CamLopa: A Hidden Wireless Camera Localization Framework via Signal Propagation Path Analysis

***Abstract*—Hidden wireless cameras pose significant privacy threats, necessitating effective detection and localization methods. However, existing localization solutions often require impractical activity spaces, expensive specialized devices, or pre-collected training data, limiting their practical deployment. To address these limitations, we introduce CamLopa, a training-free wireless camera localization framework that operates with minimal activity space constraints using low-cost, commercial-off-the-shelf (COTS) devices. CamLopa can achieve detection and localization in just 45 seconds of user activities with a Raspberry Pi board. During this short period, it analyzes the causal relationship between wireless traffic and user movement to detect the presence of a hidden camera. Upon detection, CamLopa utilizes a novel azimuth localization model based on wireless signal propagation path analysis for localization. This model leverages the time ratio of user paths crossing the First Fresnel Zone (FFZ) to determine the camera's azimuth angle. Subsequently, CamLopa refines the localization by identifying the camera's quadrant. We evaluate CamLopa across various devices and environments, demonstrating its effectiveness with a 95.37% detection accuracy for snooping cameras and an average localization error of 17.23°, under the significantly reduced activity space requirements and without the need for training. Our implementation, code, and demo are available at https://anonymous.4open.science/r/CamLoPA-Code-DFD5.**

1. Introduction

In recent years, the proliferation of wireless camera devices for home and public security has grown significantly due to their convenience and flexibility in deployment. A study by Market Research Future in 2024 [1] projected the global wireless video surveillance and monitoring market to grow at a compound annual growth rate of 16.8% from 2022 to 2030. However, the rapid adoption of wireless cameras has also raised substantial privacy concerns related to unauthorized video recording and dissemination [2], [3], [4]. Users increasingly find themselves being illegally recorded by hidden cameras in various locations, from hotel rooms to short-term rentals. For instance, a 2019 survey [5] revealed that 58% of 2,023 Airbnb guests were concerned about the possibility of hidden cameras, with 11% reporting actual discoveries of such devices. In response to these privacy threats, various jurisdictions have proposed and enacted legislation. For example, Delaware's privacy laws now strictly prohibit the use of hidden cameras in private settings without the consent of the individuals being recorded, with violations leading to severe penalties including jail time and fines [6]. These legal measures underscore the urgency of developing effective methods for detecting and localizing hidden wireless cameras [7], [8], [9].

TABLE 1: Qualitative comparison with existing approaches.

Method	Low Cost	Low User Efforts	No Training	Crowded Room
LAPD [10]	✗	✗	✓	✓
HeatDeCam [11]	✗	✓	✗	✓
Lumos [12]	✓	✗	✗	✗
SNOOPDOG [13]	✓	✗	✓	✗
MotionCompass [14]	✓	✓	✓	✗
SCamF [15]	✓	✗	✓	✗
LocCams [16]	✓	✓	✗	✓
CamLopa	✓	✓	✓	✓

Consequently, the problem of wireless camera detection and localization has attracted considerable research attention [17], [18]. However, existing solutions often face significant limitations that hinder their practical deployment. Many approaches can detect wireless cameras but cannot locate them [18], [19], [20], [21], [22]. Those capable of localization often impose complex requirements. Specifically, methods relying on lens reflection [10], [23], [24] or electromagnetic/thermal emissions [11], [25], [26] are typically cumbersome, requiring user expertise and examination of every corner of the room, making them difficult to use. Moreover, electromagnetic/thermal-based methods often necessitate costly specialized equipment. To address these shortcomings, recent research has focused on analyzing the WiFi traffic or physical layer information to locate wireless cameras. These methods usually require users to move along the edges of the room [12], [15], [27] or perform perturbations at different positions and orientations [13], [14]. The camera's location is determined by assessing the RSSI (Received Signal Strength Indicator) or traffic variations of target devices. These approaches typically necessitate the room to be nearly empty to allow user movement to different locations, which is not feasible in real-world scenarios. They are also time-consuming, requiring 10-30 minutes for camera localization and constant user movement or position adjustments. In a recent work [16], differences in WiFi Channel State Information (CSI) under Line-of-Sight (LOS) and None-Line-of-Sight (NLOS) conditions are utilized for the coarse localization of wireless cameras. This approach requires minimal user effort but its localization resolution is limited to 45^o, still taking a lot of time to

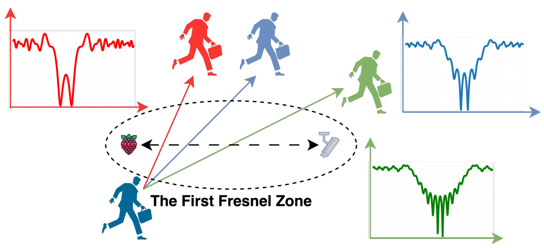


Figure 1: Different wireless signal path losses when crossing the First Fresnel Zone (FFZ) with different path lengthes.

search for devices. Additionally, it requires pre-collected training data, and the deep learning model used has poor robustness against changes in the environment and devices. (More background please refer to Appendix A)

In this paper, we introduce CamLopa, a fast and robust wireless camera detection and localization framework using low-cost commercial-off-the-shelf (COTS) devices. As shown in Table 1, CamLopa requires less activity space and user effort compared to previous studies. Our framework is inspired by the relationship between obstructions in the propagation path of wireless signals and the resulting signal attenuation. Specifically, when a large obstacle is located within the First Fresnel Zone (FFZ) between a WiFi transmitter and receiver, the transmitted signal will experience significant attenuation due to diffraction, as defined by Huygen's principle [28] and Fresnel-Kirchhoff diffraction parameters [29]. As illustrated in Figure 1, when a person crosses the FFZ, there is a drastic change in the wireless signal path loss, and the duration of this significant variation is related to the length of the path traversed through the FFZ. Since the FFZ forms an ellipse with the two devices as its foci, given a fixed distance between the two devices, the length of the path through the FFZ can be mapped to the angle of the walk relative to the LOS path (**azimuth**). CamLopa utilizes this relationship to achieve azimuth angle localization of the wireless cameras.

The technical crux of CamLopa is to address the over-complexity and lack of robustness issues in previous approaches. However, there are still two significant challenges:

1) Relationship Mapping Under Unknown User Speed: By analyzing the durations of significant wireless signal fluctuations, we can determine the time it takes for a user to traverse the FFZ. To ascertain the path length through the FFZ, we also need to know the user's speed (The challenge of constant user speed is discussed in Section 7.). In real-world scenarios, considering cost and complexity, users typically do not have specialized equipment to measure walking speed or have robots to substitute for user to move. Thus, the user's speed remains unknown, and we cannot determine the path length.

> **Q1:** How can we establish a mapping relationship between the traversal time and the azimuth angle of the hidden wireless camera without knowing the user's walking speed?

2) Errors Control Under Variable Distance and Body Size: In practical scenarios, the distance between the hidden wireless camera and the CamLopa device is also unknown, and the user's body size is variable. The user's body size significantly affects the duration of signal variations, as the signal is impacted from the moment the user enters the edge of the FFZ until he/she completely exits from it. Predefining these two values can introduce substantial errors in the aforementioned mapping relationship.

> **Q2:** How can we minimize the impacts of biased parameters and keep the errors within an acceptable range?

To overcome the above challenges, we propose a scheme called the ***orthogonal ratio***. This scheme replaces the need to measure the distance of a single path through the FFZ with the time ratio of two orthogonal paths crossing the FFZ to establish a mapping relationship with the azimuth angle. Specifically, we set two orthogonal walking paths that both pass through the CamLopa device, which is typically easy to achieve in real-world environments. We then calculate the time taken for each path to traverse the FFZ. Since the path length is the product of the time and speed, using the time ratio of the two paths eliminates the influence of the speed. Next, we develop a mapping model between the orthogonal ratio and the angle between the first path and LOS (**azimuth**) by WiFi propagation path analysis. By obtaining the orthogonal ratio in real environments, the azimuth angle of the wireless camera can be derived from the model. Besides, the orthogonal ratio remarkably reduces the impact of biased parameters such as variable distances and body sizes due to the division operation.

CamLopa operates in three stages and requires only **45** seconds of user movement to detect and locate a hidden wireless camera. In the first stage (**0**-**15**s), the system analyzes the relationship between the data stream uploaded by the camera and user activity for snooping camera detection. The encoding method of the video stream causes an increase in data volume when there is movement within the monitored area. Therefore, CamLopa first prompts the user to leave the room and collects traffic data of 15 seconds. By examining the causal relationship between the user's exit and the data stream, the system identifies whether a wireless camera is monitoring the current area. In the next stage (**15**-**35**s), the user walks along two orthogonal paths that both pass through the CamLopa equipment. The system calculates the orthogonal ratio of these two paths and determines the azimuth of the wireless camera using the azimuth model. This model only provides an angle within the range of 0-90° (e.g., for 45° and 135°, CamLopa reports 45° for both cases). To address this, we further design a scheme to determine the quadrant in which the camera is located. In the final stage (**35**-**45**s), the system prompts the user to walk along a path that coincides with the first path but does not traverse the entire FFZ. By analyzing whether the user's initial position blocks the LOS, the quadrant determination scheme identifies the quadrant in which the wireless camera is located, achieving the final localization. We implement a prototype of CamLopa on a Raspberry Pi device, which users can connect to using SSH tools on their smartphone to receive system prompts and display the results.

In summary, we make the following key contributions:

- We propose CAMLOPA, the first hidden wireless camera detection and localization framework based on the diffraction phenomenon during wireless signal propagation. This scheme is implemented using low-cost COTS devices. It has small activity space requirements, and does not require model training.
- We introduce a wireless device azimuth localization model and a quadrant determination method based on wireless signal propagation path analysis. The model is designed on the principle that diffraction causes significant attenuation of wireless signals. By combining the model with the quadrant determination method, we can achieve fast and training-free device localization.
- We evaluate CAMLOPA across various devices and environments. Experiment results show that CAMLOPA achieves the detection accuracy of 95.37% and average localization error of 17.23° for snooping wireless cameras.

