Whoosh: Non-Voice Acoustics for Low-Cost, Hands-Free, and Rapid Input on Smartwatches

Gabriel Reyes, Dingtian Zhang, Sarthak Ghosh, Pratik Shah, Jason Wu
Aman Parnami, Bailey Bercik, Thad Starner, Gregory D. Abowd, W. Keith Edwards
{*greyes*, *dingtianzhang*, *sarthak.ghosh*, *pratikshah*, *jasonwu*, *amanparnami*, *baileybercik*, *thad*, *abowd*, *keith*} *@gatech.edu*
School of Interactive Computing, College of Computing, Georgia Institute of Technology

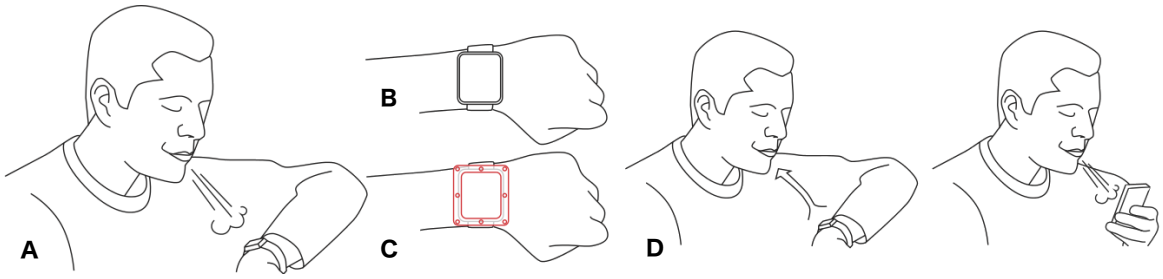

Figure 1: (A) Whoosh is an interaction technique that captures non-voice acoustic input (e.g., blowing, shooshing, other dynamic events), (B) using a commodity smartwatch without modifications and (C) with a custom-designed passive watch case. (D) Our technique enables low-cost and rapid input, including multi-device events such as "sip" on the watch and "puff" on the phone.

ABSTRACT

We present an alternate approach to smartwatch interactions using non-voice acoustic input captured by the device's microphone to complement touch and speech. *Whoosh* is an interaction technique that recognizes the type and length of acoustic events performed by the user to enable low-cost, hands-free, and rapid input on smartwatches. We build a recognition system capable of detecting non-voice events directed at and around the watch, including blows, sip-and-puff, and directional air swipes, without hardware modifications to the device. Further, inspired by the design of musical instruments, we develop a custom modification of the physical structure of the watch case to passively alter the acoustic response of events around the bezel; this physical redesign expands our input vocabulary with no additional electronics. We evaluate our technique across 8 users with 10 events exhibiting up to 90.5% ten-fold cross validation accuracy on an unmodified watch, and 14 events with 91.3% ten-fold cross validation accuracy with an instrumental watch case. Finally, we share a number of demonstration applications, including multi-device interactions, to highlight our technique with a real-time recognizer running on the watch.

ISWC '16, September 12-16 2016, Heidelberg, Germany.

ACM 978-1-4503-4460-9/16/09...$15.00.
DOI: http://dx.doi.org/10.1145/2971763.2971765

Author Keywords

Interfaces; wearable computing; smartwatches; interaction techniques; non-voice acoustics; hands-free; on-body input.

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces — Input Devices and Strategies.

INTRODUCTION

The emergence of smart devices (e.g., mobile phones, smartwatches, and head-up displays) is redefining the way we access data and produce information through everyday microinteractions [2], interactions that take less than four seconds to initiate and complete. The primary input modality for the smartwatch and mobile phone is touch. Touch offers expressive multi-touch capabilities and is intuitive. For example, recent work demonstrates the possibility of performing text entry with a smartwatch on-screen keyboard, using statistical decoding and error correction [9]. However, touch input on the small screen of a watch still requires targeted visual attention and a free hand for interaction. Traditionally, occlusion and fat-finger selection errors are two common challenges that hinder the use of these small screens [14, 27].

With advances in connectivity and computing, phones and smartwatches are capable of near real-time speech recognition. Speech provides a fluid and hands-free way of communicating intent and commands to a smart device. However, speech may be tedious and not well suited to certain microinteractions, such as repetitive input, scrolling, or swiping. In this paper, we present an approach to smartwatch input using non-voice acoustics to supplement touch and speech. Our input modality

opens up opportunities for hands-free input on small screen devices and also has implications for assistive technologies. The "sip-and-puff" technique is popular for wheelchair controls [7] and inspires some of our work. Our event set includes blow events as well as other acoustically unique sounds (e.g., shoosh, double blow, long blow) produced by modulating the shape of the mouth, tongue and throat. Non-voice acoustic input on the smartwatch can be subtle, and with the device in proximity to the mouth, can also be performed quietly or in environments with high ambient noise.

Our event set and applications are designed around familiar metaphors from mouse, touch and physical interactions. This design consideration facilitates mapping non-voice acoustic events to intuitive actions on the device. For example, a localized blow on the bezel can be used to click a corner icon, air swipes are useful for directional commands in the interface, and sip-and-puff is used to "absorb" content and "deliver" it to another device or application. Our system runs in real-time and can be installed on commodity mobile platforms that are equipped with a microphone. Further, through a completely passive watch case modification, we can facilitate robust recognition of an expanded set of input events.

