This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

Enabling "See-and-Point" Communication between Robots

Huangwei Wu, Weiguo Wang, Meng Jin, *Member, IEEE,* Zhuxuan He, Qi Cao, Xinbing Wang, *Senior Member, IEEE,* Chenghu Zhou, *Member, IEEE,*

***Abstract*—This paper proposes a novel address mapping mechanism for multi-robot communication and collaboration systems, named SPing. SPing addresses each robot with a dynamic *physical-world address* - an encoding of the robot's physical location - rather than a pre-assigned digital-world ID (e.g., the IP address). This enables a "see-and-point" communication mode for robots: a robot can establish an immediate connection pointing to any other robot it intends to collaborate with in its visual field, without relying on a pre-existing multi-robot network. This on one hand improves the robustness and usefulness of multi-robot systems in uncertain and unstructured environments where network infrastructures are unavailable. On the other hand, it makes the robots' communication behavior tightly coupled with and more supportive of the collaboration tasks in the physical world. We build an end-to-end prototype of SPing and evaluate its performance in both static and mobile scenarios. The results show that SPing can always establish a connection precisely pointing to the target receiver with an average matching rate of 99.58%, and a spatial resolution of 0.3m∼0.5m.**

***Index Terms*—Vision-based channel estimation, acoustic, multirobot collaboration, connection establishment**

I. INTRODUCTION

Multi-robot collaboration systems have been studied for decades. In most of the existing systems, the collaborative behaviors of robots are either well-scripted for specific tasks (such as assembling and packaging) or centrally controlled based on well-maintained networks [1]–[3]. However, in the incoming AIGC era, robots are expected to complete more challenging tasks, such as disaster rescue, counter-terrorism operations, and post-accident inspection. These tasks typically take place in unknown and even hazardous environments where actions cannot be pre-planned and no existing network infrastructure is available. Under these conditions, ad hoc collaboration becomes essential, where each robot selectively forms partnerships with others based on the instant surrounding status.

However, one limiting factor in achieving an ad hoc collaboration mode is the lack of suitable connection establishment technology. Specifically, traditional connection establishment typically operates in the digital world, where each node is assigned a digital ID (e.g., the IP address). A node determines which node to connect with totally based on the digital-world information (e.g., network connectivity and the routing algorithm), and establishes a connection by sending a request message with the digital ID of the target receiver. For robots operating in the physical world, however, their communications are often triggered by sensor data, such as images from a camera. A robot may decide to connect with others based on real-time visual data or specific events detected in its surroundings. Fig. 1 shows three typical cases:

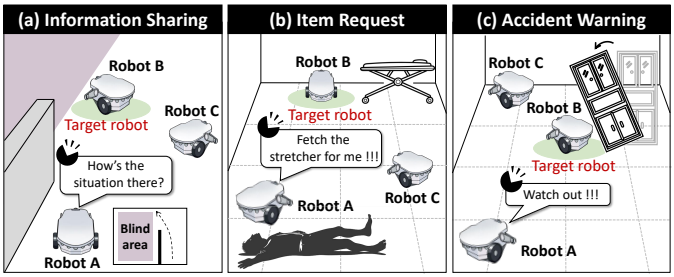


Fig. 1. Example application scenarios of SPing.

- *Information Sharing*. For better path planning, a robot (e.g., Robot A in Fig. 1 (a)) may request a partner (e.g., Robot B) who is close to the blind area for visual information.
- *Item Request*. A robot (e.g., Robot A in Fig. 1 (b)) may request the partner (e.g., Robot B), who is close to the item it requires, to help fetch it.
- *Accident Warning*. A robot (e.g., Robot A in Fig. 1 (c)) sends an instant warning to another robot (e.g., Robot B) who is not aware of the danger.

In these scenarios, traditional digital-world connection establishment technology fails as it's challenging for a robot to infer the digital address of a target robot in its vision field.

We ask a question that can we design an addressing mechanism that *operates in the physical world*? For example, can we assign each robot a physical-world address, which is an encoding of the robot's location attributes in the physical space? This mechanism analogizes the communication between two humans who do not know each other's names and thus call each other using descriptions of the physical environment around them (e.g., "the one sitting at the table"). This **physical-world addressing mechanism** enables a **"see-and-point"** communication mode: a robot can establish a connection pointing to any other robot without knowing its IP address, as long as it sees that robot in its visual field.

However, one problem here is what location attributes of a robot can be used to encode its physical-world address. One

Huangwei Wu, Meng Jin, Zhuxuan He, Qi Cao, Xinbing Wang are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {wuhuangwei, jinm, amy_he, c19975151009, xwang8}@sjtu.edu.cn.

Weiguo Wang is with the Technology Planning Department at NIO, China (E-mail: weiguo.wang2@nio.com)

Chenghu Zhou is with the Institute of Geographical Science and Natural Resources Research, Chinese Academy of Sciences, China (E-mail: zhouch@lreis.ac.cn)

(Corresponding author: Meng Jin and Weiguo Wang.)

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

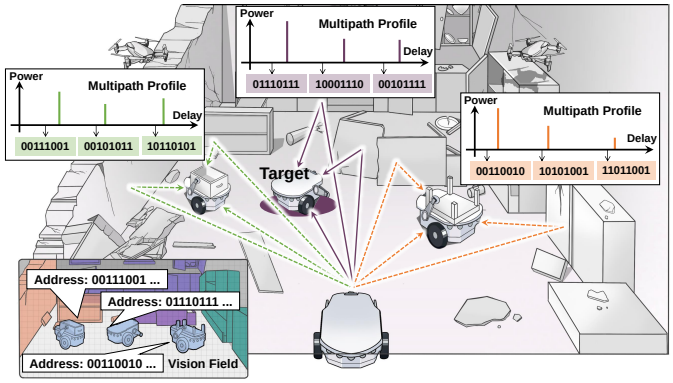


Fig. 2. Physical-world addressing mechanism.

direct solution is using the robot's 3D coordinates. However, this requires external localization infrastructures (e.g., GPS or indoor positioning systems) and well-maintained networks to make sure that each robot can keep track of both its own location and that of all the other robots [4]–[16], which requires reliable communication across robots, and thus is not always available in practice.

Another seeming alternative is to use the relative position of the target robot by measuring wireless signal timing (ToF) or directionality (AoA) [17]–[20], or using visual localization [21], [22]. However, these methods face a fundamental limitation: relative position measurements with respect to the transmitter are not valuable to the receiver, since they must first be converted to absolute locations for the receiver to determine whether it is the intended target robot. Unfortunately, this conversion again assumes an existing network and localization infrastructure in place.

Apparently, as we have shown above, all candidate approaches face a problem of circular dependency: **they require some form of prior communication link to exchange certain location or spatial information before establishing the actual communication link.** This circular dependency fundamentally makes them unsuitable for true "see-and-point" communication in unknown environments. What we need is an approach that can work without any pre-existing communication link.

We in this paper propose to leverage the *channel environment* of a robot as its physical-world address. The idea is proposed based on the fact that the channel of a receiver is location-specific [23], and more importantly, **can be "seen" by the transmitter and measured by the receiver**. Specifically, signals emitted from the transmitter will traverse multiple paths, reflecting off different reflectors, before arriving at the receiver (see Fig. 2) [24]–[35]. Since signals arriving at different locations propagate through different paths, a description of the received signals' propagation delays and powers along the paths - a multipath profile – can be used as a unique location attribution to encode the receiver's address. Since the transmitter, as a robot, is typically equipped with the ability to "see" and model the environment with its vision system, then, if it can further model the signal's propagation paths in the environment, it can foresee the receiver's multipath profile before message transmission and use it as the destination address for connection establishment. For the receiver, it can measure its multipath profile by detecting the arrival times of the multipath signals from the transmitter. This allows the receiver to obtain its physical-world address and perform address matching.

The key challenge in realizing the above idea, however, is how to achieve scene-to-channel transformation in *real-time*, using only a robot's onboard camera and computation. Specifically, the signal's interaction with the environment is complex, which depends on the locations, sizes, shapes, and materials of all the objects in the environment. Precisely modeling the environment in real-time is a nearly impossible task. More importantly, even if we can build a realistic 3D map, accurate ray tracing on the map is a time-consuming task, as we will show in Sec. III-D. These make the existing scene-to-channel methods ill-suited for our scenarios, where the transceivers are in fast mobility and instant connection establishment is required.

In this work, we solve the above challenges and develop a physical-world addressing mechanism, named SPing. The core components of SPing are: (i) a real-time 3D mapping and ray tracing technique that enables fast multipath profiling based simply on one camera shooting of the environment; (ii) a fuzzy address matching method that identifies non-exact matches between real channel measurements and vision-based simulations. and iii) a vision-based target tracking and pointing mechanism that establishes a connection even under the fast mobility of the transceivers. The **key advantage** of SPing is that it does not depend on any prior communication link, does not need any prior knowledge of the working environment, and does not rely on any external infrastructure or advanced hardware (e.g., LiDAR). This enables robots to work in uncertain and infrastructure-less environments.

We build an end-to-end prototype of SPing. To achieve accurate channel measurement on the receiver side, the connection request message is sent through the acoustic channel, where multipath signals' arrival times can be measured with fine granularity due to the low propagation speed of the acoustic signal. We test the performance of SPing in both static and mobile scenarios. The results show that SPing can establish a connection precisely pointing to the target robot, with an average true matching rate of 99.58% and an average false matching rate of 0.21%. The connection establishment delay is below 412ms, ensuring a reliable connection to the target robot at speeds of up to 2m/s.

Our key contributions can be summarized as follows:

- We propose a novel physical-world addressing mechanism, which allows robots to communicate and collaborate in a convenient "see-and-point" mode, without relying on a well-maintained multi-robot network.
- We for the first time achieve real-time transformation from the visual scene to the channel sketch using only a robot's on-board camera and computation.
- We build a working prototype of SPing, demonstrating its reliability under both static and mobile scenarios.

II. RELATED WORKS

This section introduces other potential methods to achieve "see-and-point" communication. The comparison between all

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

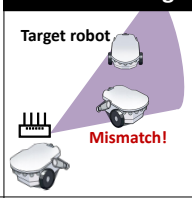
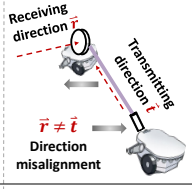
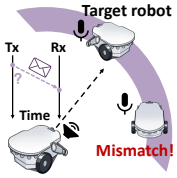
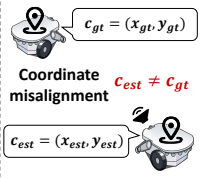
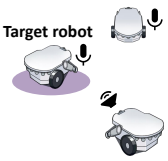
Method	Beamforming	Laser-based FSO	ToF-based	AoA-based	Coordinate-based	SPing
						
Spatial Resolution	Low (beam)	Moderate (line)	Moderate (ring)	Moderate (line)	**High (spot)**	**High (spot)**
Beam Alignment	Required	Required	**Not required**	**Not required**	**Not required**	**Not required**
Dedicated Device	Required	Required	**Not required**	Required	**Not required**	**Not required**
Pre-existing Link	**Not required**	**Not required**	Required	Required	Required	**Not required**
Coordinate System Alignment	**Not required**	**Not required**	**Not required**	Required	Required	**Not required**

Fig. 3. Comparison of different potential methods.

those potential methods and SPing is demonstrated in Fig. 3.

One intuitive idea is to restrict the signal propagation so that only the target robot can receive the signal. However, none of the existing methods could practically achieve "see and point". For example:

Beamforming. A common strategy is to use beamforming to steer the signal toward the target robot [36]–[45]. However, beamforming focuses the signal on a beam rather than a spot. All the receivers located in the target direction can receive the signal, which leads to a serious mismatch. Additionally, effective beamforming requires a sizable antenna array, which increases the cost and complexity, making it expensive and less feasible for widespread deployment on robots.

Laser-based FSO. Free-space optical (FSO) technology can focus the signal on the target robot leveraging the narrow laser beam [46]–[48]. However, FSO requires precise (mm-level) beam alignment between the transceivers [49]–[52], which is hard to retain when the transceivers are in mobility and the orientations are arbitrary. More importantly, FSO requires dedicated devices for optical transmitting, steering, and detecting, which are also costly and usually not available on robots.

Another promising idea is to use sensors to estimate various spatial information of the target robot, including relative distance (ToF), relative direction (AoA), or coordinates. The transmitter can then send this estimated information to the target for selective connection, based on whether the receiver's measurements match the transmitter's estimation. However, this method's reliance on prior communication links or coordinate system alignment makes it unsuitable for "see-and-point" communication in unknown environments:

ToF-based. Accurate ToF estimation requires tight time synchronization between transceivers [53]–[56], which in turn necessitates a pre-existing network to exchange temporal information among robots. This dependency limits its use in environments without established communication. Moreover, the connection could be inaccurately established with any robot whose distance to the transmitter matches that of the target, leading to a coarse ring-like spatial resolution.

