PrivateTalk: Activating Voice Input with Hand-On-Mouth Gesture Detected by Bluetooth Earphones

Yukang Yan[1], **Chun Yu**[12†], **Yingtian Shi**[1], **Minxing Xie**[3]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Key Laboratory of Pervasive Computing, Ministry of Education
[3]College of Computer Science and Technology, Zhejiang University
{yyk15,shiyt16}@mails.tsinghua.edu.cn chunyu@tsinghua.edu.cn xieminxing@gmail.com

ABSTRACT
We introduce PrivateTalk, an on-body interaction technique that allows users to activate voice input by performing the Hand-On-Mouth gesture during speaking. The gesture is performed as a hand partially covering the mouth from one side. PrivateTalk provides two benefits simultaneously. First, it enhances privacy by reducing the spread of voice while also concealing the lip movements from the view of other people in the environment. Second, the simple gesture removes the need for speaking wake-up words and is more accessible than a physical/software button especially when the device is not in the user's hands. To recognize the Hand-On-Mouth gesture, we propose a novel sensing technique that leverages the difference of signals received by two Bluetooth earphones worn on the left and right ear. Our evaluation shows that the gesture can be accurately detected and users consistently like PrivateTalk and consider it intuitive and effective.

Author Keywords
Voice input; hand gesture.

CCS Concepts
•**Human-centered computing** → *Gestural input;*

INTRODUCTION
Using voice input to interact with computing devices has been consistently rated as a convenient and natural interaction method by users [18, 22]. Voice input is used in a wide range of tasks, including text entry, communication, and sending voice commands. However, there are two major challenges with voice input [32]. First, users worry about the privacy risks of disclosing their personal information while speaking. Second, they suffer the inconvenience of repeatedly speaking the wake-up word or pressing a button during multiple rounds of voice input.

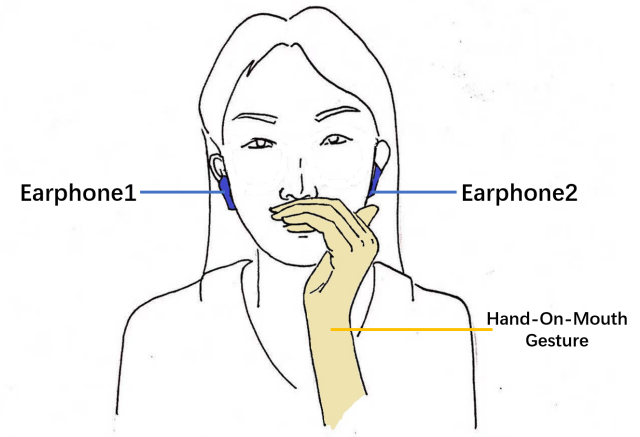

Figure 1. Illustration of the Hand-On-Mouth gesture: The user's hand covers the mouth from one side. PrivateTalk compares the audio inputs received by the two earphones for detecting this gesture.

With PrivateTalk we can address these two issues simultaneously. With PrivateTalk, a user can perform voice input by covering the mouth on one side with a hand while speaking (the Hand-On-Mouth gesture) as shown in Figure 1. While performing this gesture, the hand acts as a barrier between the user's mouth and one earphone while simultaneously enhancing the voice input to the other one. This causes a significant difference between the two audio inputs in amplitude and frequency characteristics and we leverage this to trigger voice input. This gesture naturally reduces the spread of speech while also concealing lip movements. The activation is convenient as the gesture is easy to perform and does not require visual attention. Furthermore, the gesture is intuitive to users as it is naturally used while whispering in daily life.

In this paper, we present an algorithm for detecting the Hand-On-Mouth gesture. The algorithm consists of synchronization and normalization of the two audio inputs followed by voice activity detection (VAD), feature extraction, and finally binary classification. We train an SVM-based binary classifier, using data collected in our first user study, to detect PrivateTalk. The model is evaluated in a second user study showing that the proposed algorithm can accurately detect PrivateTalk (98.33%). The study also shows that PrivateTalk mitigates users' privacy concerns, reduces disturbance to the public while also being rated by users as easy to learn and intuitive to perform.

† denotes the corresponding author.

