UbiTap: Leveraging Acoustic Dispersion for Ubiquitous Touch Interface on Solid Surfaces

Hyosu Kim
School of Computing
KAIST
hyosu.kim@kaist.ac.kr

Anish Byanjankar
School of Computing
KAIST
anish@kaist.ac.kr

Yunxin Liu
Microsoft Research
yunxin.liu@microsoft.com

Yuanchao Shu
Microsoft Research
yuanchao.shu@microsoft.com

Insik Shin*
School of Computing
KAIST
insik.shin@kaist.ac.kr

ABSTRACT

With the omnipresence of computing devices in our daily lives, interests in ubiquitous computing interfaces have grown. In response to this, various studies have introduced on-surface input techniques which use the surfaces of surrounding objects as a touch interface. However, these methods are yet struggling to support ubiquitous interaction due to their dependency on specific hardware or environments. In this paper, we propose UbiTap, an input method that turns solid surfaces into a touch input space, through the use of sound (i.e., with microphones already present in the commodity devices). More specifically, we develop a novel touch localization technique which leverages the physical phenomenon, referred to as *dispersion*, a characteristic of sound as it travels through solid surfaces, so as to address challenges which limit existing acoustic-based solutions in terms of portability, accuracy, usability, robustness, and responsiveness. Our extensive experiments with a prototype of UbiTap show that we can support sub-centimeter accuracy on various surfaces with minor user calibration effort. In our experience with real-world users, UbiTap significantly improves usability and robustness, thus enabling the emergence of more exciting applications.

CCS CONCEPTS

• Human-centered computing → Ubiquitous and mobile computing systems and tools; **• Hardware → Sound-based input / output**;

KEYWORDS

Ubiquitous Computing; Human-Computer Interaction; On-Surface Touch Interface; Sound-based Localization

*A corresponding author.

SenSys '18, November 4–7, 2018, Shenzhen, China

ACM ISBN 978-1-4503-5952-8/18/11…$15.00
https://doi.org/10.1145/3274783.3274848

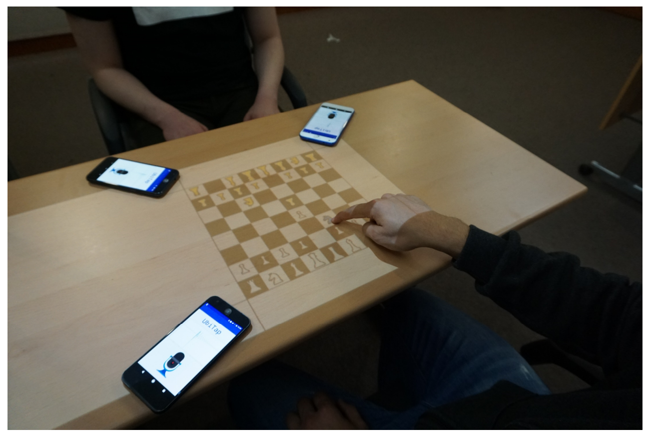


Figure 1: UbiTap enables touch inputs on a screen which is projected onto a wooden table, with the use of commodity hardware.

ACM Reference Format:
Hyosu Kim, Anish Byanjankar, Yunxin Liu, Yuanchao Shu, and Insik Shin. 2018. UbiTap: Leveraging Acoustic Dispersion for Ubiquitous Touch Interface on Solid Surfaces . In *The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18), November 4–7, 2018, Shenzhen, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3274783.3274848

1 INTRODUCTION

A plethora of computing devices, such as IoT devices, smartphones, and smart mirrors, have been used everywhere and around the clock. One of the new trends that support seamless interaction with these ubiquitous devices is the use of surfaces on objects (e.g., tables) as touch interfaces. Many works have enabled such on-the-fly interfaces by introducing techniques that track a user's fingertip on a surface. However, they either benefit from a specific infrastructure [2, 3, 8, 15, 17, 19, 21, 22, 24, 25, 29], or show a limited application range [27, 28]. Thus, none of them can yet provide the seamless interactivity in a ubiquitous manner.

Through this paper, we introduce UbiTap, a novel approach to support accurate on-surface touch inputs without a loss of portability in a different yet effective manner; by listening to sound. Figure 1 illustrates an instance of the use of UbiTap. First, it records a *touchsound*, an impact sound produced when tapping surfaces, with multiple (at least three) microphones. Note that we can easily build the multi-microphone system with microphones built in different devices. UbiTap then analyzes the propagation pattern of the received signal and identifies the location of the touch events.

However, exploiting touchsounds makes it much challenging to support in-situ touch interfaces. This is primarily because users can utilize touch input methods in a wide range of environments (e.g., in-door/outdoor, day/night, and alone/with friends), resulting in unpredictable and dynamic changes in characteristics of the touchsounds. This challenge has forced existing approaches into a trade-off situation with regard to usability, accuracy, robustness, and portability. Classification-based works [6, 23] achieve high accuracy while requiring a heavy training burden to users for every setup. Even worse, environmental changes cause unexpected variations in their features and degrade accuracy. Another direction of research seeks to reduce user efforts, by using Time-Difference-of-Arrival (TDoA) between microphones [11–13, 20, 26]. However, some of these studies [26] have shown poor accuracy due to complex acoustic phenomena, e.g., dispersion on solid surfaces. Others attempt to minimize errors but require specific types of hardware, such as a time-synchronized vibration sensor array [11–13, 20].

UbiTap addresses the limitations of the existing works through an in-depth exploration of acoustic dispersion phenomena. As sound travels along surfaces, its frequency components are transmitted at different speeds depending on their carrying frequency. Especially, 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 feasible to estimate the distance from a touch point to a microphone by leveraging the TDoA between the frequency components of the touchsound.

