ATP: Acoustic Tracking and Positioning under Multipath and Doppler Effect

Guanyu Cai, Jiliang Wang
School of Software, Tsinghua University
cgy22@mails.tsinghua.edu.cn, jiliangwang@tsinghua.edu.cn

***Abstract*—Acoustic tracking and positioning technologies using microphones and speakers have gained significant interest for applications like virtual reality, augmented reality, and IoT devices. However, existing methods still face challenges in real-world deployment due to multipath interference, Doppler frequency shift, and sampling frequency offset between devices. We propose a versatile Acoustic Tracking and Positioning (ATP) method to address these challenges. First, we propose an iterative sampling frequency offset calibration method. Next, we propose a Doppler frequency shift estimation and compensation model. Finally, we propose a fast adaptive algorithm to reconstruct the line-of-sight (LOS) signal under multipath[1]. We implement ATP in Android and PC and compare it with eight different methods. Evaluation results show that ATP achieves mean accuracy of 0.66 cm, 0.56 cm, and 1.0 cm in tracking, ranging, and positioning tasks. It is 2×, 6×, and 5.8× better than the state-of-the-art methods. ATP advances acoustic sensing for practical applications by providing a robust solution for real-world environments.**

***Index Terms*—tracking, positioning, acoustic signal, multipath, Doppler effect**

I. Introduction

The rapid development of devices on the Internet of Things (IoT) has led to an increasing interest in the use of microphones and speakers for active or passive sensing tasks such as tracking [1]–[10], ranging [11]–[13], and positioning [14]–[18]. Applications based on these tasks include Virtual Reality, Augmented Reality, mobile gaming, smart appliances, etc. Despite the availability of many methods for acoustic tracking, ranging, and positioning, they still encounter significant challenges when applied to practical scenarios:

Multipath environment. A common application scenario for acoustic sensing is the indoor environment, which suffers from dense multipath interference [3], [19]. Multipaths significantly hurt the accuracy of acoustic sensing systems. Single-tone-based approaches are inherently vulnerable to multipath interference, yet previous works ignore this problem [1], [4], [10]. Although the frequency modulated continuous wave (FMCW) can split multipath signals, the limited ultrasonic bandwidth available in commercial devices (18 kHz to 24 kHz) prevents splitting two paths separated in cm level [5], [6]. RABIT [3] proposes to apply MUSIC [20] to resolve multipath and enhance tracking accuracy. However, it requires knowing the exact number of multipaths. Meanwhile, the eigenvalue decomposition required by MUSIC is time-consuming.

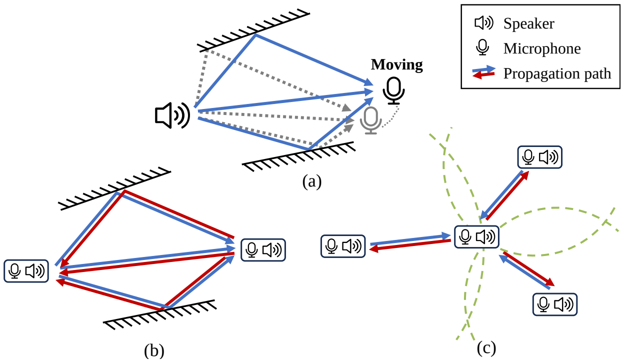

Fig. 1. ATP addresses multipath, Doppler Effect, and sampling frequency offset problems for (a) tracking, (b) ranging, and (c) positioning.

Doppler frequency shift. The Doppler frequency shift (DFS) leads to non-negligible errors for methods based on time of flight (ToF) [3], [5], [11], [12], [14], [16]. Depending on the signal used, there are speed search-based [12], [16] and fractional Fourier transform (FrFT) search-based [19] approaches to estimate the Doppler effect. Those approaches are, however, time-consuming. Other techniques [3], [5] involve simultaneously transmitting FMCW and single-tone signals. The received single-tone signals is analyzed to calculate the relative moving speed. This speed is then used to compensate for DFS on FMCW-based ToF measurements. As mentioned earlier, multipath can impact the analysis of single-tone signals, thereby reducing the accuracy of these methods. Furthermore, simultaneous transmission of two signals can lead to energy waste and frequency distortion [21].

Sampling frequency offset. Sampling frequency offset (SFO) between different mobile devices causes a frequency shift of the received signal [5]. This frequency shift further causes the estimated ToF to drift linearly over time. The methods [1], [5] require the devices to be static to measure the SFO every few minutes [5]. Our experiments also demonstrate that SFO should be frequently corrected. Thus, we need to quickly and accurately estimate the SFO while existing methods [1], [5] cannot meet the requirements.

To address the above challenges and make acoustic position-related sensing more practical, we propose a versatile solution for Acoustic Tracking, ranging, and Positioning (ATP) in environments with severe multipath and Doppler effect as shown in Fig. 1. The speaker transmits a short single-tone

[1] LOS signal arrives first, non-LOS signals follow under multipath scenarios.

(sent once) followed by triangular FMCW signals. And the microphone records the signal. We first use this short single-tone for calculating SFO. Then, we use a triangular FMCW signal to compensate for the DFS. Finally, we reconstruct the line-of-sight (LOS) signal and calculate accurate ToF. The proposed method has many applications, as demonstrated in Fig. 2, where tracking, ranging, and positioning tasks can be performed using relative ToF, two-way ranging, and trilateration. Our method can also be incorporated with other FMCW-based approaches [2], [3], [5], [7], [11], [14]. They can use our ToF results to improve their accuracy. In the design of ATP, we address the following fundamental problems:

(1) How to accurately and efficiently resolve the FMCW LOS signal? Conventional FMCW methods mix received and sent signals to produce single-tones of varying frequencies. After applying the fast Fourier Transform (FFT) and searching for peaks, we can map single-tones' frequencies to distances. Due to the narrow bandwidth and limited signal length, LOS and non-LOS (NLOS) peaks interfere. Our method efficiently focuses on only LOS and close NLOS. It adapts to remove NLOS interference and reconstruct the LOS signal, resulting in an accurate LOS frequency.

(2) How to compensate Doppler frequency shift? FMCW signals transmitted between moving devices suffer from DFS. A common method called Doppler FFT [22] can be used for speed estimation. However, it has a maximum unambiguous speed limit, e.g., 21.6 cm/s for 40 ms and 20 kHz central frequency acoustic FMCW signal. The triangular FMCW model used in radar [23] can also be used for speed measurement. However, the relatively low sound propagation speed can lead to a significant error in the acoustic signal. Other search-based methods [16], [19] are very time-consuming. We model the Doppler effect on acoustic triangular FMCW and propose an accurate and efficient method for directly calculating speed. We use this speed to compensate DFS.

(3) How to accurately and efficiently calibrate sampling frequency offset? We need to calibrate SFO frequently, e.g., through our experiments and [5], SFO keeps relatively static in only several minutes. The method [5] keeps devices static while sending FMCW and monitors distance changes to estimate SFO. This method is time-consuming because it needs to send many chirps. We measure the SFO based on the received single-tone's frequency. To minimize the overhead, we propose utilizing an iterative frequency estimation method, which is much more efficient than FFT with zero padding.

Our main contributions include:

- We investigate theoretically and experimentally the fundamental limitations of existing acoustic tracking, ranging, and positioning systems in real scenarios. We reveal that these limitations lead to substantial measurement errors, which existing methods do not address well.
- We propose ATP, a novel versatile acoustic sensing approach that can handle SFO, DFS, and multipath interference in real-world scenarios. ATP can be used in different acoustic sensing tasks, including active and passive tracking, ranging, and positioning.
- We implement ATP and evaluate its performance for tracking, ranging, and localizing tasks using smartphones and speakers. The results demonstrate that ATP achieves mean accuracy of 0.66 cm, 0.56 cm, and 1.0 cm in tracking, ranging, and positioning, respectively. It is 2×, 6×, and 5.8× better than state-of-the-art methods.