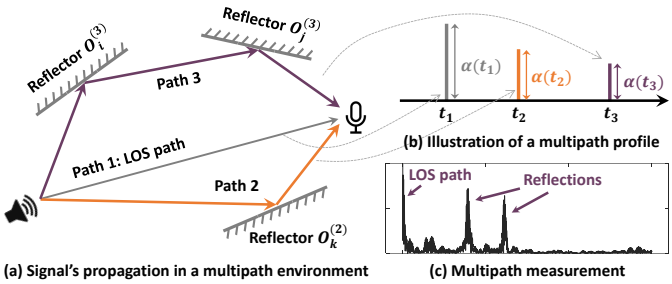


Fig. 4. Channel behavior of acoustic signal.

AoA-based. AoA estimation typically occurs at the receiver, but effective use of this method depends on inter-robot coordinate system alignment to align orientations between the transmitting and receiving robots, which typically requires a prior communication link to exchange the necessary pose information. This alignment is essential for matching the transmitter's estimation with the receiver's measurements. Additionally, AoA estimation requires a large antenna array [57], [58], which is impractical for mobile robots due to size and cost constraints and is especially challenging in environments rich in multipath interference [59]–[61]. Similar to ToF, this method risks connecting to any robot that lies in the same direction as the target robot.

Coordinate-based. Since most robots are equipped with cameras, it's intuitive to use visual-based self-localization (i.e., SLAM), paired with ad-hoc networking, to achieve "see-and-point" communication by using robots' coordinates as the physical-world addresses. However, although real-time visual SLAM technologies [62], [63] are mature for single-robot applications, they are insufficient to support a decentralized multi-robot system. This is because, in a multi-robot system, individual robots are confined to their own local coordinate systems, where collaborative SLAM [64] is required to align the local coordinate systems of individual robots.

Unfortunately, collaborative SLAM is limited by its high delay and high error accumulation. First, coordinate alignment fundamentally requires inter-robot loop closure detection [65], where robots need to identify sufficient shared visual features across their local maps. As a result, robots that enter the

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

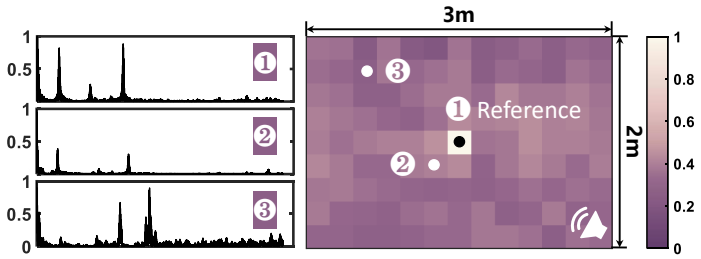


Fig. 5. Multipath channel variation across locations.

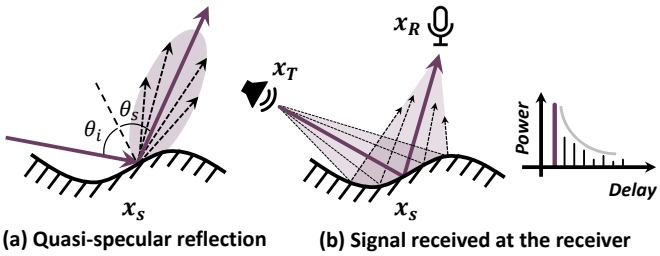


Fig. 6. Quasi-specular reflection model.

unknown environment from different entries must first explore large portions of the environment until they encounter shared scenes, leading to a high cold-start delay. Second, aligning coordinate systems involves solving a large-scale global consistency problem where measurements from different robots are often erroneous. SoTA methods rely on computationally intensive iterative optimization procedures [66], [67], which are challenging on resource-constrained embedded platforms (e.g., Raspberry Pi). Furthermore, collaborative SLAM relies on historical data that grows over time [68], leading to increasing alignment latency. Third, coordinate alignment is not a one-time operation. Even after a successful alignment, the spatial consistency between robots degrades due to odometry drift [69]. Moreover, modern SLAM systems actively perform local map optimization to correct accumulated errors [62]. This process changes the coordinates of historical trajectory points to ensure map consistency. Consequently, previously aligned coordinates become obsolete as soon as either robot optimizes its local map, forcing the system to repeatedly recalculate the alignment to track these changes.

Compared to all the potential methods discussed above, SPing is a purely software-based method, leveraging only the speakers, microphones, and cameras, which are widely available on robotic systems. It establishes a connection pointing to a spot, rather than a beam, achieving much higher spatial resolution. Additionally, it does not require directional alignment between Rx and Tx due to the broadcast property of acoustic signals. More importantly, SPing utilizes a multipath channel dependent solely on the environment and the transceivers' relative positions to establish the connection, with no need for prior communication links and coordinate system alignment between robots.

III. MODELING THE CHANNEL ENVIRONMENT

A. Signal's channel behavior

In a multipath environment, signals sent from a transmitter are usually reflected by multiple surrounding objects, which causes them to traverse different paths before reconvening at the receiver (see Fig. 4(a)). Such a multipath channel can be characterized by a *multipath profile*, which gives the power of signals along different paths, as a function of propagation delay (see Fig. 4(b)). Formally, we define the multipath profile as a time series $\psi(t)$, where each element is the power of the signal received at time t. Suppose there are P paths and the delay and amplitude of each multipath signal are t_p and $\alpha(t_p)$, respectively, we have:

$$\psi(t) = \sum_{p=1}^{P} \alpha(t_p) \cdot \delta(t - t_p) \tag{1}$$

where $\delta(\cdot)$ is the delta function. Fig. 4(c) shows an example of the multipath profile obtained in an indoor environment, where the peaks are caused by the reflectors around.

The multipath profile can be measured on the receiver side by directly detecting the arrival time of the multipath signals. However, fine-grained measurement of the multipath profile would require hardware with a high signal sampling frequency. Taking the widely used 20MHz WiFi module as an example, the multipath profile resolution achieved is $\Delta\tau = \frac{1}{20MHz} = 50ns$, which leads to a $\Delta d = \Delta\tau \cdot c = 15m$ resolution in measuring the multipath lengths (where $c = 3\times10^8 m/s$ is the light speed). This means that the receiver cannot distinguish between two signal paths if their length difference is lower than 15m. To solve this problem, we leverage the acoustic channel, which enables more fine-grained multipath profiling due to the low propagation speed of the acoustic signal. Specifically, a microphone can sample at $f = 48kHz$ while the sound speed is only $340m/s$, which leads to a 7.1mm resolution in the multipath profile measurement. We in Sec. V-C introduce how the multipath profile is measured in detail.

B. Multipath profile as address

Since signals arriving at different receivers propagate through different paths, the multipath profile can be used as a unique address for each receiver. We in Sec. V-D introduce how to transform a receiver's multipath profile to its physical-world address. In this section, we show how the multipath profile varies across locations through an experiment.

In the experiment, we place a speaker at the corner of a 2m×3m room and place a microphone at 117 different locations in the room. We measure the multipath channel at each location, and Fig. 5 (left) illustrates channels measured at three different positions. As can be seen, the three channels are quite different in both time distribution and power. We further calculate the Pearson correlation coefficient ρ [70] between the channel measured at the reference location c_r and that measured at other locations c_l by $\rho = Cov(c_r, c_l)/[\sigma(c_r)\sigma(c_l)]$, where $Cov(\cdot)$ is the covariance and $\sigma(\cdot)$ is the standard deviation. Fig. 5 (right) visualizes the correlation of each location. As expected, the correlation drops quickly as the distance increases. When the distance >0.5m, the correlation decreases to less than 0.5. So, by using the multipath profile as a receiver's physical-world address, **we can distinguish two receivers with a larger than 0.5m spacing.**

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

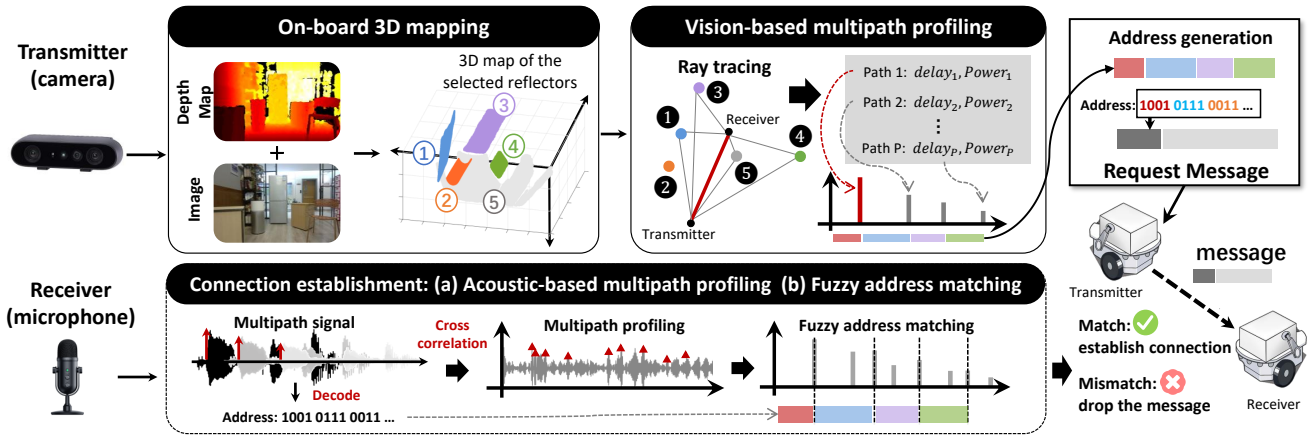


Fig. 7. System overview.

C. Scene-to-channel transformation

To obtain the receiver's physical-world address, the transmitter needs to predict the receiver's multipath profile based on the visual scene obtained by its camera. This is in theory feasible since *the multipath channel is the direct result of the environment*. Recall that the multipath profile is characterized by two attributes of the signal paths: the propagation delay t_p and the received power $\alpha(t_p)$. According to the acoustic propagation equations [71], these two attributes can be fully determined by the physical characteristics of the surrounding reflectors.

Fig. 6 (a) illustrates how the acoustic signal interacts with an object. Here, the surface of an object is modeled as a quasi-specular reflector, where the incident wave on a point is scattered to different directions with different powers. The scattered energy in each direction is decided by a scattering coefficient $\sigma(\theta_i, \theta_s)$, which is the ratio of outgoing energy to incident energy and is related to the incident and scattering angles (θ_i and θ_s) of the signal [72]. Here, θ_i and θ_s represent the angles that the incident and scattered rays make with the surface normal at the reflection point. The strongest reflection is achieved in the specular direction where $\theta_i = \theta_s$.

So, if we divide the signal emitted from the transmitter into an infinite number of rays, the rays will reflect on the object with different angles and arrive at the receiver with different delays, forming a series of peaks, as shown in Fig. 6 (b). Due to the limited time resolution of the microphone, we can approximately consider those rays as propagating along the same path p and consider the delay of the path as that of the strongest ray, which is represented as:

$$t_p = \frac{||\mathbf{x_T} - \mathbf{x_s}|| + ||\mathbf{x_s} - \mathbf{x_R}||}{v} \quad (2)$$

where $\mathbf{x_s}$ denotes the location of the incident point of the strongest ray. $\mathbf{x_T}$ and $\mathbf{x_R}$ denote the locations of the transmitter and the receiver, respectively.

The reflected energy is statistically accumulated across the surface area of illumination, and the overall attenuation from the transmitter to the receiver along the path p is:

$$\alpha(t_p) = l(\mathbf{x_T}, \mathbf{x_s}) l(\mathbf{x_s}, \mathbf{x_R}) R(m) \iint \sigma(\theta_i, \theta_s) dA \quad (3)$$

where $l(\mathbf{x}, \mathbf{y})$ is the path loss from point $\mathbf{x}$ to point $\mathbf{y}$, which is solely determined by the propagation distance [73]; $R(m)$ is the absorption coefficient of the reflector, which is related to the surface material; dA is the area of illumination; and $\sigma(\cdot)$ is the scattering coefficient at a reflection point.

Summary. Eqs. (2) and (3) tell that the multipath channel of the receiver can be fully determined by the locations, sizes, orientations, and materials of the reflectors in the environment. So if the transmitter could obtain a 3D scene model of the physical environment, it can simulate acoustic rays to trace paths that an actual acoustic signal would take in the real world, which allows it to generate the multipath profile of any receiver in the scene.

D. Challenges

Although the signal's interaction with the environment can be precisely modeled by Eqs. (2) and (3), ray-tracing the environment in real-time with onboard camera and computation is still a challenging task, due to the following limitations.

