Vernier: Accurate and Fast Acoustic Motion Tracking Using Mobile Devices

Yunting Zhang, Jiliang Wang, Weiyi Wang, Zhao Wang, Yunhao Liu
School of Software and TNList
Tsinghua University, China
{zhangyt15, wang-wy14, wangc16}@mails.tsinghua.edu.cn, {jiliangwang, yunhao}@tsinghua.edu.cn

Abstract—**Acoustic motion tracking has been viewed as a promising user interaction technique in many scenarios such as Virtual Reality (VR), Smart Appliance, video gaming, etc. Existing acoustic motion tracking approaches, however, suffer from long window of accumulated signal and time-consuming signal processing. Consequently, they are inherently difficult to achieve both high accuracy and low delay. We propose Vernier, an efficient and accurate acoustic tracking method on commodity mobile devices. In the heart of Vernier lies a novel method to efficiently and accurately derive phase change and thus moving distance. Vernier significantly reduces the tracking delay/overhead by removing the complicated frequency analysis and long window of signal accumulation, while keeping a high tracking accuracy. We implement Vernier on Android, and evaluate its performance on COTS mobile devices including Samsung Galaxy S7 and Sony L50t. Evaluation results show that Vernier outperforms previous approaches with a tracking error less than 4 mm. The tracking speed achieves 3× improvement to existing phase based approaches and 10× to Doppler Effect based approaches. Vernier is also validated in applications like controlling and drawing, and we believe it is generally applicable in many real applications.**

I. Introduction

The rapid development and prevalence of mobile devices enable various ubiquitous mobile applications. Acoustic motion tracking using mobile devices has been shown as a promising user interaction technique in many scenarios such as Smart Appliance (e.g., TV control), Virtual Reality (VR), Augmented Reality (AR), video gaming, etc., attracting many attentions and efforts. In acoustic motion tracking, a mobile phone tracks its position using received acoustic signal. For example, with acoustic motion tracking the gesture or posture of a user can be obtained, which can facilitate various applications.

Typically, inertial sensors such as accelerometer, gyroscope can be used for mobile motion tracking [24]. However, the tracking error is high (up to 60 cm even in 6s [24]) and thus accurate tracking is difficult to achieve [13]. Various approaches leverage RF signal for mobile device tracking [20] [16] [9] [23]. Those approaches usually require special hardware support or incur a high computation overhead [18] [4] [2] [1].

Recently, acoustic signal based motion tracking is proposed as a promising technique [15] [19] [11] [3] [12] [10] [5] [7]. Further, CAT [13] proposes a novel distributed Frequency Modulated Continuous Wave (FMCW) based method for mobile motion tracking. Using FMCW, the calculation of moving time is translated to calculation of frequency. CAT improves the accuracy by combining inertial sensors. Recently, LLAP [21] proposes a tracking method based on phase shift of acoustic signal. In LLAP, a mobile phone transmits an acoustic signal, which is reflected by a moving target and received by the mobile phone again. By calculating the phase shift between the original signal and the reflected signal, the signal travelling time and thus moving distance of the target can be obtained.

Existing approaches, however, have some limitations in terms tracking accuracy, overhead and delay. Most approaches require frequency analysis (e.g., FFT) to derive the frequency shift, phase, etc., which inevitably introduces a high computation overhead and delay. Moreover, tracking accuracy is also limited by window length. Achieving high accuracy requires accumulating and processing a sufficient window of signal. Thus it is difficult to achieve both low latency and high accuracy simultaneously. Those limitations hinder performance improvement for acoustic motion tracking and limit their practical application.

A. Our Approach

To address those limitations, we propose *Vernier*, an accurate and fast acoustic motion tracking approach using mobile devices. As in [24] and [13], a mobile device running Vernier receives inaudible acoustic signals, each at a certain frequency, from different signal sources (e.g., speakers on TVs). Instead of calculating the frequency shift directly (e.g., using FFT), Vernier designs a novel method to calculate the phase change due to frequency shift with a small window of signal. Then Vernier calculates the distance change to each source and derives the real-time position of the mobile device.

In the heart of Vernier, we design a novel method to efficiently calculate the phase change based on a very small window of samples (e.g., 100 samples). Our method is inspired by vernier caliper. Signal samples in our method act as the vernier while the local maximums of original signal act as the ruler. For different phase changes (length), the samples on the vernier has different matching positions (local maximum) on the ruler, which can be leveraged to further derive phase change. To further improve the efficiency, we propose a Differentiated window based phase change calculation (DW-PC) in which we calculate the phase change based on local maximum change between two windows. Further, we show

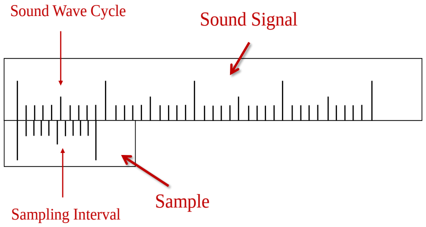


Fig. 1: Principle of our approach.

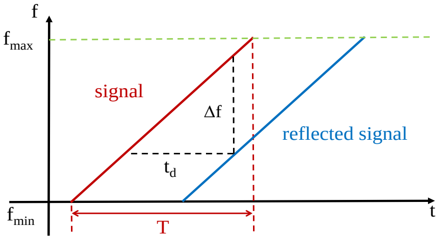


Fig. 2: Calculate the time t based on FMCW.

that our method can achieve a higher accuracy than existing approaches while has a much smaller delay and overhead.

Overall, Vernier aims to achieve the following goals: (1) accurate tracking with mm-level error, (2) a low delay in order to enable real-time applications such as mobile gaming and (3) a low computation overhead efficiently run on commodity mobile devices.

B. Summary of Main Results

We implement Vernier on Android and evaluate it on different mobile devices including SAMSUNG Galaxy S7/Sony L50t. Vernier has no special hardware requirements and can run on most commodity mobile phones. The evaluation results show that Vernier can achieve efficient tracking with a median error less than 4 mm in various scenarios at a distance of 7 m. We believe Vernier is general and can facilitate nowadays user interaction like Video Games, VR, AR, smart home applications, etc.