UIST '19, October 20–23, 2019, New Orleans, LA, USA

ACM ISBN 978-1-4503-6816-2/19/10…$15.00
https://doi.org/10.1145/3332165.3347950

RELATED WORK

User Concerns about Voice Input

Inconvenience of activation and privacy risks are the two major concerns with voice input [32]. As [8] observed, users are conscious while transmitting private information via voice in public places. The Creative Strategies 2016 survey [1] showed that 39% of smartphone users performed voice input at home but only 6% of them used it in public places. Users worry about their speech being overheard by the people surrounding them. This creates a concern for disclosing their personal information as well as for being a disturbance to others. To address this issue, unnoticeable voice input techniques or silent speech interfaces [11, 27] were developed. However, these techniques require users to speak differently from the way we talk every day. In addition, to activate voice input, users are required to speak a wake-up word [13] or press a physical/software button before each input. This is inconvenient and tiresome, especially in multiple rounds of voice input. Furthermore, accidentally mentioning the wake-up word incorrectly activates the voice input causing further inconvenience to the user.

Sound Driven Activity Detection

Using sound to detect and classify real-time events has been well studied. Voice activity detection (VAD) algorithms have been developed to detect the presence of human speech [19, 28]. Human activities (e.g., using a blender) and emergency events (screaming, gunshots [3, 10]) can be detected based on features of the sound collected by one [15, 17, 24] or multiple microphones [16]. The features have also been used in combination with accelerometer data [30]. For classifying music, using features in the frequency domain, including Mel-frequency cepstral coefficients [7] and chroma features [25] have been studied. Our study differs from all the above in that PrivateTalk detects the Hand-On-Mouth gesture by comparing the difference between two audio inputs instead of modeling events by extracting features from a single input.

Combined Use of Voice and Hand Gesture

Combining voice input with hand gestures can provide greater expressive power, flexibility and convenience [14]. "Put that there" [5] was the first to use the pointing gesture to indicate the location mentioned in the voice command. Similar hand location and posture based augmentation were applied in many areas including structural biological analysis [23], autonomous car control [21] and human-robot interaction [12]. However, one major challenge faced by all these techniques is hand pose estimation [26]. To recognize hand gesture accurately, extra sensors are often required, including cameras [33], accelerometers, EMG sensors [34], and inertial sensors [31]. PrivateTalk demonstrates a new sensing method for hand gesture which does not require accurate hand pose estimation or any extra sensors on the hand. To sense the Hand-On-Mouth gesture, we detect the asymmetrical influence that the gesture has on the two audio inputs.

HAND-ON-MOUTH GESTURE DETECTION

We now describe the algorithm for detecting the Hand-On-Mouth gesture. The setup and the process is illustrated in Figure 2 and Figure 3 respectively.

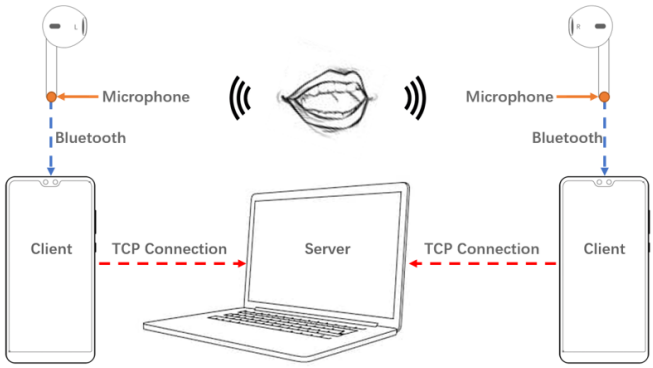

Figure 2. Illustration of the setup. Microphones on two earphones record the user's speech. Two earphones separately connect to two phones via Bluetooth connections. Two phones connect to a server running on a laptop via TCP connections.

Apparatus

There is a microphone on each earphone, however, a pair of earphones only activates one microphone during use. To record audio inputs from microphones on both ears simultaneously, we use the setup shown in Figure 2. Two pairs of earphones are connected to two different smartphones and users wear one left and one right earphone from different pairs. Thus we are able to use microphones on both ears. Smartphones receive audios from the earphones and transmit them to the server. Then the server analyzes the audio inputs and detects the Hand-On-Mouth gesture based on the result. Audio inputs are sampled at 48000 Hz.

Synchronization and Normalization

The analysis begins with the synchronization of two audio inputs. The audio inputs are transmitted to the server in the frames of every 4096 samples. We add a timestamp to each frame and synchronize the inputs by matching the frames with the nearest timestamps. Then we match audio inputs for five seconds to create a better synchronization. We split the audio inputs into arrays of short-term windows of 512 samples and calculated the Mel-scaled spectrogram [29] of the samples in each window. We perform Dynamic Time Warping (DTW) algorithm on two arrays of spectrograms. As Figure 4 demonstrates, we calculate the most frequent offsets between the aligned windows as the final offset of two audio inputs. We compensate for the offset in the incoming frames.

