Indoor Smartphone SLAM With Acoustic Echoes

Luo, Wenjie; Song, Qun; Yan, Zhenyu; Tan, Rui; Lin, Guosheng

DOI
10.1109/TMC.2023.3323393

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Mobile Computing

Indoor Smartphone SLAM With Acoustic Echoes

Wenjie Luo, Qun Song, *Member, IEEE*, Zhenyu Yan, *Member, IEEE*, Rui Tan, *Senior Member, IEEE*, and Guosheng Lin, *Member, IEEE*

***Abstract*—Indoor self-localization has become a highly desirable system function for smartphones. The existing systems based on imaging, radio frequency, and geomagnetic sensing may have sub-optimal performance when their limiting factors prevail. In this paper, we present a new indoor simultaneous localization and mapping (SLAM) system that is based on the smartphone's built-in audio hardware and inertial measurement unit (IMU). Our system uses a smartphone's loudspeaker to emit near-inaudible chirps and then the microphone to record the acoustic echoes from the indoor environment. The echoes contain the smartphone's location information with sub-meter granularity. To enable SLAM, we apply contrastive learning to train an echoic location feature (ELF) extractor, such that the loop closures on the smartphone's trajectory can be accurately detected from the associated ELF trace. The detection results effectively regulate the IMU-based trajectory reconstruction. The reconstructed trajectories are used for *trajectory map superimposition* and *room geometry reconstruction*. Extensive experiments show that our SLAM achieves median localization errors of 0.1 m, 0.53 m, and 0.4 m in a living room, an office, and a shopping mall, and outperforms both the Wi-Fi and geomagnetic SLAM systems. The room geometry reconstruction achieves up to 4× lower errors compared with the latest echo-based approaches.**

***Index Terms*—Simultaneous localization and mapping (SLAM), contrastive learning, acoustic sensing.**

I. Introduction

LOCATION awareness is a fundamental requirement for mobile operating systems. As of 2023, more than 70% of the top 100 Android apps require location information. Various smartphone's built-in sensing modalities, including Wi-Fi [1], BLE [2], GSM [3], FM radio [4], visible light [5], imaging [6], acoustic background [7], and geomagnetism [8] have been exploited for indoor location sensing. However, these sensing modalities have their own limiting factors. For instance, radio frequency (RF) signals are susceptible to electromagnetic noises. Visible light sensing suffers blockage. Visual imaging may incur privacy concerns in certain spaces and times. The acoustic background only provides room-level granularity. Therefore, exploiting new modalities based on smartphones' built-in hardware to enrich location-sensing services has been an interest of research.

Using a smartphone's audio system for active indoor location sensing receives increasing research interest [9], [10], [11], [12], [13], [14]. The active sensing uses smartphone's loudspeaker to emit excitation signals in the target indoor space and microphone to capture the acoustic echoes that carry location information. The existing approaches for location sensing can be divided into two categories. The *analytic approach* [12], [13], [14] analyzes the sound reflection processes from nearby surfaces (e.g., walls) for location estimation. However, when the indoor spaces are complicated (e.g., with irregular surfaces, many nearby objects with complex structures, etc), accurate object association becomes intractable. Thus, the existing analytic approaches often make simplifying assumptions that the major reflectors are at most two nearby walls [12], [13], [14]. The *fingerprint approach* [9], [10], [11] uses the echoes captured by the smartphone as the fingerprints of the locations and then applies supervised machine learning to build localization models. However, fingerprint data collection at spatially fine-grained locations incurs a high overhead. Thus, the existing studies focus on room-level location sensing [9], [11] or recognize a limited number of locations (11 closed locations in [10]).

Nevertheless, the fingerprint approach exhibits the potential to offer good generalizability as it does not make specific assumptions about the surroundings. To investigate whether satisfactory resolutions can be maintained when the number of fingerprinted locations increases, we conduct a measurement study of fingerprinting 128 locations using active sensing in a $16 \times 28\,\mathrm{m}^2$ office. We train a recognition model using labeled data. Results show that the fingerprint approach achieves sub-meter location sensing accuracy. Thus, acoustic echo is a promising modality for building indoor localization services on smartphones.

To unleash the fingerprint approach from the laborious training data collection process, in this paper, we aim to design a

Manuscript received 15 January 2023; revised 4 October 2023; accepted 6 October 2023. Date of publication 10 October 2023; date of current version 7 May 2024. This work was supported in part by the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU). This work was also supported in part by Singapore MOE Tier 1 (RG88/22). Recommended for acceptance by A. Conti. *(Corresponding author: Rui Tan.)*

Wenjie Luo and Rui Tan are with the Singtel Cognitive and Artificial Intelligence Lab (SCALE) for Enterprises, School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore 639798 (e-mail: wenjie005@e.ntu.edu.sg; tanrui@ntu.edu.sg).

Qun Song was with the SCSE, NTU, Singapore 639798. She is now with the Electrical Engineering, Mathematics and Computer Science (EEMCS), Delft University of Technology, 2628 Delft, CD, The Netherlands.

Zhenyu Yan was with the SCALE and SCSE, NTU, Singapore 639798. He is now with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: zyyan@ie.cuhk.edu.hk).

Guosheng Lin is with the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore 639798 (e-mail: gslin@ntu.edu.sg).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2023.3323393, provided by the authors.

Digital Object Identifier 10.1109/TMC.2023.3323393

simultaneous localization and mapping (SLAM) system using the smartphone's inertial measurement unit (IMU) data and the acoustic echoes collected by the microphone. Specifically, when a user carrying the smartphone moves in the indoor space, if he/she returns to a previously visited location, the user trajectory forms a *loop closure*. If the loop closures can be correctly detected using the acoustic echo, the IMU-based dead reckoning result, which is prone to sensor errors, can be regulated to obtain an accurate trajectory. As a result, the reconstructed trajectory and the associated echo data form a *trajectory map*.

As a key step of SLAM, loop closure detection requires an effective feature embedding to determine if two echo samples are collected at the same/different locations. However, it is challenging to find such an effective embedding for acoustic echoes. Our experiments show that the generic acoustic features, e.g., power spectral density (PSD), spectrogram, and principal component analysis (PCA), are ineffective for location discrimination. Thus, we resort to finding an effective embedding using deep neural networks (DNNs). The embeddings learned via supervised learning become ineffective on the data out of the training dataset. Differently, contrastive learning (CL) is a self-supervised learning technique that constructs effective embeddings from unlabeled data. Applied to our SLAM problem, CL can be used to learn an acoustic representation from the unlabeled echo data and only requires the information of whether two echoes are collected at close locations. Thus, we apply CL to train a feature extractor that outputs a new representation called *echoic location feature* (ELF). Then, we compute ELFs' similarity to detect loop closures.

We make the following four designs to realize the ELF-based SLAM. First, we design a trajectory-level CL procedure to learn the trajectory-specific ELFs for loop closure detection. It consists of *pre-training* a basic ELF extractor based on the incremental learning scheme, and *fine-tuning* the extractor for target room adaptation using limited unlabeled echoes. Second, we design a loop closure curation approach to remove the false positives by exploiting prior knowledge of the user's movement. Third, we design a floor-level CL procedure to superimpose the crowdsourced trajectory maps to form a single *floor map*. The procedure can effectively reconcile the differences among the ELFs from multiple trajectory maps at the same spot caused by the smartphone orientation. Lastly, we use the echoes to estimate the wall distances and then leverage the estimated distances and the rectified user trajectory for room geometry reconstruction.

The main contributions of the paper are:

- We conduct extensive measurement studies to investigate the spatial property of the acoustic echo. The acoustic echo exhibits a sub-meter spatial resolution limit and is promising for designing an accurate indoor location sensing system.
- We design CL and learn ELFs for loop closures detection. We further design ELF-SLAM using IMU data and learned ELFs on a smartphone. We also apply CL to superimpose the crowdsourced trajectory maps.
- We leverage the reconstructed trajectory and estimate the wall distance for room reconstruction. Evaluations show that our system outperforms the existing echo-based room reconstruction systems.
- We conduct extensive experiments in various indoor environments. ELF-SLAM achieves sub-meter mapping and localization accuracy and outperforms SLAM systems based on Wi-Fi and geomagnetism. We also study the allowable intensities and/or needed mitigation for various practical affecting factors, including nearby people, audible noises, and space layout changes.

Paper Organization: Section II reviews related work. Section III presents the measurement study. Section IV presents the ELF-SLAM design. Section V presents room geometry reconstruction. Section VI presents evaluation results. Section VII discusses several potential approaches to improve the system. Section VIII concludes the paper.

II. Related Work

■ *Acoustics-based indoor location sensing:* The ubiquity of speakers and microphones on consumer electronics has promoted acoustics-based indoor location sensing in the last few decades [17]. In [18], [19], [20], a device's indoor position can be estimated by receiving and analyzing the sound emitted from the deployed acoustic beacons. However, the infrastructure-based approaches may incur the undesirable overhead of deploying dedicated sound beacons or receivers. Thus, we mainly review infrastructure-free approaches as summarized in Table I. The *analytic approach* analyzes the sound propagation processes for location sensing. It either senses the sounds from the external source or generates probing signals and analyzes the echoes. VoLoc [15] uses a speaker to detect the angle of arrival of the user's voice for localization. EchoSpot [14] uses a device to emit near-inaudible signals and analyze the signals' times of flight reflected off the human body. However, Voloc and EchoSpot require prior knowledge of the sound reflectors for triangulation. Another application of the analytic approach is indoor mapping, which estimates the wall distances to the smartphone. To build the room contour, studies [12], [13] require a user to walk along the walls for data collection. Then, the IMU data is used to construct the user trajectory and the echo data is used to estimate the wall distances. These two studies presume an accurate IMU-based trajectory. However, IMU-based dead reckoning is prone to sensor errors and subject to error accumulation problems. In this work, we reconstruct the room based on the accurate trajectory reconstructed by the proposed ELF-based SLAM.