II. Prior Arts and Limitations

Acoustic tracking techniques can be divided into single-tone-based [1], [4], [7], [8], [10], FMCW-based [2], [3], [5], [6], [11], [13], [24] and other categories [8], [16], [25]–[28]. Here, we provide a brief introduction to those techniques and their limitations. Suppose a speaker is emitting a signal $s_T(t)$, and a moving microphone with a relative speed of $v(t)$ is continuously recording $s_T(t)$. The signal travels through n paths with a time-varying path length of $d_i(t)$. The sound propagation speed is c. Assume $i = 1$ indicates the LOS path and $i = 2...n$ indicates the NLOS paths.

A. Single-tone-based Approaches

We show the single-tone-based tracking approaches (i.e., $s_T(t) = \cos(2\pi f_0 t)$). We have the recorded signal

$$s_R(t) = \sum_{i=1}^{n} A_i \cos(2\pi f_0 (t - \frac{d_i(t)}{c})), \tag{1}$$

where A_i is the amplitude, f_0 is single-tone's frequency sent.

1) Tracking Based On Frequency Change: In AA-Mouse [10], $s_R(t)$ in a duration T time window $w(t)$ is

$$s_R(t)w(t) = \begin{cases} \sum_{i=1}^{n} A_i \cos(2\pi f_0 (t - \frac{d_i(t)}{c})) & \text{if } t < |T/2| \\ 0 & \text{if } t > |T/2| \end{cases}. \tag{2}$$

Applying FFT to Eq. (2), it has

$$S_R(f) * W(f) = \sum_{i=1}^{n} A_i e^{j\phi_i} \operatorname{sinc}(f - f_0(1 - \frac{v_i(t)}{c})), \tag{3}$$

where $W(f)$ is the Fourier transform of the rectangular window, which is sinc function $\operatorname{sinc}(f) = \frac{\sin(\pi f/T)}{(\pi f/T)}$. It calculates $f_1(t) = \arg\max |S_R(f) * W(f)|$. Also, $f_1(t) = f_0(1 - \frac{v_1(t)}{c})$. So $v_1(t) = c(1 - \frac{f_1(t)}{f_0})$. The relative moving distance is calculated by $d_1(t) = \int v_1(t) \mathrm{d}t$.

Due to multipath interference, $f_1(t)$ cannot be accurately measured by $\arg\max |S_R(f) * W(f)|$, leading to a nonnegligible tracking error.

2) Tracking Based On Phase: LLAP [4] multiplies the $s_R(t)$ by $\cos(2\pi f_0 t)$ and $-\sin(2\pi f_0 t)$ and passes the result through a low-pass filter. It has

$$I_R(t) = \mathrm{LPF}(s_R(t)\cos(2\pi f_0 t)) = \sum_{i=1}^{n} A_i' \cos(-2\pi f_0 \frac{d_i(t)}{c}), \tag{4}$$

$$Q_R(t) = \mathrm{LPF}(s_R(t)(-\sin(2\pi f t))) = \sum_{i=1}^{n} A_i' \sin(-2\pi f_0 \frac{d_i(t)}{c}). \tag{5}$$

If $n = 1$, it has $d_1(t) = -c \cdot \frac{\arctan(Q_R(t)/I_R(t))}{(2\pi f_0)}$ [4]. If $n > 1$, it cannot solve $d_1(t)$ based on $I_R(t)$ and $Q_R(t)$.

3) Tracking Based On Phase Without Mixing: Vernier [1] avoids the FFT, mixing, and filtering steps for improving the refresh rate. First, it counts the number of local maximum N_{max} in the time window $[0, T]$. So, the phase change $\tilde{\phi}$ can be approximated as $\tilde{\phi} = N_{max} \cdot 2\pi$. Then, the moving distance is calculated by $N_{max}\lambda - cT$. However, Vernier cannot work in multipath environments as the local maximum is distorted.

Summary of limitations. (1) Multipath: single-tone-based approaches suffer from multipath interference. Multipaths will distort the signal's frequency and phase. Thus, an error will occur in the estimated distance. (2) Error accumulation: single-tone-based tracking produces relative moving distance. It suffers from error accumulation over time.

B. FMCW-based Approaches

Many approaches resort to FMCW or the so-called chirp signal to address the multipath. When $s_T(t) = \cos(2\pi(f_0 t + \frac{1}{2}k_0 t^2))$, the recorded signal is

$$s_R(t) = \sum_{i=1}^{n} A_i \cos(2\pi(f_0(t-\frac{d_i(t)}{c})+\frac{1}{2}k_0(t-\frac{d_i(t)}{c})^2)), \quad (6)$$

where f_0 is starting frequency and k_0 is chirp rate.

By multiplying $s_T(t)$ and $s_R(t)$ and passing the result through a low pass filter, we can obtain the mixed signal

$$m_R(t) = \sum_{i=1}^{n} A_i \cos(2\pi(f_0 \frac{d_i(t)}{c} + \frac{1}{2}k_0(2t\frac{d_i(t)}{c} - (\frac{d_i(t)}{c})^2))). \quad (7)$$

1) Frequency Based ToF: In CAT [5], $m_R(t)$ in a time window $w(t)$ can be written as:

$$m_R(t)w(t) = \begin{cases} \sum_{i=1}^{n} A_i \cos(2\pi \frac{k_0 d_i(t)}{c} t + \phi_i) & \text{if } t < |T/2| \\ 0 & \text{if } t > |T/2| \end{cases}, \quad (8)$$

where $w(t)$ is the rectangle window, ϕ_i is the static phase (assume that $d_i(t)$ remains constant in rectangle window duration T). It applies FFT to Eq. (8), and have

$$M_R(f) * W(f) = \sum_{i=1}^{n} A_i e^{j\phi_i} \operatorname{sinc}(f - \frac{k_0 d_i(t)}{c}), \quad (9)$$

where $W(f)$ is the sinc function $\operatorname{sinc}(f) = \frac{\sin(\pi f/T)}{(\pi f/T)}$ which is the Fourier transform of the rectangular window. By searching for the highest n peaks and their corresponding f_i in $|M_R(f) * W(f)|$, it calculates $d_i(t) = \frac{cf_i}{k_0}$. If $n = 1$, this approach can achieve high accuracy by padding long enough zeros in FFT. When $n > 1$, n sinc functions interfere with each other and cause a peak offset to f_i, so d_i is inaccurate.

2) Cross-correlation Based ToF: BeepBeep [11] utilizes the correlation property of signals. The magnitude of cross-correlation between $s_T(t)$ and $s_R(t)$ is

$$\psi(t) = \sum_{i=1}^{n} A_i \rho \operatorname{sinc}(\pi B(t - \frac{d_i(t)}{c})), \quad (10)$$

where ρ is determined by B and T, and $\operatorname{sinc}(x) = \frac{\sin(x)}{x}$ [29]. It calculates ToF by $t_i = \arg\max(\psi(t))$. Then, it calculates $d_i = ct_i$. When $n > 1$, these n sinc functions interfere and create an offset to t_i, consequently leading to an offset of d_i.

3) Phase Based ToF: PDF [2] directly calculates the frequency of $m_R(t)$ by dividing the phase difference by the time difference. When $n = 1$, it calculates $f_1 = \frac{\phi_{m_R}(t_1) - \phi_{m_R}(t_2)}{t_1 - t_2}$ and $d_1(t) = \frac{cf_1}{k_0}$. However, it's prone to multipath and noise, just like the aforementioned phase-based methods.

