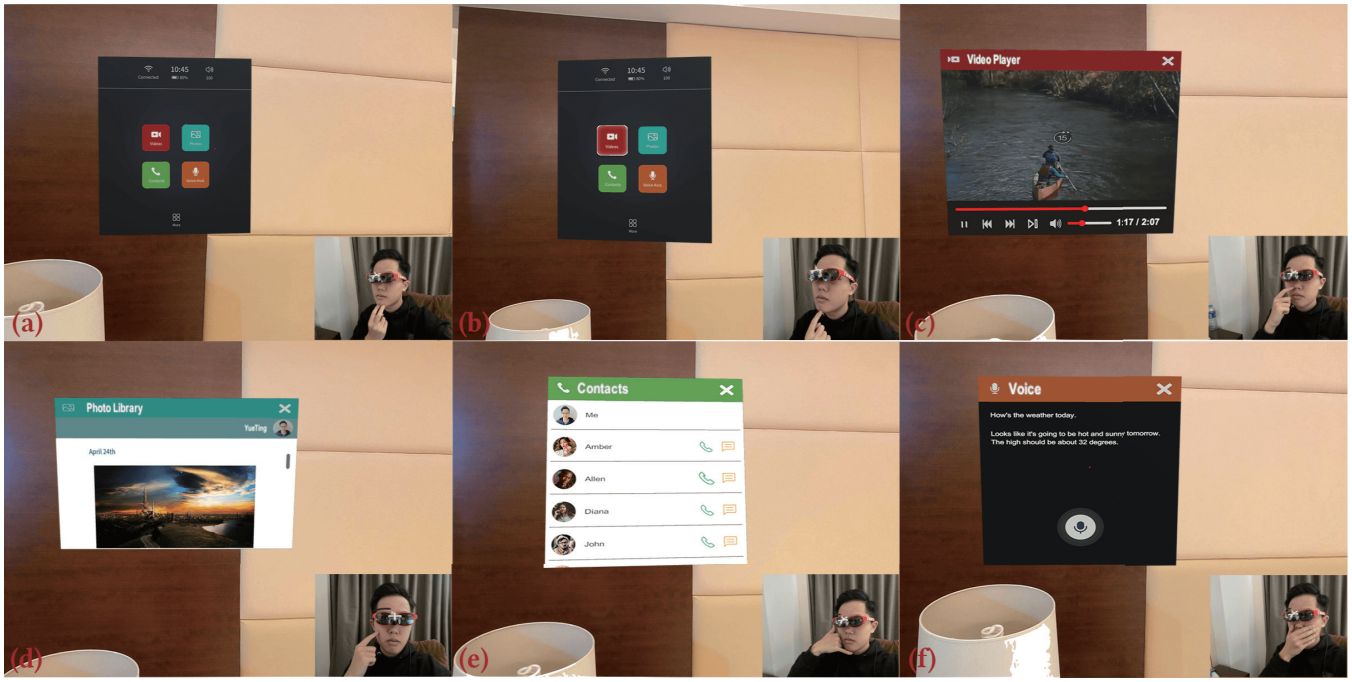

Figure 8: The demonstrated AR applications and interaction techniques: a) Returning to home page by tapping on chin with two fingers; b) Gazing at an icon and tapping on chin with one finger for confirmation; c) Gentle pushing the nose wing for fast forward or rude pushing it to play next video; d) Swiping on cheek to browse pictures; e) The phone gesture as a shortcut to open Contacts application; f) Conducting the cover mouth gesture to activate voice input, and releasing hand to close it.

simple eyes-free gesture. To ease the process of selecting a video and focus on the interaction tasks for video control, we simplified this application by loading default demo videos. The video played automatically when the user opened the Video Player application.

6.3 Photo Library

The Photo Library application included all the pictures taken by the user. To navigate the library, a user can swipe vertically with her index finger on her cheek to browse the pictures. She can also select an image by staring at it and tapping on cheek, and swipe with her two index fingers on each cheek together to scale a picture (swiping down to zoom out and swiping up to zoom in).

6.4 Contacts

When a user wanted to make a phonecall, FaceSight allowed the user to open the Contacts application quickly by conducting the Phone gesture. This gesture provided a convenient and intuitive way to access the Contacts, which largely saved the user's time from navigating the whole application list. After opening the Contacts applications, the user can then gaze at the target contact and tap on her cheek to call.

