SoundTrak: Continuous 3D Tracking of a Finger Using Active Acoustics

CHENG ZHANG, Georgia Institute of Technology
QIUYUE XUE, Peking University
ANANDGHAN WAGHMARE, Georgia Institute of Technology
SUMEET JAIN, Georgia Institute of Technology
YIMING PU, Georgia Institute of Technology
SINAN HERSEK, Georgia Institute of Technology
KENT LYONS, Technicolor Research
KENNETH A. CUNEFARE, Georgia Institute of Technology
OMER T. INAN, Georgia Institute of Technology
GREGORY D. ABOWD, Georgia Institute of Technology

The small size of wearable devices limits the efficiency and scope of possible user interactions, as inputs are typically constrained to two dimensions: the touchscreen surface. We present SoundTrak, an active acoustic sensing technique that enables a user to interact with wearable devices in the surrounding 3D space by continuously tracking the finger position with high resolution. The user wears a ring with an embedded miniature speaker sending an acoustic signal at a specific frequency (e.g., 11 kHz), which is captured by an array of miniature, inexpensive microphones on the target wearable device. A novel algorithm is designed to localize the finger's position in 3D space by extracting phase information from the received acoustic signals. We evaluated SoundTrak in a volume of space ($20cm \times 16cm \times 11cm$) around a smartwatch, and show an average accuracy of 1.3 cm. We report on results from a Fitts' Law experiment with 10 participants as the evaluation of the real-time prototype. We also present a set of applications which are supported by this 3D input technique, and show the practical challenges that need to be addressed before widespread use.

CCS Concepts: • **Human-centered computing → Sound-based input / output**; *Graphics input devices*;

Additional Key Words and Phrases: Acoustic, 3D input, Wearable, Finger Tracking

ACM Reference format:
Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A. Cunefare, Omer T. Inan, and Gregory D. Abowd. 2017. SoundTrak: Continuous 3D Tracking of a Finger Using Active Acoustics. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 2, Article 30 (June 2017), 25 pages.
DOI: http://doi.org/10.1145/3090095

This work is partially supported by Georgia Tech Wearable Computing Center Engagement Grant. Author's addresses: C. Zhang, School of Interactive Computing, Georgia Tech; Q. Xue, EECS Department, Peking University; A. Waghmare and S. Jain and Y. Pu, School of Interactive Computing, Georgia Tech; S. Hersek, School of Electrical and Computer Engineering, Georgia Tech; K. A. Cunefare, The George W. Woodruff School of Mechanical Engineering, Georgia Tech; O. T. Inan,School of Electrical and Computer Engineering, Georgia Tech; G. D. Abowd, School of Interactive Computing, Georgia Tech.

2474-9567/2017/June-ART30 $15.00
DOI: http://doi.org/10.1145/3090095

1 INTRODUCTION

The small physical size of wearable devices limits the user experience, preventing full engagement with wearable technology compared to smartphones, tablets, or laptops. Since user-device interaction is currently dominated by touch-based methods, the size of the touchscreen relative to the finger imposes significant restrictions on the interaction. Simple operations, such as pressing a button to answer a call can be performed; but, more powerful interactions such as multi-touch or gesture-based text input are difficult. There is a need to extend the input to a larger space that has the potential to support a rich set of input gestures.

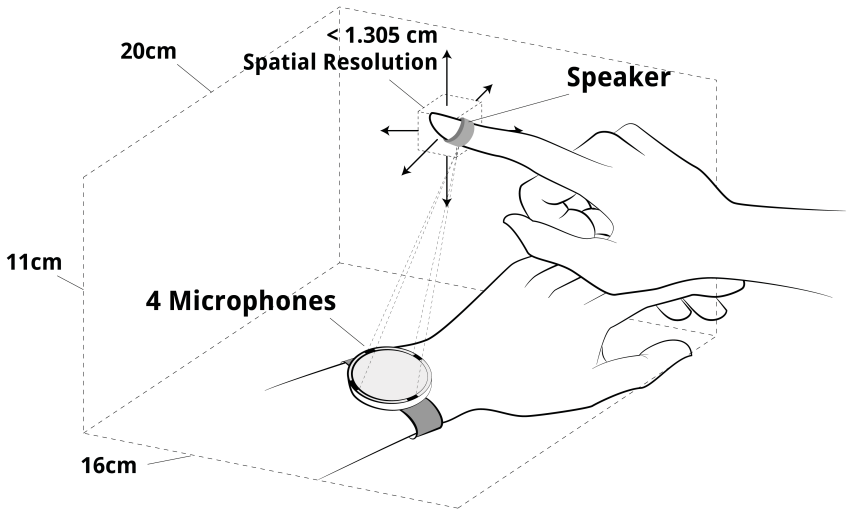

Fig. 1. The SoundTrak System

To address this challenge, researchers have already expanded the interaction to a larger space around the watch with different sensing modalities, mostly on the skin around the watch [33, 35] or watch bezel and band [34]. In this paper, we consider allowing inputs in the 3D volume of space around a wearable device.

There are a few different existing technologies to track an object in 3D space, with computer-vision based tracking (including the use of depth) being the most popular due to the widespread use of cameras. But to track the object, the camera needs to be placed in a specific orientation which limits the space for interaction. Furthermore, processing video data places strict requirements on the hardware (e.g., processing unit, battery, memory), which is currently still a limit for wearable devices. Researchers have demonstrated the possibility of locating a finger's position in 3D space using fully passive magnetic sensing [5]. However, the magnetic field attenuates dramatically with distance, limiting the range for such a solution.

To overcome these challenges, we present SoundTrak, a novel approach based on active acoustic sensing to expand the human-wearable input operation space from two to three dimensions. Our proposed method hinges on the use of a micro-speaker on the finger or entity to be tracked (e.g., a smart pen), and an array of low-cost microelectromechanical systems (MEMS) microphones placed on a wearable device (such as a smartwatch).

By tracking the accumulated phase shift of the measured signal on each element of the MEMS microphone array, we can compute the distance between the finger with the micro-speaker and the device in 3D space in real-time. SoundTrak calculates the absolute x,y,z displacement of the finger to the device with a derived physics model. Thus, no machine learning training is required prior to using this system. Based on our system evaluation, SoundTrak increases the interaction space from 5 cm^2 (the average area of a smartwatch screen) to 3520 cm^3 (the volume around the device), with an average spatial resolution of 1.3 cm, as shown in Figure 1. Our technique can be used to expand the design space not only for wearable devices (e.g., smartwatches, Google Glass, virtual reality devices) but also for many other surfaces (e.g., tabletop, blackboard).

Our paper presents the following contributions:

(1) An active acoustic sensing technique that tracks the position of the finger or entity to be tracked in 3D space.
(2) A physics-based model that derives the finger's 3D position from the phase information of the received acoustic signals.
(3) A system evaluation that demonstrates the system can track the finger's position in a 3D volume around the wearable device measuring 20cm x 16cm x 11cm (see Figure 1) with an average accuracy of 1.305 cm.
(4) A Fitts' Law based user study with 10 participants that shows the usability of our technique.
(5) A discussion of potential applications that can be supported by our system and future challenges before widespread adoption.

2 RELATED WORK

Our technique employs acoustic sensing to track the finger position, thus enabling 3D-finger-movement based inputs for wearable devices. Therefore, we review the related work in the following four categories: novel inputs for wearable technology, acoustic sensing for novel gesture input, object localization by using acoustic signals or radio frequency signals, and continuous finger tracking for wearable devices.

2.1 Novel Inputs for Wearable Technology

The wearable computing community has worked on providing appropriate input technology for tiny portable computers for years. These projects usually instrument different parts of the body with additional hardware to provide discrete input gestures. Twiddler [17] designed a one-hand chording keyboard. Some researchers built a band-like device, which allows the users to perform a set of gestures using the hands or performing gestures with the arm [6, 9, 14, 27]. Others designed devices which can be worn on the hand such as a data glove [1] or a ring on the finger [4].

Due to the recent popularity of smartwatches in the market, many research projects have been developed to solve the challenges pertaining to input on a smartwatch. Researchers have developed technology to shrink the touch point size of the finger [32], enlarge the input area [13, 33, 35], or extend input to the apparatus of the watch [16, 34]. However, the input challenges still remain because some of the solutions are currently impractical for use with wearable devices while the input gestures in others are relatively limited.

2.2 Acoustic Sensing for Novel Gesture Inputs

Using acoustics to detect input gestures has been demonstrated on wearable devices. Some projects, such as such as Skinput [9, 15] and TapSkin [33], use passive acoustic sensing capturing the acoustic signature of certain gestures. Others use active acoustic sensing to detect on-skin gestures [21]. Many recent projects have taken advantage of the Doppler Effect to recognize gross movements of the hand [2, 8, 26, 29]. DopLink [2] also uses the Doppler Effect for rapid device pairing, based on the hypothesis that the intended target device will receive the maximum frequency shift, that is, the hand will move towards the target device.

All of the above projects are based on detecting discrete input gestures, and most of them use machine learning techniques. Recognizing only discrete input gestures limits the potential gesture design space of these techniques. Furthermore, for machine learning based techniques, the training required before using the system may be unpleasant for users and may not be generalizable. The SoundTrak solution presented in this paper uses a physics-based understanding of phase shifts to derive location of the tracked speaker, resulting in a generalized solution that requires no training and can provide continuous tracking to support a broader set of gestures to detect.

2.3 Object localization using acoustic signals or radio frequency signals

Acoustics has been widely used for distance calculation for many years, especially for indoor localization. Most of these technologies actively send a chirp or a pulse signal from the sender to the receiver. The location is calculated based on the amount of delay time of the pulse. However, the resolution usually varies from room level to meter level. For instance, Cricket [25] sends a concurrent radio frequency signal and ultrasonic signals to infer distance, and it provides room-level localization by comparing the time delay between radio signal and ultrasonic signal. Swadloon [11] provides meter-level indoor localization of the phone by using both the phase information and the Doppler effect. Meter-level accuracy, however, is not good enough for continuous tracking of an input gesture for a wearable device, which is our goal in this work.

