

S-UbiTap: Leveraging Acoustic Dispersion for Ubiquitous and Scalable Touch Interface on Solid Surfaces

Anish Byanjankar, Yunxin Liu, *Senior Member, IEEE*, Yuanchao Shu, *Senior Member, IEEE*, Insik Shin, *Member, IEEE*, Myeongwon Choi, and Hyosu Kim, *Member, IEEE*,

Abstract—As various computing devices, such as smartphones, IoT devices, smart speakers etc, becomes omnipresent in our daily lives, interest in ubiquitous computing interfaces is increasing. In response to this, various studies have introduced on-surface input techniques that leverage the surface of surrounding objects as touch interfaces. However, most of them struggle to support ubiquitous interaction due to their dependency on specific hardware or environments. In this work, we propose S-UbiTap, an input method that turns any flat solid surface into a touch input space by listening to sound (i.e., with microphones already present in the commodity devices). More specifically, we develop a novel touch localization technique that leverages the physical phenomenon, called *dispersion*, which is the characteristic of sound as it travels through solid surfaces, and address the challenges that limit existing acoustic-based solutions in terms of portability, accuracy, usability, robustness, scalability, and responsiveness. Our extensive experiments with a prototype of S-UbiTap show that we can support sub-centimeter accuracy on various types of surfaces with minor user calibration effort. In addition, the accuracy is maintained even when the size of the touch input space increases. In our experience with real-world users, S-UbiTap significantly improves usability and robustness, thus enabling the emergence of more exciting applications.

Index Terms—Ubiquitous Computing; Human-Computer Interaction; On-Surface Touch Interface; Sound-based Localization; Time Difference of Arrival (TDoA); Dispersion

1 INTRODUCTION

IN the era of Internet-of-Things (IoT), a plethora of computing devices, such as smartphones, wearable devices, smart speakers, and smart mirrors, have become omnipresent. However, interactions with these ubiquitous devices are difficult owing to a significant decrease in their form factor. In response to this concern, one of the emerging trends is to use surrounding objects having a large surface, such as tables, walls, and mirrors, as touch interfaces. Several previous works have developed such on-the-fly interfaces by introducing techniques that track a user's fingertip on a surface. None of these works, however, fully support scalable, ubiquitous, and seamless interactivity in on-surface touch interfaces, mainly because of their dependency on specific infrastructure [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].

This work introduces S-UbiTap, a novel approach for supporting scalable and precise on-surface touch input localization without losing portability; by listening to sound generated while tapping a surface. Figure 1 illustrates a typical usage scenario of S-UbiTap, which converts a large



Fig. 1: S-UbiTap enables touch inputs on a screen which is projected onto a whiteboard in a classroom, with the use of commodity smartphones.

whiteboard into a touch input interface.¹ First, S-UbiTap records a *touchsound*, an impact sound produced when tapping a surface, with multiple (at least three) microphones. Such a multi-microphone system can be easily built as microphones are usually pre-installed in commodity mobile devices. Then, it pinpoints the tapping location on the surface by analyzing the propagation pattern of the recorded sounds.

However, supporting in-situ touch interfaces with touch-sounds is much challenging. This is primarily because users can utilize touch input methods in a wide variety of envi-

1. See https://youtu.be/29OsCeap_nE for more usage scenarios.

ronments, such as indoors/outdoors, day/night, small/large input space, and alone/with friends. The diverse nature of use cases results in unpredictable and dynamic changes in the characteristics of touchsounds. This challenge has compelled existing approaches to have trade-offs between scalability, accuracy, portability, robustness, and usability. Classification-based works [13], [14] achieve high accuracy while placing a heavy training burden on users for every setup. The training burden further increases with an increase in the size of a touch input space. Other works seek to reduce user efforts, by leveraging the Time-Difference-of-Arrival (TDoA) between microphones [15], [16], [17], [18], [19]. However, some of these studies [15] suffer from poor accuracy because of complex acoustic phenomena, e.g., dispersion on solid surfaces. Others tackle this problem through the usage of specific hardware, such as a time-synchronized vibration sensor array [16], [17], [18], [19].

S-UbiTap addresses the limitations of the existing works via an in-depth exploration of acoustic dispersion phenomena. As sound travels along surfaces, its frequency components are transmitted at various speeds depending on their carrying frequency [20]. In particular, on a flat surface, the propagation speed of each frequency wave remains constant across the entire area and even with changes in the surrounding environment. That is, different frequency components reach a microphone at different times and the difference in their arrival times increases in proportion to their propagation distance to the microphone. This linear relationship makes it possible to estimate the distance from a touch point to a microphone by leveraging the TDoA between the frequency components of touchsounds.

Based on this, we design a dispersion-aware touch localization system that supports accurate, usable, robust, responsive, scalable, and ubiquitous touch inputs on surfaces. The proposed system consists of three key techniques:

- We develop a simple yet robust calibration method that estimates surface-dependent parameters (e.g., propagation speeds). Specifically, based on our observations that propagation speeds are constant over the entire area for a certain flat surface, we compute the parameters with a small number of calibration points. Furthermore, for previously calibrated surfaces, there is no need for additional calibration because changes in the surroundings rarely affect the propagation speeds.
- We introduce a new arrival time detection technique that accurately pinpoints the frequency-specific arrival times of touchsounds with low computation latency. It applies different time-frequency analysis schemes depending on the frequency component's dispersion properties.
- We design a scalable touch localization algorithm. It first determines a set of effective microphones, which allows more accurate localization on large surfaces, by leveraging the dispersion phenomenon. For each selected microphone, it individually measures the propagation distance of touchsounds using the information obtained through our calibration and arrival time detection techniques. It then localizes touch inputs by combining these distance results.

We evaluate the performance of S-UbiTap by means

of our prototype implementation, which builds a 17-inch touchscreen on common objects (e.g., a wooden table, a glass mirror, and an acrylic board). Our evaluation results show that S-UbiTap can easily achieve sub-centimeter localization accuracy on common surfaces, without compromising usability. For example, on a wooden table, the 98th percentile error of 0.76 cm is achieved with only 18 calibration points. In addition, S-UbiTap maintains sub-centimeter level accuracy even in the presence of dynamic environmental changes (e.g., displacement of nearby objects) and on large surfaces. Our experiment with real-world users gives positive results, especially in terms of usability. The user study results also show that S-UbiTap is highly capable of supporting various user-friendly applications (e.g., multiplayer games).

The key contributions of our work can be summarized as follows:

- To the best of our knowledge, this work is the first attempt to explore the feasibility of exploiting dispersion phenomena to enable ubiquitous touch interfaces on solid surfaces.
- We design S-UbiTap, a novel dispersion-aware framework for on-surface touch localization, that satisfies the following requirements: a high degree of portability, accuracy, usability, robustness, scalability, and responsiveness. In particular, we present a scalable design that can be easily extended to support a large-sized touch input space.
- We implement a prototype of S-UbiTap and demonstrate its effectiveness through extensive benchmark tests and real-world user studies.

2 REQUIREMENTS AND CHALLENGES

S-UbiTap builds a ubiquitous on-surface touch input system by using microphones, which are present in most commodity devices, i.e., by capturing and analyzing *touchsounds*, produced due to touch events on surfaces. Having such technology may encourage the emergence of more user-friendly applications.

Ad-hoc touchscreen construction. Users can use a touchscreen of any size regardless of time or place. For instance, at a campsite, a group of friends may want to play board games such as chess or monopoly, but they may lack the equipment for such a game. It would be a great experience for them to convert a dining table into a virtual game board, with only the devices they carry (e.g., with the built-in microphones of smartphones and a portable projector).

Adding a touch feature to existing smart devices. We can support touch functions in surrounding smart devices, such as smart TVs and mirrors, which have inconvenient or no input methods. In particular, by using the built-in microphones of these devices, we can provide users the touch input interface at no additional cost.

Supporting large scale touch interface. Large-sized screens are common in classrooms, museums, and airports. However, very few of these digital screens interact with users because it is extremely expensive to provide touch interfaces with state-of-the-art techniques on such large screens. For example, a 15-inch capacitive touch panel costs approximately USD 100,

and scaling this up to larger screens can cost even more. In contrast, S-UbiTap can reduce the cost of constructing large-scale touch interfaces by using only a few number of microphones.

2.1 Requirements

From the above applications, we can identify the six major requirements of S-UbiTap as follows:

- *Portability.* To support the in-situ construction of touch input systems, S-UbiTap must not use a dedicated infrastructure.
- *Accuracy.* The gap between two adjacent touch points can be a few centimeters only in various use-case scenarios (e.g., keyboard applications and board games). Thus, the user's touch position should be localized with sub-centimeter accuracy.
- *Usability.* S-UbiTap needs to be easy to set up and use, requiring minimum time and effort. In other words, calibration and training time should be low.
- *Robustness.* Users should be able to use S-UbiTap in various situations. It should work accurately regardless of the environment and its accuracy should not drop with minor changes in the environment.
- *Responsiveness.* Users must receive feedback for their touch inputs without any perceivable latency. The system must be user-interactive by responding to a user's touch input within 100 ms [21].
- *Scalability.* It should be possible to support touch inputs on any size of surfaces. In particular, S-UbiTap must meet other requirements even in supporting large-sized touch interfaces.

2.2 Challenges

The relationship between touchesounds and touch locations basically depends on the physical properties of a given environment. For example, the material, size, and boundary condition of a surface can affect not only surface vibration patterns but also the path and speed of touchesound propagation. In practice, touch input systems can be used in a variety of dynamically changing environments, resulting in unpredictable and dynamic changes in touchesounds.

- *Different surfaces.* As we aim to support an in-situ touch input interface on any flat surface, users may use various kinds of surfaces, such as glass tables, wooden tables, mirrors, walls, and acrylic/plastic boards. These surfaces have different physical properties, making touchesounds have unique signatures and travel at different speeds depending on the surface..
- *Relocation of touch input space.* A touch input space can be rearranged to another area on the same surface, producing a different touchesound for the same input. As an example, families can enjoy touch-based board games at dinner tables every evening. However, it may not be possible to play exactly at the same place on the table each time.
- *Changes in the surrounding environment.* While using an on-surface touch input system, users can place or

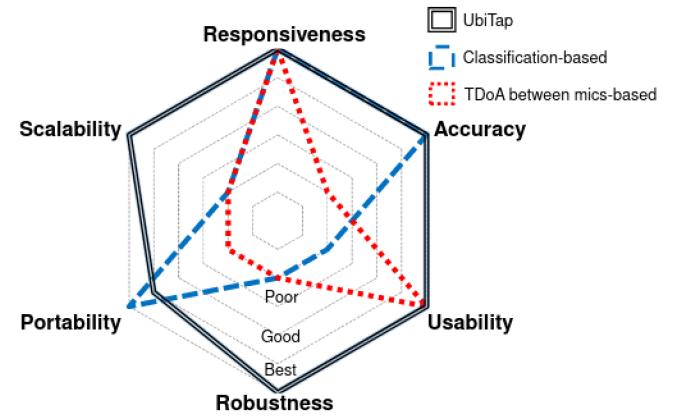


Fig. 2: Comparison of acoustic-based touch localization methods.

move objects on the surface. For example, the number of books stacked on a table can vary depending on the time, even during touch-based games on the table. This object displacement can alter multipath reflection patterns.