6.5 Voice Assistant

Our design of Voice Assistant application was inspired by the works [37, 50]. Originally, to trigger a voice assistant, a user usually needs to navigate the whole application list to find the corresponding application, or speak a specific keyword to wake up the voice assistant, such as "OK, Google." With FaceSight, a user can speak to the voice assistant directly with the cover mouth gesture, so that the voice assistant can be activated automatically and respond to the user's voice input. The voice assistant application closed immediately if the user released her hand.

7 USER STUDY: USABILITY EVALUATION

We conduct a user study to evaluate the usability of FaceSight. We invite ten users to perform twelve tasks with the help of FaceSight, and collect their subjective feedback on the interaction design, form factor, as well as the real-time performance.

7.1 Participants and Apparatus

We recruited another 10 participants (8 males, 2 females) from our university. Participants' ages ranged from 21 to 27 (mean = 23.7). All of them were familiar with mid-air hand gestures supported by Hololens. Four participants had prior experience using other input methods, such as a trackpad or head movement (using built-in sensors on HMD to sense head position and orientation as input [51]). The study was conducted with the same AR glasses and camera described in the hardware section.

7.2 Task

There are twelve tasks in total, corresponding to twelve interaction techniques introduced in *Application and Interaction Design* section. Among them, *Home* application corresponds to two tasks (Return home, Select). *Video Player* application has six tasks (pause/play, fast forward, fast rewind, play next video, play the previous video, and mute). *Photo Library* application possesses two tasks (page scrolling, picture zooming). *Contacts* application and *Voice Assistant* application both relate to a task (activate).

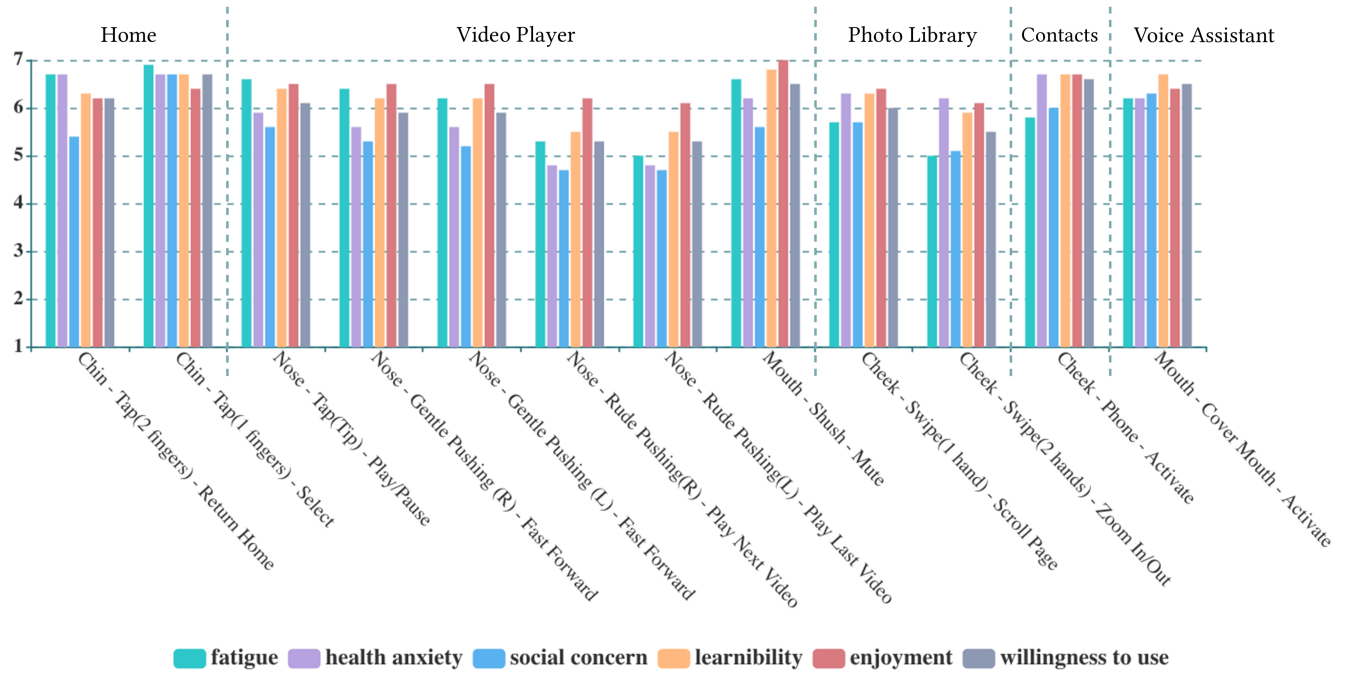

Figure 9: The subjective ratings of twelve hand-to-face interaction techniques in our customized AR applications. 1=strongly disagree, 7=strongly agree. The scores of three metrics of "fatigue, "health anxiety", and "social concern" were rated reversely.