The *fingerprint approach* collects acoustic echoes from different spots of a room and trains recognition models for location inference [7], [9], [10], [11], [16]. Early studies [7], [9], [16] apply conventional feature engineering and require either long data collection time that may incur privacy concerns or full-spectrum recording that is susceptible to interference like ambient noise [11]. DeepRoom [11] applies deep learning to reduce the requirements on recording time and spectrum usage. The studies [7], [9], [11], [16] address semantic or room-level location sensing. EchoTag [10] uses echoic fingerprints to tag up to 11 spots at centimeter spatial resolution. When the fingerprint approach is extended to a large indoor space, the blanket process

TABLE I
SUMMARY OF INFRASTRUCTURE-FREE ACOUSTIC INDOOR LOCALIZATION AND MAPPING

Approach	Study	Objective	Presumption on reflectors	Labeled training data	Resolution	Sensing technique		
						Mode	Sensing signal	Duration
Analytic	VoLoc [15]	Localization	Yes	No	Decimeters	Passive	Human voice	15 cmds
	EchoSpot [14]					Active	18-23kHz FMCW[1]	0.2 s
	BatMapper [13]	Mapping					8-16kHz chirps	0.04 s
	SAMS [12]						11-21kHz FMCW	0.03 s
Fingerprint	SurroundSense [16]	Localization	No	Yes	Semantic locations	Passive	Acoustic	60 s
	Batphone [7]				(Sub-) room-level		background	10 s
	RoomSense [9]					Active	0-24kHz MLS[2]	0.68 s
	DeepRoom [11]						20kHz tone	0.1 s
	EchoTag [10]	Location tagging			Centimeter		11-22kHz chirp	0.42 s
	ELF-SLAM	Localization & mapping		No	Sub-meter		15-20kHz chirp	0.1 s

[1]Frequency-Modulated Continuous-Wave, [2]Maximum Length Sequence

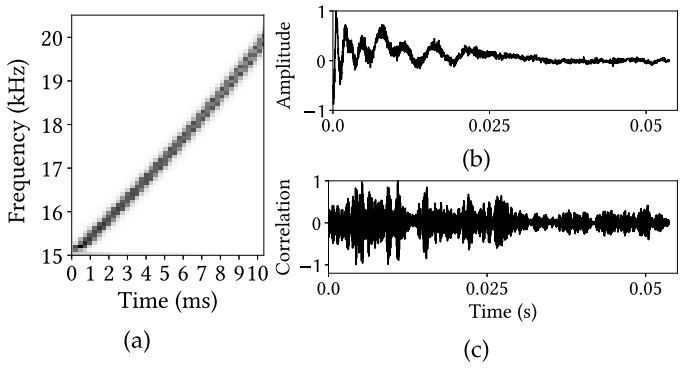


Fig. 1. (a) Probing signal: a logarithmic chirp sweeping the 15–20 kHz band within 10 ms; (b) Recorded time-series echoes; (c) Correlated result using the received echo and probing signal template.

of collecting labeled training data at dense locations incurs high overhead.

Besides location sensing, acoustic echo has been used for other applications. A detailed review can be found in [21].

■ *SLAM:* A SLAM system can construct the indoor map and localize the user device simultaneously based on a certain signal. Here, we review existing SLAM systems according to the used sensing modalities. Radar SLAM [22], mmWave SLAM [23], and Lidar SLAM [24] are based on point clouds generated by radar, mmWave and high-profile lidar, which are unavailable on most smartphones. Visual SLAM [25] uses imaging for landmark detection and map construction. The imaging may introduce privacy concerns. Wi-Fi SLAM [26] detects the received signal strength indicators (RSSIs) from nearby Wi-Fi access points. However, Wi-Fi RSSI is time-varying. Geomagnetic SLAM [27] exploits the spatially varying magnetic field. Electromagnetic radiation (EMR) SLAM [28] uses the smartphone's earphone as a side-channel sensor to sense the EMR from the powerlines. However, side-channel sensing may experience weak signal strength if the earphone is far from the powerlines. This paper employs geomagnetic, EMR, and Wi-Fi SLAMs as the main baselines for evaluation.

Compared with the previous work [21], several major extensions are made in this paper. First, a new model pre-training scheme based on incremental learning is proposed. The new training scheme addresses the lack of training data and allows the model to be updated incrementally with small datasets. Second, the impact of model pre-training and fine-tuning on ELF-SLAM is comprehensively evaluated. The results show that both steps are essential to learn effective ELFs. Third, an in-depth analysis of the spatial resolution of WiFi RSSI, geomagnetism, and ELFs is provided. Lastly, a new case study of room geometry reconstruction based on the rectified trajectory and the echoes is presented. The room geometry reconstruction performance is shown to outperform the existing echo-based room reconstruction systems.

III. MEASUREMENT STUDY

A. Signal Design and Processing

■ *Probing signal:* In this paper, we use a smartphone to emit a near-inaudible logarithmic chirp sweeping the 15–20 kHz band within 10 ms as the probing signal for location sensing, as shown in Fig. 1(a). The 15–20 kHz frequency range causes little annoyance to humans. A wide bandwidth (i.e., 5 kHz) also benefits pulse compression [29], which helps capture finer-grained spatial features. In addition, we apply a Hanning window on the chirp to reduce the damped oscillation of the speaker and increase the signal-to-noise ratio (SNR) that benefits distance measurement.

■ *Echo extraction:* We develop an application that uses the smartphone's loudspeaker to emit the probing signal and microphone to record the 100 ms echo at 44.1 ksps. We assume the smartphone is held around 30 to 40 cm in front of the chest. Ideally, the speaker and the microphone should be unobtrusive. In the received data, we discard the first 10 ms due to the direct propagation from the loudspeaker to the microphone. We also discard the subsequent 1 ms data, which usually contains the echoes reflected by the human body that is around $30-40$ cm apart from the smartphone. The subsequent 50 ms data, which are collectively referred to as *echo trace* and illustrated in Fig. 1(b), are used for location recognition. Fig. 1(c) shows the cross-correlated result between the received signal shown in Fig. 1(b) and the chirp template. The peaks in Fig. 1(c) represent echoes generated from the nearby sound reflectors.

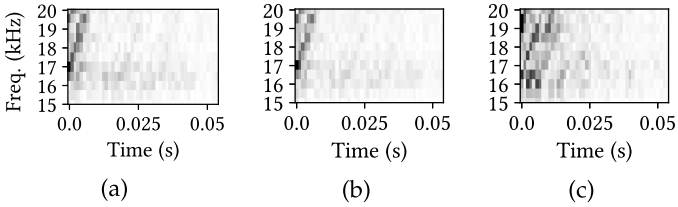


Fig. 2. Spectrograms of echoes received at three spots. The spots of (a) and (b) are 10 cm apart from each other; the spot of (c) is 2 m apart from those of (a) and (b).

■ *Acoustic spectrogram extraction:* We apply short-time Fourier transform (STFT) on echo data to extract the acoustic spectrogram feature. Specifically, a 96-point Hann window with a step length of 48 points is slid on echo data, resulting in a 49×48 spectrogram. The frequency bins that are below 15 kHz are discarded, yielding a 12×48 image as the final result. Fig. 2 shows the generated spectrograms, the spectrograms at far-part locations exhibit significant differences. The horizontal axis represents the echo length, which is 50 ms. The vertical axis represents the chirp frequency, which ranges from 15 to 20 kHz. The noticeable differences among the spectrograms suggest that collected echoes vary at different spots.

B. Spatial Distinctness of Echoes

To gain insights into acoustic echoes, we conduct a set of measurement studies in a $16 \times 28\,\mathrm{m}^2$ lab space. We use a Google Pixel 4 smartphone to excite the space at 128 spots and collect 1,700 data samples at each spot. To understand echo's distinctness limit, we use a supervised learning approach to investigate the achievable spatial resolution and scalability with respect to the number of spots. The analysis results are presented as follows.

■ *Spatial resolution:* For each location, 1,700 spectrograms are collected and split into training and testing data at an 8:2 ratio. The spot's spatial locations are used as ground truth labels. To understand how the inter-spot distance affects location recognition accuracy, we divide the 128 spots into multiple groups with different densities. As a result, the average inter-spot distances of the groups range from 0.25 m to 3 m. For each group, we use spectrograms to train a recognition model. We opt to use the ResNet model [30], which is a popular DNN architecture used for image recognition. Specifically, we select ResNet-18, a ResNet variant that achieves high recognition accuracy while maintaining a relatively low model complexity for echo data. The spot recognition accuracy and the mean localization error are both measured. Fig. 3(a) shows the measured results with respect to the average inter-spot distance. The evaluation is repeated multiple times to get the error bars. The recognition accuracies stay around 90% and the localization errors remain less than 1 m. The results suggest that the acoustic echoes can achieve sub-meter spatial resolution with increased inter-spot distance.

■ *Scalability:* To gain an understanding on acoustic echo's scalability, we gradually increase the number of spots handled by a single DNN model (denoted by k). For each k setting, we randomly draw k spots from the 128 spots, train and test a DNN model. The process is repeated 20 times for each k setting. Fig. 3(b) shows the results. The recognition accuracy gradually decreases with k and becomes flat when k exceeds 100. This result complies with the understanding that the complexity of deep learning increases with the class numbers. The mean localization error also remains within 1 m. The results suggest that a DNN model does not present a bottleneck when the number of spots increases.

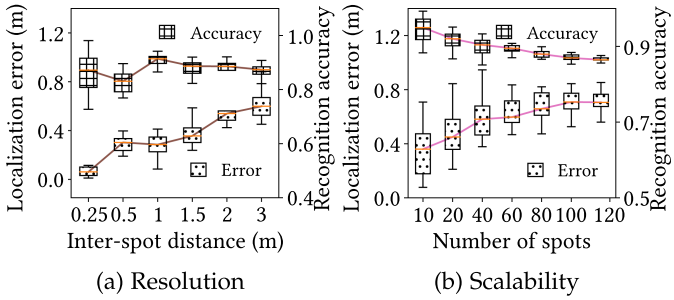


Fig. 3. Acoustic echoes' spatial distinctness.

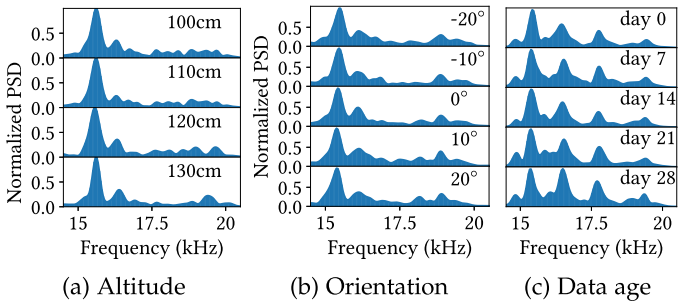


Fig. 4. PSDs of the echoes when several factors vary.

C. Robustness of Acoustic Echoes

This section studies the robustness of acoustic echoes against several potential affecting factors.

■ *Altitude:* We ask a user to hold the phone at different altitudes to simulate the users with different heights. As typical adult heights are within 150–194 cm [31] and we assume the phone is held at two-thirds of user height. The phone's altitude to the ground is around 100 cm to 130 cm. Fig. 4(a) shows the PSDs of the acoustic echoes. We can see that the altitude variations of less than 30 cm introduce little impact on echo PSDs. Hence, the acoustic echoes are robust to the user height and hand altitude variations.