In order to recognize gestures, researchers built devices with active ultrasonic technology. One of the earliest project proposing this concept was Sonic Pen [3], which involved placing the speaker in a pen to generate a sonic pulse that was received by an orthogonal pair of microphones. The distance from the sonic pen to the microphones can be measured by comparing the received pulses at different receivers. Unfortunately, the published study on Sonic Pen only presents an estimate of the resolution achievable without providing any experimental results. When the tracked device becomes small and wireless, it is usually challenging to provide synchronized signals, which are essential for calculating the time of travel between the speaker and receivers. To overcome this limit, infrared signal [28] and radio frequency (RF) signals [7] are used to synchronize between devices. Since these signals travel at the speed of light, comparing the arrival of the sound signals against the infrared/RF signals allows for a derivation of time travel of the sound. There are also some other methods based system for 3D track with high precision. Mao et al. describe [19] a high-precision Acoustic motion Tracker system (CAT) for 3D track, which uses a distributed Frequency Modulated Continuous Waveform (FMCW) to derive the change of distance to the speaker when the mobile moves from one position to another.

Recent projects also use the echo effect to detect gestures, such as Chirp [20], FingerIO [22] and others [30]. Chirp shows the ability to detect hand gestures in 3D space by measuring the elapsed time between transmitted ultrasonic signal and the received echo. It is not clear how Chirp would work for finger gesture detection, because the fingers are much smaller than the overall hand and may, subsequently result in a less significant echo effect.

SoundTrak uses phase information to calculate distance, and previous work has explored that concept. Recently, Low-Latency Acoustic Phase (LLAP) has been used to detect the sound signal reflected by a moving hand or finger [30]. The phase change of echo signals caused by the hand or finger is used to detect and classify gestures. Though this technology does not require the instrumentation of the finger, LLAP can only track 1D and 2D movement and it is not clear whether it is capable of tracking hand movement in a relatively large area. Some phase-based RF systems have also been proposed [10, 18, 24]. These systems require relatively large transmitters and higher power, suitable for room-level localization but not for wearable input.

2.4 Continuous finger tracking for wearable input

One of SoundTrak's contributions is continuous tracking in a relatively large volume above the smartwatch. Tracking of fingers in a large 3D volume is not unique to SoundTrak. Popular commercial solutions (e.g., Leap Motion and Kinect) use computer-vision and depth sensors. While there are relatively small depth sensors

Table 1. Track Technology Comparison

	Implementation	Application	Scale	Resolution	Resolution3/Scale	Sender Receiver Communication	2D or 3D
Cricket	Time Delay	Indoor Localization	Room Size	1.2m	/	No	2D
Swadloon	Doppler Effect	Indoor Localization	Room Size	0.5m	/	No	3D
Sonic Pen	Time Delay	Graphic Input	Not Mentioned	theoretically 0.2mm	/	Yes	3D
Ultrasound Position Input Device	Time Delay	Graphic Input	Not Mentioned	Not Mentioned	/	Yes	3D
SkinTrak	Electricity Phase Machine Learning	Touch Track	Arm and Hand	7.6mm	/	No	2D
FingerIO	Echo Detection	Finger Track	$500mm \times 250mm$	8mm	0.05%	No	2D
UTrack	Magnetic Sensing	Finger Track	$70mm \times 30mm \times 60mm$	4.84mm	0.08%	No	3D
SoundTrak	**Sound Phase**	**Track**	**200mm × 160mm × 110mm**	**13.05mm**	**0.06%**	**No**	**3D**

available now, it remains a challenge to embed those cameras/sensors into the current wearable devices on the market. Our empirical validation with SoundTrak involves a relatively straightforward microphone array that we formed into a casing which fits around an existing smartwatch. It would not be difficult to integrate such a microphone array into the design of a new smartwatch.

Other sensing modalities have also been utilized to develop continuous finger tracking technology. SkinTrack [35] is able to track the finger position on the skin by passing electricity through the body. However, it is hard to extend this technique to the 3D space above the skin because air has very low electrical conductivity. FingerIO [22] extends the input space for a smartwatch around the watch by detecting the echo pattern using the OFDM technique, but technique only works for a 2D space. UTrack [5] can track the finger movement in the space volume of 38400 mm^3 with a mean euclidean error of 24.54 mm for one-handed gestures using magnetic sensing. However, because the intensity of the magnetic field attenuates cubically with distance, it is very challenging to track the finger position in a larger area.

2.5 Comparison

Table 1 provides a comparison between SoundTrak and other related work discussed above. Compared with the above continuous finger tracking approaches, SoundTrak provides a 3D finger tracking technology working within the largest volume (3250,000 mm^3) and providing a low mean euclidean error (13.05 mm) without requiring communication between receivers and the sender. Furthermore, it only requires four microphones and a speaker, which can be tiny in size.

3 THEORY OF OPERATION

SoundTrak is built upon a well-understood physical relationship: the further the distance that a sound travels, the more delay is introduced on the receiver side because of the speed of sound. This delay can be accurately detected both by directly measuring time-of-flight or by examining the phase shift of the signal. The time delay introduced in sound propagation has been used previously to detect the distance from a sound source to a receiver in many different scenarios, for example in sonar-based location detection in 2D space [23].

Some research projects calculate the location of an object using the delay of an ultrasonic pulse or chirp. However, to retrieve the delay of the chirp, the sender needs to communicate the sent time of each pulse or chirp to the receiver, which may be challenging between wearable devices. Because such a communication may require additional hardware, it would increase the size and weight as well as the power consumption of a wearable device. In addition, to distinguish the consecutive pulses or chirps on the receiver, a chirp-based method may have limits on the number of pulses or chirps can be sent per second.

SoundTrak is a new 3D position calculation method that tracks the accumulated phase shift of a received acoustic signal. It does not require communication between the speaker and receivers because it can derive the

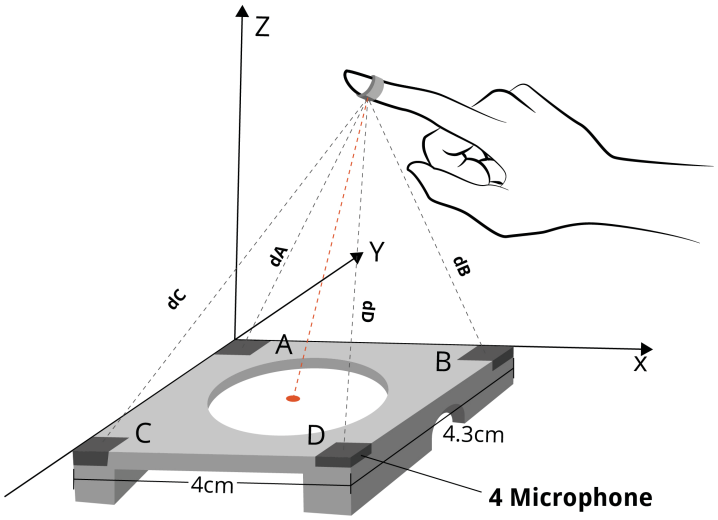

Fig. 2. The physics model

signal distance exclusively at the receiver using phase information. Thus, it is possible to build a miniature device to transmit a single frequency acoustic signal with relatively low energy consumption.

In the following sections, we will present the underlying model and a detailed algorithm to implement the physics model.

3.1 Geometric Model

Figure 2 shows the geometric relationship in 3D space between the finger-mounted speaker and a custom designed rectangular-shaped watch case. We use A $(x_a, y_a, 0)$, B $(x_b, y_b, 0)$, C $(x_c, y_c, 0)$, and D $(x_d, y_d, 0)$ to denote the corners of the watch case where the receivers are placed, and use A $(x_a, y_a, 0)$ as the grid origin of the coordinate system. The distance from the speaker on the finger to each receiver is denoted as dA, dB, dC, and dD. SoundTrak is built to calculate the coordinates (x,y,z) of the finger, which represents the finger's position in 3D space around the watch.

$$(x - x_a)^2 + (y - y_a)^2 + z^2 = dA^2 \tag{1}$$
$$(x - x_b)^2 + (y - y_b)^2 + z^2 = dB^2 \tag{2}$$
$$(x - x_c)^2 + (y - y_c)^2 + z^2 = dC^2 \tag{3}$$
$$(x - x_d)^2 + (y - y_d)^2 + z^2 = dD^2 \tag{4}$$

Based on Figure 2, we can derive equations 1, 2, 3, 4 to represent the geometric relationships between locations. We need to derive the value of x,y,z using these equations. If we assume all the other variables are known in these equations, x,y,z can be calculated using any three of the four equations. There are four combinations of the three equations: 1 2 3, 1 3 4, 2 3 4, 1 2 4. For example if we choose the equations 1 2 3 to solve the value of x, y, z, we get get equations 5 6 7. In these equations, $x_a, y_a, x_b, y_b, x_c, y_c$ are all preset constants, which represents the coordinates of receivers. Solving equations 5 6 7 gets us the location of the finger.

$$x = \frac{(y_c - y_a)(dA^2 - dB^2 - y_a^2 + y_b^2 - x_a^2 + x_b^2) - (y_b - y_a)(dA^2 - dC^2 - y_a^2 + y_c^2 - x_a^2 + x_c^2)}{2(x_b - x_a)(y_c - y_a) - 2(x_c - x_a)(y_b - y_a)} \tag{5}$$

$$y = \frac{dA^2 - dC^2 - y_a^2 + y_c^2 - x_a^2 + x_c^2 - 2(x_c - x_a)x}{2(y_c - y_a)} \tag{6}$$

$$z = \sqrt{dA^2 - (x - x_a)^2 - (y - y_a)^2} \tag{7}$$

3.2 Solution to the geometric model

The set of equations 5 6 7, although mathematically solvable, are computationally demanding. Solving these equations on a computing device requires non-negligible time which adds delay for further processing. Since we aimed to make SoundTrak a real-time system, we chose not to solve these equations by the conventional methods to avoid any processing delays. We chose to solve these equations by finding the value of dA, dB, dC, and dD using acoustic data and substituting these values into equations 5, 6, and 7 to solve them.