Limitations of existing works. Figure 2 indicates that prior acoustic-based approaches have limitations with regard to fulfilling the aforementioned requirements of S-UbiTap.

- *Classification-based: Low usability, robustness, and scalability.* One class of the previous works [13], [14] collects a set of heavy training data to characterize touchesounds for a given environment. That is, this sacrifices usability for accuracy. In addition, the burden of training further increases with an increase in the size of the input space. Even worse, the signatures of touchesounds, the classification features of these methods, change unpredictably depending on the environment, causing a significant performance drop.
- *TDoA between microphones-based: Low accuracy, portability, and scalability.* Other works [15], [16], [17], [18], [19] have tried to reduce calibration efforts by leveraging the TDoA between microphones. However, for a precise TDoA measurement, these works require dedicated hardware containing perfectly time-synchronized microphones [15], [18], [19], or geophones [16] or accelerometer, [17]. Moreover, Toffee [15] suffers from low accuracy because surface-borne sound undergoes dispersion and causes variations in the TDoA depending on the frequencies. In addition, all the existing approaches are designed to localize touch inputs only on small-sized surfaces. For instance, the possibility of supporting larger touch input areas with multiple sensors has not been explored in these works.

Conversely, S-UbiTap shows high performance in all of the categories, as shown in Figure 2. The key idea of S-UbiTap offering this level of quality is that it leverages the dispersion properties of touchesounds. In the remainder of this paper, we first describe the properties of dispersion that provide grounds for developing S-UbiTap (Section 3). We then describe how such properties are incorporated into the design of S-UbiTap (Section 4). Note that S-UbiTap uses at least three microphones, causing a subtle decrease

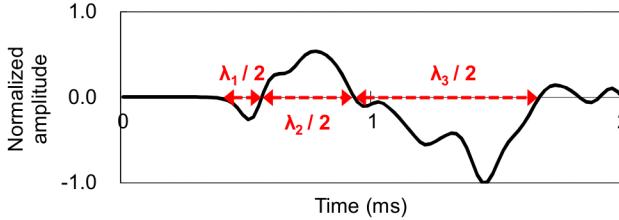


Fig. 3: Early part of a touchsound: higher frequency waves arrive earlier than lower ones (λ_i indicates the i -th approximate local wavelength of the touchsound). Note that, in our preliminary observations, we captured touchsounds on a wooden table by using a fingernail tip.

in portability as shown in Figure 2. However, our system design reduces a user's burden to construct such a multi-microphone system by allowing them to use microphones of different devices (e.g., microphones in smartphones and smartwatches).

3 ACOUSTIC DISPERSION

This section deeply explores the core characteristics of surface-borne touchsounds, i.e., *acoustic dispersion*.

Whenever a surface is tapped, the impact of the tap creates a touchsound. The touchsound spreads through the surface in a transverse manner, causing the air pressure around its passage to change. At this time, acoustic dispersion is observed [20]. A solid-like surface is a dispersive medium and transmits waves of different frequencies at different speeds. Therefore, different frequency components of the touchsound arrive at microphones at different times. The touchsound is also transferred from the touch location to the microphones through non-dispersive mediums, such as air. However, as air-borne sound is much slower than surface-borne sound, we can easily distinguish them and analyze the dispersive behaviors in audio recordings. For example, Figure 3 shows that the local wavelength of the touchsound increases over time. This indicates that on solid surfaces, the propagation speed of a lower-frequency wave is slower than that of a higher-frequency wave.

Key principle 1. Sounds having different frequencies propagate at different speeds $V(f)$, where f is their carrying frequency.

The physical properties of a surface (e.g., density, thickness, Young's modulus, and Poisson ratio) determine the propagation speed $V(f)$ on that surface [20]. For example, touchsounds travel slower on a wooden table than on a steel table due to the difference in their stiffness. On a common flat surface, such as an office desk and a mirror, $V(f)$ is almost constant over the entire area. This is because each part of the surface has similar physical properties. In addition, as $V(f)$ is dependent on the physical properties of the surface, $V(f)$ remains unaltered with changes in the ambient environment (e.g., ambient noise and nearby objects). Note that on surfaces with irregular or directional properties, $V(f)$ might differ from region to region. This is discussed in Section 9.

Key principle 2. For a certain surface, if each part of the surface has similar physical characteristics, $V(f)$ is constant regardless of the touch location and the surrounding environment.

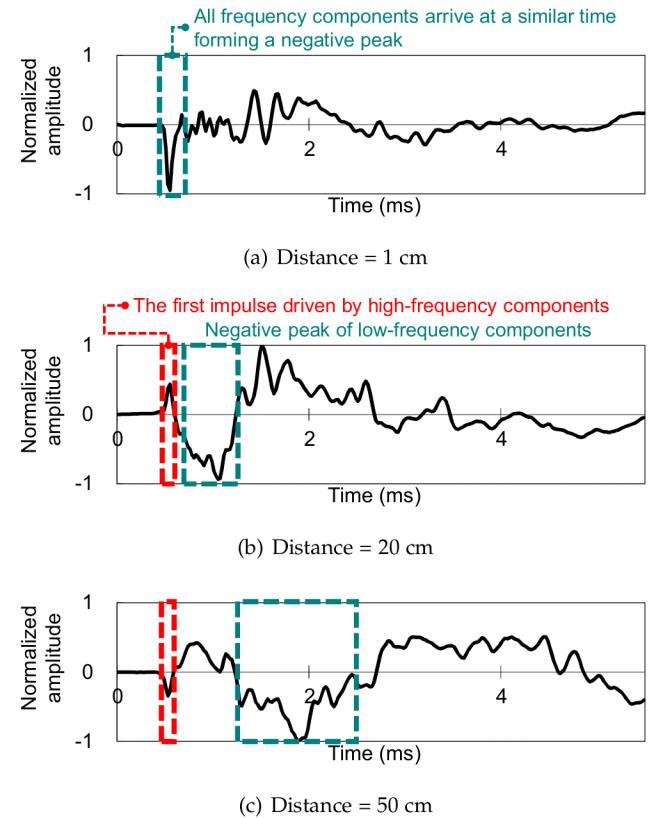


Fig. 4: Structure of a touchsound; the arrival time difference between frequency components grows as the propagation distance increases.

Figure 4 shows the effects of the dispersion phenomenon on the arrival time of touchsounds. When a touch input is made near a microphone (e.g., 1 cm away), all the frequency components of the touchsound arrive at a similar time, forming a negative peak (shown in Figure 4(a)). The microphone is typically placed on a surface, but the touch impact deforms the surface in the opposite direction, i.e., downward. This causes the touchsound to be captured in a shape with a negative peak. However, as the propagation distance increases, other smaller peaks are observed before the negative peak, shown in Figure 4(b) and 4(c). We can also observe that the arrival time difference between the first impulse and the negative peak grows gradually with the increase in the propagation distance. Such a difference in arrival time occurs because the high-frequency components (i.e., above 18 kHz) propagate much faster than the low-frequency components (e.g., under 1 kHz).

Key principle 3. The TDoA between two different frequency components of a touchsound is linearly proportional to the propagation distance of the touchsound to a microphone (denoted as D), as follows:

$$T^A(f_i) - T^A(f_j) = D \cdot \left(\frac{1}{V(f_i)} - \frac{1}{V(f_j)} \right), \quad (1)$$

where $T^A(f)$ is the arrival time of the touchsound at frequency f .

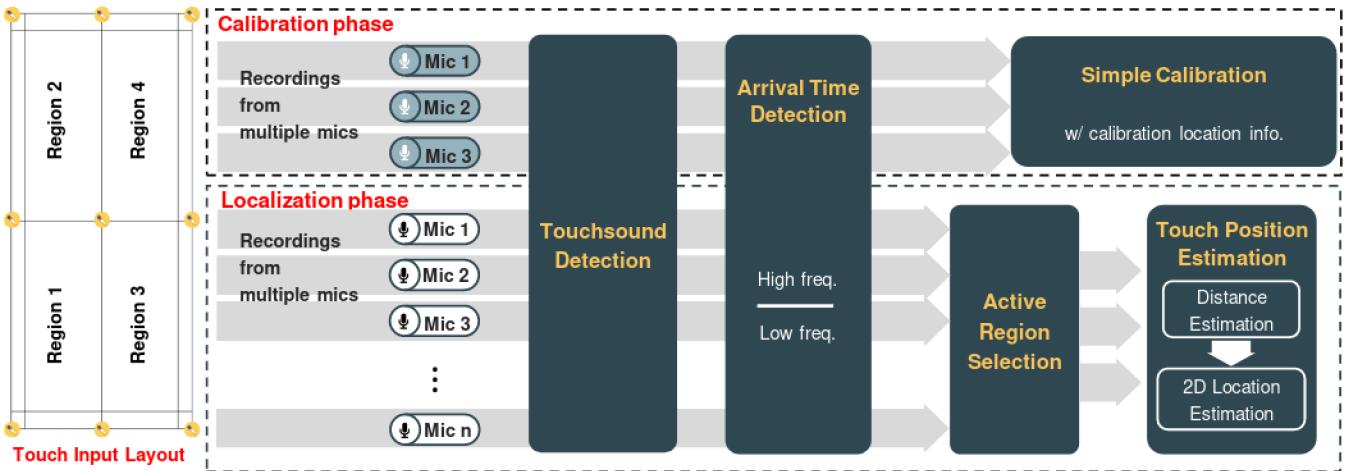


Fig. 5: Architecture of S-UbiTap.

4 S-UBITAP SYSTEM DESIGN

S-UbiTap is designed with the primary goal of supporting scalable, accurate, usable, robust, responsive, and portable in-situ touch input interfaces. This design goal can be achieved by leveraging the dispersive principles of touchsound. In particular, four key techniques are developed to realize this goal: i) arrival time detection, ii) calibration, iii) active region selection, and iv) touch position estimation as shown in Figure 5. First, once a touch input is made, the arrival time of each frequency component is estimated. S-UbiTap then calculates the TDoA between high- and low-frequency components. Note that despite travelling the same propagation distance, calculated TDoA values can have significant differences depending on the material of the surface. Thus, these surface-dependent parameters are first determined in the calibration phase and used to locate the touch location in the localization phase.