After the synchronization, we normalize the audio inputs. When a user first puts on the two earphones, there is an amplitude difference between them due to their specific positions and orientations on the ear. We calculate this difference as the baseline and normalize the two audio inputs to remove its influence on the detection of the Hand-On-Mouth gesture. For each pair of aligned windows after synchronization, we compute the ratio of the areas under two waves. We use the average of these ratios as the baseline ratio. We apply this baseline ratio on the incoming samples of two audio inputs to ensure when users speak without the Hand-On-Mouth gesture, amplitudes of two audios are of little difference.

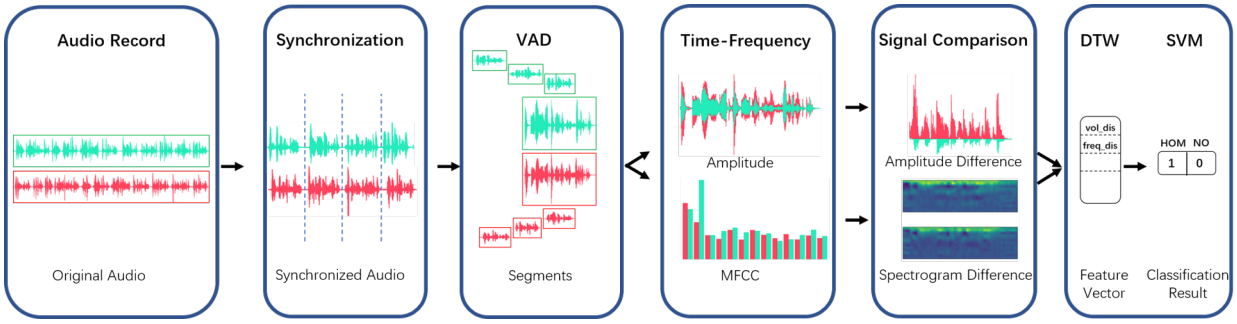

Figure 3. Overview of the Hand-On-Mouth gesture detection. Audio inputs of two earphones are labeled with red and green colors.

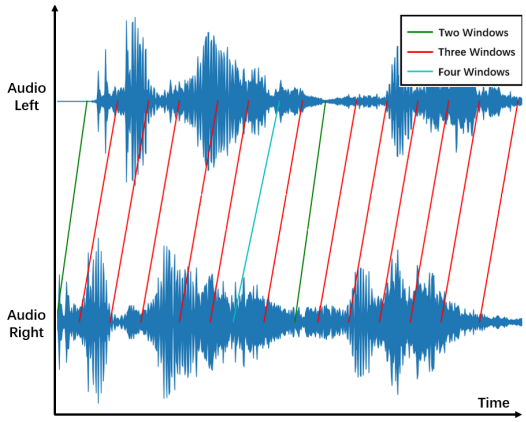

Figure 4. Synchronization of two audio inputs. The colored lines connect the aligned windows. 512 samples of data are between successive lines. In this example, the most frequent offset is three windows (red lines).

Voice Activity Detection

Audio segments which contain human speech are extracted by voice activity detection (VAD). As a result, PrivateTalk only reacts to voice input from users rather than environmental noises. The implementation of VAD has been well studied [9, 35] and thus we apply the state-of-the-art Google WebRTC VAD [2] in our implementation.

When users perform voice input with PrivateTalk, sometimes only one microphone can detect human speech because the other one is blocked by the hand and the audio is not strong enough. Our recognition results are robust even in such situations as we find that the quality of the audio measured from even one microphone is capable of supporting audio-to-text translation tasks. Thus, even if we detect human speech in one of the two microphones we continue with the gesture recognition task. Next, we extract the segments from both inputs for further analysis. Pairs of audio segments are generated after this process. Since we only use audio segments after the VAD, we are confident that our audio segment pairs contain human speech.