Our work makes the following contributions:

- We describe an interaction technique using non-voice acoustic input for smartwatch interactions that enables low-cost, hands-free, and rapid input.
- We introduce the use of passive, 3D-printed smartwatch cases to expand the expressivity of events by introducing air swipes, circular blows, and bezel blows.
- We provide empirical evidence of our recognition system performance and limitations, through studies with 8 participants in the laboratory and 4 participants in-the-wild.

RELATED WORK

Interaction Techniques for Wrist-Worn Devices

Several custom-hardware solutions sense the surrounding surface area of the wristwatch. Prior work includes approaches with bio-acoustics [1, 6, 11], electromyography (EMG) [26], capacitance [23], pressure sensing [5], proximity sensing [17], and vision [15]. These solutions are capable of providing a diverse input vocabulary, but suffer from limitations such as bulkiness of hardware, occlusions for line-of-sight solutions, cost and complexity. Instead, we use the microphone already present in most commodity devices to enable new sensing capabilities with a diverse event set.

Prior work also focuses on developing custom watch devices. Facet provides a multi-display wristband consisting of multiple independent smartwatch screens, enabling a rich set of touch input techniques [18]. Xiao et al. provide a multi-degree-of-freedom mechanical watch face that supports continuous 2D pan and twist, as well as binary tilt and click [30]. Oakley et al. use proximity sensing around the watch face to capture interaction on the edge of small devices [19]. WatchIt introduces a custom watch band for eyes-free interaction [22]. While modifying the case or band around a smartwatch with electronics may provide additional interaction capabilities, it will also place varying degrees of power constraints on a device with limited battery capacity. We present the use of a passive 3D-printed watch case — dubbed "FluteCase" — to increase the expressivity of our event set, with no additional demands on battery or computation. While our technique does require an active microphone and continuous analysis, the majority of smartwatches today are already "always-on" for hotword detection (e.g., "Ok Google"). It could be possible to modify this device functionality to include recognition of Whoosh events.

Non-Voice Acoustic User Interfaces

Speech offers an expressive alternative to on-screen input, but non-speech acoustics may also provide a secondary input channel. Various types of non-speech input such as humming and whistling are used to provide continuous input [28], and Igarashi et al. demonstrate how duration, pitch and tonguing of sounds are used for interactive controls [13]. Closely related, others present interaction techniques using prosodic features of speech and non-verbal metrics [10, 20]. Sakamoto et al. propose a technique to augment touch interactions on a mobile device with non-voice sounds as a parallel input modality [25].

Blowing is another type of non-voice acoustic interaction in the literature, used for selection, gaming, entertainment, accessibility, or text entry [8]. Zyxio's SensaWaft uses a MEMS-based sensor array in a headworn microphone to detect blowing, enabling bidirectional controls for scrolling, zooming, and rotating a button dial [29]. BLUI is a fingerprinting technique that localizes where a person is blowing on a laptop screen and demonstrates accuracy of over 95% for 4x4 regions with a single microphone [21]. Our work is inspired by BLUI's initial results, and Whoosh seeks to not only localize blowing on a different form factor with a significantly smaller screen but also to capture how a person is acoustically interacting with the device. Blowatch proposes blowing air at a smartphone prototype with four external microphones simulating smartwatch interactions [3], and presents a taxonomy for blowing events on a watch. We use and extend this taxonomy in our event set, by including blows and other types of non-voice acoustic input, while implementing our system on a commodity smartwatch with a single microphone.

INTERACTION TECHNIQUES WITH UNMODIFIED WATCH

We describe the space of input events we explore with an unmodified smartwatch equipped with a single monophonic microphone, and draw analogies to common mouse and touch inputs.

Directed Blows

The *short blow* is the most basic event in our set with a quick blow toward the center of the watch screen. Based on empirical data from pilot studies, the typical length of a short blow is 200ms. In general, the spectrogram for this event (shown in Figure 2A) exhibits saturation when blowing normally with the device close to the mouth. For our users, we observed 10-20cm to be the typical distance between the mouth and the device while blowing at the smartwatch. The *double blow* extends the recognition of a discrete blow while capturing consecutive short blows directed at the center of the screen,

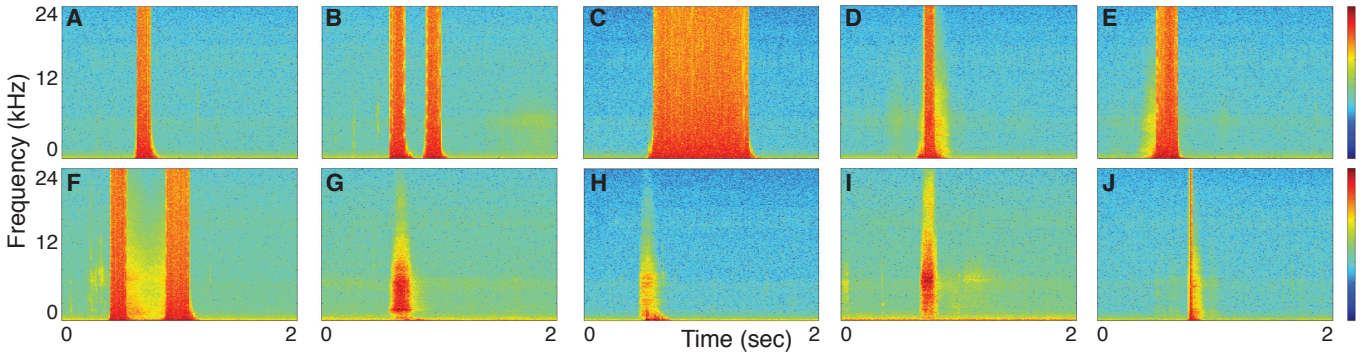

Figure 2: Spectrogram figures for unmodified watch events. The events displayed are: (A) short blow, (B) double blow, (C) long blow, (D) swipe up, (E) swipe down, (F) clockwise blow, (G) shoosh, (H) open exhale, (I)-(J) sip-and-puff.

lasting between 400-500ms (see Figure 2B). The double blow is analogous to a double click. The *long blow* consists of a continuous blow aimed at the center of the watch screen, typically longer than 500ms based on pilot study data (see Figure 2C). The long blow is analogous to a press-and-hold interaction from touch events.