2. Channel State Information (CSI)

WiFi CSI [30], [31], [32], [33], [34], [35] describes various effects that a WiFi signal undergoes during propagation, including multipath effects, attenuation, phase shift, and more. This process of influence can be represented as follows [36], [37]:

$$Y = H \cdot X + N, \tag{1}$$

where Y and X are the received and transmitted signals, respectively. N is the additive white Gaussian noise, and H is a complex matrix representing CSI. And this complex matrix can be expressed as follows:

$$H(f) = |H(f)|e^{j\theta(f)}, \tag{2}$$

where $H(f)$ is the channel response at frequency f, $|H(f)|$ is the magnitude of the CSI, representing the variation in signal strength, and $\theta(f)$ is the phase shift of the CSI, representing the variation in signal phase. The magnitude of the CSI can be used to characterize signal attenuation. The received CSI is a superposition of signals of all the propagation paths, and its Channel Frequency Response (CFR) can be represented as [38]:

$$H(f,t) = \sum_{m \in \Phi} a_m(f,t) e^{-j2\pi \frac{d_m(t)}{\lambda}}, \tag{3}$$

where f and t represent center frequency and time stamp, respectively, and m is the multi-path component. $a_m(f,t)$ and $d_m(t)$ denote the complex attenuation and propagation length of the mth multi-path component, respectively. Φ denotes the set of multi-path components and λ is the signal wavelength. When there are changes in only one path, the CSI can be used to approximate the attenuation occurring on that path. Specifically, paths with no changes and those with changes can be categorized as static and dynamic paths as follows [39]:

$$\begin{aligned} H(f,t) &= H_s(f,t) + H_d(f,t) \\ &= \sum_{m_s \in \Phi_s} a_{m_s}(f,t) e^{-j2\pi \frac{d_{m_s}(t)}{\lambda}} \\ &+ \sum_{m_d \in \Phi_d} a_{m_d}(f,t) e^{-j2\pi \frac{d_{m_d}(t)}{\lambda}}, \end{aligned} \tag{4}$$

where $H_s(f,t)$ and $H_d(f,t)$ denote the static and dynamic components, respectively. Φ_s represents the set of static paths, e.g., reflected off the walls and furniture and static body parts, while Φ_d denotes the set of dynamic paths, e.g., reflected off the moving human. When there is only one person moving in the room, CSI can be used to characterize the signal attenuation and multipath effects caused by this person's movement.

Next, we briefly explain the Fresnel zone model, which is widely used to analyze the diffraction and reflection effects of wireless and light signals along their propagation path. This model helps in understanding how signal strength varies with distance and obstacles. The Fresnel zones can be described as a series of concentric ellipses with the wireless signal transmitter and receiver as the focal points [40] (see the Appendix B).

$$|TxQ_n| + |Q_nRx| - |TxRx| = n\lambda/2, \tag{5}$$

where Q_n is a point at the boundary of the nth Fresnel zone, and Tx and Rx represent the transmitter and receiver, respectively. Since the phase difference of waves within the First Fresnel Zone (FFZ) is relatively small, most of the energy is concentrated in this region. In wireless communication and wave propagation, the energy within the FFZ typically accounts for about 60% to 70% of the total transmitted energy. Obstacles outside the FFZ primarily cause signal reflection [41], [42], [43]. The attenuation due to reflection is minimal, and the total signal energy affected by obstacles outside the FFZ is relatively small. As a result, when obstacles moves in the outside of the FFZ, the total received signal energy does not change significantly. Instead, the movement mainly causes multipath effects, leading to phase changes in the CSI. Conversely, obstacles within the FFZ mainly cause diffraction [29], [40]. The attenuation due to diffraction is substantial, and since a significant amount of signal energy is transmitted within the FFZ, the received signal experiences substantial attenuation, which can be clearly characterized by the magnitude of the CSI.

In practical systems, we can use open-source tools such as csitool [44], picosense [45], and nexmon_csi [46], [47] to obtain CSI from various network cards, including Intel 5300, AX210/AX200, and bcm43455c0 (Raspberry Pi B3+/B4). The actual size of the extracted CSI matrix depends on the number of antennas and subcarriers [48], [49], and the obtained CSI is a 4-dimensional tensor $H \in \mathbb{C}^{\mathbb{N}\times\mathbb{M}\times\mathbb{K}\times\mathbb{T}}$, and $\mathbb{M}$, $\mathbb{K}$, and $\mathbb{T}$ represent the number of receive antennas, transmit antennas, subcarriers, and packets, respectively.

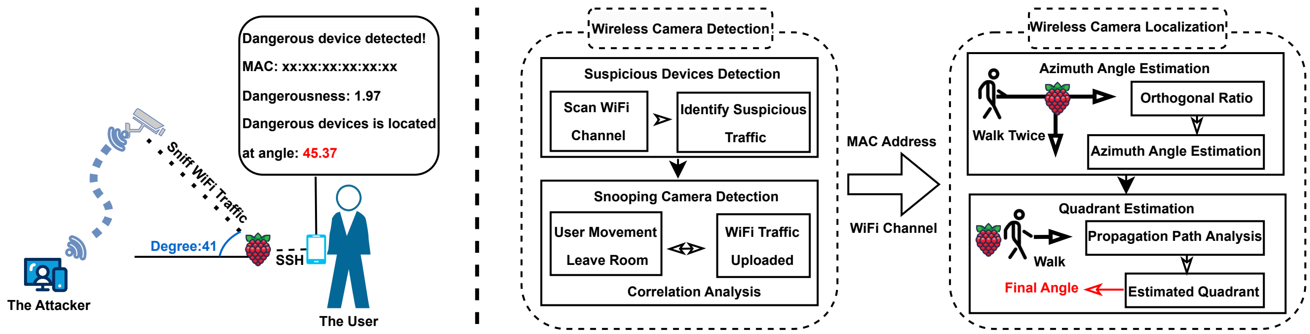


Figure 2: Overview of CamLopa. CamLopa is implemented using a low-cost Raspberry Pi, which can connect via SSH to the user's phone for prompts and notifications. The operation of CamLopa is divided into two phases: wireless camera detection and localization. The detection stage determines whether a wireless camera is monitoring the current area, while the localization stage precisely locates the identified camera.

3. Overview

3.1. Threat Model

Our work focuses on a scenario where an attacker places a hidden wireless camera in a room to monitor the user in real-time. This scenario aligns with current state-of-the-art methods [12], [15], [16], [50], [51] for detecting and locating hidden cameras. It is also supported by several real-world cases [52], [53], in which attackers have been caught live-streaming users in private spaces—an effective and convenient method for gathering private information. The adversary covertly deploys a hidden camera within the victim's room, communicating with it via encrypted wireless communication. We focus on WiFi as the communication channel in this paper, given its widespread use for remote monitoring in commercial devices. Below, we describe the real-world settings for both the attacker and the user.

Attacker: The attacker could be the host or a previous guest intending to monitor users in the room.

- The attacker can fully control the room before the user checks in, such as changing the environment and installing hidden wireless cameras.
- The attacker uses COTS camera devices to spy on users and can control the cameras through an app. Similar to previous studies [12], [13], [15], [54], [55], we assume the attacker does not alter the firmware, network protocols or wireless transmission behaviors of these camera devices, as these tasks generally require a high level of expertise.
- The attacker has complete control over the WiFi network to which the hidden wireless cameras connect. He can configure the WiFi network's wireless channels, encryption methods, and access modes.

User: The user's requirement is to detect and locate hidden wireless cameras within the room.

- The user can access the physical space to search and move around. But in a real environment, his movement is limited and obstructed by the furniture, making it difficult to meet the activity space requirements of most previous studies [12], [13], [14], [15].
- The user does not have any knowledge of the hidden wireless cameras. He is unaware of the WiFi network being used, the channel of the WiFi network, or the cameras' locations. However, the user has control over the CamLopa device, including its placement and the configuration of its network connection.
- The user does not have control over the WiFi network to which the wireless cameras are connected. However, he can use existing tools (e.g., tcpdump, Wireshark) to sniff WiFi 802.11 packets broadcast in the air. The user carries no additional measuring tools except for a Raspberry Pi equipped with CamLopa.

3.2. Workflow of CamLopa

CamLopa requires the user to perform three walks (45 seconds) to detect and locate the hidden wireless camera according to the prompts of CamLopa. It then provides feedback with the estimated azimuth angle of the hidden wireless camera. The overall structure of CamLopa is shown in Figure 2 and it operates in two phases:

Hidden Wireless Camera Detection. CamLopa first scans the surrounding WiFi networks and captures packets on all active 802.11 wireless channels for analysis. If it detects a device that is continuously uploading data, it identifies this device as suspicious and forwards its MAC address and channel index to the snooping camera detection module. The snooping camera detection module will prompt the user to leave the room and sniff packets from this channel for 15 seconds. It then analyzes the upload traffic of the suspicious device according to the MAC address. If the traffic pattern matches the user's departure phase, the detection module will report that the device is monitoring the current area. Next, the module will forward the device's MAC address and channel index to the following localization phase.

Hidden Wireless Camera Localization. Upon receiving the MAC address of the snooping wireless camera and the WiFi channel of the connected Access Point (AP), CamLopa prompts the user to walk along two orthogonal paths (see Figure 6) cross the CamLopa device, such as a Raspberry

Pi board. Specifically, the device sniffs the WiFi packets transmitted from the target MAC on the specified channel over 10 seconds for each path, extracting CSI to calculate the orthogonal ratio and determine the azimuth angle using the proposed azimuth localization model. These paths intersect in a T-shape, with the intersection point being the location of the CamLopa device. After calculating the azimuth angle, CamLopa prompts the user to walk along a path coinciding with the first path but starting in front of the CamLopa device, collecting 10 seconds of CSI. Next, using the quadrant determination model, CamLopa calculates the quadrant in which the target device is located to obtain the final azimuth angle of the hidden wireless camera.

4. Wireless Camera Detection

CamLopa detects the presence of snooping wireless cameras in the environment through wireless traffic analysis by: (i) searching for suspicious devices, and (ii) detecting snooping wireless cameras.

4.1. Searching for Suspicious Devices

In real-world environments, there are usually many wireless networks and devices connected to WiFi around the user. Analyzing all devices to detect cameras monitoring the area is highly inefficient. Therefore, CamLopa first identifies suspicious devices to narrow down the detection scope. Video stream packets are typically large and stable, and surveillance cameras continuously and frequently upload data. CamLopa starts by scanning the surrounding WiFi networks to detect all APs, even those with Hidden Service Set Identifiers (SSIDs). According to [56], CamLopa excludes APs that do not meet the minimum RSSI requirements for video streaming, namely, below -67 dBm (please refer to Appendix C). In practice, the requirements for RSSI slightly relaxed to avoid missed detections. It then sequentially scans the channels of the remaining APs, sniffing and capturing 802.11 packets for 5 seconds to determine if any devices are continuously uploading data.

For the captured 802.11 packets, CamLopa first classifies them by source MAC address into different end devices. Next, it filters out Management-Type and Control-Type frames, leaving only Data-Type frames for further analysis, as application layer data is encapsulated within Data-Type frames [57]. After protocol filtering, CamLopa aggregates all Data-Type frames corresponding to each device and calculates the average size of the payload portion. Finally, CamLopa determines the presence of any suspicious devices as follows:

$$\mathbf{S_{mac}} = \begin{cases} \mathbf{true} & \text{if } \bar{s}_{mac} > T_s \& l > T_l \& \mathbf{mac} \neq \mathbf{m_{ap}}, \\ \mathbf{false} & \text{else }. \end{cases} \tag{6}$$

Here, $\mathbf{S_{mac}}$ represents the determination of whether the device with MAC address $\mathbf{mac}$ is suspicious. $\bar{s}_{mac}$, T_s, l, $\mathbf{map}$, and T_l denote the average size of all packet payloads, the size threshold, the count of packets, the MAC address of APs, and the count threshold, respectively. This equation indicates that if a device sends a large number of packets within 5 seconds and the average packet length is long, it is likely uploading a video stream. After identifying suspicious devices, CamLopa forwards their MAC addresses and 802.11 channel index to the snooping camera detection module. This module then sequentially assesses the risk of each device to determine whether they are monitoring the current area.