Feature Extraction

To compare the difference between the two audio segments, features in audio amplitude and frequency characteristics are extracted. Audio segments are split into windows of 20 ms with overlaps of 10 ms. We calculate the amplitude difference between two audio segments (Figure 3 - "Amplitude") as in Equation 1. The average amplitudes of the windows form two arrays as Amp_Left/Right. We perform the DTW algorithm to measure the degree of difference between two arrays, which has removed the influence of the segment length (the number of windows). We calculate the difference in Mel-frequency cepstral coefficients (MFCC) as in Equation 1. First, we erase the influence of amplitude difference by normalizing the amplitudes of the segments as in Equation 2. Then in each window, we calculate thirteen MFCC values with 40 filters and the FFT window length of 512 samples. This results in two matrices containing MFCC vectors of the windows (Figure 3 - "Spectrogram Difference"), which are MFCC_Left/Right. We also apply the DTW algorithm to measure the difference between matrices. Feature_Amplitude and Feature_MFCC form the final feature vector.

$$\begin{aligned} Feature\_Amplitude &= DTW(Amp\_Left, Amp\_Right) \\ Feature\_MFCC &= DTW(MFCC\_Left, MFCC\_Right) \end{aligned} \quad (1)$$

$$segment\_normalized = \frac{(segment - mean(segment))}{max(absolute(segment))} \quad (2)$$

Classification

Based on these features of the audio segments, we use an SVM-based binary classifier to detect whether users are performing the Hand-On-Mouth gesture while speaking. An SVM model with a linear kernel was trained with the data we collected (as described next in the USER TEST section). We use the SVM-based classifier with a linear kernel because the current input feature vector is simple, and the accuracy of the model is acceptable. In the future when more functionality, for example, user verification, is to be added, we can consider more complex models. Other machine learning models will be explored to improve detection accuracy.

USER TEST

The goal of this experiment was to test whether the Hand-On-Mouth gesture can be detected accurately and whether PrivateTalk can improve the user experience of voice input. We invited twelve users to speak voice commands in two conditions (PrivateTalk V.S. normal speech). We recorded the audio data and collected users' subjective feedback.

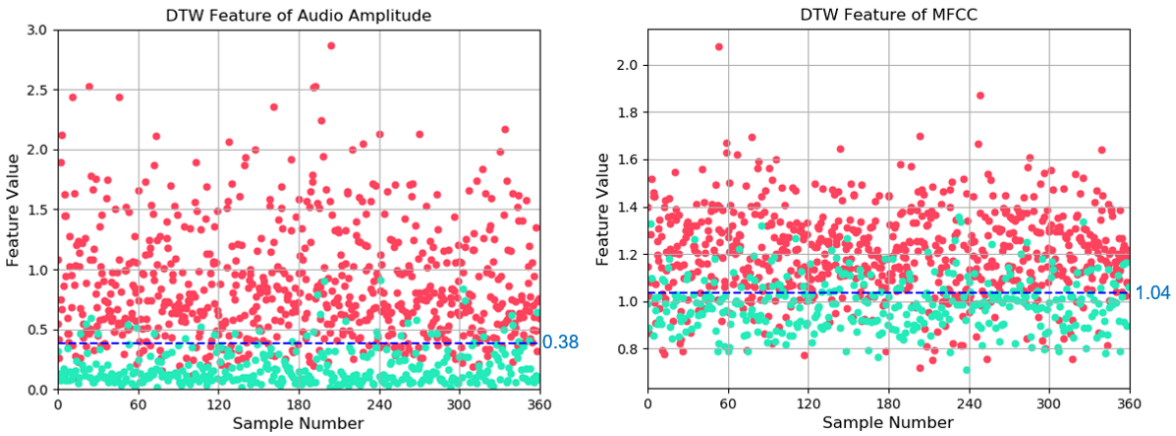

Figure 5. Distribution of the Feature_Amplitude and Feature_MFCC extracted from the collected audio data. Blue dash lines show the best empirical thresholds for distinguishing two types of audios. 720 samples of PrivateTalk conditions and 360 samples of normal speech condition are visualized.

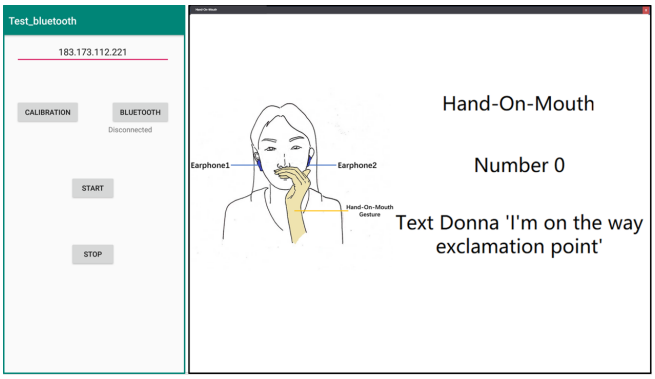

Figure 6. The user interfaces on the smartphones and the laptop. On the smartphone, "Bluetooth" button is used to connect the phone to the earphone; "Start" button is used to connect the phone to the server; "Calibration" button is to trigger the synchronization and normalization. On the laptop, the condition ("Hand-On-Mouth"), the index of the task ("Number 0") and the current voice command are shown.