Air Swipes

Similar to swipe gestures in touch-based interactions, air swipes are directional events captured as air passes over the watch screen and wind noise is captured by the microphone (located at the bottom center of the bezel). Typical length for swipes is 300ms. A *swipe up* is a continuous blow from bottom-to-top of the bezel across the screen. Conversely, the *swipe down* begins at the top of the bezel and ends at the bottom. See Figures 2D-E. A *circular blow* is a continuous swipe performed in a clockwise direction around the bezel. The blow starts and ends at the bottom center of the bezel, where the microphone is located. Based on training data, we observed circular blows lasting up to 1 second (see Figure 2F).

Non-Voice Sounds

A *shoosh* sound is produced by modulating a blow with curled tongue and pursed lips. A shoosh is typically used to indicate a form of silence or dismissal in the interface. Typical length is about 200ms (see Figure 2G). An *open-mouth* consists of the user exhaling toward the watch screen with their mouth open. This action is similar to fogging your eyeglasses with your breath. Typical length is about 300ms (see Figure 2H). A *sip* is performed with pursed lips similar to using a drinking straw. Compared to other events, the sip is an inhale and can be used to indicate directionality away from the device. A *puff* accompanies the sip event. A strong "p" sound distinguishes the puff from a short blow. Typical length for sip-and-puff is about 200ms (see Figures 2I-J).

WHOOSH

Whoosh runs in real-time on the smartwatch and performs audio recognition on incoming microphone data.

Theory of Operation

The main parts of voice and acoustic production are the lungs, the larynx or vibrator, and the resonator system. Air is exhaled out of the lungs and passes through the larynx, which contains the vocal folds. For blow events, air passes through relaxed folds and lung capacity determines the forcefulness of the blow. For non-voice sounds, the airstream passes between the vocal folds as they vibrate between 100Hz to 1kHz. The muscles in the larynx control the pitch based on the length and tension of the vocal cord. As the folds vibrate, they produce a buzzing sound at different frequencies, similar to the mouthpiece of a trumpet. The resonator system, consisting of the throat, nose, and mouth, alter the pathway to produce human speech and other sounds, similar to the structures of a musical instrument.

Our system focuses on both the wind noise detected by the microphone while blowing, as well as non-speech human sounds. Depending on the forcefulness of a blow or non-voice event, proximity to the microphone, and the direction of the user's mouth, we observe different phenomena. A *blow* event may produce either a broadband frequency response through the microphone or exhibit distortion from clipping caused by non-linear behavior of the electronic components and power supply limitations (Figures 2A-F). We use this distortion to our advantage to minimize false positives and uniquely identify particular events. Other events such as *shoosh*, *open*, and *sip-and-puff* exhibit distinct spectral patterns with energy up to approximately 10kHz (Figures 2G-J). After isolating an event and extracting features based on its frequency response, we infer the type and location of the event based on pre-trained audio fingerprinting using a machine learning classifier.

Implementation

We use an LG G (Android) Watch with a single microphone located at the bottom center of the bezel of its 1.65 inch touch screen, as well as a Motorola Droid Turbo (Android) smart phone to explore multi-device interaction. The microphones on both devices are sampled at 48kHz using the default microphone source, 16-bit PCM encoding, with no audio gain or noise suppression. TarsosDSP[1] handles audio management and recording. The library delivers a *float[]* audio buffer at preset frame intervals for processing in real-time.

Segmentation: This task focuses on determining when an event of interest occurs within the audio stream. For training

[1]https://github.com/JorenSix/TarsosDSP

purposes, users are prompted to provide input and audio is recorded for each event individually in 2-second windows. In our offline analysis, we implement a form of silence detector using a rolling variance to isolate the beginning and end of the event in each audio file. We use a forward and backward threshold search to account for events with silence occurring during the event (e.g., double blow), and empirically determine a threshold robust to noise around the event. We then expand the trimmed window outward around the isolated data by a preset number of frames to ensure full capture of the event and pass it to our feature extractor. For our real-time pipeline, we use a silence detector based on the energy of an audio frame (approximately 20ms). We maintain a buffer of audio frames that comprise an input event and use a heuristic timing threshold when silence is detected during an actual event (e.g., double blow). Once the event input buffer is full, the audio data is passed to the feature extractor.

Feature extraction: In order to capture directional events, we divide our segmented signal into two window slices of equal length. Dividing the audio signal in two halves aids in capturing salient features occurring about the center of the event. Mel-frequency cepstral coefficients (MFCCs) are a set of acoustic features modeling the human auditory system's non-linear response to different spectral bands. We calculate a 26-dimension MFCC with band edges from 0Hz to half the sampling rate at 24kHz. We calculate the sum of each MFCC coefficient for all frames (20ms frame, 10ms overlap) in each half of the audio signal, with the energy as the first coefficient. The MFCC vectors for each half add up to a total of 52 features. We use an additional 26 features based on the deltas of the MFCC coefficients. The features are normalized for classification. We run principal component analysis (PCA) on these features to facilitate our real-time classification.