(1) Limited sensing capability. Signal's interaction with the environment depends on the physical characteristics of *all the objects* in the scene. Thus, accurate multipath profiling requires a full omnidirectional scan of the environment, which is however difficult to accomplish using the onboard camera due to its limited FoV (field of view). Some existing works (e.g., SpaceBeam [74]) address this problem by mapping the environment in advance, through a comprehensive visual-SLAM-based site survey. This on one hand requires prior access to the environment, on the other hand, assumes a quasi-static environment so that the 3D map built in advance can be directly used for ray tracing. In our scenario, however, the robots are expected to operate in unknown and highly dynamic environments for urgent and unplanned tasks, where advanced environment mapping is neither available nor effective. Furthermore, current visual-SLAM algorithms prioritize accurate reconstruction of all objects in view, resulting in redundant visual information that impedes the effective multipath profile generation.

(2) Limited computation resource. Even if we can perform a full scan of the environment, performing real-time ray tracing using only the onboard computation is still an impossible task. Specifically, to perform ray tracing, one needs to first generate

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

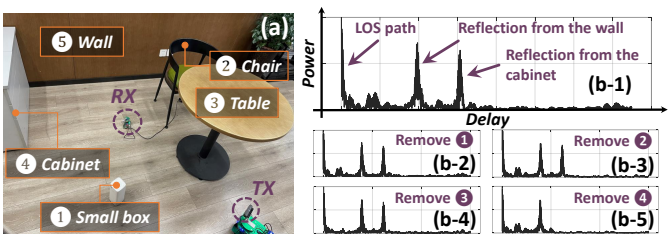


Fig. 8. Reflection from the dominant reflectors.

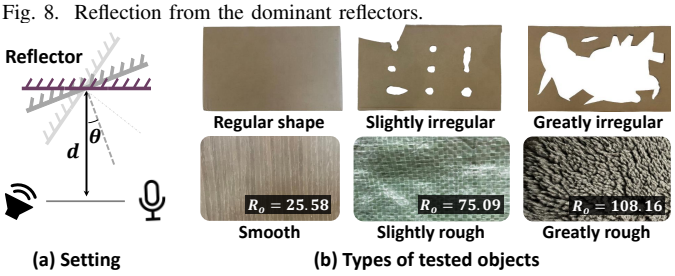


Fig. 9. Tested objects.

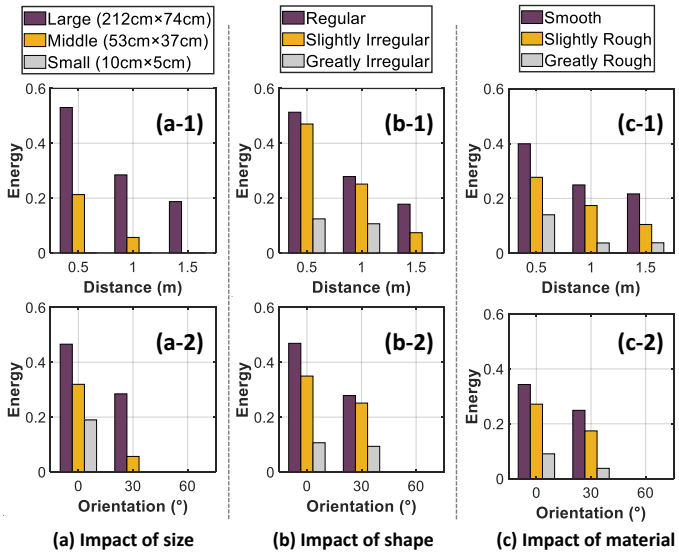


Fig. 10. Factors that affect reflection energy.

a 3D mesh of the environment which consists of millions of (say N) geometric primitives, and then generate a set of (say M) candidate rays to interact with those primitives. Suppose that we consider d-order reflection among objects, the time complexity of ray tracing is approximately $O(d \cdot M \cdot N)$ [75]. Although some recent efforts [76], [77] can reduce the complexity to $O(d \cdot M \cdot logN + N \cdot logN)$, the time overhead is still unacceptable (9.6s, based on the experiment in Sec. VI-C) for real-time operation on embedded devices.

We address the above challenge based on an observation that the multipath channel is dominated by a small number of reflectors, which are large, regular-shaped, and located close to the LOS path between transceivers. Thus, SPing proposes an onboard 3D mapping module to identify and reconstruct the dominant reflectors and a vision-based multipath profiling module to accelerate ray tracing with simplified surface modeling and an intersection detection process. More details will be introduced in Sec. V.

IV. BASIC IDEA OF SPING

SPing is a novel system that enables "see-and-point" communication between robots. On the sender side, SPing can reconstruct and ray-trace the environment in real-time, using only the onboard camera and computation. It simulates the multipath profile of the target robot, quantizes it as a destination address, and embeds it in the request message. On the receiver side, SPing first measures the multipath channel by detecting the arrival time of the message's multipath signal. It then matches the measured channel with the destination address for connection establishment. As illustrated in Fig. 7, SPing consists of the following modules.

(1) Onboard 3D mapping

SPing's 3D mapping module can generate a sufficiently accurate 3D model of the environment in real-time, using only the onboard camera. It achieves this based on the observation that, although the environment can be complex and cluttered with reflectors, only a small number of reflectors dominate the multipath channel. The dominant reflectors are usually those large and regular-shaped objects located near the LOS path, which can be captured by the onboard camera of the transmitter. We in Sec. VI-G show that **considering only two dominant reflectors is sufficient to generate distinct multipath profiles for different receivers.** Moreover, our experimental results in Fig. 10 show that **a reflector with a reflection area of 1.5m^2 is sufficient to reliably generate multipath. Such reflectors are common in indoor environments (e.g., floor, wall, cabinets, sofas, etc.).** Based on this observation, SPing's 3D mapping module first identifies the dominant reflectors on the captured image, based on the reflectors' sizes, shapes, locations, etc. It then generates 3D models for only those selected reflectors, which reduces the complexity in the subsequent ray-tracing process, meanwhile maintaining the accuracy. Note that SPing's 3D mapping module does not incur additional computational overhead since most robotic systems have already integrated visual-SLAM functions for obstacle avoidance and autonomous navigation [78]. The intermediate results of visual-SLAM (e.g., the depth map and the detected objects) could be reused by SPing.

(2) Vision-based multipath profiling

To achieve real-time ray tracing for vision-based multipath profiling, SPing leverages the fact that the surfaces of the dominant reflectors are usually regular-shaped, which can be characterized by the quadric surface equation with a small number of coefficients. Thus, instead of generating a dense mesh of the objects and traversing all the primitives for path detection, SPing directly models the entire surface and checks whether a ray intersects with the surface. Considering that the number of surfaces is far fewer than the number of primitives, computation overhead can be greatly reduced.

(3) Connection establishment

To establish a connection to the receiver, SPing first generates a destination address based on the predicted multipath profile of the receiver. Then it broadcasts a request message which contains the address. After receiving the message, the receiver measures the multipath profile by detecting the arrival time of the multipath signals. Then, a fuzzy address matching

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

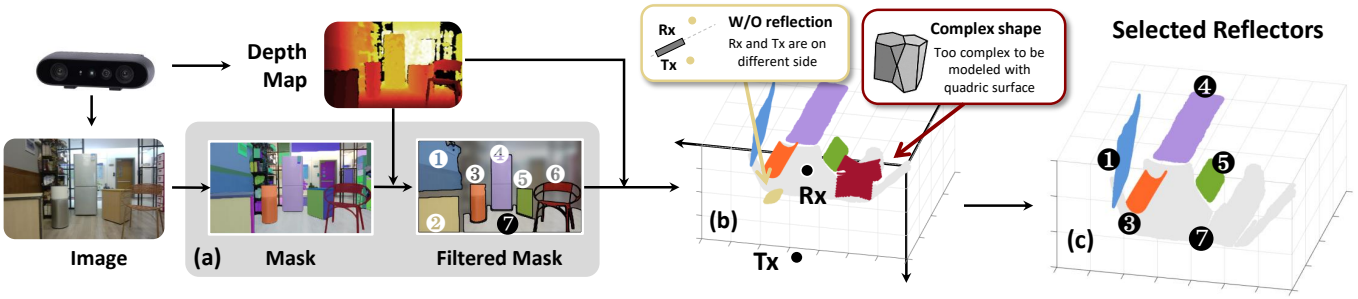


Fig. 11. Framework of the 3D mapping module.

method is proposed to match the address generated from the vision-based simulated multipath profile to the real channel measurement. If matched, the receiver receives the subsequent messages, otherwise it discards the messages.

V. SPING'S DESIGN

We in this section lay out the design details of SPing. We first show how SPing generates a 3D map of the environment in Sec. V-A. Then in Secs. V-B and V-C, we show how the sender and the receiver generate the multipath profile based on the visual and acoustic signals, respectively. Finally, we in Sec. V-D show how the connection is built by matching the vision-based and acoustic-based multipath profiles.

A. Onboard 3D mapping

SPing's 3D mapping module is designed based on the assumption that: only a small number of objects, which reflect more energy to the environment, dominate the multipath channel. So, reconstructing only those dominant reflectors can already generate a sufficiently accurate multipath profile for each robot. To verify this assumption, we observe the multipath channel of an indoor environment with five different reflectors, as shown in Fig. 8 (a). Fig. 8 (b-1) shows the multipath channel of the environment when all five objects exist. We then remove the object ❶-❹ from the environment separately and measure the corresponding channels. As can be seen in Figs. 8 (b-2)-(b-4), objects ❶-❸ have little impact on the channel. Fig. 8 (b-5) indicates that the object ❹ (cabinet) contributes to the second NLOS peak in the channel. Although the object ❺ (wall) is hard to remove, we could infer that it contributes to the first NLOS peak based on the above results.

We further explore the factors that affect the amount of energy an object reflects, and show that we could identify the dominant reflectors based on their size, shape, material, and location. The experimental settings are demonstrated in Fig. 9 (b). For each object, we test the amount of energy it reflects when putting it at different orientations θ and distances d to the transceiver (see Fig. 9 (a)). The results are shown in Fig. 10. We make three important observations: i) objects with large, smooth, and regular-shaped surfaces reflect more energy to the environment; ii) an object reflects more energy when it is located closer to either of the transceivers; and iii) a smaller incident angle also brings larger reflection energy.

Inspired by the above observation, we propose an object detection method, which identifies the dominant objects based on the objects' physical characteristics mentioned above. However, it is time-consuming to estimate all the characteristics of all the objects in the vision field. Hence, SPing operates in a recursive way: it first estimates the locations, sizes, and roughness (characteristics that can be obtained before the 3D reconstruction) of all the objects, based on which it screens out most of the objects. It then reconstructs the surface of the remaining objects and filters out surfaces with irregular shapes (based on reconstruction residues) and those with weak reflection characteristics (based on surface orientations). Dominant objects are finally selected by ranking the roughly estimated signal propagation distances.

Fig. 11 shows the basic idea of the object detection method. It first extracts all the objects using a fast image segmentation algorithm YOLOv8n-seg [79], which generates masks of arbitrary objects from a single RGB image. It then estimates the location of each object based on the sparse points sampled on the depth map. Then, according to the spatial perspective principle [80], SPing makes a rough estimation of an object's actual size S_A, based on its distances to the transmitter d and the size of its mask S_M as $S_A = d^2 \cdot \frac{S_M}{L_f^2}$, where L_f is the focal length of the camera. After obtaining the objects' locations and sizes, SPing screens out the objects that are too small or too far away from the LOS path.

The filtered mask set is then processed by a roughness estimation algorithm [81], which outputs the roughness level (R_o) of an object by calculating the root mean square of pixel gradient magnitudes from its corresponding region in the RGB image. Objects with $R_o > 100$ (e.g., the carpet shown in Fig. 9) are filtered out. The remaining masks, along with the depth map, are then used to generate the 3D points, which are then processed by a surface fitting algorithm to detect the piecewise surfaces of each object. Specifically, we model the surfaces by the quadric surface equation:

$$S(p) = A \cdot x^2 + B \cdot y^2 + C \cdot z^2 + D \cdot x + E \cdot y + F \cdot z + G = 0 \quad (4)$$

where $p = \{A, B, C, D, E, F, G\}$ are the surface coefficients. The quadratic terms A, B, and C characterize the curvature of the surface, the linear terms D, E, and F determine the position of the surface center, and the constant term G, along with all other coefficients, jointly determine the final scale and position of the surface. Eq. (4) characterizes most everyday surfaces such as planes, cylinders, cones, etc. To construct the 3D surface of an object, SPing optimizes the coefficients p to fit the 3D points $\mathbf{P}$ based on the least squares method (LSM). Mathematically, it computes:

$$P^* = \arg\min_p Res(S(p), \mathbf{P}) \quad (5)$$

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

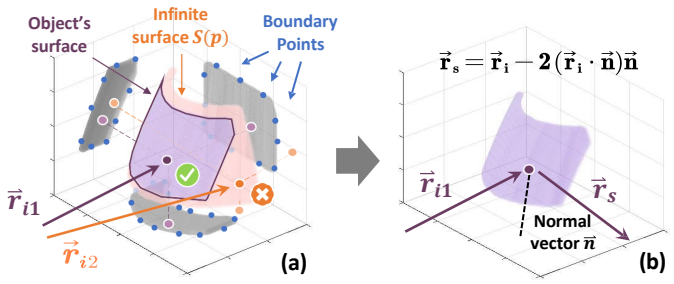


Fig. 12. Vision-based Multipath Profiling: (a) intersection detection; (b) ray generation.

where $Res(\bullet)$ captures the sum of squared Euclidean distances from each point in $\mathbf{P}$ to its closest point on the surface $S(p)$, which quantifies the goodness-of-fitting of the constructed surface. Clearly, $Res(\bullet)$ will be minimized when we get the optimal coefficients P^*, because the geometric shape characterized by $S(P^*) = 0$ lies closest to all the 3D points in $\mathbf{P}$ simultaneously.