4.2. Detecting Snooping Cameras

Before uploading video streams, cameras typically apply encoding to compress the data and reduce the upload volume. Most video compression standards, such as H.264 [58] and H.265 [59], achieve high compression rates through inter-frame prediction. Specifically, standard video compression algorithms use three types of frames to compress video: I (Intra-coded picture) frames, P (Predicted picture) frames and B (Bi-directionally predicted picture) frames

When there is any activity in the area monitored by the wireless camera, the camera traffic increases due to the higher number of P and B frames that need to be transmitted [13], [15]. Conversely, if the scene transitions to a stationary one, the number of disturbed pixels decreases, reducing the camera traffic. If a person first moves and then remains still within the camera's monitored area, it will result in a unique camera traffic pattern (traffic decreasing) that corresponds to the user's motion. This causal effect can be used to detect whether a hidden wireless camera is snooping on the current area. CamLopa leverages this causal relationship to detect snooping cameras. Specifically, CamLopa prompts the user to leave the room within 15 seconds. It then calculates the data throughput of each suspicious device per second and checks for traffic patterns where the throughput is initially high and then decreases. If such a pattern is detected, the device is identified as a snooping camera, and its risk level is determined based on the ratio of the data throughput in the first half to that in the second half. A sample of the data throughputs during the user's exit from the room is shown in Figure 3.

Upon detecting a snooping camera, CamLopa forwards the camera's MAC address and associated WiFi channel index to the wireless camera localization module. It then initiates the localization process for the detected camera.

5. Wireless Camera Localization

CamLopa localizes snooping cameras in two stages: (i) azimuth localization and (ii) quadrant determination.

5.1. Diffraction Attenuation in Wireless Signal Propagation

Diffraction allows radio signals to propagate around the curved surface of the earth, beyond the horizon, and behind obstacles [40]. This phenomenon can be explained

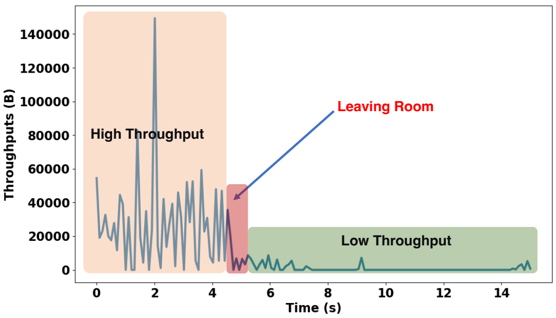


Figure 3: Throughput during the user's exit from the room.

using Huygen's principle, which states that all points on a wavefront can be considered as point sources generating secondary wavelets. These secondary wavelets are combined in the direction of propagation to form a new wavefront. Diffraction occurs due to the propagation of these secondary wavelets into shadowed regions. Empirical studies [41], [43], [60] suggest that when an obstacle is within the FFZ, it primarily causes the diffraction of wireless signals. Conversely, when the obstacle is outside the FFZ, it mainly causes the reflection of signals.

In Figure 4, assuming the height of a point Q from the LOS path is h, and its projection onto the LOS path has distances d_1 and d_2 from Tx and Rx, respectively, the path difference between the signal propagating through this point and the LOS path Δd can be expressed as [40]:

$$\Delta d \approx \frac{h^2}{2}\frac{d_1+d_2}{d_1 d_2}. \tag{7}$$

The corresponding phase difference is:

$$\phi = \frac{2\pi d}{\lambda} = \frac{\pi h^2}{\lambda}\frac{d_1+d_2}{d_1 d_2}. \tag{8}$$

Equation 8 can typically be expressed using the Fresnel-Kirchoff diffraction parameter v as follows:

$$\phi = \frac{\pi}{2}v^2. \tag{9}$$

The Fresnel-Kirchoff diffraction parameter v can be represented as:

$$v = h\sqrt{\frac{2(d_1+d_2)}{\lambda d_1 d_2}}. \tag{10}$$

The Fresnel-Kirchoff diffraction parameter originates from the combination of the Fresnel approximation and Kirchhoff's diffraction theory. This parameter is used to describe the diffraction effect that occurs when a wave encounters an obstacle or aperture. The magnitude of v is related to the significance of the diffraction effect. A smaller v indicates a smaller obstacle size or greater distance, resulting in a less significant diffraction effect. Conversely, a larger v indicates a more pronounced diffraction effect, where the wave experiences noticeable diffraction when encountering an obstacle and continues to propagate around it. The radius

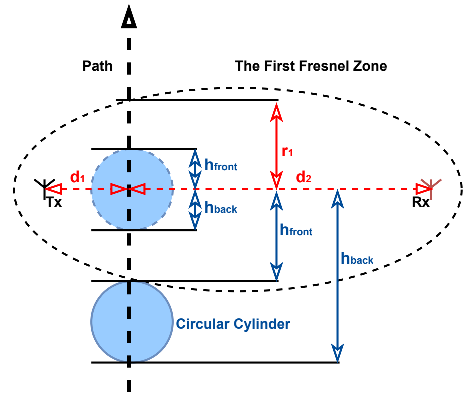


Figure 4: A moving cylinder across the FFZ.

(The perpendicular distance from Q to the LOS path.) of the FFZ can be expressed as [40]:

$$r_1 = \sqrt{\frac{\lambda d_1 d_2}{d_1+d_2}}. \tag{11}$$

Thus, the Fresnel-Kirchoff diffraction parameter can be represented as:

$$v = h\sqrt{\frac{2(d_1+d_2)}{\lambda d_1 d_2}} = h\frac{\sqrt{2}}{r_1}. \tag{12}$$

In wireless communication systems, only a portion of the signal's energy can diffract around an obstacle, allowing only part of the blocked energy to reach the receiver. Therefore, when an obstacle obstructs part of the Fresnel zone, the received energy is the vector sum of the contributions from all the unobstructed portions of the Fresnel zone. If an infinitely long object is positioned at a distance h from the LOS path, the ratio of the electric field strength E_d affected by diffraction to the unobstructed electric field strength E_o is given by [40]:

$$\frac{E_d}{E_o} = F(v) = \frac{1+j}{2}\int_v^{\infty} exp(\frac{-j\pi t^2}{2})dt, \tag{13}$$

where $F(v)$ is the complex Fresnel integral.

In practical scenarios, a human body can be approximated as a cylinder to analyze the signal attenuation caused by diffraction along the propagation path. As shown in Figure 4, both ends of the cylinder induce diffraction effects, where h_{front} and h_{back} represent the distances from the front and back edges of the cylinder to the LOS path, respectively. The signal attenuation caused by diffraction at the front and back edges can be expressed as:

$$F(v_{front}) = \frac{1+j}{2}\int_{v_{front}}^{\infty} exp(\frac{-j\pi t^2}{2})dt, \tag{14}$$

$$F(v_{back}) = \frac{1+j}{2}\int_{-\infty}^{v_{back}} exp(\frac{-j\pi t^2}{2})dt. \tag{15}$$

The diffraction gain due to the presence of a cylinder is given by:

$$G_d(dB) = 20log|F(v_{front}) + F(v_{back})|. \tag{16}$$

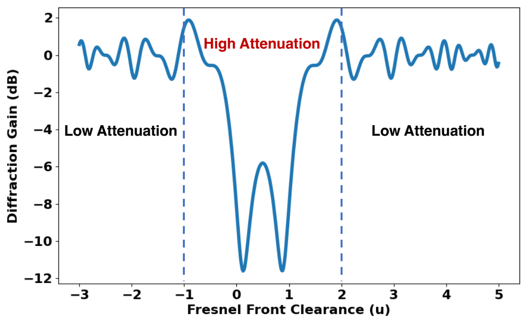


Figure 5: Diffraction gain variation corresponding to Figure 4.

To intuitively demonstrate the diffraction attenuation caused by obstruction, we use the example of a cylinder with a radius equal to the FFZ radius. To simplify the setup, we assume the cylinder crosses the FFZ vertically (as shown in Figure 4) and introduce Fresnel clearance u [60] to indicate the percentage of crossing:

$$u = \frac{h}{r_1}, \tag{17}$$

$$v = h\sqrt{\frac{2(d_1 + d_2)}{\lambda d_1 d_2}} = h\frac{\sqrt{2}}{r_1} = \sqrt{2}u. \tag{18}$$

The diffraction gain during the cylinder's traversal of the FFZ is shown in Figure 5. It is obvious that the cylinder causes significant signal attenuation due to diffraction from the moment it touches the FFZ ($u_{front} = -1$) until it completely exits the FFZ ($u_{front} = 2$).

5.2. Azimuth Localization

Section 5.1 highlights that the period of significant wireless signal attenuation can be used to determine the time taken for an obstacle (the user) to cross the first Fresnel zone (FFZ). Below, we list several key points:

- The location of the CamLoPA device is known.
- As discussed in Section 2, CSI can represent the attenuation of WiFi signals.
- When the positions of transmitter (camera) and receiver (CamLoPA) are fixed, and the obstacle (user) walks in a straight line past the receiver and through the FFZ, the length of the path traversing the FFZ is related to the angle between the walking path and LOS (azimuth).

Based on the above key points, it is evident that if the user's walking speed and the distance between the transmitter and receiver are known, the azimuth angle of the wireless camera can be calculated using the time of significant CSI attenuation. Furthermore, an important corollary is derived:

> **Corollary:** In an indoor environment, for a camera to effectively monitor an area of interest, its LOS must remain unobstructed. Therefore, if the azimuth of the wireless camera is known, the camera is likely located at the first obstacle encountered along that angle.

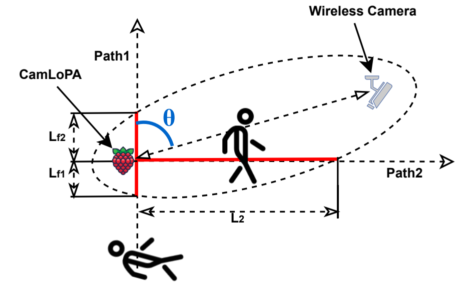


Figure 6: The illustration of azimuth localization.

From the corollary, we know that in an indoor environment, effective localization of a wireless camera can be achieved by knowing the azimuth angle information, even without distance information. However, some challenges arise in practice:

- Users' walking speeds are difficult to obtain.
- Some users may be unaware of their own sizes.
- The distance between the CamLoPA device and the wireless camera is unknown.