Technical challenges. The most important problem when designing such a dispersion-aware touch localization framework is to pinpoint the arrival time of individual frequency waves ($T^A(f)$) in an accurate and robust manner. However, in practice, such pinpointing of various individual frequency components is very complicated for multiple reasons.

- Interferences across different frequency components occur frequently. For example, adjacent frequency components can arrive at a similar time and cause interference. Furthermore, reflections of higher frequency components can disrupt the arriving waves of lower-frequency components due to differences in their propagation speed. This necessitates the use of high-resolution time-frequency analysis methods, but their high computational complexity can compromise the responsiveness of the system.
- As the size of touchable areas increases, touchsounds get attenuated, i.e., the Signal-to-Noise Ratio (SNR) of touchsounds decreases. More specifically, as observed in Section 9, high-frequency components of touchsounds have much lower amplitudes compared to low-frequency components, causing difficulties in accurately estimating the arrival time of high-frequency components on large surfaces. This eventually causes more errors in localiza-

tion.

We design S-UbiTap to address these challenges and meet the six fundamental requirements for ubiquitous touch interfaces as follows:

- *Accuracy and responsiveness improvement.* We develop an accurate, yet fast method to detect the frequency-specific arrival times of touchsounds, by applying different time-frequency analysis techniques to the high- and low-frequency components based on their dispersion properties.
- *Portability improvement.* S-UbiTap individually estimates the propagation distance of touchsounds for each microphone. The measured propagation distance from each microphone is then combined to pinpoint the 2D location of the touch inputs. Such individual microphone measurement allows us to use microphones present in different devices even without the need for time-synchronization between them.
- *Usability and robustness improvement.* Our accurate arrival time estimation can be synergistic with the dispersive principles, to enhance usability and robustness. For example, the high accuracy of measuring arrival times and the consistency of propagation speeds across the entire surface enable the precise estimation of surface-dependent speeds with only a few calibration points. Furthermore, for a calibrated surface, we can always maintain a high degree of localization accuracy with the robustness of propagation speeds against changes in the surrounding environments.
- *Scalability improvement.* We enable scalable on-surface touch interfaces without compromising accuracy. We first partition a large surface into multiple smaller regions based on the placement of microphones (see Figure 5). We then determine the region in which a touch input is made, called the active region, and conduct precise localization for the region with the corresponding microphones.

S-UbiTap assumes the following four usage conditions: i) touchsounds have an energy level high enough to analyze, ii) users have three or more microphones to use a distance-based

localization approach, iii) there is an on-surface screen, and iv) the screen's size and position relative to the microphones are known in advance.

The first condition can be fulfilled simply by asking users to generate a touchsound with a significant energy level by tapping the surface with hard objects like pens, stylus, or even fingernail tips. We can also easily satisfy the second condition through the collaborative use of multiple commodity devices. It is very common for a single user to carry a smartphone, smartwatch, and, tablets. In addition, such microphone array can also be constructed by using devices of different users in case of multi-user environments. For the third requirement, we can instantly construct on-surface screens using a portable projector (e.g., Samsung bean [22]). We can also simply print out the layout of the input interfaces, e.g., keyboard, on a paper if the layout is static. Finally, the fourth condition can be satisfied by placing microphones at pre-defined locations, such as a corner of the screen. At this time, human errors in placing these microphones can be minimized by leveraging phone-to-phone localization techniques [23], [24].

5 ARRIVAL TIME MEASUREMENT

In this section, we describe how S-UbiTap accurately detects touchsounds in sensor streams and pinpoints their frequency-specific arrival times with a reasonable amount of computation latency.

5.1 Touchsound Detection

Extracting touchsounds from the audio stream/recording can be done simply using a sound level threshold because touchsounds have high amplitudes, as illustrated in Figure 4. However, the use of this simple approach cannot distinguish touchsounds from bursty noise (e.g., human voices) and thus entails false positives. S-UbiTap addresses this with the help of motion sensors, e.g., gyroscopes, as proposed in UbiK [13]. This previous work showed that motion sensor readings are only altered by surface vibrations, but not by acoustic noise propagated through the air. This ensures robust touchsound detection even in the presence of noise².

Touchsound detection. The energy level of audio and motion signals are first examined to determine the existence of touchsounds. Towards this, S-UbiTap calculates the accumulated energy in sound at time t , denoted as $E^S(t)$, as,

$$E^S(t) = \sum_{i=t-T^W}^t X^S(i)^2, \quad (2)$$

where $X^S(t)$ is the received audio signal at time t , and T^W , the window size, is empirically set to 1 ms. S-UbiTap then compares $E^S(t)$ with the threshold ϵ^S that is configured as a quarter of the maximum accumulated energy measured during the calibration step. Similarly, the accumulated energy level of gyroscope readings $E^M(t)$ and its threshold ϵ^M are calculated in the same way. Thus, the existence of a touch input is declared at time t , if i) both $E^S(t)$ and $E^M(t)$ exceed

2. UbiK has achieved nearly 100% detection accuracy in noisy environments, such as in food courts [13]. In this paper, we do not conduct further experiments for touchsound detection.

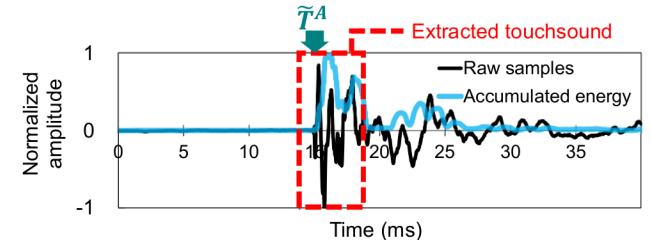


Fig. 6: Typical structure of a touchsound; the highest energy level is observed in an early part of the touchsound due to directly-arriving components.

ϵ^S and ϵ^M , respectively, and ii) the time gap from the last touch event exceeds a safe margin. We set the margin to 200 ms, a typical input interval observed in a previous field study [25].

Touchsound segmentation. After detecting a touchsound, its early part is extracted for the estimation of arrival times. Let assume that a touchsound is detected at time t . S-UbiTap approximates the starting point of the touchsound, \tilde{T}^A , in time window $[t - T^G, t + T^G]$. Here, the guard interval T^G is set to 20 ms, the typical duration of a touchsound. Then, the low-frequency component (below 5 kHz) of the signal is removed for the following reasons: i) waves of higher-frequency components arrive earlier than those of lower-frequency components and ii) ambient noise (e.g., human voices, audible signals) has a high level of energy at the low frequencies. The filtered signal is then used to calculate the accumulated energy at each time instant and find the maximum level E^* among them. S-UbiTap then determines \tilde{T}^A as the first time instant where the accumulated energy is greater than $E^*/20$ and takes the original audio signal (i.e., the non-filtered raw signal) between $[\tilde{T}^A - T^W, \tilde{T}^A + T^W + T^G/4]$ (see Figure 6).

5.2 Pinpointing Arrival Times

As briefly described in Section 4, there is a trade-off between accuracy and responsiveness when measuring the arrival times of touchsounds. Using high-resolution time-frequency analysis (e.g., Wigner-Ville distribution (WVD)) provides an accurate estimation of arrival time, but with high computational complexity (e.g., $O(k^2 \log k)$, where k is the sample length). Therefore, incorporating such high-resolution methods leads to a significant performance drop in terms of responsiveness. By contrast, computationally-efficient techniques (e.g., short-term Fourier and continuous wavelet transforms) can also be used for the arrival time estimation. However, these techniques compromise accuracy because of their low-resolution support. Hence, in order to address this trade-off, we further explore the frequency-dependent characteristics of touchsounds and leverage these observations in designing our arrival time measurement technique.

Arrival time estimation of high-frequency waves. High-frequency components (e.g., > 18 kHz) waves are observed at the very early part of a received touchsound (shown in Figure 4). This implies that analyzing a small number of samples (e.g., 1 ms) of the signal is sufficient to pinpoint

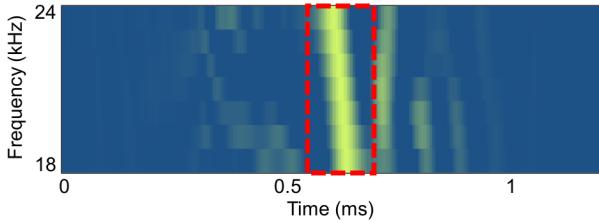


Fig. 7: Extracted WVD features of a touchsound at high frequencies; the directly-arriving sound shows high magnitude due to its shortest propagation length. Note that the brighter the WVD feature is, the louder the sound is at the corresponding time and frequency.

high-frequency components. In other words, the search range is sufficiently small, allowing us to use high-resolution analysis methods, such as WVD, without compromising responsiveness.

Based on this idea, S-UbiTap segments the first 256 samples (i.e., approximately 1.3 ms with a sampling rate of 192 kHz) from a touchsound obtained in the detection step. Then, a Hanning window [26] is applied to assign more weights to signals near \tilde{T}^A and WVD features are extracted from the windowed samples. Let the extracted WVD feature at time t and at frequency f be denoted by $WVD(t, f)$. For each frequency f_i^H in \mathcal{F}^H , a set of high frequencies, S-UbiTap estimates its arrival time $T^A(f_i^H)$ as the first time instant that shows a high magnitude (i.e., the highlighted points in Figure 7) as follows:

$$T^A(f_i^H) = \min\{t \mid \sum_{k=-N^G}^{N^G} WVD(t, f_{i+k}^H) \geq \frac{1}{2} WVD^*(f_i^H)\}, \quad (3)$$

where N^G , a guard frequency interval for increasing noise robustness, is empirically set to 2 and $WVD^*(f_i^H)$ is the maximum accumulated magnitude at frequency f_i^H , i.e., $\max_{\forall j} \sum_{k=-N^G}^{N^G} WVD(j, f_{i+k}^H)$. It is noteworthy that \mathcal{F}^H is configured as a range from 18 kHz to 24 kHz because touchsounds in this range show a reasonable degree of amplitude on most common surfaces (e.g., wood, glass, and metal). More specifically, each frequency in \mathcal{F}^H is separated by 1 kHz.

Arrival time estimation of low-frequency waves. As observed in Section 3, the arrival time difference between low- and high-frequency components varies with respect to the change in the propagation distance of touchsounds. For example, the time gap increases from 1 ms to 2 ms with an increase in propagation distance from 20 cm to 50 (see Figure 4). This will increase further with an increase in the propagation distance. Therefore, the application of high-resolution techniques (Wigner-Ville Distribution) induces a high computational complexity, thereby affecting the responsiveness of the system. Instead, S-UbiTap designs a simple approach based on the characteristics of touchsounds at low frequencies. As shown in Figure 4, a low-frequency wave always arrives in the form of a negative peak and shows the highest level of energy. This means that by simply pinpointing the minimum peak of the signal, we can accurately estimate the arrival of low-frequency components.

