Fine-grained Multi-user Device-Free Gesture Tracking on Today's Smart Speakers

Ningzhi Zhu
ShanghaiTech University
zhunzh@shanghaitech.edu.cn

Huangxun Chen
Huawei Theory Lab
chen.huangxun@huawei.com

Zhice Yang
ShanghaiTech University
yangzhc@shanghaitech.edu.cn

***Abstract*—Smart speakers play an important role in smart home envision. Active acoustic sensing can enable convenient gesture interaction on smart speakers to complement voice interaction in mandatory quiet scenarios and address privacy concerns. However, existing solutions did not consider the impact of the widely adopted uniform circular geometry of commercial smart speakers on gesture tracking. To fill this gap, we propose SparseTrack to achieve fine-grained multi-user device-free gesture tracking on commercial smart speakers. We cast gesture tracking to sparse recovery intuition to address signal coherence issue on uniform circular mic-array. We then synthesize wideband measurement to eliminate spatial ambiguity caused by the insufficient spatial sampling rate of today's smart speakers in the ultrasonic frequency band. We further design a robust trace extraction approach and properly handle the impact of the doppler effect on gesture tracking. We implement SparseTrack on COTS circular mic-array and conduct extensive evaluations. The results show that our system can simultaneously track up to 4 users' gestures with a mean tracking error of 2.66 cm.**

***Index Terms*—Acoustic Sensing, Device-free, Smart Speaker.**

I. Introduction

Smart speakers play an essential role in the envisions of smart space including home, office, ward, etc. They act as the control hub to acquire user's commands, analyze user's intent, and offer smart services. To fully achieve this intelligence, it is indispensable to empower them with convenient interaction approaches. Voice/speech recognition is the major interaction way of current commercial smart speakers. However, mandatory quiet areas such as the living room with a sleeping baby and hospital wards may limit the usage of voice interaction. In addition, voice interaction has been criticized for privacy issues [1].

To this end, researchers tried to repurpose the smart speaker as an active sonar to track the gestures of nearby users to provide another way of control and interaction. Recent efforts [2] have demonstrated the feasibility of a non-uniform linear microphone array (mic-array). But compared with linear array's 180° azimuthal coverage, circular array's 360° coverage is preferred in voice picking. The dominating array layout of most commercial products (see Table I) is the Uniform Circular mic-Array (UCA). However, the UCA layout poses significant challenges in multi-user tracking and also excludes most existing solutions that are dedicated to the linear layout.

Firstly, device-free gesture tracking relies on signals reflected by the targets as shown in Fig. 1. Signals reflected by different targets are attenuated and delayed copies of the signal emitted by the speaker, *i.e.*, the reflections are coherent signals.

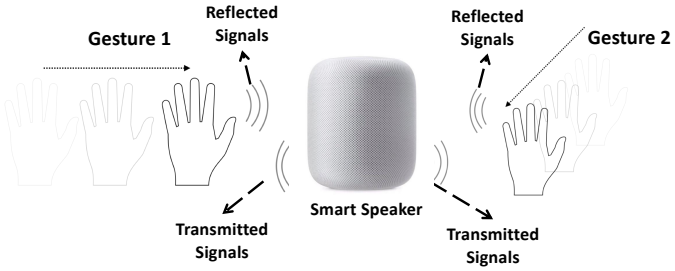


Fig. 1: **Gesture Tracking on Smart Speaker.** The smart speaker tracks the surrounding activities, *e.g.*, gesture commands from one or multiple users, through analyzing the reflected acoustic signals like an active sonar.

Product	Mic Layout	Mic Spacing
Amazon Echo	6-mic uniform circular array	4.96 cm
Amazon Echo Dot	4-mic uniform circular array	7.00 cm
Apple Homepod	6-mic uniform circular array	7.10 cm
Sonos One	6-mic uniform circular array	5.99 cm

TABLE I: Mic-array Specifications of Smart Speakers.

Signal coherence degrades the performance of subspace-based direction and location tracking methods, *e.g.*, MUSIC [3], [4]. Previous work based on MUSIC [2], [5] performs the spatial smoothing operation on the linear array to decorrelate the signals [6]. However, unlike linear arrays, UCA lacks the orientational invariant structure [7], hence it is still an open issue to handle signal coherence under the constraints of smart speakers: aperture size and the number of microphones [8].

Secondly, the microphones of today's smart speakers are usually spaced for several centimeters (see Table I) to increase the aperture size for better spatial resolution in picking human voice. However, active acoustic sensing usually emits inaudible ultrasound (17-23 kHz) to prevent annoyance. As the ultrasound wavelength (1.5-2 cm) is much smaller than the actual spacing of these smart speaker mic-arrays, the spatial sampling rate is not sufficient, which leads to the spatial aliasing analogous to the case of insufficient temporal sampling [9]. Existing work [2] adopts the non-uniform array to overcome this problem, but most commercial mic-array has uniform geometry. Without a sufficient spatial sampling rate, gesture tracking techniques will encounter severe ambiguity issues and performance degradation.

To practically enable gesture tracking on today's smart speakers, we propose a fine-grained gesture tracking system, SparseTrack, which has no microphone geometry assumption

and is fully compatible with commercial UCAs. It works harmonically with coherent signals and insufficient spatial sampling rate in the ultrasound band.

Specifically, signal coherence is a well-known headache for subspace-based sensing methods. Instead of countering the inherent defect of the estimation technique itself, we argue to pursuit a better cure by adopting more compatible techniques. Our approach is based on an important observation on reflector sparsity, *i.e.*, the number of significant moving reflectors that could have contributed to the overall reflected signal is limited, which enlightens us to treat gesture tracking as a sparse recovery problem. Therefore, we rigorously model gesture tracking on commodity circular mic-array from the perspective of sparse recovery to achieve accurate movement tracking.

To address the ambiguity issue caused by the insufficient spatial sampling rate, we observe that the ambiguous positions are related to the frequency of the acoustic sensing signals. Thus, we actively emit wideband OFDM signals and synthesize the measurements of different frequency components to make the true position more pronounced than ambiguous ones. In addition, we develop a robust approach to extract gesture traces from noisy measurements. We further design a velocity-aware scheme to account for the Doppler effect when the gesture is performed at high speed.

The contribution of this work is summarized as follows:

- We identify the incompatibility between the state-of-the-art gesture tracking solutions and today's commercial smart speakers, reveal the key challenges on signal coherence and spatial sampling rate, and propose effective countermeasures.
- We formulate the device-free gesture tracking as a sparse recovery problem and propose SparseTrack to realize fine-grained multi-user device-free gesture tracking on commodity smart speakers.
- We implement SparseTrack on COTS circular mic-array and conduct extensive evaluations. The results show that our system can simultaneously track up to 4 users' gestures with a mean tracking error of 2.66 cm.