Classification: We use a support vector machine (SVM) algorithm trained using Weka's sequential minimal optimization (SMO) implementation with a cubic polykernel and default parameters.

TECHNICAL EVALUATION FOR UNMODIFIED WATCH

We conduct a technical evaluation of our interaction techniques with an unmodified watch in the usability lab of our institution. Eight participants (5 male, 3 female, ages 22-34) are part of our user study. Participants wear the watch on the left hand. To begin, researchers perform a demonstration of each technique. Participants familiarize themselves with our data collection application with a practice round. During the evaluation, a visual stimulus is presented to the participant on the watch screen prompting them to perform a given event. Audio is recorded for 2 seconds after the prompt, with a one second pause between events. Each event is recorded in an individually labeled audio file for segmentation. The participants perform four rounds of data collection for 10 events. In each round, participants perform 5 examples for each event. The order of the stimulus is randomized across each round. In summary, our user study includes 8 participants x 10 events x 4 rounds x 5 samples per event for a total of 1600 event samples. We discard a total of four instances where the researcher observes the participant perform the wrong event or our segmentation determines the event is not fully captured within the time window. Participants are allowed to rest, drink water, remove the watch, or leave the room between sessions if desired.

Per-User and General Classifiers

We evaluate our technique by applying 10-fold cross validation on each individuals' collected instances. The overall average per-user accuracy across 8 participants and 10 events is 90.5% (sd=3.9%). P1 achieves the highest accuracy at 98.5% and P7 achieves the lowest accuracy at 86.0%. We observe that P7 held the device farther from the mouth than other participants, roughly more than 20cm. The distance away from the mouth results in weaker signals at the microphone making it difficult to distinguish between events. We present the confusion matrix of our results in Figure 3. The lowest precision of 78.8% is observed for the double blow event, mostly confused with a short blow. In some cases, participants perform the double blow quickly, effectively being recognized as a short blow. The shoosh event achieves the highest accuracy at 98.8%. We also evaluate how our system generalizes across users. Preliminary leave-one-participant-out analysis (i.e., test with one participant, train with the rest) across 8 participants and 10 events results in overall accuracy of 71.3% (sd=7.2%).

Evaluation of Activation Event In-The-Wild

We demonstrated our recognizer is capable of discriminating 10 events with our in-lab study and our technique is feasible on smartwatches available today. However, to minimize unintentional activation during real-world use, activation events are designed to distinguish intentional interactions from everyday activities. Activation events should ideally be extremely resistant to false positives while achieving high recognition rates [24]. Once the system is activated, all other input events

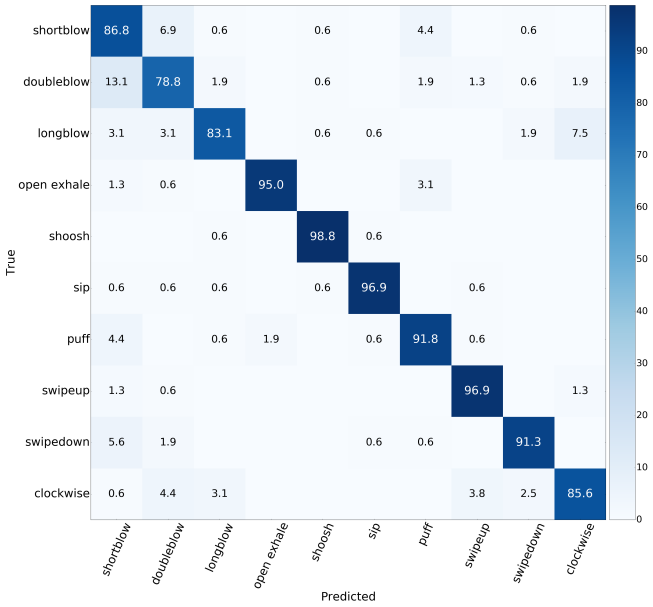

Figure 3: Confusion matrix averaged across all users (in %). Rows represent ground truth and columns are predicted values.

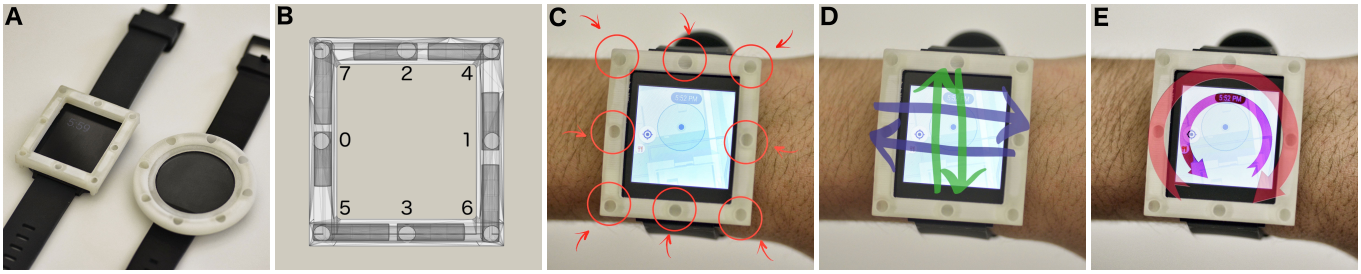

Figure 4: FluteCase and interactions. (A) FluteCase designs for both square and round watch faces, (B) Translucent rendering of the square FluteCase's 3D model with tube labels, (C) Bezel blows, (D) Air swipes, and (E) Circular blows.

are recognized. We design an activation trigger consisting of a double blow. While double blow presents some confusion with our classifier (see Figure 3), we believe it would be robust against detecting noise. We present results for false negatives with double blow performed in-the-wild by 4 participants, and false positives with noise only recorded by 4 researchers.