S-UbiTap builds a Butterworth filter bank [27] to separate input signals into 20 different low-frequency components. For instance, the i -th filter has an order of 6 with a center frequency of f_i^L , which is set to $i \cdot 100$ Hz. Based on this filter bank design, S-UbiTap breaks the detected touchsound into multiple subband signals. Note that before the subband separation process, it applies a half-Hanning window (the first half of the window) to the first $\min_{f \in \mathcal{F}^H} T^A(f)$ samples so as to mitigate the effect of the noise captured before the touchsound. It then finds $T^A(f_i^L)$, the arrival time of the touchsound at f_i^L , as the point at which the lowest amplitude appears in its corresponding subband signal FB_i ,

$$T^A(f_i^L) = \arg \min_{t=0,1,\dots,T_i^U} FB_i(t), \quad (4)$$

where T_i^U , the upper limit of $T^A(f_i^L)$, is set to $T^A(f_{i-1}^L)$ if $i \geq 2$, or is set to the length of the detected sound otherwise. That is, to determine $T^A(f_i^L)$, we consider the dispersion phenomenon that makes waves with higher frequencies arrive earlier than those with lower frequencies.

6 TOUCH LOCALIZATION

The basic idea for localizing touch inputs is to leverage the linear relationship between the propagation distance of touchsounds and the TDoA between frequency components. Let $\Delta T^A(f^H, f^L)$ denote the TDoA between high-frequency f^H and low-frequency f^L waves, measured by our arrival time estimation technique. $\Delta T^A(f^H, f^L)$ increases linearly when the propagation distance of the touchsounds D grows (see Figure 8). However, it contains non-zero but constant errors, denoted by $U(f^H, f^L)$. This arises because S-UbiTap identifies the arrival time of each frequency component as the time instant that shows an energy level high enough to distinguish the signal from noise. Such a threshold-based method inherently leads to errors because the estimated arrival times are not identical to the actual times. Thus, the relationship between $\Delta T^A(f^H, f^L)$ and D can be represented as follows:

$$\Delta T^A(f^H, f^L) = D \cdot I(f^H, f^L) + U(f^H, f^L), \quad (5)$$

where $I(f^H, f^L)$ is the difference between the inverse propagation speed at f^H and f^L , i.e., $V(f^H)^{-1} - V(f^L)^{-1}$. Note that using different types of touch tools can vary $U(f^H, f^L)$. We will show the impact of touch tool variations in Section 8.

Based on this relationship, S-UbiTap computes the environment-dependent parameters (in the calibration phase) or the propagation distance of touchsounds (in the localization phase). Then, it localizes touch inputs with the distance information obtained from multiple microphones. Note that, for large-sized surfaces, we split the area into multiple small regions based on the placement of microphones. Figure 9 illustrates a typical layout of the large-sized surfaces. It consists of multiple sub-regions, each of which has three or more microphones placed on the corners of the region. Based on this, we conduct calibration and localization independently for each region.

6.1 Simple Calibration

Before enabling touch interaction on a certain surface, S-UbiTap conducts user-involved calibration to configure

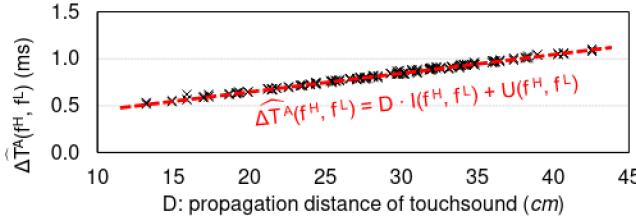


Fig. 8: Relationship between the propagation distance of touchsounds and the TDoA between high-frequency ($f^H = 18\text{ kHz}$) and low-frequency ($f^L = 1\text{ kHz}$) waves.

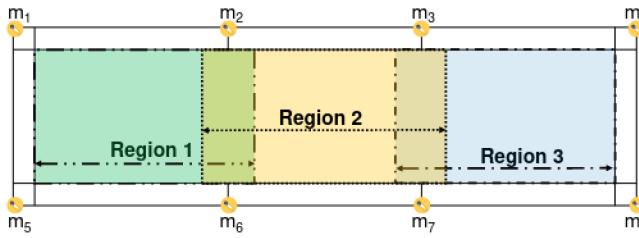


Fig. 9: Layout of a very large touch surface divided into smaller regions. There are some overlapping regions to properly handle errors in selecting active regions.

not only the surface-dependent values $I(f^H, f^L)$ but also the estimation errors $U(f^H, f^L)$.

First, users are asked to tap N^C pre-defined locations on a touch input space. For each microphone, S-UbiTap then extracts the frequency-specific arrival times of the calibration inputs and computes $I_i(f^H, f^L)$ as the mean value of the slopes, calculated for every pair of the calibration inputs,

$$I_i(f^H, f^L) = \frac{1}{\binom{N^C}{2}} \sum_{j=1}^{N^C-1} \sum_{k=j+1}^{N^C} \frac{\widehat{\Delta T}_{i,j}(f^H, f^L) - \widehat{\Delta T}_{i,k}(f^H, f^L)}{D_{i,j}^C - D_{i,k}^C}, \quad (6)$$

where $D_{i,j}^C$ is the distance between the i -th microphone and the j -th calibration point and $\widehat{\Delta T}_{i,j}(f^H, f^L)$ indicates the TDoA between high (f^H) and low (f^L) frequency components, estimated at the i -th microphone with the j -th calibration input. S-UbiTap finally calculates the estimation error $U_i(f^H, f^L)$ as follows:

$$U_i(f^H, f^L) = \frac{1}{N^C} \sum_{j=1}^{N^C} \widehat{\Delta T}_{i,j}(f^H, f^L) - I_i(f^H, f^L) \cdot D_{i,j}^C. \quad (7)$$

We compute the environment-dependent parameters for all pairs of f^H and f^L , where $f^H \in \mathcal{F}^H$ and $f^L \in \mathcal{F}^L$.

As discussed earlier, when a large-sized surface is used, we divide the single large surface into multiple smaller regions (see Figure 9). At this time, we collect a set of calibration data and calculate $I(f^H, f^L)$ and $U(f^H, f^L)$ individually for each region by using the microphones placed on the region. For example, in Figure 9, the four microphones m_1, m_2, m_5 , and m_6 are used to conduct the calibration for the region 1.

6.2 Active Region Selection

To support precise localization on large-sized surfaces, we use touchsounds collected by microphones close to touch locations, i.e., the microphones placed on a region where a touch input is made, called *active region*. This is because these microphones, we call them effective microphones, can capture touchsounds with higher SNR compared to others. One naive approach is to select effective microphones based on the amplitude of their received touchsound. However, given that different devices have different microphone properties, such as sensitivity, this simple approach provides poor accuracy in choosing the effective microphones or requires additional calibration for better selection. Another way is to leverage the TDoA between microphones. However, this works well only when all the microphones are perfectly synchronized.

Instead, S-UbiTap determines an active region by leveraging the dispersive characteristics of touchsounds. Specifically, we use the TDoA between only low-frequency components. As discussed in Section 4, when touchsounds travel long distances, estimating the arrival time of high-frequency components is quite difficult because of their low amplitude. Such attenuation is observed in low-frequency components too. However, they still show a sufficient SNR for the arrival time measurement. Based on this, for each microphone, S-UbiTap first computes the TDoA between low-frequency components, denoted by $\widehat{\Delta T}_{i,j}^A(f^L, f_j^L)$, where both f_i^L and f_j^L are in \mathcal{F}^L . $\widehat{\Delta T}_{i,j}^A(f^L, f_j^L)$ linearly increases with an increase in the traveling distance of touchsounds. We then choose top- k microphones with the lowest TDoA values and determine the active region as the region where the highest number of microphones among the selected ones are placed. Note that we empirically set k to three. We further improve the accuracy in the active region selection by repeatedly performing this process for every pair of low-frequency components. In other words, we make $\binom{|\mathcal{F}^L|}{2}$ decisions for the active region selection and conduct a majority voting.

6.3 Touch Position Estimation

Once an active region is selected, S-UbiTap identifies the 2D location of an actual touch input based on the measured TDoA values and calibrated parameters. First, for each microphone placed on the selected region, S-UbiTap calculates \widehat{D}_i , the distance between the touch location and the i -th microphone, as follows:

$$\widehat{D}_i = \frac{1}{N^H \cdot N^L} \sum_{j=1}^{N^H} \sum_{k=1}^{N^L} \frac{\widehat{\Delta T}_i^A(f_j^H, f_k^L) - U_i(f_j^H, f_k^L)}{I_i(f_j^H, f_k^L)}, \quad (8)$$

where N^H and N^L represent the numbers of high- and low-frequency components, respectively.

S-UbiTap then localizes the touch input using a least squared error method with the distance information measured from multiple microphones. It first establishes a set of possible touch positions \mathcal{P} that contains 2,500 evenly-distributed positions on a touch input space. For each

candidate position P_i in \mathcal{P} , S-UbiTap computes its squared error SE_i as follows:

$$SE_i = \sum_{j=1}^{N^M} (\| M_j - P_i \| - \widehat{D}_j)^2, \quad (9)$$

where N^M is the number of microphones and M_j is the location of the j -th microphone. Finally, S-UbiTap determines the touch location as the point P_i that shows the least squared error among all possible positions in \mathcal{P} .

Note that there might be an error in selecting active regions, especially when touch inputs are given near the border of regions. Therefore, we slightly extend (approximately 10 cm) the touchable area for each region as shown in Figure 9 and localize touch inputs on top of this extended layout. This allows S-UbiTap to accurately identify the location of touch inputs even if their active region is selected incorrectly.

7 IMPLEMENTATION

We implemented a prototype of S-UbiTap on commodity smartphones (e.g., Pixel, Pixel XL, Pixel 4 XL, and Samsung S9) running different versions of the Android platform.

System configurations. The implementation captures raw sensor data using microphones and gyroscope sensors; this is done with a sampling rate of 192 kHz for the microphones and approximately 220 Hz for the motion sensors. It receives acoustic streams from an audio device buffer every 10 ms. Specifically, it records sound using UNPROCESSED audio source to eliminate the effect of manufacturer-specific preprocessing techniques.

Hardware limitations. There are several limitations when using microphones and existing mobile devices. First, when a touch input is made near the microphones, the received sound is clipped as it arrives with a very high amplitude. To avoid audio clipping, the microphones should be kept slightly away (e.g., 5 to 10 cm away) from the touch input space.

Another limitation is the location of the built-in microphones in mobile devices. Mobile devices have multiple microphones to support advanced features. However, microphone placement is not optimized for touch localization. They are positioned at opposite locations, such as the top and bottom of a smartphone. Therefore, depending on the touch location, some microphones can capture touch sounds well, whereas others cannot, which can reduce localization accuracy. To address these limitations, S-UbiTap prototype supports multi-device environments that use the built-in microphones of many mobile devices together. In particular, in the multi-microphone system, each microphone is placed facing a touch input space to avoid the microphones directivity issues, i.e., to further enhance localization accuracy, as shown in Figure 1.