CamLoPA introduces the ***orthogonal ratio*** to address the challenge of obtaining crucial parameters (e.g., speed and distance). As shown in Figure 6, CamLoPA prompts the user to walk along two orthogonal paths, both of which pass by the CamLoPA device. In real-world environments, finding such paths is usually feasible. CamLoPA then calculates the time it takes to traverse the FFZ along each path (represented by the red lines) based on the periods of significant CSI attenuation and computes their ratio. The azimuth angle θ (the angle of the Path 1 relative to the LOS path) is estimated using a model that relates this ratio to the azimuth. The orthogonal ratio-based method eliminates the impact of walking speed and reduces errors due to unknown distances between devices and the user's size.

Next, we provide a detailed explanation of the azimuth localization model based on the orthogonal ratio. As explained in Section 5.1, the duration of significant CSI attenuation corresponds to the time it takes for the user to traverse from entering to exiting the FFZ. Therefore, for Path 1, the walking distance that causes significant attenuation can be calculated as follows:

$$L_1 = B_s + L_f, \tag{19}$$

where B_s and L_f represent the user's body size and the length of Path 1 within the FFZ (red line in Figure 6). L_f can be further divided into L_{f1}, the distance from the FFZ boundary to CamLoPA, and L_{f2}, the distance from CamLoPA to the FFZ boundary. Combined with Equation 5, we have the following equations:

$$L_{f1} + \sqrt{d^2 + L_{f1}^2 - 2dL_{f1}\cos\theta} - d = \frac{\lambda}{2}, \tag{20}$$

$$L_{f2} + \sqrt{d^2 + L_{f2}^2 - 2dL_{f1}\cos(\pi - \theta)} - d = \frac{\lambda}{2}, \tag{21}$$

where d is the distance between T_x and R_x. Treating L_{f1} and L_{f2} as unknown, they can be solved as follows:

$$L_{f1} = \frac{\lambda^2 + 4d\lambda}{4(2d + \lambda - 2d\cos\theta)}, \tag{22}$$

$$L_{f2} = \frac{\lambda^2 + 4d\lambda}{4(2d + \lambda + 2d\cos\theta)}. \tag{23}$$

Path 2 does not cross the entire FFZ, and thus the length of its path that perturbs the CSI is only the distance from CamLopa to the FFZ boundary:

$$L_2 + \sqrt{d^2 + L_2^2 - 2dL_2\cos(\frac{\pi}{2} - \theta)} = \frac{\lambda}{2}. \tag{24}$$

Treating L_2 as unknown, it can be solved as follows:

$$L_2 = \frac{\lambda^2 + 4d\lambda}{4(2d + \lambda - 2d\sin\theta)}. \tag{25}$$

The orthogonal ratio is calculated as:

$$\begin{aligned} R_o &= \frac{T_1}{T_2} = \frac{T_1 v_s}{T_2 v_s} = \frac{L_1}{L_2} = \frac{4B_s(2d + \lambda - 2d\sin\theta)}{\lambda^2 + 4d\lambda} \\ &+ \frac{4(2d + \lambda - 2d\sin\theta)}{4(2d + \lambda - 2d\cos\theta)} + \frac{4(2d + \lambda - 2d\sin\theta)}{4(2d + \lambda - 2d\cos\theta)} \\ &= \frac{4B_s(2d + \lambda - 2d\sin\theta)}{\lambda^2 + 4d\lambda} + \frac{8(2d + \lambda)(2d + \lambda - 2d\sin\theta)}{(2d + \lambda)^2 - (2d\cos\theta)^2}, \end{aligned} \tag{26}$$

where T_1 and T_2 are the periods during which the user's movement along Paths 1 and 2 causes significant CSI attenuation, and v_s is the user's walking speed. By taking the ratio, the influence of the speed can be eliminated. After obtaining R_o, the Newton-Raphson method can be used to solve for θ.

Next, we analyze the errors introduced by setting fixed values of B_s and d. We conducted an analysis of the L_1-θ and R_o-θ relationship models separately. Figure 7 shows the variations of L_1 and R_o relative to the azimuth angle θ for $B_s = 0.15$, 0.25, and 0.45, which are reasonable based on common sense. It can be observed that the error caused by B_s is more pronounced near $\theta = 90^o$. The error in the L_1-based method due to changes in B_s is significant, while the R_o-based method effectively mitigates the error caused by the variations of B_s. Figure 8 illustrates the variations of L_1 and R_o relative to the azimuth angle θ for $d = 1$, 3, and 6, which are plausible ranges for indoor wireless camera deployment. It can be observed that the error caused by d is more significant around 0/180^o. Compared to the L_1-based approach (with an theoretical maximum error approaching 20^o), the theoretical maximum error of R_o (15^o) is more advantageous. Furthermore, the variations in the walking speed due to different users' habits can introduce greater errors in the L_1-based scheme. It is clear that the orthogonal ratio-based scheme employed by CamLopa nearly eliminates the bias caused by unknown speeds and user body sizes while minimizing the errors due to the unknown distance between the transmitter and receiver. Even under the condition of maximum theoretical error, the localization results remain highly practical in real indoor environments due to the limited number of potential hiding spots for wireless cameras. Due to the superiority of the orthogonal ratio strategy, in this paper, CamLopa sets $d = 3$ and $B_s = 0.25$ as fixed values according to realistic scenarios, and users walk for 10 seconds along each path.

5.3. Quadrant Determination

From Figures 7 and 8 (i.e., R_o leading to two possible values of θ), we can also observe that the predicted θ using R_o has two possible values, making it impossible to determine whether the camera is in the first or second quadrant. Therefore, further quadrant determination is necessary.

To achieve quadrant determination, CamLopa prompts the user to walk again in the same direction as Path 1 for 10 seconds, but starting from a position in front of the CamLopa device. The quadrant can then be determined based on changes in the CSI. The rationale is that if the wireless camera is located in the first quadrant, the user standing at the starting position will block the LOS signal between the two devices, causing significant signal variations due to the diffraction effect when the user moves. Conversely, if the wireless camera is behind the user, the user's movement will only cause signal fluctuations due to reflection. Specifically, CamLopa determines the quadrant as follows:

$$\mathbf{Q_{mac}} = \begin{cases} \mathbf{2} & \text{if } \frac{\max(CSI_3)}{\min(CSI_3)} < T_q * \frac{\max(CSI_1)}{\min(CSI_1)}, \\ \mathbf{1} & \text{else }. \end{cases} \tag{27}$$

Equation 27 means that if the extent of the CSI fluctuation caused by Path 3 is less than T_q times the extent of the CSI fluctuation caused by Path 1, the camera is determined to be in the second quadrant; otherwise, it is in the first quadrant.

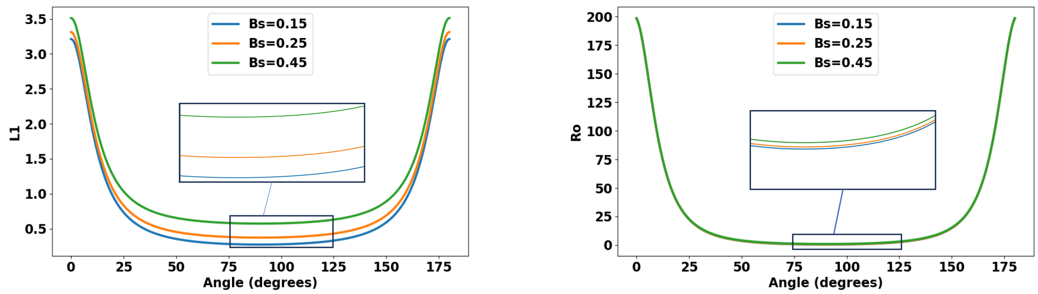
Since movement within the range of 180-360^o does not cross the LOS, CamLopa can only locate devices within the range of 0-180^o. However, in real-world environments, the user's available space is usually near walls, thus a single measurement by CamLopa remains highly useful. If the condition of moving near walls is not met, CamLopa requires two measurements.

6. Implementation and Evaluation

We implemented CamLopa in multiple rooms and diverse hidden wireless cameras, and this section presents the implementation details of CamLopa.

6.1. Prototype

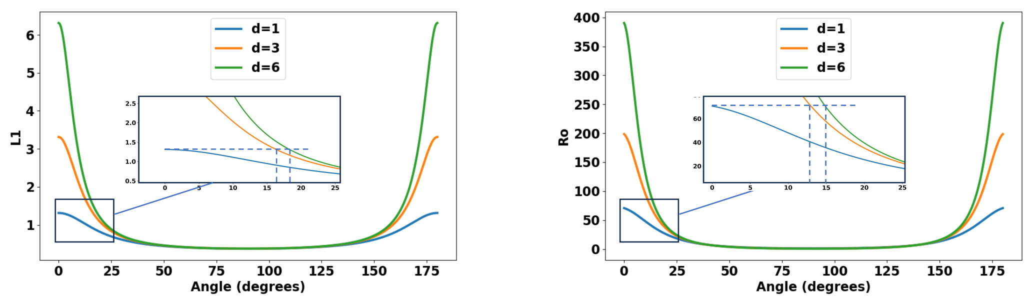
The prototype of CamLopa is shown in Figure 9. The Raspberry Pi uses its built-in wireless NIC with the nexmon tool [46] to modify the kernel for CSI extraction. However, the modified driver for extract CSI cannot sniff 802.11 packets, therefore we set up an external network card (NIC1) with monitoring capabilities to sniff 802.11 packets. NIC2 is a standard wireless network card used for communication between the CamLopa device and the user's smartphone. The user's smartphone can receive prompts and localization



(a) The variations of L_1.

(b) The variations of R_o.

Figure 7: The variations of L_1 and R_o relative to θ with B_s changes.



(a) The variations of L_1.

(b) The variations of R_o.

Figure 8: The variations of L_1 and R_o relative to θ with d changes.

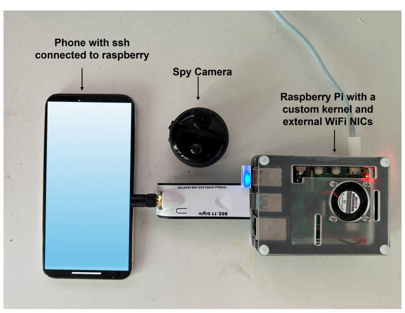


Figure 9: The prototype of CamLopa.

results from CamLopa via SSH tools. More details please refer to Appendix E.