4) MUSIC Based ToF: RABIT [3] uses MUSIC, a super-resolution algorithm for separating sine waves. It first computes the auto-correlation matrix of $m_R(t)$. It next conducts eigenvalue decomposition on this matrix to separate the signal and noise components. It then constructs a pseudo-spectrum with noise components and a defined steering vector. It locates peaks in the pseudo-spectrum to determine f_1. It calculates $d_1 = \frac{cf_1}{k_0}$. However, the MUSIC algorithm has a time complexity of $\mathcal{O}(N^3)$, much slower than the FFT algorithm's $\mathcal{O}(N \log N)$. Additionally, prior knowledge of the number of multipath components is required for MUSIC.

Summary of limitations. (1) Multipath: FMCW-based approaches suffer from interference from NLOS. (2) Doppler effect: FMCW approaches either require mixing $s_T(t)$ with $s_R(t)$ (CAT, PDF, and MUSIC) or correlating $s_T(t)$ with $s_R(t)$ (BeepBeep). They require that k_0 in $s_R(t)$ is the same as in $s_T(t)$. Doppler effect changes k_0 in the received $s_R(t)$, causing an error in the distance result.

C. Other Approaches

Some works attempt to use Received signal strength [27], [28], [30]. This strength can be utilized for low-precision tracking and positioning purposes. Other works attempt to use channel impulse response (CIR), which represents channel information. CIR can also be used to analyze propagation delay [8], [25], [26].

III. System Design

Our system is designed to achieve the following objectives:

- Anti-multipath interference: It should perform well in multipath environments.
- Anti-Doppler effect: It should work well with fast-moving devices (e.g., 1m/s).
- Accurate: It should be accurate enough to produce results with mm-level error.
- Fast: It should be fast enough to provide online results.

A. System Overview

In our system, the speaker (e.g., those on the smartphone, computer, or smart speaker) transmits the audio signal to the microphone. The audio signal consists of short single-tones and triangular FMCWs.

The system consists of four components, as shown in Fig. 2. (1) SFO estimation: we use short single-tones to calibrate SFO. This step is only performed at the beginning of tracking. (2) DFS estimation: we use cross-correlation to calculate the coarse delay τ_1 of triangular FMCW. Then, we use our model to calculate relative moving speed v and residual delay τ_2. (3) LOS estimation: we use v to construct a triangular FMCW under the Doppler effect. Then, we align and mix the constructed FMCW with the received FMCW and apply the

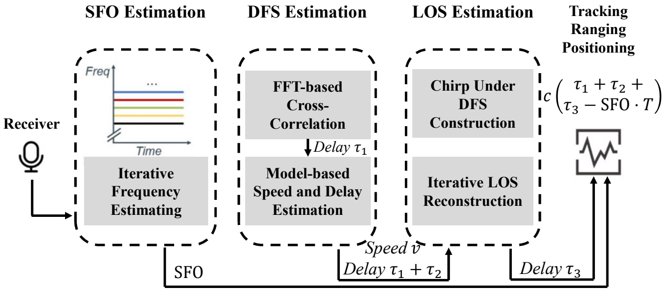
SFO Estimation
DFS Estimation
LOS Estimation
Tracking
Ranging
Positioning
Receiver
Freq
...
Time
Iterative Frequency Estimating
FFT-based Cross-Correlation
Delay τ_1
Model-based Speed and Delay Estimation
Chirp Under DFS Construction
Iterative LOS Reconstruction
$c\begin{pmatrix}\tau_1+\tau_2+\\ \tau_3-\text{SFO}\cdot T\end{pmatrix}$
SFO
Speed v
Delay $\tau_1+\tau_2$
Delay τ_3

Fig. 2. The main workflow of ATP.

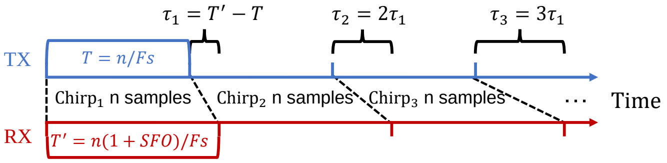

Fig. 3. Sampling frequency offset problem.

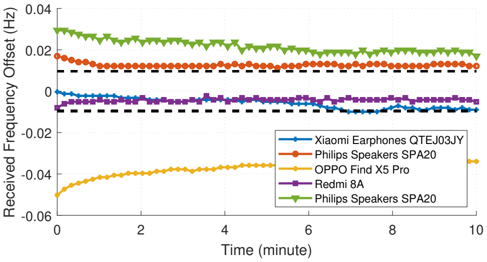

Fig. 4. The received frequency offset of a 20kHz single-tone transmitted by different devices. The black dashed line represents an error of 1 cm/min.

adaptive LOS reconstruction algorithm to the mixed signal. The algorithm reconstructs accurate LOS frequency. Finally, we map LOS's frequency to residual delay τ_3. (4) Tracking and Positioning: We use estimated SFO to correct this distance. The corrected distance is $c(\tau_1 + \tau_2 + \tau_3 - \text{SFO} \cdot T)$, where T is tracking duration. The distance is used for tracking, ranging (based on two-way ranging), and positioning (based on trilateration).

B. Sampling Frequency Offset Estimation

SFO occurs when the transmitter and receiver have different clocks, leading to errors in ToF measurements. When sending or receiving the same number of samples, the sender and receiver will experience different time durations. Fig. 3 shows an example to demonstrate the problem. In this scenario, a speaker sends a chirp with n samples. The sampling frequency of the transmitter is Fs, while the sampling frequency of the receiver is $\frac{Fs}{1+\text{SFO}}$. The delay of $\text{chirp}_1, \text{chirp}_2, ..., \text{chirp}_\text{m}$ will add up to $\tau_1, 2\tau_1, ..., m\tau_1$.

To further show the necessity of fast and accurate estimation of SFO, we transmit a 20 kHz single-tone by different devices, including smartphones, earphones, and loudspeakers. We use a Xiaomi Mi 11 to record the signal. By utilizing FFT with a sliding window of 10 seconds and a zero padding length of $5 \cdot 10^7$, we obtain the frequency offset of the received signal, as shown in Fig. 4. When the received frequency offset is 0.0096 Hz, the accumulated error in one minute is $0.0096/20000 \cdot 60 \cdot 346 = 1$ cm. According to the measured frequency offset in Fig 4, Oppo Find x5 Pro has an error of up to 5 cm in a minute. We also show that SFO can vary across time, even for the same device. For example, we repeat the experiment of the Philips Speaker SPA20 two times and find that the calculated frequency offset varies significantly in those two times.

We calculate the SFO $= \frac{f_0' - f_0}{f_0}$, where f_0 is the sent frequency and f_0' is the received frequency. We may use FFT to calculate f_0'. However, to measure SFO accurately, we need to pad many zeros in FFT. We analyze the consumption of FFT. Assume the tracking period is T, the distance measurement error Δd is

$$\Delta d = cT \cdot \text{SFO}, \tag{11}$$

where c is the sound propagation speed. Given $f_0 = 20$ kHz and $c = 346$ m/s, if we want to keep tracking errors under 1 cm/min, we should keep the SFO error in $4.817 \cdot 10^{-7}$. It means the frequency estimation error should be less than $4.817 \cdot 10^{-7} \cdot 20000 = 0.0096$ Hz. So, the FFT bin resolution should be higher than $2 \cdot 0.0096 = 0.0192$ Hz. When $Fs = 48$ kHz, the single-tone sequence length should be longer than $Fs/0.0192 = 2.5 \cdot 10^6$ (i.e., $1/0.0192 = 52.08$ s). Performing a $2.5 \cdot 10^6$ point FFT is time and power-consuming.

To reduce overhead, we design an SFO estimation method based on iterative frequency estimation of received single-tones [31]. Assume the transmitted signal is K single-tones with frequency f_i. We set $f_{i+1} - f_i = 1$ kHz to minimize the interference between them when doing discrete Fourier transform (DFT) [4], [5]. The received signal is

$$s(n) = \sum_{i=1}^{K} A_i e^{j(2\pi \frac{f_i'}{Fs} n + \phi_i)}, \tag{12}$$

where A_i is the attenuation, $f_i' = f_i(1 + \text{SFO})$ is the received frequency, and ϕ_i is the initial phase.