II. Related Work

In this section, we summarize recent efforts on gesture tracking including acoustic-based and RF-based approaches.

Device-free Acoustic Sensing. There have been many early works [10]–[12] supporting the classification of a pre-defined set of gestures using active acoustic sensing. SilentSign [13] uses acoustic sensing and verification model to enable convenient signature verification in smart devices. In terms of acoustic-based gesture tracking, many previous works focus on near-device interaction. FingerIO [14] can enable a mobile phone to track the nearby finger with an average accuracy of 8 mm. VSkin [15] supports capturing finger movements on the back of a mobile phone. Both LLAP [16] and Strata [17] resort to the phases of audio signals to obtain finer resolution. These works focus on tracking a single reflector. Recently, RTrack [2] enables multi-user tracking by integrating a non-uniform linear mic-array, 2D MUSIC algorithm, and an RNN-based neural network. However, circular array's 360° azimuthal coverage is more preferred in voice picking and widely adopted on commercial products. Motivated by this trend, our work aims to fill the gap between the circular mic-array geometry of smart speakers and existing multi-user tracking solutions. We enable multi-user 2D gesture tracking on a uniform circular mic-array without the training burden.

Device-free RF Sensing. Besides acoustic signals, RF signals are also extensively exploited as the sensing medium. WiFi routers and software-defined radios (SDRs) are two common platforms for RF sensing. WiSee [18], WiAG [19], and Widar [20] support the classification of a pre-defined set of gestures using active RF sensing. WiTrack [21] utilizes wideband FMCW emitted by a customized SDR to achieve decimeter-level device-free trajectory tracking. It is worth mentioning that sparse representation has also been adopted in RF sensing. ROArry [22] accurately localizes a target device even under low SNRs by casting DOA estimation into a sparse recovery problem. However, our target is the device-free tracking scenario, and we need to cope with severe signal coherence. WiDeo [23] shows that the RF backscatter sensing can be formulated using sparsity and compressive sensing intuition, and achieves a 7 cm median error on tracking moving persons. Compared with WiDeo, we focus on acoustic signals, which expose distinct behaviors as the RF signals, such as insufficient spatial sampling rate, the Doppler effect, *etc.*

III. Problem Formulation

A. Signal Model of Uniform Circular Mic-array

We give a mathematical signal model which paves the way to design device-free tracking on today's smart speakers.

1) Basics of Device-free Tracking on Smart Speaker: Fig. 2 shows three major components of a smart speaker: a speaker array, a microphone array consisting of several microphones, and a micro-controller. Device-free gesture tracking generally starts with active signal emission by the speaker array, thereupon reflection signals collection by the mic-array, finally reflector localization and trajectory extraction by the micro-controller.

The basic principle of device-free tracking is that distributed microphones receive differently delayed versions of the emitted signal, which can be parameterized by the distance and orientation of reflectors with respect to the mic-array. Assuming transmitted signals is $s_0(t)$, D reflectors exist and i-th reflector is located at different distance d_i and angle-of-arrival (AoA) θ_i *w.r.t.* the smart speaker with L microphones. The signal $x_k(t)$ arriving back at the k-th microphone is $x_k(t) = \sum_{i=1}^{D} f_i(s_0(t), d_i, \theta_i)$, where $f_i(\cdot)$ characterizes the distortion applied by i-th reflector. The goal of device-free tracking is to estimate reflectors' location parameters $[(d_i, \theta_i), i = 1, 2, ..., D]$ that compose the mic-array measurements $X = [x_0(t), x_1(t), ..., x_{L-1}(t)]^T$.

2) Signal Model on Uniform Circular Mic-array: Distinct from recent modeling efforts on linear mic-array [2], we choose to model device-free tracking on uniform circular mic-array rigorously, so that most commercial smart speakers

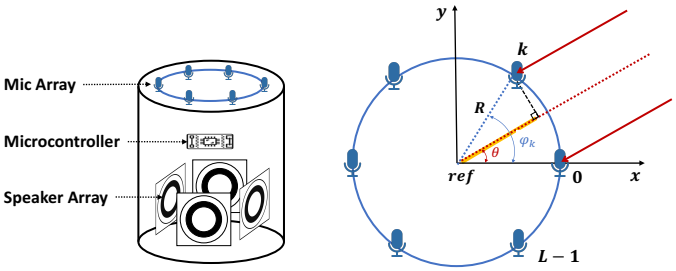


Fig. 2: **Typical Architecture of a Smart Speaker and the Signal Model of the Uniform Circular mic-Array (UCA).** Commercial smart speakers adopt UCA to achieve uniform spatial coverage. Our signal modeling focuses on UCA, but in fact, it can be extended to arbitrary mic geometry layout.

enumerated in Table I can benefit from our proposed design instantly. As shown in Fig. 2, given a circular mic-array with radius R and L equally-distributed microphone elements, the 0-th element is located at the x-axis of the coordinate frame, and the coordinate origin is treated as the reference point `ref`. Thus, the angle between x-axis and k-th element $\varphi_k = 2\pi k/L$ and the coordinates of k-th element $P_k = [R\cos\varphi_k, R\sin\varphi_k]$.

For clarity, we will first illustrate the model of single reflector localization, followed by the multi-reflector version. Given a reflector at distance d_i and AoA θ_i *w.r.t.* `ref`, we can first obtain the `ref` phase $\phi_{ref} = 2\pi f\frac{2d_i}{c}$, where c denotes the sound velocity and f denotes the frequency of emitted signal $s_0(t)$. Then, the phase offset between k-th element and the `ref` under circular mic-geometry can be parameterized by AoA θ_i: $\Delta\phi_k = 2\pi f\Delta d_k/c = 2\pi fR\cos(\varphi_k - \theta_i)/c$, where Δd_k denotes the signal path difference between k-th mic and `ref` and is depicted intuitively by the yellow thick line in Fig. 2. Therefore, the signal phase at k-th element is $\phi_k(d_i,\theta_i) = \phi_{ref} + \Delta\phi_k = 2\pi f(2d_i - R\cos[2\pi k/L - \theta_i])/c$. Thus, given a reflector i with (d_i,θ_i), we have:

$$\begin{aligned} X &= [x_0(t), x_1(t), ..., x_{L-1}(t)]^T \\ &= [e^{-j\phi_0(d_i,\theta_i)}, ..., e^{-j\phi_{L-1}(d_i,\theta_i)}]^T c_i s_0(t) \\ &= a(d_i,\theta_i)c_i s_0(t), \end{aligned} \tag{1}$$

where $a(d_i,\theta_i) = [e^{-j\phi_0(d_i,\theta_i)}, ..., e^{-j\phi_{L-1}(d_i,\theta_i)}]^T$ denotes the steering vector for reflector i, and constant c_i represents the propagation and scattering attenuation of reflection signals from reflector i.