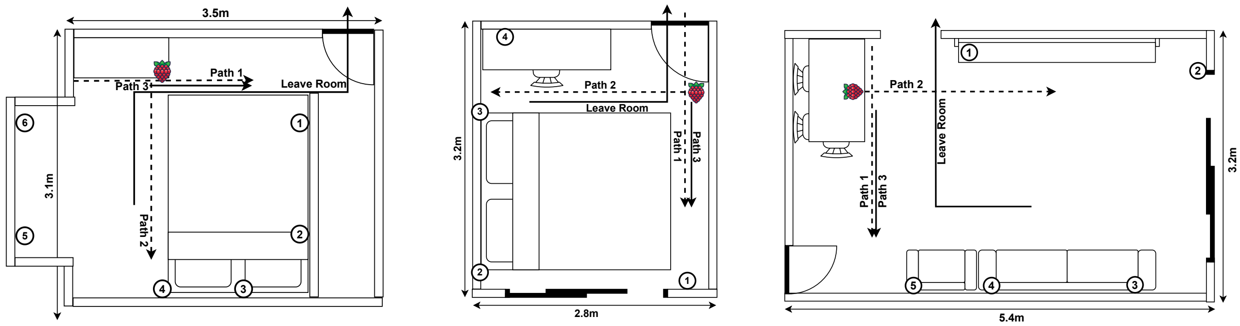
6.2. Experimental Setup

We evaluated the performance of CamLopa using seven different wireless cameras (details provided in Appendix D). All devices were purchased from online shopping platforms, and the cameras were connected to a 2.4GHz WiFi network. The experiments were conducted in a real residential setting, spanning three different rooms, each containing various obstacles such as furniture and household items. The experimental environment included numerous WiFi devices and APs operating both within and around the test house. Since the experiments were conducted in actual home environments over an extended period, only the residents participated to ensure privacy. The validation experiments were carried out over a total duration of two months.

The layout of three rooms are shown in Figure 10, and the location of cameras please refer to Appdedix D. Rooms 1 and 2 (Figures 10a and 10b) are bedrooms, while room 3 is a living room (Figure 10c). In real environments, private spaces like bedrooms and hotel rooms have limited activity space, restricting the feasibility of previous methods that rely on extensive indoor scanning. As shown in Figure 13, the cameras we used have an average QoS data packet length ranging from 369 to 1050 bytes during video stream uploads, with upload speeds ranging from 35 to 130 packets per second. Therefore, in our experiments, T_s and T_l are set to 300 bytes and 150 packets (30 packets * 5 seconds), respectively. The T_q for quadrant localization is empirically set to 0.6.

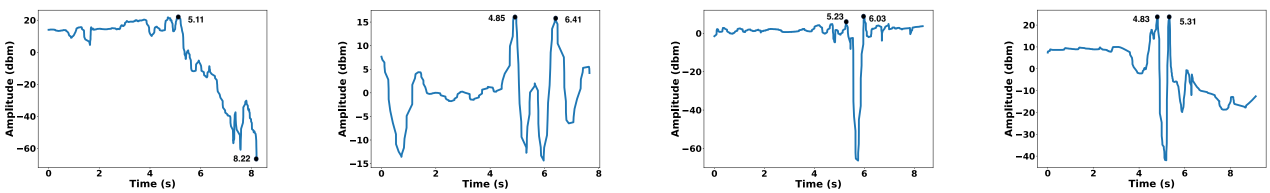
6.3. CSI Analysis and Algorithm Implementation

In this section, we analyze the relationship between the CSI influenced by user activity and the azimuth of the camera. Furthermore, we elaborate on the design of the algorithm for extracting attenuation time from the CSI. The variation in CSI amplitude during localization for a camera at different azimuth angles are shown in Figure 11. It can be observed that the CSI amplitude variation is significantly influenced by the azimuth angle of the wireless camera relative to CamLopa. Generally, the larger the angle, the shorter the duration of significant fluctuations in CSI from Path 1 (CSI 1), while the duration of significant fluctuations

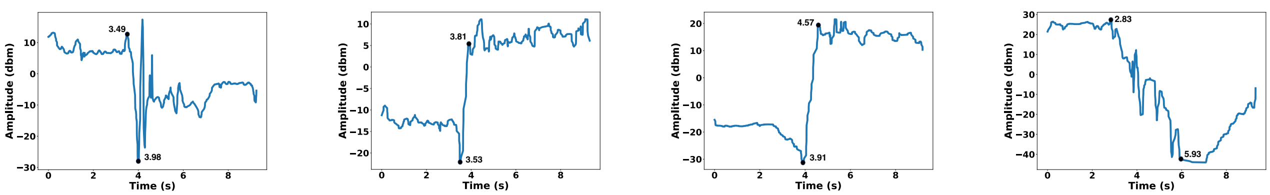


(a) Room1 (b) Room2 (c) Room3

Figure 10: The layout of three rooms.



(a) 4.11°, path1. (b) 28.61°, path1. (c) 60.28°, path1. (d) 88.54°, path1.



(e) 4.11°, path2. (f) 28.61°, path2. (g) 60.28°, path2. (h) 88.54°, path2.

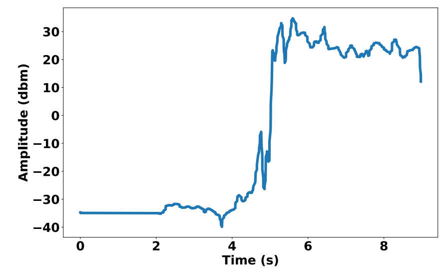
Figure 11: The CSI amplitude during localization. The black dots represent the start and end points of significant CSI fluctuations for each path. By dividing the duration of significant attenuation of path 1 by that of path 2, we obtain R_o, which is then used to calculate θ according to Equation 26. In (c) and (g), R_o is calculated as $\frac{0.8}{0.66} = 1.21$, and substituting this into Equation 26 yields $\theta = 72.18°$. The calculations for the others follow the same procedure.

in CSI from Path 2 (CSI 2) increases. These experimental results validate the feasibility of the azimuth localization scheme proposed by CamLoPa. Additionally, here are some practical consideration:

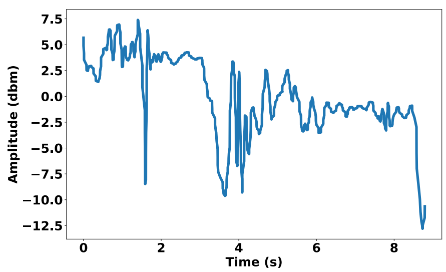
- The fluctuation duration of CSI 2 may not accurately reflect the actual path length causing the fluctuation, as it takes time for the user to accelerate from a stationary state to walking.
- When the angle is too small (0 degrees) or too large (90 degrees), the calculated R_o significantly deviates from the theoretical R_o. This is due to the limited indoor space usually causes the user to stop after a short distance due to obstacles.

To obtain the duration of significant CSI fluctuations, we use different methods for CSI 1 and CSI 2. For CSI 1, we first identify the lowest point and then use the calculated inverse to find the start and end points of the fluctuation. For CSI 2, we first calculate the mean values of the initial and later segments, then we construct a piecewise waveform where the values of the initial and later segments are equal to the calculated means. By adjusting the position of the segmentation, we find the point that best matches the waveform with CSI 2 to determine the midpoint of the fluctuation. We then calculate the inverse to identify the start and end points of the fluctuation. Additionally, based on our first observation, we scale the calculated fluctuation duration for CSI 2 to eliminate errors. For activities that cause fluctuations exceeding a certain duration, we increase the fluctuation time to mitigate the effect noted in the second observation. As shown in Fig 11, CamPoLA achieves localization of cameras depolyed at different positions.

Figure 12 shows the variations in CSI 3 (corresponding to Path 3) when the wireless camera is located in different quadrants. It is obvious that the quadrant localization scheme proposed by CamLoPa is also effective. Since CSI consists of many different subcarriers, and different subcarriers have varying sensitivities to user activity (with higher amplitudes indicating lower sensitivity), CamLoPa focuses only on the periods of significant attenuation. Therefore, we select the five subcarriers with the highest amplitudes, average them after filtering, and use this average as the final input for CamLoPa to calculate R_o and the quadrant.

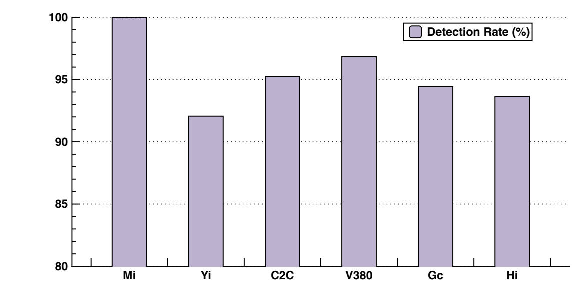


(a) 28.61°, path3.



(b) 130.1°, path3.

Figure 12: The CSI amplitude during quadrant determination. When the camera is located in the first quadrant (a), the user's starting position blocks the LOS, resulting in significant fluctuations during movement. In contrast, when the camera is located in the second quadrant (b), the user does not block the LOS, leading to minor fluctuations.



Device	Mi	Yi	C2C	V380	Gc	Hi
Packet Length	1050	761	632	369	640	828
Packets/s	51	67	63	35	112	130

Figure 13: Snooping camera detection performance.

6.4. Performance of Wireless Camera Detection

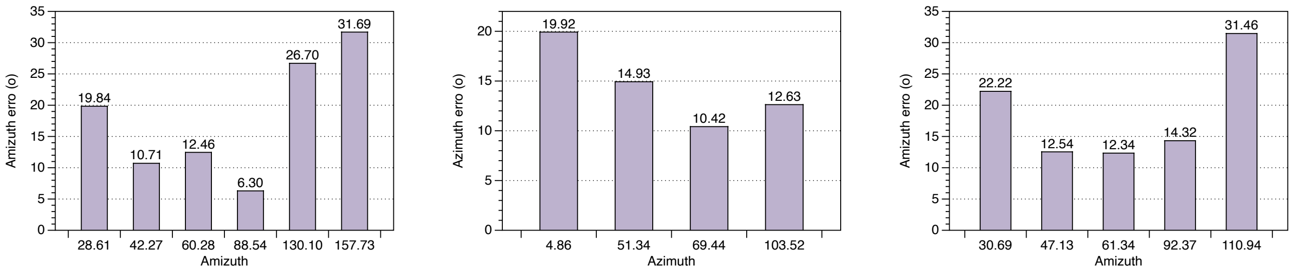
CAMLOPA detects wireless cameras monitoring the current area by first identifying suspicious devices, prompting the user to leave the room, and monitoring throughput changes to detect snooping hidden wireless cameras. CAMLOPA achieves an 84.35% success rate in identifying suspicious wireless cameras across all devices. The probability of identifying the 360 camera as a suspicious device is 0, while the accuracy of detecting other wireless cameras as suspicious devices reaches 98.41%. This discrepancy occurs because, during traffic sniffing, the 360 wireless camera only allows the capture of ACK Block and Request-to-Send packets, but not QoS data packets. This limitation may be due to the special data transmission methods or protocols they use, which prevent its traffic from being intercepted, thus hindering detection and previous methods based on WiFi traffic all cannot work [12], [13], [14], [15]. However, the nexmon tool used by CAMLOPA can still capture the CSI for the 360 camera from WiFi traffic. The snooping camera detection results are shown in Figure 13. CAMLOPA achieves a 95.37% success rate in detecting snooping cameras for six types of cameras across three rooms, except for the 360 wireless camera. For devices similar to the 360 camera, we believe that wireless camera detection can still be achieved by querying the OUI of the captured Request-to-Send packet's leaked MAC address. By constructing an OUI table of all available devices using device name information from shopping platforms and MAC address lookup websites, it is possible to identify the device type. However, CAMLOPA cannot determine whether the camera is monitoring the current area using this method.