Our major contributions include:

- We propose the design of Vernier, an accurate and fast motion tracking approach on mobile devices, which leverages a novel method to efficiently and accurately derive phase change and thus moving distance.
- We analyze the performance of Vernier and compared it with existing approaches. The analysis result shows the performance improvement of Vernier.
- We implement Vernier on Android and evaluate it on different mobile devices including SAMSUNG Galaxy S7/Sony L50t. The evaluate results show that Vernier can achieve efficient tracking with a median error less than 4 mm in various scenarios at a distance of 7 m.

The remainder of this paper is organized as follows. Section II analyzes the limitations of existing approaches. Section III presents the main design of our approach. Section IV shows implementation parameters in real applications. Section V shows the evaluation results. Section VI concludes this work.

II. Prior Arts

We briefly introduce the basic mechanisms of existing acoustic motion tracking approaches and their practical limitations.

A. Tracking based on Doppler Effect

Many approaches track mobile device based on Doppler Effect [24] [14] [8] [17] [6]. Suppose a sound source is emitting a signal and a moving receiver receives the signal. Due to Doppler Effect, the receiver's relative speed v to the sound source can be calculated as:

$$v = \frac{F_\triangle}{F_0} c \tag{1}$$

where F_0 is the original frequency of the signal, $F_\triangle$ is the frequency shift due to Doppler Effect, and c is the speed of sound. Therefore, the moving distance for time T can be calculated as $d = \int_0^T v dt$. As a result, given the initial position, the target can then be tracked.

The key step in Doppler Effect based tracking is to calculate the frequency shift (F_c). By applying frequency analysis (e.g., STFT) to the received acoustic signal, the spectrum distribution of the received signal can be obtained. Given the frequency of original signal (e.g., sine wave) [24], the frequency shift F_c is calculated. In practice, the frequency analysis (e.g., STFT) is applied to a moving window. Thus the accuracy of frequency D_F can be calculated as:

$$\hat{F} = \frac{F_s}{L_w} \tag{2}$$

where L_w is the window length and F_s is the sampling rate. Note that padding the signal with zeros cannot improve the frequency resolution [21]. Combing Eq. (1) and (2), we can derive the resolution of moving speed as:

$$\hat{v} = \frac{\hat{F}}{F_0} c = \frac{F_s}{L_w F_0} c. \tag{3}$$

We can see that the accuracy of moving speed (and thus distance) is related to the window size L_w. A larger window can provide better frequency domain resolution and higher moving speed accuracy. On the other hand, a larger window contains more samples and causes a larger delay. For a typical window $L_w = 1764$ samples and a sampling rate $F_s = 44100$ Hz [24] [21], the accuracy of spectrum D_F is $\frac{F_s}{L_w} = \frac{44100}{1764} = 25$ Hz. Suppose the frequency $F_0 = 20000$ Hz and the speed of sound wave $c = 340$ m/s, the moving speed resolution is $\hat{v} = \frac{25 \times 340}{20000} = 0.425$ m/s. This indicates that the accumulated distance error in 1 second can be up to 0.425 m. The corresponding delay using such a window is $1764/44100 = 40$ ms.

Moreover, Doppler shift is subject to high noise. Detecting Doppler shift needs to detect the frequency with the highest energy. However, the frequency with highest energy may be difficult to determine due to noise as shown in [21] .

We can see that approaches using Doppler Effect, which require window-based frequency analysis, introduce inevitable computation overhead. High accuracy and small delay is difficult to achieve simultaneously in practice due to the relation between window size and accuracy.

B. Tracking based on FMCW

A *Frequency Modulated Continuous Wave* (*FMCW*) or *chirp* is a signal with linearly increasing Frequency. An FMCW of length T with frequency ranging from f_{min} to f_{max} can be denoted as

$$R(t) = \cos(2\pi(f_{min} + \frac{B}{2T}t)t). \tag{4}$$

where $B = f_{max} - f_{min}$ is the bandwidth.

Assume a mobile phone needs to measure the length of path an FMCW travels, e.g., the round-trip distance to a reflected object. By using FMCW, the travelling time calculation can be translated to frequency calculation. The mobile phone first transmits an FMCW signal, which is directly received by the mobile phone itself. Meanwhile, the signal travels along the reflected path and is received by the mobile phone again. The received signal can be denoted as $R'(t) = \alpha R(t - t_d)$, where t_d is the time delay for travelling along the path and α is the attenuation. Note CAT [13] removes the requirement of receiving reflected signal and synchronization between receiver and signal source by a distributed FMCW. But the basic idea of distance calculation is similar. As shown in Figure 2, the distance d can be calculated as

$$d = \frac{c \cdot t_d}{2}. \tag{5}$$

The time t_d can be calculated by the frequency difference Δf between two FMCW signals. In practice, we multiply the two signal signals $R(t)$ and $R'(t)$ according to $\cos A \cos B = \frac{1}{2}(\cos(A+B) + \cos(A-B))$. By filtering the high frequency component $\cos(A+B)$, we have:

$$V(t) = \alpha \cos(2\pi(f_{min}t_d + B\frac{(2tt_d - t_d^2)}{2T})). \tag{6}$$

From Eq. (6), we have $\Delta f = \frac{Bt_d}{T}$ where Δf is the frequency of $V(t)$. Thus we have

$$t_d = \frac{\Delta f \cdot T}{B}. \tag{7}$$

According to Eq. (7) and (5), the travelling distance can therefore be calculated as

$$d = \frac{\Delta f \cdot c \cdot T}{B}. \tag{8}$$

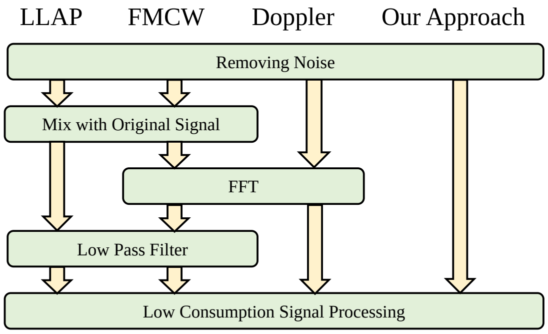


Fig. 3: Working flows of different approaches.