Generalizing it to the scene with D reflectors with $(d_i,\ \theta_i)$, $i = 1, ..., D$ *w.r.t.* `ref`, we have:

$$\begin{aligned} X &= [\sum_{i=1}^{D} c_i e^{-j\phi_0(d_i,\theta_i)}, ..., \sum_{i=1}^{D} c_i e^{-j\phi_{L-1}(d_i,\theta_i)}]^T s_0(t) \\ &= [a(d_1,\theta_1), ..., a(d_D,\theta_D)] \cdot [c_1, ..., c_D]^T s_0(t), \end{aligned} \tag{2}$$

If we define the steering matrix and signal vector as $A = [a(d_1,\theta_1), ..., a(d_D,\theta_D)]$ and use $S = [c_1, ..., c_D]^T s_0(t) = [s_1(t), ..., s_D(t)]^T$ to absorb the whole attenuation effects on transmitted signals, and further consider the noise $\mathcal{N}$, then the signal model of multi-reflectors localization is:

$$X = A \cdot S + \mathcal{N}. \tag{3}$$

Algorithm 1 The Matching Pursuit Solver

Input: measurement vector X, dictionary Dic.
Output: list of position-strength tuple $(Pos_n, c_n)_{n=1}^N$.
  Initialization: $n \leftarrow 1$, $R_n \leftarrow X$;
  while not reach the stop condition **do**
    $vec_n = \arg\max_{vec_i}[R_n \cdot vec_i/|vec_i|^2], vec_i \in Dic$;
    $Pos_n \leftarrow$ position coordinates extracted from vec_n;
    $c_n \leftarrow R_n \cdot vec_n/|vec_n|^2$;
    $R_{n+1} \leftarrow R_n - c_n vec_n$;
    $n \leftarrow n + 1$;
  end while

B. Casting to the Sparse Recovery Intuition

There are many previous attempts trying to solve the signal localization model of Equation 3. Sub-space based methods, such as MUSIC [2], [5], are widely used with linear arrays but have difficulties on UCA when handling coherent signals. We note that this limitation is not caused by the properties of physical signals, but by the mathematical properties of the sub-space solver. Thus, we propose to consider the tracking problem from the perspective of sparse recovery, which is immune to signal coherence and more compatible with our target scenario. The intuition is that, as static reflectors can be eliminated through signal cancellation [2], there are limited significant moving reflectors in an environment. Thus, the tracking problem can be formulated to find the smallest number of scaled and shifted reflection signals that could make up the overall signals received by the mic-array.

Technically, we expand the steering matrix A to an over-complete matrix with N ($N \gg D$) dimension $A' = [a(d_1,\theta_1), a(d_2,\theta_2), ..., a(d_N,\theta_N)]$, where $a(d_i,\theta_i)$ represents a possible reflector at distance d_i and AoA θ_i *w.r.t.* `ref`. Vector S is also expanded to an N dimension sparse vector $S' = [0, 0, ...s_1, 0, ...s_2...s_D...0]^T$. Thus, Equation 3 becomes

$$X = A' \cdot S' + \mathcal{N}. \tag{4}$$

If a true reflector with $a(d_i,\theta_i)$ exists, the corresponding coefficient will be s_i, otherwise, it is 0. In our scenario, as the transmitted signal $s_0(t)$ is pre-known, we further incorporate $s_0(t)$ into A':

$$X = [s_0(t)a(d_1,\theta_1)...s_0(t)a(d_N,\theta_N)]\cdot C = Dic\cdot C + \mathcal{N}, \tag{5}$$

where $Dic = [vec(d_1,\theta_1,t)...vec(d_N,\theta_N,t)]$ is called the *dictionary*. It is a pre-calculated matrix with each element, *i.e.*, vec_k, storing the time domain waveform reflected from a single location (d_k,θ_k). Similar to S', $C = [0, ...c_1, ...c_D...0]^T$ is a sparse vector representing the strength of the reflection from certain locations. When the number of significant reflector D is much less than the dimension N of the dictionary, the tracking problem can be formulated to a sparse recovery problem, *i.e.*, finding a sparse vector C to represent the measurements X vector in the dictionary space:

$$\min \|C\|_0 \ \ s.t. \ \ \|X - Dic\cdot C\|_2 \le \varepsilon \tag{6}$$

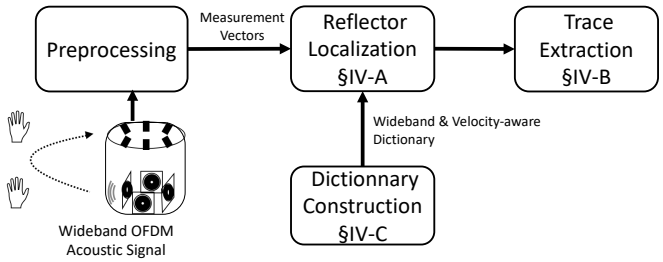


Fig. 3: **SparseTrack Overview.** We first adopt the preprocessing module from [2] to eliminate the self-interference and the static reflectors, while preserving reflection signals from users' gestures. Then we apply the sparse recovery framework to localize the reflectors (§IV-A). After that, we extract the gesture traces from continuous localization measurements (§IV-B). Finally, we discuss the detailed dictionary construction and the design in handling the Doppler effect (§IV-C).

Given a recovered C, we can derive reflectors' position $(Pos_n)_{n=1}^{D}$ from the position of nonzero coefficients, and also obtain the reflection strength from coefficients values.

In this work, we adopt an iterative greedy solver, matching pursuit (MP) [24] to solve the above sparse recovery problem. The MP solver framework is shown in Algorithm 1, which iteratively finds a vector from Dic with maximum projection to the measurement X, and then subtracts it from the measurement X until the number of iterations reaches the threshold.

IV. Design

Our target scenario is shown in Fig. 1 and the basic workflow is shown in Fig. 3. The SparseTrack system emits ultrasonic signals through the speaker and collects reflected signals via a UCA. To fully enable device-free gesture tracking, we still face three challenges. Firstly, we should address the spatial ambiguity issue caused by the insufficient spatial sampling rate. Secondly, we need to handle noisy measurements to extract gesture traces robustly. Thirdly, we should consider the dynamic nature of gestures and eliminate the negative impact of the Doppler shift on gesture tracking. In this section, we elaborate on our designs to address them.