6.5. Performance of Wireless Camera Localization

Overall Performance: The localization results across three rooms are shown in Figure 14, where CAMLOPA achieves an average azimuth localization error of 17.23 degrees for wireless hidden cameras. CAMLOPA demonstrates higher localization accuracy for cameras placed within the 40-90° range, while accuracy decreases for cameras located in the second quadrant or near 0°. This discrepancy is attributed to errors introduced by the quadrant determination scheme and path length limitations. The primary source of quadrant determination error is the human torso, which is relatively large and can introduce significant noise into the reflected signals. Such errors in quadrant localization can lead to azimuth errors of up to 180°. To mitigate this, searching the opposite location can help identify the correct position. For cameras near 90°, the algorithm described in Section 6.3 tends to output predictions close to 90°, resulting in lower localization errors. Overall, CAMLOPA achieves high accuracy with low user efforts, minimal space requirements and no need for training.

Robustness: As shown in Figuree 15, CAMLOPA maintains consistent localization performance across different camera types, demonstrating its robustness to device variations. The azimuth localization errors for CAMLOPA across three rooms were 17.95°, 14.48°, and 18.58°, respectively, further emphasizing its resilience to environmental changes. This robustness is a result of CAMLOPA's localization algorithm, which is a model-based method. Learning-based methods used in previous approaches [16] require extensive training data to ensure robustness.

Influence of T_q: We also conducted ablation experiments in Rooms 1 and 2 to determine the optimal value for the threshold T_q. Using classification accuracy as the evaluation metric, the results (accuracy: thresholds) were: (0.1: 0.5, 0.3: 0.6, 0.5: 0.8, 0.6: 0.85, 0.7: 0.8, 0.9: 0.6). The results were consistent across both rooms, leading to the selection of $T_q = 0.6$ as the optimal threshold.



(a) Room1. (b) Room2. (c) Room3.

Figure 14: Localization results of hidden cameras deployed at different positions.

6.6. Comparative Study

Performance Comparison: Most previous localization methods [12], [13], [15] typically evaluate in nearly empty rooms and use distance as the evaluation metric, making direct comparisons with our approach challenging. Additionally, many of these studies have not been open-sourced. Therefore, we compare CAMLOPA with the SOTA method LocCams [16]. LocCams collects CSI while the user holds the device in four different orientations. It then uses a pre-trained deep learning model to identify which orientations have their LOS paths blocked, with the mid-direction of the blocked LOS paths considered the device's azimuth. We conducted experiments in Room 2 using two cameras (360 and Gc) across four different locations. The results, presented in Table 2, include in-domain (ID), cross-device (CD), and cross-device-room (CDR) comparisons. The findings clearly demonstrate that CAMLOPA outperforms LocCams, showing better overall accuracy and robustness.

Cost, Time, and User Effort Comparison: The total cost of our system is $82.71 (Raspberry Pi: $79.20 + USB network adapter: $3.51). In comparison, LocCams uses a Nexus 5, priced at $99.99 on Amazon. Other traffic-based systems such as SNOOPDOG [13], Lumos [12], and ScamF [15] also use Raspberry Pi, while MotionCompass [14] uses an Android device (note that only certain smartphones allow root access for collecting CSI or traffic, meaning smartphone-based platforms often incur additional hardware costs). RF/infrared-based solutions, such as HeatDeCam [11] and LAPD [10], require more expensive equipment (over $300). In terms of time, LocCams is the fastest, taking only 0.5 minutes for localization. CAMLOPA requires 1.5-2 minutes, but this additional time significantly improves both accuracy and robustness. MotionCompass, based on traffic patterns, takes around 3 minutes. Other RSSI/traffic-based systems typically takes 15-30 minutes [12], [13], [15]. For user efforts, MotionCompass require the user to walk several straight paths that span both monitored and unmonitored areas, which can be difficult to achieve in real-world environments. Other RSSI/traffic-based systems require users to walk around the perimeter of the room multiple times or constantly adjust a laptop's position to cover most areas, which is also impractical. LocCams requires the least user effort, as users only need to perform a few turns. CAMLOPA, requiring users to walk three orthogonal paths, has the second-lowest effort requirement, while offering significant improvements in performance. Moreover, such paths are easy to find in everyday environments, such as hotels.

TABLE 2: Comparison with other methods.

Method	CAMLOPA	LocCam ID	LocCam CD	LocCam CDR
360	**17.60**	25.10	30.22	40.32
Gc	**15.13**	27.55	38.90	43.39

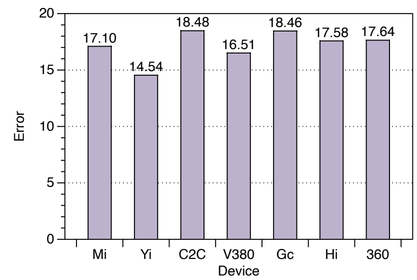


Figure 15: Localization results across different device.

7. Disscussion

In this section, we discuss the limitations of CAMLOPA, the potential risks, and possible improvements.

Non-WiFi Cameras. The fundamental principle behind CAMLOPA's detection and localization of wireless cameras limits its applicability to live streaming spy cameras on WiFi networks. It does not extend to cameras that use local storage, cellular networks, or Ethernet. However, most recent crime cases have involved WiFi spy cameras [15] because they are easy to deploy and manage, and their prevalence is rapidly increasing in the commercial market. Therefore, CAMLOPA is suitable for many scenarios. To expand the detection range, infrared or optical methods [10], [11] would still be needed.

MAC Address Randomization. Although some devices employ MAC address randomization [61] to enhance security, this does not affect CAMLOPA's detection and localization capabilities. This is because devices, even with MAC address randomization, use a consistent MAC address for communication once a network connection is established.

Non-VBR Devices. When CAMLOPA detects whether a camera is monitoring the current area, the device's traffic must be encoded using a Variable Bit Rate (VBR) algorithm. While this algorithm is used by the vast majority of wireless camera devices, if a camera is specifically designed to encode video/audio information at a constant bit rate (CBR), CamLoPA may only be able to roughly detect its presence using the OUI table. However, CamLoPA can still locate such devices through the proposed localization scheme.

False Positives and Misdiscard. To evaluate the false positive rate of detection, we simulated potential activities that could trigger false alarms in Room 1 by setting up a computer uploading files and having another computer and smartphone engaged in video conferencing. Only 6.67% of the samples resulted in false positives. Furthermore, devices that generate significant traffic like camera indoors are typically under user control, which makes it unlikely for them to cause interference. Even if devices in neighboring rooms trigger false alarms, they would primarily increase the workload rather than posing a security risk. Our approach filters out routers with weak RSSI values. While the position of the wireless camera may differ from the CamLoPA device, leading to potentially different RSSI values, this could result in misdiscarding some devices. To mitigate this, we implemented a margin of tolerance by slightly lowering the RSSI threshold (by 5 dBm) below the level required for reliable streaming quality to prevent incorrectly exclusion.

Evading CamLoPA. We acknowledge that more powerful attackers may have ways to evade CamLoPA. Attackers could modify the behavior of hidden cameras by customizing hardware or altering firmware to change the packet size or arrival intervals, thus avoiding detection. These methods could prevent CamLoPA from detecting them. However, such tactics require a high level of expertise from the attacker. The localization module, based on wireless signal propagation path analysis, can still function normally by using the device's MAC address and WiFi channel. Avoiding localization would require modifying the network card hardware to control the WiFi signal's transmission power, causing it to constantly change and disrupt the signal attenuation trend caused by user activity. This also requires attackers to have specialized knowledge, and modifying network card hardware is considerably challenging. According to the latest research [62], **the majority of surveillance tools still rely on commercially available devices**, thus we have not consider adaptive attack in our evaluation.

Limitations and Fault Tolerance. CamLoPA can only localize wireless cameras within the 0-180^o range. However, in real-world environments, it is relatively easy to find a location near a wall to place the CamLoPA device, and it can perform two rounds of positioning to achieve 360^o localization. Another limitation is that CamLoPA assumes users walk along two orthogonal straight paths at a constant speed, which may introduce faults in real-world scenarios. However, in actual environments, the layout of indoor furniture (such as floor stripes, walls, and furniture) can help guide users to maintain two straight walking paths. Additionally, users can easily control their walking speed within a certain range to minimize the biases. Our experiments were conducted in real-world environments, without any special measures to assist the users in walking in a straight line and control speed. The results demonstrate the robustness of our approach to these liminations. For fault tolerance, although CamLoPA's localization results are not perfectly precise in confined indoor spaces, it significantly reduce the search area and reduce user efforts for the user compare to previous studies.

TABLE 3: Evaluation with Challenging Environments.

Materials	Normal	Plastic	Textile	Metal
360	17.60	16.51	16.06	22.42
Gc	15.13	17.62	14.79	39.79

Multiple Cameras. While we evaluated CamLoPA in single-camera scenarios, it can easily be extended to situations involving multiple cameras. During the camera detection phase, a single user walking can detect multiple cameras by clustering the MAC addresses of all captured packets. However, when capturing CSI, the Nexmon tool can only obtain packets from one MAC address at a time. As a result, to localize multiple cameras, the user must repeat the localization process for each individual camera.

Challenging Environments. In real-world settings, attackers may attempt to disguise hidden cameras using various objects. To assess the performance of CamLoPA under such conditions, we evaluated its effectiveness when cameras were obscured by different materials. The results, presented in Table 3, show that common materials like plastic and textiles had minimal impact on CamLoPA's performance. However, metal caused a significant degradation in performance. This is because metal absorbs wireless signals, which not only impairs CamLoPA's localization capabilities but also degrades overall network quality. As a result, attackers are unlikely to use metal to conceal WiFi cameras.

Future Work for Improvement. Next, we aim to further reduce user effort and eliminate localization errors caused by user activity. This will involve using low cost 3D-printed kits with metal obstructions as peripherals. By controlling the metal obstructions to rotate around the Raspberry Pi, we can perturb the CSI. Constructing a corresponding CSI-azimuth model will enable more precise localization with no user effort. We plan to explore building indoor wireless device maps based on our localization technology. Combine this map with WiFi traffic and CSI will help us study new smart home related risks and develop defensive measures.

8. Conclusion

In this paper, we propose CamLoPA, a framework for detecting and locating wireless hidden cameras based on wireless signal propagation path analysis, specifically focusing on diffraction attenuation. CamLoPA establishes a relationship between the signal attenuation caused by user activity and the location of the wireless camera. We evaluate the performance of CamLoPA through comprehensive experiments in real-world conditions. Compared to current methods, CamLoPA offers several advantages: it is cost-effective, requires no training, demands less activity space, and involves minimal user effort. However, CamLoPA still has some limitations. In future work, we aim to further reduce user effort and minimize localization errors through the use of low-cost peripherals.