Participants

We invited twelve users from a local campus to participate in this experiment. Four were female and eight were male. The average age was 23.75 (STD = 1.69). All users had used voice input and had known wake-up word method and button press method to activate voice input before this experiment.

Apparatus

We used the setup described in HAND-ON-MOUTH GESTURE DETECTION section (illustrated by Figure 2). We developed an Android application that ran on the phones to record and transmit the audios. We ran a server on the laptop, which received audio data from smartphones through TCP connections. The interface of the Android application is shown by Figure 6 - Left. The command set consisted of 30 frequently used voice commands [4]. We developed a program on the laptop to show the instructions to the participants. For each task, the current condition, the index of the task and the target command were shown (Figure 6 - Right) in this program. User sat comfortably during the whole experiment, which was conducted in a quiet office room.

Procedure

The experimenter introduced PrivateTalk interaction and the Hand-On-Mouth gesture to the participant. Then the participant put on two earphones. The applications on two phones were activated and connected to the server. After that, the participant spoke 30 commands in two conditions (PrivateTalk/normal speech). In PrivateTalk condition, the participant spoke each voice command twice with two directions of the Hand-On-Mouth gesture. Between two conditions, the participant took a rest. The experiment took fifteen minutes on average. Each participant filled in a questionnaire to report their subjective feedback and comments.

Classification Result

We generated 1080 pairs of audio segments of voice commands from twelve participants (720 in PrivateTalk condition and 360 in normal speech condition). First, we used empirical thresholds of Feature_Amplitude and Feature_MFCC to distinguish PrivateTalk and normal speech. We calculated the best thresholds to distinguish two types of speech. Using the threshold of 1.04 for Feature_MFCC, the precision and recall of PrivateTalk were 82.76% and 80.22%, and the averaged accuracy of PrivateTalk and normal speech was 81.83%. Using the threshold of 0.38 for Feature_Amplitude, the precision and recall of PrivateTalk were 94.25% and 94.44%, and the average accuracy was 92.52%. Then we trained an SVM model using both Feature_Amplitude and Feature_MFCC as the input features. In Leave-One-Out cross-validation, the precision and recall were 95.47% and 97.60%, and the average accuracy was 96.36%. The classification results showed that when the Hand-On-Mouth gesture was performed, the audios from two microphones had significant differences in both audio amplitude and frequency characteristics. We calculated the accuracy of PrivateTalk and normal speech respectively and reported the average value. This erased the influence of the unbalance of the data set.

Subjective Feedback

Participants scored for the experience of voice input with PrivateTalk on seven aspects. They provided ratings on the Seven-point Likert scale. The results are visualized in Figure 7.

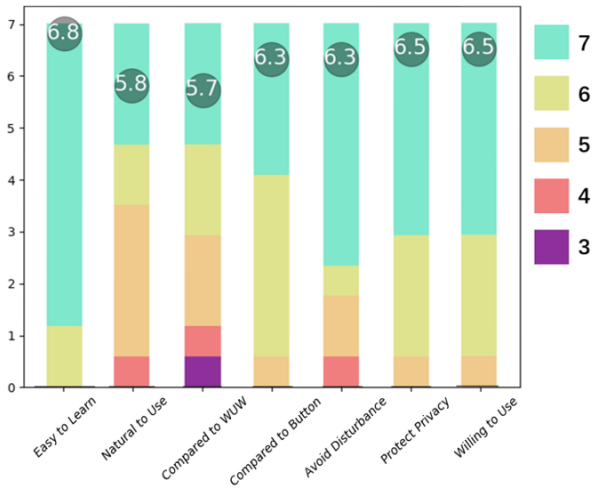

Figure 7. Distribution of subjective scores. White numbers in the circles are the averages of the scores.

The overall results showed that participants were very positive to the experience of PrivateTalk (all seven average scores are higher than 5.7). 11/12 participants rated the highest score for "Easy to Learn" and 11/12 rated positive scores for "Natural to Use". When comparing the activation methods, only one participant reported the preference of wake-up word over PrivateTalk. He explained that he felt calling the name (wake-up word) to be more polite to start a conversation. Other participants all preferred PrivateTalk than wake-up word or button press method. Participants also commented that compared to normal speech condition, PrivateTalk provided a stronger sense of safety and would cause less disturbance which would be helpful while performing voice input in public places.