■ *Phone orientation:* As a smartphone's loudspeaker and microphone are not omnidirectional, the received signal at the same spot could be affected by the phone's orientation. Fig. 4(b) shows the echo PSDs when the phone has orientation deviations from $-20°$ to $20°$. The results show that the orientation deviations within $40°$ do not introduce many changes on the collected echoes. However, the echoes exhibit larger differences when

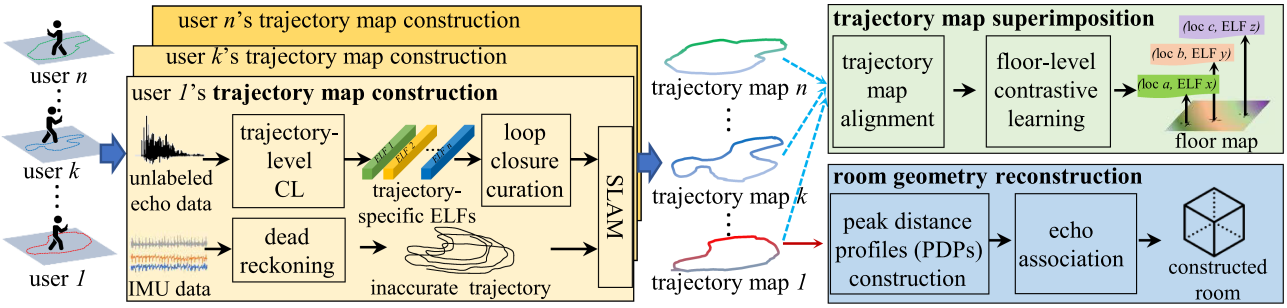


Fig. 5. Overview of ELF-SLAM. It consists of three parts, namely, trajectory map construction, trajectory map superimposition, and room geometry construction.

the orientation deviation increases. Thus, the impact of phone orientation must be taken into consideration when multiple echo traces collected at the same spot are in different orientations. We address the phone orientation issue in Section IV-E.

■ *Temporal stability:* We evaluate acoustic echoes' temporal stability at a fixed location over one month. The layout of the room has no significant changes in this period. From Fig. 4(c), the echo PSDs remain consistent over time. In practice, the constructed floor map can be updated whenever a user contributes a trajectory map. In Appendix B, available online, we further evaluate the impact of significant layout changes on our system and a mitigation approach beyond map update.

IV. Design of ELF-SLAM

The spatial distinctness of acoustic echoes shown in the measurement study is the basis of the fingerprint approach. To unleash the fingerprint approach from laborious training data collection, we design ELF-SLAM based on acoustic echoes and IMU data captured by a smartphone.

A. Approach Overview

Fig. 5 illustrates the overview of ELF-SLAM. The mapping phase of ELF-SLAM consists of *trajectory map construction* and *trajectory map superimposition:* The former focuses on a single trajectory and the latter combines available trajectories such that the combined map can cover the mostly visited spots in the target space.

■ *Trajectory map construction:* The acoustic echoes and IMU data are collected simultaneously using the developed program on the phone. The IMU-based dead reckoning [32] is used to reconstruct the user's trajectory and loop closures are detected using the collected acoustic echoes. The dead reckoning relies on the regulation provided by the loop closure to combat its long-run drifting problem. Due to the ineffectiveness of generic acoustic features, ELF-SLAM extracts a custom ELF using CL for loop closure detection. This trajectory-level CL consists of model *pre-training* that is based on incremental learning and *fine-tuning* using genuine data collected in the target space. ELF-SLAM detects loop closures based on a proposed similarity metric called echo sequence similarity (ESS) between two sequences of ELF traces. Then, a clustering-based approach is developed to remove the false positive loop closures. Lastly, a graph-based optimization constructs an accurate trajectory map of ELFs for the user.

■ *Trajectory map superimposition:* A unified floor map will be obtained after superimposing multiple trajectory maps through crowdsensing. The superimposition reconciles different trajectory maps' ELFs that are collected at the same spot but in different phone orientations. To achieve this, different users' trajectory maps are first aligned into a common coordinate system. The alignment can be achieved based on the initial positions of the users (e.g., the room entrance) and/or prior knowledge about the accessible passages of the target indoor space [33]. Then, we apply the floor-level CL to train a floor-wide ELF extractor using the acoustic data from all trajectory maps. Thus, the floor map covers all spots on the available trajectory maps, where each spot is associated with a unique floor-level ELF.

B. Graph-Based SLAM Formulation

Graph-based SLAM [34] constructs a graph with nodes representing the agent's poses and edges representing the kinetic constraints relating two poses. In this paper, by letting $\mathbf{x}_k$ denote the node (i.e., location) corresponding to the kth detected footstep based on the IMU data, the acoustic echo trace captured between the kth and $(k+1)$th footsteps is the measurement associated with the node $\mathbf{x}_k$ and used to detect whether $\mathbf{x}_k$ is at the same location as any previous node (i.e., loop closure detection). The edge connecting two nodes is associated with the IMU-based odometry. The user trajectory is estimated via graph-based optimization after the loop closures are identified. The estimation method is as follows. For a total of N detected footsteps, let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote the sequence of nodes describing the user trajectory and $\mathbf{u}_{i,j}$ denote the edge constraint between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. Let $\mathcal{C}$ denote the set of footstep index pairs of the detected loop closures. The essence of the trajectory reconstruction can be described by:

$$\mathbf{X}^* = \operatorname{argmin} \sum_{\forall i \in [1,\ldots,N-1]} \left\| f\left(\mathbf{x}_i, \mathbf{u}_{i,i+1}\right) - \mathbf{x}_{i+1} \right\|^2 + \sum_{\forall \langle i,j \rangle \in \mathcal{C}} \left\| f\left(\mathbf{x}_i, \mathbf{u}_{i,j}\right) - \mathbf{x}_j \right\|^2,$$

where $\|\mathbf{x} - \mathbf{y}\|$ denotes the euclidean distance between $\mathbf{x}$ and $\mathbf{y}$, $f(\mathbf{x}_i, \mathbf{u}_{i,j})$ represents the prediction of $\mathbf{x}_j$ based on $\mathbf{x}_i$ and $\mathbf{u}_{i,j}$.

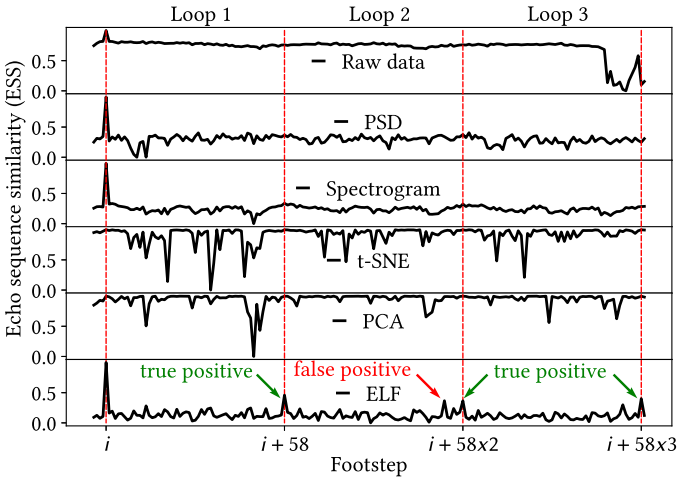


Fig. 6. ESS traces with respect to footstep i. Peaks indicate loop closures at footstep $i + 58$, $i + 58 \times 2$, and $i + 58 \times 3$.

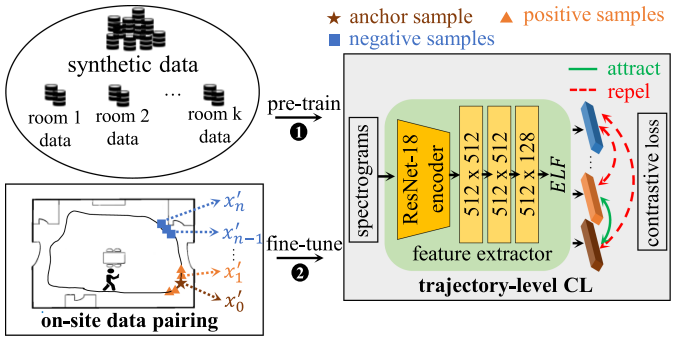


Fig. 7. Trajectory-level CL to learn trajectory-specific ELFs.

In this paper, the SLAM algorithm is implemented using a general graph optimization framework [35], which also addresses the uncertainty of the prediction $f(\mathbf{x}_i, \mathbf{u}_{i,j})$.

C. ELF for Loop Closure Detection

Constructing an effective feature embedding for loop closure detection is critical to SLAM. In this section, we first demonstrate the ineffectiveness of the generic features and then propose using CL to construct a learning-based feature.

1) Ineffectiveness of Generic Features.: We conduct a controlled experiment to evaluate several generic acoustic features' performance on loop closure detection. A user is asked to walk 4 rounds to collect the echo data in a lab space. Each round consists of the same 58 footsteps. We extract the following features of the echo data: PSD, spectrogram, t-SNE, and PCA. Then, we compute the similarity between the features collected at footstep i in the first round with those at all footsteps in all rounds. The echo sequence similarity (ESS) is used as the metric to compute the similarity of the echo data obtained at two footsteps, which is defined as follows. For two footsteps i and j at which K_i and K_j numbers of echoes are collected, the ESS between them is obtained by averaging the $K_i \times K_j$ pair-wise cosine similarity among the two sets of echoes. Fig. 6 shows the resulting ESS traces with footstep i (where $i = 4$) in the first round and j being all footsteps of all rounds sequentially. In this experiment, for footstep i, loop closures are formed at the footsteps $i + 58$, $i + 58 \times 2$, and $i + 58 \times 3$. If the used feature is effective, ESS peaks should be observed at these footsteps. However, from the plots in the first five rows of Fig. 6, no salient peaks are observed. This suggests that the raw data and the used generic features are ineffective for loop closure detection. Note that although t-distributed stochastic neighbor embedding (t-SNE) [36] is effective for finding feature embeddings of clustered data, it is ineffective on the echo samples collected in the spatial continuum that do not exhibit clustered patterns.

The ineffectiveness of generic features for loop closure detection motivates us to apply CL to construct a custom feature, i.e., ELF. CL [37] is a popular unsupervised learning technique that aims to learn useful representations from unlabeled data. CL maximizes the agreement between similar samples while minimizing the agreement between dissimilar samples during the model training. The quality of feature learning relies on the effectiveness of data pairing, which constructs similar samples and dissimilar samples from unlabeled data.

In what follows, we present our CL design to learn the ELF for loop closure detection. Fig. 7 depicts the workflow, which consists of three steps, i.e., *data pairing*, *model pre-training*, and *model fine-tuning*.