A. Ambiguity-free Reflector Localization

1) The Spatial Ambiguity Problem: As analyzed before, spatial ambiguity exists because most smart speakers are designed to work on the audible band but the active acoustic-based tracking system works on the ultrasonic band for annoyance reduction. Thus, for a narrowband ultrasonic signal with frequency f, the spacings of microphones on commodity smart speakers (Table I) are generally larger than half of its wavelength, which causes spatial aliasing or ambiguities in localization. Technically, for a reflector at certain range d and direction θ, there may be multiple possible direction parameters $\theta_0, \theta_1 ... \theta_n$ making similar phase values as: $\phi_k(d,\theta_0) \approx \phi_k(d,\theta_1) + 2k_1\pi \approx ... \approx \phi_k(d,\theta_n) + 2k_n\pi$. In other words, there are multiple similar vectors in the dictionary Dic: $vec(d,\theta_0) \approx ... \approx vec(d,\theta_n)$. The ambiguity issue also exists in the range domain: $vec(d_0,\theta) \approx ... \approx vec(d_m,\theta)$.

Here we take a specific example to show the problem. Suppose the spacing R=5 cm, the frequency of emitted signal f=17 kHz, and the location of the reflector (d,θ) = (150 cm, 30°). Adopting the sparse recovery and MP solver mentioned in Section III-B, the projections among all $vec(d,\theta)$ in Dic are shown in Fig. 4a, where the color corresponds to the magnitude value of projections. MP solver needs to select the $vec(d_i,\theta_i)$ with maximum projection as the location estimation. However, the serious ambiguity in Fig. 4a makes it hard to identify the ground truth at (150cm, 30°).

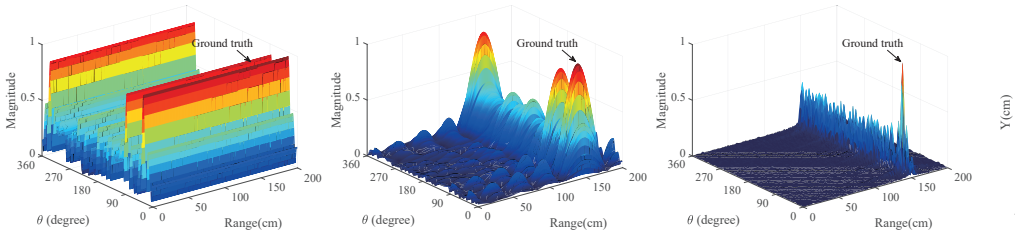
2) Synthesizing Wideband Measurements: To address the spatial ambiguity issue in UCA, we propose to synthesize wideband measurements. Instead of emitting a single-frequency signal, we utilize a wideband signal with K frequency components $f_1, f_2 ... f_K$, and corresponding dictionaries $Dic_1, Dic_2 ... Dic_K$. Our key observation is that the measurement from each frequency component f_i experiences different ambiguities, but all measurements include the positions of true reflectors. For example, the ambiguity of f_1 is $[\theta_0, \theta_i ... \theta_j]$, the ambiguity of f_2 is $[\theta_0, \theta_p ... \theta_q]$, the ambiguity of f_K is $[\theta_0, \theta_h ... \theta_k]$. Their only intersection will be the true direction θ_0. Thus, when we synthesize all measurements together, the measurements of the true position are enhanced and become possible for accurate localization. By leveraging the frequency diversity of the transmitted signal, we can address the spatial ambiguity issue without enforcing the uniform array spacing to be non-uniform spacing like [2].

Technically, our wideband-based sparse recovery can be formulated as follows: we use OFDM to generate a wideband signal with K subcarriers. According to Equation 5, for each subcarrier f_k, we have $X_k = Dic_k \cdot C_k$, In our scenario, reflector sparsity holds for dictionaries of different subcarriers, *i.e.* $C_k = C_l \ for \ k \neq l$. It means that the reflectors' position and attenuation coefficient at each single position is the same for different subcarriers, which is not contradictory to the frequency selective fading after the overlap of the multipath effect. Thus, it is equivalent to solve multiple-dictionary (MD) joint optimization problem as follows [25]:

$$\begin{aligned} &\min \|C_k\|_0, \ for \ k = 1, 2 ... K \\ &s.t. \|X_k - Dic_k \cdot C_k\|_2 \leq \varepsilon \ and \ C_k = C_l \ for \ k \neq l \end{aligned} \tag{7}$$

We leverage the constraint $C_k = C_l$ for $k \neq l$ to transform the above optimization to a single-dictionary version. Specifically, we stack X_k vertically as $X = [X_1, X_2 ... X_K]^T$ such that $X = Dic \cdot C$ where Dic is also stacked by Dic_k and $C = C_1 = C_2 = ... = C_K$. Then, we can utilize the MP solver in Algorithm 1 to derive C.

We take the same scenario in Fig. 4a to intuitively show the effectiveness of our design. We first test OFDM signals with 10 subcarriers from 17 kHz to 17.5 kHz. Compared with the single-frequency case, Fig. 4b shows a significant reduction of ambiguities in the range dimension. Then, we test OFDM signals with 120 subcarriers from 17 kHz to 23 kHz. As shown in Fig. 4c, the ambiguity issue disappears and the ground truth location (150cm, 30°) can be clearly identified. The final



(a) Single Frequency (17kHz) (b) 10 Subcarriers (17-17.5kHz) (c) 120 Subcarriers (17-23kHz)

Fig. 4: **Using Wideband Signal to Eliminate Spatial Ambiguity**. Different narrow band signals result in different ambiguities but all contain the correct reflector information. Accurate results can be obtained after synthesizing them together.

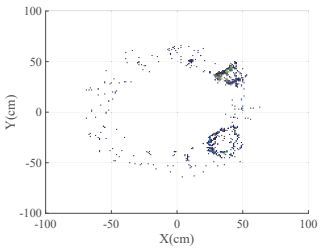


Fig. 5: **Example Outputs of Reflector Localization**. Two gestures are performed simultaneously in 8 seconds.

parameters of the OFDM signals emitted by SparseTrack are shown in Section V.

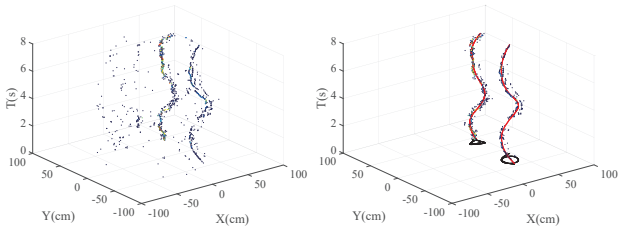
Before locating the reflector, it is worth noting here that we adopt the interference cancellation method in [2] to filter out the interference from direct transmission of the speaker and reflection of static clutters and human bodies. Specifically, we record the direct transmission and environmental reflection in the static scene. We then subtract this recorded signal from the received signals. We further estimate the channel information to remove the residual reflectors like human bodies as in [2].

B. Robust Gesture Tracking

1) Trace Extraction: In this section, we aim to extract gesture traces from the output of reflector localization. A sample output of our ambiguity-free reflection localization is shown in Fig. 5. Two users are required to 'draw a circle' and 'draw a triangle' simultaneously. During the drawing period, the smart speaker repeatedly transmits and receives OFDM symbols. The OFDM symbols are synchronized through a chirp header. We run the ambiguity-free reflector localization algorithm on each symbol, and the accumulated localization results are shown in Fig. 5.