At any time t,

$$dA_t = dA_0 + \Delta dA \tag{8}$$

$$dB_t = dB_0 + \Delta dB \tag{9}$$

$$dC_t = dC_0 + \Delta dC \tag{10}$$

$$dD_t = dD_0 + \Delta dD \tag{11}$$

Where dA_0, dB_0, dC_0, and dD_0 represent the distances of the speaker from the receivers A, B, C, and D, respectively, at time t=0 and ΔdA, ΔdB, ΔdC, and ΔdD reflects the change in dA, dB, dC, and dD over time t. We start by solving these equations by keeping the speaker at a known location at time t=0, which gives us the value of dA_0, dB_0, dC_0, and dD_0, or the initial calibration point. For our setup, we chose the initial calibration point as the center of the watch (shown as the red dot in Figure 2. This helps us directly find out the value dA_0, dB_0, dC_0, dD_0 as $dA_0 = dB_0 = dC_0 = dD_0 = \frac{\sqrt{(x_a - x_b)^2 + (y_a - y_c)^2}}{2}$.

The values dA_0, dB_0, dC_0, dD_0 are known from the initial calibration and ΔdA, ΔdB, ΔdC, ΔdD is calculated from the acoustic data which helps us calculate dA, dB, dC, dD at any time t and in turn find the value of x,y,z.

3.3 Calculating distance between the speaker and receivers using phase information

By definition, phase is the position of a point in time on a waveform cycle.[1] Its value falls into the range of 0 to 360 degrees (or 0 to 2π radians). The phase value can be used to calculate the relative displacement between two waves with the same frequency. This relative displacement d between the waves can be calculated using equation 12, where $\Delta\varphi$ is the value of the phase shift, and λ is the wavelength of the sound wave. The wavelength of sound is a well-known value that is a result of the speed and frequency of sound as shown in equation 13.

$$d = -\frac{\Delta\varphi \times \lambda}{360} \tag{12}$$

$$\lambda = \frac{v}{f} \tag{13}$$

SoundTrak uses a sinusoidal wave as our choice of signal and using phase information from the signal received from the speaker, we calculate the distance between the finger and the points A(dA), B(dB), C(dC), D(dD). However, converting the phase information to absolute distance will encounter several technical challenges.

[1] https://en.wikipedia.org/wiki/Phase_(waves)

3.3.1 Challenges. Firstly, the phase information of the signal received at the receiver cannot alone be used to determine the distance between the speaker and receiver. The distance is usually extracted by using the phase difference between the signal being sent from the speaker and the receiving signal at the receiver at any given time. However, such a comparison requires the sender to be able to communicate the time-stamp of each wave cycle to the receivers, which can be challenging for wearable devices. How to synchronize the time stamp between devices is the first challenge.

Secondly, the phase only provides information pertaining to one cycle of the periodic wave. This means that if we plan to calculate relative displacement using phase, the maximum displacement will be limited by the range of phase values in a cycle. And from equation 12, the maximum displacement value which can be calculated using the phase is the wavelength of the signal. In the context of finger tracking, this means that if the displacement of the finger is greater than the wavelength of the signal, calculating displacement using phase may become difficult. For instance, the wavelength of 11025 Hz while propagating in air is about 3.2 centimeters. If the displacement of the finger is greater than 3.2 centimeters, the displacement will contain more than one wave cycle as shown in Figure 4a. Therefore, the same phase values may appear multiple times. This will cause ambiguity and the system may not be able to determine the phase shift without additional information.

Lastly, during our early experiments, we observed an initial phase bias at each microphone, which is different from one to another and remains constant over time. It is hard to compare the phase value in the presence of such device-specific phase bias.

3.3.2 Solutions. SoundTrak uses the following solutions to addresses the above challenges.

Firstly, to solve the problem of removing any communication between the sender and the receiver, we introduce the concept of a reference signal. The reference signal in our system is a replica copy of the signal originating from the speaker and persists at the receiving end. At the receiving end, the reference signal is used while calculating the phase shift/difference between the signal originating from the speaker to the signal being received at the receiver. To obtain the phase value at the speaker, receivers can refer to this reference signal instead of relying on a communication channel between the speaker and the receiver.

In our system, we collect the reference signal ahead of time from the same function generator which is used to drive the speaker. The reason to use a prerecorded signal instead of generating our own reference signal at the same frequency is that we found a different periodicity of the signal on different machines (probably due to the different quality of the clocks). We recorded about six seconds of the sine wave signal for each frequency from the function generator. We then removed the incomplete period at the start and end of the signal to make the recorded signal periodic. Therefore, we can manually extend the recorded signal for real-time processing. In summary, by using a prerecorded sender signal as reference signal, no communication between the receiver and sender is required when the system is running in real-time.

The other problem is to find the absolute displacement given only the phase of the receiving signal. To calculate the absolute displacement only using phase information, it is required to know the phase value of the signal at the transmitting end as well as at the receiving signal. But as we do not have any communication channel between the sending and receiving end, to increase the practicality of the system, we do not know the value of the phase at the transmitting end. This value of phase at the transmitting end is a random value based on the sent time. To eliminate this initial value, we do not track the phase but the phase shift using the reference signal as presented in the function GET_PHASE_SHIFT in Algorithm 1. The phase shift result we get is the real phase shift plus a random initial value. However, we can get displacement from the last known location to the current location by comparing the phase shift change. For example at data point 500 in Figure 3b, the value -300 indicates that compared with the start value 0, the phase shift has changed by -300, which means the distance between speaker and receiver changed $\frac{300\times\lambda}{360}$ based on equation 12. Therefore, if we accumulate the distance change between adjacent points from the beginning of the system, and the initial distance at the beginning is

known, we can then calculate the absolute location at any given time. To do this, we use the calibration point as the starting position, which is at the center of the four speakers at z = 0. Since we know the coordinates of the calibration position, by accumulating the displacements at each receiver over time, we can calculate the distance from the current position to the starting point.

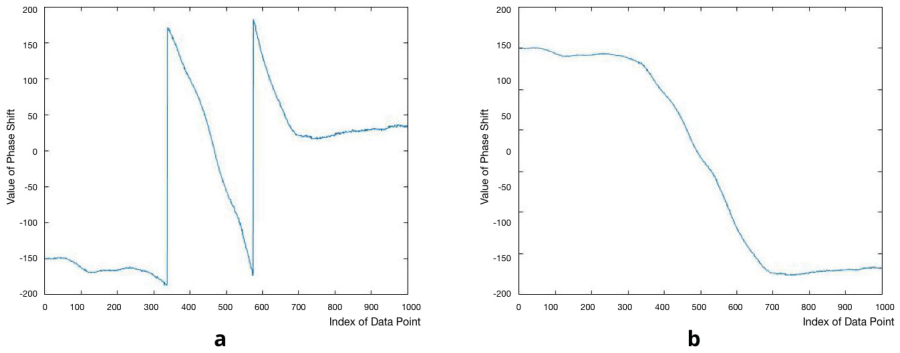

Fig. 3. Adding extra -360 degrees to overcome periodic limitation. **a.** The phase shift result. **b.** The phase shift result after correction. The x-axis represents the index of each data point, and the y-axis represents the value of phase shift at each data point.

Secondly, to solve the problem of the maximum displacement value limited to just one wavelength, we continuously track the phase shift to detect a change of cycle in the periodic wave. For this, whenever we detect a sudden change of about +/−360 degrees in the phase shift value, we accommodate for this change by adding correction values to the original phase shift value. There can be two reasons for this sudden change of the phase shift value. The first reason is that when subtracting the two signals, one of the two signals exceeds its period and the other one remains in the current period, as shown in Figure 4a. The second possibility is that the phase shift value exceeds a period because the distance between transmitter and receiver changed more than a wavelength, as shown in Figure 3a.

Figure 4 shows the result of directly subtracting the reference signal phase (Figure 4b) from the received signal phase (Figure 4a), which would introduce a peak when any of the signal exceeds the edge of one period as shown. We can easily solve this issue by extending the phase value which exceeds the period slightly over the range -180 to 180 degrees until both signals move to the next period. Thus we get a reasonable phase shift result as shown in Figure 4d. The resulting phase shift is a constant, which means location did not change in between the two given times. A closer observation shows that there are only 20 points in Figure 4, which equals $20/44100 = 0.0004$s and hence the distance could not change significantly in such a short time. This technique addresses the first reason of a sudden change in phase shift due to one signal exceeding its period while another does not.