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

- We develop a simple yet robust calibration method which estimates surface-dependent parameters (e.g., propagation speeds). In particular, based on the principle which holds that propagation speeds are constant across the entire surface, we compute the parameters with a small number of calibration points. Furthermore, changes in the surroundings rarely affect the propagation speeds, thus allowing precise touch localization without additional calibrations for previously calibrated surfaces.
- We introduce a new arrival time detection technique which accurately pinpoints the frequency-specific arrival times of touchsounds with low computation latency. It applies different time-frequency analysis schemes depending on each frequency wave's dispersion properties.
- We design an algorithm which determines touch positions. For each microphone, it individually measures the propagation distance of touchsounds by using the information obtained through our calibration and arrival time detection techniques. It then localizes touch inputs by combining these distance results. This individual calculation enables freely to use the built-in microphones of different devices without time synchronization between them.

We evaluate the performance of UbiTap with 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 UbiTap can easily achieve subcentimeter 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, UbiTap maintains accuracy of less than a centimeter even in the presence of dynamic environmental changes (e.g., displacement of nearby objects). Our experiment with real-world users is positive, especially in terms of usability. The user study results also show that 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 for enabling ubiquitous touch interfaces on solid surfaces.
- We design UbiTap, a novel dispersion-aware framework for on-surface touch localization, which satisfies all the following requirements: a high degree of portability, accuracy, usability, robustness, and responsiveness.
- We implement a prototype of UbiTap and demonstrate its effectiveness through extensive benchmark tests and real-world user studies.

2 REQUIREMENTS AND CHALLENGES

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

Ad-hoc touchscreen construction. Users can use a touchscreen 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. It is too expensive to provide touch interfaces with state-of-the-art techniques on such large screens. For example, a 15-inch capacitive touch panel costs about US $100, and scaling this up to larger screens can cost even more. In contrast, given that UbiTap uses the built-in microphones of these devices, there is no additional cost to support such interfaces.

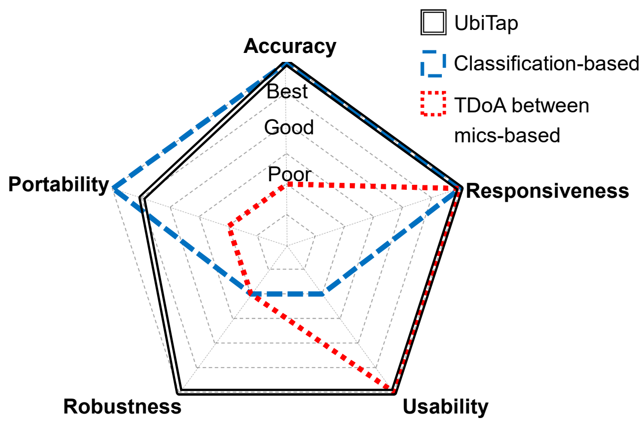


Figure 2: Comparison of acoustic-based touch localization methods.

2.1 Requirements

From the above applications, we can identify the five major requirements of UbiTap as follows:

- *Portability.* To support the in-situ construction of touch input systems, UbiTap must not rely on the help of a dedicated infrastructure.
- *Accuracy.* The gap between each touch point could be a few centimeters in keyboard applications or chess games. A user's touch inputs should be localized with sub-centimeter accuracy.
- *Usability.* UbiTap needs to be easy to set up and use and should not require much time and effort by users.
- *Robustness.* Users should be able to use UbiTap in various situations. It should work accurately regardless of the environments.
- *Responsiveness.* Users should acquire feedback for their touch inputs without any noticeable latency. For example, user-interactive applications must respond to user inputs within 100 ms [5].

2.2 Challenges

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

- *Different surfaces.* Users may use different object surfaces such as wooden tables, mirrors, or plastic boards. Touchsounds on different surfaces have unique signatures and travel at different speeds, due to differences in the physical properties of different surfaces.
- *Relocation of touch input space.* A touch input space can be rearranged to another area on the same surface, producing a different touchsound 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 applications, users can place or move objects (e.g., bags or books) on surfaces. The 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 UbiTap.

- *Classification-based: Low usability and robustness.* One class of prior works [6, 23] collects a set of heavy training data to characterize touchsounds for a given environment. However, this sacrifices usability for accuracy. Moreover, the signatures of touchsounds, the classification features of these methods, change unpredictably depending on the environment, causing a significant performance drop.
- *TDoA between microphones-based: Low accuracy and portability.* Other works have utilized TDoA between microphones to reduce calibration efforts [11–13, 20, 26]. However, for a precise TDoA measurement, they require dedicated hardware which contains multiple microphones [12, 13, 26], geophones [11], or accelerometers [20], each of which is perfectly time-synchronized. Moreover, some [26] suffer from low accuracy, as surface-borne sound undergoes dispersion, causing variations in the TDoA depending on the frequencies.

In contrast to these studies, UbiTap can support a high level of performance in all categories (see Figure 2). The key idea of UbiTap offering this level of quality is that it leverages the dispersion properties of touchsounds. In the remainder of this paper, we first observe the properties of dispersion that provide grounds for developing UbiTap (Section 3). We then describe how such properties are incorporated into the design of UbiTap (Section 4). Note that UbiTap uses at least three microphones, causing a subtle decrease 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.