After constructing the surfaces, we identify and filter out those irregular-shaped surfaces. Specifically, surfaces with high fitting residual Res are considered irregular because a high fitting residual means that the surface is too complex to be modeled with the quadric surface equation. Then, for each remaining surface, SPing checks whether there exist possible reflection paths between the surface and the transceivers. This is done by checking whether the transmitter and the receiver are on the same side of the reflector. If not, the surface is considered invalid and filtered out. Finally, SPing ranks the remaining surfaces based on their distance to the LOS path, and selects top-K of them for multipath profiling.

B. Vision-based multipath profiling

Now we have generated the 3D map of the dominant reflectors, we then focus on how to generate the multipath profile of the target receiver from the 3D map. This is typically achieved by ray tracing. In general, ray tracing works by initializing a set of rays from the transmitter in different directions. For each ray $\vec{r}$, it checks whether the ray intersects with any object in the environment. If the ray intersects directly with the receiver, the tracing for this ray terminates. Otherwise, it creates a new ray $\vec{r_s}$ in the direction of the specular reflection. This is governed by the law of reflection, which tells that the angle of incidence equals the angle of reflection. Specifically,

$$\vec{r_s} = \vec{r_i} - 2(\vec{r_i} \cdot \vec{n})\vec{n} \tag{6}$$

where $\vec{r_i}$ is the unit vector indicating the direction of the incident signal, and $\vec{n}$ is the norm vector of the surface at a specific point. For the point (x_0, y_0, z_0), the norm vector could be calculated as the gradient of the surface equation by $\vec{n}(x_0, y_0, z_0) = \nabla S(p)\Big|(x_0, y_0, z_0)$. For each new ray, it proceeds to recursively find a path to the receiver, unless a predetermined limit on the number of reflections is reached.

However, accomplishing the above process in real-time is impossible due to the time-consuming intersection detection process. Specifically, to determine whether a ray intersects with an object, the traditional ray-tracing algorithm has to first generate a dense 3D mesh of the object, which usually consists of millions of geometric primitives (polygons, usually triangles). Then, for each ray, it traverses *all the polygons* to check whether the ray interacts with a certain polygon, which means solving the Ray-Plane intersection function millions of times. Although some recent efforts try to speed up this process by modeling the 3D environment with a BVH (bounding volume hierarchy) tree structure, which reduces the number of intersection detections, the latency for tree construction (a few seconds) is still unacceptable (see Sec. VI-C).

To solve the above problem, we leverage the fact that the surfaces of the selected objects are usually regular-shaped, which can be characterized by the quadric surface equation. Thus, instead of modeling the surface with millions of geometric primitives and traversing through all the polygons, we consider the surface as a whole and directly check whether a ray intersects with the surface. Specifically, for each ray $\vec{r_i} = o + t \cdot \vec{d}$ that starts at point o and goes in the direction $\vec{d}$, SPing first determines the point at which the ray intersects the surface (as shown in Fig. 12 (a)). This is achieved by substituting the ray equation into the surface equation (Eq. (4)) as $S(p) = 0$. After solving the equation to get the value t^*, we can compute the coordinates of the intersection point by substituting t^* back into the ray's equation $\vec{r_i}$. However, since Eq. (4) defines an infinite surface, so for each intersection point, we need to further check whether it is located within the boundary of the object's surface. This can be achieved with a fast ray-segment interaction algorithm. Specifically, before the ray tracing process, we project the surface's 3D points to the three coordinate planes and detect the boundary of the projected plane points by performing convex hull detection [82] (as shown in Fig. 12 (a)). Assuming that the surface is convex, then for each candidate point, if its projections on all three planes are located within the boundary, we consider it as an intersection point and create a new ray in the specular output direction.

After finding all the primary paths, SPing estimates the delay t_p and power $\alpha(t_p)$ of each path based on Eqs. (2) and (3), respectively. Then the multipath profile can be generated based on Eq. (1). Note that since the surface material is difficult to obtain from the visual signal when estimating the power of each path, SPing assumes all the reflectors are smooth by setting their absorption coefficients as $R(m) = 1$. This brings errors in signal power estimation. We show in Sec. V-D that using only the delay of each path is sufficient to generate a distinct physical-world address for the receiver.

C. Acoustic-based multipath profiling

In this section, we introduce how the target robot (the receiver) measures the multipath channel based on the acoustic signal emitted from the transmitter. This is done by detecting the arrival time of the multipath signals. Specifically, any wireless packet starts with a known preamble signal [83], [84]. Once detecting a packet, the receiver calculates the cross-correlation between the received signal and the preamble template. The correlation result produces a peak when a delayed version of the preamble signal is detected [85]. We choose the chirp signal as the preamble since the auto-correlation of the chirp signal has a narrow peak so that we can separate paths with a small time difference.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

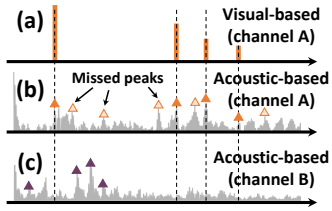


Fig. 13. Vision-based and acoustic-based profiles.

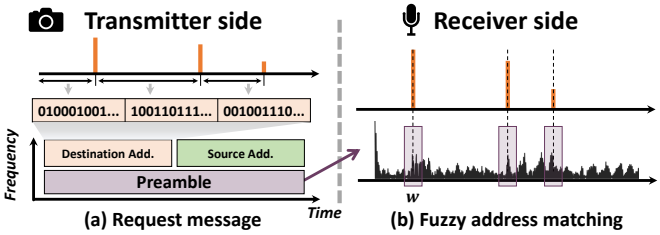


Fig. 14. Process of address generation and matching.

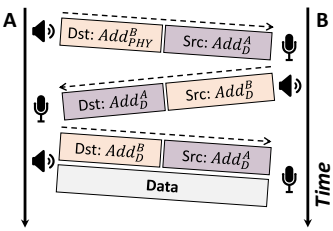


Fig. 15. Acoustic connection establishment process.

D. Address matching

So far, we have introduced how SPing extracts the multipath profile from the visual and the acoustic signals, which we term as *vision-based* and *acoustic-based* profiles, respectively. Now we introduce how we match these two profiles for the subsequent connection establishment process.

We meet a challenge here: due to the imperfect 3D mapping of the environment, there exist differences between the acoustic-based and the vision-based multipath profiles. Figs. 13 (a) and (b) show an example of such a mismatch. We can observe that: i) due to the lack of material information, SPing cannot accurately estimate the power of each path; and ii) since SPing cannot build a full 3D map of the environment, only part of the paths are reconstructed in the vision-based profile. To solve this challenge, SPing generates the destination address leveraging only the delay of each path, which can be measured with high accuracy and granularity. Moreover, we propose a fuzzy address matching method to identify non-exact matches between real multipath measurements and vision-based simulations.

(1) Address generation

We use the delay of the paths for address generation. The reasons are two-fold. First, the delay is determined solely based on the signal's propagation distance, without relying on unpredictable factors such as the material of the surfaces. Second, due to the low propagation speed of the acoustic signal, the signal's propagation distance can be measured with high resolution (as mentioned in Sec. III-A). This makes it possible to generate addresses with high location-distinguishability. Figs. 13 (b) and (c) compare the multipath profiles of two devices which are 1m apart. We can observe an obvious time offset between the peaks in the two profiles.

To convert the multipath profile to bits, the transmitter selects L peaks with the highest energy in the multipath profile and calculates the interval between each two successive peaks. Each interval is translated to a B-bit binary sequence as an address symbol (as shown in Fig. 14 (a)). Then we concatenate the address symbols to generate a $B \cdot L$-bit address. We empirically set the symbol length $B = 12$ bits and set the address length $L = 3$ symbols based on the experimental result in Sec. VI-G.

(2) Fuzzy address matching

To establish a connection to the target robot, the transmitter broadcasts a request message that contains the physical-world address of that robot. Fig. 14 (a) shows the format of the request message, which consists of a preamble, a physical-world destination address, and a temporary source address. The preamble is an 80-ms chirp signal on the 17-23kHz band. The address is encoded as a 60-ms 36-bit sequence on the 12-16.6kHz band, using the coding scheme in [84].

To perform address matching, the receiver first obtains the acoustic-based multipath profile using the preamble signal (See. V-C) and then recovers the vision-based multipath profile from the destination address. It then compares the acoustic-based and the vision-based multipath profiles using a fuzzy matching algorithm, as shown in Fig. 14 (b). Specifically, it just checks whether the peaks in the vision-based profile occur in the acoustic-based one, leveraging a J-shape detection algorithm [85]. For each peak, it checks with a window of length W to tolerate possible offset due to 3D mapping errors or hardware imperfections in peak detection. W is empirically set as 1ms as shown in Sec. VI-G. Then, if the error is less than 1ms, the peaks on the two profiles can be matched successfully. Otherwise, the matching procedure fails, and thus the receiver will drop the request message. The transmitter then has to retransmit the request message to build the connection.

E. Connection establishment

We in this section introduce the connection establishment process illustrated in Fig. 15. In this process, the transmitter A first generates the physical-world address of the receiver B (denoted as Add^B_{PHY}) based on the vision-based profile. It then sends a request message embedded with both Add^B_{PHY} (as the destination address) and its own digital ID Add^A_D (as the source address). After receiving this message, B responds with an ACK that includes both its own ID (as the source address) and that of A (as the destination address). After this round-trip handshake, the two robots could start the actual data transfer using their IDs as addresses.

The above process considers only the case with one pair of robots. To handle the case where many robots are in range of each other, we design a coordination mechanism which involves the following two modules:

(1) Collision-free transmission. This module avoids the collision between multiple transmitters by performing carrier sense. Specifically, each transmitter first checks if the wireless medium is idle before transmission and will take a backoff if the channel is busy. This largely avoids contentions among robots. In the rare cases where the contention still occurs, since the transmitter will not receive the ACK signal from the target receiver, it will repeat the above process to retransmit the message.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

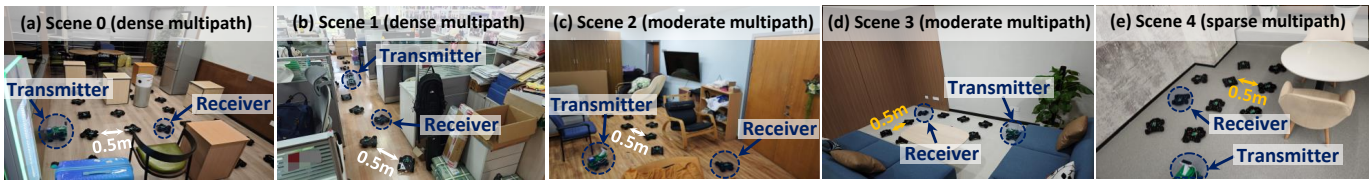


Fig. 16. Experimental environments.

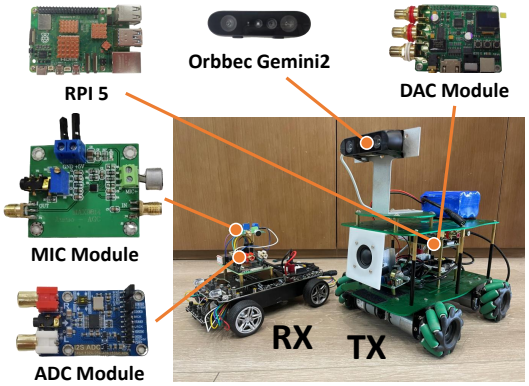


Fig. 17. Hardware implementation.