EVALUATION

We invited another twelve new users to evaluate PrivateTalk in the environment of a real lab office, where there were sounds of keyboard typing and chatting as the environmental noises. We invited five extra participants to evaluate the disturbance felt by surrounding people while users perform voice input using PrivateTalk.

Participants

Twelve users who did not participate in the USER TEST section participated in this experiment. None of the training data of the SVM-based classifier that we used in this experiment was from these users. The average age was 23.83 (STD = 1.28). Six were male and six were female. All participants had used voice input before this experiment.

Apparatus

We used the same experimental setup as in the USER TEST section. The applied SVM-based classifier was trained with the speech data of all twelve participants. The experiment was conducted in a real lab office on a local campus. PrivateTalk ran in real-time in this experiment.

Procedure

The experimenter first introduced the PrivateTalk interaction and the Hand-On-Mouth gesture. The user put on earphones and connected the phones to the server by pressing the "Start" button. Then the user pressed the "Calibration" button and spoke three sentences ("Here is the first/second/third command") in a row. After that, the user commenced the 30 commands in both PrivateTalk and normal speech conditions. In PrivateTalk condition, the user was free to cover the mouth from either the left or the right. The order of two conditions was counter-balanced across users. After each command was spoken, if the algorithm detected the existence of the Hand-On-Mouth gesture, a sound of "OK" was played in the earphones as the response. Between two conditions, the user took a rest. During the rest, we queried the other five participants on whether they noticed or were disturbed by the user's voice inputs. They sat around the user who performed voice input and they read books during the experiment.

Results

We recorded 720 recognition results of user intention to activate PrivateTalk. Further, we collected the reports from five extra participants on the feeling of disturbance.

Accuracy

The overall classification accuracy was 98.33%. Errors included 8/360 PrivateTalk speeches that were recognized as normal speeches (activation failures) and 4/360 normal speeches that were recognized as PrivateTalk speeches (false positives). Three activation failures were caused by performing the Hand-On-Mouth gesture in a nonstandard way. One participant covered the mouth with the hand not being in contact with the face, and he encountered three activation failures in a row at the beginning. We instructed him to cover the mouth tightly from one side. Then he performed all the following trials successfully. All the other participants performed the standard gesture from the beginning. Although most participants intuitively cover the mouth tightly, it will be meaningful to measure the required degree of contact (i.e., how much gap between the hand and the mouth) for successful activation of PrivateTalk. We regard this factor as future work.

Disturbance

We asked the five extra participants to score for the disturbance level in Five-point Likert scale. The average score for PrivateTalk condition was 3.15 (STD = 0.23) and that for normal speech condition was 3.55 (STD = 0.19). It shows that the disturbance was reduced by the Hand-On-Mouth gesture as the score for PrivateTalk was lower. An interesting finding was that with PrivateTalk, although surrounding people noticed the user's voice input, they thought it to be more acceptable. They felt it more polite when users performed the Hand-On-Mouth gesture to lower the volume of voice. More importantly, the user's own experience was improved. As reported, they felt less embarrassed while speaking commands with PrivateTalk, especially for the commands that contained personal information, for example, "Text Donna I'm on the way" and the commands that they felt "silly", for example, "what is the result of 71 times 241?". Most users commented that they would use PrivateTalk to perform voice input in public places in real life.

DISCUSSION

In this paper, we propose PrivateTalk interaction, implement the detection algorithm, evaluate the user experience and recognition performance through user studies. Based on the results, we discuss the interaction potential of PrivateTalk and the devices that PrivateTalk can be applicable to.

Voice and Gesture

PrivateTalk demonstrates leveraging hand gestures to supplement voice-based interaction. Gestures provide increased power of expression and the ability to protect privacy. In our technique, the Hand-On-Mouth gesture is used to activate voice input and protect privacy simultaneously. Our study shows that hand gestures can be detected by their influence on the voice input data. In the future, by receiving audio inputs from more than two microphones we could expand the interaction space to include detection of more than one hand gesture. This can be done, for example, by adapting our current algorithm, supplementing the input feature vector with the correlation matrices between each pair of available microphones. These gestures can serve to trigger other functions than activating voice input, including indicating the purpose of voice input (e.g., send voice messages V.S. translate it into text). The number of gestures that can be detected using this method and the mappings between gestures and functions will be our future work.