The second reason for a sudden phase shift is when the distance between transmitter and receiver changes by more than a wavelength. Figure 3a shows the phase shift result of a continuously "moving away" gesture, for which the phase shift value should keep decreasing based on equation 12, but it suddenly increases 360 degrees when it exceeds a period. We solve this problem by subtracting 360 degrees for the phase shift value after the sudden change, thus making it smooth. Figure 3b shows the phase shift result after correction for the periodic problem. This method is feasible with one assumption, that the phase shift value between two adjacent data points would not be close to 360 degrees, i.e., the finger will not move about a wavelength far in a unit

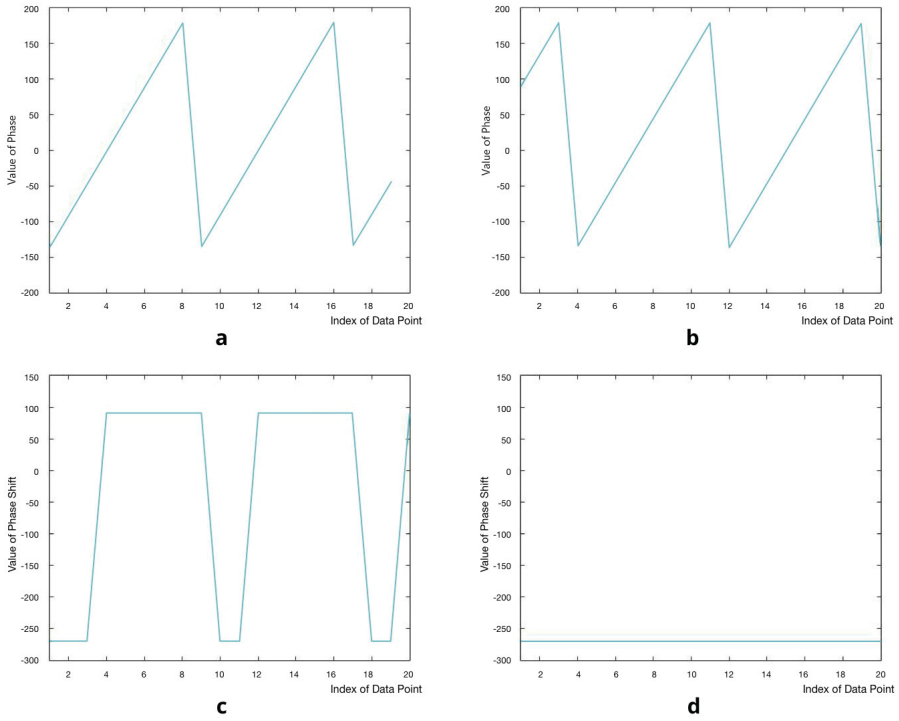

Fig. 4. Processing phase at different stages. **a.** The phase value of a received signal. **b.** The phase value of a reference signal. **c.** Result of subtracting the reference signal phase from the received signal phase **d.** The phase value after extending it beyond a cycle. The x-axis represents the index of each data point, and y-axis represents the value of phase at each data point

sample time. For example, if we use 11025Hz (the wavelength will be 0.032 meters) as the signal frequency and 44100 Hz as the sample rate, it means that the finger will not move at a speed that is faster than 1411.2 m/s (0.032 meters/(1/44100)seconds), which is a reasonable assumption. Algorithm 1 below describes this phase shift adjustment.

ALGORITHM 1: Algorithm to overcome periodic limitation

if $phase\_shift - previous\_phase\_shift \approx 360$ **then**
    phase_shift=phase_shift-360 ;
    period_add=period_add-360 ;
if $phase\_shift - previous\_phase\_shift \approx -360$ **then**
    phase_shift=phase_shift+360
    period_add=period_add+360 ;

Finally, to address the different initial phase offset , we record the offset at each receiver during the initial calibration process and accommodate for these offsets in all future calculations.

4 SYSTEM DESIGN

To validate our model and implementation of SoundTrak, we built a hardware prototype and data processing pipeline. For our initial prototype, the user needs to wear the speaker in a casing resembling a finger ring.

4.1 Hardware Design

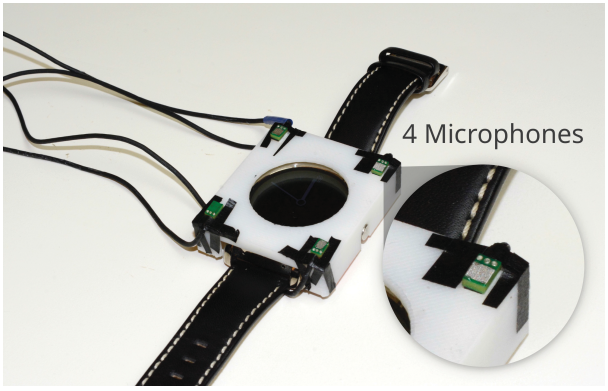

Fig. 5. LG G Watch with SoundTrak

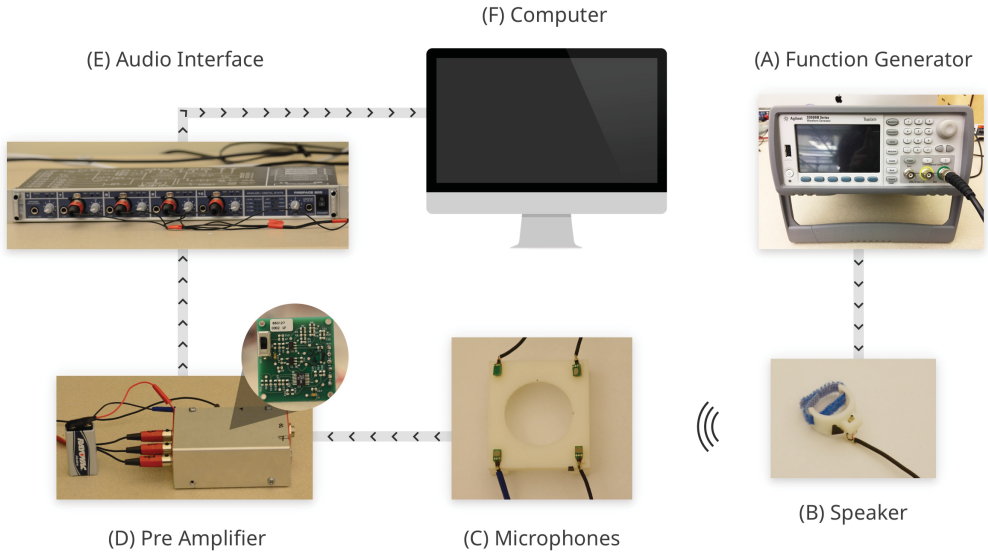

Fig. 6. Hardware setup for SoundTrak

Figure 6 shows the hardware setup for SoundTrak. The hardware of the prototype contains three major parts: the sender, the receivers, and the data acquisition hardware. We use a miniature speaker (Knowless FK-23451, 5.00mm Length *2.73mm Width * 1.98mm Height) as the sender, which is attached to a 3D printed ring (Figure 6B). The speaker is driven by a function generator (Agilent 33500B, Figure 6A). The system has four MEMS microphones (STMicroelectronics MP33AB01) as receivers (Figure 6C). Each microphone is connected to a separate customized preamplifier (Figure 6D). The output from these preamplifiers then goes to an audio interface (Fireface 800, Figure 6E) for data acquisition. We wrote a thin client in C with the PortAudio library to collect data on an iMac (Figure 6F). The data is transmitted via a socket to a Java program for real-time processing on the same iMac.

4.2 Algorithm

The real-time processing algorithm (see Algorithm 2 pseudocode) contains three major functions. UPDATELOCATION is the general data pipeline of calculating locations from the input sound data. CALIBRATION records the $correction\_value$ at each receiver, which includes information of the sensor bias and the initial distance between speaker and receiver at calibration. The $period\_add$ value is used to accommodate the period change of the $phase\_shift$ value. Using the equations 1, 2, 3, and 4, the function LOCATION_FROM_FOUR_RECEIVERS calculates the current coordinates of the ring/speaker. There are three combinations of equations from these four equations used to obtain the four location values, as described in Section 3.1. Finally, the speaker location is assigned as an average of the four location values.

4.3 Choice of frequency

In theory, SoundTrak does not have the limit of which frequency can be used in the system. However, we chose the signal frequency based on the following practical considerations.

First, we use a Fast Fourier Transform (FFT) to calculate the phase value of every $N = 512$ points of the received signal. The results of FFT are a list of bins, each of which represents the information of frequency $k \times SampleRate/N$, where k is an integer from 1 to $N/2$ and the sample rate is 44100 Hz. Therefore, we need to use the frequency related to one of the bins. Second, we observed that the phase value is more sensitive to noise when the frequency is under 3k Hz, as more environmental noise exists in this frequency range. Third and most important, the higher the frequency is, the shorter the wave length will be. So for higher frequencies, the same amount of phase shift represents a shorter distance which can have potentially higher resolution. Therefore, we tested our system with a frequency of 11025Hz for system evaluation and 16537.5Hz (inaudible) for the Fitts' Law test for the ease of FFT calculation.

4.4 Sensor placement

Four microphones are placed at the corners of a rectangle with a length of 4 cm and a width of 4.3 cm. The sensor placement is mostly determined by the practical challenge of deploying the sensors on top of a commercial smartwatch. Using a more advanced sensor placement would potentially increase the range of operation and the accuracy of the system. We plan to further investigate this in the future.

5 EVALUATION

5.1 Assessing spatial resolution

5.1.1 Experiment. To measure the tracking accuracy of SoundTrak in 3D space we designed an experiment which involved comparison of values recorded by the system against a predefined set of ground truth coordinates. We used an off-the-shelf Makerbot (Replicator) to position the speaker accurately in 3D space as shown in

ALGORITHM 2: Iterative Algorithm

UPDATELOCATIONsound data
Begin receiving sound data
$period\_add = 0$
call function *CALIBRATION*
$previous\_phase\_shift = phase\_shift$
while $gettingdata$, **do**
  for *every receiver,* **do**
    bandpass filter
    $phase\_shift \leftarrow GET\_PHASE\_SHIFT$
    $phase\_shift = phase\_shift + correction\_value + period\_add$
    if $phase\_shift - previous\_phase\_shift \approx 360$ **then**
      phase_shift=phase_shift-360 ;
      period_add=period_add-360 ;
    if $phase\_shift - previous\_phase\_shift \approx -360$ **then**
      phase_shift=phase_shift+360
      period_add=period_add+360 ;
    $pre\_phase\_shift = phase\_shift$
  end
  return $Location \leftarrow LOCATION\_FROM\_FOUR\_RECEIVERS$
end
CALIBRATION
for *every receiver,* **do**
  bandpass filter
  $phase\_shift \leftarrow GET\_PHASE\_SHIFT$
  calculate $correction\_value = -phase\_shift - 360 * initial\_distance/wavelength$
  $phase\_shift = phase\_shift + correction\_value$
  $pre\_phase\_shift = phase\_shift$
end
return $location \leftarrow LOCATION\_FROM\_FOUR\_RECEIVERS$
GET_PHASE_SHIFT
for *each sample point in received data stream,* **do**
  $phase \leftarrow FFT$
  $phase\_shift = phase - reference\_phase$
end
return $phase\_shift$
LOCATION_FROM_FOUR_RECEIVERS
$location1 \leftarrow$ *calculate location by solving equations (1),(2),(3)*
$location2 \leftarrow$ *calculate location by solving equations (2),(3),(4)*
$location3 \leftarrow$ *calculate location by solving equations (3),(4),(1)*
$location4 \leftarrow$ *calculate location by solving equations (4),(1),(2)*
final_location $\leftarrow$ *average of location1 , location2, location3 and location4*
return $final\_location$