To reduce the noise in the gesture traces in Fig. 5, SparseTrack leverages the time domain to expand the localization outputs to the time-space 3D space as in Fig. 6a. In the time-space 3D space, the noise points are scattered from the real reflector traces. Therefore, we utilize the clustering technique (DBSCAN algorithm in our prototype) to identify large clustered point sets, *i.e.*, those highly likely induced by gestures. Then we conduct 3D curve fitting to bridge these clusters respectively (to account for missing points between clusters). Finally, we project the 3D curves back to the 2D plane to extract gesture traces. The circle and the triangle are revealed in the X-Y plane of Fig. 6b.

2) Reducing Computational Cost: In this section, we introduce our optimization to better support real-time tracking. We first analyze the computation cost per OFDM symbol. Given L microphones, K subcarriers, N_θ search steps in direction domain, N_d search steps in range domain and N_{iter} iterations for MP solver of the reflector localization, the time complexity per symbol will be $O(\eta) = O(N_{iter} \cdot N_\theta \cdot N_d \cdot L \cdot K)$. Since human hand movement is relatively slow, we can reduce the size of the spatial search window by setting smaller $N_\theta \cdot N_d$ without degrading tracking performance. Specifically, the MP solver can search in a window of 20 cm$\times$20 cm with a 1 cm search step for each gesture per OFDM symbol. After certain numbers of initial symbols (10 in our prototype), the spatial search window will slide and track the gesture according to the latest extracted traces.



(a) (b)

Fig. 6: **Leverage Time Domain Information to Extract Gesture Traces**. (a) Casting the location measurements into the time-space 3D space to scatter noise points. (b) Using clustering and curve fitting to extract the noise-free trace.

C. Combating Reflector Dynamics

1) The Impact of Doppler Effect: Due to the low velocity of the sound, it is known that the Doppler effect has obvious impact on acoustic measurement. When the reflector is moving quickly, *e.g.*, waving the hand in an interactive game, the frequency of the reflected signals is shifted. This will downgrade the performance of gesture tracking, because the signals received by the mic-array are no longer the delayed version of the transmitted OFDM symbols. Specifically, suppose the original frequency-domain sequence of OFDM signal is $[s(k), k = 0, 1...N-1]$, the time-domain sequence is

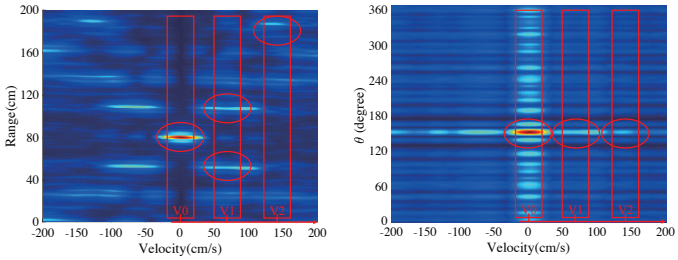
$$t(n) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) e^{j\frac{2\pi n k}{N}} \ (n = 0, ...N-1) \tag{8}$$

Due to the Doppler effect, the time-domain sequence becomes:

$$td(n) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) e^{j\frac{2\pi n (k+\varepsilon)}{N}} \ (n = 0, ...N-1) \tag{9}$$

where ε is the normalized frequency offset, *i.e.*, frequency offset divided by the subcarrier interval. The corresponding frequency-domain sequence is:

$$sd(k) = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{i=0}^{N-1} s(i) e^{j\frac{2\pi n}{N}(i+\varepsilon-k)} \ (k = 0, ...N-1) \tag{10}$$



(a) Range (b) Direction

Fig. 7: **Impact of the Doppler Effect.** With the velocity increasing from V0 to V2, frequency shifts make the range measurement (a) in wrong position and the direction measurement (b) in reduced signal strength.

In the ideal case without the Doppler effect, the transmitted and received signals are $X_k = A_k S_k = A_k[c_1, ..., c_D]^T s(k)$ for each subcarrier. However, in the real situation, the corresponding signals are $A_k[c_1, ..., c_D]^T sd(k)$. When the hand is moving, $\varepsilon \neq 0$, thus $sd(k) \neq s(k)$. Given motion speed v, the frequency offset for signals with frequency f_i will be $\Delta f_i = 2f_i \cdot \frac{v}{c}$. Thus, the difference between $sd(k)$ and $s(k)$, and hence the error is larger with the greater moving speed.

We conduct simulation experiments to illustrate the impact of the Doppler effect. Suppose a reflector at (80 cm, 150°) with different instantaneous velocities from -200 to 200 cm/s, where the positive ones denote moving close and negative ones denote moving away. We always adopt the dictionary Dic_k calculated from the static case for the MP solver to localize the (d, θ) under different instantaneous velocities, *i.e.*, without handling the Doppler effect properly. As shown in Fig. 7, with the increase of velocity, the results of the range measurement become more dispersed and deviated from the ground truth. Performance degradation can also be observed in the direction measurement, *i.e.*, weaker signal strength in true direction with increasing velocities.

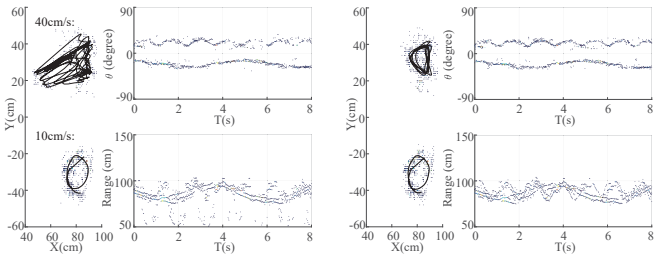
We also test a real scenario where one user performs gestures with a velocity around 10 cm/s, and the other performs gestures with a velocity around 40 cm/s. As shown in Fig. 8a, if using the default dictionary, the range measurements often deviate from the correct value. Both range and direction measurements of the quicker gesture are often missing, which is caused by the low signal strength. Thus, the dictionary Dic_k for the MP solver should adapt to the gesture dynamics accordingly to reduce localization errors.

2) Velocity-aware Dictionary: Based on the above analysis, we propose to construct the velocity-aware dictionary to address the Doppler effect problem. Specifically, we expand the original 2D dictionary (d, θ) to a 3D dictionary (d, θ, v). We denote $sd_{(k,v_i)}$ for velocity v_i. Thus, the matching dictionary for velocity v_i should be:

$$Dic_{(k,v_i)} = [sd_{(k,v_i)} a_k(d_1, \theta_1) ... sd_{(k,v_i)} a_k(d_N, \theta_N)] \quad (11)$$

We then stack dictionaries for different velocities together:

$$Dic3D_k = [Dic_{(k,v1)}, Dic_{(k,v2)} ... Dic_{(k,vi)}] \quad (12)$$



(a) w/o Velocity-aware Dictionary (b) w/ Velocity-aware Dictionary

Fig. 8: **Trace Extraction under the Doppler Effect.** (a) The localization performance is interfered with by the reflector movement. (b) The impact of the Doppler effect can be compensated by including the velocity in the dictionary.