7.3 Procedure

The study lasted about 60 minutes. We firstly gave a brief interview, asking the participants' demographic information and their prior experience with AR Glasses. An experimenter then introduced the concept of FaceSight and the tasks in this study. After that, we started the experiment using a randomized task order. For each task, they performed a gesture to instruct AR glasses to execute a specific command. They can freely attempt until they fully understood that technique. After the whole tasks were completed, we asked them to give 7-point Likert scale scores for their agreements with 6 metrics to evaluate their experience (7 means strongly agree and 1 means strongly disagree). The metrics included:

- Fatigue: "Performing the gesture makes me tired." (the scores were reversed)
- Health anxiety: "The gesture leads to my health consideration." (the scores were reversed)
- Social concern: "Performing the gesture would raise my social concern" (the scores were reversed)
- Learnability: "The technique was easy to learn."
- Enjoyment: "The technique was fun to use."
- Willingness to use: "I would use the technique on my AR glasses."

We also asked them to reflect their feelings on the form factor with three statements along a 7-point Likert scale:

- Did you socially accept the form factor?
- Did you feel concerned about your face being exposed to a camera?
- Did you perceive narrowing a sight for real world because of camera physical protrusion?

In the 7-point Likert scale, 7 means the most positive feeling, while 1 means the most negative feeling. For example, if a user rated "7" for the first question, that represented she can fully accept the form factor and would use in any practical scenario; If a user rated "1", that defined she felt extremely embarrassed and had no willingness to use in public settings at all. For the first question, we required participants to focus on the placement of the add-on camera, not consider AR glasses and camera together. For the second question, they can see the images captured from the camera on a laptop's screen when they are wearing the AR glasses.

7.4 Results

7.4.1 Subjective Feedback on Interaction Design. Figure 9 showed the mean scores that participants gave to the gestures (fatigue, health anxiety, and social concern) and interaction techniques (learnability, enjoyment, willingness to use). In general, participants appreciated the interaction techniques in five example applications. They felt that the interaction techniques were fun to use and easy to learn, especially performing *Shush* gesture to mute an application got "7" points in the "fun to use" metric from every participant. However, one participant (P10) felt concerning that, "Performing shush gesture in public could be offensive to other people." With *Voice Assistant* application. While covering the mouth, P1 recommended that interaction can be expanded by further tapping on the cheek, such as answering a true or false question asked by the voice assistant. Some participants also gave suggestions on how to improve the interaction technique. For example, P10 said, "Spread two fingers on one cheek for zooming is more natural than using two hands." and four participants(P3, P6, P8, P9) mentioned, "For selection, I would like to tap on the cheek, not on the chin."

Participants consistently preferred subtle gestures, such as simply tapping on a particular facial region. Compared with swiping or nose pushing gestures, these ones required less physical effort, hygiene concern, and social awkwardness. They also appreciated the cover mouth gestures "This could lessen my social worries since my mouth would not expose to the public" From our observations, participants tended to use the hand that was on the same side as the target region. For example, if one's target is the left nose wing or left cheek, she would intuitively use her left hand, and vice versa.

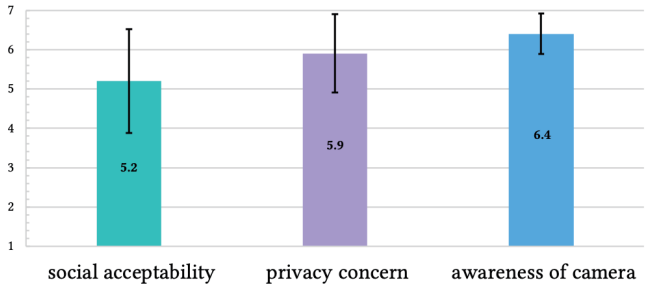

Figure 10: The subjective ratings on the form factor. 7 means the most positive feeling, and 1 means the most negative feeling. Error bars represent standard deviation.

We also collected comments related to the facial regions. Some participants thought "Nose has a well-defined shape, and the fingertip can be seen by the peripheral vision while touching nose, which helps complete a task accurately." and "Conducting gestures on chin costs less on health considerations and social concern." We also received negative feedback. Two females (P2, P5) expressed, "I would not swipe on my cheek, as it could break my makeup. But tapping on the cheek is tolerable." and P3 reported, "hand-to-chin gestures are more likely to use fingernail rather than finger pulp, making me a little bit uncomfortable." Meanwhile, a rhinitis patient (P9) showed a low willingness to interact with the nose. Concerning this, the mapping of a hand-to-face gesture and its function should better allow customized by users themselves.