It is also required to derive the frequency of signal $V(t)$ (e.g., using FFT). According to Eq. (2), the resolution of frequency is $\hat{F} = F_s/L_w$. Thus, the accuracy of distance can be calculated as

$$\hat{d} = \frac{F_s \cdot c \cdot T}{L_w \cdot B}. \tag{9}$$

Since $L_w = F_s \cdot T$, we have

$$\hat{d} = \frac{c}{B}. \tag{10}$$

Eq. (10) shows that the accuracy is only related to B. For $B = 10$ kHz [13], which is very large for acoustic signal on mobile, the accuracy is $\hat{d} = 340/10000 = 0.034$ m.

FMCW based approaches require multiplying two signals (to derive Δf), frequency analysis (e.g., FFT) and low pass filtering (to remove the high frequency component).

C. Tracking based on Phase

Recently, LLAP [21] proposes a method for mobile tracking based on low latency acoustic phase [22]. Suppose a sound signal $R(t) = \cos 2\pi ft$ travels through a path p with time-varying path length of $d_p(t)$. According to [21], the received sound signal from path p can therefore be represented as

$$R_p(t) = 2A'_p \cos(2\pi ft - 2\pi fd_p(t)/c) \tag{11}$$

where $2A'_p$ is the amplitude of the received signal, the term $2\pi fd_p(t)/c$ comes from the phase lag caused by the propagation delay of $d_p(t)/c$ and c is the speed of sound. The key idea is to obtain the phase from the received signal $R_p(t)$. Based on the phase, the change of path length $d_p(t)$ can be obtained. By multiplying the received signal with the signal source $\cos 2\pi ft$, we have

$$R(t)R_p(t) = A'_p(\cos(-2\pi f\frac{d_p(t)}{c}) + \cos(4\pi ft - 2\pi f\frac{d_p(t)}{c})). \tag{12}$$

The high frequency component $\cos(4\pi ft - 2\pi fd_p(t)/c)$ can be removed by a low pass filter. Therefore, we can obtain $I_p(t) = A'_p(\cos(-2\pi fd_p(t)/c)$. Similarly, multiplying the received signal $R_p(t)$ with $\sin(2\pi ft)$, we obtain $Q_p(t) = A'_p \sin(-2\pi fd_p(t)/c)$. Then based on $I_p(t)$ and $Q_p(t)$, we can calculate the phase $-2\pi fd_p(t)/c = arctg(Q_p(t)/I_p(t))$. Therefore, the path length change in a short time period can be calculated by the phase change.

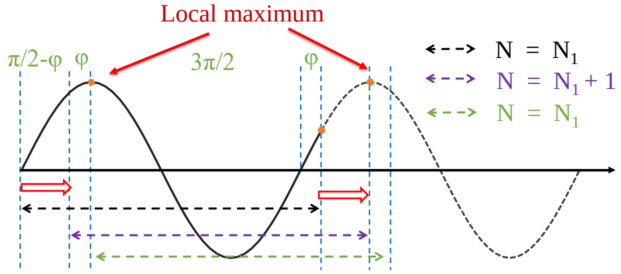


Fig. 4: Local maximum number and phase change.

D. Summary

We summarize the main working flow of different approaches in Figure 3. Both Doppler based approach and FMCW based approach require frequency analysis and filtering, which incur extra overhead on mobile devices. Moreover, the frequency analysis and filtering introduce an inevitable delay, e.g., accumulating a window of samples for processing. They also inherently have a limited resolution in distance measurement. Phase based approach significantly improve the accuracy. It still requires multiplying the received signal with a given signal. It also requires different filters for signal processing, which incurs a relative high computation overhead and a non-negligible delay.

The analysis coincides with the experimental results in those approaches: (1) For Doppler Effect based approach [24], the median error for tracking is around 1.4 cm and quickly increases over time due to error accumulation. The tracking delay is 40 ms. (2) For FMCW based approach [13], the median tracking error is 6 mm by combining inertial sensors. The tracking delay is at least 40 ms due to the length of STFT window. (3) For phase based tracking [21], the 1D tracking accuracy is 3.5 mm the tracking latency is 15 ms. The effective range for tracking is within 40 cm according to their experiments.

III. Vernier Design

The design Vernier has the following goals:

- Accurate. The approach should be accurate with error in mm-level.
- Efficient. It should be efficient and incurs a low overhead. It should be able to run on commodity mobile phones without specific hardware support.
- Low latency. It should be able to calculate the position with a very small delay to satisfy real-time applications such as mobile gaming, VR, etc.

A. 1D Tracking

We first introduce our approach for 1D case. Then we show how to extend it 2D and 3D cases. Considering a static sound source transmits an acoustic signal of frequency F_0 and a moving receiver (e.g., mobile phone) receives the sound signal. For example, the signal source is the TV speaker and the mobile phone is held by a user. The goal for 1D tracking is to derive the mobile phone's moving distance d to the sound source. The distance can be calculated as $d = \int_t v(t)dt$. Denote the sampling rate as F_s and the frequency for the received signal as $F_c = F_\Delta + F_0$. Due to Doppler Effect, for a time period of length T, we have

$$d = \frac{c}{F_0}\int_t F_\Delta dt = \frac{c}{F_0}\int_t (F_c - F_0)dt = \frac{c\tilde{\phi}}{2\pi F_0} - cT \quad (13)$$

where $\tilde{\phi}$ is the phase change for the received signal in a time period of length T and λ is the wavelength of acoustic signal at frequency F_0. From Eq. (13), we translate distance calculation during a time period $[0, T]$ to calculation of the phase change $\tilde{\phi}$. The phase change can be calculated by the start phase and end phase during the time period. Denote ϕ_0 as the phase at time 0 and ϕ_T the phase at time T, we have $\tilde{\phi} = \phi_0 - \phi_T$.