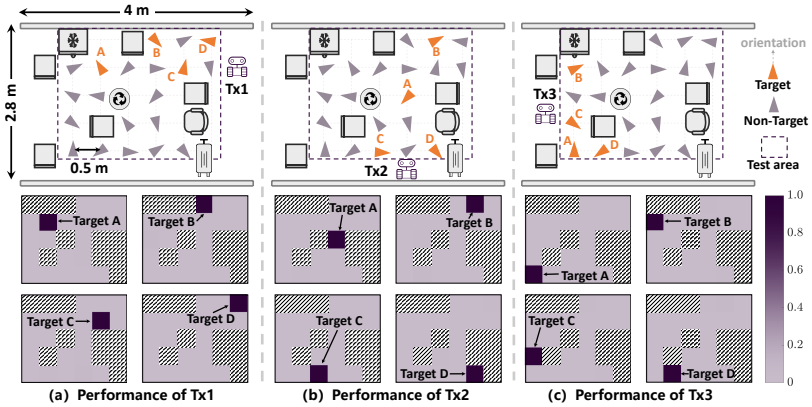


Fig. 18. Overall performance.

(2) Selective acknowledgment. This module considers the case where multiple robots, which are close to each other, are activated by the same physical-world address. In this case, they will ACK the transmitter with different IDs (note that the responses will not collide due to the carrier sense mechanism). Since SPing aims at connecting with an expected location instead of any specific robot, the transmitter can respond to some or all of them based on the task requirements. For example, if the transmitter just needs a collaborator located at a certain position, it will choose either of the robots to respond. While, if it needs to send a warning to robots that are located in a hazardous area, it will respond to all of them.

F. Target tracking

As the multipath channel decorrelates fast across locations, any movement of the target receiver can result in a mismatch between the vision-based predicted channel and the acoustic-based measurement result. To address this problem, we propose a target tracking module to predict the future location of the robot and generate the multipath profile in advance. Specifically, after the 3D mapping process, SPing first detects the mobility of the target robot by comparing the robot's Region of Interest (ROI) in image frames over time, using the lightweight object detection model YOLOv8n-det [86]. Once mobility is detected, a Kalman Filter (KF) based tracking mechanism is triggered. Firstly, the target's 3D locations at different time steps are calculated using the ROIs and the depth maps. Then, based on these locations from the previous N_{pre} time steps, the KF algorithm can predict the target's location in the next step.

However, the above target tracking process suffers high latency, which limits the maximum movement speed of the robot that can be tolerated. We find that the bottleneck in the latency comes from the YOLOv8n-det model. Specifically, YOLO-series models perform target identification and ROI localization simultaneously. It uses a complex CNN network [87] to extract both the appearance-related and the location-related features of the target. Considering that only the location-related features are important for target tracking, we compress the model by decreasing the number of feature filters in the convolution layers, which reduces the model size by 95% and the tracking latency by 83%, without sacrificing the ROI localization accuracy.

VI. EVALUATION

A. Experimental methodology

Hardware implementation. We implement SPing on two Raspberry Pi 5 for evaluation. As shown in Fig. 17, at the transmitter side, the RGB image and the depth map used for 3D mapping are obtained from an Orbbec Gemini2 depth camera that supports over 30FPS output [88]. The camera uses binocular structured light for depth perception [89]. Specifically, it actively projects a light with a known pattern onto an object and obtain the depth information of the object by analyzing the pattern of the reflected signal from two different camera views. The audio signal is first generated by a PCM5122 DAC and then amplified by a TDA7297 amplifier before being transmitted by a 15W speaker. At the receiver side, the audio signal is first collected by a microphone and then amplified by a MAX9814 amplifier. A WM8782S ADC is used to transform the audio into digital signal. The sampling rate is set at 48kHz.

Software implementation. We implement SPing with C++ and leverage multi-core programming for acceleration with the OpenMP library. We also accelerate the code in ARM-based Raspberry Pi with NEON instructions [90]. All the deep learning models are implemented by the NCNN framework [91]. The YOLOv8n-seg model is trained with 20% size of the SAM dataset [92].

Metrics. We evaluate the performance of SPing with two metrics: the True Matching Ratio (**TMR**) and the False Matching Ratio (**FMR**). The TMR refers to the ratio that the transmitter is successfully matched with a target receiver that

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

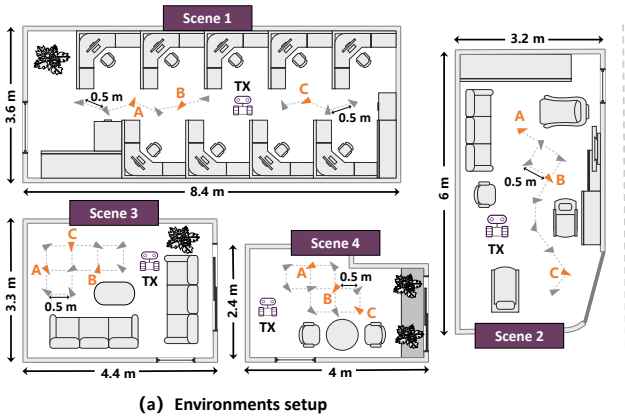


Fig. 19. Performance in different environments.

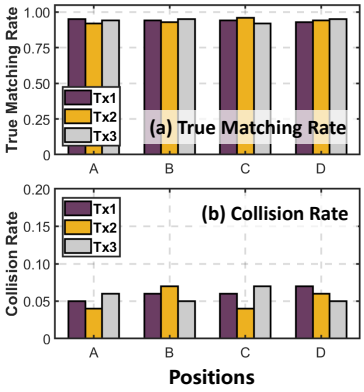


Fig. 20. Performance under simultaneous transmission.

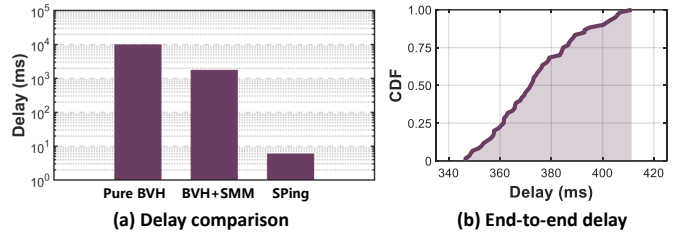


Fig. 21. Delay evaluation.

is located at the desired position, calculated by n_s/N. Here, n_s is the number of successful matches and N is the number of total attempts. FMR refers to the ratio that the transmitter is falsely matched with a non-target receiver that is located away from the desired place and should not respond to the request message, calculated by n_f/N. Here, n_f is the number of false matches. In experiments, N is set as 100 by default.

Experiment settings. We conducted experiments in five different indoor environments with dense (Scene 0&1), moderate (Scene 2&3), and sparse (Scene 4) multipath (see Fig. 16). The floor maps are shown in Figs. 18 and 19. We consider both static and mobile scenarios. In the mobile scenarios, we let the receiver move with different trajectories and speeds. The setting of the trajectory is shown in Fig. 25.

Baseline. We compare SPing with a "visual SLAM + coordinate-based addressing + ad-hoc networking" baseline. Specifically, each robot runs the visual SLAM method ORB-SLAM3 [62] to estimate its local trajectory in its own coordinate system, while a decentralized collaborative SLAM framework — Swarm-SLAM [93] — is used to align the local coordinate systems between robots, enabling coordinate-based addressing in a unified global map. The ad-hoc network is established via Wi-Fi on Raspberry Pi 5.

The above baseline system follows this workflow: When the transmitter sees the receiver, it estimates the receiver's relative position using the RGB-D camera, then transforms it to the receiver's coordinate system based on the transformation matrix obtained from the most recent alignment result. It then broadcasts the estimated coordinates as the target address. A matching attempt is considered successful if the deviation between the estimated coordinates and the receiver's self-localized coordinates is within a tolerance threshold (denoted as TH in Fig. 23 (b)).

Note that Swarm-SLAM is a SoTA framework designed explicitly for multi-robot systems operating over ad hoc networks. It greatly improves data exchange efficiency and reduces communication overhead during collaborative SLAM. We implemented the baseline system using the open-source ROS 2 package [94], [95] on two robots identical to the platforms used for the SPing evaluation (Tx robot in Fig. 17).

B. Overall performance

We first evaluate the overall performance of SPing. The experiment is conducted in a 2.8m×4m room (as shown in Fig. 16 (a)), where common objects in daily life, such as desks, suitcases, and a refrigerator, are scattered throughout the testing area. We deploy the transmitters at three different positions as the purple robot icons indicate, so that a receiver located at any of the positions marked as triangles could be visible to at least one of the transmitters. The distance between two adjacent receivers is 0.5m. For each transmitter, we select four receivers in its vision field as the target receiver, marked as A∼D in each figure. In the experiment, the transmitter sequentially generates the physical-world address for each target receiver and sends a connection request message containing the address, during which we observe the true matching rate (TMR) and false matching rate (FMR) for the target and non-target receiver, respectively.

Figs. 18 (a)-(c) show the result achieved when the transmitter is placed at positions Tx1-Tx3. In each figure, we show the matching rate achieved on each receiver, when the transmitter sends a message to different target receivers. A darker color represents a higher matching rate. The mosaic blocks indicate the obstacles' locations. We observe that in all the cases, SPing can always precisely establish a connection pointing to the target receiver, without disturbing other non-target receivers. Specifically, the matching rate of the target position keeps higher than 96%, while that of a non-target position keeps lower than 1%, even for the non-target positions which are only 0.5m away from the target position. The results show that SPing achieves an average TMR of 99.58% and an average FMR of 0.21%.

Performance in various environments. We further repeat the experiments in Fig 18 in another four environments with different clutter levels, as shown in Fig. 19 (a). In each environment, the transmitter is placed in a fixed position. Three

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

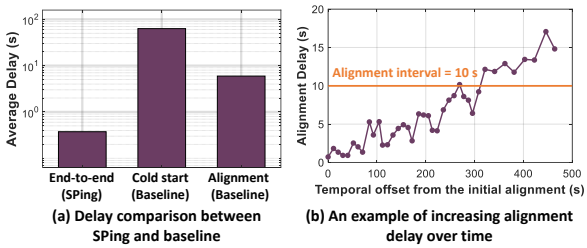


Fig. 22. Time overhead of the baseline system.

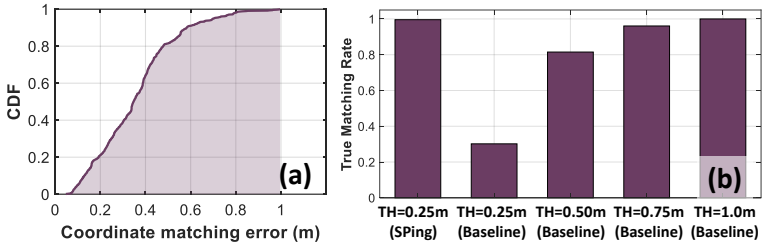


Fig. 23. Accuracy of coordinate-based addressing using baseline.

of the 10 receiving positions are selected as target positions. Figs. 19 (b-1) and (b-2) show the TMR and FMR achieved in different environments when pointing to different target positions. As can be seen, SPing achieves a better FMR in environments with more reflectors. For example, it on average achieves a 0.26% FMR in Scene 1 and achieves a 1.26% FMR in Scene 4.

Performance of collision avoidance. In this experiment, we consider the case with multiple transmitters. We let three robots (located on positions Tx1-Tx3 in Fig. 18) send request messages with random intervals (averaged at 1s). They perform carrier sense to avoid collision between messages. Figs. 20 (a) and (b) show the TMR and the collision rate of each transmitter. Here, the collision rate R_c is defined as $R_c = n_c/N$, where N is the number of the transmitted message, and n_c is the number of messages that fail to be decoded due to collision. We can see that the matching rate is higher than 92% and the collision rate is lower than 7%.

In summary, we could see that SPing is robust to cluttered and multipath-rich indoor environments. Unlike past approaches which consider multipath as detrimental, SPing leverages the rich multipath signals to create a highly distinctive physical-world address, thereby enhancing spatial resolution.

C. Time overhead evaluation

We further evaluate the latency of SPing on Raspberry Pi 5. We first compare SPing's multipath profiling delay with BVH [76], a commonly used ray tracing technique. We consider two baseline methods: i) pure BVH; and ii) BVH + SMM, where we first extract the point cloud of dominant reflectors using SPing's 3D mapping module (SMM), then use BVH to ray trace only the extracted part. The experiment is performed in 10 real environments with various complex layouts. Fig. 21 (a) shows the averaged latency of the three methods. As illustrated, pure BVH achieves a 9.6s latency. Adding the SMM module reduces the latency by 82.42%. Adding SPing's multipath profiling module (SMP) further reduces the latency to 5.8ms. SMP outperforms the BVH method to a great extent in time overhead because it avoids the time-consuming mesh generation and tree-structure construction process.

Fig. 21 (b) shows the end-to-end latency of SPing, where the worst-case performance is 411.3ms, indicating that SPing is suitable for mobile scenarios. The latency of SPing is mainly from the 3D mapping module, which consumes 308.5ms at worst. Future work will explore reducing the latency by designing an end-to-end DNN-based 3D mapping module for fast dominant reflector detection and modeling.