Figure 7. The ground truth coordinates were selected from a 3520 cm^3 3D volume space of ($16cm \times 20cm \times 11cm$) with all the coordinate combinations from the set X (cm)= $\{-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10\}$, Y (cm)= $\{-8, -6, -4, -2, 0, 2, 4, 6, 8\}$ and Z (cm) = $\{1, 3, 5, 7, 9, 11\}$. The z-axis refers to the vertical direction (up and down) and the x-axis refers to the horizontal direction (left and right), when the user is facing the Makerbot. The

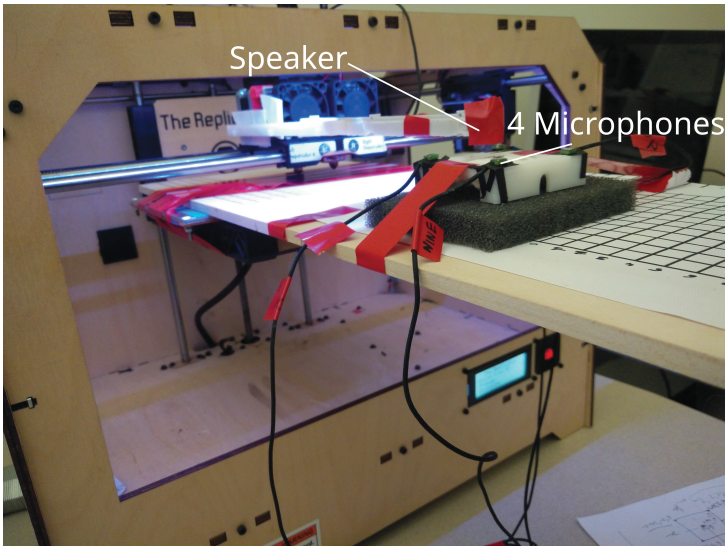

Fig. 7. System Evaluation Setup with MakerBot

experiment was conducted in a shared large room including uncontrolled common office noise (e.g., people conversation, air conditioner, phone ringing). The Makerbot used for this experiment is moved to a specific coordinate in 3D by injecting corresponding GCode (low-level instructional codes to control RepRap based machines) commands over its serial port. The SoundTrak speaker was connected to the Makerbot's left extruder. The receivers were placed on an extended wooden plank on the Makerbot's platform. The extension provides flexibility to choose a calibration point and coordinate axes of our choice and moreover it also minimized the effect of noise produced due to the movement of the Makerbot's motor. The data collection pipeline was semi-automatic and two co-authors synchronized the Makerbot's movement and the corresponding location data logging to maximize accuracy and minimize time synchronization issues. Location data for 594 location coordinates was collected in under 2 hours using this approach. The results and accuracy of the system are discussed in the following section.

5.1.2 Results. Figure 8 depicts the ground truth coordinates for the system evaluation and Figure 9 shows the corresponding calculated values by the SoundTrak system. For a total of 594 location coordinates in the entire volume of 3520000 mm^3, the mean Euclidean error is 13.05 mm. By providing centimeter-level resolution consistently over a large volume space, the system can support a large variety of interactions for the human finger. These results further show that the system is very well suited as a three-dimensional input system for wearable technology, such as a smartwatch. From our evaluation results, the accuracy of the system is maximum for z = 9 cm (mean error = 0.92 cm) in the Z plane, for x = 0 mm (mean error = 0.87cm) in the X plane and for y = -2cm (mean error = 0.93 cm) in the Y plane.

Figure 10 contains six heat maps for Z ranging 1cm to 11 cm. The coordinate system and the placement of the watch used for the system evaluation are represented in the topmost image. From the heatmaps, it is evident that SoundTrak achieves maximum accuracy for the region within Y= -2cm to 6cm and X = - 4cm to 6cm for all the Z range. For Z = 1cm, the location calculation accuracy deteriorates as we move the speaker in the negative X and Y direction. Although the mean error is minimum for Z = 9cm, it is noteworthy that the variance is lesser for the Z = 7cm. Overall, SoundTrak is more accurate for $Z > 5$cm compared to $Z < 5$cm. The optimal region for 3D

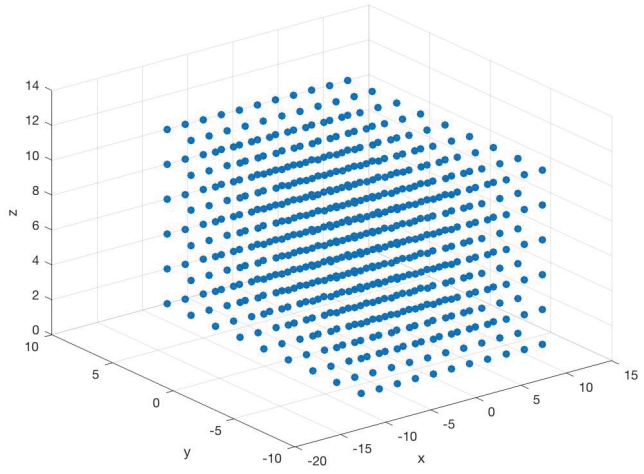

Fig. 8. The predefined positions in the system evaluation

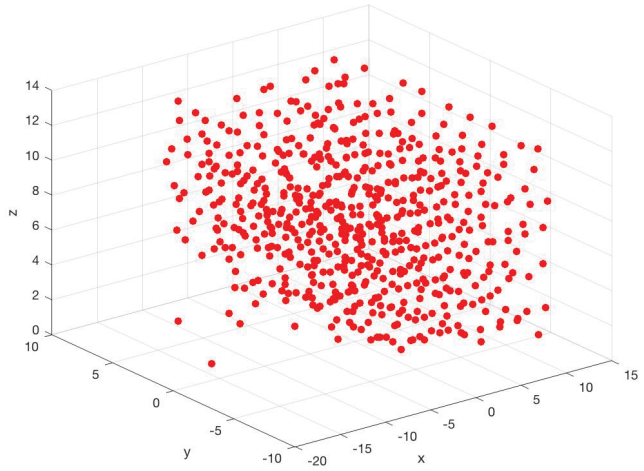

Fig. 9. The positions calculated by SoundTrak in the system evaluation

input is defined by the volume Z=7cm to 9cm, X = -2cm to 6cm and Y=2cm to 4cm. Point A in the coordinate system is the reference point and all the values are calculated considering point A as the origin of the system for the testing.

From Figure 8 we observe that for the ground truth values of Z = 1cm the predicted values are less accurate. We attribute the higher error rate when z is smaller to two possible causes. First, the speaker is not omnidirectional, it outputs unbalanced energy in different directions. The direction where the speaker is pointing to would receive

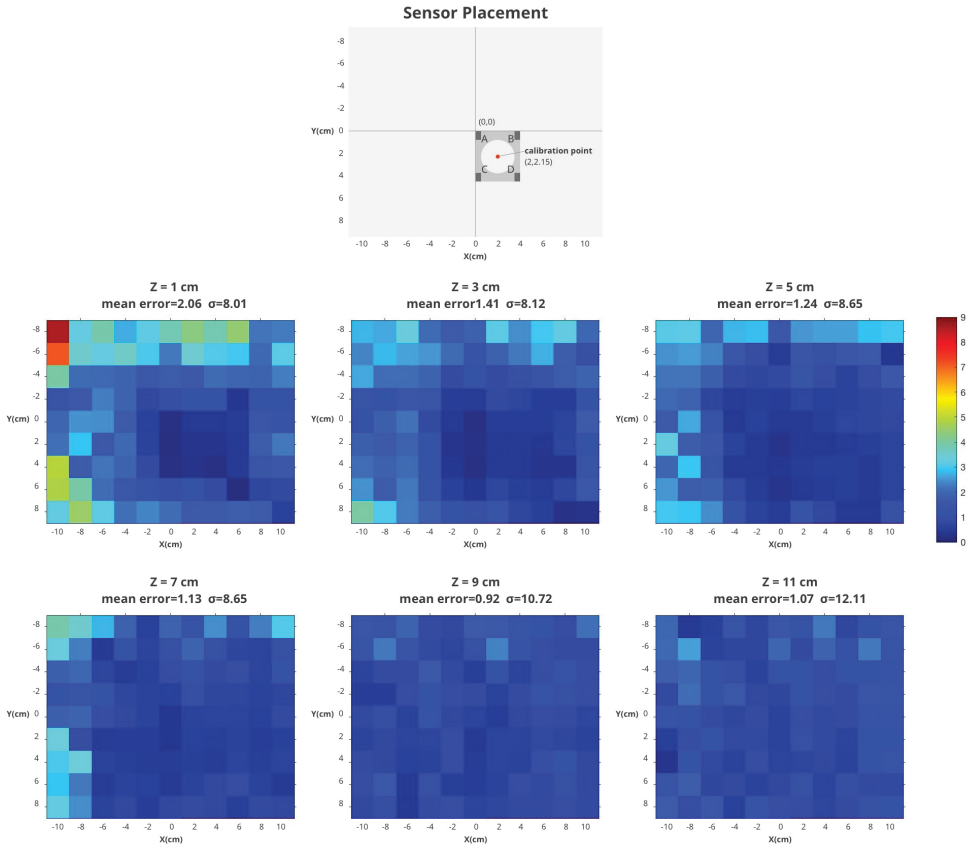

Fig. 10. Heatmap of euclidean errors calculated along x-y planes at different z (height) values

more energy than the direction where it is perpendicular to the speaker. Second, the reflected signal on the surface can be relatively stronger when the distance between the speaker and the surface is closer (e.g. z = 1cm), resulting in higher errors.