7.4.2 Subjective Feelings about Form Factor. Figure 10 reflected the subjective ratings about the augmented camera use of FaceSight in terms of three statements (social acceptability of the form factor, privacy concern, and awareness of camera). Participants generally accepted the form factor (average=5.2 ±1.32) except P10, who rated 2 points. They also remained positive attitudes on the privacy issue (average=5.9 ±0.99). For the issue of camera obstructing real world view (average=6.4 ±0.52), two participants did not notice the camera protrusion during the whole process, while the other eight participants did. Still, most of them felt, "The device appears highly transparent and unclear. It is only visible while focusing on it." The results indicated to mount a camera compactly on AR glasses was socially acceptable and would not affect the user experience.

7.4.3 Observations of Real-Time Performance. From our observations, participants can be proficient with the system after around ten minutes of practicing. More than half participants reported "The gestures of *cover mouth*, *phone*, and *shush* can be robustly recognized." To our surprise, participants familiarized with gentle or rude pushing nose wing gestures with just a few learning. For the swiping gestures, four participants fail to scroll the page successfully in their earlier trials. One reason is that they were not familiar with the boundaries of the camera's viewing angle. After they followed the experimenter's instruction and training, all of them can scroll a page smoothly without frustration.

The hand-to-chin gestures of *Tap (1 finger)* and *Tap (2 fingers)* were easily misidentified. However, the performance had improvements while they spread two fingers or lifted their palm to the same height as the chin, since this could make more hand features expose to the camera. We also noticed that frequent head movements would suffer the performance, which could involve background noises and affect the algorithm accuracy.

8 DISCUSSIONS

In this section, we discuss the potential use cases, practical deployment issues of FaceSight.

8.1 Applicable Platform and Use Cases

FaceSight offers an eyes-free, tangible input modality that helps users leverage proprioceptive to complete AR tasks naturally, such as basic commands (e.g., confirming an action, browsing a page, zooming a picture), application shortcut, or mode switching, as demonstrated by the application examples. Based on our findings, hand-to-face interactions are well-suited to input AR HMD, with high enjoyment and less learning effort. However, considering the sanitation concerns resulting from frequent hand-face contact [31], FaceSight is not appropriate for highly repetitive task, like text entry. Moreover, FaceSight also has the potential to be embedded on other platforms, such as a pair of regular eyeglasses, which could allow users to make a quick response to their smartphone or surrounding intelligent devices, such as issuing commands, or dealing with notifications. VR HMDs are also compatible with FaceSight by installing a camera on the device or integrating it into the box that can enjoy the interaction benefits.

8.2 Form Factor

The hardware setup of FaceSight is compact. Due to the camera's proximity to human eyes, thus it is perceived to be highly transparent and unfocused. According to our study, all participants showed positive feelings about the augmented device (Figure 10). They reported that the slight and unnoticeable camera occlusion is acceptable, which neither disturbed their user experience nor raised discomfort. Also, The camera size can be further reduced. We can re-design AR glasses to integrate the module into the device and leave only the lens outside. It can even adopt a motor-based mechanical design to popup the lens whenever needs. The camera we used costs about 30$.

8.3 IR versus RGB Illumination and Sensing

In this paper, we realize FaceSight based on IR illumination and sensing. The benefits are as follows: 1) The IR camera usually accompanies lighting bulbs, providing active illumination without bothering users that promises the system's usability in a completely dark environment. 2) The illumination scope and power can be customized to satisfy the usages in specific scenarios. An appropriate setup can simplify the segmentation algorithm and reduce potential privacy concerns. 3) IR sensing scheme is widely applied in commercial human-computer interaction devices, such as Leap Motion and Kinect, proving the practical feasibility of both computing hardware and software. An alternative solution is adopting RGB sensing. RGB camera provides rich color information that is beneficial to implement skin segmentation and gesture detection. However, the shortcoming is that the algorithm significantly relies on the lighting conditions. Low illumination in the environment will affect the sensing quality and usability.

8.4 Midas Touch Problem

Prior studies [2, 31] indicated that people often touched their face with hands in their daily lives. Therefore, in practical use, we need to design interaction mechanisms to avoid Midas touch (i.e., trigger a command inadvertently) when using FaceSight. A solution is to create a mode-switch method to let users explicitly specify the interaction mode. For example, a user first performs a delimiter gesture and then completes the rest interaction. The delimiter gesture itself still requires to be robustly recognized and is rarely to be acted unintentionally. Other strategies include repeating the action twice to confirm the intention, specifically collecting unintended hand-to-face action data and training an extra classification model.