D. Comparison with the baseline

Then, we evaluate the baseline and compare its results with SPing. The experiments were conducted in the same indoor environment as shown in Fig. 18. We conducted 10 separate trials, each lasting 10 minutes. In each trial, the two robots entered the environment from two randomly selected starting points among Tx1-Tx3. Throughout the tests, the robots explored the scene with random trajectories at speeds ranging from 0.1 to 0.2 m/s. During collaborative SLAM, the coordinate system alignment interval for Swarm-SLAM was set to its default value of 10s. Whenever a robot detected its peer within visual range, it attempted to establish a connection via coordinate-based addressing at 2-second intervals following the workflow introduced in Sec. VI-A.

To fully demonstrate the limitations of the baseline, we use additional metrics here: i) *Cold-start delay*, which is the elapsed time from the start of a SLAM trial to the first time the system starts a coordinate system alignment process.; ii) *Alignment delay*, which is the duration of the coordinate system alignment process; iii) *Coordinate matching error*, which is the Euclidean distance between the transmitter's estimated coordinate and the receiver's self-localized coordinates, both represented in the same local coordinate system.

The results are presented in Figs. 22 & 23. As shown in Fig. 22 (a), the baseline suffers from a prohibitive average cold-start delay of 62.97s, primarily due to the fact that coordinate system alignment requires the accumulation of sufficient scene overlap to be initiated. The average delay for coordinate system alignment is 6.15s, with a large standard deviation of 5.84s. This is because the delay of coordinate system alignment grows over time as historical trajectory data accumulates. Fig. 22 (b) shows that, as the mission progresses, this delay exceeds the system's 10-s update interval after running for only several minutes, making timely alignment unfeasible on low-end platforms like Raspberry Pi. In contrast, SPing is able to establish an instant connection in an unknown environment with an average end-to-end delay of only 0.37 seconds, making it over 150 times faster than the baseline system. SPing does not depend on prior visual information or the prohibitive computational overhead associated with the frequent alignment process.

Regarding accuracy, Fig. 23 indicates that even with periodic alignment, the baseline fails to maintain reliable spatial consistency between robots compared to SPing. The system exhibits an average coordinate matching error of 0.36m, resulting in a true matching rate of only 81.50% at a 0.5-m tolerance threshold. This inaccuracy is attributed to inherent odometry drift and the dynamic nature of SLAM back-end

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

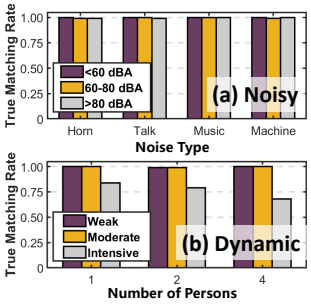


Fig. 24. Performance in hostile environments.

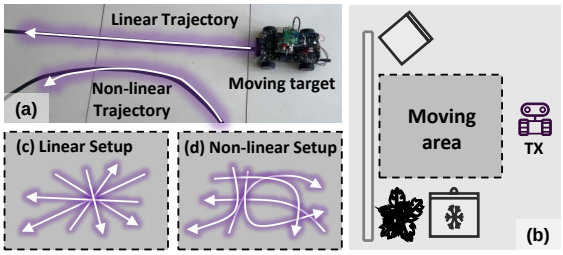


Fig. 25. Setting of the mobile evaluation.

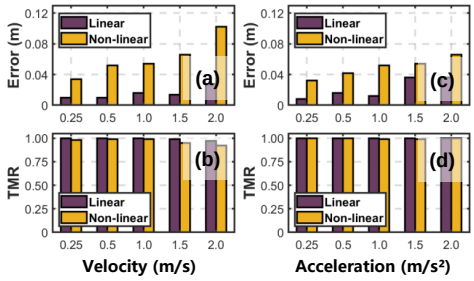


Fig. 26. Performance under: uniform ((a)-(b)) and variable (c)-(d)) motions.

optimization, which continuously alters local map coordinates and invalidates previous alignment relationships. In contrast, SPing is able to distinguish target receivers spaced only 0.5 m apart (equivalent to a 0.25-m tolerance threshold) with a 99.58% true matching rate.

Collectively, these results confirm that the SLAM-based baseline system lacks the real-time capability and the addressing accuracy and stability required for "See-and-Point" communication, consistent with the analysis in Sec. II.

E. Performance in hostile environments

This section evaluates SPing's performance in hostile environments with environmental dynamics and ambient noises. All the experiments are performed in Scene 0 (Fig. 16 (a)).

■ **Impact of ambient noises.**

Although **SPing operates in the frequency band that is higher than most everyday noises** (>8kHz [96]), it could still be affected by ambient noises due to the nonlinearity in acoustic devices. We in this section conduct an experiment to examine how different ambient noises affect the performance of SPing. We consider four types of noises: train horn, human talk, rock music, and machinery noise. We collect different free audio sources of these four types of noises and replay them during the experiment. For each kind of noise, we play them at different levels of sound pressure. The results are shown in Fig. 24 (a). As can be seen, SPing consistently achieves a nearly 100% TMR under all kinds of noises.

■ **Impact of environmental dynamics.**

We then examine the performance under dynamic occlusions. With the approval of IRB, we invite four volunteers to participate in the experiment. They are allowed to perform three types of activities in the testing area, which simulates three levels of dynamics: i) weak dynamic, where volunteers are allowed to sit in any preferred location and talk to each other; ii) moderate dynamic, where the volunteers are allowed to move around the testing areas, but will not disturb the LOS path between the transceivers; and iii) intensive dynamic, where the volunteers will occasionally block the LOS path.

Fig. 24 (b) shows the TMR achieved under different levels of dynamics and with different amounts of volunteers. The results tell that: i) SPing is resistant to small-scale dynamics with a clear LOS path since it uses only the dominant reflectors for address generation, which are hard to be blocked due to their large sizes. It achieves consistently high TMR ($\geq$99%) as long as the LOS path between the transceivers is not blocked. ii) The matching rate suffers obvious degradation when the LOS path is blocked.

F. Mobile scenario evaluation

In this section, we consider a practical scenario where the receivers are in mobility. Fig. 25 shows the setting of the experiment, where we fix the location of the transmitter and the reflectors and put the receiver on a vehicle which could move with controllable trajectory and speed. We consider 10 different trajectories, including the linear and the non-linear ones, as illustrated in Figs. 25 (c) & (d). For each trajectory, we let the receiver move with two motion patterns: i) uniform motion, where the receiver moves at a constant speed. We tested five different speeds, 0.25m/s, 0.5m/s, 1m/s, 1.5m/s, and 2m/s. ii) variable motion, where the receiver accelerates from 0.5m/s with a constant acceleration. We tested five different accelerations, 0.25m/s^2, 0.5m/s^2, 1m/s^2, 1.5m/s^2, and 2m/s^2.

Figs. 26 (a) & (c) show the average error between the predicted receiver location and the ground truth under different motion patterns. We can see that although the error slightly increases as mobility becomes increasingly intense, the tracking error remains below 0.1m. Figs. 26 (b) & (d) show the matching rate results. As can be seen, SPing constantly achieves a higher than 97% matching rate, when the robot moves in a straight line at speeds of up to 2m/s, which is comparable to the operating speeds of mainstream commercial robots (e.g., Boston Dynamics [97]). The matching rate decreases when the robot moves along non-linear trajectories. This is because SPing predicts the location of the target robot using the KF algorithm, which assumes a linear motion of the robot. The performance of SPing can be further improved by using a more advanced target-tracking algorithm.

G. Impact of different parameters

■ **Impact of the spacing between receivers**

We in this experiment evaluate the spatial resolution of SPing. Specifically, we test SPing's performance in differentiating receivers with different spacings. Fig. 27 shows the results obtained in environments with dense (Scene 0&1), moderate (Scene 2&3), and sparse (Scene 4) multipath. As expected, SPing achieves better spatial resolution in a multipath-rich environment. For example, it achieves a 0.3m resolution in Scene 0 while achieving a 0.5m resolution in Scene 4. We note that even in the case where receivers are too close to be differentiated, SPing can still coordinate their transmission leveraging the mechanism proposed in Sec. V-E.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

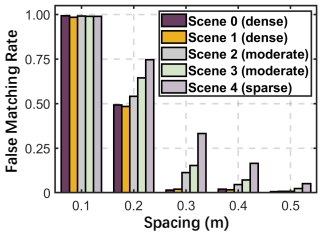
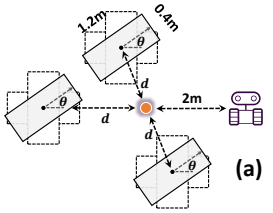


Fig. 27. Impact of spacing.

Fig. 28. Impact of reflector numbers.



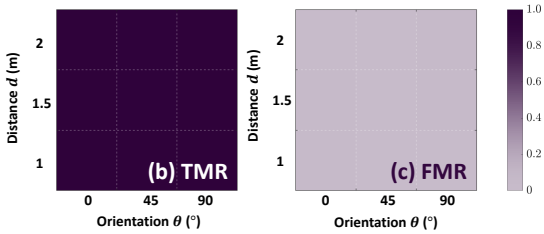


Fig. 29. Impact of reflector's orientation and distance to the receiver: (a) experiment setting; (b) TMR performance; (c) FMR performance.

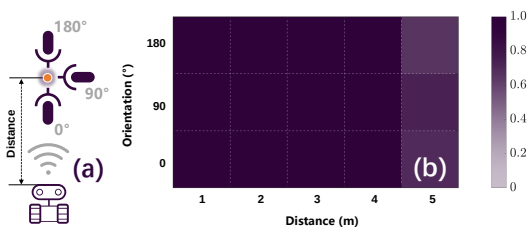


Fig. 30. Impact of orientation and distance between transceivers.

■ **Impact of the number of reflectors.**

Using more reflectors helps to generate a more distinct multipath profile for the target receiver, which increases the location-distinguishability of SPing. To evaluate the impact of the number of reflectors, we perform an experiment in a testing area with three dominant reflectors, as shown in Fig. 28 (a). Figs. 28 (b1-b3) show the matching rate of the target and non-target areas achieved when using different numbers of reflectors. Evidently, SPing achieves a higher location-distinguishability when using more reflectors for address generation. When using only one reflector (e.g., reflector ❶), SPing cannot distinguish the five receiving positions along the line parallel to the surface of the reflector. When adding reflector ❷, SPing can narrow down the target area to a 0.25 m^2 spot. When using all three reflectors, SPing can further decrease the FMR of the non-target positions.

We repeat the above experiment in another 10 different environments, each with 1-3 randomly placed reflectors. Fig. 28 (c) shows the average TMR and FMR achieved under different numbers of reflectors. As can be seen, even with only two reflectors, SPing can already establish a connection precisely to only the target receiver, with higher than 98% TMR and lower than 3% FMR.

■ **Impact of location and orientation of the reflectors.**

Besides the number of reflectors, the orientations and locations of the reflectors, which determine the path power, can also affect the performance of SPing. In this experiment, we consider a scenario with three reflectors, as shown in Fig. 29 (a). We vary the average reflector-to-receiver distance d from 1m to 2m. The orientation θ of the reflector are varied from 0° to 90°. Under each setting, we observe the TMR of the target receiver and the FMR of a non-target receiver which is located 0.5m away from the target receiver.

The results are shown in Figs. 29 (b) and (c). As can be seen, SPing consistently achieves a higher than 97% TMR and lower than 3% FMR across different settings of the reflectors' orientations and locations. SPing achieves this because it leverages only the delay of the paths to generate the physical address, so the variation in the power of each path does not affect the address generation and matching process.

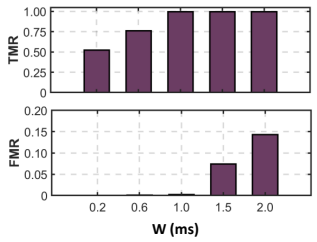


Fig. 31. Performance under different W.

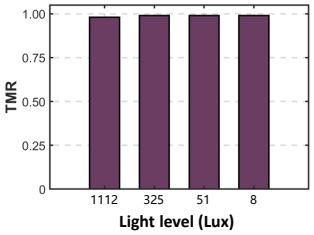


Fig. 32. Performance under different light levels.

■ **Impact of transmission distance and direction.**

In this experiment, we vary the distance between the transmitter and the receiver from 1m to 5m. For each distance, we vary the orientation of the transmitter from 0° to 180°, as shown in Figs. 30 (a) and (b) show the TMR achieved under each setting of the distance and the orientation. The results tell that: i) The TMR keeps higher than 98% within the range of 4m. The bottleneck comes from the YOLO-based target detection module, which inaccurately detects the target with too small size on the image. ii) The receiver's orientation does not have an apparent impact on the performance.