We summarize our method in Algorithm 1. First, we apply FFT without zero padding to $s(n)$ and find K peak indexes as $\hat{p}_1, \hat{p}_2, ..., \hat{p}_K$ in line 1. Then, we iteratively estimate an accurate offset $\hat{\delta}_i$ in line 6. The received single-tones' frequencies are calculated as $\frac{\hat{p}_i + \hat{\delta}_Q}{N} Fs$ in line 8. Finally, SFO is calculated by transmitted f_i and estimated received $\hat{f}_i'$ in line 10.

We prove that our algorithm can estimate the received frequency with an error of order N^{-2} for $K = 1$. We omit the subscript i for simplicity. The core of Algorithm 1 lies in accurately estimating f', which can be expressed as

$$f' = \frac{\hat{p} + \delta}{N} Fs, \tag{13}$$

where N is the number of samples, $\hat{p}$ is the index of the maximum value of $|\text{FFT}\{s(n)\}|$ and δ lies in the range $[-0.5, 0.5]$. We aim to obtain an accurate estimation $\hat{\delta}$ of δ. We have the DFT coefficients

$$S_{\pm 0.5} = \sum_{n=0}^{N-1} s(n) e^{-j2\pi \frac{\hat{p} \pm 0.5}{N} n}. \tag{14}$$

Algorithm 1 SFO_Estimation

```
Input: $s(n)$ and $f_i$
Output: SFO
$\hat{p}_1, \hat{p}_2, ..., \hat{p}_K = \text{K\_argmax}(|\,\text{FFT}\{s(n)\}|)$
for $i = 1$ to $K$ do
    $\hat{\delta}_0 = 0$
    for $q = 1$ to $Q$ do
        $S_{\pm 0.5} = \sum_{n=0}^{N-1} s(n) e^{-j2\pi \frac{\hat{p}_i + \hat{\delta}_{q-1} \pm 0.5}{N} n}$
        $\hat{\delta}_q = \frac{1}{2} \text{Real}\{\frac{S_{0.5} + S_{-0.5}}{S_{0.5} - S_{-0.5}}\} + \hat{\delta}_{q-1}$
    end for
    $\hat{f}'_i = \frac{\hat{p}_i + \hat{\delta}_Q}{N} Fs$
end for
$\text{SFO} = \frac{1}{K} \sum_i^K (\frac{\hat{f}'_i - f_i}{f_i})$
return SFO
```

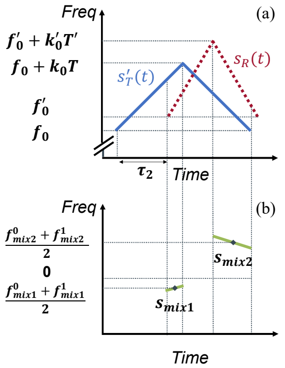

Fig. 5. Doppler frequency shift estimation. (a) Coarsely aligned signal $s_R(t)$ and $s'_T(t)$. They have different chirp rates due to the Doppler effect. (b) Mixing $s_R(t)$ and $\overline{s'_T(t)}$ produces two chirps, s_{mix1} and s_{mix2}, with very low chirp rates. We estimate the speed by analyzing their frequencies.

Substituting the expression of $s(n)$ for $K = 1$ into Eq. (14), we obtain a geometric series, and have

$$S_{\pm 0.5} = Ae^{j\phi} \sum_{n=0}^{N-1} e^{j2\pi \frac{\delta \mp 0.5}{N} n} = Ae^{j\phi} \frac{1 + e^{j2\pi\delta}}{1 - e^{j2\pi \frac{\delta \mp 0.5}{N}}}. \quad (15)$$

For $(\delta \mp 0.5) \ll N$, we can approximate $e^{j2\pi \frac{\delta \mp 0.5}{N}}$ as $1 + j2\pi \frac{\delta \mp 0.5}{N}$ by Taylor series expansion. Eq. (15) becomes

$$S_{\pm 0.5} = -NAe^{j\phi} \frac{1 + e^{j2\pi\delta}}{j2\pi\delta} \frac{\delta}{\delta \mp 0.5} = b \frac{\delta}{\delta \mp 0.5}. \quad (16)$$

So we have $\frac{1}{2} \text{Real}\{\frac{S_{0.5} + S_{-0.5}}{S_{0.5} - S_{-0.5}}\} = \delta$. We substitute estimated δ to Eq. (13) to obtain $\hat{f}'$. The bias resulting from Taylor series expansion approximation from Eq. (15) to Eq. (16) is of order N^{-2}. This bias is minimized from iterative estimating $\hat{\delta}$ in Q times. This algorithm is based on DFT, so we can extend it to $K > 1$ when $f_1, f_2, ..., f_K$ is separated by a distance. Moreover, Algorithm 1 has a time complexity of $\mathcal{O}(KQN \log N)$, where $Q, K \ll N$.

C. Doppler Frequency Shift Estimation

We leverage the triangular chirp for Doppler frequency shift compensation in this step. A complex triangular chirp consists of an up-chirp with increasing frequency followed by a down-chirp with decreasing frequency, which can be expressed as

$$s'_T(t) = \begin{cases} e^{j(2\pi(f_0 t + \frac{1}{2} k_0 t^2))} & \text{if } 0 \le t < T \\ e^{j(2\pi((f_0 + k_0 T)(t-T) - \frac{1}{2} k_0 (t-T)^2))} & \text{if } T \le t < 2T \end{cases}, \quad (17)$$

where f_0 is the starting frequency, k_0 is the up-chirp's chirp rate, and T is the time duration of the up-chirp/down-chirp. In practice, a speaker transmits the real version of $s'_T(t)$, i.e., $s_T(t) = \text{Real}\{s'_T(t)\}$. When the microphone moves with relative speed v, the parameters of received signal $s_R(t)$ will change to

$$\begin{cases} f'_0 = f_0(1 + v/c) \\ T' = T/(1 + v/c) \\ k'_0 = k_0(1 + v/c)^2 \end{cases}. \quad (18)$$

The ToF/delay can be estimated by cross-correlating the $s_R(t)$ and $s_T(t)$. However, the Doppler effect changes the received chirp's parameters and causes a ToF estimation error. We need to estimate v and compensate the Doppler effect. Traditional FMCW radar method [22] can estimate the velocity, but it has a maximum speed limit (e.g., 21.6 cm/s for 40 ms and 20 kHz central frequency acoustic FMCW signal). Speed search-based [12], [16] and FrFT search-based [19] approaches are time-consuming. We propose a novel method to estimate moving speed and ToF by solving a cubic and a linear equation. First, we use an FFT-based matched filter (Cross-Correlation) to estimate the coarse ToF of $s_R(t)$. The match filter's output $R(\tau)$ can be denoted as

$$R(\tau) = \text{F}^{-1}\{\text{Conj}\{\text{F}\{s_T(t)\}\}\,\text{F}\{s_R(t)\}\}, \quad (19)$$

where τ is the time delay, F, F^{-1} and Conj are FFT, IFFT, and complex conjugation operations, respectively. The time complexity of Eq. 19 is $\mathcal{O}(N \log N)$. We search for the coarse delay $\tau_1 = \arg\max(|R(\tau)|)$.