3 ACOUSTIC DISPERSION

In this section, we deeply explore the core characteristics of surface-borne touchsounds, i.e., *acoustic dispersion.*

When tapping on a surface, a touchsound occurs and spreads through the surface in a transverse manner, causing the air pressure around its passage to change. Acoustic dispersion occurs as the touchsound moves along the surface [16]. A solid-like surface is a dispersive medium and transmits waves of different frequencies at different speeds. Therefore, different frequency components of the touchsound pass the area near microphones at different times. The touchsound is also transferred from the touch location to the microphones through non-dispersive mediums, such as air. However, since surface-borne sound travels much faster than air-borne one, we can easily distinguish between them and capture the dispersive behaviors in audio recordings. For example, as indicated in Figure 3,

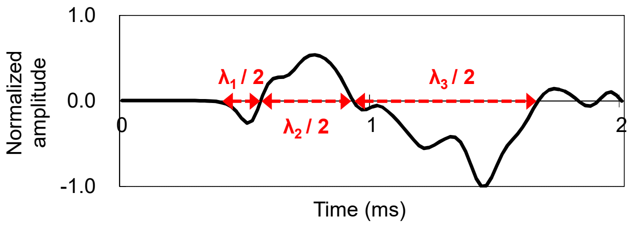
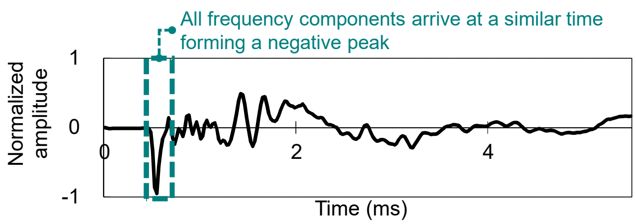
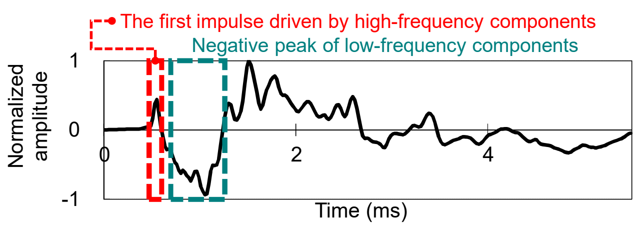


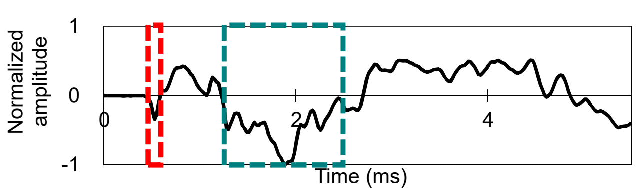
Figure 3: Early part of a touchsound: higher frequency waves arrive earlier than lower ones (λ_i indicates the i-th approximate local wavelength of the touchsound). Note that, in our preliminary observations, we captured touchsounds on a wooden table by using a fingernail tip.



(a) Distance = 1 cm



(b) Distance = 20 cm



(c) Distance = 50 cm

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

the local wavelength of the touchsound increases over time. This occurs because, on solid surfaces, the propagation speed of a lower frequency wave is slower than that of a higher frequency wave.

Key principle 1. *Different frequency components of a touchsound propagate at different speeds $V(f)$, where f is their carrying frequency.*

The propagation speed $V(f)$ is determined by the physical properties of surfaces (e.g., thickness, density, Young's modulus, and Poisson ratio). For example, touchsounds travel faster on steel surfaces than on wooden surfaces due to stiffness differences. On common flat surfaces such as office desks and mirrors, $V(f)$ is almost constant over the entire area because each part of the surface has similar physical properties. Note that, on surfaces with irregular or directional properties, $V(f)$ may vary from region to region. This is discussed in Section 9. Furthermore, $V(f)$ is not affected by changes in the ambient environment (e.g., ambient noise and nearby objects).

Key principle 2. *For a certain surface, $V(f)$ is constant regardless of the touch location and the surrounding environment.*

Figure 4 shows how the dispersion phenomenon affects the arrival times 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 (see 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. On the other hand, with a longer propagation distance, other initial peaks are observed before the negative peak, as shown in Figures 4(b) and (c). This is because the high-frequency components (e.g., above 18 kHz) of the touchsounds arrive much earlier than the low-frequency components (e.g., under 1 kHz). In particular, the figures show that the time difference between the first impulse and the negative peak grows gradually as their propagation distance increases.

It is important to note that key principle 2 (i.e., the consistency of $V(f)$) yields the following principle:

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.

4 UBITAP SYSTEM DESIGN

The main design goal of UbiTap is to support accurate, usable, robust, responsive, and portable touch input interfaces. To achieve this, we leverage the dispersive principles of touchsounds in designing UbiTap. In particular, we develop three key techniques: 1) arrival time detection, 2) simple calibration, and 3) touch position estimation (see Figure 5). Once a touch input is made, UbiTap initially estimates when each frequency component of the touchsound arrives at the microphones. It then uses the TDoA between the frequency components to determine surface-dependent parameters (in a calibration phase) or the location of the touch input (in a 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, this is quite complicated because different frequency waves often interfere with each other. For example, adjacent frequency components can arrive

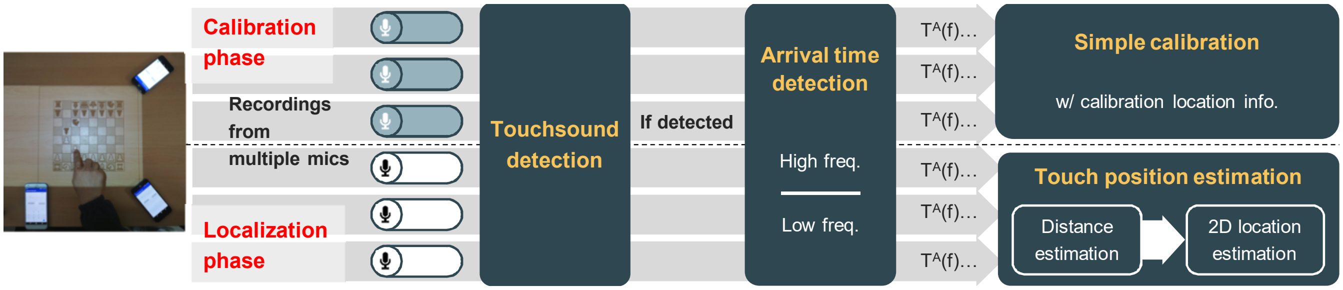


Figure 5: Architecture of UbiTap.

at similar times and cause interference. In addition, reflections of waves at a higher frequency can disrupt those at a lower frequency that arrive directly because their propagation speeds are different. This necessitates the use of high-resolution time-frequency analysis methods, but their high computational complexity can compromise the responsiveness of the system.