5.2 Evaluation of the drift

5.2.1 Experiment. SoundTrak integrates the phase changes over time to calculate the location of the finger. However, noise produced by hardware and the environment cannot be completely avoided in such systems. This noise in the signal can result in errors while calculating the phase values. When integrated, even a minimal error at each data point may cause a non-negligible influence on the accuracy over time. Furthermore, if the accuracy drops significantly, the participant may have to recalibrate the system again, which would degrade the user experience.

To evaluate the drift of our system over time, we conducted another experiment using the same frequency (11025 Hz) and hardware setup as the system evaluation experiment described above. We recorded the data for 20 minutes at each of the three locations, where the position on x and y-axis are 0 cm, and the position on z-axis is 1,5,9 cm for each session respectively. 20 minutes is a relatively large time interval for a user interacting with a

smartwach continuously and offers a good estimate of how the system would perform over even larger intervals of time. To avoid human error, we generated a file containing location information along with the time-stamp for each location point.

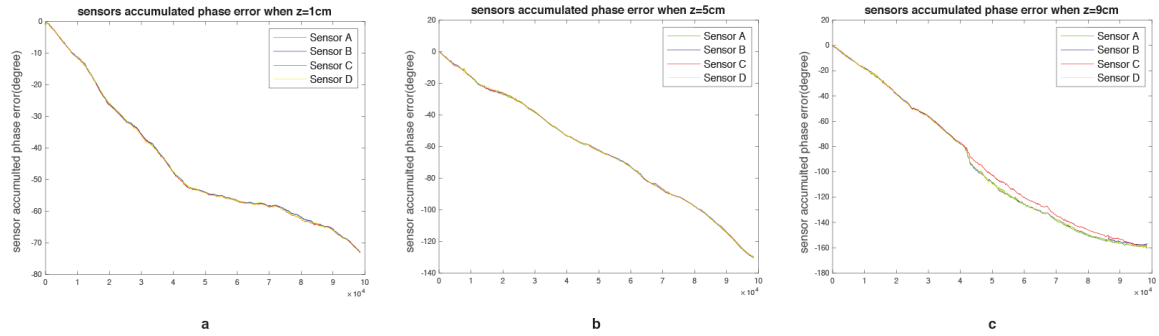

Fig. 11. Accumulated phase changes on three axes for 20 mins when z = 1,5,9 cm

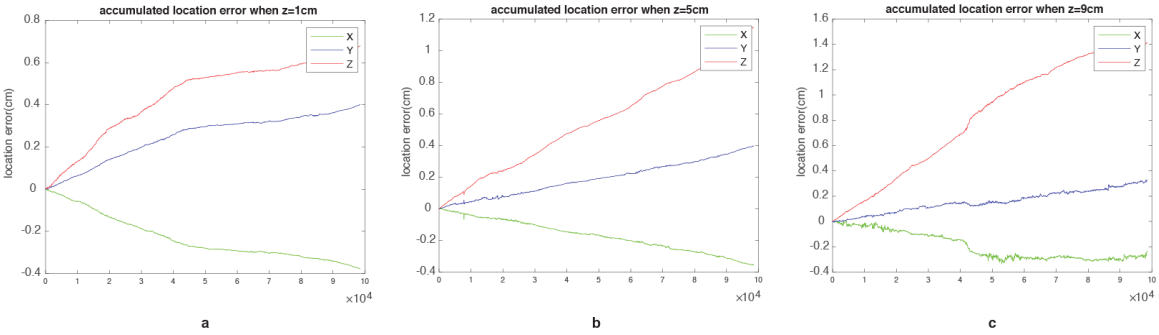

Fig. 12. Accumulated location changes on three axes for 20 mins when z = 1,5,9 cm

5.2.2 Results. To calculate the drift of phase values on each microphone, we subtract the starting phase value from all the later phase values to derive the relative phase changes over 20 minutes. The results are shown in Figure 11. The phase drift on different sensors are similar at each location. The phase decreased 73, 130, 159 degrees respectively for z = 1, 5, 9 cm in 20 minutes.

Based on these phase value, we further calculated the location drift on x,y,z-axis by using the formula presented in Section 3. Similarly to the method we used to derive the phase drift, we subtracted the original coordinates to all the later ones, and the results are presented in Figure 12. In the three 20-minute experiment, the x coordinates decreased 0.375 cm, 0.35 cm and 0.33 cm, the y coordinates increased 0.398cm, 0.397 cm, and 0.314 cm, and the z coordinates increased 0.68 cm, 1.148 cm and 1.412 cm respectively.

The above results show a greater drift for the z-axis than the x and y axes. The reason is the coordinate on z-axis is calculated based on the coordinates of x and y axes, as formula 7 described. Any drift of the location along either the x or y axis would be exaggerated when calculating the position on the z-axis. Therefore, while designing interactions in 3D space using SoundTrak, the x-y plane sould be considered as having a higher spatial location accuracy.

There are two possible reasons that may cause the drift. One is the unreliable clock of the functional generator that may introduce accumulated drift over time. Since we did the z=1cm session first and z= 9cm session last, it is possible that the longer time the system is running , the larger drift may be observed. Another possible reason is the system suffers more from multipath effect when the distance between the speaker to the receivers are larger, as we will discuss in details in section 6.2.

Despite the drift error, considering most of the interaction on wearable devices would not exceed 20 minutes per session, even such a drift exists, the user may not need to recalibrate the system during an interaction session. However, we plan to further investigate this system drift while the speaker is in motion in the future.

In addition, as Figure 11 shows, the phase value does not decrease at a constant rate. Sometimes the phase would suffer a relative and sudden large drop such as the center of the Figure 11c shows. We attribute this to the random environmental noise presented during the whole experiment. Various types of environmental noise appeared during the experimental session, such as the air conditioner noise, loud music played by students, phone rings, conversations among students, as well as the sound caused by opening and closing of the doors. These are all natural sounds that would be expected in normal use of SoundTrak.

5.3 User Evaluation

To better understand the usability of SoundTrak from the user's perspective and how it could be used in practice, we conducted a Fitts' Law study. Though SoundTrak is able to track a finger's position in 3D space, we conducted the user study in two 2D planes (vertical and horizontal). The reason is that it is more likely that SoundTrak will be used for devices that still have a 2D output device in the near future. Understanding how SoundTrak would potentially enhance the interaction with the 2D screen is important. Therefore, we decided to conduct a Fitts' Law study, which is widely used to evaluate rapid and targeted movement on 2D screens. In order to map the movement in a 3D space to a 2D screen, we split the 3D coordinate system into the X-Y plane and the X-Z plane, which are horizontal and vertical to the watch face, respectively in the study.

5.3.1 Participants and Apparatus. In this user study, we recruited 10 participants with an average age of 26.2 (6 males) from a university campus. All of them were first-time users. During the study, participants were sitting in front of the computer running the Fitts' Law software and resting their left arm on the desk. They were wearing an LG Watch Urbane with a 3D printed case on their left wrist. As shown in Figure 5, four microphones were placed at the corners of the case on the watch. A miniature speaker was taped to the nail of the index finger on their right hand.

To complete a Fitts' Law task, participants were asked to control a cursor on a nearby computer screen by moving and pointing the speaker on their right index finger around sensors on the watch on their left wrist. A trackpad was placed on the desk next to the participant's left hand. They were instructed to tap the trackpad to confirm a selection for a Fitts' Law task. The computer was a 27-inch iMac with a resolution of 2560 x 1440 pixels. We mapped every 100 pixels on the screen to 1-centimeter movement. To stabilize the cursor on the screen, we drew the current cursor at the position which was the average of the last 30 location values obtained by SoundTrak.

5.3.2 Procedure and Design. At the beginning of the study, a researcher helped the participant to put on the watch and the speaker. The participants were then given an introduction of how to interact with the test software, including mapping the finger movement in the relevant 2D plane to the cursor movement on the screen and tapping the trackpad to confirm the selection using their left hand.

The Fitts' law software we used in the study was developed by MacKenzie et al.[2]. Tasks were designed based on the ISO 9241 standard, as modeled by Fitts' Law. We had 13 targets in the test as shown in Figure 13. Participants

[2]http://www.yorku.ca/mack/FittsLawSoftware/

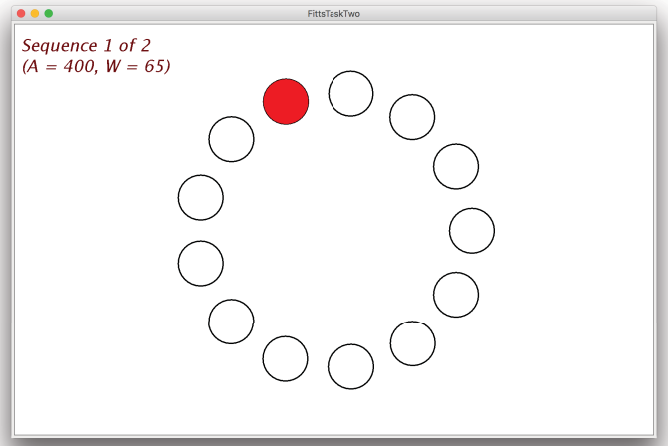

Fig. 13. Fitts' Law test interface with 13 targets.

were instructed to move the cursor to select the highlighted target using SoundTrak as accurately and quickly as possible. The system highlighted the next target after selection. A trial ended after all targets were selected. We set the parameters to be two conditions (X-Y plane, X-Z plane), two distances (400 and 500 pixels), and three target sizes(40, 65 and 90 pixels). For each condition, there were four sessions. Each session consists of 6 trials (6 = 2 distances x 3 target sizes). In each trail, the participant selected 13 targets. The system was calibrated at the beginning of each trial. In total, each participant completed 48 trials (2*4 sessions, with 6 trials in each session). The order of conditions and the order of trials were randomized. The movement time and the error rate were recorded.