8.5 Impact of Head and Glass Movement

The head-worn camera can capture the human face stably regardless of head movements. However, frequent head movements in a complex environment could involve more background noises that would influence the robustness of segmentation algorithm. For the glass movement, except for leading the

camera to rest on the nose, most of the cases are negligible because the offset is too slight to affect the image quality.

9 LIMITATION AND FUTURE WORK

We discuss some limitations and future works of FaceSight in this section. First, infrared sensing is susceptible to interference from infrared light in the environment. In this work, our experiment was conducted in an ideal indoor environment. We did not study the problem of infrared interference in depth. Our rule-based segmentation approach may generate unsatisfactory results in some circumstances (e.g., in outdoor space), where we could lose some utilized features, like intensity contrast between cheek and nose. We would further test our system in those conditions and research a more robust segmentation algorithm (e.g., semantic segmentation model) to address this issue.

Second, in this study, all gestures were designed by authors, as the goal was to explore the design possibilities. In the future, we would recruit users from different cultures and conduct a study to curate the gestures and their applicability carefully, with special attention towards social acceptability. Third, our current system runs the computer-vision algorithm on a remote server, which takes advantage of GPUs to perform CNN model inference. For practical use, it is important to research a lightweight classification algorithm that can run locally on AR glasses, with particular concerns of efficiency and power consumption.

Currently, we only explored the hand-to-face interaction with a downward-looking camera mounted on the nose edge of AR glasses. The camera also has the potentials to sense facial expressions (e.g., smiling, puffing the cheek) [28, 52], which gives a unique opportunity for hands-free interaction. Moreover, commercial AR glasses usually contain microphone sensors, bringing additional information (the audio signal resulting from hand-face contact [48]) that may further improve the contact detection performance. We expect to incorporate these features into the future version of FaceSight.

10 CONCLUSION

We present FaceSight, a novel camera-based sensing technique to enable hand-to-face gesture interaction on AR glasses. FaceSight uses a downward-looking camera to capture a users' lower face and leverages infrared illumination and sensing to enhance the quality and signal-to-noise ratio of the image. Thanks to the high-resolution image of hand and face, FaceSight is capable of detecting rich and subtle hand gestures performed on the face. To explore the interaction potential of FaceSight, we proposed a rich gesture set that contained 21 hand-to-face gestures associating with the nose, mouth, chin, and cheek. Out of them, ten novel gestures have not been present in prior literature. We also demonstrated the value of these hand-to-face gestures by implementing customized AR applications. Our recognition algorithm takes advantage of the high contrast of the image, and can detect touch contact and classify hand-to-face gestures in high accuracy (83.06% in recognizing all gestures simultaneously), reflecting the advantage of the sensing scheme of FaceSight. We conclude this work by discussing the potential usages, issues for practical deployment, limitations of the current work. In conclusion, our exploration suggests that FaceSight has great potential to realize and advance hand-to-face gesture interaction on AR glasses.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 61521002, No. 61572276, and National Key R&D Program of China No. 2019AAA0105200, and also by Beijing Key Lab of Networked Multimedia, the Institute for Guo Qiang, Tsinghua University, Institute for Artificial Intelligence, Tsinghua University (THUAI), and Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. 2019. MeCap: Whole-Body Digitization for Low-Cost VR/AR Headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 453–462. https://doi.org/10.1145/3332165.3347889

[2] Wladimir J. Alonso, Francielle C. Nascimento, Joseph Shapiro, and Cynthia Schuck-Paim. 2013. Facing Ubiquitous Viruses: When Hand Washing Is Not Enough. *Clinical Infectious Diseases* 56, 4 (02 2013), 617–617. https://doi.org/10.1093/cid/cis961 arXiv:https://academic.oup.com/cid/article-pdf/56/4/617/1009241/cis961.pdf

[3] Abdelkareem Bedri, Diana Li, Rushil Khurana, Kunal Bhuwalka, and Mayank Goel. 2020. FitByte: Automatic Diet Monitoring in Unconstrained Situations Using Multimodal Sensing on Eyeglasses. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376869

[4] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 549–556. https://doi.org/10.1145/2807442.2807450

[5] Xiang 'Anthony' Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. 2014. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 159–168. https://doi.org/10.1145/2556288.2556955

[6] Christine Dierk, Sarah Sterman, Molly Jane Pearce Nicholas, and Eric Paulos. 2018. HäirIÖ: Human Hair as Interactive Material. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction* (Stockholm, Sweden) *(TEI '18)*. Association for Computing Machinery, New York, NY, USA, 148–157. https://doi.org/10.1145/3173225.3173232

[7] Mohamed Elgharib, Mallikarjun BR, Ayush Tewari, Hyeongwoo Kim, Wentao Liu, Hans-Peter Seidel, and Christian Theobalt. 2019. EgoFace: Egocentric Face Performance Capture and Videorealistic Reenactment. arXiv:1905.10822 [cs.CV]

[8] 2020. Nreal Light AR glasses. Website. 2020. https://www.nreal.ai/.