False positives (fp): We record smartwatch ambient data where there is no intentional input from 4 researchers on this project. In total, we record 11hr:36min of ambient audio at 48kHz sampling rate. Our data collection is limited only by the battery life of the smartwatches used, and the longest recording is approximately 3 hours on a LG G Watch. We apply a highpass filter above 15kHz to isolate any activation event that exhibits clipping and remove most ambient noise below 4-7kHz. We then apply a peak detection algorithm to identify double blow events. Our recognizer mislabels noise as a double blow event 15 times, resulting in a 1.3 fp/hour rate. Most confusion is observed during hand washing at the sink and several forceful coughs.

False negatives (fn): We recruit 4 participants from our previous study to perform the double blow activation gesture in-the-wild. We ask each participant to wear the watch for at least 4 hours during the day and perform the double blow when prompted. Our application prompts participants by vibrating the watch and presenting on-screen feedback. Prompts are delivered using a random Poisson process with an average delivery time of 4 minutes. To preserve battery, we record only one minute of audio data after prompting the user. In total, the four participants were prompted 174 times. We discard 22 missed instances where the participant did not perform the gesture, based on visual analysis and acoustic inspection of the data. Our peak detection search algorithm correctly identified the double blow 149 out of the remaining 152 instances, resulting in 98.0% accuracy.

DISCUSSION & LIMITATIONS FOR UNMODIFIED WATCH

During our initial exploration into the design of our event set, we experimented with swipes in all directions (i.e., up, down, left, and right). Given that we use a smartwatch with a single front-facing microphone at the bottom center of the bezel, we found intuitively that the location of the microphone was key to discriminating between events. A continuous blow approaching from the left or the right and passing over the microphone appear symmetrical, and thus are difficult to discriminate in the recognizer. In contrast, swipes up and down begin either at or away from the microphone, making it easier to recognize them as unique events.

Additionally, we also experimented with localizing directional blows on the arm to the left and right sides of the watch, as well at the top, bottom, left, and right target areas on the bezel. In prior work [21], researchers localize up to 5x5 events with a single microphone on a laptop screen roughly an order of magnitude larger than a smartwatch screen. However, on our platform, the single microphone and small size of the watch did not provide the ability to readily disambiguate such inputs. To address these limitations and expand the Whoosh vocabulary, we design a custom 3D-printed watch case.

FLUTECASE: A PASSIVE 3D-PRINTED WATCH CASE

FluteCase is a custom 3D-printed watch case for both square and circular smartwatches that alters the acoustic response of blowing events on and around the smartwatch. The case provides a low-cost and entirely passive (meaning no electronics nor battery usage) means of expanding the range of inputs that are recognized by our system. In this section, we describe the phenomena and inspiration for our design of passive modification of the physical structure of the watch, based on wind instruments and prior work altering the speaker-microphone pathway on mobile phones [16].

Acoustic Phenomena

When air is blown into a tube-shaped resonator, standing waves are created that cause the air to vibrate and produce sound. For closed pipe wind instruments like ours, the pitch of the vibration is determined by the length of the tube. For example, the Greek pan flute has multiple tubes with different lengths open at one end for blowing and is closed at the other end. Closed pipe resonators do not require finger operation and their fundamental air resonant frequencies are defined by:

$$f = \frac{v}{\lambda} \quad \text{[Hz]} \quad (1)$$

where f is the air resonant frequency, v is the speed of sound, $\lambda = kL$ is the wavelength, where k is a constant determined by open or closed pipe and L is the length of the pipe. Generally, the shorter the pipe is, the higher the resonant frequencies produced.

Design of the FluteCase

We draw inspiration from the structure of closed pipe instruments to design our 3D-printed FluteCase. We develop both

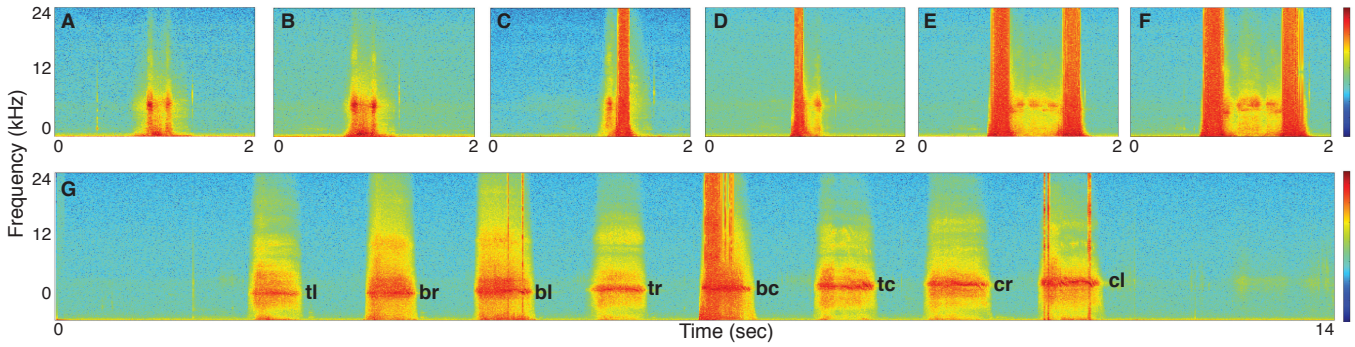

Figure 5: Spectrograms for FluteCase events: (A) swipe left, (B) swipe right, (C) swipe down, (D) swipe up, (E) clockwise, (F) counterclockwise, (G) bezel blows (labeled on the figure) starting from the lowest to highest resonant frequency.