UbiTap addresses these challenges and meets the five fundamental requirements for ubiquitous touch interfaces as follow:

- *Accuracy and responsiveness improvement.* We develop an accurate, yet fast method for detecting 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.* For each microphone, we individually estimate the propagation distance of touchsounds with its own sensing data. We then combine each microphone's distance results to localize touch events in 2D space. This per-microphone measurement enables us collaboratively to use the microphones of different devices 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 both 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 robustness of propagation speeds against changes in the surrounding environments.

Usage conditions. UbiTap assumes the following four usage conditions: 1) touchsounds have an energy level sufficient to analyze, 2) users have at least three microphones to use a distance-based localization approach, 3) there is an on-surface screen, and 4) the screen's size and position relative to the microphones are known in advance.

First, users can make noticeable touchsounds by tapping surfaces with hard objects such as fingernail tips or pens. We can also easily meet the second condition with the collective use of multiple commodity devices. It is common to see a single user carry multiple mobile devices such as smartphones and tablets, which equipped with microphones. In addition, the microphone array can be easily constructed using the devices of different users in the case of multi-user environments. With regard to the third condition, we can install an instant screen using a portable projector (e.g., Samsung Beam [9]). We can also simply print out the layout of the input interfaces, e.g., keyboard, on a paper if the layout is static. The last condition can be satisfied in various ways. For example, we can ask users to place their microphones at pre-defined locations such as each corner of a screen. Furthermore, to minimize human errors when positioning the microphones, we can use existing phone-to-phone localization techniques [14, 30].

5 ARRIVAL TIME MEASUREMENT

In this section, we describe how 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

To extract touchsounds from audio recordings precisely, we can simply use a sound level threshold, because touchsounds have high amplitudes as depicted in Figure 4. However, in the presence of bursty noise (e.g., human voices), this simple method may entail an increase in false positive results. UbiTap addresses this with the help of motion sensors, e.g., gyroscopes, as proposed in UbiK [23]. This previous work showed that motion readings vary by touch events that cause surface vibration, but not by acoustic noise that propagates through the air. Thus, the sensor fusion technique can detect touchsounds more robustly against noise[1].

UbiTap first examines the energy level of audio and motion signals to determine when a touch occurs. It computes $E^S(t)$, the accumulated sound energy level at time t, as,

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

where $X^S(t)$ is the received audio signal at time t, and T^W, the window size, is empirically set to 1 ms. UbiTap then compares $E^S(t)$ with the threshold ϵ^S, which is configured as a quarter of the maximum accumulated energy measured during the calibration step. 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 1) both $E^S(t)$ and

[1] UbiK has achieved nearly 100% detection accuracy in noisy environments, such as in food courts [23]. In this paper, we do not conduct further experiments for touchsound detection.

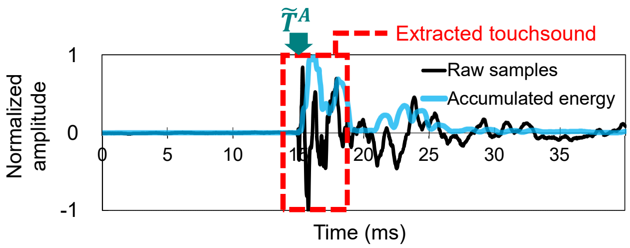


Figure 6: Typical structure of a touchsound; the highest energy level is observed in an early part of the touchsound due to directly-arriving components.

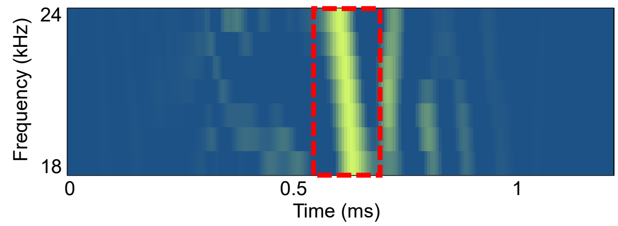


Figure 7: Extracted WVD features of a touchsound at high frequencies; the directly-arriving sound shows high magnitude due to its shortest propagation length. Note that the brighter the WVD feature is, the louder the sound is at the corresponding time and frequency.

$E^M(t)$ exceed ϵ^S and ϵ^M, respectively, and 2) 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 [10].

Touchsound segmentation. Once a touch event is detected, UbiTap extracts the early part of its touchsound, which will be used to measure arrival times. Here, we assume that a touch event is detected at time t. UbiTap identifies the approximate starting point of its touchsound, $\widetilde{T}^A$, in time window $[t - T^G, t + T^G]$. We set a guard interval T^G to 20 ms, which is a typical duration of touchsounds. First, it filters out low-frequency (< 5 kHz) components based on the following two principles: 1) waves at a higher frequency arrive earlier than those at a lower frequency and 2) ambient noise (e.g., human voices) has a high level of energy at the low frequencies. With the filtered samples, UbiTap computes the accumulated energy level for each time instant and finds E^*, the maximum level among them. It finally determines $\widetilde{T}^A$ as the first time instant at which the accumulated energy is larger than $E^*/20$ and then takes the original (i.e., non-filtered) sound between $[\widetilde{T}^A - T^W, \widetilde{T}^A - T^W + T^G/4]$ (see Figure 6).

5.2 Pinpointing Arrival Times

As discussed in Section 4, there is a trade-off between accuracy and responsiveness when measuring the arrival times of touchsounds. For example, using high-resolution time-frequency analysis methods such as the Wigner-Ville distribution (WVD) helps to estimate the arrival times more precisely. However, in terms of responsiveness, these high-resolution methods inherently lead to a significant performance drop, due to their high computational complexity (e.g., $O(k^2 \log k)$, where k is the sample length). Instead, we can apply computationally-efficient techniques, including short-term Fourier and continuous wavelet transforms, but these techniques compromise accuracy because of their low resolution support. To address this trade-off, UbiTap further explores the frequency-dependent characteristics of touchsounds and leverages the results of such observations in designing our arrival time measurement technique.