Due to the Doppler effect, τ_1 differs from the real delay [16]. Now, we show how to calculate the accurate time delay. We first align $s_R(t)$ and $s'_T(t)$ by τ_1. Then we mix $s_R(t)$ and $\overline{s'_T(t)}$ (conjugate of $s'_T(t)$) to obtain two chirps s_{mix1} and s_{mix2} with a low chirp rate as shown in Fig. 5. Their starting and ending frequencies f^0_{mix1}, f^1_{mix1}, f^0_{mix2} and f^1_{mix2} are

$$\begin{cases} f^0_{mix1} = -(f_0 + k_0\tau_2) + f'_0 \\ f^1_{mix1} = -(f_0 + k_0 T) + f'_0 + k'_0(T - \tau_2) \\ f^0_{mix2} = -(f_0 + k_0 T - k_0(\tau_2 + T' - T)) + f'_0 + k'_0 T' \\ f^1_{mix2} = -f_0 + (f'_0 + k'_0 T' - k'_0(2T - \tau_2 - T')) \end{cases}, \quad (20)$$

where τ_2 is the remained time offset between $s_R(t)$ and $s'_T(t)$ after aligning them with τ_1. In fact, we cannot directly access f^0_{mix1}, f^1_{mix1}, f^0_{mix2}, and f^1_{mix2} from the FFT result of s_{mix1} and s_{mix2}. These Two mixed signals with very low chirp rates will generate two peaks f^{pk}_{mix1} and f^{pk}_{mix2} in the frequency domain. We obtain $f^{pk}_{mix1} = \frac{f^0_{mix1} + f^1_{mix1}}{2}$ and $f^{pk}_{mix2} = \frac{f^0_{mix2} + f^1_{mix2}}{2}$. By summing the frequencies of the

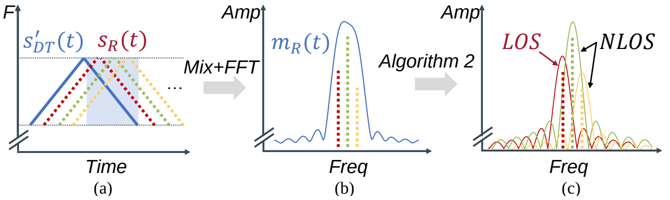

Fig. 6. Multipath interference elimination. (a) The received triangular chirps under three paths. (b) The result after mixing and FFT (blue background part in (a)). There is only one peak as three paths interfere. (c) Our algorithm iteratively reconstructs the real frequency of these three paths.

two peaks and multiplying the result by 2, and subsequently substituting Eq. 20, we obtain

$$2(f^{pk}_{mix1} + f^{pk}_{mix2}) = 4f'_0 - 4f_0 + 3k'_0T' + k_0T' - 3k_0T - k'_0T. \quad (21)$$

Substituting Eq. (18) into Eq. (21), we have

$$2(f^{pk}_{mix1} + f^{pk}_{mix2}) = 4\frac{vf_0}{c} + 3k_0T(1+\frac{v}{c}) \\ + \frac{k_0T}{1+\frac{v}{c}} - 3k_0T - k_0T(1+\frac{v}{c})^2. \quad (22)$$

We eliminate the variable τ_2 by the above method, leaving only v as unknown in Eq. (22). Eq. (22) is a cubic equation of v. We quickly obtain the relative speed v between the transmitter and receiver by solving it.

After solving v, we also have a linear equation of τ_2 as

$$2(f^{pk}_{mix1} - f^{pk}_{mix2}) = 3k_0T(1+\frac{v}{c})^2 - 3k_0T(1+\frac{v}{c}) - 2k_0\tau_2 \\ + k_0T - 2k_0\tau_2(1+\frac{v}{c})^2 - \frac{k_0T}{(1+\frac{v}{c})}, \quad (23)$$

We solve Eq. (23) to obtain τ_2.

D. LOS Estimation

Now, we have a finer estimation of ToF $\tau_1 + \tau_2$ by compensating the Doppler effect. Next, we show how to address multipath interference and calculate the final ToF.

Some works attempt to use deep learning methods to separate mutual interference between multiple signals, yet deep learning-based methods require significant computational resources and are susceptible to environmental changes [13], [32]–[36]. Other Model-based methods, e.g., MUSIC, have also been used to split multipath [3]. It requires knowing an accurate number of paths and has $\mathcal{O}(N^3)$ time complexity.

We only care about the arrival time of the LOS path. Signal traveling through the LOS path will arrive first, followed by NLOS. So, We iteratively reconstruct LOS and close NLOS to restore the accurate delay of the LOS signal.

Our multipath interference elimination workflow is shown in Fig. 6. We substitute v to Eq. 18 and use produced f'_0, T', k'_0 to construct a transmitted signal $s'_{DT}(t)$ under the Doppler effect. Next, we align $s_R(t)$ with $s'_{DT}(t)$ using the estimated delay of $\tau_1 + \tau_2$. Once aligned, we mix complex conjugate of $s'_{DT}(t)$ and $s_R(t)$ to produce single-tones $m_R(t)$. Then we apply Algorithm 2 to $m_R(t)$ to calculate LOS's delay τ_3.

Algorithm 2 LOS_Estimation

```
Input: $m_R(t)$, chirp rate $k'_0$ of $s'_{DT}(t)$
Output: LOS delay $\tau_3$
1: Initialize adaptive ratio $R$, stop criteria $E$.
2: $\hat{s} = m_R(t)$, $i = 1$, $h = [\,]$
3: repeat
4:     $e_{curr} = INT\_MAX$ /*A large value*/
5:     repeat
6:         $f_i, a_i, \phi_i = \arg\max(|\,\mathrm{FFT}\{\hat{s} - h[1:i-1]\}|)$
7:         $h[i] = a_i \exp(j(2\pi f_i t + \phi_i))$
8:         for $l = 1$ to $i - 1$ do
9:             $f_l, a_l, \phi_l = \arg\max(|\,\mathrm{FFT}\{\hat{s} - h[1:l-1] - h[l+1:i]\}|)$ /*Refine the estimated single-tones*/
10:            $h[l] = a_l \exp(j(2\pi f_l t + \phi_l))$
11:        end for
12:        $e_{pre} = e_{curr}$
13:        $e_{curr} = \sqrt{\sum_t (\hat{s} - h[1:i])^2}$
14:    until $|e_{curr} - e_{pre}| < E$
15:    $i = i + 1$
16: until $|a_{i-1}| < R \cdot \max(|a|)$
17: return $\frac{\min(f)}{k'_0}$
```

In Algorithm 2, the adaptive ratio R, is used as a threshold to stop our algorithm. E controls the number of iterations that single-tones' parameters are updated. The algorithm iteratively estimates the parameters of potential single-tone signals in line 6. This estimation is performed by canceling estimated single-tones and then applying FFT to estimate the remaining strongest single-tone. After estimating one single-tone, we refine previously estimated single-tones in line 9. We repeat this refining process until the power change of the estimated single-tones below E in line 14. Next, we start the new estimating and refining steps until the next estimated single-tone's power is below a threshold controlled by R in line 16. Finally, we choose the minimum estimated frequency as LOS's frequency because LOS arrives first. This frequency is finally mapped to delay $\tau_3 = \frac{\min(f)}{k'_0}$.

Our adaptive LOS reconstruction algorithm has several advantages. First, there is no need to know the number of multipaths beforehand. Our algorithm monitors the power of single-tones around the LOS. It stops once the next reconstructed single-tone's power is below a threshold controlled by R in line 16. Second, our method can solve the peaks merging problem shown in Fig. 6.(b). Third, when signals' sidelobes add up, there may be a fake peak with considerable height before LOS. Our method can eliminate these fake peaks by iteratively removing the main peaks. Last, the time complexity of Algorithm 2 is $\mathcal{O}(MN\log N)$, where M is the FFT called times, and $M \ll N$.

E. Tracking, Ranging, and Positioning

Combining all the individual steps, Algorithm 3 presents the pseudo-code for our final system. The output d is used for tracking, ranging, and positioning.