As shown in Fig. 8b, with the help of the velocity-aware dictionary, both the range and direction localization have much better accuracy and continuity. It is worth mentioning that the time complexity per symbol now becomes $O(\eta) = O(N_{iter} \cdot N_v \cdot N_\theta \cdot N_d \cdot L \cdot K)$ after adding N_v search steps in the velocity domain, but it still supports real-time gesture tracking as evaluated in Section VI-B.

V. Implementation

This section describes the implementation of the SparseTrack prototype and our parameter selection.

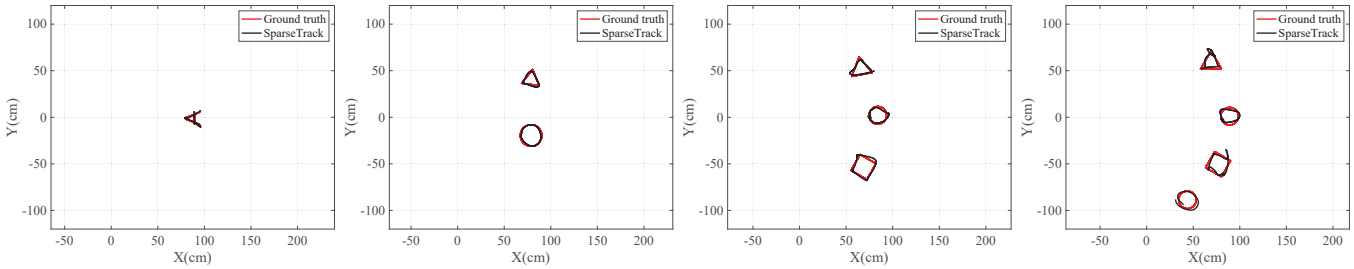
Prototype. We implement the SparseTrack prototype with a Raspberry Pi 3 board, commercial off-the-shelf (COTS) speakers, and a COTS mic-array. As shown in Fig. 10a, the prototype has a similar layout as commodity smart speakers, such as Amazon Echo. Specifically, we use four Edifier M1250 speakers to emit signals to achieve 360° azimuthal coverage. The mic-array is a ReSpeaker 6-Mic circular uniform array with 4.7 cm spacing, which is widely used for prototyping smart speakers. We note that the Raspberry Pi micro-controller is only for controlling and data logging, and not powerful enough for audio signal processing. We emit the ultrasonic signals and record the reflected signals into WAV files. Then the recorded signals are analyzed in MATLAB using a MacBook laptop with an Intel i5 processor and 16 GB memory.

Parameters of the Emitted OFDM Signals. We repeatedly transmit and receive OFDM symbols for acoustic sensing. The default OFDM symbol has 6 kHz bandwidth (17 kHz - 23 kHz) and 960 time-domain samples under 48 kHz sampling rate, *i.e.*, the duration of each symbol is 20 ms. There are 480 subcarriers in total, but only the frequency coefficient of 120 subcarriers within 17 kHz - 23 kHz is a non-zero random sequence composed by [-1,1]. Unless otherwise specified, we transmit and receive 400 symbols (8 seconds) for every single experiment to calculate tracking error in evaluation. And we add a chirp sequence before these OFDM symbols as the preamble for synchronization.

VI. Evaluation

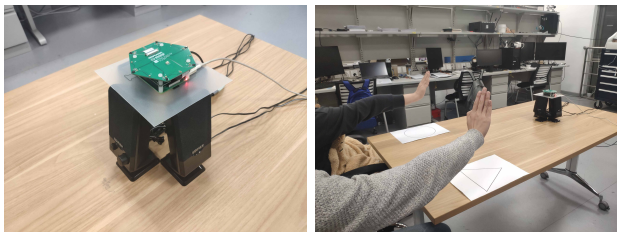
A. Evaluation Settings and Methodology

As shown in Fig. 10b, we ask the volunteers to sit in front of the speakers and draw different shapes according to the



(a) Single User (A) (b) Two Users (c) Three Users (d) Four Users

Fig. 9: **Tracking Results Example.** The smart speaker is located at (0,0). SparseTrack can simultaneously track up to 4 users.



(a) SparseTrack Prototype (b) Evaluation Settings

Fig. 10: **Prototype and Evaluation Settings**.

predefined templates on the table. They are required to follow the size and shape of the template. The evaluated templates include simple shapes like triangles and circles and complex shapes like uppercase English letters and Arabic numbers. These templates are used as the ground truth. We collect 167 traces in total under different experiment conditions, which will be detailed in the corresponding subsections. The tracking error is measured in the symbol basis. It is defined as the minimum Euclidean distance from the estimated hand positions by the symbol to the trajectory of the template.

We first conduct micro-benchmark to show the impact of OFDM parameters selection, and the effectiveness of our design components on handling spatial ambiguity and the Doppler effect. We also measure the end-to-end latency of our system to evaluate the real-time tracking capability. Then we provide an overall evaluation on tracking accuracy and detailed evaluation on different impacting factors in practice, including the number of users, ranges, directions, ambient environment. In general, SparseTrack can simultaneously track 1 to 4 users' gestures. Some examples are shown in Fig. 9.

Unless otherwise specified, we ask two volunteers to sit 1 m away from the speakers with a 30° bearing angle in a lab environment. They are required to draw different shapes along the templates with hands with a speed of around 10 cm/s. We change the experiment parameters based on this default setting to evaluate different aspects of our system and the impact of various factors.

B. Micro-benchmark

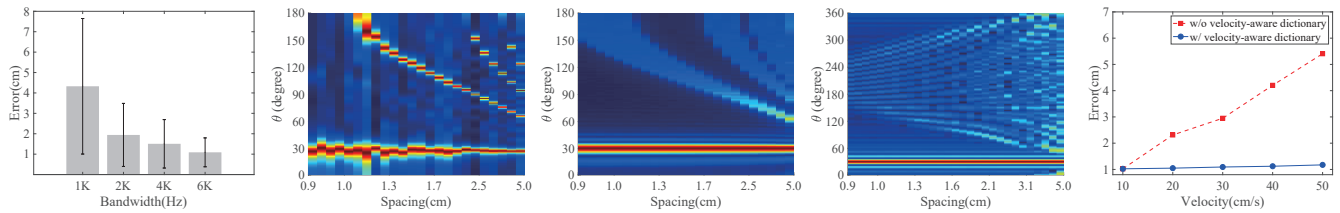
OFDM Parameter Selection. We determine the parameters of the emitted OFDM signals, *i.e.*, the bandwidth and the symbol duration. To choose proper signal bandwidth, we emit various bandwidths of OFDM signals including 1 kHz, 2 kHz, 4 kHz, 6 kHz to test the tracking performance. The mean and variance of errors are shown in Fig. 11a. The tracking accuracy is improved with larger bandwidth, which validates the benefits of synthesizing wideband measurement. Since acoustic signals with frequencies higher than 17 kHz are inaudible for most people and frequencies higher than 23 kHz are subject to severe attenuation in our platform, the OFDM sub-carriers are set to 17 kHz - 23 kHz. In addition, we set the duration of the emitted OFDM symbol as 960 samples under 48 kHz sampling rate, *i.e.*, the duration is 20 ms, because 20 ms allows a sensing range of 3.4 m, which can meet the requirements of most indoor applications.