Arrival time estimation of high-frequency waves. Arrivals of high-frequency (e.g., > 18 kHz) waves are observed in the very early stage of received touchsounds (see Figure 4). This implies that we can estimate the arrival times of touchsounds at high frequencies, by analyzing only a small number of samples (e.g., 1 ms). Hence, as the search range is sufficiently small, we can use high-resolution analysis methods such as WVD while not compromising latency much.

Capturing this, UbiTap takes the first 256 samples (i.e., approximately 1.3 ms with a sampling rate of 192 kHz) from a touchsound, which is obtained in our detection step. It then applies a Hanning window [18] to assign more weights to signals near $\widetilde{T}^A$, and extracts WVD features from the windowed samples. Let $WVD(t, f)$ denote the magnitude of the extracted WVD feature at time t and at frequency f. For each frequency f_i^H in $\mathcal{F}^H$, a set of high frequencies, UbiTap estimates its arrival time $T^A(f_i^H)$ as the first time instant, which 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, depending on the distance between the touch location and the microphone, the arrival time difference between high- and low-frequency waves varies drastically. For example, the time gap changes from 1 ms to 2 ms as the distance increases from 20 cm to 50 cm (see Figures 4(b) and (c)). It can increase even further with a longer propagation distance. Thus, we should search the arrival times of low-frequency components in a wide range of samples, making it difficult to apply computationally-heavy methods. Instead, UbiTap designs a simple technique based on the characteristics of touchsounds at low frequencies. As presented in Figure 4, low-frequency waves arrive in the form of a negative peak and show the highest energy level. In other words, by finding the minimum peak, we can simply but accurately estimate when low-frequency waves arrive.

UbiTap constructs a Butterworth filter bank [1] to separate input signals into 20 components, each of which carries a different frequency. 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 the filter bank design, 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,

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

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 which 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 $\Delta\widehat{T^A}(f^H, f^L)$ denote the TDoA between high-frequency f^H and low-frequency f^L waves, measured by our arrival time estimation technique. $\Delta\widehat{T^A}(f^H, f^L)$ 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 UbiTap identifies the arrival time of each frequency component as the time instant which 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 $\Delta\widehat{T^A}(f^H, f^L)$ and D can be represented as follows:

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

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 discuss the impact of touch tool variations in Section 9.

Capturing this relationship, UbiTap computes the environment-dependent parameters (in a calibration phase) or the propagation distance of touchsounds (in a localization phase). Then, it localizes touch inputs with the distance information obtained from multiple microphones.

6.1 Simple Calibration

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

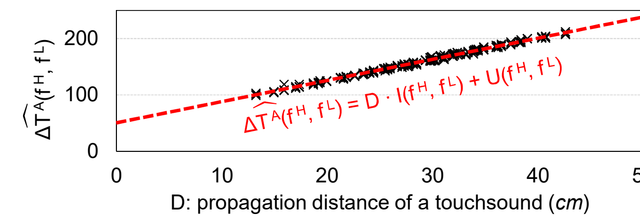


Figure 8: Relationship between the propagation distance of touchsounds and the TDoA between high-frequency (f^H = 18 kHz) and low-frequency (f^L = 1 kHz) waves.

every pair of the calibration inputs,

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

where $D^C_{i,j}$ is the distance between the i-th microphone and the j-th calibration point and $\Delta\widehat{T^A_{i,j}}(f^H, f^L)$ 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. UbiTap finally calculates the estimation error $U_i(f^H, f^L)$ as follows:

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

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

6.2 Touch Position Estimation

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

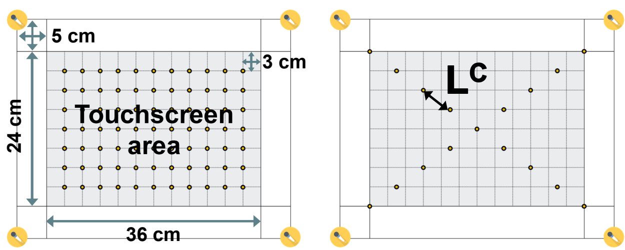
$$\widehat{D_i} = \frac{1}{N^H \cdot N^L} \sum_{j=1}^{N^H} \sum_{k=1}^{N^L} \frac{\Delta\widehat{T^A_i}(f_j^H, f_k^L) - U_i(f_j^H, f_k^L)}{I_i(f_j^H, f_k^L)}, \quad (8)$$

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

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 $\mathcal{P}$, 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, UbiTap declares the touch location as the point P_i which shows the least squared error among all possible positions in $\mathcal{P}$.



(a) Touchscreen setup. (b) Calibration setup.

Figure 9: Default setup for our micro-benchmark tests.

7 IMPLEMENTATION

We implemented a prototype of UbiTap on commodity smartphones (e.g., Pixel and Pixel XL) running Android 8.0.0.

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 about 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 pre-processing 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 too high amplitude. To avoid audio clipping problems, 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 existing mobile devices which makes it difficult to configure a microphone array on a single device. 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 touchsounds well, whereas others cannot, which can reduce localization accuracy. To address these limitations, the prototype of UbiTap supports multi-device environments that use the built-in microphones of many mobile devices together.

8 EVALUATION

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

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

8.1 Micro-Benchmark Tests

We evaluate how well UbiTap addresses each challenge which hinders the realization of an accurate, usable, robust, and responsive touch localization.

8.1.1 *Evaluation Setup and Methodology*

We conducted experiments with an on-surface touchscreen measuring 36 cm by 24 cm (i.e., a 17-inch screen) (see Figure 9(a)). 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 type. To observe the effects of different surfaces, we also used other surfaces, including a glass mirror and an acrylic board, which have great potential for smart devices such as smart mirrors.