5.3.3 Results.

Learning Effect. To understand the learning effect, we conducted repeated-measures ANOVA on the results. Fig 14 shows the learning effect for each plane on movement time. The analysis shows the effect was very significant for the X-Y plane ($F_{3,9} = 15.42$, $p < 0.001$), but not significant for the X-Z plane ($F_{3,9} = 3.59$, $p > 0.05$). However, the difference from session 2 to session 4 is not significant for both the X-Y plane ($F_{2,9} = 4.82$, $p > 0.1$) and the X-Z plane ($F_{3,9} = 2.67$, $p > 0.1$) . Therefore, to avoid potential issues caused by the learning effect, session 1 is considered as a training session and all analyses of the results are based on session 2–4 only.

Movement Time. The average movement time per trial for the X-Y plane is 2310.87 ms with a standard deviation(SD) of 249.81, and for the X-Z plane is 2594.52 ms (SD = 362.72). The difference is considered to be very statistically significant ($p = 0.0038$). The analysis of the movement time by the effective index of difficulty (IDe) also indicates that it takes longer time to accomplish more difficult tasks.

Error Rate and Throughput. The average error rate per trial for the X-Y plane is 6.03% (SD = 3.38), and for the X-Z plane is 7.91% (SD = 3.93). However, the difference is not statistically significant ($p = 0.0694$).

The average throughput (bps) for the X-Y plane is 1.41 (SD = 0.12), and for the X-Z plane is 1.31 (SD = 0.16). The analysis shows a significant difference between these two planes ($p = 0.0199$). One possible reason is that most of the 2D tracking devices are designed to be operated in the X-Y plane. It is more challenging for most participants to perform movement in the X-Z plane.

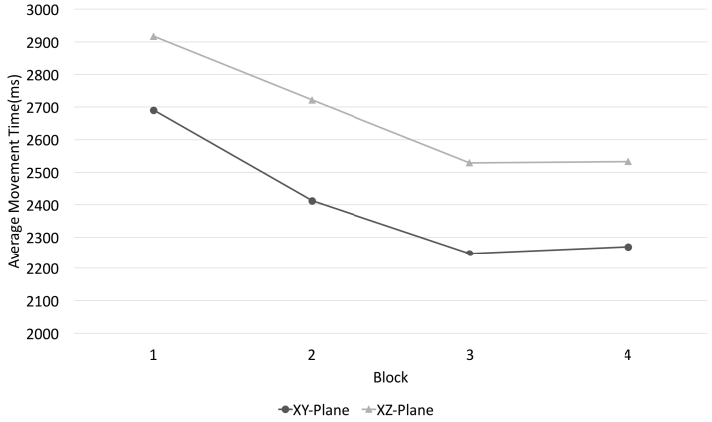

Fig. 14. Average movement time by plane and session

Implications. The results of the study demonstrate the learnability of our system. Participants were able to get used to the proposed interaction on the watch after only one training session. However, we also noticed the significant difference of the performance between the X-Y plane and the X-Z plane in the study. We attribute this difference to two factors. First, according to the results presented in Sections 5.1 and 5.2, the calculated location on z-axis is less accurate than the calculated locations on the x and y axes. The lower accuracy could potentially influence the interaction experience and the performance of the participants. The other possible reason is that many participants reported they were more used to performing gestures on the horizontal plane (X-Y plane) because this is how the interaction is designed on most of their computing devices (e.g., phones, watches, IPads). Therefore, the interaction designer may want to first consider X-Y plane while designing finer input gestures to achieve a better user experience. This does not rule out use of the X-Z plane, but does caution about needed precision of the input gesture.

6 DISCUSSION

6.1 Applications

6.1.1 Gesture control for wearables. Interactions with wearable devices are currently limited by the size and the placement of devices. For instance, using fingers to zoom in/out on a map on the smartwatch is constrained by the small screen size. By allowing the finger movement in 3D space, SoundTrak enables an extended and flexible gesture control for wearable devices. Not limited by the physical size of devices, users are able to perform a much larger set of gestures. It is also possible to establish a new set of 3D gestures for smartwatches built upon SoundTrak that can improve the usability of smartwatches in general. For instance, it is not convenient to launch an app on the smartwatch at this moment, since a user needs to go through a long list on the screen for selection. However, if the list is too long, repeating the sliding gesture is needed. With SoundTrak, the user can easily select the application by moving the finger in one direction in the relative large 3D space around without the need to repeat the same gesture again.

Our current SoundTrak prototype tracked only one finger, but there is not fundamental reason why this technique could not be extended to multiple tracked fingers. By placing SoundTrak rings on multiple fingers, users can easily perform multifinger gestures, such as the standard pinch gesture in the peripheral space around the watch to zoom in/out. A much wider range of 3D gestures can now be explored.

6.1.2 Text input. SoundTrak can also be used for text input for devices that do not have a physical keyboard. By tracing the finger movement in real-time, SoundTrak allows users to draw anything in the air and detects the input. For instance, users can reply short messages quickly from their smartwatch using SoundTrak.

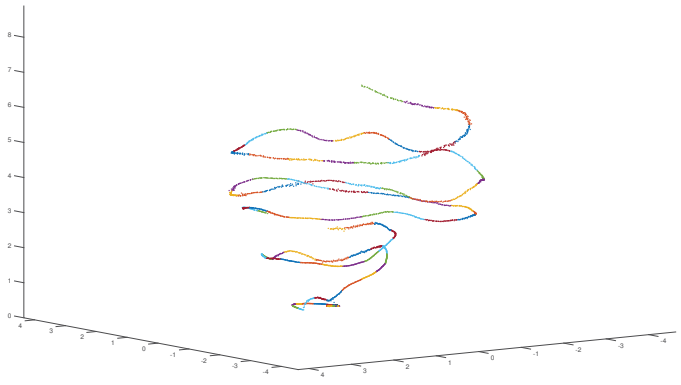

Fig. 15. Drawing Spiral in 3D with SoundTrak

6.1.3 Drawing in 3D space. Most of the current drawing applications are based on 2D tracking technology. Therefore, it is only able to draw the picture in a 2D plane. Though a few smartwatches provides drawing application, it is particularly challenging to draw on a picture on the tiny screen of the watch. SoundTrak enables the user to draw the picture in 3D space around the wearable devices. Therefore, the user can draw picture in 3D space at any time and at any location with their wearable. Figure 15 shows the sample 3D pictures drawn by using SoundTrak.

We should note that we have currently been promoting SoundTrak as a way to provide 3D interaction with a wearable device, such as a smartwatch or a head-mounted device like Google Glass. There is no reason why this 3D interaction should be limited to those devices, and in the case of 3D drawing, we can easily see a motivation for providing this kind of 3D tracking above or around any screen (e.g., table, smartphone, laptop).

6.1.4 Improve interaction on wearable devices with physical spatial awareness. Most of the existing tracking technologies on wearable can only track the relative movement of the finger. The wearable is not aware of the finger's position in the 3D space. Providing the physical spatial awareness to the wearable can be potentially used to redesign the interaction for improving the interaction efficiency and user experience. For instance, we can define any space around the wearable as the location of a virtual button. By moving the finger to that predefined space, a user can click a button. However, clicking the virtual button accurately in 3D space can be challenging. We plan to further explore this in the future.

6.1.5 3D input for other scenarios. As already suggested, the technology presented in SoundTrak is not limited to detecting a single finger's position on wearable devices. It can also be used in other scenarios, such as turning the area around a horizontal or vertical surface into a 3D interaction area. If multiple speakers can be attached to different fingers, it is also possible to capture multiple fingers' position simultaneously. The availability of such "multi point" sensing in 3D space can open up a whole new space for novel gestures interactions with wearable and other devices.

6.2 The Effect of Multipath and Environmental Echoes

Like other acoustic based systems, the transmitted acoustic signal does not exclusively travel in the direct path from speaker to microphone; it can be scattered by the fingers or other environmental objects along the way. Therefore, multipath problems may exist. The phase of received multipath signals are different than the straight line path, which may contribute to the error in our system. However, this is not a significant issue in our system.

The multipath signal only contributes to a relatively small fraction of the received signal. With a frequency of 11 kHz, and a 2 cm diameter finger, the dimensionless wavenumber with respect to radius is $ka = 2$. With that ka, the scattered pressure in all directions is roughly 3 dB down with respect to the incident wave amplitude [12], which means that the scattering loss is about one half. What's more, the Inverse Square Law [31] states that the signal intensity is inversely proportional to the square of the distance from the signal source, which can be applied on the sound traveling process in air, both for direct signal and the scattered signal. Therefore, the scattered field incident back to the microphones is always comprised of contributions that have suffered greater spherical spreading loss (longer path length) than the direct field, and also scattering loss. Therefore, their contribution at the microphone array is a small fraction of the total sensed field.

For environmental echoes, presumably from structures even further away than the finger-watch system, the strength of the multipath field over those longer propagation paths is even lower.

Combining these factors together, we think the multipath signals and environmental echoes only contribute to a very small portion of the received signal and would not cause significant errors in most of the cases.

6.3 Power Consumption

SoundTrak introduces a four microphone array and a finger-mounted speaker. The four microphones can be connected to a mother device (e.g., a smartwatch) for power. The speaker on the ring is a separate module and needs to be powered separately. From our experiments, the speaker has very low power requirements and can also be a passive device, harvesting energy wirelessly. In our setup, we used a 2V peak-to-peak voltage (Vpp) to drive the speaker. The impedance of the speaker was about 1000 Ohms when working at approximately 10k Hz. From these two values we can estimate the power consumption of the speaker to be nearly 10^{-4} W. For this small scale of power, wireless charging can be deployed to power the speaker and the size of the charging module can be built small enough to fit in a ring. There are off-the-shelf wireless charging products for small wearable devices now, such as TIDA-00329 designed by Texas Instruments. The tiny wireless charging receiver features an ultra-small size (5.23 mm x 5.48 mm) and is capable of up to 2W power delivery, which perfectly matches our requirements. The wireless power source typically needs to be within a certain range (e.g., 2 meters) from the speaker, which is a constraint to keep in mind.