Handling Spatial Ambiguity. We first compare our design with the 2D MUSIC approach [2]. To ensure a fair comparison, the chirp signal of RTrack is set to 20 ms and spans between 17 kHz to 23 kHz. In the simulation, both SparseTrack and RTrack use the same uniform linear array (ULA) with a 25 cm aperture size. The ground truth signal comes from 30° with 10 dB SNR. We modify the mic spacing via changing the number of the mic in ULA, *i.e.*, given an array aperture of 25 cm, 30-mic array corresponds to mic spacing of around 0.9 cm, while 6-mic array corresponds to the mic spacing of 5 cm.

As shown in Fig. 11b and Fig. 11c, when the mic spacing is less than half of the ultrasonic wavelength, both approaches experience no ambiguity. However, the ambiguity problem of the 2D MUSIC approach becomes obvious when the mic spacing increases. On the contrary, SparseTrack can still clearly identify the correct position, which validates the effectiveness of the ambiguity-free reflector localization design.

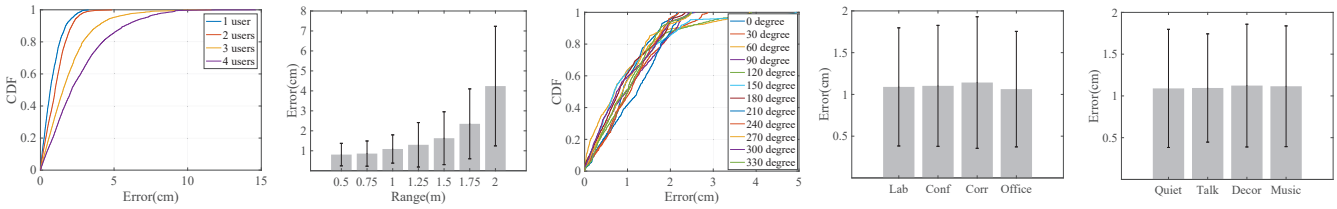
Since RTrack does not work in UCA, we only evaluate SparseTrack in the UCA case. We set the radius of UCA as 5 cm, and change the number of the mic from 35 to 6. The corresponding mic spacing is changed from 0.9 cm to 5 cm. The results are shown in Fig. 11d. SparseTrack can handle the spatial ambiguity issue even when the mic spacing is much larger than half of the wavelength. Thus, SparseTrack can support gesture tracking on today's smart speakers with circular geometry.

Handling the Doppler Effect. To evaluate the effectiveness of the velocity-aware dictionary, we ask one volunteer to draw shapes under default settings, but another user to draw shapes with different speeds from 10 cm/s to 50 cm/s. We conduct gesture tracking with both the static dictionary and the velocity-aware dictionary for the user with different gesture speeds to compare the tracking performance of two dictionaries under the Doppler effect.



(a) OFDM Bandwidth (b) 2D MUSIC with ULA (c) SparseTrack with ULA (d) SparseTrack with UCA (e) Gesture Speed

Fig. 11: **Micro-benchmark Evaluation.** SparseTrack leverages wideband signals (Fig. 11a) to address the spatial ambiguity issue on both ULA (Fig. 11c) and UCA (Fig. 11d). In both cases, it can correctly identify the ground truth at 30°. In addition, it eliminates the impact of the Doppler effect through a velocity-aware dictionary (Fig. 11e).



(a) Tracking Accuracy (b) Impact of Range (c) Impact of Direction (d) Impact of Location (e) Impact of Noise

Fig. 12: **Overall Performance.** SparseTrack can track 4 users' gestures simultaneously with a mean tracking error of 2.66 cm (Fig. 12a). It achieves good tracking performance within the range of 1.5 m (Fig. 12b) and works robustly under different directions (Fig. 12c), locations (Fig. 12d), and ambient noise (Fig. 12e).

Fig. 11e shows that when the user moves hand with a velocity of 10 cm/s, both the static dictionary and the velocity-aware dictionary achieve a mean error of around 1.0 cm. However, when the motion speed increases, the static dictionary leads to continuous performance degradation, while the performance of the velocity-aware dictionary remains stable. This comparison validates the effectiveness and necessity of our design on handling the Doppler effect.

System Latency. In this part, we show that our design can support real-time gesture tracking. Table II shows the processing time of each component in SparseTrack for tracking 1 to 4 users. Specifically, we set the spatial search window as 20 cm×20 cm with 1 cm×1 cm step size, and the velocity search window as -50 cm/s to 50 cm/s with 10 cm/s step size. The iteration numbers of the MP solver for 1, 2, 3, and 4 users are 2, 4, 6, and 8 respectively. As shown in Table II, the time consumption of pre-processing module mainly comes from operations on interference cancellation, subcarrier calculation, and selection for each microphone, which takes a constant time of 2.0 ms for different user numbers.

The time consumption of the reflector localization module mainly depends on the MP solver, which searches reflectors over the search window. With the increasing number of users, two factors, *i.e.*, the number of tracking windows and the number of MP iterations also increase. Thus, the processing time of 2, 3, and 4 users cases increases accordingly, which is approximately 4, 9, and 16 times of that of the 1-user case. The time consumption of the trace extraction module is mainly from the process of updating the temporary dictionary for the slid window. Since each user has a search window and the corresponding dictionary, the processing time of 2, 3, and 4 users cases are approximately 2, 3, and 4 times of that of the 1-user case. The end-to-end latency for 1-4 users is 7.1 ms, 20.5 ms, 43.2 ms, and 76.6 ms respectively, which are sufficiently short to support real-time gesture tracking [1].

User Number	Pre-processing	Reflector Localization	Trace Extraction	Total Latency
1	2.0	3.7	1.4	7.1
2	2.0	15.8	2.7	20.5
3	2.0	37.4	3.8	43.2
4	2.0	69.6	5.0	76.6

TABLE II: Processing Time (Unit: ms).