During the experiments, we first asked a single user to tap all calibration points 5 times before she used the touchscreen. Figure 9(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 $N^C = ((40\ cm/L^C) + 1) \cdot 2$. 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 9(a). In particular, all circles were tapped sequentially, each repeated 10 times. The user utilized her fingernail tip as a touch tool through entire experiments.

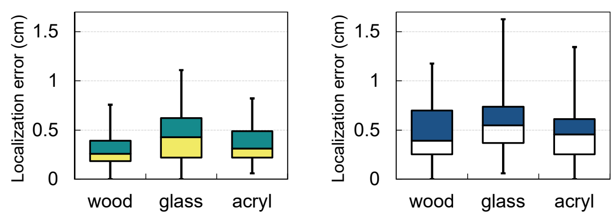
Baseline system. We mainly compare the performance of UbiTap with one of the state-of-the-art works, UbiK [23]. It is worth noting that we implemented a classification-based localization algorithm in the same way as mentioned in the previous work. Especially, to collect a set of training data, we requested users to touch all possible touch locations 5 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 a set of possible touch locations $\mathcal{P}$ is predetermined (e.g., a chess board). For a fair comparison, we measure the touch accuracy of each system as the probability that touch inputs are correctly classified as their expected touch locations within the pre-defined set.

8.1.2 *Accuracy test*

First, we evaluate the maximum localization accuracy, which UbiTap can provide on common surfaces such as a wooden table, a glass mirror, and an acrylic board. Therefore, to observe the



(a) With 4 microphones. (b) With 3 microphnoes.

Figure 10: Localization accuracy of UbiTap on various surfaces. The whiskers indicate the 2nd and the 98th percentiles of localization errors, respectively.

maximum performance, we asked the user to conduct intensive calibration ($L^C = 1\ cm$).

Overall accuracy. Figure 10(a) indicates that, in all environments, 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 UbiTap accurately captures the dispersive characteristics of touchsounds for each surface. Such a high degree of accuracy thus enables 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. UbiTap also achieves a similar degree of accuracy with previous classification-based works (these comparison results are not shown due to page limitations). Note that, on the glass mirror, the performance of UbiTap slightly decreases because touchsounds 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, UbiTap requires at least three microphones to identify the location of touch inputs. To verify how well UbiTap 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. Figures 10(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, 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 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.

8.1.3 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 UbiTap. Toward this, a user conducted experiments several times with changing 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 efforts. We observed that UbiTap can achieve a high degree of localization accuracy even with a single tapping time, i.e., with low effort, if there is no human error when collecting calibration data.

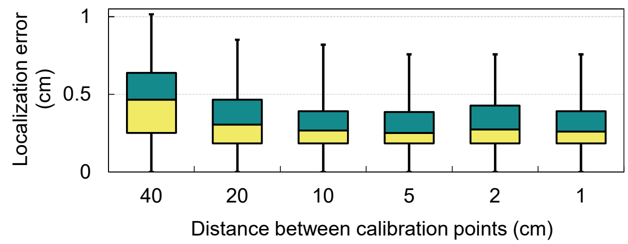


Figure 11: Impact of the number of calibration points.

Table 1: Localization accuracy of 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

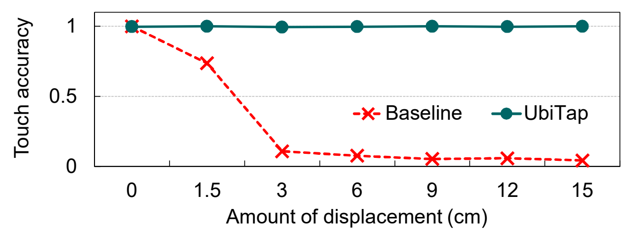


Figure 12: Robustness comparison against the displacement of a touchscreen.

Figure 11 illustrates the localization accuracy of UbiTap over different intervals between calibration points. Even with a small number of calibration points (e.g., $L^C = 40\ cm$), 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 UbiTap can compute the environment-dependent parameters I and U more accurately by collecting calibration data more densely. A further decrease in L^C can improve the localization accuracy, but only slightly. Thus, UbiTap only requires users to make a small number of calibration inputs ($L^C = 5\ cm$) to enable touch features on large screens (e.g., less than 20 inputs for a 17-inch screen). That is, UbiTap achieves a high degree of both accuracy and usability, whereas prior works suffers from a 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.

8.1.4 Robustness test

In this experiment, we observe how robustly UbiTap localizes touch inputs against dynamic environmental changes. To do this, after

Table 2: Localization accuracy of 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

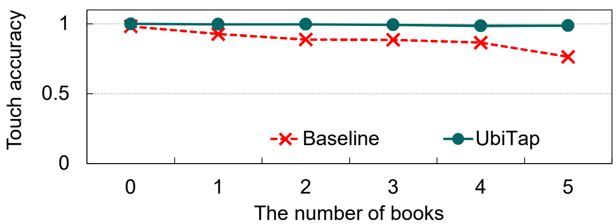


Figure 13: Robustness comparison against surrounding object placement changes.

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 shown in Table 1, UbiTap achieves stable and high localization accuracy regardless of the position of the screen, as the calibrated parameters I and U, on which 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). However, the classification-based work, i.e., the baseline, experiences significant performance degradation with a change of the screen's location due to the use of location-dependent features for classification (see Figure 12).

Against changes of surrounding objects. We evaluate the effects of nearby objects by stacking different numbers of books close to the touchscreen (e.g., 5 cm away) after calibration. As presented in Figure 13, 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. UbiTap, on the other hand, shows high robustness against such reflection pattern changes (e.g., 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 2).

Against ambient noise. Acoustic-based localization algorithms can be inherently vulnerable to noise. To observe how noise affects the performance of 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

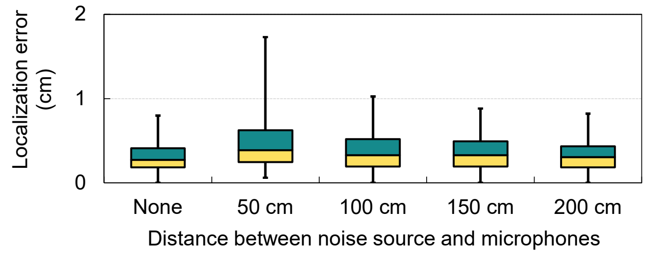


Figure 14: Noise robustness of UbiTap. The sound pressure level of the reproduced noise, 10 cm away from its source, is in the range of 60 and 70 dBA.