2) Learning-Based ELF: Data pairing forms similar samples and dissimilar samples needed by CL. The spatial perturbations for similar sample construction, such as resizing, cropping, and blurring in image recognition tasks may destruct the subtle location-related information embedded in the echo signal. Our data pairing design is based on the empirical observation that the echoes are similar if collected at close locations and different if collected at locations apart. This is illustrated in Fig. 2, the spectrograms of the acoustic echoes received at two nearby spots exhibit similar patterns, whereas the spectrogram of the echo received at a faraway spot is different. Thus, we construct similar data samples using echoes collected at close locations and dissimilar data samples using echoes at locations apart. Specifically, echoes collected consecutively are treated as similar samples. For each training step, we randomly select 256 pairs of similar samples as a training batch from the entire dataset. According to our design in Section III, the time difference between two consecutive echoes is 0.1 s, the spatial distance between these two echoes is about 0.14 m. This average separation is smaller than the achievable spatial resolution of the echo modality as evaluated in Section III. Thus, using two consecutive echoes during the user's movement as similar samples is a good heuristic. The data from different pairs are treated as dissimilar samples.

Model pre-training exploits CL to build a basic ELF extractor, which will be specialized by the model fine-tuning step. CL often requires abundant unlabeled training data to learn useful feature representation. However, there is a lack of publicly available echo data for model pre-training. To address this issue, we propose an incremental learning-based model pre-training scheme. It consists of two steps. **First**, we utilize a room acoustics simulator, `pyroomacoustic` [38],

to generate a substantial amount of simulated training data. The `pyroomacoustic` package offers an intuitive application programming interface (API) for simulating sound reverberation within indoor environments. Specifically, a `SoundSource`, a `MicrophoneArray`, and a `Room` are constructed for data collection. The `SoundSource` emits the designed inaudible chirp, while the `MicrophoneArray` records the sound in the constructed room. The `pyroomacoustic` employs the image source modeling method to simulate sound propagation in indoor spaces. The `SoundSource` and the `MicrophoneArray` are placed 15 cm apart to mimic their relative positions on a smartphone. Multiple rooms with varying configurations are generated to enhance location-related data diversity. Within each of these rooms, extensive echo data is collected at fine-grained points to cover a wide range of locations. We adhere to the procedures outlined in Section III-A to extract the spectrogram feature and train a base feature extractor using CL. This enables us to obtain a base model capable of discerning locations effectively. **Second**, we adapt the pre-trained model to real indoor environments by incrementally updating the feature extractor when new training data are contributed by new users. Though using simulated data for model pre-training allows us to obtain a location-aware feature extractor, the simulated and the real captured data still exhibit differences as the simulator cannot fully model room conditions (e.g., the wall reflection/absorption coefficients, small reflectors like chairs, tables, etc) and the audio hardware characteristics of a smartphone. By incrementally updating the pre-trained model using collected echo data, it can gradually learn environmental and hardware-related features. Therefore, the pre-trained model can be fine-tuned to a new environment more quickly and accurately. The beneficial performance of the proposed model pre-training scheme is presented in Section VI-D.

The architecture of the feature extractor is adapted from [39], which consists of a ResNet-18 encoder and a 3-layer projection head, the model architecture is shown in the right part of Fig. 7. The input of the model is the echo's spectrogram and the output is a 128-dimensional ELF. We minimize the following contrastive loss for model training:

$$\ell_{i,j} = -\log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right)/\tau\right)}{\sum_{k=1}^{2M} \mathbf{1}_{[k\neq i]} \exp\left(\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_k\right)/\tau\right)},$$

where $\mathbf{1}_{[k\neq i]} \in \{0, 1\}$ is evaluated to 1 if and only if $k \neq i$, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, $\boldsymbol{z}$ is the feature vector, i and j indicate a similar data pair, M is batch size, and τ is the temperature parameter. With the above contrastive loss, the pre-training increases the feature similarity for echoes at close locations and decreases the feature similarity for those at locations apart. As a result, the loop closure detection can be implemented by comparing the ELFs in terms of the cosine similarity.

Model fine-tuning uses limited unlabeled data collected by users in the target space to adapt the pre-trained model such that the environment-specific characteristics can be captured. We follow the same CL procedure described above and construct the data pairs using genuine data for model fine-tuning. The resulting model can generate trajectory-specific ELFs in the target space.

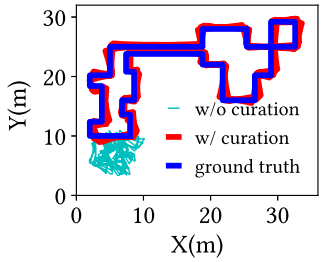


Fig. 8. Reconstructed traj w/wo loop closure curation.

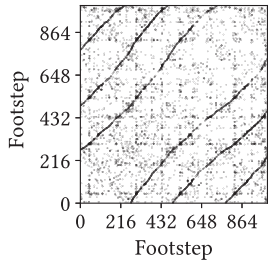


Fig. 9. ESS matrix with trend curves.

The last row of Fig. 6 shows the ESS trace computed using ELF. Peaks at footstep $i + 58$, $i + 58 \times 2$, and $i + 58 \times 3$ marked by the green arrows are effectively detected loop closures. A visualization of ELF is shown in Appendix A.1, available online. However, an unexpected peak close to the footstep $i + 58 \times 2$ marked by a red arrow is also observed. It represents a false positive loop closure based on ELF. Unfortunately, the SLAM is often sensitive to false positive loop closures – a small number of false positives can degrade the SLAM performance [28]. Thus, a loop closure curation algorithm is needed to remove the false positives.

D. Loop Closure Curation

We propose a clustering-based algorithm which is based on the ESS matrix defined as follows to curate the loop closures.

ESS matrix: Consider a user's trajectory that consists of N footsteps. The pair-wise ESSs between any two footsteps form a $(N-1) \times (N-1)$ ESS matrix (0 indexed), where the (i, j)th element is the ELF-based ESS between the footsteps i and j. Thus, the ESS matrix is symmetric. Two ELFs have a high similarity if a large ESS is observed, signaling a potential loop closure. We apply a threshold value of 0.4 to identify most true positives while capturing acceptably low false positives to be removed shortly. The ESS matrix is binarized by the threshold, where the positive elements represent loop closure candidates. Fig. 9 shows the constructed ESS matrix using the ELFs collected in a shopping mall. Both horizontal and vertical axes are footstep numbers. The black dots in the ESS matrix represent the positive elements.

Clustering-based approach for loop closure curation: The goal of loop closure curation is to remove the false positives

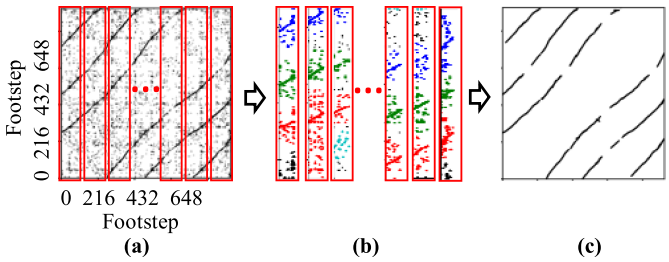


Fig. 10. Loop closure curation: (a) Slicing, (b) clustering in each slice, and (c) concatenated line regression results.

from the ESS matrix. The true positives form trend lines in the binarized ESS matrix due to the user movement. For example, consider an ideal case in which the user walks at a constant speed, the true loop closures of footsteps 0, 1,..., and 10 are footsteps $0+L$, $1+L$,..., and $10+L$, where L is the loop length. As a result, the $(0, 0+L)$th, $(1, 1+L)$th,..., and $(10, 10+L)$th elements of the ESS matrix should be positives and form a trend line. In contrast, the false positives tend to appear at random positions, as shown in Fig. 9. This observation inspires us to propose a clustering-based approach to isolate the true positives from the scattered false positives, which is described as follows.

First, we slice the ESS matrix at a length of 16 footsteps as illustrated in Fig. 10(a). With the slicing, it is easier to identify the true positive clusters in each slice. Then, we apply the DBSCAN clustering algorithm [40] to identify the clusters. This is illustrated by Fig. 10(b), where the clusters are differentiated by colors. Although some false positives are classified by DBSCAN as outliers, the remaining false positives close to the trend curves are still in the clusters. To remove these false positives, we apply the RANSAC [41] linear regression algorithm to detect a line approximating the trend curve in each cluster. RANSAC is a preferred regression algorithm when there are many outliers. Concatenation of the regressed lines across all slices form the clean trend curves as shown in Fig. 10(c). The trend curves formed by the positives are effectively isolated from the scattered noises. Since the negative impact of a false positive on SLAM outweighs that of a false negative, we further curate the loop closures by only retaining the positives that conform to the symmetric property. Specifically, if the positive at the $(i, j)^{\text{th}}$ of the ESS matrix has no counterpart positive at the $(j, i)^{\text{th}}$, the positive is excluded.

Fig. 8 shows the necessity of loop closure curation. First, we use all positives in Fig. 9 as the loop closure information to construct the trajectory. The plot labeled "w/o curation" in Fig. 8 shows the constructed trajectory. We can see that the false positives devastate the trajectory optimization. The plot labeled "w/ curation" in Fig. 8 shows the trajectory using the curated loop closures. Its shape is close to the ground truth as marked by the blue line. The result demonstrates the effectiveness of the proposed loop closure curation.

E. Trajectory Map Superimposition

A single user's trajectory maps have limited coverage in a room. For real applications, it is desirable to combine trajectory maps from many users to form a floor map to cover most/all accessible locations of an indoor space. We assume that the absolute starting position of each trajectory map can be known. In practice, location tagging [10] can be used to recognize the actual entrance. With the known absolute position, the trajectory maps can be collated into a common coordinate system. However, the trajectory maps crossing the same spot from different directions have different echoes, due to the echoes' dependency on phone orientation. Thus, we need to reconcile such differences.

We propose a floor-level CL approach to train a unified feature extractor for map superimposition. It shares the same model pre-training workflow as the trajectory-level CL except the data pairing approach for model fine-tuning. Specifically, the echo data collected at the same location regardless of the phone orientation are treated as similar pairs, whereas those collected from different locations are treated as dissimilar pairs. The model trained via the floor-level CL outputs the floor-level ELFs covering spots from all trajectory maps. As the quality of the floor map is related to its spatial coverage, this floor-level CL approach needs to scale well with the number of locations. In Section VI, we evaluate this approach in handling 4,000 fine-grained spots with four phone orientations at each location.

When falling back to the scheme of learning a location recognition model using supervised learning, a possible approach to mitigate the echo data's sensitivity on phone orientation is to construct a training dataset with echo data and location labels (regardless of orientation) from all the trajectory maps. In Section VI, we evaluate the localization performance of this supervised learning approach with our approach in terms of the quality of the floor map.