Algorithm 3 ATP system

```
Input: recorded signal $s(n)$, transmitted single-tones' frequencies $f$, transmitted triangular FMCWs' parameters $f_0, T, k_0$, working time $T$, and sound speed $c$
Output: distance $d$
SFO = SFO_Estimation($s(n)$, $f$)
$\tau_1, \tau_2, v$ = DFS_Estimation($s(n), f_0, T, k_0$)
Construct $f'_0, T', k'_0, s'_{DT}(t)$ using $v$. Mix Conj$\{s'_{DT}(t)\}$ and $s_R(t)$ to obtain $m_R(t)$.
$\tau_3$ = LOS_Estimation($m_R(t)$, $k'_0$)
$d = (\tau_1 + \tau_2 + \tau_3 - \text{SFO} \cdot T)c$
```

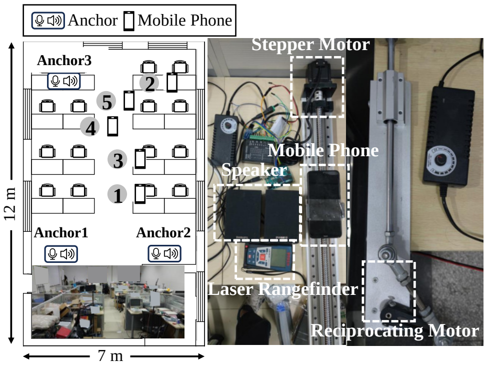

Fig. 7. Floor map of experiment environment and experiment devices.

IV. Implementation

We implement ATP in the remote mode on the Android platform based on LibAS framework [37]. LibAS is a cross-platform framework for developing acoustic sensing apps. The recorded audio in Android is sent to a PC in real time. We perform signal processing using MATLAB on a PC with AMD Ryzen 7 5800H CPU, and the results are sent back to mobile phones in real time. In the SFO estimation stage, we use 1 s single-tones ranging from 18 to 22 kHz with a 1 kHz interval for all methods. In the tracking stage, ATP uses 40 ms triangular FMCW from 18 kHz to 22 kHz. Other FMCW-based methods use the same FMCW and a 23 kHz single-tone to compensate for the Doppler effect. Single-tone-based methods use the same single-tons as used in the SFO estimation stage. For all FFT operations, we pad the signals to 20 times their original length with zeros. We use $Q = 3, K = 5$ in Algorithm 1, and $E = 10^{-3}, R = 0.3$ in Algorithm 2. We implement ATP for tracking, ranging, and positioning based on relative distance change [5], two-way ranging [11], and trilateration [38].

V. Evaluation

A. Evaluation Setup

The experiment environment and devices are shown in Fig. 7. We conducted our experiments in an office environment with normal working activities and environmental noise. We have conducted experiments including SFO estimation, 1-D tracking with different durations, speeds, distances, and noise levels, 2-D tracking, ranging, and positioning. We ensure LOS between speakers and microphones because sound waves hardly penetrate obstacles. We use a stepper motor to control the receiver's movement precisely at sub-millimeter levels. To track movement at varying speeds, we use a reciprocating motor with a fixed 15 cm moving range and a speed range of 0 to 100 cm/s. We use a laser distance meter with sub-millimeter precision to obtain ground-truth measurements for ranging and localization.

TABLE I
Solving multipath time consumption

Algo 2	MUSIC (M=40)	MUSIC (M=80)	MUSIC (M=160)
8.0 ms	12.6 ms	41.9 ms	172.2 ms

B. Comparison

We compare our method, ATP, with the existing approaches.

- AAMouse [10]: an acoustic tracking method by measuring single-tone's frequency change.
- LLAP [4]: an acoustic tracking method by measuring single-tone's phase change.
- Vernier [1]: an acoustic tracking method by measuring single-tone's phase change without mixing step.
- CAT [5]: an acoustic tracking method by mapping mixed signal's frequency to distance.
- BeepBeep [38]: an acoustic ranging and positioning method by Cross-Correlation. We use the FMCW signal.
- BeepBeep-GCC-PHAT [15]: an improved version of BeepBeep using generalized cross-correlation phase transform (GCC-PHAT) instead of Cross-Correlation.
- PDF [2]: an acoustic tracking method by calculating mixed signal's frequency based on phase without FFT and then mapping this frequency to distance.
- RABIT [3]: an acoustic tracking method utilizing MUSIC to estimate mixed signal's frequency and then mapping this frequency to distance. We set RABIT's auto-correlation order M to 40 to achieve the online working goal according to the time consumption shown in Table.I.

We further compare ATP with ATP$^{|}$, ATP†, and ATP‡.

- ATP$^{|}$: ATP without SFO estimation and compensation.
- ATP†: ATP without DFS estimation and compensation.
- ATP‡: ATP without LOS estimation and reconstruction.

It should be noted that AAMouse, LLAP, and Vernier are based on single-tone and can only be used for tracking. In contrast, CAT, BeepBeep, BeepBeep-GCC-PHAT, PDF, and RABIT approaches are based on FMCW and can be used in tracking, ranging, and positioning.

C. Sampling Frequency Offset Estimation

We first measure the SFO estimation accuracy. We set one device to transmit a 20 kHz single-tone and another to record the signal. We vary the distance between them.

The accuracy of SFO estimation depends on the precise calculation of the received single-tone frequency. Thus, we

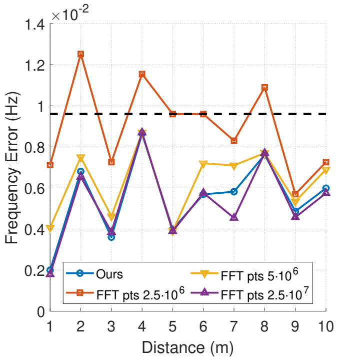

Fig. 8. SFO Estimation accuracy. The black dash line marks a 1 cm/min tracking error.

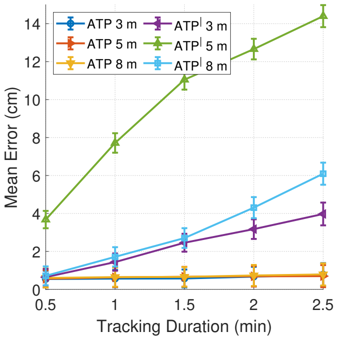

Fig. 9. Long-time tracking.

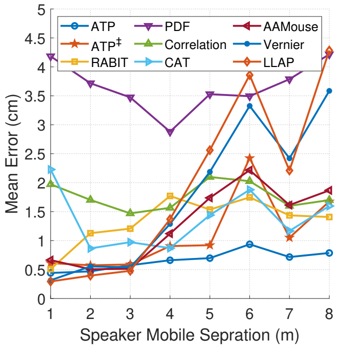

Fig. 10. 1-D tracking for different separations.

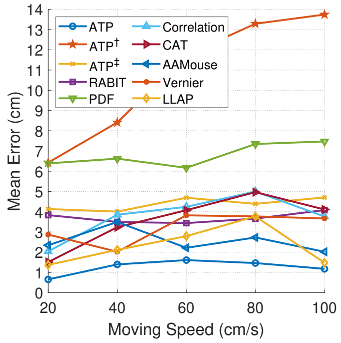

Fig. 11. 1-D tracking for different speeds.

TABLE II
SFO ESTIMATION TIME AND MEMORY CONSUMPTION

Algo 1	$2.5 \cdot 10^6$ pts FFT	$5 \cdot 10^6$ pts FFT	$2.5 \cdot 10^7$ pts FFT
1.8 ms	187.5 ms	242.5 ms	1893.5 ms
16 kb	39140 kb	78284 kb	391388 kb

compare the frequency estimation error of Algorithm 1 to that of FFT with zero-padding in Fig. 8. Our method's mean frequency estimation error is 0.0055 Hz, while FFT method with zero-padding of lengths $2.5 \cdot 10^6$, $5 \cdot 10^6$, and $2.5 \cdot 10^7$ has mean errors of 0.009, 0.0063, and 0.0053 Hz, respectively. With a longer length of zero padding, FFT has less error in frequency estimation but higher computation overhead. We further examine the time and memory consumption for different methods, as shown in Table.II. To achieve equivalent accuracy, FFT with $2.5 \cdot 10^7$ points requires 1052× more time and 24462× more memory than our method.