[9] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 283–292. https://doi.org/10.1145/2047196.2047233

[10] Sean G. Gustafson, Bernhard Rabe, and Patrick M. Baudisch. 2013. Understanding Palm-Based Imaginary Interfaces: The Role of Visual and Tactile Cues When Browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 889–898. https://doi.org/10.1145/2470654.2466114

[11] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: Wearable Multitouch Interaction Everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 441–450. https://doi.org/10.1145/2047196.2047255

[12] Chris Harrison, Shilpa Ramamurthy, and Scott E. Hudson. 2012. On-Body Interaction: Armed and Dangerous. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction* (Kingston, Ontario, Canada) *(TEI '12)*. Association for Computing Machinery, New York, NY, USA, 69–76. https://doi.org/10.1145/2148131.2148148

[13] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the Body as an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 453–462. https://doi.org/10.1145/1753326.1753394

[14] R. Herpers, M. Michaelis, K. H. Lichtenauer, and G. Sommer. 1996. Edge and Keypoint Detection in Facial Regions. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96) (FG '96)*. IEEE Computer Society, USA, 212.

[15] Yasha Iravantchi, Yang Zhang, Evi Bernitsas, Mayank Goel, and Chris Harrison. 2019. Interferi: Gesture Sensing Using On-Body Acoustic Interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 276, 13 pages. https://doi.org/10.1145/3290605.3300506

[16] Takashi Kikuchi, Yuta Sugiura, Katsutoshi Masai, Maki Sugimoto, and Bruce H. Thomas. 2017. EarTouch: Turning the Ear into an Input Surface. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Vienna, Austria) *(MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, Article 27, 6 pages. https://doi.org/10.1145/

3098279.3098538

[17] Gierad Laput, Robert Xiao, Xiang "Anthony" Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin Buttons: Cheap, Small, Low-Powered and Clickable Fixed-Icon Laser Projectors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 389–394. https://doi.org/10.1145/2642918.2647356

[18] DoYoung Lee, Youryang Lee, Yonghwan Shin, and Ian Oakley. 2018. Designing Socially Acceptable Hand-to-Face Input. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 711–723. https://doi.org/10.1145/3242587.3242642

[19] 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* (Maui, Hawaii) *(ISWC '17)*. Association for Computing Machinery, New York, NY, USA, 94–97. https://doi.org/10.1145/3123021.3123060

[20] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. 2015. Facial Performance Sensing Head-Mounted Display. *ACM Trans. Graph.* 34, 4, Article 47 (July 2015), 9 pages. https://doi.org/10.1145/2766939

[21] Chen Liang, Chun Yu, Xiaoying Wei, Xuhai Xu, Yongquan Hu, Yuntao Wang, and Yuanchun Shi. 2021. Auth+Track: Enabling Authentication Free Interaction on Smartphone by Continuous User Tracking. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Tokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3411764.3445624

[22] Roman Lissermann, Jochen Huber, Aristotelis Hadjakos, Suranga Nanayakkara, and Max Mühlhäuser. 2014. EarPut: Augmenting Ear-Worn Devices for Ear-Based Interaction. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design* (Sydney, New South Wales, Australia) *(OzCHI '14)*. Association for Computing Machinery, New York, NY, USA, 300–307. https://doi.org/10.1145/2686612.2686655

[23] Guanhong Liu, Yizheng Gu, Yiwen Yin, Chun Yu, Yuntao Wang, Haipeng Mi, and Yuanchun Shi. 2020. Keep the Phone in Your Pocket: Enabling Smartphone Operation with an IMU Ring for Visually Impaired People. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 58 (June 2020), 23 pages. https://doi.org/10.1145/3397308

[24] Mona Hosseinkhani Loorak, Wei Zhou, Ha Trinh, Jian Zhao, and Wei Li. 2019. Hand-Over-Face Input Sensing for Interaction with Smartphones through the Built-in Camera. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 32, 12 pages. https://doi.org/10.1145/3338286.3340143

[25] Marwa Mahmoud, Tadas Baltrušaitis, Peter Robinson, and Laurel D. Riek. 2011. 3D Corpus of Spontaneous Complex Mental States. In *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction - Volume Part I* (Memphis, TN) *(ACII'11)*. Springer-Verlag, Berlin, Heidelberg, 205–214.