1) Sampling based phase calculation: We show how to use the samples to derive the phase change $\tilde{\phi}$ in a time window $[0, T]$ containing n samples. Intuitively, the samples contain the information of phase change. For example, the number of local maximum (or minimum) N_{max} should correspond to the maximum number of cycles contained in the signal, as long as the sampling frequency F_s is larger than the Nyquist sampling rate. Therefore, the phase change $\tilde{\phi}$ can be approximated as $\tilde{\phi} = N_{max} \cdot 2\pi$. Combined with Eq. (13), we can approximate the moving distance $N_{max}\lambda - cT$. It can be seen that the approximation error is less than a wavelength, i.e. $\lambda = c/F = 1.7$ cm when $F_0 = 20000$ Hz.

We further show how to improve the accuracy in practice. First, we have the following lemma.

Lemma 1: The expected number of local maximums for a signal of phase change $2\pi N + \phi_0$ $(0 \le \phi_0 < 2\pi)$ is $N + \phi_0/2\pi$.

Proof 1: Without loss of generality, we assume $0 \le \phi_0 \le \pi/2$. To calculate the expected number of local maximum. We set N_1 as the number of local maximum when $\phi_0 = 0$. We assume the start of the signal is uniformly distributed in a cycle, i.e., $[0, 2\pi]$. As shown in Figure 4, the expected number of local maximum is calculated by

$$\bar{N} = \int_0^{\frac{\pi}{2}-\phi_0} N_1 + \int_{\frac{\pi}{2}-\phi_0}^{\frac{\pi}{2}} (N_1 + 1) + \int_{\frac{\pi}{2}}^{2\pi} N_1 = N_1 + \frac{\phi_0}{2\pi}. \quad (14)$$

Similarly, we can extend the proof to the case of $\pi/2 < \phi_0 < 2\pi$.

Lemma 1 indicates that by calculating the expected number of local maximum, we can derive the phase change of the signal. Meanwhile, local maximum can be extended to any relatively fixed points in each cycle, e.g., local minimum.

2) Moving window based phase change estimation: In practice, a key challenge is how to obtain the expected number of local maximum. According to Lemma 1, it requires uniformly distributed sampling windows. However, as long as the first window is given, all following windows are determined given the fixed sample frequency. An intuitive approach is to randomly choose windows, which introduces a long delay to process all windows. We show how to derive the phase change based on the local maximum with discrete samples. Without loss of generality, we consider a signal of p cycles containing

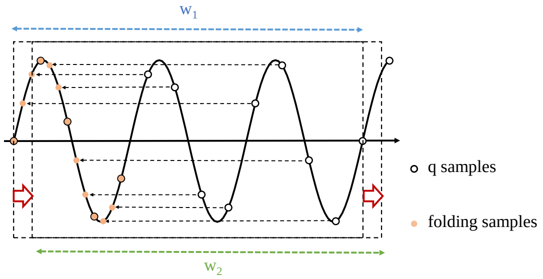


Fig. 5: Phase change calaulation ($q = 13$ and $p = 3$).

q samples as shown in Figure 5. Note p and q can be simply calculated by the smallest integer satisfying $p/q = F_c/F_s$. For example, if $F_s = 44100$ Hz and $F_c = 20000$ Hz, we have $p = 200$ and $q = 441$. For the ith sample of phase $\phi[i]$, denote its relative phase as $\phi[i] \bmod 2\pi$.

Lemma 2: The relative phases of q samples are uniformly distributed in $[0, 2\pi]$.

Proof 2: Without loss of generality, assume the signal has an initial phase 0. The relative phase of the ith sample can be calculated as $ip2\pi/q \bmod 2\pi = (ip \bmod q)2\pi/q$. The result of $ip \bmod q$ are pairwise distinct for $0 \leq i < q$. Therefore, the relative phases of q samples are evenly distributed in $[0, 2\pi]$. For example, as shown in Figure 5, there are 13 samples covering 3 cycles, i.e., $q = 13$ and $p = 3$. Folding those 13 samples into a single cycle results in uniformly distributed samples in the cycle.

3) Differentiated window based phase change estimation: CW-PE still incurs a high overhead as the window needs to be moved q times. We further propose an efficient method to improve the efficiency, namely Differentiated Window based Sample Counting for Phase Change Calculation (DW-PC).

Assume there are two windows w_1 and w_2, each of which contains q samples that cover p cycles of signal. Denote the q samples in w_1 and w_2 by $m_i(1 \leq i \leq q)$ and $m'_i(1 \leq i \leq q)$. We show that the phase change between m_1 and m'_1 can be calculated based on samples in w_1 and w_2. For each sample $m_i(0 < i \leq q)$ in w_1, define the Local Maximum Prefix (LMP) $l_i(0 < 0 \leq q)$ as the number of local maximum from the beginning of w_1 to m_i. Define the Local Maximum Prefix Sum (LMPS) of w_1 as $L = \sum_{i=1}^{q} l_i$. Similarly, the LMPS of w_2 is denoted as L'. We have the following lemma.

Lemma 3: Assume the LMPS of m_1 and m'_1 are L and L' respectively, the phase change between m_1 and m'_1 is $(L' - L)\frac{2\pi}{pq}$.

Proof 3: Lemma 2 shows that the relative phase of q samples are evenly distributed in $[0, 2\pi]$ with inter-distance $2\pi/q$. As shown in Figure 5, we can virtually fold all samples into a cycle to obtain uniformly distributed samples in the cycle. Moving the window by $2\pi/q$ causes the local maximum prefix of exactly one sample increases (decreases) by 1. As a result, the LMPS is increased by 1. Therefore, if the LMPS is increased by n, i.e. $L - L' = n$, the window is moved by $n2\pi/q$. Thus the phase change between m_1 and m'_1 is $(L' - L)2\pi/q$.