References

[1] Ankit Gupta. Wireless monitoring and surveillance market, by component, type, connectivity, end-user - forecast till 2030. https://www.marketresearchfuture.com/reports/wireless-monitoring-surveillance-market-975, 2024.

[2] Yun Ye, Song Ci, Aggelos K Katsaggelos, Yanwei Liu, and Yi Qian. Wireless video surveillance: A survey. *IEEE Access*, 1:646–660, 2013.

[3] Shrirang Mare, Franziska Roesner, and Tadayoshi Kohno. Smart devices in airbnbs: Considering privacy and security for both guests and hosts. *Proceedings on Privacy Enhancing Technologies*, 2020.

[4] David Janssen. Many airbnbs have cameras installed, especially in the us, canada and singapore. , 2023.

[5] Jim Dalrymple. More than 1 in 10 airbnb guests have found hidden cameras: Survey. https://www.inman.com/2019/06/07/morethan-1-in-10-airbnb-guest-have-found-cameras-in-rentals-survey/, 2019.

[6] The security camera laws in delaware. https://www.cambasket.com/the-security-camera-laws-in-delaware/.

[7] Dinesh Sathyamoorthy, Mohd Jalis Md Jelas, and Shalini Shafii. Wireless spy devices: A review of technologies and detection methods. *Editorial Board*, 7:130, 2014.

[8] Veronica Valeros and Sebastian Garcia. Spy vs. spy: A modern study of microphone bugs operation and detection. *Chaos Computer Club eV*, 2017.

[9] Tian Liu, Ziyu Liu, Jun Huang, Rui Tan, and Zhen Tan. Detecting wireless spy cameras via stimulating and probing. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 243–255, 2018.

[10] Sriram Sami, Sean Rui Xiang Tan, Bangjie Sun, and Jun Han. Lapd: Hidden spy camera detection using smartphone time-of-flight sensors. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 288–301, 2021.

[11] Zhiyuan Yu, Zhuohang Li, Yuanhaur Chang, Skylar Fong, Jian Liu, and Ning Zhang. Heatdecam: detecting hidden spy cameras via thermal emissions. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3107–3120, 2022.

[12] Rahul Anand Sharma, Elahe Soltanaghaei, Anthony Rowe, and Vyas Sekar. Lumos: Identifying and localizing diverse hidden {IoT} devices in an unfamiliar environment. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1095–1112, 2022.

[13] Akash Deep Singh, Luis Garcia, Joseph Noor, and Mani Srivastava. I always feel like somebody's sensing me! a framework to detect, identify, and localize clandestine wireless sensors. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1829–1846, 2021.

[14] Yan He, Qiuye He, Song Fang, and Yao Liu. Motioncompass: pinpointing wireless camera via motion-activated traffic. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 215–227, 2021.

[15] Jeongyoon Heo, Sangwon Gil, Youngman Jung, Jinmok Kim, Donguk Kim, Woojin Park, Yongdae Kim, Kang G Shin, and Choong-Hoon Lee. Are there wireless hidden cameras spying on me? In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 714–726, 2022.

[16] Yangyang Gu, Jing Chen, Cong Wu, Kun He, Ziming Zhao, and Ruiying Du. Loccams: An efficient and robust approach for detecting and localizing hidden wireless cameras via commodity devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–24, 2024.

[17] Yushi Cheng, Xiaoyu Ji, Tianyang Lu, and Wenyuan Xu. Dewicam: Detecting hidden wireless cameras via smartphones. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 1–13, 2018.

[18] Kevin Wu and Brent Lagesse. Do you see what i see? detecting hidden streaming cameras through similarity of simultaneous observation. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*, pages 1–10. IEEE, 2019.

[19] Xiaoyu Ji, Yushi Cheng, Wenyuan Xu, and Xinyan Zhou. User presence inference via encrypted traffic of wireless camera in smart homes. *Security and Communication Networks*, 2018(1):3980371, 2018.

[20] Yushi Cheng, Xiaoyu Ji, Tianyang Lu, and Wenyuan Xu. On detecting hidden wireless cameras: A traffic pattern-based approach. *IEEE Transactions on Mobile Computing*, 19(4):907–921, 2019.

[21] Muhammad Salman, Nguyen Dao, Uichin Lee, and Youngtae Noh. Csi: Despy: enabling effortless spy camera detection via passive sensing of user activities and bitrate variations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–27, 2022.

[22] Dinhnguyen Dao, Muhammad Salman, and Youngtae Noh. Deepdespy: a deep learning-based wireless spy camera detection system. *IEEE Access*, 9:145486–145497, 2021.

[23] Jakobi Teknik. Spy hidden camera detector. https://apps.apple.com/us/app/spy-hidden-cameradetector/id925967783?mt=8, 2023.

[24] LLC LSC. Hidden camera detector. https://apps.apple.com/us/app/hidden-camera-detector/id532882360, 2023.

[25] Ziwei Liu, Feng Lin, Chao Wang, Yijie Shen, Zhongjie Ba, Li Lu, Wenyao Xu, and Kui Ren. Camradar: hidden camera detection leveraging amplitude-modulated sensor images embedded in electromagnetic emanations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):1–25, 2023.

[26] Agustin Zuniga, Naser Hossein Motlagh, Mohammad A Hoque, Sasu Tarkoma, Huber Flores, and Petteri Nurmi. See no evil: Discovering covert surveillance devices using thermal imaging. *IEEE Pervasive Computing*, 21(4):33–42, 2022.

[27] Yongqiang Ma, Xiangyang Luo, Ruixiang Li, Shaoyong Du, and Wenyan Liu. Lenser: A channel state information based indoor localization scheme for malicious devices. In *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pages 461–470. IEEE, 2023.

[28] Bevan B Baker and Edward Thomas Copson. *The mathematical theory of Huygens' principle*, volume 329. American Mathematical Soc., 2003.

[29] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[30] Chen Chen, Gang Zhou, and Youfang Lin. Cross-domain wifi sensing with channel state information: A survey. *ACM Computing Surveys*, 55(11):1–37, 2023.

[31] Jinyi Liu, Wenwei Li, Tao Gu, Ruiyang Gao, Bin Chen, Fusang Zhang, Dan Wu, and Daqing Zhang. Towards a dynamic fresnel zone model to wifi-based human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(2):1–24, 2023.

[32] Enze Yi, Dan Wu, Jie Xiong, Fusang Zhang, Kai Niu, Wenwei Li, and Daqing Zhang. {BFMSense}:{WiFi} sensing using beamforming feedback matrix. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 1697–1712, 2024.

[33] Xin Li, Hongbo Wang, Zhe Chen, Zhiping Jiang, and Jun Luo. Uwb-fi: Pushing wi-fi towards ultra-wideband for fine-granularity sensing. In *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, pages 42–55, 2024.

[34] Hongbo Wang, Jingyang Hu, Tianyue Zheng, Jingzhi Hu, Zhe Chen, Hongbo Jiang, Yuanjin Zheng, and Jun Luo. Muki-fi: Multi-person keystroke inference with bfi-enabled wi-fi sensing. *IEEE Transactions on Mobile Computing*, 2024.

[35] Daqing Zhang, Kai Niu, Jie Xiong, Fusang Zhang, and Xuanzhi Wang. Wifi/4g/5g based wireless sensing: Theories, applications and future directions. In *Integrated Sensing and Communications*, pages 387–417. Springer, 2023.

[36] Jinyang Huang, Bin Liu, Chenglin Miao, Xiang Zhang, Jiancun Liu, Lu Su, Zhi Liu, and Yu Gu. Phyfinatt: An undetectable attack framework against phy layer fingerprint-based wifi authentication. *IEEE Transactions on Mobile Computing*, 2023.

[37] Xiang Zhang, Yu Gu, Huan Yan, Yantong Wang, Mianxiong Dong, Kaoru Ota, Fuji Ren, and Yusheng Ji. Wital: A cots wifi devices based vital signs monitoring system using nlos sensing model. *IEEE Transactions on Human-Machine Systems*, 53(3):629–641, 2023.

[38] Xiang Zhang, Jingyang Huang, Huan Yan, Peng Zhao, Guohang Zhuang, Zhi Liu, and Bin Liu. Wiopen: A robust wi-fi-based open-set gesture recognition framework. *arXiv preprint arXiv:2402.00822*, 2024.

[39] Yu Gu, Xiang Zhang, Huan Yan, Jingyang Huang, Zhi Liu, Mianxiong Dong, and Fuji Ren. Wife: Wifi and vision based unobtrusive emotion recognition via gesture and facial expression. *IEEE Transactions on Affective Computing*, 2023.

[40] Theodore S Rappaport. *Wireless communications: principles and practice*. Cambridge University Press, 2024.

[41] Fusang Zhang, Kai Niu, Jie Xiong, Beihong Jin, Tao Gu, Yuhang Jiang, and Daqing Zhang. Towards a diffraction-based sensing approach on human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):1–25, 2019.

[42] Zhiyun Yao, Xuanzhi Wang, Kai Niu, Rong Zheng, Junzhe Wang, and Daqing Zhang. Wiprofile: Unlocking diffraction effects for sub-centimeter target profiling using commodity wifi devices. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pages 185–199, 2024.

[43] Xuanzhi Wang, Anlan Yu, Kai Niu, Weiyan Shi, Junzhe Wang, Zhiyun Yao, Rahul C Shah, Hong Lu, and Daqing Zhang. Understanding the diffraction model in static multipath-rich environments for wifi sensing system design. *IEEE Transactions on Mobile Computing*, 2024.

[44] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM computer communication review*, 41(1):53–53, 2011.

[45] Rui Li, Yu Duan, Rui Du, Fangxin Xu, Hangbin Zhao, Yang Sun, Yiyan Zhang, Daiyang Zhang, Yiming Liu, Zhiping Jiang, and Tony Xiao Han. Reshaping wifi isac with high-coherence hardware capabilities. *IEEE Communications Magazine*, 62(9):114–120, 2024.

[46] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. Free your csi: A channel state information extraction platform for modern wi-fi chipsets. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pages 21–28, 2019.

[47] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. Nexmon: The c-based firmware patching framework. https://nexmon.org, 2017.

[48] Yongsen Ma, Gang Zhou, and Shuangquan Wang. Wifi sensing with channel state information: A survey. *ACM Computing Surveys (CSUR)*, 52(3):1–36, 2019.

[49] Yu Gu, Xiang Zhang, Yantong Wang, Meng Wang, Huan Yan, Yusheng Ji, Zhi Liu, Jianhua Li, and Mianxiong Dong. Wigrunt: Wifi-enabled gesture recognition using dual-attention network. *IEEE Transactions on Human-Machine Systems*, 52(4):736–746, 2022.