We further conduct a long-time 1-D tracking experiment, separating the speaker and microphone at 3 m, 5 m, and 8 m, as shown in Fig 9. ATP's tracking error remains relatively consistent across varying tracking durations. While ATP$^{|}$'s tracking error increases linearly over time because of SFO. The tracking error of ATP is less than 0.78 cm, whereas ATP$^{|}$ can have a tracking error of up to 14.4 cm when tracking for 2.5 minutes. ATP$^{|}$ has a larger error in 5 m than 8 m because SFO causes a larger error than an increase in distance. Our SFO estimation and compensation method effectively minimizes the accumulation of tracking errors over time.

D. 1-D Tracking Accuracy

In this experiment, we measure the 1-D tracking error. We vary the distance between the microphone and the speaker. The moving distance is 20 cm, and the moving speed is 2.6 cm/s, resulting in a negligible impact from the Doppler effect. Fig. 10 shows that ATP has an error under 0.57 cm in 3 m and an error under 0.94 cm in 8 m, respectively. ATP is better than all others as it overcomes multipath interference through Algorithm. 2. ATP‡ is impacted by multipath, resulting in a 2.6× higher error rate than ATP. As the distance increases, the Signal-to-noise ratio (SNR) decreases, and the multipath's impact strengthens. So, the tracking error of all methods increases. Within a 3 m range, single-tone-based methods AAmouse, LLAP, and Vernier typically show errors under 0.66 cm, similar to ATP and consistent with their respective research papers [1], [4], [10]. This is because there is typically a lighter influence from multipath at close distances. When tracking at ranges greater than 3 m, the performance of single-tone-based methods degrades significantly. This is because there is increasing interference from multipath signals, and these methods are inherently vulnerable to multipath. PDF can work well in clean environments. However, its performance is poor for varying distances in the presence of multipath interference, as it employs phase and time differences to calculate frequency, making it vulnerable to noise and interference. The performance of the MUSIC-based RABIT method is similar to the FFT-based CAT method, as the auto-correlation order M is limited due to the $\mathcal{O}(N^3)$ time complexity of MUSIC. CAT and Correlation methods also suffer from multipath, and their tracking error increases.

The significant errors observed for all methods at 6 m suggest that multipath interference is more significant at that distance, potentially overshadowing the effects of distance variation.

E. 1-D Tracking With Different Speeds

In this experiment, we show the tracking error at different moving speeds. We set the distance between the microphone and the speaker to 5 m and control the moving speed of the speaker. Fig. 11 shows the result. ATP has a mean error of 1.3 cm, which is better than other methods as it overcomes the Doppler effect and multipath interference by our speed estimation method and LOS reconstruction algorithm. ATP† without DFS estimation has an 8.5× higher error and ATP‡ without LOS estimation has a 3.5× higher error than ATP. The error of different methods is stable to varying speeds because all those methods consider the Doppler effect: (1) single-tone-based approaches leverage the Doppler effect for tracking, (2) Other FMCW-based methods utilize single-tone

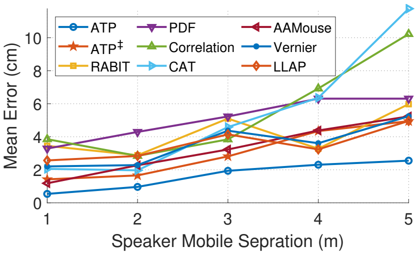

Fig. 12. 2-D tracking accuracy.

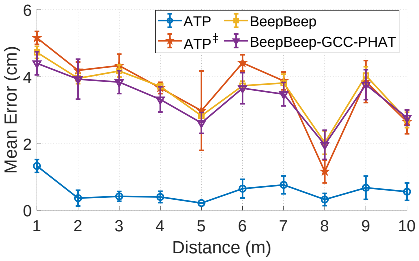

Fig. 13. Ranging accuracy.

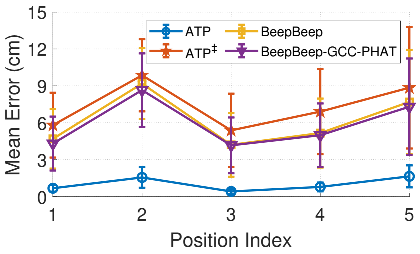

Fig. 14. Positioning accuracy.

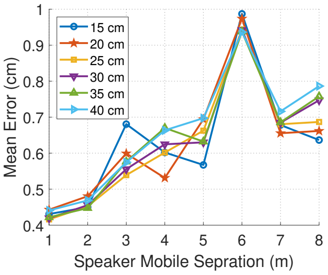

Fig. 15. tracking with different moving distance accuracy.

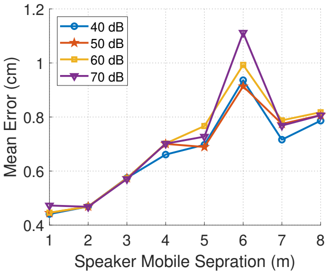

Fig. 16. tracking with noise accuracy.

for compensating the Doppler effect, and (3) ATP compensates Doppler effect by triangular FMCW. Our triangular FMCW-based Doppler best compensates the Doppler effect among those methods, especially in multipath environments. What's more, it does not need to send additional single-tones.

F. 2-D Tracking Accuracy

In this experiment, we measure the 2-D tracking error. We separate two speakers by 50 cm. We move the receiver by 5 cm from each distance. Fig. 12 shows the tracking error. Our novel Doppler estimation and LOS reconstruction methods result in ATP having a mean error of 1.6 cm, and this error is better than that of all other methods. ATP is also 1.7× better than ATP‡. The relationship between tracking errors for different methods is similar to that in 1D tracking.

G. Ranging Accuracy

We compare our method with ATP‡, BeepBeep, and GCC-PHAT-based BeepBeep. The result is shown in Fig. 13. ATP has an average ranging error of 0.56 cm. ATP‡, BeepBeep, and BeepBeep-GCC-PHAT have an average error of 4.1 cm, 3.6 cm, and 3.4 cm, respectively. ATP overcomes multipath interference, resulting in a better result.

H. Positioning Accuracy

We implement a trilateration-based positioning system by placing three anchors. Then, we place a smartphone in five locations, as depicted in Fig. 7. The accuracy of trilateration relies on the accuracy of ranging. ATP with better ranging accuracy has a positioning error of 1.0 cm, as shown in Fig. 14. ATP‡, BeepBeep, and BeepBeep-GCC-PHAT have a positioning error of 7.4 cm, 5.9 cm, and 6.2 cm, respectively.

I. Impact Of Moving Distance On Tracking Error

We measure the 1-D tracking error of different moving distances for ATP. As shown in Fig. 15, ATP's tracking error remains relatively consistent with different moving distances. It is because our FMCW-based tracking method produces each distance based on ToF. Each estimated distance is independent, so there is no cumulative error.

J. Impact Of Noise Intensity On Tracking Error

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). The result is shown in Fig. 16. We can see that the error increases slightly as the noise level increases. It remains small for all distances.

K. Latency

The average time consumption for SFO estimation, DFS estimation, LOS estimation, and other components is 1.8 ms, 3.1 ms, 8 ms, and 0.9 ms, respectively. Since SFO estimation is performed only once initially, ATP's latency is $3.1+8+0.9 = 12$ ms when processing a 40 ms triangular FMCW, enabling it to work online.