F. Localization

When a trajectory map or floor map is available, a smartphone's location can be determined by capturing the echoes in response to the chirps. We consider two localization approaches, i.e., *one-shot localization* and *trajectory localization*, depending on whether the user is standing still or walking. In the former, an ELF sequence containing multiple consecutive echoes collected at a spot is matched against the map in terms of the ESS to determine the location. In the latter, both the ELF sequence and the IMU data during the user's movement over a short time period are used for localization. Specifically, we apply dead reckoning to the IMU data to estimate the user's trajectory, and then apply a curve matching algorithm [42] to find the candidate segments in the map that resemble the user's trajectory. Among the candidate segments, the one with the largest average ESS from the captured ELF sequence is the output of the trajectory localization.

V. Room Geometry Reconstruction

Accurate smartphone-based room geometry sensing is desirable for indoor navigation systems, virtual/augmented reality applications and network condition prediction, etc. In this section, we use the reconstructed user trajectory and the collected acoustic echoes to construct the contour of a polyhedron room with a fixed height. Specifically, the user is required to walk

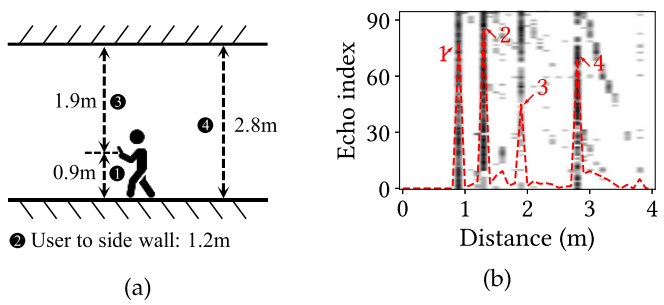


Fig. 11. (a) User holds a phone and moves along a side wall at 1.2 m away. (b) Constructed peak distance profile.

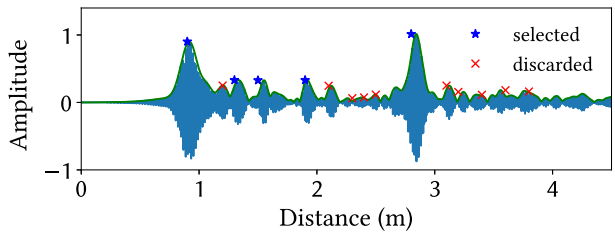


Fig. 12. Peaks detection on the cross-correlated signal.

along the sidewalls and form a complete loop. After the IMU trajectory is rectified using the *trajectory map construction*, the wall distances are estimated using the acoustic echoes. Next, the room geometry is determined by the trajectory and the estimated wall distances. In what follows, we present our room reconstruction procedures.

A. Wall Distance Measurement

We conducted a measurement study to verify if the recorded echoes are effective for measuring the phone-wall distances. A user is asked to hold a smartphone and walk along a sidewall in a living room for a few meters. A total of 95 echo traces are collected in the experiment. Fig. 13(a) shows the layout of the tested environment, where sofas, a table and a TV occupy the room. Fig. 11(a) shows the user's distances to the walls: the *phone-floor*, *phone-sidewall*, and *phone-ceiling* are 0.9 m, 1.2 m, and 1.6 m, respectively. The *room height* is 2.8 m.

Peaks selection: We cross-correlate the received signal with the chirp template to find the peaks generated by main reflectors in a room (e.g., walls, floor and ceiling). Fig. 12 shows an example, where the peaks represent the main reflectors. To determine the peaks' index, we apply the envelope detector on the correlated signals and search for the local maximas. The conversion of peaks' index to distance is calculated by $\frac{nc}{2f}$, where n is a peak's index, c is the speed of sound in air, f is the microphone's sampling rate (i.e., $44,100$ Hz). The result represents the estimated distance between the reflector and the smartphone.

The cross-correlated signals contain many peaks generated by nearby objects. It is difficult to associate each peak with its corresponding reflector. However, for room reconstruction, we only need to identify peaks from the main reflectors, e.g., side walls, the ceiling, and the floor. Our peak selection procedure is as follows. First, we normalize the cross-correlated signal to between 0 and 1. Next, we discard peaks with an amplitude less than 0.15, as they are generated from smaller objects and can be safely disregarded. In Fig. 12, the peaks marked by stars are retained and red crosses are discarded. We further discard peaks beyond 4 m from the smartphone. This is based on the assumption that when a user walks along the sidewalls for data collection, the smartphone's distance to the sidewalls, ceiling, and floor can be maintained within 4 m. In addition, the peak candidates with a distance larger than 4 m are generally caused by the multi-path reflections, which are difficult for object association.

Peak distance profile: In each echo trace, we apply the extraction and selection procedures described above to generate the candidate peaks. Then, we stack peaks selected from all echo traces to form a peak distance profile (PDP) as shown in Fig. 11(b). The horizontal axis represents the peaks' distance to the smartphone and the vertical axis represents the echo's index. The grayscale intensity represents the peaks' amplitude, which ranges from 0.15 to 1. A darker dot represents a higher amplitude. In Fig. 11(b), four vertical lines are formed by the peaks. Lines 1, 2, and 3 are located at the distances of 0.9 m, 1.2 m, and 1.6 m, respectively. These lines correspond to the distances of *phone-floor*, *phone-sidewall*, and *phone-ceiling*. Note that line 4 located at 2.8 m is also observed, whose distance is equal to the *room height*. This line is generated by the echoes that travel a full round in the vertical direction of a room (i.e., via smartphone $\rightarrow$ floor $\rightarrow$ ceiling $\rightarrow$ smartphone, or smartphone $\rightarrow$ ceiling $\rightarrow$ floor $\rightarrow$ smartphone). The red line in Fig. 11(b) represents the summation of the peaks' amplitude along the vertical axis. We can see that the peaks corresponding to the distances of the *phone-floor*, *phone-sidewall*, *phone-ceiling*, and *room height* stand out in the PDP. The reason is that the walls' distance to the phone remains constant while a user is walking along a sidewall, while other objects' distance changes (e.g., TV, sofas, etc). As shown in Fig. 11(b), although it is difficult to associate the peaks to the objects in a single echo trace, aggregation of echoes collected along a specific wall renders peaks from main reflectors more salient than those of the furniture inside the room. Thus, the PDP is effective to find the phone-wall distances along a sidewall.

B. Room Geometry Reconstruction Procedure

We describe the room geometry reconstruction procedure in this section. As shown in Fig. 5, the room reconstruction consists of *PDPs construction* and *Peaks association*.

PDPs construction: We construct PDP for each sidewall to obtain wall distances. The wall numbers are determined based on the shape of constructed user trajectory. Note that we use the rectified user trajectory to get a more accurate approximation. We track the heading directions of the IMU data and record the sheer direction changes as the corners between walls. The sidewall numbers are equal to the detected corners. Then, we split the echoes into clusters based on the timestamps of the detected corners. Since the IMU data and the echoes are collected

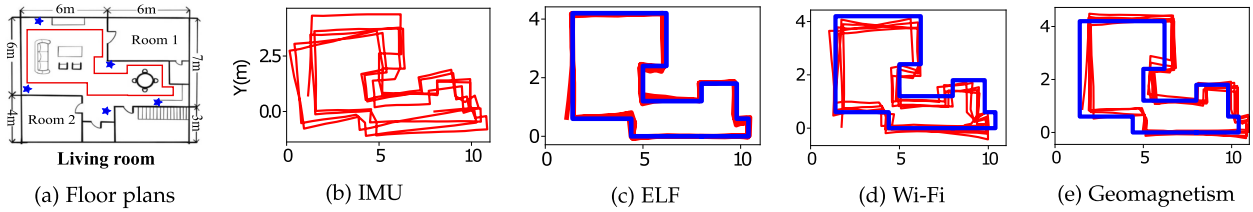


Fig. 13. Floor plans and trajectory reconstruction results in the living room.

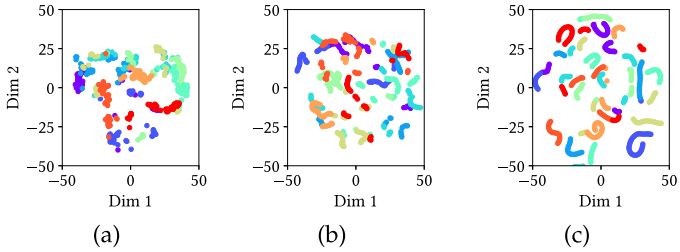


Fig. 14. t-SNE visualization of finetuned feature from different pre-trained models: (a) pre-trained using cross-entropy loss (CEL), (b) pre-trained via CL, using synthetic data (SYN) only, (c) pre-trained via CL, using proposed incremental learning scheme (INC).

simultaneously, each cluster contains the echo traces collected while the user walks along a specific sidewall. We use the echo traces in each cluster to construct PDPs. If the echoes are correctly associated with the *phone-floor*, *phone-sidewall*, *phone-ceiling*, and the *room height* distances in PDPs, the room's geometry is also determined.

Peaks association: To correctly associate the *phone-floor*, *phone-sidewall*, *phone-ceiling*, and the *room height* distances to the peaks in the constructed PDPs, we leverage the knowledge that the *room height* equals to the summation of the *phone-floor* and *phone-ceiling* distances. These three distances generally remain constant while a user holds the phone and moves within the room. Thus, we can determine these three distances in each PDP and then identify the subsequent largest peak as the *phone-sidewall* distance. The procedure is as follows. First, we combine PDPs from all sidewalls to form a unified PDP (u-PDP). Since the *phone-floor*, *phone-ceiling*, and the *room height* distances remain consistent in each PDP, their appearance will be more salient in the aggregated u-PDP. In u-PDP, we identify the echo with the largest peak as *phone-floor* distance. This is because the used bottom microphone for recording is closer to the floor when held by a user. Thus, the peak amplitude at *phone-floor* distance generally has the largest value. We then associate the *phone-ceiling* distance and *room height* by looking for peaks that have the summation relationship with the identified *phone-floor* distance in the u-PDP. To reduce ambiguity, we assume the *phone-ceiling* distance is larger than *phone-floor* distance. Then, we visit each PDP and exclude the echoes that are closest to the identified *phone-floor*, *phone-ceiling* and *room height* distances. The subsequent echo with the largest peak is identified as the *phone-sidewall* distance. Finally, we use the rectified trajectory and the estimated wall distances to determine the vertexes of the polyhedron.