Algorithm 1 DW-PC(m, $\tilde{\phi}$)

```
Input: $m[i](i = 1, 2, \ldots)$, the samples continuously feeded from the sampling component.
Output: the phase change $\tilde{\phi}[i](i = 1, 2, \ldots)$.
1: $\tilde{\phi}[1] = 0$
2: $N_{max} = LMPS(m[1], m[2], \ldots, m[q])$
3: for i = 2; ;i++ do
4:     $N'_{max} = LMPS(m[i], m[i+1], \ldots, m[i+q-1])$
5:     $\tilde{\phi}[i] = (N'_{max} - N_{max}) \cdot 2\pi/q$
6: end for
```

Lemma 3 shows the relationship between the LMPS difference and phase change. According to Lemma 3, we can use the LMPS difference of two windows to estimate the phase change between the start of two windows. If the LMPS difference of two windows is n, the phase change $\tilde{\phi}$ can be calculated as $n2\pi/q$. It can also be seen that the error e_ϕ is at most $2\pi/q$. Otherwise, the LMPS difference of those two windows should not be n. Based on the phase change $\tilde{\phi}$, according to Eq. (13), we can calculate the moving distance by phase change.

According to Eq. (13), the moving distance error can be calculated as $\frac{c \cdot e_\phi}{2\pi F_0}$. For $F_0 = 20000$ and $q = 100$, we can see that the distance error by this method is only about 0.17 mm. Based on DW-PC, a mobile phone can continuously measure the moving distance. It can be seen that DW-PC can even update the moving distance for each sample, supporting efficient and accurate position measurement and motion tracking. For example, when q is set to 100, only 100 samples are required for each window, i.e., DW-PC can calculate the moving distance with a delay of $100/F_s = 2.3$ ms.

Algorithm 1 shows the simplified major steps of DW-PC. The array $\tilde{\phi}[\cdot]$ is used to store the phase change. Line 1-2 initialize the parameters. Line 4 calculates the N'_{max} for window w_2. Line 5 calculates the phase change based on Lemma 3. It can be seen that DW-PC measures the phase change with at most a linear computation overhead to the window length (calculate the local maximum and LMCPS). Usually, the window length is very small (e.g., 100), leading to a very small computation overhead. Therefore, DW-PC can support accurate and efficient distance movement measurement. The performance of DW-PC is also validated in Section V.

B. 2D/3D Tracking

2D and 3D tracking can be achieved based on 1D tracking. Assume the distance between two speakers A and B is d_0 in 2D tracking. As shown in Figure 6, we build the axis with A as the origin and x-axis along the direction from A to B. Assume the mobile phone moves from X_0 to X_1 and the position of X_0 is known.

We show how to calculate the new position X_1 by DW-PC. First, we can calculate the distance a_1 and a_2 towards signal source A and B by DW-PC. Therefore, we can calculate the length $\overline{X_1A} = \overline{X_0A} - a_1$ and $\overline{X_1B} = \overline{X_0B} - a_2$. Accordingly, we can calculate $\cos\alpha = \frac{d_0^2 + \overline{X_1A}^2 - \overline{X_1B}}{2d_0\overline{X_1A}}$. The position (x_1, y_1) of X_1 can be calculated as $x_1 = \overline{X_1A} \cdot \cos\alpha$

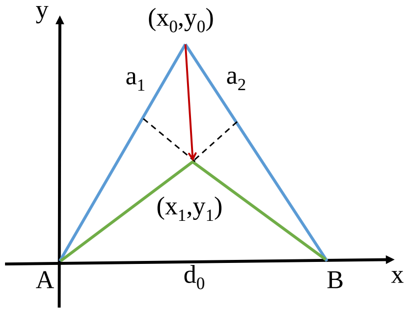


Fig. 6: 2D tracking based on DW-PC.

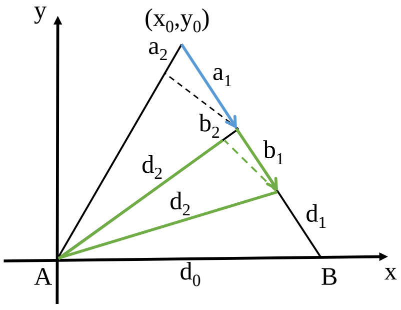


Fig. 7: Initial position of the mobile phone.

and $y_1 = \overline{X_1A} \cdot \sin\alpha$. Similarly, 3D tracking can be achieved by 3 signal sources. Here we omit the details.

C. Initial Position of Signal Source

There are two types of information that should be determined for most acoustic motion tracking approaches [24] [13] [21], i.e., the initial position of mobile phone and the initial position of signal source. The first requirement is to calculate the initial position of the signal source. Assume there are two signal sources A and B, as shown in Figure 7, calculating the initial position is equal to calculate the distance between two signal sources. As DW-PC can directly measure the distance a mobile phone has moved, we move the mobile phone from signal source A to source B. The distance between two signal sources A and B can then be calculated by DW-PC.

D. Initial Position of Mobile Phone

Another important step is to measure the initial position of mobile phone. In [24] [13], particular filtering method is used to derive the initial position. Intuitively, a large collection of possible initial positions are generated, each of which is tested according to the movement information. Finally, the centroid of the remaining particulars is calculated as the initial position. This introduces a high overhead and a relatively high measurement error [24].

In our approach, we show how to derive the initial position using DW-PC. We propose a method in which a user only needs to move the mobile phone for a certain distance towards a signal source or move the mobile phone from a signal source to any position to calculate the initial position. We call this method moving while initialization (MOWI).

As shown in Figure 7, assume the distance between A and B is d_0. Here we mainly show how to measure the initial position by moving the mobile phone towards the signal source. The method by moving the mobile phone from the signal source to any initial position is similar. Suppose the initial position of mobile phone is point X. A user moves the mobile phone from X to Z, passing a point Y. During the moving process, we can calculate the distance from X to Y and Y to Z using DW-PC. Thus we can calculate the distance for a_1, a_2 for the movement from X to Y, and b_1 and b_2 for the movement from Y to Z respectively. Denote the angle $\angle XBA$ as α, the distance $\overline{ZB}$ as d_1 and the distance $\overline{ZA}$ as d_2, we have

$$\begin{cases} \cos\alpha = \frac{d_1^2+d_0^2-d_2^2}{2d_0d_1} \\ \cos\alpha = \frac{(d_1+b_1)^2+d_0^2-(d_2+b_2)^2}{2d_0(d_1+b_1)} \\ \cos\alpha = \frac{(d_1+b_1+a_1)^2+d_0^2-(d_2+b_2+a_2)^2}{2d_0(d_1+b_1+a_1)} \end{cases} \tag{15}$$

Solving this equation, we obtain

$$d_2 = \frac{a_1^2b_1 - a_2^2b_1 + a_1b_1^2 - 2a_2b_1b_2 + a_1b_2^2}{2(a_2b_1 - a_1b_2)}. \tag{16}$$

Plugging d_2 to the equation array, we can obtain the value of d_1. We omit the details for the lengthy formula of d_1. Based on d_1 and d_2, we can obtain the coordination (x, y) of X.