VI. Conclusion

We presented ATP, an efficient and accurate tracking, ranging, and positioning approach. We theoretically and experimentally analyze the limitations of prior arts. We overcome the fundamental challenges of existing approaches, including multipath interference, Doppler frequency shift, and sampling frequency offsets. We implement our prototype on smartphones and conduct extensive experiments to evaluate the performance of ATP. The result shows that ATP can achieve mean accuracy of 0.66 cm, 0.56 cm, and 1.0 cm in tracking, ranging, and positioning, respectively. We believe ATP's design can support various mobile applications, from video gaming to VR and AR.

Acknowledgment

This work is in part supported by National Key R&D Program of China 2022YFC3801300, National Natural Science Foundation of China (U22A2031, No. 61932013, 62172250).

References

[1] Yunting Zhang, Jiliang Wang, Weiyi Wang, Zhao Wang, and Yunhao Liu. Vernier: Accurate and fast acoustic motion tracking using mobile devices. In *Proceedings of IEEE INFOCOM*, 2018.

[2] Haiming Cheng and Wei Lou. Push the limit of device-free acoustic sensing on commercial mobile devices. In *Proceedings of IEEE INFOCOM*, 2021.

[3] Wenguang Mao, Zaiwei Zhang, Lili Qiu, Jian He, Yuchen Cui, and Sangki Yun. Indoor follow me drone. In *Proceedings of ACM MobiSys*, 2017.

[4] Wei Wang, Alex X Liu, and Ke Sun. Device-free gesture tracking using acoustic signals. In *Proceedings of ACM MobiCom*, 2016.

[5] Wenguang Mao, Jian He, and Lili Qiu. Cat: high-precision acoustic motion tracking. In *Proceedings of ACM MobiCom*, 2016.

[6] Anran Wang and Shyamnath Gollakota. Millisonic: Pushing the limits of acoustic motion tracking. In *Proceedings of ACM CHI*, 2019.

[7] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of ACM CHI*, 2016.

[8] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. Strata: Fine-grained acoustic-based device-free tracking. In *Proceedings of ACM MobiSys*, 2017.

[9] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A Cunefare, Omer T Inan, and Gregory D Abowd. Soundtrak: Continuous 3d tracking of a finger using active acoustics. *Proceedings of ACM IMWUT*, 1(2), 2017.

[10] Sangki Yun, Yi-Chao Chen, and Lili Qiu. Turning a mobile device into a mouse in the air. In *Proceedings of ACM MobiSys*, 2015.

[11] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of ACM SenSys*, 2007.

[12] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proceedings of ACM MobiSys*, 2012.

[13] Wenguang Mao, Wei Sun, Mei Wang, and Lili Qiu. Deeprange: Acoustic ranging via deep learning. *Proceedings of ACM IMWUT*, 4(4), 2020.

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

[15] Viktor Erdélyi, Trung-Kien Le, Bobby Bhattacharjee, Peter Druschel, and Nobutaka Ono. Sonoloc: Scalable positioning of commodity mobile devices. In *Proceedings of ACM MobiSys*, 2018.

[16] Weiguo Wang, Luca Mottola, Yuan He, Jinming Li, Yimiao Sun, Shuai Li, Hua Jing, and Yulei Wang. Micnest: Long-range instant acoustic localization of drones in precise landing. In *Proceedings of ACM SenSys*, 2022.

[17] Kaikai Liu, Xinxin Liu, and Xiaolin Li. Guoguo: Enabling fine-grained smartphone localization via acoustic anchors. *IEEE Transactions on mobile computing*, 15(5), 2015.

[18] Hongzi Zhu, Yuxiao Zhang, Zifan Liu, Shan Chang, and Yingying Chen. Hyperear: Indoor remote object finding with a single phone. In *Proceedings of IEEE ICDCS*, pages 678–687, 2019.

[19] Lei Zhang, Minlin Chen, Xinheng Wang, and Zhi Wang. Toa estimation of chirp signal in dense multipath environment for low-cost acoustic ranging. *IEEE Transactions on Instrumentation and Measurement*, 68(2), 2018.

[20] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3), 1986.

[21] Dong Li, Shirui Cao, Sunghoon Ivan Lee, and Jie Xiong. Experience: practical problems for acoustic sensing. In *Proceedings of ACM Mobicom*, 2022.

[22] Sandeep Rao. Introduction to mmwave sensing: Fmcw radars. *Texas Instruments (TI) mmWave Training Series*, 2017.

[23] Jau-Jr Lin, Yuan-Ping Li, Wei-Chiang Hsu, and Ta-Sung Lee. Design of an fmcw radar baseband signal processing system for automotive application. *SpringerPlus*, 5, 2016.

[24] Kun Qian, Chenshu Wu, Fu Xiao, Yue Zheng, Yi Zhang, Zheng Yang, and Yunhao Liu. Acousticcardiogram: Monitoring heartbeats using acoustic signals on smart devices. In *Proceedings of IEEE INFOCOM*. IEEE, 2018.

[25] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing*, 21(5), 2020.

[26] Yi Zhang, Weiying Hou, Zheng Yang, and Chenshu Wu. Vecare: Statistical acoustic sensing for automotive in-cabin monitoring. In *Proceedings of USENIX NSDI*, 2023.

[27] Mingshi Chen, Panlong Yang, Jie Xiong, Maotian Zhang, Youngki Lee, Chaocan Xiang, and Chang Tian. Your table can be an input panel: Acoustic-based device-free interaction recognition. *Proceedings of ACM IMWUT*, 3(1), 2019.

[28] Xiangyu Xu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Linghe Kong, and Minglu Li. Breathlistener: Fine-grained breathing monitoring in driving environments utilizing acoustic signals. In *Proceedings of ACM MobiSys*, 2019.

[29] Charles E. Cook. Linear fm signal formats for beacon and communication systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-10(4):471–478, 1974.

[30] Linsong Cheng, Zhao Wang, Yunting Zhang, Weiyi Wang, Weimin Xu, and Jiliang Wang. Acouradar: Towards single source based acoustic localization. In *Proceedings of IEEE INFOCOM*, 2020.

[31] Elias Aboutanios and Bernard Mulgrew. Iterative frequency estimation by interpolation on fourier coefficients. *IEEE Transactions on signal processing*, 53(4), 2005.

[32] Zheng Yang, Yi Zhang, Kun Qian, and Chenshu Wu. Slnet: A spectrogram learning neural network for deep wireless sensing. In *Proceedings of USENIX NSDI*, 2023.

[33] Yetong Cao, Chao Cai, Anbo Yu, Fan Li, and Jun Luo. Earace: Empowering versatile acoustic sensing via earable active noise cancellation platform. *Proceedings of ACM IMWUT*, 7(2), 2023.

[34] Li Zhang, Jinhui Bao, Yi Xu, Qiuyu Wang, Jingao Xu, and Danyang Li. From coarse to fine: Two-stage indoor localization with multisensor fusion. *Tsinghua Science and Technology*, 28(3):552–565, 2023.

[35] En Wang, Mijia Zhang, Bo Yang, Yongjian Yang, and Jie Wu. Large-scale spatiotemporal fracture data completion in sparse crowdsensing. *IEEE Transactions on Mobile Computing*, 2023.

[36] Fei HUANG, Guangxia LI, Haichao WANG, Shiwei TIAN, YANG Yang, and Jinghui CHANG. Navigation for uav pair-supported relaying in unknown iot systems with deep reinforcement learning. *Chinese Journal of Electronics*, 31(3):416–429, 2022.

[37] Yu-Chih Tung, Duc Bui, and Kang G Shin. Cross-platform support for rapid development of mobile acoustic sensing applications. In *Proceedings of ACM MobiSys*, 2018.

[38] Chunyi Peng, Guobin Shen, and Yongguang Zhang. Beepbeep: A high-accuracy acoustic-based system for ranging and localization using cots devices. *ACM Transactions on Embedded Computing Systems*, 11(1), 2012.