Applicability of PrivateTalk

In our implementation, we receive audio inputs from two earphones and detect the Hand-On-Mouth gesture by comparing the difference between the inputs. This method can be applied to other computing devices including AR headsets and smart glasses which are equipped with multiple microphones at different locations. As discussed above, this can be used to allow for a larger set of gestures by adapting our current algorithm. Currently, we cannot activate both microphones of one pair of earphones. We expect this to be technically solved in the future and then PrivateTalk can be applied to off-the-shelf earphones. In addition, the always-on nature of PrivateTalk increases the power consumption of the earphones. We determine empirically that continuous use of approximately 3 hours can be supported after the battery of the earphones has to be fully charged.

LIMITATION AND FUTURE WORK

We discuss several limitations of PrivateTalk, which are mainly due to the current implementation. Accidental activation might occur if nearby people talk. For example, when a person talks to the user from the left, the audio input of the left earphone will be louder and the amplitude difference between two inputs might be large enough to trigger PrivateTalk. This can be solved by adding a user verification function [6, 20] which recognizes voiceprint and only responds to the authenticated user. The current implementation requires the user to cover the mouth tightly. If the user performs the Hand-On-Mouth gesture without contacting the face, the difference between two audio inputs may be not large enough for the activation. Although most users intuitively cover the mouth tightly (as discussed in the EVALUATION section), the required degree of contact for successful activation should be studied in the future. Evaluation in the wild should be conducted. As we tested, in the environment of a real lab, PrivateTalk can work properly. However, we can expect that in a noisier environment (e.g., in a restaurant), the noise may overwhelm the voice input from users. To overcome this, a more sophisticated voice filter should be applied. In addition, our training set for detecting the Hand-On-Mouth gesture only collected 30 commands. Although the detection algorithm is independent of the spoken utterances, we hope to collect a more comprehensive dataset in the future.

CONCLUSIONS

This paper presents PrivateTalk, a new on-body gesture-based technique for activating voice input. Users activate voice input by performing the Hand-On-Mouth gesture while speaking. Evaluation results showed that the gesture was accurately detected (98.33%); users thought PrivateTalk to be easy to learn, natural to adopt and useful for protecting their privacy; PrivateTalk also helped reduce the disturbance to the people around the user. Interestingly, PrivateTalk was valued not only for its disturbance mitigating property but also for improving the acceptability of using voice input in public areas. In the future, we will combine the use of PrivateTalk with other techniques including speaker verification and denoising algorithms, which will help in the ever increasing adoption of voice input for interaction.

ACKNOWLEDGEMENT

The authors thank all participants. This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001200, the Natural Science Foundation of China under Grant No. 61672314 and No. 61572276, and also by Beijing Key Lab of Networked Multimedia.

REFERENCES

[1] 2017. Creative Strategies. Website. (2017). Retrieved March 26, 2019 from `https://creativestrategies.com/voice-assistant-anyone-yes-please-but-not-in-public/`.

[2] 2019a. Google WebRTC VAD. Website. (2019). Retrieved March 26, 2019 from `https://webrtc.org/`.

[3] 2019b. ShotSpotter. Website. (2019). Retrieved March 26, 2019 from `https://www.shotspotter.com/`.

[4] 2019c. Siri. Website. (2019). Retrieved March 26, 2019 from `https://www.apple.com/siri/`.

[5] Richard A Bolt. 1980. *"Put-that-there": Voice and gesture at the graphics interface*. Vol. 14. ACM.

[6] William M Campbell, Douglas E Sturim, and Douglas A Reynolds. 2006. Support vector machines using GMM supervectors for speaker verification. *IEEE signal processing letters* 13, 5 (2006), 308–311.

[7] Jianfeng Chen, Alvin Harvey Kam, Jianmin Zhang, Ning Liu, and Louis Shue. 2005. Bathroom activity monitoring based on sound. In *International Conference on Pervasive Computing*. Springer, 47–61.

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

[9] Florian Eyben, Felix Weninger, Stefano Squartini, and Björn Schuller. 2013. Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 483–487.

[10] Pasquale Foggia, Nicolai Petkov, Alessia Saggese, Nicola Strisciuglio, and Mario Vento. 2015. Reliable detection of audio events in highly noisy environments. *Pattern Recognition Letters* 65 (2015), 22–28.

[11] Masaaki Fukumoto. 2018. SilentVoice: Unnoticeable Voice Input by Ingressive Speech. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 237–246. DOI: http://dx.doi.org/10.1145/3242587.3242603

[12] Jun Hu, Zhongyu Jiang, Xionghao Ding, Taijiang Mu, and Peter Hall. 2018. VGPN: Voice-Guided Pointing Robot Navigation for Humans. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 1107–1112.