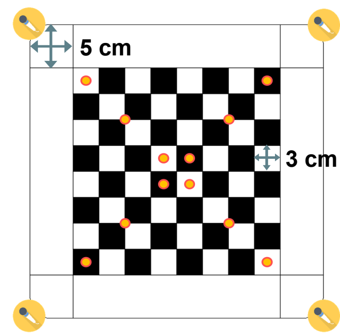


Figure 15: User study setup. The orange circles on the touchscreen indicate calibration points for UbiTap.

normal conversation. Figure 14 shows that, when the location of the noise source is far (e.g., more than 1 m) from the microphones, UbiTap exhibits stable and high localization accuracy. With 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 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 them with a much higher amplitude level than noise. Thus, these structural advantages help to provide a high degree of localization accuracy even in the presence of noise.

8.1.5 Responsiveness test

With our implementation, we examine how fast UbiTap provide feedback to a user's inputs. Hence, we simply measured the computation time required to localize each touch input (1,000 in total) on a Google Pixel smartphone (with a processor speed of 1.6 GHz). 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) [5]. This implies that users can make use of such interactive applications on top of UbiTap without any noticeable latency.

8.2 User Study

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

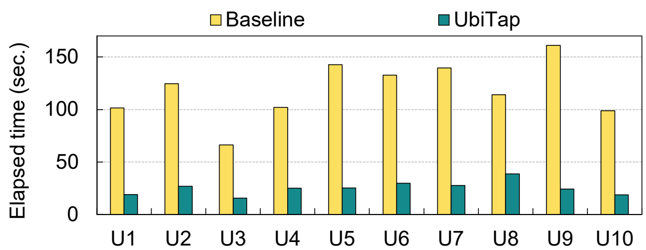


Figure 16: Elapsed time to conduct calibration.

User study design. Ten college students (4 females / 6 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 15). 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 with 12 pre-defined locations, as indicated in Figure 15. 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 did calibration, and then the participant played the game 200 times with the instructor's calibration results. For each experiment, we also compare UbiTap with the baseline implementation. In particular, 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 UbiTap to enable on-surface touch inputs.

First, UbiTap is very easy to use and does not sacrifice accuracy. Figure 16 compares the times required for calibration. The baseline system must go through all available touch locations before use, increasing the calibration time substantially. For example, every user required more than one minute for calibration. In the worst case (U9), the time increases to 161 seconds. UbiTap, on the other hand, shows a 4.8x decrease in the calibration time on average. Furthermore, it does not compromise accuracy as indicated in Table 3. Such a large difference in usability caused most participants to have a more positive feel with UbiTap compared to the baseline.

Second, UbiTap can support a greater variety and more user-friendly scenarios. Figure 17 shows the capability of both systems to support multi-user environments. The performance of the baseline varies depending on the similarity between users who shares a touchscreen. It is only effective when the users generate touchsounds which are very similar to those by others (e.g., U2, U6 and U8 - U10). Otherwise, the touch accuracy is degraded to 71.1% (U4). However, the localization algorithm of 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, UbiTap leverages the arrival times of touchsounds,

Table 3: Touch accuracy in a single-user scenario.

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

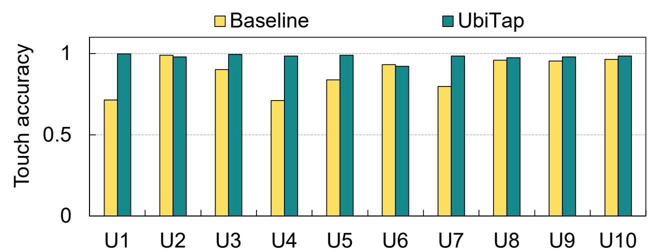


Figure 17: Touch accuracy in a multi-user scenario.

which is primarily affected by the properties of surfaces. Thus, with an increase in the environmental robustness, UbiTap enhances the user experience and even enables new types of applications, such as multiplayer games.

9 DISCUSSION

Supports for various interaction methods. 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 more various interaction modalities. For example, with multi-touch and swipe inputs, users may easily zoom in and out of screens, as they do on smartphones. Thus, our future work includes an investigation to support diverse interaction methods.

Adaptive parameter configuration. We set most of the parameters of UbiTap (e.g., frequency ranges) in a static way, based on our empirical observations. Although we demonstrate that, with this static decision, UbiTap can perform well in diverse environments, there is still a room for further improvement. For example, since touchsounds show different frequency characteristics depending on the properties of surfaces, we can increase localization accuracy by dynamically analyzing calibration inputs and determining the optimal range of low- and high-frequencies for each surface. Therefore, in our future work, we will develop adaptive techniques to optimize the parameters for a given environment.

Environmental constraints. Throughout our experiments, we show that UbiTap is a very powerful and useful touch input method, which can be used in numerous real-word environment (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 show 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 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 due to early reflections.
- *Large obstacles.* If there are large objects such as beams or arms in the middle of the direct propagation path of touchsounds, 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 microphones can affect the performance of 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 [4] becomes high.
- *Touch tool variations.* In our experiments, we show that if users keep using similar touch tools, e.g., fingernail tips, 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 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 unpredictbly, leading to a decrease in localization accuracy.

To prevent such undesirable environments, we can provide users 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 [14, 30]. In addition, by comparing the frequency signatures (e.g., 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 localize touch inputs made on surfaces by using IR sensors [2, 8, 19, 21, 22, 24, 25], capacitive touch panels [15, 17], visible light sensors [29], or wearable devices [3]. 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 [28] finds the location of near-surface fingertips based on the principle of reflection symmetry. CamK [27] 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.