8 EVALUATION

In this section, we evaluate S-UbiTap by answering the following questions:

- *Accuracy.* How accurately can S-UbiTap localize touch events on various surfaces?

- *Usability.* How much user effort is required for conducting calibration?
- *Robustness.* How robustly can S-UbiTap identify touch locations in dynamically-changing environments?
- *Responsiveness.* Does S-UbiTap respond to a user's touch inputs without any noticeable latency?
- *Scalability.* How accurately can S-UbiTap support touch interfaces on large surfaces?
- *Real-world user experience.* How does a touch-based application running on S-UbiTap work with real-world users?

8.1 Evaluation Setup and Methodology

We basically conducted experiments with an on-surface touchscreen measuring 36 cm x 24 cm (i.e., a 17-inch screen) (see Figure 10(a)). Note that the size of the screen may differ depending on the experiment. The touchscreen was constructed by using one portable projector (SK Smart Beam) and the back-side microphones of four smartphones (two Google Pixel and two Google Pixel XL devices). Note that we positioned the smartphones at some distance from each corner of the projected screen so as to avoid the audio clipping problem, as discussed in Section 7. The screen was installed on a wooden table measuring 160 cm x 80 cm x 72 cm in an office. We chose a wooden surface as our default environment because it is the most commonly used surface type. To observe the effects of different surfaces, we also used other surfaces, including a glass mirror and an acrylic board, which have immense potential for smart devices such as smart mirrors.

During the experiments, we first asked a single user to tap all calibration points five times before using the touchscreen. Figure 10(b) illustrates the points used for the calibration. Depending on the purpose of each experiment, we changed the total number of calibration points N^C by adjusting the distance between two consecutive calibration points L^C , where $N^C = ((40 \text{ cm}/L^C) + 1) \cdot 2$. The user then made touch inputs on the displayed circles (77 in total), each of which is separated by 3 cm on both the X and Y axes, as shown in Figure 10(a). Specifically, all circles were tapped sequentially, each repeated 10 times. The user utilized her fingernail tip as a touch tool throughout the experiments.

Baseline system. We compare the performance of S-UbiTap with one of the state-of-the-art methods, UbiK [13]. It is worth noting that we implemented a classification-based localization algorithm in the same way as mentioned in the previous work. In particular, to collect a set of training data, we requested users to touch all possible touch locations five times before they used the system.

Metrics. We use two metrics: the localization error and the touch accuracy.

- *Localization error.* We define the localization error as the distance between the estimated touch location and the ground truth.
- *Touch accuracy.* The baseline work was designed under the assumption that the set of possible touch locations \mathcal{P} is pre-determined (e.g., a chess board). For a fair comparison, we measure the touch accuracy of each

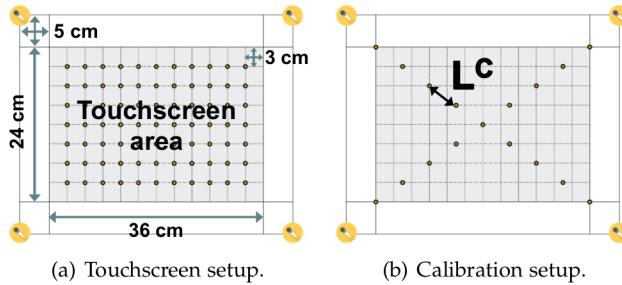


Fig. 10: Default setup for our micro-benchmark tests.

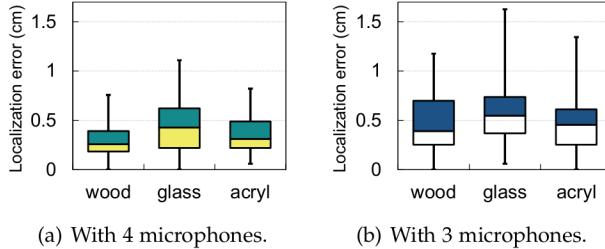


Fig. 11: Localization error of S-UbiTap on various surfaces. The whiskers indicate the 2nd and the 98th percentiles of localization errors, respectively.

system as the probability that touch inputs are correctly classified as their expected touch locations within the pre-defined set.

8.2 Accuracy test

We first evaluate the maximum localization accuracy, which S-UbiTap can provide on common surfaces, such as a wooden table, a glass mirror, and an acrylic board. Therefore, to observe the maximum performance, we asked the user to conduct intensive calibration ($L^C = 1 \text{ cm}$).

Overall accuracy. Figure 11(a) indicates that in all environments, S-UbiTap satisfies the sub-centimeter accuracy requirement. For example, the 98th percentiles for localization errors are 0.76 cm (on the wooden table), 1.11 cm (on the glass mirror), and 0.82 cm (on the acrylic board). These results stem from the fact that S-UbiTap accurately captures the dispersive characteristics of touchesounds for each surface. Such a high degree of accuracy thus enables S-UbiTap to support a very wide range of real-world applications, including keyboards and board games, which require a fine-grained input system, in various environments. S-UbiTap also achieves a similar degree of accuracy with previous classification-based works. Note that on the glass mirror, the performance of S-UbiTap slightly decreases because touchesounds propagate faster on glass surfaces than they do on other surfaces. The high propagation speed incurs that reflections arrive with a small time difference compared to direct sounds. Errors thus increase when estimating the arrival times of direct sounds due to the interference from reflections.

Impact of the number of microphones. As noted in Section 4, S-UbiTap requires at least three microphones to identify the location of touch inputs. To verify how well S-UbiTap

TABLE 1: Localization error of S-UbiTap with various touch tools (unit = centimeter).

Touch tools	A nail tip	A stylus pen	A pen
Average	0.43	0.76	0.92
Stdev.	0.27	0.37	1.66

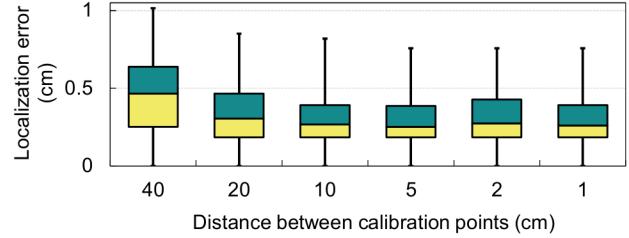


Fig. 12: Impact of the number of calibration points.

works with the least number of microphones, we conducted additional experiments by using three smartphones placed at the left-bottom, right-bottom, and left-top corners of a screen. Figure 11(a) and (b) demonstrate that on every surface, using a smaller number of microphones leads to a slight increase in localization errors. However, despite the increased errors, S-UbiTap still provides a reasonable level of performance with three microphones. For example, its 98th percentile error remains lower than 1.6 cm. Hence, we believe that S-UbiTap can support various touch-based applications, e.g., board games, even with a small number of microphones. In addition, with the availability of more sensors, we can enhance the user experience further with more accurate touch localization.

Impact of touch tools. Different touch tools might generate different touchesounds and affect the performance of S-UbiTap. Table 1 summarizes how accurately S-UbiTap localizes touch inputs when users utilize various types of touch tools, such as nail tips, stylus pens, and pens. S-UbiTap always provides high accuracy regardless of the touch tool. This is because it relies on the dispersion phenomena, which depends on the characteristics of surfaces, not touch tools.

8.3 Usability test

We examine how much effort is required for calibration, i.e., the impact of the number of calibration points on the performance of S-UbiTap. Toward this, a user conducted experiments several times, varying the number of calibration points. It should be noted that the number of tapping times for each calibration point is also an important aspect when estimating the calibration effort. We observed that S-UbiTap can achieve a high degree of localization accuracy even with a single tap, i.e., with low effort, if there is no human error when collecting calibration data.

Figure 12 illustrates the localization accuracy of S-UbiTap over different intervals between calibration points. Even with a small number of calibration points (e.g., $L^C = 40 \text{ cm}$), it shows a high level of localization accuracy (an error of 1.02 cm at the 98th percentile). The error decreases by 0.26 cm as L^C is decreased to 5 cm as S-UbiTap can compute the environment-dependent parameters I and U more accurately

TABLE 2: Localization error of S-UbiTap under different touchscreen displacements (unit = centimeter).

Displacement	0	1.5	3	6	9	12	15
Average	0.31	0.31	0.31	0.36	0.46	0.38	0.40
Stdev.	0.20	0.17	0.19	0.19	0.23	0.22	0.23

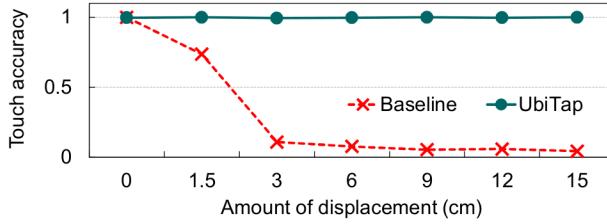


Fig. 13: Robustness comparison against the displacement of a touchscreen.

by collecting calibration data more densely. A further decrease in L^C can improve the localization accuracy, but only slightly. Thus, S-UbiTap only requires users to make a small number of calibration inputs ($L^C = 5 \text{ cm}$) to enable touch functionalities on large screens (e.g., less than 20 inputs for a 17-inch screen). That is, S-UbiTap achieves high accuracy and usability, whereas prior works suffer from the trade-off between them. The baseline, for example, requires at least 36×24 calibration points to provide sub-centimeter accuracy on the 17-inch screen.

8.4 Robustness test

In this experiment, we observe how robustly S-UbiTap localizes touch inputs against dynamic environmental changes. To do this, after calibration (with L^C of 5 cm), we made changes in the environment by 1) re-positioning the touchscreen, 2) putting objects on the surface, or 3) making ambient noise.

Against changes in the position of a touchscreen. In this experiment, the touchscreen, including the portable projector and the smartphones, was moved horizontally after calibration. As presented in Table 2, S-UbiTap achieves stable and high localization accuracy regardless of the position of the screen. This is because the calibrated parameters I and U , on which S-UbiTap relies, remain constant over the entire surface. For example, even in the severe case of 15-cm displacement, it shows a high degree of accuracy (localization errors of 0.36 cm and 0.88 cm on average and at the 98th percentile, respectively). By contrast, the classification-based work, i.e., the baseline, experiences significant performance degradation with a change in the screen's location due to the use of location-dependent features for classification (see Figure 13).

Against changes of surrounding objects. We evaluate the effects of nearby objects by stacking various numbers of books close to the touchscreen (e.g., 5 cm away) after calibration. As presented in Figure 14, the touch accuracy of the baseline decreases as more books are placed on the surface. This occurs because multipath reflection patterns change more when the number of placed objects increases. S-UbiTap, by contrast, shows high robustness against such

TABLE 3: Localization error of S-UbiTap with surrounding object placement changes (unit = centimeter). Each book weighs 2.5 kg.