[13] VZ Këpuska and TB Klein. 2009. A novel wake-up-word speech recognition system, wake-up-word recognition task, technology and evaluation. *Nonlinear Analysis: Theory, Methods & Applications* 71, 12 (2009), e2772–e2789.

[14] David Michael Krum, Olugbenga Omoteso, William Ribarsky, Thad Starner, and Larry F Hodges. 2002. *Speech and gesture multimodal control of a whole Earth 3D visualization environment*. Technical Report. Georgia Institute of Technology.

[15] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. In *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, 213–224.

[16] Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic sensors: Towards general-purpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3986–3999.

[17] Hong Lu, Wei Pan, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. 2009. SoundSense: scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 165–178.

[18] François Portet, Michel Vacher, Caroline Golanski, Camille Roux, and Brigitte Meillon. 2013. Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects. *Personal and Ubiquitous Computing* 17, 1 (2013), 127–144.

[19] Javier Ramırez, José C Segura, Carmen Benıtez, Angel De La Torre, and Antonio Rubio. 2004. Efficient voice activity detection algorithms using long-term speech information. *Speech communication* 42, 3-4 (2004), 271–287.

[20] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. 2000. Speaker verification using adapted Gaussian mixture models. *Digital signal processing* 10, 1-3 (2000), 19–41.

[21] Pablo Sauras-Perez, Andrea Gil, Jasprit Singh Gill, Pierluigi Pisu, and Joachim Taiber. 2017. *VoGe: A Voice and Gesture System for Interacting with Autonomous Cars*. Technical Report. SAE Technical Paper.

[22] Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope. 2010. âĂIJYour Word is my CommandâĂİ: google search by voice: A case study. In *Advances in speech recognition*. Springer, 61–90.

[23] Rajeev Sharma, Michael Zeller, Vladimir I Pavlovic, Thomas S Huang, Zion Lo, Stephen Chu, Yunxin Zhao, James C Phillips, and Klaus Schulten. 2000. Speech/gesture interface to a visual-computing environment. *IEEE Computer Graphics and Applications* 20, 2 (2000), 29–37.

[24] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras. 2012. Audio-based human activity recognition using Non-Markovian Ensemble Voting. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. 509–514. DOI:http://dx.doi.org/10.1109/ROMAN.2012.6343802

[25] Johannes A Stork, Luciano Spinello, Jens Silva, and Kai O Arras. 2012. Audio-based human activity recognition using non-markovian ensemble voting. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 509–514.

[26] Steven Strachan, Roderick Murray-Smith, and Sile O'Modhrain. 2007. BodySpace: Inferring Body Pose for Natural Control of a Music Player. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems (CHI EA '07)*. ACM, New York, NY, USA, 2001–2006. DOI:http://dx.doi.org/10.1145/1240866.1240939

[27] Ke Sun, Chun Yu, Weinan Shi, Lan Liu, and Yuanchun Shi. 2018. Lip-Interact: Improving Mobile Device Interaction with Silent Speech Commands. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 581–593. DOI: http://dx.doi.org/10.1145/3242587.3242599

[28] S Gökhun Tanyer and Hamza Ozer. 2000. Voice activity detection in nonstationary noise. *IEEE Transactions on speech and audio processing* 8, 4 (2000), 478–482.

[29] Amirsina Torfi. 2018. Speechpy-a library for speech processing and recognition. *arXiv preprint arXiv:1803.01094* (2018).

[30] Jamie A Ward, Paul Lukowicz, Gerhard Troster, and Thad E Starner. 2006. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE transactions on pattern analysis and machine intelligence* 28, 10 (2006), 1553–1567.

[31] Jian Wu, Lu Sun, and Roozbeh Jafari. 2016. A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors. *IEEE journal of biomedical and health informatics* 20, 5 (2016), 1281–1290.

[32] Kuan-Ning Wu. 2016. Voice Assistant. (2016).

[33] Mohammed Yeasin and Subhasis Chaudhuri. 2000. Visual understanding of dynamic hand gestures. *Pattern Recognition* 33, 11 (2000), 1805–1817.

[34] Xu Zhang, Xiang Chen, Yun Li, Vuokko Lantz, Kongqiao Wang, and Jihai Yang. 2011. A framework for hand gesture recognition based on accelerometer and EMG sensors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 41, 6 (2011), 1064–1076.

[35] Xiao-Lei Zhang and Ji Wu. 2013. Deep belief networks based voice activity detection. *IEEE Transactions on Audio, Speech, and Language Processing* 21, 4 (2013), 697–710.