square and round watch versions to suit a variety of commercial devices (see Figure 4A). The cases have 8 closed pipe tubes of different lengths, each with an open hole. The tubes' "head" (the end with the open hole) and "tail" (the closed end) are connected to each other. In the case of a circular smartwatch, the head and tail form a ring shape around the watch display. A base that fits the shape and size of the watch bezel attaches tightly to the watch. The eight tubes are designed to resonate at eight distinct frequencies between 2kHz to 10kHz, allowing blows near particular regions of the watch face to be readily disambiguated. For replicability, we describe the dimensions of the square case used during our user evaluation. The overall width, length, and depth of the square case are 45.60 mm, 51.06 mm, and 5.58 mm, respectively. The diameter of each hole is 4.05 mm. The width of each circular tube is constant at 4.096 mm. The length of each tube is defined by:

$$L = 14.956 * 2^{\frac{i}{12}} \qquad \text{[mm]} \quad (2)$$

where L is the length of each tube as a function of i, which denotes the ith tube (labeled in Figure 4B).

INTERACTION TECHNIQUES WITH FLUTECASE

The FluteCase design greatly expands the range of non-voice acoustic interactions with smartwatches, allowing recognition of an additional 6 swiping blows and 8 bezel blows. A blow event over each FluteCase hole creates a slightly audible tone generated by the airflow entering the resonator tube. We use the same recognition pipeline described for the unmodified watch scenario, as our segmentation is adaptive to variable event lengths. Bezel blows and swiping blows are shown visually in Figures 4C-E.

Swiping Blows

Blowing over two or more FluteCase holes in a swiping fashion enables six additional input events. *Air swipes* are single blows across the watch face traversing two holes in the following directions: left-right, right-left, top-bottom, or bottom-top. *Circular blows* are swipes along the edge of the watch, traversing all holes, in a clockwise or counterclockwise direction beginning at the bottom center location. The spectrograms for all swiping blows are shown in Figures 5A-F.

Bezel Blows

Bezel blows are discrete events performed by the user in a single-action blowing at one of the eight holes distributed evenly around the watch case. *Corner* bezel blows consist of a continuous blow at one of the four corner targets of the watch case. These events are: topleft (tl), topright (tr), bottomleft (bl), bottomright (br). The next set of events are *D-pad* bezel blows. In this spatial arrangement, a continuous blow is directed at FluteCase target locations emulating a D-pad keypad configuration. These events are the remaining bezel locations: topcenter (tc), centerleft (cl), centerright (cr), and bottomcenter (bc). The spectrogram for bezel blows starting from the lowest to highest resonant frequency is shown in Figure 5G.

TECHNICAL EVALUATION FOR FLUTECASE

We conduct a technical evaluation of our new event set with a FluteCase-mounted smartwatch using the same device, participants, and pipeline as the previous study. Our data collection for this condition includes 8 participants x 14 events x 4 rounds x 5 samples per event for a total of 2240 event samples. We discard a total of 61 instances (roughly less than 5 out of 160 samples per event) in which participants either perform the wrong event or the event is not fully captured within the time window. We evaluate how the system performs on a per-user level using 10-fold cross validation and how it generalizes across participants using leave-one-participant-out analysis.

For 10-fold cross validation, the average accuracy across 8 users and 14 events is 91.4% (sd=5.3%). P10 achieves the highest accuracy at 96.4% and P7 achieves the lowest accuracy at 80.4%. We present the confusion matrix of our results in Figure 6. The lowest precision of 81.6% is observed for both the clockwise and counterclockwise events. Both of these are more complex gestures that require blowing over all eight bezel locations. The bottomcenter event is the most accurate with accuracy of 99.4%. We suggest the main reason for the highest accuracy is that the microphone is located directly underneath the bottomcenter bezel hole, resulting in a clearer signal. Preliminary leave-one-participant-out analysis across 8 participants and 14 events results in overall accuracy of 79.7% (sd=9.7%).

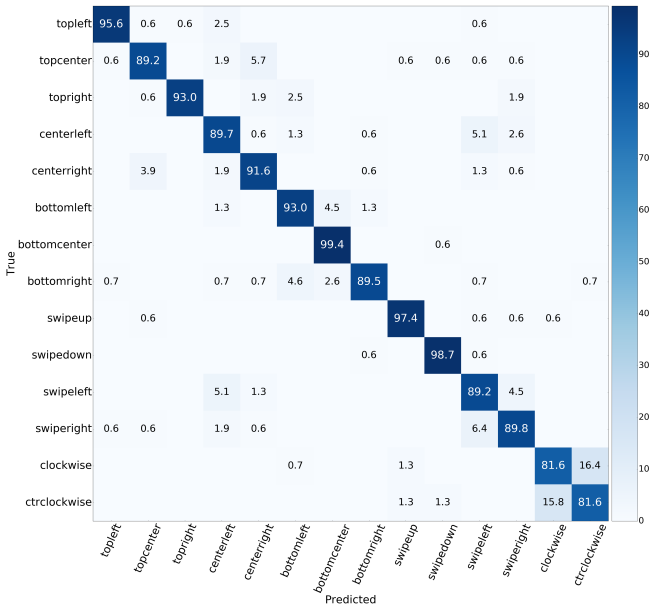

Figure 6: Confusion matrix for FluteCase across all users (in %). Rows are ground truth and columns are predicted values.