# of books	0	1	2	3	4	5
Average	0.32	0.32	0.31	0.32	0.33	0.32
Stdev.	0.19	0.19	0.19	0.18	0.21	0.20

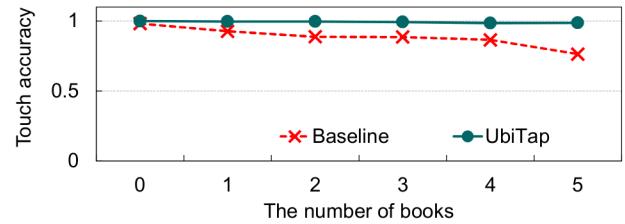


Fig. 14: Robustness comparison against surrounding object placement changes.

reflection pattern changes (i.e., almost 100% touch accuracy in all cases) because it localizes touch events based on an analysis of directly arriving touchsounds. More specifically, it consistently provides an average error of less than 0.33 cm even when numerous objects are newly placed (see Table 3).

Against ambient noise. Acoustic-based localization algorithms can be inherently vulnerable to noise. To observe how noise affects the performance of S-UbiTap, we reproduced noise recorded in the real world (e.g., a cafe), while changing the distance of the speaker from the microphones. Specifically, we set the sound pressure level of the reproduced noise to 60 - 70 dBA, similar to the loudness of a normal conversation. Figure 15 shows that when the location of the noise source is far (e.g., more than 1 m) from the microphones, S-UbiTap exhibits stable and high localization accuracy. With the noise produced close to the microphones, i.e., at a distance of 50 cm, the error increases to 1.73 cm at the 98th percentile, but S-UbiTap still provides reasonable performance (e.g., 95% touch inputs are localized with errors of less than 1.02 cm). This is because touchsounds are typically produced very close to microphones, and arrive at the microphones with a much higher amplitude level than noise. These structural advantages help to achieve a high degree of localization accuracy even in the presence of noise.

8.5 Responsiveness test

With our implementation, we examine how fast S-UbiTap provide feedback to a user's inputs. Hence, we simply measured the computation time required to localize each touch input (1,000 times in total) on a Google Pixel smartphone (with a processor speed of 1.6 GHz). S-UbiTap shows a running time of only 33.4 ms on average with a standard deviation of 5.5 ms, which is much lower than the minimum responsiveness requirement for user-interactive applications (e.g., a latency of 100 ms) [21]. This implies that users can make use of such interactive applications on top of S-UbiTap without any noticeable latency.

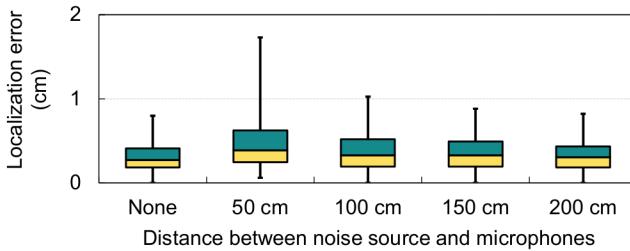
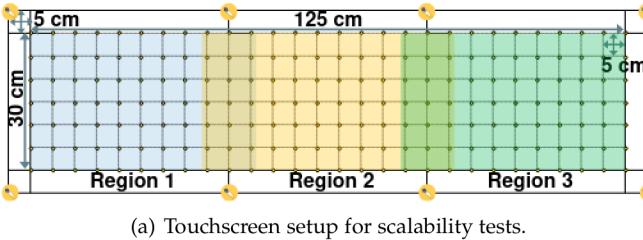
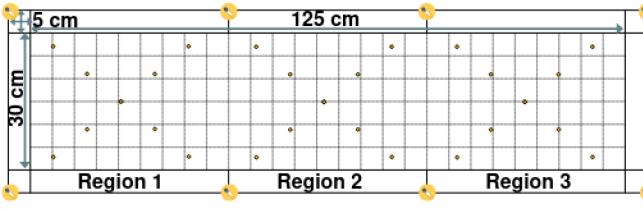


Fig. 15: Noise robustness of S-UbiTap. The sound pressure level of the reproduced noise, 10 cm away from its source, is in the range of 60 and 70 dBA.



(a) Touchscreen setup for scalability tests.



(b) Calibration setup for scalability tests.

Fig. 16: Setup for scalability tests. Note that a large-sized touch screen is divided into smaller overlapped regions.

8.6 Scalability Test

Figure 16 shows the configuration of a large-sized touchscreen used to evaluate the scalability of S-UbiTap. We placed eight smartphones, each of which is equipped with microphones, on a glass plate and divided the surface into three sub-regions. During this experiment, we asked a single user to collect a set of calibration data for each region by tapping each calibration point (see Figure 16(b)). After that, the user made touch inputs by tapping the displayed circles as depicted in Figure 16(a).

Active region selection performance. High accuracy of region selection is essential to achieve accurate localization on large surfaces. Selection of an active region can be done using various approaches apart from leveraging the arrival time of low-frequency components. For example, the energy of touches perceived by each microphone is directly proportional to the distance between touch locations and microphones. In addition, the Time of Arrival(ToA) of touches on each microphone also depends on the propagation distance. In this experiment, we compare the performance of our proposed method with energy-based and ToA-based approaches each of which works as follows:

- *Energy-based active region selection.* It selects the region where the sum of squared amplitudes obtained from its corresponding microphones is higher than other regions.

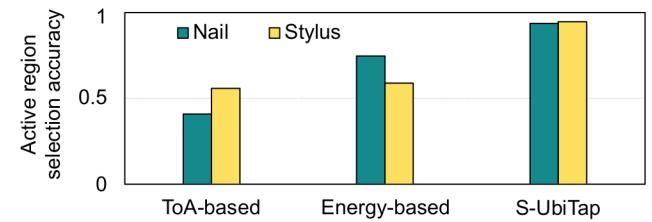


Fig. 17: Active region selection accuracy over different methods.

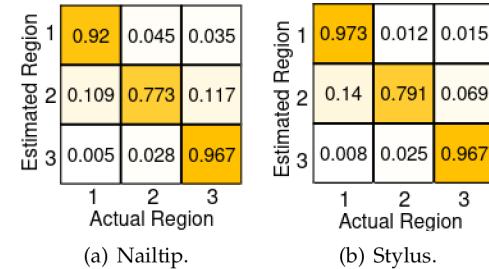


Fig. 18: Confusion matrix for active region selection accuracy of S-UbiTap.

- *ToA-based active region selection.* For each microphone, it measures an arrival time of a touchsound through the energy criterion algorithm [28]. It then decides an active region by comparing their ToA values.

As illustrated in Figure 17, ToA-based and energy-based methods suffer from lower accuracy (48.30 % and 66.74 % on average for each method, respectively). Different microphones have non-identical timing and sensitivity characteristics. This results in non-negligible errors in estimating ToA and energy and eventually in selecting active regions. However, S-UbiTap leverages TDoA between low-frequency components which can be measured without requiring any synchronization between microphones. This allows S-UbiTap to achieve a high degree of accuracy in the active region selection (higher than 90.0% regardless of touch tools). Furthermore, Figure 18 shows that most errors occur between neighboring regions. In particular, more than 80% of such estimation errors happen when touch inputs are made near the common edge between two regions.

Localization accuracy. We then evaluate the effect of our active region selection method on localizing touch inputs. In particular, we compare the following three methods.

- *Without Active Region Selection (ARS).* It conducts localization by using the distance information obtained from all the microphones, i.e., without selecting active regions.
- *With an ARS.* It first selects an active region for each touch input by leveraging the TDoA between low-frequency components. After that, it uses the microphones on the region to identify the location of the input.
- *With an ideal ARS.* It assumes that the region where a touch input is made is already known. Based on this information, it localizes the input by using only the microphones placed on the region.

As shown in Figure 19, using all the microphones, i.e.,

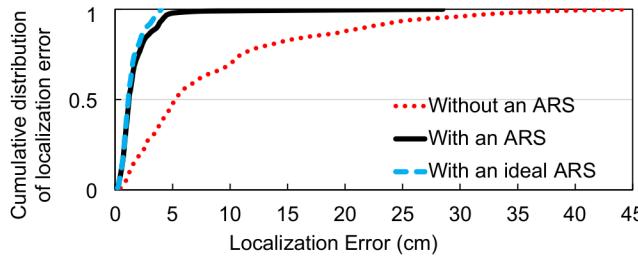


Fig. 19: Effect of active region selection (ARS) in localizing touch inputs on large surfaces.

TABLE 4: Localization error of S-UbiTap using the different number of microphones on a large surface (unit = centimeter).

# of mics	3	4	6	8
Average	11.08	10.57	1.94	1.90
Stdev.	22.21	21.38	3.19	2.69

without an ARS, results in numerous errors in localizing touch inputs on large surfaces (a median error of 5.3 cm). When microphones are placed far from touch locations, touchesounds with very low amplitudes are captured and the error in estimating arrival times significantly increases. However, our proposed ARS helps localizes touch inputs much more accurately using only microphones placed close to the inputs. Furthermore, although the method has some errors in selecting regions as shown in Figure 17, S-UbiTap achieves an accuracy quite similar to that of the ideal implementation (a median error of 1.2 cm). As described above, most errors occur in neighboring areas between two regions and touchesounds generated in these areas are captured with a sufficient level of intensity by the microphones placed on both regions. Therefore, S-UbiTap can properly handle these errors, which occur during the active region selection, by just extending the estimation range of each region as described in Section 6.3.

Impact of the number of microphones on large surfaces. We verify the number of microphones required to support accurate touch localization on large surfaces by conducting an additional experiment. Specifically, in this experiment, we covered four cases, each of which uses a different number of microphones. For example, in the case of using three and four microphones, we only used microphones placed at the corners of Region 2, as depicted in Figure 16(a). In particular, in the three-microphone case, we measured touch localization accuracy using the left-top, right-top, and left-bottom corner microphones of Region 2. We then used additional two microphones placed at the left-top corner of Region 1 and the right-top corner of Region 3 in the six-microphone case.