VI. System Evaluation

A. Experiment Setup

Evaluation environments: We evaluate ELF-SLAM in three indoor environments, i.e., a living room ($60\,\mathrm{m}^2$) shown in Fig. 13(a), an office ($360\,\mathrm{m}^2$), and a shopping mall ($2,000\,\mathrm{m}^2$). The floorplans of the latter two can be found in Appendix C, available online. To conduct comparative evaluation side by side, we employ the SLAM systems using two smartphone's built-in sensing modalities, i.e., Wi-Fi RSSI and geomagnetism, as the baselines. This is the same as the evaluation methodology adopted in [28] that studies powerline EMR SLAM. Note that we also compare the results of ELF-SLAM and EMR SLAM. To implement Wi-Fi SLAM, we deploy Wi-Fi access points (APs) in the living room and office, as illustrated by the stars in the floorplans. The shopping mall has dense APs deployed by the tenants. The number of Wi-Fi APs observable is around 5 to 10 when conducting experiments in the mall. Note that random people hung around in the shopping mall during the data collection.

Data collection: We develop an Android app on a Google Pixel 4 smartphone to collect acoustic echoes, Wi-Fi RSSI, geomagnetic field signals, and IMU data. The app uses the available `WifiManager` Android API to scan the Wi-Fi APs and collect RSSI data at a sampling rate of 0.8 sps. We do not use Wi-Fi channel state information (CSI), because CSI sampling requires rooting the smartphone [43]. The app uses the phone's built-in magnetometer to sample the geomagnetic field at 50 sps. During data collection, the smartphone is held around 30 to 40 cm in front of the user's chest. The data is collected by walking on a marked trajectory for multiple rounds in each of the evaluated environments. Note that the purpose of trajectory marking is to obtain the location ground truth.

Loop closure detection for baseline modalities: For Wi-Fi SLAM, we use the euclidean distance between two Wi-Fi RSSI vectors for loop closure detection [28]. For geomagnetic SLAM, we first normalize the triaxial magnetic data and then apply dynamic time warping for loop closure detection [27]. We apply the same loop closure curation and graph-based optimization algorithms on all modalities.

Model training details: The model training is implemented using PyTorch [44]. The model is trained for 200 epochs with

TABLE II
MAPPING ERROR STATISTICS (UNIT: METER)

Modality	Living room			Office			Mall		
	$\tilde{x}^1$	$\bar{x}^2$	Q3[3]	$\tilde{x}$	$\bar{x}$	Q3	$\tilde{x}$	$\bar{x}$	Q3
ELF	**0.10**	**0.10**	**0.14**	**0.63**	**0.63**	**0.80**	**0.45**	**0.53**	**0.69**
ELF w/o pre-train	0.73	0.82	1.25	1.69	1.68	1.94	1.16	1.14	1.42
ELF w/o fine-tune	1.26	1.32	1.82	2.45	2.56	3.07	2.28	2.34	3.44
Wi-Fi	0.44	0.45	0.55	1.52	1.54	2.06	1.24	1.26	1.54
Geomag	0.56	0.55	0.64	1.14	1.24	1.82	0.79	0.81	1.05

[1]Median error, [2]Mean error, [3]Third quartile of the error

TABLE III
LOCALIZATION ERROR STATISTICS (UNIT: METER)

Modality	Living room			Office			Mall		
	$\tilde{x}^1$	$\bar{x}^2$	Q3[3]	$\tilde{x}$	$\bar{x}$	Q3	$\tilde{x}$	$\bar{x}$	Q3
One-shot localization									
ELF	**0.10**	**0.29**	**0.14**	**0.54**	**0.60**	**0.80**	**0.42**	**0.79**	**0.67**
Wi-Fi	1.67	2.17	3.30	3.44	4.27	6.16	3.04	3.86	5.22
Geomag	1.06	2.31	3.93	2.19	3.95	5.38	12.5	13.4	19.3
Trajectory localization									
ELF	**0.10**	**0.22**	**0.64**	**0.41**	**0.54**	**0.97**	**0.47**	**0.53**	**0.86**
WiFi	0.54	0.73	1.13	1.74	1.86	3.09	1.46	1.90	3.73
Geomag	0.56	0.56	0.78	1.75	1.81	2.39	8.70	8.29	14.6

[1]Median error, [2]Mean error, [3]Third quartile of the error

a batch size of 256. The learning rate is set to 0.0001. The temperature parameter τ is set to 0.1. The model is trained on a workstation equipped with two NVIDIA GeForce RTX 2080 Ti GPUs. The model pre-training and fine-tuning are implemented using the same hyperparameters. The model training time depends on the used data volume. On our workstation, the model training time is around 15 minutes when the model is trained on 120 minutes of data.

B. Trajectory Map Construction Performance

Fig. 13 shows the map construction results of three modalities in the living room. The results of the office and the shopping mall can be found in Appendix C, available online. The trajectories reconstructed by ELF-SLAM are the closest to the ground truth among the three modalities in all evaluated environments. Table II lists the detailed mapping error statistics. ELF-SLAM achieves sub-meter mapping accuracy in all environments, whereas Wi-Fi SLAM and geomagnetic SLAM's mapping errors increase in the large indoor space, i.e., office and mall. In [28], EMR SLAM using the smartphone earphone as the side-channel sensor yields about 1 m to 2 m median mapping errors in the evaluated office and lab spaces. Thus, ELF-SLAM outperforms Wi-Fi SLAM, geomagnetic SLAM, and EMR SLAM in map construction.

C. Localization Performance

We evaluate both the one-shot localization and trajectory localization of the three sensing modalities. Table III lists the localization error statistics. For one-shot localization, ELF-SLAM achieves sub-meter median error in the three environments and outperforms both Wi-Fi SLAM and geomagnetic SLAM. For trajectory localization, each short trajectory consists of 8 consecutive footsteps. For Wi-Fi and geomagnetic SLAMs, the trajectory localization errors are less than the one-shot localization errors. For ELF-SLAM, trajectory localization does not bring much accuracy improvement over one-shot localization, because the latter has already achieved a high localization accuracy.

We also conduct experiments in the living room to study the impact of various affecting factors on ELF-SLAM, including nearby people, audible noises, and space layout changes. The results can be found in Appendix B, available online.

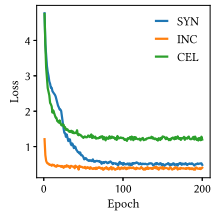


Fig. 15. Fine-tuning loss trend.

D. In-Depth Analysis

1) Impact of Model Pre-Training: We investigate the necessity of the CL pre-training by comparing the trajectory map reconstruction performance using the ELF extractors learned with and without the model pre-training step. The row "ELF w/o pre-train" in Table II is for the case without model pre-training. Compared with the result with model pre-training (row "ELF"), the median mapping errors increase to 0.73 m, 1.16 m, and 1.69 m in the three environments, respectively. This result shows that model pre-training is essential to learn effective ELFs. We further evaluate the effectiveness of the proposed incremental learning-based scheme. We visualize the fine-tuned feature embeddings by applying different pre-trained schemes. The first scheme adopts cross-entropy loss (CEL) and applies supervised learning for model pre-training. The second scheme applies CL with synthetic data (SYN) only for model pre-training, and the third scheme applies CL with the proposed incremental learning (INC), i.e., incrementally updating the pre-trained model with real data. Fig. 14 shows the t-SNE [36] visualization of fine-tuned features, where (a), (b), and (c) corresponds to three pre-training schemes, respectively. The ELFs are extracted from the data collected in the office and colors represent different locations. We can see that the feature embeddings learned using the proposed incremental learning scheme are more compact and distinct than those learned using the other two schemes. The pre-training scheme based on the CEL yields the worst result. We further investigate the finetuning loss using different pre-trained models, the results are shown in Fig. 15. It is observed that finetuning using the model pre-trained from the proposed incremental learning converges fastest and yields the smallest loss, whereas the finetuning from the model pre-trained by CEL yields the largest loss. This observation shows that the proposed incremental contrastive learning is effective to learn location-dependent features and the pre-trained model can be finetuned with fewer epochs. The model trained via CEL does not generalize well if the labeling information is missing.

We also investigate how much data is needed by the proposed incremental learning scheme to achieve optimal performance. We first train the model using the synthetic data only. Then, we

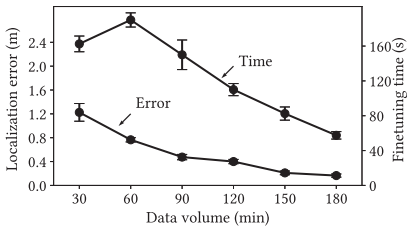


Fig. 16. Evaluation of fine-tuning time and localization performance.

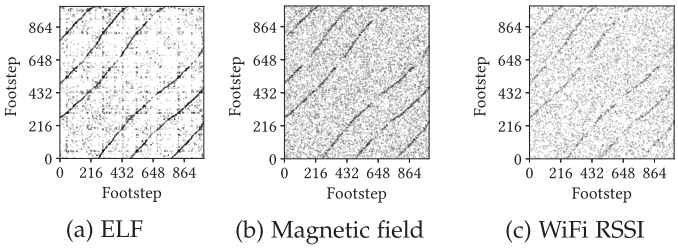


Fig. 17. ESS matrixes of different modalities in the mall.

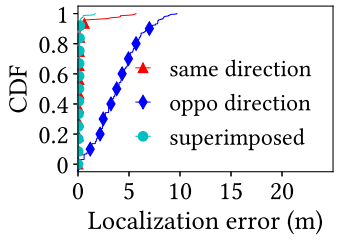


Fig. 18. Superimposition of small-scale real echoes.

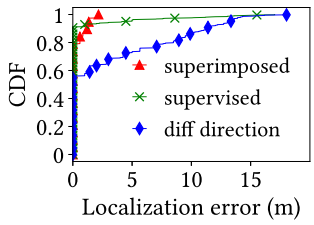


Fig. 19. Superimposition of large-scale synthetic echoes.

update the model by incrementally adding the real data. Fig. 16 shows the fine-tuning time and localization errors versus the model pre-training data volume. The data used for fine-tuning is collected in the living room and the total length is around 10 minutes. The results show that the localization error decreases as the model is pre-trained using more data. The localization error saturates when pre-training data reaches 120 minuntes. In addition, the required model fine-tuning time also decreases as it requires fewer training epochs to converge.

2) Impact of Model Fine-Tuning: We investigate the impact of model finetuning on trajectory map construction performance. The row "ELF w/o fine-tune" in Table II is for the case without model fine-tuning. The ELFs extracted without model fine-tuning cannot provide any loop closure information to be used by the SLAM optimization algorithm. The results are reported using the un-rectified IMU trajectory. Thus, model fine-tuning is essential to learn effective ELFs.