DEMONSTRATION APPLICATIONS

We implement several demonstration applications that highlight the potential for Whoosh interactions, with both the unmodified watch and FluteCase. We refer the reader to the video accompanying this paper for live demonstrations of each application.

Unmodified Watch Applications

Notifications: Quick access to notifications or quick actions without having to use a mobile phone is arguably one of the most compelling features of a smartwatch. Our notifications app shows examples of how Whoosh facilitates such interactions. Our app enables *discrete selection* between one or two buttons on-screen. We use our event recognizer to silence or dismiss an incoming call notification with a shoosh event, and answer the call with a single blow acknowledgment.

Authentication: A person can also use a sequence of Whoosh events as an additional layer of security on their devices. The smartwatch can automatically lock whenever the user removes it from the wrist. In our application, a lock screen pops up and a pre-determined sequence of Whoosh events is used to unlock the device. Whoosh events on the watch could also be used as a physical authentication challenge to complete a purchase on another device (e.g., mobile phone or desktop).

Speech + Whoosh: Whoosh events can be combined with speech to create a mixed interaction modality. In our messaging application, a user dictates the content of a text message and uses Whoosh events to manipulate the text. A long blow is used to backspace and a short blow is used to send the message when complete. The user quickly mode switches between speech and Whoosh input using a double blow, or could potentially use a flick of the wrist.

Multi-Device Handoff: When Whoosh is run in parallel on both the smartwatch and phone, it can enable a robust set of multi-device events. For example, we explore interactions between the watch and the phone held in the same hand. Inspired by the stitching technique [12] and Duet [4], we support two multi-device events: watch-to-phone and phone-to-watch *sip-and-puff*. Sip-and-puff provides an intuitive metaphor to transfer tasks from one device to another. In our demonstration, a sip event on the watch "absorbs" content on the watch screen and a puff event remotely delivers the content to the phone. This allows, for instance, a user who receives an email notification on the watch to transfer and view the entire message on a larger device.

FluteCase Applications

Maps: Whoosh enhances navigation on a map by providing the following actions. Panning the map is enabled by *bezel blows*. In our application, the map shifts in the direction towards the FluteCase hole that the user blows. Continuously blowing into the same hole could keep the map moving in that direction. A total of eight panning directions are enabled with the FluteCase. Zooming is enabled by *circular blows*. In our demonstration, a circular blow in the clockwise direction will zoom in the map while the counterclockwise direction will zoom out. An *air swipe* up or down allows the user to traverse layers of hierarchical content. In our application, a swipe down reveals the various map views (e.g., satellite, terrain) and a swipe up returns up the stack.

Application Shortcuts: Smartwatches are intended to minimize the time between intent and action [2]. In our demonstration, eight app icons are displayed on the watch home screen aligned with the FluteCase holes. The user blows at any of the FluteCase target locations to open the associated app on the watch itself or potentially on the mobile phone.

DISCUSSION AND FUTURE WORK

Using audio for interaction can always present potential privacy concerns as the device may capture spoken input from the user. Whoosh focuses on non-speech audio recognition based on extracted features and does not store any raw audio. Furthermore, the Whoosh recognizer is lightweight and runs in real-time on the device. Thus, we do not require sending audio to the cloud for additional computational power and processing.

Whoosh is well-suited as a complementary input modality for smartwatches. A multimodal approach enables more complex and potentially parallel forms of input. "Chording" or combining Whoosh events with touch, speech, or motion provides a new set of fluid interactions. One potential example includes "clutch" mechanisms. An air swipe might be used to trigger sending an SMS. A flick of the wrist after the event could cancel, or a touch down during the event could immediately confirm the intended action. Such a mechanism provides a lightweight confirmation step for microinteractions that are irreversible.

The Whoosh recognizer running on a commodity watch without modification enables various simple microinteractions. FluteCase enables an expanded set of interactions but requires

modifying the physical structure of the watch. To achieve a richer vocabulary, there is a trade-off between passive approaches such as FluteCase and other solutions at the hardware level. Potential opportunities, which we have not yet explored, include increasing the number of microphones or altering microphone placement.

We have begun exploring the use of Whoosh in-the-wild. Our initial evaluation of our activation event with the unmodified watch assesses false positives and negatives. Our segmentation is tuned to be robust to noise (as demonstrated in our video figure). For future work, we want to further assess how environmental sounds, such as wind and other unforeseen ambient noise, may potentially impact our system. We could also redesign the FluteCase to minimize the pathway for ambient sounds. Improving the machine learning pipeline could also address these issues. User variability across events is another challenge. Personalization of our classification models to dynamically incorporate new users' data may improve user independent performance.