IV. Implementation

We implement Vernier on Android 6.0.1 as an App. The signal sources of Vernier Tracker can be most COTS speakers like the speakers on TV. In our implementation, we use the speaker (SV S840B) as shown in Figure 8 (a). The speakers is connected to a mobile phone which can play audio files containing waves of different frequency. Instead of using a group of sine and chirp signals on different frequency bands [13], our approach uses sine waves (e.g., 20000 Hz and 17500 Hz for 2D tracking in our implementation). The sine wave files are generated on a desktop computer. Vernier on Android receives and analyzes the received signal, and displays the real-time location on the screen. Meanwhile, Vernier Tracker can also record all signal data for further analysis and comparison in evaluation.

A. Moving Distance Measurement

We use the equipment in Figure 9 (a) to measure distance accurately. The mobile phone is fixed on the platform of the equipment. We can move the platform horizontally and vertically by rolling the rocker. Figure 9 (b) shows the measured distance on the mobile app. In the app, we draw a virtual rule for 10 mm.

There are 25 scales on the rocker and the platform moves 1.25 mm when the rocker rolling one circle (0.05 mm for each scale). We can move the platform horizontally and vertically so we can obtain the ground truth for the mobile phone position.

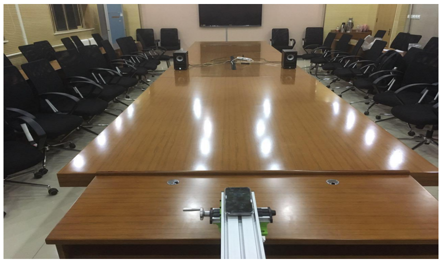
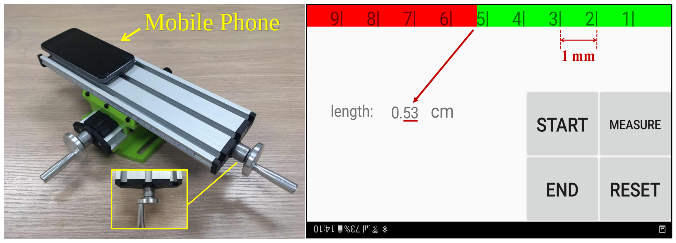


Fig. 8: Experiment scenario.



length: 0.53 cm

(a) (b)

Fig. 9: (a) Moving distance measurement; (b) mobile app.

B. Clock Inconsistency

In practice implementation, we find that there exists a clock inconsistency for the generated signal and received signal, which further leads to a distance measurement error. We conduct an experiment to validate the impact of clock inconsistency. We noticed the received signal frequency, even when the mobile phone is static, is different from the signal source. This leads to a non-zero moving speed and a continuously increasing distance. To address the frequency inconsistency, we propose a linear frequency compensate (FC) to calibrate the frequency for the signal source and mobile phone.

Assume the frequency shift between the mobile phone and the signal source is α. A signal at frequency F_0 is received at frequency $(1+\alpha)F_0$. By keeping the mobile phone static, we calibrate the frequency as follows. If there is no frequency drift, the calculated phase change by DW-PC for a time period T should be $TF_0 2\pi$. Assume the calculated the phase change by DW-PC for a time period T is ϕ, we can calculate the frequency drift $\alpha = \frac{\phi}{2\pi T}$. We use α to compensate the frequency shift between the signal source and mobile phone.

V. Evaluation

A. Evaluation Methodology

We mainly evaluate the performance of Vernier from the following aspects.

- Tracking accuracy: we show the accuracy of Vernier in motion tracking compared with other approaches.
- Delay: the time consumption of Vernier and other approaches.
- Robustness: performance in different application environments.
- Overall performance: we also evaluate the overall performance for different tracking paths.

B. Tracking Accuracy

We first measure the 1D distance tracking error. In this experiment, we vary the initial distance from the mobile phone to the speaker and calculate the corresponding distance measurement error. The results show that the error is under 2 mm even when the distance between the mobile phone and speaker is 7 m. The result is shown in Figure 10 (a). Figure 10 (b) shows the detailed measurement error of different moving distance for our approach. We move the mobile phone for different distance from 1 cm to 10 cm (larger-scale measurement is hard to achieve because of the limitation of our equipment as shown in Figure 9 (a)). For each distance, we measure the moving distance for 30 times. The accumulated error is small for different moving distance. This enables our approach for many applications, such as video gaming, VR, smart appliances control, etc.

We further measure the tracking accuracy in 2D case. In this experiment, we move the mobile phone following the a path of ”L” whose size is about 2 cm × 2 cm. Figure 11 (a) shows the tracking error of different distance from the mobile phone to the speakers. Figure 11 (b) shows how the tracking error influenced by the speakers separation when the mobile phone is 3 m away from the speaker. We can see the error for different distance is slightly larger than that in 1D. Nevertheless, the error is still under 4 mm.