Acoustic-based techniques. Similar to UbiTap, some works have utilized the built-in microphones of commodity devices. Classification-based techniques [6, 23] show a high degree of touch localization accuracy with a heavy training intensity. They thus hamper usability and accuracy is vulnerable to environmental changes. Toffee [26] calculates touch locations based on the TDoA between microphones, especially with much less 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 [7, 11], accelerometers [20], or microphones [12, 13], all of which require a specific type of infrastructure.

11 CONCLUSION

In this paper, we presented the design and implementation of UbiTap, which enables accurate, usable, robust, responsive and ubiquitous touch inputs on solid surfaces, with the use of sound. In particular, we explored the fundamental characteristics of the dispersion phenomenon. We then used our observations to design touch localization algorithms including simple calibration, arrival time measurement, and touch position estimation techniques. Our evaluation with a prototype of UbiTap demonstrated that it can support sub-centimeter localization accuracy on many environments (e.g., different surfaces and dynamically-changing environments), 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. Consequently, we believe that UbiTap can bring up new applications and even new interaction method in ubiquitous computing.

ACKNOWLEDGMENTS

This work was supported in part by Microsoft Research Asia, the Korean Government MSIT (NRF-2018R1A5A1059921), and IITP (2014-0-00065). Also, we would like to thank our anonymous reviewers and shepherd, for their valuable comments.

REFERENCES

[1] Stephen Butterworth. 1930. On the theory of filter amplifiers. *Wireless Engineer* 7, 6 (1930), 536–541.
[2] Florian Echtler, Andreas Dippon, Marcus Tönnis, and Gudrun Klinker. 2009. Inverted FTIR: easy multitouch sensing for flatscreens. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces.*

[3] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*.

[4] Richard B Langley et al. 1999. Dilution of precision. *GPS world* 10, 5 (1999), 52–59.

[5] Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. 2015. 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*.

[6] Jian Liu, Yingying Chen, Marco Gruteser, and Yan Wang. 2017. VibSense: Sensing Touches on Ubiquitous Surfaces through Vibration. In *Proceedings of the Annual IEEE International Conference on Sensing, Communication, and Networking*.

[7] Mostafa Mirshekari, Shijia Pan, Jonathon Fagert, Eve M Schooler, Pei Zhang, and Hae Young Noh. 2018. Occupant localization using footstep-induced structural vibration. *Mechanical Systems and Signal Processing* 112 (2018), 77–97.

[8] Leap Motion. Retrieved Febuary 21, 2018 from https://www.leapmotion.com/.

[9] Samsung NEWSROOM. Retrieved Febuary 10, 2018 from https://news.samsung.com/global/share-the-fun-with-samsung-galaxy-beam.

[10] Tom Page. 2013. Usability of text input interfaces in smartphones. *Journal of Design Research* 11, 1 (2013), 39–56.

[11] Shijia Pan, Ceferino Gabriel Ramirez, Mostafa Mirshekari, Jonathon Fagert, Albert Jin Chung, Chih Chi Hu, John Paul Shen, Hae Young Noh, and Pei Zhang. 2018. Surfacevibe: vibration-based tap & swipe tracking on ubiquitous surfaces. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks*.

[12] Joseph A Paradiso, Che King Leo, Nisha Checka, and Kaijen Hsiao. 2002. Passive acoustic knock tracking for interactive windows. In *Proceedings of the SIGCHI Extended Abstracts on Human Factors in Computing Systems*.

[13] Duc Truong Pham, Ze Ji, Ming Yang, Zuobin Wang, and Mostafa Al-Kutubi. 2007. A novel human-computer interface based on passive acoustic localisation. In *Proceedings of the International Conference on Human-Computer Interaction*.

[14] Jian Qiu, David Chu, Xiangying Meng, and Thomas Moscibroda. 2011. On the feasibility of real-time phone-to-phone 3d localization. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*.

[15] Jun Rekimoto. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[16] Annie Ross and Germain Ostiguy. 2007. Propagation of the initial transient noise from an impacted plate. *Journal of Sound and Vibration* 301, 1-2 (2007), 28–42.

[17] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[18] Julius O. Smith. 2011. *Spectral Audio Signal Processing*. W3K publishing.

[19] SONY. Retrieved Febuary 28, 2018 from https://www.sonymobile.com/global-en/products/smart-products/xperia-touch/#gref.

[20] Amir Sulaiman, Kirill Poletkin, and Andy WH Khong. 2010. Source localization in the presence of dispersion for next generation touch interface. In *Cyberworlds (CW), 2010 International Conference on*. IEEE, 82–86.

[21] Yoshiki Takeoka, Takashi Miyaki, and Jun Rekimoto. 2010. 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*.

[22] SMART Technologies. Retrieved Febuary 28, 2018 from https://education.smarttech.com/.

[23] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. 2014. 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*.

[24] Andrew D Wilson. 2005. PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*.

[25] Robert Xiao, Chris Harrison, and Scott E Hudson. 2013. 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*.

[26] Robert Xiao, Greg Lew, James Marsanico, Divya Hariharan, Scott Hudson, and Chris Harrison. 2014. 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*.

[27] Yafeng Yin, Qun Li, Lei Xie, Shanhe Yi, Edmund Novak, and Sanglu Lu. 2018. CamK: Camera-based Keystroke Detection and Localization for Small Mobile Devices. *IEEE Transactions on Mobile Computing* PP, 99 (2018), 1–1.

[28] Chungkuk Yoo, Inseok Hwang, Eric Rozner, Yu Gu, and Robert F Dickerson. 2016. 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*.

[29] Chi Zhang, Josh Tabor, Jialiang Zhang, and Xinyu Zhang. 2015. Extending mobile interaction through near-field visible light sensing. In *Proceedings of the Annual International Conference on Mobile Computing and Networking*.

[30] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. 2012. 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*.