CONCLUSION

Whoosh is a sensing technique that uses non-voice acoustic input for microinteractions on smartwatches. Our system exploits the unique signature of sounds generated by the user to enable low-cost, hands-free, and rapid input on commodity devices. We evaluate the performance of our unmodified watch recognizer with 8 participants and 10 events. Our recognition system achieves 90.5% accuracy using a single classifier and per-user cross validation models. Using FluteCase, our 3D-printed passive case around the smartwatch, we alter the acoustic response captured by the microphone to enable 14 additional interactions and achieve 91.3% accuracy with per-user cross validation. We detect and classify non-voice acoustic signals in real-time on the device. We conclude with a set of example applications that highlight our technique and demonstrate the design space and opportunities enabled by Whoosh.

ACKNOWLEDGMENTS

This work is generously supported by a Google Faculty Research Award and PhD Fellowship. Special thanks to Kent Lyons, collaborators at Georgia Tech ECE and Music Technology, our anonymous reviewers, and user study participants for their contributions.

REFERENCES

1. B. Amento, W. Hill, and L. Terveen. The Sound of One Hand: A Wrist-mounted Bio-acoustic Fingertip Gesture Interface. *CHI EA '02*. 724–725.
2. D. L. Ashbrook. 2010. Enabling Mobile Microinteractions. Ph.D. Dissertation. Georgia Institute of Technology, Atlanta, GA, USA.
3. W.-H. Chen. Blowatch: Blowable and Hands-Free Interaction for Smartwatches. *CHI EA '15*. 103–108.
4. X. A. Chen, T. Grossman, D. J. Wigdor, and G. Fitzmaurice. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. *CHI '14*. 159–168.
5. A. Dementyev and J. A. Paradiso. WristFlex: Low-power Gesture Input with Wrist-worn Pressure Sensors. *UIST '14*. 161–166.
6. T. Deyle, S. Palinko, E. S. Poole, and T. Starner. Hambone: A Bio-Acoustic Gesture Interface. *ISWC '07*. 3–10.
7. L. Fehr, W. E. Langbein, and S. B. Skaar. Adequacy of Power Wheelchair Control Interfaces for Persons with Severe Disabilities: A Clinical Survey. *J. of Rehab R&D '00*. 353–360.
8. J. F. Filho, W. Prata, and T. Valle. Advances on Breathing Based Text Input for Mobile Devices. *Universal Access in Human-Computer Interaction '15*. 279–287.
9. M. Gordon, T. Ouyang, and S. Zhai. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. *CHI '16*. 3817–3821.
10. S. Harada, J. A. Landay, J. Malkin, X. Li, and J. A. Bilmes. The Vocal Joystick:: Evaluation of Voice-based Cursor Control Techniques. *Assets '06*. 197–204.
11. C. Harrison, D. Tan, and D. Morris. Skinput: Appropriating the Body as an Input Surface. *CHI '10*. 453–462.
12. K. Hinckley, G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith. Stitching: Pen Gestures That Span Multiple Displays. *AVI '04*. 23–31.
13. T. Igarashi and J. F. Hughes. Voice As Sound: Using Non-verbal Voice Input for Interactive Control. *UIST '01*. 155–156.
14. A. K. Karlson and B. B. Bederson. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. *INTERACT '07*. 324–338.
15. D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. *UIST '12*. 167–176.
16. G. Laput, E. Brockmeyer, S. E. Hudson, and C. Harrison. Acoustruments: Passive, Acoustically-Driven, Interactive Controls for Handheld Devices. *CHI '15*. 2161–2170.
17. G. Laput, R. Xiao, X. A. Chen, S. E. Hudson, and C. Harrison. Skin Buttons: Cheap, Small, Low-powered and Clickable Fixed-icon Laser Projectors. *UIST '14*. 389–394.
18. K. Lyons, D. Nguyen, D. Ashbrook, and S. White. Facet: A Multi-segment Wrist Worn System. *UIST '12*. 123–130.
19. I. Oakley and D. Lee. Interaction on the Edge: Offset Sensing for Small Devices. *CHI '14*. 169–178.
20. A. Olwal and S. Feiner. Interaction Techniques Using Prosodic Features of Speech and Audio Localization. *IUI '05*. 284–286.
21. S. N. Patel and G. D. Abowd. BLUI: Low-Cost Localized Blowable User Interfaces. *UIST '07*. 217–220.
22. S. T. Perrault, E. Lecolinet, J. Eagan, and Y. Guiard. Watchit: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets. *CHI '13*. 1451–1460.
23. J. Rekimoto. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. *ISWC '01*. 21–27.
24. J. Ruiz and Y. Li. DoubleFlip: A Motion Gesture Delimiter for Mobile Interaction. *CHI '11*. 2717–2720.
25. D. Sakamoto, T. Komatsu, and T. Igarashi. Voice Augmented Manipulation: Using Paralinguistic Information to Manipulate Mobile Devices. *MobileHCI '13*. 69–78.
26. T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay. Enabling Always-Available Input with Muscle-Computer Interfaces. *UIST '09*. 167–176.
27. K. A. Siek, Y. Rogers, and K. H. Connelly. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. *INTERACT '05*. 267–280.
28. A. J. Sporka, S. H. Kurniawan, M. Mahmud, and P. Slavik. Non-speech Input and Speech Recognition for Real-time Control of Computer Games. *Assets '06*. 213–220.
29. H. Wallop. CES 2010: breath-controlled mobile phones to be made? Telegraph.co.uk. Accessed: 2016-07-14. 2010.
30. R. Xiao, G. Laput, and C. Harrison. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. *CHI '14*. 193–196.