Table 4 demonstrates that using a small number (e.g., three and four) of microphones on large surfaces lead to a drastic increase in localization errors. To accurately identify touch locations, S-UbiTap requires the use of at least three microphones that are placed close to the locations. However, in the three- and four-microphone cases, only one or two microphones are available for localizing the inputs made on Region 1 and 3, causing the severe performance drops. On

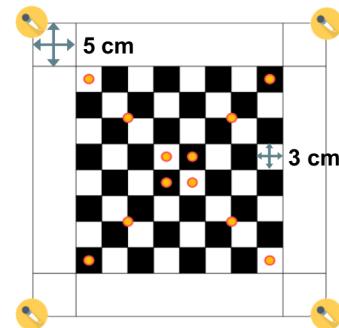


Fig. 20: User study setup. The orange circles on the touch-screen indicate calibration points for S-UbiTap.

the other hand, if we can put three or more microphones on each sub-region, S-UbiTap can precisely select an active region for each touch input and localize the inputs using the microphones for the selected region. For example, as shown in Figure 4, in the six- and eight-microphone cases, S-UbiTap provides a few-centimeter accuracy even on the large surface. Hence, we confirmed that S-UbiTap can support accurate touch input systems on a large surface only if at least three microphones are available for each sub-region of the surface.

8.7 User Study

We conducted a user study to evaluate the effects of S-UbiTap on real-world users, especially in terms of usability.

User study design. Ten college students (four females / six males) participated and played a touch-based game called *KingChaser*. The basic rule of KingChaser is simply to find and touch the position of a randomly placed king piece on an 8x8 chess board. The users used a portable projector and four smartphones to build a touch screen on a wooden table (see Figure 20). The size of each chess square was 3 cm, similar to the size of a typical chess board.

To encompass various scenarios, we conducted experiments in two different environments: single-user and multi-user. In the former case, each user conducted calibration by means of 12 pre-defined locations, as indicated in Figure 20. The users then played KingChaser 200 times with their own calibration results. In the latter case, we assumed that two different users (a target user and an instructor) utilize the touchscreen together. In other words, the instructor first completed the calibration, and then the participant played the game 200 times with the instructor's calibration results. For each experiment, we also compared S-UbiTap with the baseline implementation. Specifically, users were asked to collect a set of training data by tapping each cell on a chess board (64 cells in total) before using the baseline. Note that, during this experiment, all users used their own fingernail tip as a touch tool.

User study results. In this experiment, we observed the two key benefits of S-UbiTap to enable on-surface touch inputs.

First, S-UbiTap is very easy to use and does not sacrifice accuracy. Figure 21 compares the time required for calibration. The baseline system must go through all available touch locations before use, increasing the calibration time substantially. For example, every user took more than one minute for calibration. In the worst case (U9), the time increases to 161

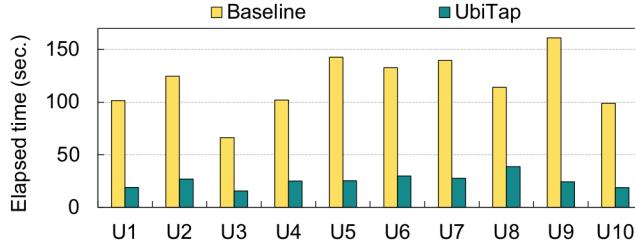


Fig. 21: Elapsed time to conduct calibration.

TABLE 5: Touch accuracy in a single-user scenario.

	Baseline	UbiTap
Average	98.5%	98.5%
Standard deviation	1.4%	1.7%

seconds. S-UbiTap, by contrast, shows a 4.8-fold decrease in the calibration time on average. Furthermore, it does not compromise accuracy as indicated in Table 5. Such a large difference in usability caused most participants to have a more positive experience with S-UbiTap compared to the baseline.

Second, S-UbiTap can support a wider variety and more user-friendly scenarios. Figure 22 shows the capability of both systems to support multi-user environments. The performance of the baseline varies depending on the similarity between users who share a touchscreen. It is only effective when the users generate touchsounds that are very similar to those generated by others (e.g., U2, U6, and U8 – U10). If not, the touch accuracy is degraded to 71.1% (U4). However, the localization algorithm of S-UbiTap provides more stable and higher performance in all cases (e.g., an average touch accuracy of 97.9% with a standard deviation of 2.1%) because when localizing touch inputs, S-UbiTap leverages the arrival times of touchsounds, which is primarily affected by the properties of surfaces. Thus, with an increase in the environmental robustness, S-UbiTap enhances the user experience and even enables new types of applications, such as multiplayer games.

9 DISCUSSION

Supports for various interaction methods. S-UbiTap mainly focuses on enabling single tap inputs, which allows users to interact with computing devices easily and efficiently. Such usability and efficiency can be enhanced by supporting a wider variety of interaction modalities. For example, with multi-touch and swipe inputs, users may easily zoom in and out of screens, as they do on smartphones.

Adaptive parameter configuration. We set most of the parameters of S-UbiTap (e.g., frequency ranges) in a static manner, based on our empirical observations. Although we demonstrate that, with this static decision, S-UbiTap can perform well in diverse environments, there is still a room for further improvement. For example, given that touchsounds show different frequency characteristics depending on the properties of surfaces, we can increase localization accuracy by dynamically analyzing calibration inputs and determining

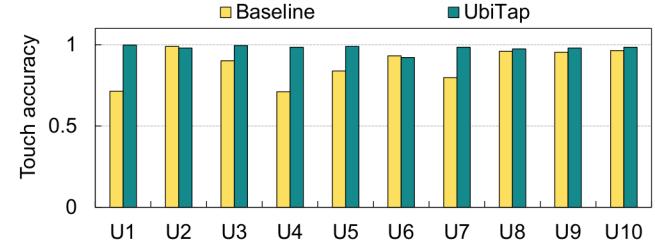


Fig. 22: Touch accuracy in a multi-user scenario.

the optimal range of low- and high-frequencies for each surface.

Environmental constraints. Through our experiments, we show that S-UbiTap is a very powerful and useful touch input method, which can be used in numerous real-world environments (e.g., various surfaces and dynamically-changing / noisy / multi-user scenarios). However, it still shows non-negligible localization errors in the following situations:

- *Surfaces with irregular/directional properties.* Our localization algorithm is designed based on the principle that the propagation speed of surface-borne sound is constant across the entire surface. However, if a surface is curved, each area can have a different curvature, making the propagation speed different even on the same surface. In addition, on surfaces with directional properties, the propagation speed varies depending on the propagation direction.
- *Near the edges of a surface.* When S-UbiTap is installed near the edges of a surface, it experiences a drop in accuracy because the reflections from the edges arrive at nearly the same time as the directly-arriving sound. Similarly, as the size of a surface decreases, the distance between the edges of the surface and microphones can decrease, causing errors to increase because of early reflections.
- *Large obstacles.* If there are large objects such as beams or arms in the middle of the direct propagation path of touchsounds, S-UbiTap will experience a decrease in localization accuracy because the touchsounds are scattered or absorbed by the obstacles.
- *Close proximity between microphones.* The geometry of the microphones can affect the performance of S-UbiTap. To verify the impact of the microphone geometry, we conducted simple experiments. Our experimental results show that, as the relative distance between microphones decreases, the localization error grows. For example, when we set the distance between two horizontally-placed phones (e.g., the top-left and top-right ones) as 10 cm, the 98th percentile of localization errors grows to 1.98 cm on a wooden table. This occurs because the geometric dilution of precision (GDOP) value [29] becomes too high.
- *Touch tool variations.* In our experiments, we show that if users keep using similar touch tools, e.g., fingernail tips, S-UbiTap can provide high accuracy even in multi-user scenarios. However, when using touch interfaces, users may change touch tools (e.g., from fingernail tips to pens).

These dynamic changes can affect the performance of S-UbiTap because a surface's deformation patterns, i.e., the structure of touchsounds, can vary depending on the touch tools. Hence, arrival time estimation errors (U) change unpredictably, leading to a decrease in localization accuracy.

To prevent such undesirable environments, we can provide users with guidelines. For example, during the calibration phase, we can determine whether the TDoA between frequency components, which is measured with calibration inputs, increases proportionally to the propagation distance of touchsounds as shown in Figure 8. If not, we can ask users to build the touch input system on other surfaces. We can also help users to avoid high GDOP problems by automatically measuring the relative position of microphones with existing phone-to-phone localization techniques [23], [24]. In addition, by comparing the frequency signatures (i.e., spectrum) of the test and calibration data, we can detect dynamic changes in touch tools and provide feedback to users.

10 RELATED WORKS

Infrastructure-based techniques. Several works to support on-surface touch interfaces have been presented with promising results. They precisely localize touch inputs made on surfaces by using IR sensors [2], [3], [4], [5], [6], [7], [8], capacitive touch panels [9], [10] [30], visible light sensors [11], or wearable devices [12], [31], [32]. However, requiring such dedicated hardware has limited their portability. In other words, none of them are yet ready for the ubiquitous use.

Vision-based techniques. Commodity cameras have been leveraged to enable on-surface touch interactivity without a loss of portability. SymmetriSense [33] finds the location of near-surface fingertips based on the principle of reflection symmetry. CamK [34] identifies which content is touched, by comparing the locations of fingertips with the location of the contents in captured images. Both techniques, however, are only feasible in limited environments. For example, SymmetriSense can operate only on glossy surfaces on which reflections are produced, and CamK assumes that the touch input space has a static layout, such as a keyboard. S-UbiTap, on the other hand, can be used in more diverse environments, e.g., on common flat surfaces and without requiring a certain input layout.

Acoustic-based techniques. Similar to S-UbiTap, EarSense [35] has leveraged acoustic dispersion and commodity earphones, but for classification of teeth gestures. Other works have also utilized built-in microphones of commodity devices to localize tap on finger knuckles [36] and to enable motion tracking [37]. However, unlike the above studies, to enable acoustic-based on-surface touch interfaces, it is required to use touchsounds generated on the surface of daily object's(mirrors, table surfaces etc.). Therefore, it is necessary to design touch localization techniques based on a deep understanding of the touchsounds' generation and propagation.

Another class of works have attempted to support ubiquitous on-surface touch inputs by listening to touchsounds. Classification-based techniques [13], [14], [38], [39] show a high degree of touch localization accuracy with

a heavy training intensity. They thus hamper usability and are vulnerable to environmental changes. Toffee [15] calculates touch locations based on the TDoA between microphones, especially with much fewer calibration requirements. However, its accuracy has fallen by several tens of cm because it does not properly address the dispersive characteristics of touchsounds. Other works have tried to overcome this limitation, but with the use of an array of surface-mounted geophones [16], [40], accelerometers [17], or microphones [18], [19], all of which require a specific type of infrastructure which are uncommon in commodity computing devices. Thus, the existing works are still limited in terms of accuracy, usability, or portability, making it difficult to use them in practice.

11 CONCLUSION

In this paper, we presented the design and implementation of S-UbiTap, an accurate, usable, robust, scalable, responsive, and ubiquitous touch interface for solid surfaces. Specifically, we harnessed the fundamental characteristics of the dispersion phenomenon. We then used our observations to design touch localization algorithms including simple calibration, arrival time measurement, active region selection, and touch position estimation techniques. Our evaluation using a prototype of S-UbiTap demonstrated that it can support sub-centimeter localization accuracy on many environments (e.g., different surfaces, dynamically-changing environments, and large surfaces), without compromising usability and responsiveness. Our experience with real-world users was also very positive, showing considerable improvements in usability and robustness compared to existing works.