■ **Impact of window length during fuzzy address matching.**

As mentioned in Sec. V-D, SPing detects the path peak using a window to tolerate possible peak position errors. The window length W can impact the TMR and FMR of SPing: a larger window can tolerate a larger peak error, which increases the TMR but may incur a higher FMR. We perform a series of experiments to select an appropriate window size W that balances the TMR and FMR. Specifically, we compare the average TMR and FMR under different values of W in Scene 0. As shown in Fig. 31, when we use a short window (e.g., when W=0.2ms), the FMR is nearly 0%, but the TMR is only 52.2%. As W increases to 2ms, TMR keeps higher than 99%, but FMR reaches 14.29% (at 2ms). Based on the results in Fig. 31, we set the window size as W=1ms, which can produce both high TMR (99.58%) and low FMR (0.21%).

■ **Impact of lighting levels.**

We in this experiment evaluate SPing's performance under

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

poor lighting conditions. Before presenting the experimental results, we first clarify that SPing could work in poor lighting conditions because it leverages an RGB-D depth camera to acquire the depth map and the RGB image for 3D mapping. Here, as mentioned in Sec. VI-A, the depth perception depends on the binocular structured lighting technology, which does not depend on ambient light, and thus, the lighting condition has little impact on the depth map. Although the RGB image may distort in dark environments, which may affect the performance of image segmentation in the 3D mapping process, modern cameras usually feature automatic exposure adjustment to improve the image quality. Additionally, the segmentation model has been trained on millions of image samples, giving it adequate robustness.

We validate the robustness of SPing in an indoor room under four different light conditions: i) Ample daylight (1112 Lux): We open the window and make the room sun-filled in the daytime; ii) Ample artificial light (325 Lux): We turn on the room's ceiling light at night; iii) Weak daylight (51 Lux): We draw the curtains in the daytime; iv) Weak artificial light (8 Lux): We used a phone flashlight to illuminate the experiment area. We reuse the experimental setting in Fig. 28 (a), and the results in Fig. 32 show that SPing could even work under a weak phone flashlight with higher than 98% TMR.

VII. DISCUSSION

Reliance on visual access to receivers. SPing requires visual access to receivers for connection establishment. This is the prerequisite for robots to accomplish visually triggered tasks. Specifically, the so-called "see-and-point" communication paradigm we want to enable in this work means that a robot can select any other robot in its vision field as a partner to communicate or cooperate with. Consider the case shown in Fig. 1 (b), Robot A wants to fetch an item which is far away, it may request one of its partners (e.g., Robot B), who it sees being close to the item it requires, to help fetch it. Note that one would not like to ask a robot who is not even in its vision filed for help. Traditional methods struggle to address this problem (i.e., how can a robot instantly build a connection pointing to a specific robot in its vision field, without any pre-existing network). SPing is the first to fill this critical gap.

Defense against malicious attackers. The design of SPing does not include explicit mechanisms to defend against a malicious attacker that spoofs the system by sending the ACK signal even if its location does not match the predicted multi-path profile. This may leave the system open to impersonation or denial-of-service attacks. As a physical-layer technology for connection establishment, SPing delegates security to upper layers and is compatible with the existing authentication and encryption mechanisms [98], [99]. For instance, a transmitter can verify a receiver's authenticity by checking if the round-trip time (RTT) of the message (which reflects the propagation distance of the signal) aligns with the visually-measured distance to the target. A mismatch would expose a spoofer at a different physical location. Furthermore, we could also leverage authentication mechanisms that have been widely used in existing protocols, such as WPA2 in Wi-Fi, for receiver authentication. Specifically, we can utilize the Advanced Encryption Standard (AES) [100] to encrypt the message, making the shared message more robust against unauthorized access and data interception.

Communication overhead for large swarms. When SPing is deployed in large swarms, a robot may continuously listen for chirps sent from other robots and run the matching process. This may raise concerns about computational overhead and power consumption. What we want to clarify here is that the chirp listening and matching will not incur unacceptable computational cost in SPing. Specifically, the proposed fuzzy address matching requires an average latency of only 8.04ms with a maximum memory usage of 6.9MB on Raspberry Pi 5 to process a request message. Considering that the Raspberry Pi 5 consumes less than 7W of power under stressful tasks, SPing can continuously handle over one million request messages powered by a lightweight 5000mAh battery. Furthermore, modern SoCs generally contain multiple cores, enabling parallel execution of SPing's matching algorithm and other important tasks. It should also be noted that the matching process occurs only for initial connections between robots. Once the digital IDs are exchanged, the robots can revert to traditional communication schemes.

VIII. CONCLUSION

We in this paper propose SPing, a physical-world addressing mechanism for autonomous robots. SPing enables a "see-and-point" communication mode: a robot can select any of the robots that it sees in its vision field, and establish an instant connection pointing to that robot, without knowing the robot's IP address. We build an end-to-end prototype of SPing and evaluate its performance in various environments. The results show that SPing can always establish a connection precisely pointing to the target robot, even though the robot is in fast mobility. It achieves a 99.58% true matching rate and a 0.21% false matching rate, even when a non-target robot is located 0.3m∼0.5m away from the target robot.

REFERENCES

[1] J. P. Queralta, J. Taipalmaa, P. B. Can, V. K. Sarker, G. T. Nguyen, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," in *IEEE Access*, 2020.

[2] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," in *Autonomous Robots*, 2016.

[3] D. Sun, Z. Liu, and X. Zhang, "Dynamic cooperative communications with mutual information accumulation for mobile robots in industrial internet of things," *Sensors*, vol. 24, no. 13, p. 4362, 2024.

[4] H. Dong, Y. Xie, X. Zhang, W. Wang, X. Zhang, and J. He, "Gpsmirror: Expanding accurate gps positioning to shadowed and indoor regions with backscatter," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3592511

[5] S. Yue, H. He, P. Cao, K. Zha, M. Koizumi, and D. Katabi, "Cornerradar: Rf-based indoor localization around corners," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 1, Mar. 2022. [Online]. Available: https://doi.org/10.1145/3517226

[6] J. Yang, B. Dong, and J. Wang, "Vuloc: Accurate uwb localization for countless targets without synchronization," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 3, Sep. 2022. [Online]. Available: https://doi.org/10.1145/3550286

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

[7] D. Guo, C. Gu, L. Jiang, W. Luo, and R. Tan, "Illoc: In-hall localization with standard lorawan uplink frames," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 1, Mar. 2022. [Online]. Available: https://doi.org/10.1145/3517245

[8] Y. Ding, D. Jiang, Y. Liu, D. Zhang, and T. He, "Smartloc: Indoor localization with smartphone anchors for on-demand delivery," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 4, Dec. 2022. [Online]. Available: https://doi.org/10.1145/3494972

[9] H. Li, X. Chen, J. Wang, D. Wu, and X. Liu, "Dafi: Wifi-based device-free indoor localization via domain adaptation," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 4, Dec. 2022. [Online]. Available: https://doi.org/10.1145/3494954

[10] M. Zhao, T. Chang, A. Arun, R. Ayyalasomayajula, C. Zhang, and D. Bharadia, "Uloc: Low-power, scalable and cm-accurate uwb-tag localization and tracking for indoor applications," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2021.

[11] J. Xiong and K. Jamieson, "Arraytrack: a fine-grained indoor location system," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation (NSDI 13)*, 2013.

[12] B. Liang, P. Wang, R. Zhao, H. Guo, P. Zhang, J. Guo, S. Zhu, H. H. Liu, X. Zhang, and C. Xu, "RF-Chord: Towards deployable RFID localization system for logistic networks," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023.

[13] Y.-L. Wei, C.-J. Huang, H.-M. Tsai, and K. C.-J. Lin, "Celli: Indoor positioning using polarized sweeping light beams," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '17, 2017.

[14] H. Abdelnasser, R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef, "Semanticslam: Using environment landmarks for unsupervised indoor localization," *IEEE Transactions on Mobile Computing*, 2016.

[15] D. Jaisinghani, R. K. Balan, V. Naik, A. Misra, and Y. Lee, "Experiences & challenges with server-side wifi indoor localization using existing infrastructure," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ser. MobiQuitous '18, 2018.

[16] J. Xiong, K. Jamieson, and K. Sundaresan, "Synchronicity: pushing the envelope of fine-grained localization with distributed mimo," in *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless*, ser. HotWireless '14, 2014.

[17] C. Cai, H. Pu, P. Wang, Z. Chen, and J. Luo, "We hear your pace: Passive acoustic localization of multiple walking persons," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2021.

[18] W. Wang, L. Mottola, Y. He, J. Li, Y. Sun, S. Li, H. Jing, and Y. Wang, "Micnest: Long-range instant acoustic localization of drones in precise landing," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '22, 2023.

[19] K. M. Bae, H. Moon, and S. M. Kim, "Supersight: Sub-cm nlos localization for mmwave backscatter," in *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, ser. MOBISYS '24, 2024.

[20] R. Nandakumar, V. Iyer, and S. Gollakota, "3d localization for sub-centimeter sized devices," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '18, 2018.

[21] J. Xu, H. Cao, Z. Yang, L. Shangguan, J. Zhang, X. He, and Y. Liu, "SwarmMap: Scaling up real-time collaborative visual SLAM at the edge," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022.

[22] J. Xu, H. Cao, D. Li, K. Huang, C. Qian, L. Shangguan, and Z. Yang, "Edge assisted mobile semantic visual slam," in *IEEE INFOCOM 2020*, 2020.

[23] W. Xu, Z. Li, W. Xue, X. Yu, B. Wei, J. Wang, C. Luo, W. Li, and A. Y. Zomaya, "Inaudiblekey: Generic inaudible acoustic signal based key agreement protocol for mobile devices," in *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2021)*, 2021.

[24] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 17)*, 2017.

[25] C.-Y. Hsu, R. Hristov, G.-H. Lee, M. Zhao, and D. Katabi, "Enabling identification and behavioral sensing in homes using radio reflections," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19, 2019.

[26] Y. Bai, N. Garg, and N. Roy, "Spidr: ultra-low-power acoustic spatial sensing for micro-robot navigation," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '22, 2022.

[27] Y. Xie, J. Xiong, M. Li, and K. Jamieson, "md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19, 2019.

[28] S. Shen, D. Chen, Y.-L. Wei, Z. Yang, and R. R. Choudhury, "Voice localization using nearby wall reflections," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '20, 2020.

[29] Y. Su, F. Zhang, K. Niu, T. Wang, B. Jin, Z. Wang, Y. Jiang, D. Zhang, L. Qiu, and J. Xiong, "Embracing distributed acoustic sensing in car cabin for children presence detection," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 1, Mar. 2024. [Online]. Available: https://doi.org/10.1145/3643548

[30] Z. Wang, Y. Wang, M. Tian, and J. Shen, "Hearfire: Indoor fire detection via inaudible acoustic sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, Jan. 2023. [Online]. Available: https://doi.org/10.1145/3569500

[31] M. Chen, L. Lu, J. Wang, J. Yu, Y. Chen, Z. Wang, Z. Ba, F. Lin, and K. Ren, "Voicecloak: Adversarial example enabled voice de-identification with balanced privacy and utility," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, no. 2, Jun. 2023. [Online]. Available: https://doi.org/10.1145/3596266

[32] S. Mahmud, V. Parikh, Q. Liang, K. Li, R. Zhang, A. Ajit, V. Gunda, D. Agarwal, F. Guimbretiere, and C. Zhang, "Actsonic: Recognizing everyday activities from inaudible acoustic wave around the body," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 4, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3699752

[33] T. C. Yu, G. Hu, R. Zhang, H. Lim, S. Mahmud, C.-J. Lee, K. Li, D. Agarwal, S. Nie, J. Oh, F. Guimbretière, and C. Zhang, "Ring-a-pose: A ring for continuous hand pose tracking," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 4, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3699741

[34] S. Wang, L. Zhong, Y. Fu, L. Chen, J. Ren, and Y. Zhang, "Uface: Your smartphone can "hear" your facial expression!" *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 1, Mar. 2024. [Online]. Available: https://doi.org/10.1145/3643546

[35] Z. Xu, T. Liu, R. Jiang, P. Hu, Z. Guo, and C. Liu, "Aface: Range-flexible anti-spoofing face authentication via smartphone acoustic sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 1, Mar. 2024. [Online]. Available: https://doi.org/10.1145/3643510

[36] Z. Chen, X. Zhang, S. Wang, Y. Xu, J. Xiong, and X. Wang, "Enabling practical large-scale mimo in wlans with hybrid beamforming," *IEEE/ACM Transactions on Networking*, 2021.

[37] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "BeamSpy: Enabling robust 60 GHz links under blockage," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016.