[26] Marwa Mahmoud, Tadas Baltrušaitis, and Peter Robinson. 2016. Automatic Analysis of Naturalistic Hand-Over-Face Gestures. *ACM Trans. Interact. Intell. Syst.* 6, 2, Article 19 (July 2016), 18 pages. https://doi.org/10.1145/2946796

[27] Marwa Mahmoud and Peter Robinson. 2011. Interpreting Hand-Over-Face Gestures. In *Affective Computing and Intelligent Interaction*, Sidney D'Mello, Arthur Graesser, Björn Schuller, and Jean-Claude Martin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 248–255.

[28] Katsutoshi Masai, Yuta Sugiura, Masa Ogata, Kai Kunze, Masahiko Inami, and Maki Sugimoto. 2016. Facial Expression Recognition in Daily Life by Embedded Photo Reflective Sensors on Smart Eyewear. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (Sonoma, California, USA) *(IUI '16)*. Association for Computing Machinery, New York, NY, USA, 317–326. https://doi.org/10.1145/2856767.2856770

[29] 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* (Seoul, Republic of Korea) *(AH '18)*. Association for Computing Machinery, New York, NY, USA, Article 23, 5 pages. https://doi.org/10.1145/3174910.3174924

[30] C. Metzger, M. Anderson, and T. Starner. 2004. FreeDigiter: a contact-free device for gesture control. In *Eighth International Symposium on Wearable Computers*, Vol. 1. IEEE, Piscataway, NJ, USA, 18–21. https://doi.org/10.1109/ISWC.2004.23

[31] Mark Nicas and Daniel Best. 2008. A Study Quantifying the Hand-to-Face Contact Rate and Its Potential Application to Predicting Respiratory Tract Infection. *Journal of Occupational and Environmental Hygiene* 5, 6 (2008), 347–352. https://doi.org/10.1080/15459620802003896

[32] J. Nie, Y. Hu, Y. Wang, S. Xia, and X. Jiang. 2020. SPIDERS: Low-Cost Wireless Glasses for Continuous In-Situ Bio-Signal Acquisition and Emotion Recognition. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE Computer Society, Los Alamitos, CA, USA, 27–39. https://doi.org/10.1109/IoTDI49375.2020.00011

[33] Masa Ogata, Yuta Sugiura, Yasutoshi Makino, Masahiko Inami, and Michita Imai. 2013. SenSkin: Adapting Skin as a Soft Interface. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 539–544. https://doi.org/10.1145/2501988.2502039

[34] Kyle Olszewski, Joseph J. Lim, Shunsuke Saito, and Hao Li. 2016. High-Fidelity Facial and Speech Animation for VR HMDs. *ACM Trans. Graph.* 35, 6, Article 221 (Nov. 2016), 14 pages. https://doi.org/10.1145/2980179.2980252

[35] Ondrej Polacek, Thomas Grill, and Manfred Tscheligi. 2013. NoseTapping: What Else Can You Do with Your Nose?. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia* (Luleå, Sweden) *(MUM '13)*. Association for Computing Machinery, New York, NY, USA, Article 32, 9 pages. https://doi.org/10.1145/2541831.2541867

[36] Manuel Prätorius, Aaron Scherzinger, and Klaus Hinrichs. 2015. SkInteract: An On-body Interaction System Based on Skin-Texture Recognition. In *Human-Computer Interaction – INTERACT 2015*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.). Springer International Publishing, Cham, 425–432.

[37] Yue Qin, Chun Yu, Zhaoheng Li, Mingyuan Zhong, Yukang Yan, and Yuanchun Shi. 2021. ProxiMic: Convenient Voice Activation via Close-to-Mic Speech Detected by a Single Microphone. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Tokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3411764.3445687

[38] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. 2016. EgoCap: Egocentric Marker-Less Motion Capture with Two Fisheye Cameras. *ACM Trans. Graph.* 35, 6, Article 162 (Nov. 2016), 11 pages. https://doi.org/10.1145/2980179.2980235

[39] Sami Ronkainen, Jonna Häkkilä, Saana Kaleva, Ashley Colley, and Jukka Linjama. 2007. Tap Input as an Embedded Interaction Method for Mobile Devices. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (Baton Rouge, Louisiana) *(TEI '07)*. Association for Computing Machinery, New York, NY, USA, 263–270. https://doi.org/10.1145/1226969.1227023