The primary future directions lie in 1) improving the accuracy of localizing on-surface touch inputs through the collaborative use of different types of sensors and 2) supporting more diverse on-surface interactions, especially multi-touch inputs, by using blind source separation techniques [41]. We believe that extending the capability of S-UbiTap can further promote the emergence of new and interesting applications and even new interaction methods in ubiquitous computing environments.

ACKNOWLEDGMENTS

This work was supported in part by the Chung-Ang University Graduate Research Scholarship in 2021 and by the National Research Foundation of Korea (NRF-2022R1C1C1012664 and NRF-2020R1A2C2005479).

REFERENCES

- [1] H. Kim, A. Byanjankar, Y. Liu, Y. Shu, and I. Shin, "Ubitap: Leveraging acoustic dispersion for ubiquitous touch interface on solid surfaces," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 211–223.
- [2] Y. Takeoka, T. Miyaki, and J. Rekimoto, "Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, 2010.
- [3] SONY, "Xperia™ touch," retrieved February 28, 2018 from <https://www.sonymobile.com/global-en/products/smart-products/xperia-touch/#gref>.
- [4] S. Technologies, "Smart board," retrieved February 28, 2018 from <https://education.smarttech.com/>.

- [5] F. Echtler, A. Dippon, M. Tönnis, and G. Klinker, "Inverted ftir: easy multitouch sensing for flatscreens," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, 2009.
- [6] L. Motion, "Leapmotion," retrieved Febuary 21, 2018 from <https://www.leapmotion.com/>.
- [7] A. D. Wilson, "Playanywhere: a compact interactive tabletop projection-vision system," in *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 2005.
- [8] R. Xiao, C. Harrison, and S. E. Hudson, "Worldkit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013.
- [9] J. Rekimoto, "Smartskin: an infrastructure for freehand manipulation on interactive surfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2002.
- [10] M. Sato, I. Poupyrev, and C. Harrison, "Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012.
- [11] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending mobile interaction through near-field visible light sensing," in *Proceedings of the Annual International Conference on Mobile Computing and Networking*, 2015.
- [12] C. Harrison, H. Benko, and A. D. Wilson, "Omnitouch: wearable multitouch interaction everywhere," in *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 2011.
- [13] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization," in *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services*, 2014.
- [14] J. Liu, Y. Chen, M. Gruteser, and Y. Wang, "Vibsense: Sensing touches on ubiquitous surfaces through vibration," in *Proceedings of the Annual IEEE International Conference on Sensing, Communication, and Networking*, 2017.
- [15] R. Xiao, G. Lew, J. Marsanico, D. Hariharan, S. Hudson, and C. Harrison, "Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation," in *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices & Services*, 2014.
- [16] S. Pan, C. G. Ramirez, M. Mirshekari, J. Fagert, A. J. Chung, C. C. Hu, J. P. Shen, H. Y. Noh, and P. Zhang, "Surfacevibe: vibration-based tap & swipe tracking on ubiquitous surfaces," in *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2018.
- [17] A. Sulaiman, K. Poletkin, and A. W. Khong, "Source localization in the presence of dispersion for next generation touch interface," in *Cyberworlds (CW), 2010 International Conference on*. IEEE, 2010, pp. 82–86.
- [18] D. T. Pham, Z. Ji, M. Yang, Z. Wang, and M. Al-Kutubi, "A novel human-computer interface based on passive acoustic localisation," in *Proceedings of the International Conference on Human-Computer Interaction*, 2007.
- [19] J. A. Paradiso, C. K. Leo, N. Checka, and K. Hsiao, "Passive acoustic knock tracking for interactive windows," in *Proceedings of the SIGCHI Extended Abstracts on Human Factors in Computing Systems*, 2002.
- [20] A. Ross and G. Ostiguy, "Propagation of the initial transient noise from an impacted plate," *Journal of Sound and Vibration*, vol. 301, no. 1-2, pp. 28–42, 2007.
- [21] K. Lee, D. Chu, E. Cuervo, J. Kopf, Y. Degtyarev, S. Grizan, A. Wolman, and J. Flinn, "Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming," in *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services*, 2015.
- [22] S. NEWSROOM, "Share the fun with samsung galaxy beam," retrieved Febuary 10, 2018 from <https://news.samsung.com/global/share-the-fun-with-samsung-galaxy-beam>.
- [23] J. Qiu, D. Chu, X. Meng, and T. Moscibroda, "On the feasibility of real-time phone-to-phone 3d localization," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2011.
- [24] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, "Swordfight: Enabling a new class of phone-to-phone action games on commodity phones," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, 2012.
- [25] T. Page, "Usability of text input interfaces in smartphones," *Journal of Design Research*, vol. 11, no. 1, pp. 39–56, 2013.
- [26] J. O. Smith, *Spectral Audio Signal Processing*. W3K publishing, 2011.
- [27] S. Butterworth, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.
- [28] S. M. Markalous, S. Tenbohlen, and K. Feser, "Detection and location of partial discharges in power transformers using acoustic and electromagnetic signals," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 15, no. 6, pp. 1576–1583, 2008.
- [29] R. B. Langley *et al.*, "Dilution of precision," *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.
- [30] S. Mayer, X. Xu, and C. Harrison, "Super-resolution capacitive touchscreens," 2021.
- [31] C. Lian, X. Ren, Y. Zhao, X. Zhang, R. Chen, S. Wang, X. Sha, and W. J. Li, "Towards a virtual keyboard scheme based on wearing one motion sensor ring on each hand," *IEEE Sensors Journal*, 2020.
- [32] J. Gong, A. Gupta, and H. Benko, "Acustico: Surface tap detection and localization using wrist-based acoustic tdoa sensing," in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 2020, pp. 406–419.
- [33] C. Yoo, I. Hwang, E. Rozner, Y. Gu, and R. F. Dickerson, "Symmetrisense: Enabling near-surface interactivity on glossy surfaces using a single commodity smartphone," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2016.
- [34] Y. Yin, Q. Li, L. Xie, S. Yi, E. Novak, and S. Lu, "Camk: Camera-based keystroke detection and localization for small mobile devices," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2018.
- [35] J. Prakash, Z. Yang, Y.-L. Wei, H. Hassanieh, and R. R. Choudhury, "Earsense: earphones as a teeth activity sensor," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–13.
- [36] W. Chen, L. Chen, Y. Huang, X. Zhang, L. Wang, R. Ruby, and K. Wu, "Taprint: Secure text input for commodity smart wristbands," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.
- [37] G. Cao, K. Yuan, J. Xiong, P. Yang, Y. Yan, H. Zhou, and X.-Y. Li, "Earphonetrack: involving earphones into the ecosystem of acoustic motion tracking," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 95–108.
- [38] H. Kim, B. Joe, and Y. Liu, "Tapsnoop: Leveraging tap sounds to infer tapstrokes on touchscreen devices," *IEEE Access*, vol. 8, pp. 14 737–14 748, 2020.
- [39] Z. Xiao, T. Chen, Y. Liu, and Z. Li, "Mobile phones know your keystrokes through the sounds from finger's tapping on the screen," in *40th IEEE International Conference on Distributed Computing Systems*. IEEE, 2020.
- [40] M. Mirshekari, S. Pan, J. Fagert, E. M. Schooler, P. Zhang, and H. Y. Noh, "Occupant localization using footstep-induced structural vibration," *Mechanical Systems and Signal Processing*, vol. 112, pp. 77–97, 2018.
- [41] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 2009–2025, 1998.

Anish Byanjankar received the B.Eng. degree in Computer Engineering from Kathmandu University, Dhulikhel, Nepal in 2015, and the M.S. degree in Computer Science from KAIST, Daejeon, Korea in 2019.





Yunxin Liu is a Guoqiang Professor at Institute for AI Industry Research (AIR), Tsinghua University. He received his B.S., M.S., and Ph.D. degrees from University of Science and Technology of China, Tsinghua University, and Shanghai Jiao Tong University, respectively. Prior to joining Tsinghua in 2021, he was a Principal Researcher and the Research Manager of Heterogeneous and Extreme Computing (HEX) group at Microsoft Research Asia. His research interests are mobile and edge computing.



Insik Shin (M'11) is a professor in the School of Computing at KAIST, Korea. He received a Ph.D. degree from the University of Pennsylvania, USA, an MS degree from Stanford University, USA, and a BS degree from Korea University, Korea, all in Computer (& Information) Science. Prior to joining KAIST in 2008, he has been a post-doctoral research fellow at Malardalen University, Sweden, and a visiting scholar at University of Illinois at Urbana-Champaign, USA. His research interests include real-time embedded systems, systems security, mobile computing, and cyber-physical systems. He serves on program committees of top international conferences, including RTSS, RTAS and ECRTS. He is a recipient of several best (student) paper awards, including MobiCom 2019, RTSS 2003, RTAS 2012, and RTSS 2012, KAIST Excellence Award, KAIST Technology Innovation Awards, and Naver Young Faculty Award.



Myeongwon Choi received his B.S. degree in the School of Computer Science and Engineering from Chung-Ang University in 2021. He is currently a graduate student at Chung-Ang University, School of Computer Science and Engineering, Seoul, South Korea. His research interests include human-computer interaction, cyber-physical systems, mobile and ubiquitous systems, and privacy.



Yuanchao Shu (M'15 – SM'19) is currently a Principal Researcher with Microsoft. His research interests lie broadly in mobile, sensing and networked systems, with a major focus on topics including edge video analytics and location-based systems. His previous research results have led to over 40 publications at top-tier peer-reviewed conferences and journals including MobiCom, MobiSys, SenSys, USENIX Security, NSDI, JSAC, TMC, TPDS etc. Dr. Shu currently serves on the editorial board of IEEE Transactions of Wireless Communications, ACM Transactions on Sensor Networks, and was a member of the organizing committee and TPC of conferences including MobiCom, SenSys, SEC, Globecom, ICC etc. He won five Best Paper/Demo (Runner-Up) Awards, and was the recipient of ACM China Doctoral Dissertation Award and IBM PhD Fellowship. Dr. Shu received his Ph.D. from Zhejiang University, and was also a joint Ph.D. student in the EECS Department at the University of Michigan, Ann Arbor.



Hyosu Kim received his B.S. degree in Department of Computer Engineering from Sungkyunkwan University in 2010. He then received his M.S. and Ph.D. degrees in School of Computing from Korean Advanced Institute of Science and Technology (KAIST), Republic of Korea in 2012 and 2018, respectively. He is currently an assistant professor at Chung-Ang University, School of Computer Science and Engineering, Seoul, South Korea. His research interests include IoT, cyber-physical systems, mobile and ubiquitous systems, and privacy.