C. Delay Performance

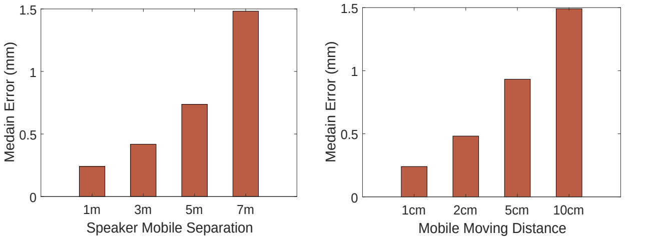
In our evaluation, we implement most recent acoustic tracking approaches including Doppler Effect based approach [24] (denoted by Doppler), phase based approach [21] (denoted by LLAP) and FMCW based approach [13] (denoted by FMCW) for comparison. For fair comparison, we use the same recorded signal in performance comparison for different approaches. The FMCW based approach [13] requires both sine wave and chirp signal for tracking, so we generate chirp signals from 8500 Hz to 18500 Hz for this approach. The mobile phone used in our evaluation is Samsung Galaxy S7 with Android 6.0.1.

We implement an active version of this approach by using the phase calculation method proposed in [21]. We directly use the received signal from the speaker instead of the reflected signal. By using such a method, the tracking range becomes much larger than before. We denote such a method Phase+. Figure 12 (a) shows average time consumption for each sample using different approaches on Android device.

D. Robustness

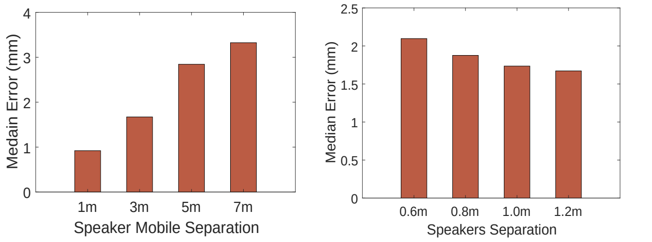
In this experiment, we mainly show how our approach can work in different environments for practical scenarios. We evaluate the performance from the following aspects:

- Different intensity of noise.
- Different devices.
- Different multipath scenarios.



(a) (b)

Fig. 10: 1-D accuracy: (a) Different initial distance; (b) Different moving distance.



(a) (b)

Fig. 11: 2-D accuracy: (a) Different initial distance; (b) Different speaker separation.

The impact of noise intensity. In this experiment, we vary the noise volume to different levels, i.e., around 40 db (library room), 50 db (air conditioner's noise), 60 db (human talking) and 70 db (noisy street). Then we evaluate the performance of Vernier under different levels. The result is shown in Figure 12 (b). We can see that the error increases as the noise level increases. The overall error for all distances is still very small.

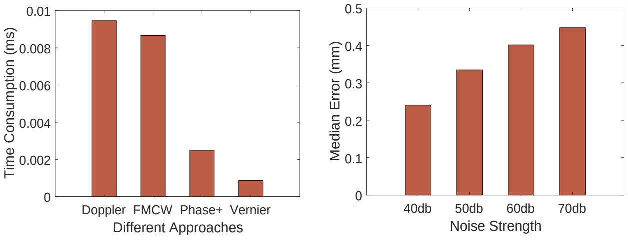
The impact of device. We also tested other mobile phones (e.g., Sony L50t) and other speakers and the results are similar. Figure 13 (a) shows the results on different devices when the mobile phone moves 1 cm.

The impact of multipath scenarios. Ultra-sound has a strong directionality because of its short wavelength. As a result, the influence of multipath effect in active tracking system is especially weak. Figure 13 (b) shows the distance measurement error in scenario with/without the reflection path by the surface of the desk when the mobile phone moves 1 cm at the distance of 1 m. The result demonstrates that the influence of multipath effect in our experiment is slight.

E. Overall Performance

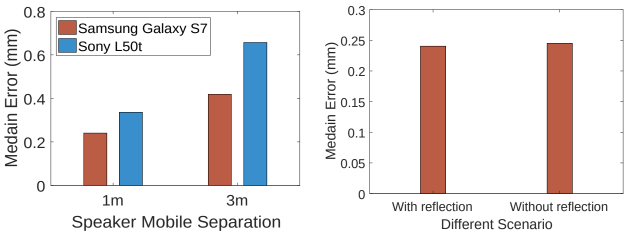
We evaluate Vernier Tracker using the method as in [13]: the similarity between the Vernier Tracker reported trace and the standard drawing template. In this experiment, we examine the performance of Vernier to draw different figures. we print different templates (banana, snake, hat and rabbit) and move the mobile phone following the curve of printed templates.

As shown in Figure 14, we plot the tracking results and compare them with the original templates. As we can see, Vernier can follow the curves. All the details in the original templates can be plotted, indicating a high accuracy of our approach. It should also be noted that somewhere in the drawing may not be as smooth as the original template. We check the data and found that this may due to unstable drawing as it is very difficult to control the drawing exactly and smoothly following the original curve. Nevertheless, the results demonstrate that Vernier preserves the details of the original templates and can be used in real applications.



(a) (b)

Fig. 12: (a) Time consumption of different approaches; (b) Median error of different noise intensity.

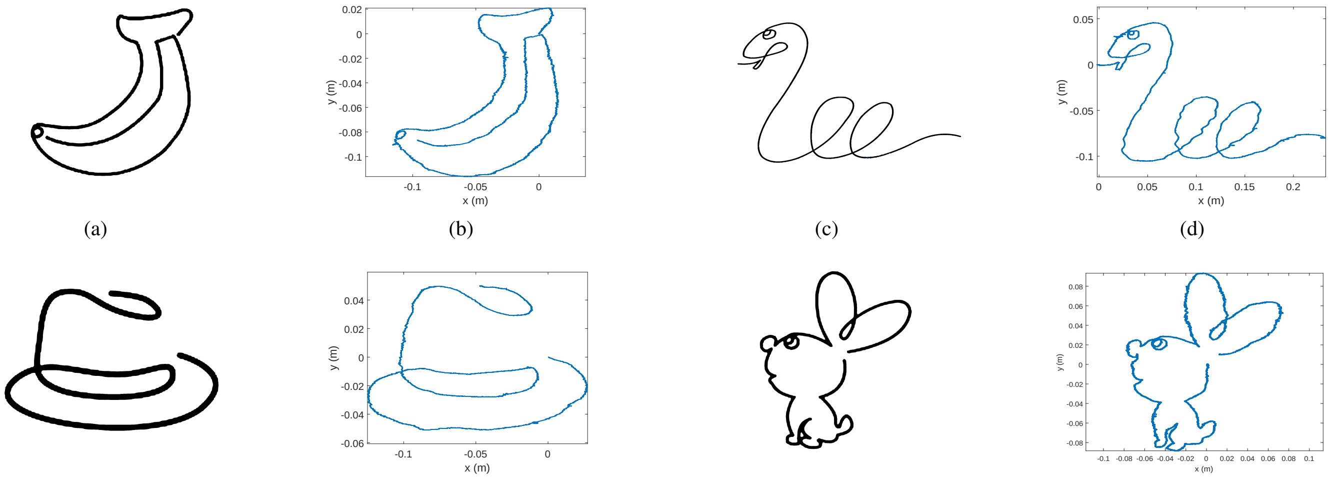


(a) (b)