3) Spatial Distinctness of Different Modalities: We analyze the spatial distinctness of ELF, WiFi RSSI and geomagnetic field. We construct the ESS matrixes of three modalities using the data collected in the shopping mall. Fig. 17 shows the results. In each ESS matrix, the true positive loop closures form the trend curves and the false positive loop closures appear as random noises. We compare the true positives and the false positives among different modalities: the number of true positives detected using geomagnetic field and WiFi RSSI is around 75% and 40% of that of the ELF. Meanwhile, the number of false positives detected using the geomagnetic field and the WiFi RSSI are around 10 and 4 times higher than that of the ELF. ELF generates more true positive loop closures and fewer false positives compared with the other two modalities. Thus, ELF is more spatial distinct than the geomagnetic field and the WiFi RSSI and achieves the best SLAM performance.

E. Trajectory Map Superimposition

We use one-shot localization to evaluate the performance of map superimposition as described in Section IV-E.

Evaluation on a small-scale dataset: The experiments are conducted in the living room. We follow the marked trajectory in Fig. 13(a) and walk in two opposite directions to generate two different trajectory maps. Then, we apply the proposed map superimposition to obtain a unified map and evaluate the localization performance. Fig. 18 shows the results. The plot labeled "same direction" is obtained when the smartphone's orientation at the localization phase is the same as the used map. The median localization error is 0.1 m. The plot labeled "oppo direction" is for the case when the smartphone's orientation at the localization phase is different from the trajectory map. The median localization error increases up to 3.8 m. The increased error is caused by the phone orientation deviations. The plot labeled "superimposed" shows the localization results using the proposed map superimposition via the CL. The median localization error is 0.1 m, which is the same as the "same direction" result. This small-scale experiment in the living room shows that the CL-based map superimposition can improve the ELF-based localization performance when trajectory maps are constructed using echoes from opposite directions.

Evaluation on a large-scale synthetic dataset: We also evaluate whether the proposed map superimposition is scalable to handle massive echo data when many trajectory maps are available. We omit the trajectory map construction step and only focus on evaluating the superimposition performance. Similarly, we evaluate the one-shot localization performance on the constructed floor map. To allow a large-scale evaluation, we use the `pyroomacoustic` simulator to generate the synthetic echoes in an indoor space that has a polyhedron shape as shown in Fig. 20(a). The data is collected from 4,000 spots in the grey area. The distance between two neighbor spots is 10 cm. At each spot, we simulate the scenario where the echoes are collected by a phone in different orientations. In Fig. 20(a), the red arrows at spot A represent the simulated orientations. We collect 100 echo

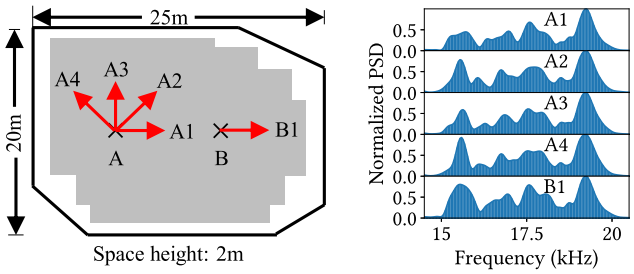


Fig. 20. (a) Simulated space. (b) Spot A's echo PSDs on directions 1 to 4 and Spot B's echo PSD at direction 1.

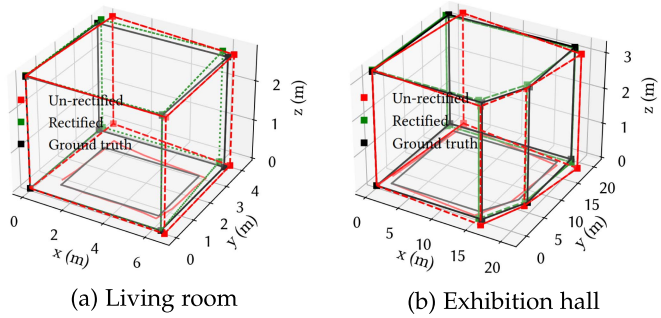


Fig. 21. Room construction results.

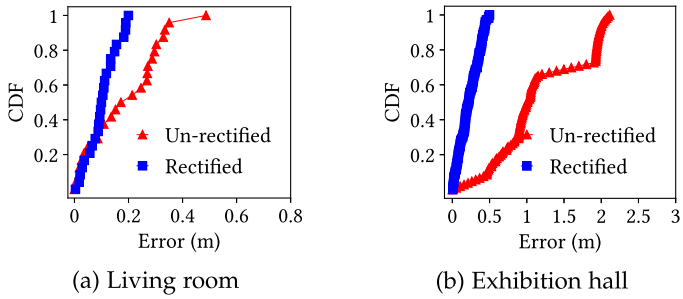


Fig. 22. Room construction performance.

samples for each orientation. We apply random perturbations to the echo data such that they are slightly different. As a result, we generate 16 million echo samples in the simulated room. The first four rows of Fig. 20(b) show the synthetic echoes' PSDs for four directions at spot A. They are slightly different from each other. The last row of Fig. 20(b) shows the echo's PSD at spot B. It is different from all PSDs obtained at spot A. This shows that the simulator can generate both orientation- and location-dependent echoes that can be used to evaluate map superimposition performance. Note that it is infeasible to collect such a large-scale dataset in a real environment. Based on our estimation, collecting the same amount of data in the real world requires about 400 hours of manual labor.

ELF visualization after map superimposition is shown in Appendix A.2, available online. The result shows that map superimposition is effective in reconciling the ELFs' differences due to phone orientations. Fig. 19 shows the localization results on the synthetic data. The plot labeled "diff direction" shows the CDF when the CL-based map superimposition is not applied and the evaluated samples are in a different phone orientation from that in the trajectory map. The mean localization error is 3.2 m. This poor result shows the necessity of differences reconciliation. The plot labeled "superimposed" shows the results obtained using the floor map constructed by the floor-level CL. The mean localization error decreases to 0.24 m. We also employ the supervised fingerprint approach as a baseline, which forms the training dataset by labeling the echoes synthesized at the same spot with the same location label and trains a DNN to classify the 4,000 spots. The CDF curve labeled "supervised" shows the results. The mean localization error is 0.56 m. The supervised fingerprint approach is inferior to the proposed solution that performs localization using the floor map.

F. Room Geometry Reconstruction

Evaluation environments: We conduct experiments in two polyhedron-shape rooms. The first one is a $4 \times 6.5 \times 2.8\,\mathrm{m}^3$ living room filled with furniture like TV and sofas. The second one is an $18 \times 20 \times 3.2\,\mathrm{m}^3$ relatively empty exhibition hall. In each room, a user holds the smartphone and walks along the sidewalls to collect the IMU and the echo data.

Evaluation results: Fig. 21 shows the room reconstructions of both rooms. The polyhedron labeled "Un-rectified" is the estimated room shape using the un-rectified IMU trajectory and the estimated wall distances. This plot represents the essence of [12], [13], where the performance of the mapping relies on the accuracy of the estimated IMU trajectory. The polyhedron labeled "Rectified" is constructed using the rectified IMU trajectory via the ELF-SLAM and the estimated wall distances. The results show that the room geometry constructed upon the rectified user trajectories is closer to the ground truth compared to those constructed using the un-rectified trajectories. We calculate the distances between the constructed and the ground-truth walls to obtain the CDF of the room reconstruction errors. Fig. 22 shows the results. The median construction errors for the "Un-rectified" approach are about 0.32 m, and 1.2 m in the living room and the exhibition hall, respectively. The errors decrease to about 0.15 m and 0.35 m for "Rectified" approach, representing a 2× and 4× error reduction. Thus, our room geometry reconstruction outperforms [12], [13] that rely on the un-rectified IMU results.

Application consideration: Our system requires the user to walk a full loop along the walls of a room. The amount of data needed depends on the size of the room. Considering the average human walking speed of 1.2 m/s, the time needed for data collection is about 18 s and 64 s in the living room and the exhibition hall, respectively. Thus, the data collection for room geometry reconstruction incurs little overhead.

G. System Overhead

We evaluate the computation overheads of the ELF-based SLAM on a Google Pixel 4 smartphone. Specifically, we perform real-time one-shot localization on the floor map constructed by the floor-level CL. To customize the ELF extractor for the phone, we use `Pytorch-Mobile` [45] to optimize and compress

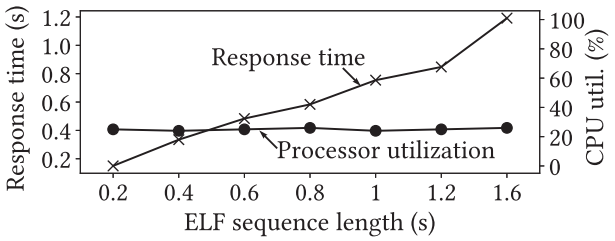


Fig. 23. Model execution overhead.

the model to about 96 MB. The real-time localization module performs one-shot localization using the 2D map.

1) App's Response Time and Processor Utilization.: At the localization stage, the smartphone processor utilization remains at around 20% when we vary the ELF sequence length from 0.2 s to 1.6 s, as shown in Fig. 23. The storage of ELFs requires moderate memory. The disk usage of storing 4,000 spots' ELFs is less than 4 MB. We also measure the app's response time, which includes the ELF's extraction time and feature matching time against the floor map. The App's response time increases from 0.18 s to 1.2 s, when the ELF sequence length varies from 0.2 s to 1.6 s. The increased response time is from the localization phase, because the computation overhead of the feature matching increases with the ELF sequence length. From Appendix B.1, available online, by setting the ELF sequence length to be 0.6 s, our system achieves 0.1 m median localization error, while the corresponding measured response time is about 0.5 s. Thus, the user can get the localization result in about 1.1 s.

2) App's Network Bandwidth and Battery Usage.: The app's bandwidth usage is around 90 kbps while continuously transmitting echo and IMU data to the cloud server for map construction. This data rate is similar to that of Advanced Audio Coding (AAC), a widely adopted standard for lossy audio compression. Note that as the localization phase of ELF-SLAM is performed locally on the phone, it requires no data transmission. We use the `battery historian` [46] to estimate the app's energy usage. The app's energy usage per hour is around 270 mAh when the app performs localization continuously. This energy usage is similar to that of the Google Map app in continuous navigation, i.e., around 280 mAh and much lower than a visual SLAM [47], whose measured energy consumption is around 450 mAh. Thus, our ELF-based localization system introduces acceptable overhead.