[50] Christopher Wampler, Selcuk Uluagac, and Raheem Beyah. Information leakage in encrypted ip video traffic. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2015.

[51] Ben Nassi, Raz Ben-Netanel, Adi Shamir, and Yuval Elovici. Drones' cryptanalysis-smashing cryptography with a flicker. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1397–1414. IEEE, 2019.

[52] S. Fussell. Airbnb has a hidden-camera problem. https://www.theatlantic.com/technology/archive/2019/03/what-happens-when-youfind-cameras-your-airbnb/585007/, 2024.

[53] S. Jeong and J. Griffiths. Hundreds of south korean motel guests were secretly filmed and live-streamed online. https://www.cnn.com/2019/03/20/asia/southkorea-hotel-spy-cam-intl/index.html, 2019.

[54] Jorge Ortiz, Catherine Crawford, and Franck Le. Devicemien: network device behavior modeling for identifying unknown iot devices. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, pages 106–117, 2019.

[55] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759, 2018.

[56] Metageek. The basics: Understanding rssi. https://www.metageek.com/training/resources/understanding-rssi/, 2019.

[57] Jianfeng Li, Shuohan Wu, Hao Zhou, Xiapu Luo, Ting Wang, Yangyang Liu, and Xiaobo Ma. Packet-level open-world app fingerprinting on wireless traffic. In *The 2022 Network and Distributed System Security Symposium (NDSS'22)*, 2022.

[58] Geert Van der Auwera, Prasanth T David, and Martin Reisslein. Traffic characteristics of h. 264/avc variable bit rate video. *IEEE Communications Magazine*, 46(11):164–174, 2008.

[59] Zhaoqing Pan, Jianjun Lei, Yun Zhang, Xingming Sun, and Sam Kwong. Fast motion estimation based on content property for low-complexity h. 265/hevc encoder. *IEEE Transactions on Broadcasting*, 62(3):675–684, 2016.

[60] Fusang Zhang, Daqing Zhang, Jie Xiong, Hao Wang, Kai Niu, Beihong Jin, and Yuxiang Wang. From fresnel diffraction model to fine-grained human respiration sensing with commodity wi-fi devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), mar 2018.

[61] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S Cardoso, and Frank Piessens. Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia conference on computer and communications security*, pages 413–424, 2016.

[62] Rose Ceccio, Sophie Stephenson, Varun Chadha, Danny Yuxing Huang, and Rahul Chatterjee. Sneaky spy devices and defective detectors: the ecosystem of intimate partner surveillance with covert devices. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 123–140, 2023.

[63] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*, pages 2177–2184. IEEE, 2017.

[64] Matthew Gast. *802.11 wireless networks: the definitive guide*. O'Reilly Media, Inc., 2005.

[65] IEEE. Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379, 2021.

Appendix A.
Background: Detecting and Locating Hidden Wireless Cameras

Current wireless hidden camera detection methods generally rely on information leaked through wireless channels or other side channels when the camera is in operation. For example, wireless communication can unintentionally leak information through certain out-of-band channels, which has recently been leveraged for detecting the presence of wireless devices. Sathyamoorthy et al. [7] and Valero et al. [8] highlight the importance of carefully setting the received power threshold to avoid false positives or missed detections. Approaches like LAPD [10], CamRadar [25], and Heatdecam [11] rely on thermal/electromagnetic emissions and lens reflections to detect cameras in operation. These methods typically use specialized, often expensive sensors to capture side-channel information for detection. While effective in locating devices within the Line-of-Sight (LOS), these techniques require detection equipment to be in close proximity to the hidden camera to capture subtle changes in the signals, making them impractical for ordinary users and ineffective in hard-to-reach areas.

Some methods leverage WiFi packet sniffing to detect wireless cameras, as these cameras transmit data packets during operation. Systems like Dewicam [17], Cheng et al. [20], Liu et al. [9], and Miettinen et al. [63] achieved detection by learning the traffic characteristics of wireless cameras. However, machine learning-based approaches often face robustness issues due to their dependence on large training datasets. SNOOPDOG [13] and ScamF [15] focus on the causal relationship between wireless camera traffic and human activity, where significant movement within the monitored area increases encoded data traffic. This relationship provides valuable information for detecting surveillance. Motioncompass [14] and LocCams [16] also leverage side-channel information, such as the Organizationally Unique Identifier (OUI) in the MAC address, which can reveal the device's manufacturer and type.

The localization of wireless hidden cameras also relies on side-channel information leakage, but not all types of side-channel data are suitable for simultaneous detection and localization. Methods based on thermal/electromagnetic emissions [11], [25] and lens reflections [10] can detect and localize cameras by identifying regions with abnormal signals. However, these methods share similar limitations for localization as they do for detection: they are difficult to deploy and require proximity to the hidden camera [16]. Detection schemes that rely on traffic analysis require additional effort to achieve localization. For instance, these methods often depend on changes in RSSI strength or data flow as the user carrying the detection device moves around the space to infer the camera's location [12], [13], [15]. These schemes typically require the room to be nearly empty, which may not be feasible in real-world environments with furniture, as the user's mobility is constrained and they may not be able to approach the hidden camera. Recently, Loccams [16] introduced a method that uses CSI to determine whether the user is blocking the LOS path between the positioning equipment and the wireless camera, allowing for a rough estimate of the camera's location. However, this method has a localization resolution of only 45 degrees, and its deep learning-based approach suffers from poor robustness for environments and devices change.

TABLE 4: Received Signal Strength Indication (RSSI).

Signal Strength	Conclusion	Describe	Required for
-30 dBm	Amazing	Max achievable signal strength. Not typical or desirable in the real world.	N/A
-67 dBm	Very Good	Minimum signal strength for applications that require very reliable, timely delivery of data packets.	VoIP, video stream
-70 dBm	Okay	Minimum signal strength for reliable packet delivery.	Email, web
-80 dBm	Not Good	Minimum signal strength for basic connectivity. Packet delivery may be unreliable.	N/A
-90 dBm	Unusable	Approaching or drowning in the noise floor. Any functionality is highly unlikely.	N/A

Appendix B.
Fresnel Zone Visualization

The visualization of the Fresnel zones described in Section 2 is shown in Figure 16, consisting of a series of concentric ellipses.

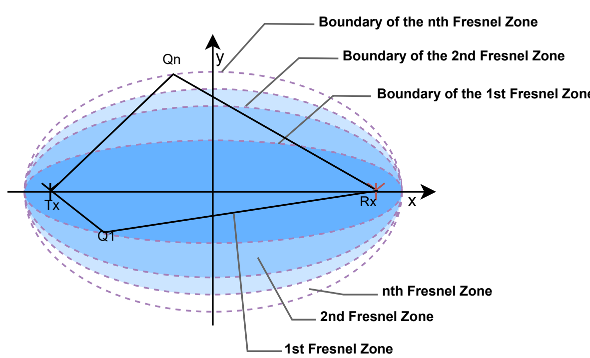


Figure 16: Illustration of Fresnel Zone.

Appendix C.
More Details of Camera Detection

We present the Received Signal Strength Indication (RSSI) requirements for various applications in Table 4. In practice, when CamLopa filters out APs based on RSSI, it retains a 5 dBm margin to avoid the risk of misdiscard.

The structure of an 802.11 wireless frame [64], [65] is shown in Figure 17. It consists of an unencrypted header

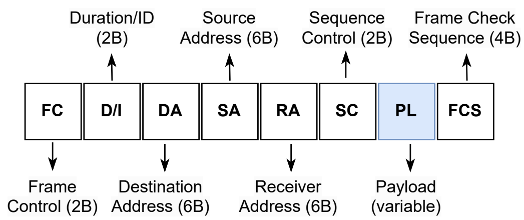


Figure 17: IEEE 802.11 wireless frame.

and an encrypted data payload. The header contains essential unencrypted information, such as addresses, while the payload is typically encrypted using WEP, WPA, or WPA2.

Regarding video compression standards, three types of frames are commonly used to compress video: I (Intra-coded picture) frames: these frames contain complete image information and can be decoded independently of other frames, P (Predicted picture) frames: these frames encode residual information and require information from preceding I frames for decoding, and B (Bi-directionally predicted picture) frames: these frames can construct images using changes from preceding I or P frames, subsequent I or P frames, or interpolations between preceding and subsequent I/P frames. Among these frame types, B frames are the most compressible, followed by P frames, and finally, I frames. In video footage captured by the camera, significant changes between frames lead to an increase in the number of P and B frames, which in turn results in higher upload traffic.

Appendix D. More Details of Evaluation Setting

We evaluated the performance of CamLopa on seven different wireless cameras, as listed in Table 5

TABLE 5: Cameras used in experiments.

Camera	Abbreviation	Cost
XiaoMi Cloud Camera2	Mi	24.5
XiaoYi Smart Camera Y4	Yi	20.4
EZVIZ C2C	C2C	24.5
360 Cloud Camera 8Pro	360	24.5
V380 Camera	V380	13.6
Guangchun Mini Camera	Gc	31.4
HiLEME Mini Camera	Hi	18.4

For hidden camera detection and localization. As shown in Figure 10, in each room, we select several potential locations suitable for monitoring the entire room to place the cameras for the experiments. The azimuths (path 1 as x-axis) of each point in room 1 are 28.61°, 42.27°, 60.28°, 88.54°, 130.1°, and 157.73 °, in room 2 are 4.86°, 51.34°, 69.44°, and 103.52 °, and in room 3 are 110.94°, 92.37°, 61.34°, 47.13°, and 30.69 °.

Appendix E. More Details of Prototype Implementation

CamLopa requires sniffing 802.11 packets to obtain CSI. Currently, most mobile devices require special permissions to perform sniffing, and due to the closed-source nature of wireless network card manufacturers, CSI extraction is only possible with certain network cards. However, acquiring this data poses no technical challenge but only involves permission issues. To ensure system applicability, we did not implement CamLopa on specific phone or computer platforms capable of extracting CSI. Instead, we chose the open-source, low-cost COTS device, the Raspberry Pi, as the platform for CamLopa.

Our code and demo are available at https://anonymous.4open.science/r/CamLoPA-Code-DFD5. The CamLopa prototype relies on the Raspberry Pi 4B hardware. The system is built on Raspberry Pi OS (kernel version 4.9, firmware version 7_45_189) and requires Python 3. Before using the system, you must first install the nexmoncsi tool and the necessary Python dependencies. Please ensure that you do not use upgrade commands during system setup, as updating the firmware may cause nexmoncsi to malfunction. Additionally, since this system version is older and no longer maintained, some required packages must be installed using the apt-get command instead of pip. After the review process, we will package the image and virtual environment, along with the necessary dependencies, and provide a download link to facilitate system replication for future users. During the installation of nexmoncsi, wireless network functionality is temporarily disabled. To restore wireless connectivity on the Raspberry Pi, you will need to manually activate the wireless interface and configure the network settings.