[40] T. Scott Saponas. 2009. Enabling Always-Available Input: Through on-Body Interfaces. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI EA '09)*. Association for Computing Machinery, New York, NY, USA, 3117–3120. https://doi.org/10.1145/1520340.1520441

[41] Marcos Serrano, Barrett M. Ens, and Pourang P. Irani. 2014. Exploring the Use of Hand-to-Face Input for Interacting with Head-Worn Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3181–3190. https://doi.org/10.1145/2556288.2556984

[42] Emi Tamaki, Takashi Miyak, and Jun Rekimoto. 2010. BrainyHand: A Wearable Computing Device without HMD and It's Interaction Techniques. In *Proceedings of the International Conference on Advanced Visual Interfaces* (Roma, Italy) *(AVI '10)*. Association for Computing Machinery, New York, NY, USA, 387–388. https://doi.org/10.1145/1842993.1843070

[43] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. 2019. xR-EgoPose: Egocentric 3D Human Pose from an HMD Camera. arXiv:1907.10045 [cs.CV]

[44] Cheng-Yao Wang, Min-Chieh Hsiu, Po-Tsung Chiu, Chiao-Hui Chang, Liwei Chan, Bing-Yu Chen, and Mike Y. Chen. 2015. PalmGesture: Using Palms as Gesture Interfaces for Eyes-Free Input. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 217–226. https://doi.org/10.1145/2785830.2785885

[45] Ruolin Wang, Chun Yu, Xing-Dong Yang, Weijie He, and Yuanchun Shi. 2019. EarTouch: Facilitating Smartphone Use for Visually Impaired People in Mobile and Public Scenarios. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 24, 13 pages. https://doi.org/10.1145/3290605.3300254

[46] Yue Wu, Tal Hassner, KangGeon Kim, Gerard Medioni, and Prem Natarajan. 2017. Facial landmark detection with tweaked convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 3067–3074.

[47] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, P. Fua, H. P. Seidel, and C. Theobalt. 2019. Mo2Cap2: Real-time Mobile 3D Motion Capture with a Cap-mounted Fisheye Camera. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 2093–2101. https://doi.org/10.1109/TVCG.2019.2898650

[48] Xuhai Xu, Haitian Shi, Xin Yi, WenJia Liu, Yukang Yan, Yuanchun Shi, Alex Mariakakis, Jennifer Mankoff, and Anind K. Dey. 2020. EarBuddy: Enabling On-Face Interaction via Wireless Earbuds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376836

[49] 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* (Gothenburg, Sweden) *(VRST '17)*. Association for Computing Machinery, New York, NY, USA, Article 19, 8 pages. https://doi.org/10.1145/3139131.3139146

[50] Yukang Yan, Chun Yu, Yingtian Shi, and Minxing Xie. 2019. PrivateTalk: Activating Voice Input with Hand-On-Mouth Gesture Detected by Bluetooth Earphones. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1013–1020. https://doi.org/10.1145/3332165.3347950

[51] Yukang Yan, Chun Yu, Xin Yi, and Yuanchun Shi. 2018. HeadGesture: Hands-Free Input Approach Leveraging Head Movements for HMD Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 198 (Dec. 2018), 23 pages. https://doi.org/10.1145/3287076

[52] Yukang Yan, Chun Yu, Wengrui Zheng, Ruining Tang, Xuhai Xu, and Yuanchun Shi. 2020. FrownOnError: Interrupting Responses from Smart Speakers by Facial Expressions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376810

[53] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. 2019. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 963–971. https://doi.org/10.1145/3332165.3347867

[54] Bo Yi, Xiang Cao, Morten Fjeld, and Shengdong Zhao. 2012. Exploring user motivations for eyes-free interaction on mobile devices. *Conference on Human Factors in Computing Systems - Proceedings* c (2012), 2789–2792. https://doi.org/10.1145/2207676.2208678

[55] Chun Yu, Xiaoying Wei, Shubh Vachher, Yue Qin, Chen Liang, Yueting Weng, Yizheng Gu, and Yuanchun Shi. 2019. HandSee: Enabling Full Hand Interaction on Smartphone with Front Camera-Based Stereo Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300935

[56] Adam Zarek, Daniel Wigdor, and Karan Singh. 2012. SNOUT: One-Handed Use of Capacitive Touch Devices. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (Capri Island, Italy) *(AVI '12)*. Association for Computing Machinery, New York, NY, USA, 140–147. https://doi.org/10.1145/2254556.2254583