VII. Discussion

ELF-SLAM is an acoustic-based indoor location sensing system and its performance can be affected by various factors as evaluated in this paper. We discuss several potential approaches that can be considered to improve the localization system's performance in future work. First, the soft information (SI)-based approach [48], [49], [50] can be employed to enhance the robustness of the system. ELF-SLAM uses hard information (i.e., the estimated distance) for map construction. However, the estimated distance is not always accurate. To address this issue, the SI-based approach considers the uncertainty of the measurement and generates Gaussian distribution to improve the accuracy of the map. Second, different sources of information (e.g., map information, smartphone inertial measurements, geomagnetic field, WiFi RSSI, etc) can be incorporated for cooperative localization [51], [52]. Such a cooperative approach can be more robust to environmental changes and can reduce the uncertainty of the localization as compared with the system developed upon the single modality.

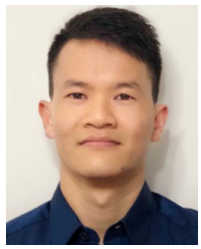
VIII. Conclusion

This paper presents ELF-SLAM, an indoor smartphone SLAM system using acoustic echoes. ELF-SLAM uses a smartphone's audio hardware to emit near-inaudible chirps and record acoustic echoes in an indoor space, then uses the echoes to detect loop closures that regulate the IMU-based dead reckoning. To effectively capture loop closures, we design a trajectory-level contrastive learning procedure and apply it to the echoes to learn ELFs. Then, we design a clustering-based approach to remove the false detection results and curate the loop closures. Third, we apply the rectified trajectory map to reconstruct the room's geometry. Lastly, we design floor-level contrastive learning to superimpose the trajectory maps. Our extensive experiments show that ELF-SLAM achieves sub-meter accuracy in both mapping and localization, and outperforms both Wi-Fi RSSI and geomagnetic SLAMs. The room geometry reconstruction also outperforms the latest echo-based systems.

References

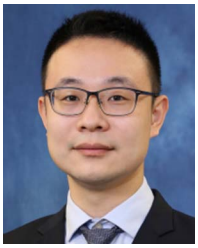
[1] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst., Appl. Serv.*, 2005, pp. 205–218.
[2] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe, "ALPS: A bluetooth and ultrasound platform for mapping and localization," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 73–84.
[3] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes, "Learning and recognizing the places we go," in *Proc. Int. Conf. Ubiquitous Comput.*, Springer, 2005, pp. 159–176.
[4] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 169–182.
[5] C. Zhang and X. Zhang, "LiTell: Robust indoor localization using unmodified light fixtures," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 230–242.
[6] R. Gao et al., "Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment," *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 460–474, Feb. 2016.
[7] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proc. 9th Int. Conf. Mobile Syst., Appl. Serv.*, 2011, pp. 155–168.
[8] S. He and K. G. Shin, "Geomagnetism for smartphone-based indoor localization: Challenges, advances, and comparisons," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–37, 2017.
[9] M. Rossi, J. Seiter, O. Amft, S. Buchmeier, and G. Tröster, "RoomSense: An indoor positioning system for smartphones using active sound probing," in *Proc. 4th Augmented Hum. Int. Conf.*, 2013, pp. 89–95.
[10] Y.-C. Tung and K. G. Shin, "EchoTag: Accurate infrastructure-free indoor location tagging with smartphones," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 525–536.
[11] Q. Song, C. Gu, and R. Tan, "Deep room recognition using inaudible echos," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 1–28, 2018.
[12] S. Pradhan, G. Baig, W. Mao, L. Qiu, G. Chen, and B. Yang, "Smartphone-based acoustic indoor space mapping," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 2, pp. 1–26, 2018.

[13] B. Zhou, M. Elbadry, R. Gao, and F. Ye, "BatMapper: Acoustic sensing based indoor floor plan construction using smartphones," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl. Serv.*, 2017, pp. 42–55.
[14] J. Lian, J. Lou, L. Chen, and X. Yuan, "EchoSpot: Spotting your locations via acoustic sensing," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 3, pp. 1–21, 2021.
[15] S. Shen, D. Chen, Y.-L. Wei, Z. Yang, and R. R. Choudhury, "Voice localization using nearby wall reflections," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–14.
[16] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 261–272.
[17] J. Aparicio, F. J. Álvarez, A. Hernández, and S. Holm, "A survey on acoustic positioning systems for location-based services," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–36, 2022.
[18] W. Huang et al., "Swadloon: Direction finding and indoor localization using acoustic signal by shaking smartphones," *IEEE Trans. Mobile Comput.*, vol. 14, no. 10, pp. 2145–2157, Oct. 2015.
[19] H. Yang et al., "Smartphone-based indoor localization system using inertial sensor and acoustic transmitter/receiver," *IEEE Sensors J.*, vol. 16, no. 22, pp. 8051–8061, Nov. 2016.
[20] J. Ureña et al., "Acoustic local positioning with encoded emission beacons," *Proc. IEEE*, vol. 106, no. 6, pp. 1042–1062, Jun. 2018.
[21] W. Luo, Q. Song, Z. Yan, R. Tan, and G. Lin, "Indoor smartphone SLAM with learned echoic location features," in *Proc. 20th ACM Conf. Embedded Netw. Sensor Syst.*, 2022, pp. 489–503.
[22] J. W. Marck, A. Mohamoud, E. vd Houwen, and R. van Heijster, "Indoor radar SLAM A radar application for vision and GPS denied environments," in *Proc. IEEE Eur. Radar Conf.*, 2013, pp. 471–474.
[23] F. Guidi, A. Guerra, and D. Dardari, "Personal mobile radars with millimeter-wave massive arrays for indoor mapping," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1471–1484, Jun. 2016.
[24] D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3D LiDAR-based online mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5000–5007.
[25] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 1, pp. 1–11, 2017.
[26] A. Arun, R. Ayyalasomayajula, W. Hunter, and D. Bharadia, "P2SLAM: Bearing based WiFi SLAM for indoor robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 2, pp. 3326–3333, Apr. 2022.
[27] S. Wang, H. Wen, R. Clark, and N. Trigoni, "Keyframe based large-scale indoor localisation using geomagnetic field and motion pattern," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1910–1917.
[28] C. X. Lu et al., "Simultaneous localization and mapping with power network electromagnetic field," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 607–622.
[29] P. Lazik and A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," in *Proc. 10th ACM Conf. Embedded Netw. Sensor Syst.*, 2012, pp. 99–112.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[31] 2019. [Online]. Available: https://ourworldindata.org/human-height
[32] M. Angermann and P. Robertson, "FootSLAM: Pedestrian floor without exteroceptive sensors—Hitchhiking on human perception and cognition," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1840–1848, May 2012.
[33] M. Montemerlo et al., "FastSLAM: A factored solution to the floor problem," *J. Mach. Learn. Res.*, 2002.
[34] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, Winter 2010.
[35] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3607–3613.
[36] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[37] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, 2020, Art. no. 2.
[38] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 351–355.
[39] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 22 243–22 255, 2020.
[40] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," *Knowl. Discov. Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.
[41] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[42] M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan, "Curve matching for open 2D curves," *Pattern Recognit. Lett.*, vol. 30, no. 1, pp. 1–10, 2009.
[43] S. M. Hernandez and E. Bulut, "Performing WiFi sensing with off-the-shelf smartphones," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2020, pp. 1–3.
[44] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 8026–8037, 2019.
[45] Pytorch Mobile, 2022. [Online]. Available: https://pytorch.org/mobile/home/
[46] 2017. [Online]. Available: https://github.com/google/battery-historian
[47] 2016. [Online]. Available: https://github.com/sunzuolei/mvo_android
[48] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for Localization-of-Things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Nov. 2019.
[49] A. Conti et al., "Location awareness in beyond 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 22–27, Nov. 2021.
[50] G. Torsoli, M. Z. Win, and A. Conti, "Blockage intelligence in complex environments for beyond 5G localization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1688–1701, Jun. 2023.
[51] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, Feb. 2012.
[52] Z. Liu, W. Dai, and M. Z. Win, "Mercury: An infrastructure-free system for network localization and navigation," *IEEE Trans. Mobile Comput.*, vol. 17, no. 5, pp. 1119–1133, May 2018.



Wenjie Luo received the BEng degree from the School of Electrical and Electrical Engineering at Nanyang Technological University (NTU) in Singapore in 2015 and the PhD degree from the School of Computer Science and Engineering, NTU, in 2023. His research interests include integrating the first principles in the machine learning algorithm to improve the performance of AIoT sensing applications. He was a recipient of the IPSN 2021 Best Artifact Award Runner-up.

Qun Song (Member, IEEE) received the BEng degree in computer science from Nankai University in 2018 and the PhD degree in computer science from Nanyang Technological University in 2022. She is currently an Assistant Professor with the Embedded Systems group of the Faculty of Electrical Engineering, Mathematics and Computer Science at the Delft University of Technology, the Netherlands. Her research interests include cyber–physical systems, sensing, and ubiquitous computing. She was a recipient of the IPSN 2021 Best Artifact Award Runner-up and SenSys 2022 Best Paper Candidate. She serves on the technical program committees of various international conferences related to her research areas including SenSys, ACM e-Energy, and IEEE/ACM CHASE.



Zhenyu Yan (Member, IEEE) received the PhD degree from the School of Computer Science and Engineering, Nanyang Technological University. He serves as a research Assistant Professor with the Department of Information Engineering, The Chinese University of Hong Kong. He has been honored with the Kan Tong Po International Fellowship from the Royal Society in the U.K. and the Rising Star Award from ACM SIGBED China. His work has also been recognized with a Best Paper Award Runner-up at ACM MobiCom 2022 and a Best Artifact Award Runner-up at ACM/IEEE IPSN 2021. His research focuses on the artificial intelligence of things, smart sensing systems, and security and privacy.

Rui Tan (Senior Member, IEEE) received the BS and MS degrees from Shanghai Jiao Tong University, in 2004 and 2007, respectively, and the PhD degree in computer science from City University of Hong Kong, in 2010. He is an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. Previously, he was a research scientist (2012–2015) and a senior research scientist (2015) with Advanced Digital Sciences Center, a Singapore-based research center of University of Illinois at Urbana-Champaign, and a postdoctoral research associate (2010–2012) with Michigan State University. His research interests include cyber-physical systems, sensor networks, and pervasive computing systems. He is the recipient of Best Paper Award, Best Paper Award Runner-Up/Finalist from ICCPS 2022 and 2023, SenSys 2021, IPSN 2014 and 2017, CPSR-SG 2017, and PerCom 2013. He is currently serving as an associate editor of *ACM Transactions on Sensor Networks*. He is the TPC co-chair of e-Energy'23 and EWSN'24, and general co-chair of e-Energy'24. He received the Distinguished TPC Member recognition thrice from INFOCOM in 2017, 2020, and 2022.

Guosheng Lin (Member, IEEE) received the PhD degree from the University of Adelaide in 2014. He is an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are generally in computer vision and machine learning including scene understanding, 3D vision and generative learning.