6.4 Removing initial calibration gesture

For future work, we aim to remove the initial calibration position. With an alternate geometrical model, it is possible to realize this. The alternate geometric model makes use of the fact that the difference in the phase shifts of the receivers can be used to find the difference of individual distance of the speaker to the different receivers. For example, the difference of phase shift between receiver A and receiver B can be used to calculate the difference of distance between receiver A and speaker, and receiver B and speaker. If the phase shifts of the four receivers are known, the location of the speaker can be calculated using it.

If the distance of speaker to receiver A is dA and the distance of speaker to receiver B is dB then $dAB = dA - dB$ is the difference of distance dA and dB and similarly dAB,dAC,dAD. In this model, assume receiver A and receiver B as the two focal points of a hyperbola. Based on the definition of a hyperbola and the information dAB, we can get equation 14. In the same way, assuming receiver A and receiver C as the focal points, we get equation 15 and

when receiver C and receiver D are the focal points we get equation 16. The location (x,y,z) of the speaker can be calculated as the intersection of the three hyperbolas.

$$\frac{(x-\frac{xb}{2})^2}{(\frac{dAB}{2})^2} - \frac{y^2+z^2}{(\frac{xb}{2})^2-(\frac{dAB}{2})^2} = 1 \tag{14}$$

$$\frac{(y-\frac{yc}{2})^2}{(\frac{dAC}{2})^2} - \frac{x^2+z^2}{(\frac{yc}{2})^2-(\frac{dAC}{2})^2} = 1 \tag{15}$$

$$\frac{(x-\frac{xb}{2})^2}{(\frac{dCD}{2})^2} - \frac{(y-yc)^2+z^2}{(\frac{xb}{2})^2-(\frac{dCD}{2})^2} = 1 \tag{16}$$

While mathematically correct, this geometric model is computationally inefficient. So we did not implement this in our original prototype. In future work, we plan to find an approach to solving these equations in a reasonable time to support our application.

6.5 Improve the localization accuracy

The accuracy of the system can further be improved by applying several techniques. Two major techniques are discussed in this paper. The first is to partition the 3D space into multiple sub segments and recursively find location in smaller sub segments starting from the largest one. With decreasing size of segments, we propose to use increasing frequency of audio signals to accurately narrow down to the location. Larger frequencies tend to give better accuracy in smaller spaces but fail to do that in larger spaces and the converse is true for smaller frequencies. By using this recursive approach we can calculate more accurate locations. This approach certainly requires the speaker to be capable of emitting multiple frequencies or the availability of multiple speaker at the same location.

The other technique which can help to increase the accuracy is by using an omnidirectional speaker. As we have discussed in the previous session, an unbalanced energy output at all directions reduces the overall accuracy. Using an omnidirectional speaker to output balanced energy at different directions would potentially increase the accuracy.

6.6 Limitations

Our system detects the delta change in position at each unit of time and predicts the new location of the speaker using a previous known location. This way of continuous tracking works well when the speaker is always in audible reach of the microphones but fails to calculate the location accurately when the audio signal gets blocked or when the finger moves out of the range. In this scenario the system will not have enough data to calculate the location and may give unpredictable output. A recalibration of the system is required. In practice, this means that if a user unwillingly moves finger out of the range of tracking system, a recalibration would be needed. Providing feedback to prevent this remains a challenge.

In addition, although our system does not impose any practical limit on the speed of movement of the speaker, if the speed is too high, the accuracy may be influenced. Because the high moving speed may exaggerate the frequency shift due to the Doppler Effect, which may result in unexpected phase changes.

7 CONCLUSION

To address the input challenge of wearable devices, we introduced SoundTrak, an active acoustic sensing technique that can track the finger's position in 3D space with a mean euclidean distance of 1.305 cm in a volume of 3520 cm^3 using four MEMS microphones and one speaker. A Fitts'law test with 10 participants was conducted to verify

the usability of our technology. We also presented an discussion about the potential applications and the practical challenges before it can be widely adopted.

8 ACKNOWLEDGMENTS

We thank the reviewers for their constructive feedback and the participants for participating the user study. This work is partially supported by Georgia Tech Wearable Computing Center Engagement Grant.

REFERENCES

[1] Christoph Amma, Marcus Georgi, and Tanja Schultz. 2012. Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3D-space handwriting with inertial sensors. In *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 52–59.

[2] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. 2013. DopLink: using the doppler effect for multi-device interaction. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 583–586.

[3] AE Brenner and P De Bruyne. 1970. A sonic pen: a digital stylus system. *IEEE Trans. Comput.* 19, 6 (1970), 546–548.

[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 (UIST '15)*. ACM, New York, NY, USA, 549–556. https://doi.org/10.1145/2807442.2807450

[5] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D Input Using Two Magnetic Sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 237–244. https://doi.org/10.1145/2501988.2502035

[6] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-power Gesture Input with Wrist-worn Pressure Sensors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*. Association for Computing Machinery (ACM). https://doi.org/10.1145/2642918.2647396

[7] EPOS. 2007. EPOS Ultrasonic Pen. (2007). http://tce.technion.ac.il/wp-content/uploads/sites/8/2015/04/Nathan-Altman.pdf [Online; accessed 11-February-2017].

[8] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Soundwave: using the doppler effect to sense gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1911–1914.

[9] 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*. ACM, 453–462.

[10] Cory Hekimian-Williams, Brandon Grant, Xiuwen Liu, Zhenghao Zhang, and Piyush Kumar. 2010. Accurate localization of RFID tags using phase difference. In *RFID, 2010 IEEE International Conference on*. IEEE, 89–96.

[11] Wenchao Huang, Yan Xiong, Xiang-Yang Li, Hao Lin, Xufei Mao, Panlong Yang, and Yunhao Liu. 2014. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 370–378.

[12] Miguel C Junger and David Feit. 1986. *Sound, structures, and their interaction*. Vol. 225. MIT press Cambridge, MA.

[13] 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 - UIST '14*. Association for Computing Machinery (ACM). https://doi.org/10.1145/2642918.2647356

[14] Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. 2015. EM-Sense: Touch Recognition of Uninstrumented, Electrical and Electromechanical Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 157–166. https://doi.org/10.1145/2807442.2807481

[15] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. 2015. Snooping keystrokes with mm-level audio ranging on a single phone. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 142–154.

[16] Kent Lyons, David Nguyen, Daniel Ashbrook, and Sean White. 2012. Facet: A Multi-segment Wrist Worn System. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 123–130. https://doi.org/10.1145/2380116.2380134

[17] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. 2004. Twiddler typing. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*. Association for Computing Machinery (ACM). https://doi.org/10.1145/985692.985777

[18] Yunfei Ma, Xiaonan Hui, and Edwin C Kan. 2016. 3d real-time indoor localization via broadband nonlinear backscatter in passive devices with centimeter precision. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 216–229.

[19] Wenguang Mao, Jian He, and Lili Qiu. 2016. CAT: high-precision acoustic motion tracking. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 69–81.
[20] Chirp Microsystems. 2013. Chirp. (2013). http://www.chirpmicro.com/technology.html [Online; accessed 11-February-2017].
[21] Adiyan Mujibiya, Xiang Cao, Desney S. Tan, Dan Morris, Shwetak N. Patel, and Jun Rekimoto. 2013. The Sound of Touch: On-body Touch and Gesture Sensing Based on Transdermal Ultrasound Propagation. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces (ITS '13)*. ACM, New York, NY, USA, 189–198. https://doi.org/10.1145/2512349.2512821
[22] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1515–1525. https://doi.org/10.1145/2858036.2858580
[23] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1515–1525.
[24] Pavel V Nikitin, Rene Martinez, Shashi Ramamurthy, Hunter Leland, Gary Spiess, and KVS Rao. 2010. Phase based spatial identification of UHF RFID tags. In *RFID, 2010 IEEE International Conference on*. IEEE, 102–109.
[25] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. 2000. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 32–43.
[26] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: enabling fine-grained hand gesture detection by decoding echo signal. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 474–485.
[27] T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling Always-available Input with Muscle-computer Interfaces. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 167–176. https://doi.org/10.1145/1622176.1622208
[28] Mark J Stefik and J Courtenay Heater. 1989. Ultrasound position input device. (March 21 1989). US Patent 4,814,552.
[29] Zheng Sun, Aveek Purohit, Raja Bose, and Pei Zhang. 2013. Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 263–276.
[30] Wei Wang, Alex X Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 82–94.
[31] Wikipedia. 2017. Inverse Square Law. (2017). https://en.wikipedia.org/wiki/Inverse-square_law [Online; accessed 11-February-2017].
[32] Haijun Xia, Tovi Grossman, and George Fitzmaurice. 2015. NanoStylus: Enhancing Input on Ultra-Small Displays with a Finger-Mounted Stylus. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*. Association for Computing Machinery (ACM). https://doi.org/10.1145/2807442.2807500
[33] Cheng Zhang, AbdelKareem Bedri, Gabriel Reyes, Bailey Bercik, Omer T. Inan, Thad E. Starner, and Gregory D. Abowd. 2016. TapSkin: Recognizing On-Skin Input for Smartwatches. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 13–22. https://doi.org/10.1145/2992154.2992187
[34] Cheng Zhang, Junrui Yang, Caleb Southern, Thad E. Starner, and Gregory D. Abowd. 2016. WatchOut: Extending Interactions on a Smartwatch with Inertial Sensing. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC '16)*. ACM, New York, NY, USA, 136–143. https://doi.org/10.1145/2971763.2971775
[35] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. SkinTrack: Using the Body As an Electrical Waveguide for Continuous Finger Tracking on the Skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1491–1503. https://doi.org/10.1145/2858036.2858082

Received November 2016; revised February 2017; accepted March 2017