C. Overall Performance

Tracking Accuracy. In this part, we evaluate the tracking accuracy of SparseTrack. We evaluate the scenarios with 1, 2, 3, and 4 users respectively. Each volunteer sat 1 m away from the speakers with a 30° bearing angle. They were asked to draw different shapes over templates using their hands. The drawing samples include simple shapes like triangle, circle and complex shapes like letters and Arabic numbers are shown in Fig. 9, where the red lines indicate ground truth while the black ones indicate the outputs of our tracking system.

Fig. 12a shows the cumulative distribution function (CDF) of the tracking errors for the cases of 1 to 4 users. The mean errors are 0.82 cm, 1.09 cm, 1.90 cm, and 2.66 cm respectively. Thus, our system can support fine-grained gesture tracking even when four users perform gestures simultaneously. We then evaluate four practical impact factors in the following.

Impact of Sensing Ranges. We ask two volunteers to draw shapes under default settings with various ranges to the smart speaker from 0.5 m to 2 m. The mean and variance of tracking error at different ranges are shown in Fig. 12b. The tracking

[1]Since the OFDM symbol is 20 ms, we can process one out of four OFDM symbols for the four user case. 80 ms per location sample is sufficient for most tracking applications.

performance degrades when the range increases. The mean tracking error increases to 4.24 cm at 2 m. The reason behind this is the SNR of reflection signals drops rapidly with the increase of sensing range, which is a common issue for device-free sensing systems.

Compared with the performance of RTrack [2], another reason limiting the range of SparseTrack is the microphone orientation. RTrack uses self-made microphones which orientate to the direction of the reflection. However, the orientation of our commercial mic-array is upward. The microphone directivity weakens the received signal. Based on the current settings, our system can provide good tracking performance within 1.5 m, which can be easily improved by using mic-arrays with more suitable packaging format.

Impact of Sensing Directions. We ask one volunteer to sit at 345° and another one to sit from 0° to 330° with a step size of 30°. Fig. 12c shows the CDF of the tracking errors for the cases of 0° to 330°. As the CDFs have similar trends and the mean errors are all around 1.0 cm, SparseTrack achieves stable performance across different directions. This is because SparseTrack adopts a circular speaker and microphone array to support transmitting and receiving acoustic signals for 360° directions, which is superior to linear- array-based solutions Compared with existing methods like 2D MUSIC, SparseTrack can work well on commercial smart speakers with uniform circular mic-array to support omnidirectional device-free gesture tracking.

Impact of Ambient Locations. We evaluate the system performance at four typical indoor scenarios: the lab, the conference room, the corridor, and the office, with different space span and density of furniture. The mean and variance of tracking error at different locations are shown in Fig. 12d. It is seen that the mean errors are all around 1.0 cm across different locations with different degrees of space crowdedness. Thus, our system can be deployed in common indoor scenarios to achieve robust tracking performance for various applications. This is because the interference cancellation effectively removes the reflections from static objects in the environment.

Impact of Ambient Sound. We test three types of ambient sound in daily lives, including talking, decoration, and music, with typical decibel levels of 60 dB, 70 dB, and 78 dB respectively. We play the sound sources with a loudspeaker and place it 0.5 m away from the mic-array of the smart speaker. As shown in Fig. 12e, our system has tracking errors around 1.0 cm under the impact of different ambient sounds, which indicates that the tracking performance of SparseTrack is not affected by the noise. Although the decibel levels of ambient sound are high, our tracking system utilizes ultrasonic frequencies and filters out the non-ultrasonic ones. Thus, it works robustly in most indoor scenarios.

VII. CONCLUSION

In this work, we propose SparseTrack to achieve fine-grained multi-user device-free gesture tracking on today's smart speakers with uniform circular geometry. We cast device-free tracking to sparse recovery intuition to address signal coherence issues on circular mic-arrays. We address practical challenges on the insufficient spatial sampling rate, doppler effect, and trace extraction and implement SparseTrack on COTS circular mic-array.

ACKNOWLEDGEMENTS

We thank anonymous reviewers for their valuable comments. This work is supported (in part) by the ShanghaiTech Startup Fund, the "Chen Guang" Program 17CG66 supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission, and NSFC 62002224.

REFERENCES

[1] K. Sun, C. Chen, and X. Zhang, ""alexa, stop spying on me!": Speech privacy protection against voice assistants," ACM Sensys 2020.
[2] W. Mao, M. Wang, W. Sun, L. Qiu, S. Pradhan, and Y.-C. Chen, "Rnn-based room scale hand motion tracking," ACM MobiCom 2019.
[3] R. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Transactions on Antennas and Propagation 1986.
[4] F. Belfiori, W. van Rossum, and P. Hoogeboom, "Application of 2d music algorithm to range-azimuth fmcw radar data," IEEE European Radar Conference 2012.
[5] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," ACM SIGCOMM 2015.
[6] Tie-Jun Shan and T. Kailath, "Adaptive beamforming for coherent signals and interference," IEEE Transactions on Acoustics, Speech, and Signal Processing 1985.
[7] H. Wang, K. R. Liu, and H. Anderson, "Spatial smoothing for arrays with arbitrary geometry," IEEE ICASSP 1994.
[8] F. Belloni and V. Koivunen, "Beamspace transform for uca: Error analysis and bias reduction," *IEEE Transactions on Signal Processing*, 2006.
[9] J. Dmochowski, J. Benesty, and S. Affes, "On spatial aliasing in microphone arrays," *IEEE Transactions on Signal Processing*, 2009.
[10] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: Using the doppler effect to sense gestures," ACM CHI 2012.
[11] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan, "Audiogest: Enabling fine-grained hand gesture detection by decoding echo signal," ACM UbiComp 2016.
[12] K. Ling, H. Dai, Y. Liu, and A. X. Liu, "Ultragesture: Fine-grained gesture sensing and recognition," IEEE SECON 2018.
[13] M. Chen, J. Lin, Y. Zou, R. Ruby, and K. Wu, "Silentsign: Device-free handwritten signature verification through acoustic sensing," IEEE PerCom 2020.
[14] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," ACM CHI 2016.
[15] K. Sun, T. Zhao, W. Wang, and L. Xie, "Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals," ACM MobiCom 2018.
[16] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," ACM MobiCom 2016.
[17] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," ACM MobiSys 2017.
[18] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," ACM MobiCom 2013.
[19] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using wifi," ACM MobiSys 2017.
[20] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," ACM MobiSys 2019.
[21] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," USENIX NSDI 2014.
[22] W. Gong and J. Liu, "Robust indoor wireless localization using sparse recovery," IEEE ICDCS 2017.
[23] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using rf backscatter," USENIX NSDI 2015.
[24] S. G. Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," IEEE Transactions on Signal Processing 1993.
[25] Z. Tang, G. Blacquiere, and G. Leus, "Aliasing-free wideband beamforming using sparse signal representation," *IEEE Transactions on Signal Processing*, 2011.