[38] L. Fan, L. Xie, W. Zhou, C. Wang, Y. Bu, and S. Lu, "Beamforming for sensing: Hybrid beamforming based on transmitter-receiver collaboration for millimeter-wave sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 2, May 2024. [Online]. Available: https://doi.org/10.1145/3659619

[39] F. Zhang, Z. Chang, J. Xiong, R. Zheng, J. Ma, K. Niu, B. Jin, and D. Zhang, "Unlocking the beamforming potential of lora for long-range multi-target respiration sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 2, Jun. 2021. [Online]. Available: https://doi.org/10.1145/3463526

[40] S. He, W. Ma, H. Dong, L. Xiao, and T. Jiang, "C-cube: Rethinking distributed beamforming for concurrent charging in backscatter networks," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, Jan. 2023. [Online]. Available: https://doi.org/10.1145/3570342

[41] Y. Feng, J. Zhao, C. Wang, L. Xie, and S. Lu, "3d bounding box estimation based on cots mmwave radar via moving scanning," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 4, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3699758

[42] Q. Yang, H. Wu, Q. Huang, J. Zhang, H. Chen, W. Li, X. Tao, and Q. Zhang, "Side-lobe can know more: Towards simultaneous communication and sensing for mmwave," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, Jan. 2023. [Online]. Available: https://doi.org/10.1145/3569498

[43] Y. Yang, H. Xu, Q. Chen, J. Cao, and Y. Wang, "Multi-vib: Precise multi-point vibration monitoring using mmwave radar," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 4, Jan. 2023. [Online]. Available: https://doi.org/10.1145/3569496

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

[44] K. Prabhath, S. K. Jayaweera, and S. A. Lane, "Intelligent reflecting surface orientation optimization to enhance the performance of wireless communications systems," in *2022 International Workshop on Antenna Technology (iWAT)*. IEEE, 2022, pp. 231–234.

[45] K. Prabhath and S. K. Jayaweera, "Optimal adaptation of 3d beamformers in uav networks," in *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2023, pp. 207–212.

[46] M. Curran, M. S. Rahman, H. Gupta, K. Zheng, J. Longtin, S. R. Das, and T. Mohamed, "Fsonet: A wireless backhaul for multi-gigabit picocells using steerable free space optics," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017.

[47] M. Ghobadi, R. Mahajan, A. Phanishayee, N. Devanur, J. Kulkarni, G. Ranade, P.-A. Blanche, H. Rastegarfar, M. Glick, and D. Kilper, "Projector: Agile reconfigurable data center interconnect," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016.

[48] C. J. Carver, H. Schwartz, Q. Shao, N. Shade, J. Lazzaro, X. Wang, J. Liu, E. Fossum, and X. Zhou, "Catch me if you can: Laser tethering with highly mobile targets," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024.

[49] H. Gupta, M. Curran, J. Longtin, T. Rockwell, K. Zheng, and M. Dasari, "Cyclops: an fso-based wireless link for vr headsets," in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022.

[50] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer, "Firefly: a reconfigurable wireless data center fabric using free-space optics," *SIGCOMM Comput. Commun. Rev.*, 2014.

[51] M. Curran, K. Zheng, H. Gupta, and J. Longtin, "Handling rack vibrations in fso-based data center architectures," in *2018 International Conference on Optical Network Design and Modeling (ONDM)*, 2018.

[52] X. Su, R. Zhang, H. Zhou, H. Song, K. Zou, H. Song, Y. Duan, K. Pang, N. Hu, Y. Zhou, R. W. Boyd, M. Tur, and A. E. Willner, "Experimental demonstration of enhanced misalignment tolerance for recovering phase and amplitude encoding in a pilot-assisted self-coherent free-space optical link," in *2022 Conference on Lasers and Electro-Optics (CLEO)*, 2022.

[53] P. Martin, A. Symington, and M. Srivastava, "Slats: Simultaneous localization and time synchronization," in *ACM Trans. Cyber-Phys. Syst.*, 2018.

[54] J. Jung, K. Kim, W. Lee, and H. Kim, "Poster: Asynchronous acoustic localization using commercial devices," in *ACM SenSys*, 2015.

[55] C. Cai, Z. Chen, H. Pu, L. Ye, M. Hu, and J. Luo, "Acute: acoustic thermometer empowered by a single smartphone," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, ser. SenSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 28–41. [Online]. Available: https://doi.org/10.1145/3384419.3430714

[56] X. Zhang, L. Chen, M. Feng, and T. Jiang, "Toward reliable non-line-of-sight localization using multipath reflections," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 1, Mar. 2022. [Online]. Available: https://doi.org/10.1145/3517244

[57] Y. Fu, Y. Zhang, H. Pan, Y. Lu, X. Li, L. Chen, J. Ren, X. Li, X. Zhang, and Y. Zhang, "Pushing the limits of acoustic spatial perception via incident angle encoding," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, no. 2, May 2024. [Online]. Available: https://doi.org/10.1145/3659583

[58] Y.-H. Su, C. J. Yang, E. Hwang, and A. P. Sample, "Single packet, single channel, switched antenna array for rf localization," vol. 7, no. 2, Jun. 2023. [Online]. Available: https://doi.org/10.1145/3596263

[59] T. C. Tai, K. C. J. Lin, and Y. C. Tseng, "Toward reliable localization by unequal aoa tracking," in *ACM MobiSys*, 2019.

[60] M. R. Figueroa, P. K. Bishoyi, and M. Petrova, "Cooperative multi-monostatic sensing for object localization in 6g networks," in *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, 2024.

[61] T. Zhang, D. Zhang, G. Wang, Y. Li, Y. Hu, Q. sun, and Y. Chen, "Rloc: Towards robust indoor localization by quantifying uncertainty," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, no. 4, Jan. 2024. [Online]. Available: https://doi.org/10.1145/3631437

[62] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE transactions on robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[63] M. Labbé and F. Michaud, "Multi-session visual slam for illumination-invariant re-localization in indoor environments," *Frontiers in Robotics and AI*, vol. 9, p. 801886, 2022.

[64] P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, "Towards collaborative simultaneous localization and mapping: a survey of the current research landscape," *Field Robotics*, vol. 2, pp. 971–1000, 2022.

[65] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

[66] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed trajectory estimation with privacy and communication constraints: a two-stage distributed gauss-seidel approach," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5261–5268.

[67] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[68] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2017.

[69] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[70] D. Freedman, R. Pisani, and R. Purves, "Statistics (international student edition)," *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[71] T. Forrest, "From sender to receiver: Propagation and environmental effects on acoustic signals," in *Integrative and Comparative Biology - INTEGR COMP BIOL*, 1994.

[72] L. Savioja and U. P. Svensson, "Overview of geometrical room acoustic modeling techniques," in *The Journal of the Acoustical Society of America*, 2015.

[73] S. Siltanen, T. Lokki, S. Kiminki, and L. Savioja, "The room acoustic rendering equation," in *The Journal of the Acoustical Society of America*, 2007.

[74] T. Woodford, X. Zhang, E. Chai, K. Sundaresan, and A. Khojastepour, "Spacebeam: Lidar-driven one-shot mmwave beam management," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021.

[75] A. Adewale, "Performance evaluation of monte carlo based ray tracer," in *The Journal of Computational Science Education*, 2021.

[76] D. Meister, S. Ogaki, C. Benthin, M. Doyle, M. Guthe, and J. Bittner, "A survey on bounding volume hierarchies for ray tracing," in *Computer Graphics Forum*, 2021.

[77] T. Viitanen, M. Koskela, P. Jääskeläinen, H. Kultala, and J. Takala, "Mergetree: A fast hardware hlbvh constructor for animated ray tracing," in *ACM Trans. Graph.*, 2017.

[78] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1689–1696.

[79] Ultralytics, "Instance segmentation," https://docs.ultralytics.com/tasks/segment/.

[80] H. Li, S. Wang, Z. Bai, H. Wang, S. Li, and S. Wen, "Research on 3D reconstruction of binocular vision based on thermal infrared," in *Sensors (Basel)*, 2023.

[81] S. Nadimi, V. Angelidakis, M. Otsubo, and A. Ghanbarzadeh, "How can the effect of particle surface roughness on the contact area be predicted?" *Computers and Geotechnics*, vol. 150, p. 104890, 2022.

[82] S. N. Metodiev and K. Ushkala, "An integral active contour model for convex hull and boundary extraction," in *Advances in Visual Computing*, 2009.

[83] T. Chen, J. Chan, and S. Gollakota, "Underwater messaging using mobile devices," in *Proceedings of the ACM SIGCOMM 2022 Conference*, ser. SIGCOMM '22, 2022.

[84] K. Qian, Y. Lu, Z. Yang, K. Zhang, K. Huang, X. Cai, C. Wu, and Y. Liu, "AIRCODE: Hidden Screen-Camera communication on an invisible and inaudible dual channel," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, 2021.

[85] S. K., T. H. Kim, J. Y. Ha, S. H. Lim, S. C. Shin, J. W. Choi, C. Kwak, and S. Choi, "Near-ultrasound communication for tv's 2nd screen services," in *MobiCom*, 2016.

[86] Ultralytics, "Object detection," https://docs.ultralytics.com/tasks/detect/.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3665613

[87] M. Derakhshani, S. Masoudnia, A. Shaker, O. Mersa, M. Sadeghi, M. Rastegari, and B. Araabi, "Assisted excitation of activations: A learning technique to improve object detectors," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[88] ORBBEC, "Orbbec gemini2," https://www.orbbec.com/products/stereo-vision-camera/gemini-2/.
[89] R. Han, H. Yan, and L. Ma, "Research on 3d reconstruction methods based on binocular structured light vision," in *Journal of Physics: Conference Series*, vol. 1744, no. 3. IOP Publishing, 2021, p. 032002.
[90] A. Developer, "Neon instructions," https://developer.arm.com/documentation/dht0002/a/Introducing-NEON/NEON-architecture-overview/NEON-instructions.
[91] Tencent, "Ncnn," https://github.com/Tencent/ncnn.
[92] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. Berg, W. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
[93] P.-Y. Lajoie and G. Beltrame, "Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 475–482, 2023.
[94] M. Lab, "Swarm-slam github repository," https://github.com/MISTLab/Swarm-SLAM, 2022.
[95] S. Zang, "Orb_slam3_ros2 github repository," https://github.com/zang09/ORB_SLAM3_ROS2, 2023.
[96] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su, "Messages behind the sound: real-time hidden acoustic signal capture with smartphones," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016.
[97] B. Dynamics, "Boston dynamics spot," https://www.flymotionus.com/products/spot-robot.
[98] S. A. A. Ahadi, N. rakesh, and S. varshney, "Overview on public wi-fi security threat evil twin attack detection," in *2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, 2020, pp. 1–6.
[99] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.
[100] M. A. Alrammahi and H. Kaur, "Development of advanced encryption standard (aes) cryptography algorithm for wi-fi security protocol," *International Journal of Advanced Research in Computer Science*, vol. 5, no. 3, pp. 62–67, 2014.

Huangwei Wu received his B.S. degree in Electronic Information Engineering from the College of Electronics and Information Engineering, Sichuan University, Chengdu, China. He is currently a Ph.D. candidate in Information and Communication Engineering at the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include vision-aided communication, acoustic communication and sensing, and AI in the physical layer.

Weiguo Wang received his PhD. degree in Tsinghua University, and his B.E. degree in the University of Electronic Science and Technology of China (UESTC). He is currently a researcher at NIO. His research interests include acoustic sensing and mobile computing.

Meng Jin (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Northwest University, Xi'an, China, in 2012, 2015, and 2018, respectively. She was a Post-Doctoral Researcher with the School of Software and BNRist, Tsinghua University. She is currently an Associate Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. Her main research interests include backscatter communication, wireless network coexistence at 2.4 GHz, mobile sensing, and clock synchronization. She is a member of ACM.

Zhuxuan He received the B.S. degree from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2024. She is currently working towards a M.Sc. degree in Information and Communication Engineering at the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. Her research interests include vision-aided communication, and acoustic sensing/communication.

Qi Cao received the B.S. degree from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2024. She is currently working towards a M.Sc. degree in the School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore. Her research interests include natural language processing (NLP), and information fusion.

Xinbing Wang received the BS degree (Hons.) from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 1998, the MS degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001, and the PhD degree, majoring in the electrical and computer engineering and minoring in mathematics, from North Carolina State University, Raleigh, North Carolina, in 2006. He is currently a distinguished professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. He has been an Editor at Large for the IEEE/ACM Transactions on Networking, and the Associate Editor for IEEE Transactions on Information Theory, and a member of the Technical Program Committees of several conferences including IEEE INFOCOM 2009−-2023.

Chenghu Zhou received the BS degree in geography from Nanjing University, Nanjing, China, in 1984, and the MS and PhD degrees in geographic information system from the Chinese Academy of Sciences (CAS), Beijing, China, in 1987 and 1992, respectively. He is currently an academician with the Chinese Academy of Sciences, China, where he is also a research professor with the Institute of Geographical Sciences and Natural Resources Research, and a professor with the School of Geography and Ocean Science, Nanjing University, China. His research interests include spatial and temporal data mining, geographic modeling, hydrology and water resources, and geographic information systems and remote sensing applications.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on March 24,2026 at 12:50:33 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.