Fig. 13: (a) Median error on different devices; (b) Median error in with/without reflection scenario.

VI. Conclusion

In this paper, we present Vernier, an efficient and accurate acoustic motion tracking approach on commodity mobile devices. We address the fundamental limitations of existing approaches in terms tracking accuracy, overhead and delay. In Vernier, we present a novel differentiated window based sample counting for phase estimate and mobile motion tracking. We theoretically show that Vernier can achieve accurate motion tracking with a window much smaller than existing approaches while incurring a small computation overhead and delay. We implement Vernier in Android and examine its performance with Samsung Galaxy S7 and Sony L50t. We conduct extensive experiments to evaluate the performance of Vernier. The results show that Vernier can achieve accurate motion tracking with error less than 4 mm in 7 m. We believe the design of Vernier is general and can facilitate various mobile applications such as video gaming, VR, AR, etc.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 14: Using 2D tracking to draw different templates. (a) (c) (e) (g) are the original templates of banana, snake, hat and rabbit. (b) (d) (e) (h) are drawing results.

VII. Acknowledgements

This work is in part supported by National Natural Science Fund China for Excellent Young Scholars (No. 61722210)NSFC, key program (No. 61532012, 61432015), NSFC No. 61572277, 61529202. Jiliang Wang is the corresponding author.

References

[1] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller. 3d tracking via body radio reflections. In *Proceedings of USENIX NSDI*, 2014.

[2] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings of IEEE INFOCOM*, volume 2, pages 775–784. Ieee, 2000.

[3] X. Bian, G. D. Abowd, and J. M. Rehg. Using sound source localization in a home environment. In H. W. Gellersen, R. Want, and A. Schmidt, editors, *Proceedings of Pervasive Computing*, 2005.

[4] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson. Phaser: enabling phased array signal processing on commodity wifi access points. In *Proceedings of ACM MobiCom*, pages 153–164. ACM, 2014.

[5] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, W. Xi, Z. Jiang, and Z. Wang. Twins: Device-free object tracking using passive tags. In *Transactions on Networking (TON)*, volume 25, pages 1605–1617. IEEE/ACM, 2016.

[6] W. Huang, Y. Xiong, X. Li, H. Lin, X. Mao, P. Yang, Y. Liu, and X. Wang. Swadloon: Direction finding and indoor localization using acoustic signal by shaking smartphones. *IEEE Transactions on Mobile Computing*, 14(10):2145–2157, 2015.

[7] Y. Jiang, Z. Li, and J. Wang. Ptrack: Enhancing the applicability of pedestrian tracking with wearables. In *Proceedings of IEEE ICDCS*, 2017.

[8] K. Kalgaonkar and B. Raj. One-handed gesture recognition using ultrasonic doppler sonar. In *Proceedings of IEEE Acoustics, Speech and Signal Processing*, 2009.

[9] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of ACM MobiCom*, 2014.

[10] P. Lazik and A. Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the ACM SenSys*, 2012.

[11] K. Liu, X. Liu, and X. Li. Guoguo: Enabling fine-grained smartphone localization via acoustic anchors. *IEEE Transactions on Mobile Computing*, 15(5):1144–1156, 2016.

[12] C. V. Lopes, A. Haghighat, A. Mandal, T. Givargis, and P. Baldi. Localization of off-the-shelf mobile devices using audible sound: Architectures, protocols and performance assessment. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10(2):38–50, Apr. 2006.

[13] W. Mao, J. He, and L. Qiu. Cat: High-precision acoustic motion tracking. In *Proceedings of ACM MOBICOM*, 2016.

[14] J. Paradiso, C. Abler, K. Hsiao, and M. Reynolds. magic carpet: physical sensing for immersive environments. In *Proceedings of ACM CHI*, 1997.

[15] J. Qiu, D. Chu, X. Meng, and T. Moscibroda. On the feasibility of real-time phone-to-phone 3d localization. In *Proceedings of ACM SenSys*, 2011.

[16] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. In *Proceedings of ACM MobiCom*, 2015.

[17] S. Tarzia, R. Dick, P. Dinda, and G. Memik. Sonar-based measurement of user presence and attention. In *Proceedings of ACM UbiComp*, 2009.

[18] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *Proceeding of USENIX NSDI*, pages 165–178, 2016.

[19] R. Wand, A. Hopper, V. Falcao, and J. Gibbons. The active badage location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1997.

[20] J. Wang, D. Vasisht, and D. Katabi. Rf-idraw: Virtual touch screen in the air using rf signals. In *Proceedings of ACM SIGCOMM*, 2014.

[21] W. Wang, A. X. Liu, and K. Sun. Device-free gesture tracking using acoustic signals. In *Proceedings of ACM MOBICOM*, 2016.

[22] T. Wei and X. Zhang. mtrack: High-precision passive tracking using millimeter wave radios. In *Proceedings of ACM MobiCom*, 2015.

[23] J. Xiong and K. Jamieson. Arraytrack: a fine-grained indoor location system. In *Proceedings of USENIX NSDI*, pages 71–84, 2013.

[24] S. Yun, Y.-C. Chen, and L. Qiu. Turning a mobile device into a mouse in the air. In *Proceedings of ACM MobiSys*, 2015.