NIRF
Detecting Cameras That Hide Behind Screen

Ye, Hanting; van der Kolk, Niels; Wang, Qing

DOI
10.1145/3680207.3723496

Licence
CC BY

Publication date
2025

Document Version
Final published version

Published in
ACM MobiCom 2025 - Proceedings of the 2025 the 31st Annual International Conference on Mobile Computing and Networking

Citation (APA)
Ye, H., van der Kolk, N., & Wang, Q. (2025). NIRF: Detecting Cameras That Hide Behind Screen. In *ACM MobiCom 2025 - Proceedings of the 2025 the 31st Annual International Conference on Mobile Computing and Networking* (pp. 620-634). ACM. https://doi.org/10.1145/3680207.3723496

RESEARCH-ARTICLE

NIRF: Detecting Cameras That Hide Behind Screen

HANTING YE, Delft University of Technology, Delft, Zuid-Holland, Netherlands

NIELS VAN DER KOLK, Delft University of Technology, Delft, Zuid-Holland, Netherlands

QING WANG, Delft University of Technology, Delft, Zuid-Holland, Netherlands

Published: 03 November 2025

Citation in BibTeX format

ACM MOBICOM '25: 31st Annual International Conference on Mobile Computing and Networking
November 4 - 8, 2025
Hong Kong, China

Conference Sponsors:
SIGMOBILE

ACM MOBICOM '25: Proceedings of the 31st Annual International Conference on Mobile Computing and Networking (November 2025)
https://doi.org/10.1145/3680207.3723496
ISBN: 9798400711299

NIRF: Detecting Cameras That Hide Behind Screen

Hanting Ye, Niels van der Kolk, Qing Wang
Delft University of Technology, The Netherlands

Abstract

Hidden spy cameras are a growing global threat to personal privacy. With the emergence of translucent screen technology, a new security risk has arisen: cameras can now hide *behind* devices' screens like TVs and monitors that are common in private places, e.g., hotel rooms. The screen's *covering* over the hidden camera not only makes the cameras behind it unnoticeable to human eyes but also makes existing camera detection methods less effective. Inspired by recent advances in representing real-world scenes accurately using neural networks, we propose *Neural Infrared Reflectance Field (NIRF)* to learn the intricate optical properties of the screen and the cameras hidden behind it. Through NIRF, we design a new camera detection system by leveraging the unique reflective properties of behind-screen cameras and screens. We evaluate NIRF with thorough experiments on five smartphones. Our NIRF archives over 90% detection rate and is robust to different conditions, including varied backgrounds, ambient light levels, screen protectors, and screen contents. Besides, we conduct a field study by deploying 18 common spy cameras behind a 65-inch translucent TV and recruiting 27 people to compare NIRF with commercial hidden camera detectors. NIRF achieves an 89.5% detection rate, significantly outperforming the best commercial hidden camera detector that only has a 14.4% detection rate of behind-screen cameras.

CCS Concepts

• **Human-centered computing → Mobile computing**; • **Security and privacy → Privacy protections**.

Keywords

Deep learning, hidden behind-screen camera, privacy

ACM Reference Format:
Hanting Ye, Niels van der Kolk, Qing Wang. 2025. NIRF: Detecting Cameras That Hide Behind Screen . In *The 31st Annual International Conference on Mobile Computing and Networking (ACM MobiCom '25), November 4–8, 2025, Hong Kong, China.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3680207.3723496

ACM MobiCom '25, Hong Kong, China

ACM ISBN 979-8-4007-1129-9/2025/11
https://doi.org/10.1145/3680207.3723496

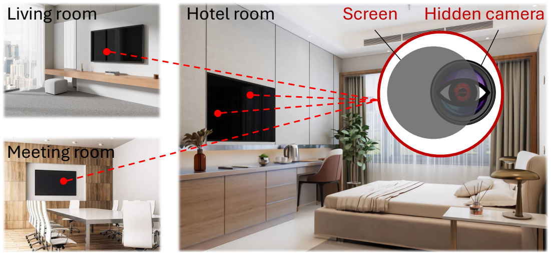


Figure 1: Illustration of the cameras hiding behind screens.

1 Introduction

Hidden spy cameras recording people in private spaces have increasingly become a global problem. For instance, in South Korea alone, a total of 5,541 hidden camera-related crimes were reported in 2021 [14]. Locations such as Airbnb accommodations are particularly attractive to hackers for installing these hidden cameras. A 2019 survey of rental accommodations revealed that 11% of visitors had actually discovered a hidden camera during one of their stays [28].

When entering an unfamiliar room, people worried about spy cameras often check devices like telephones, power sockets, and smoke alarms for possible hidden cameras. However, one crucial point has been overlooked: *the screens*, the main interface for displaying information, are rarely considered as a potential hiding spot for spy cameras. Screens are commonly found in hotel rooms and Airbnb rentals, as well as in meeting rooms. It's hard for people to imagine that a hidden camera could be placed *behind* a screen because the screen can block the camera, making the camera useless. However, recent advancements in translucent screen technology have made it possible to hide spy cameras behind screens! Translucent screen has been gradually applied to various electronic display devices, including but not limited to smartphones, laptops [55], and TVs [26], to maximize the screen-to-body ratio to improve the user experience. This technology allows a small part of the device's screen to be made translucent, still allowing light to pass through and reach the front camera placed behind/under the screen for selfies. Though the screen is translucent and light can pass through it, the front (selfie) camera placed behind the screen is *invisible* to human eyes, as highlighted in [39, 42, 51]. This reality creates a significant potential for behind-screen cameras to become a new attack vector in spy camera crimes. The screen's cover makes the camera even less noticeable, as the behind-screen camera cannot be seen whether the screen is ON (displaying content) or OFF (in standby mode)

[42, 51]. Combined with the small size of spy cameras—the lens diameter is often just 1 mm [7, 44, 54]—a translucent screen region only needs to be about 1 mm^2. Furthermore, public discussions and online forums have expressed growing concerns that this technology could be integrated not only into smartphone screens but also into any other types of screens, including TVs and monitors, potentially enabling covert surveillance [24, 25]. This raises significant privacy risks, as such hidden cameras could be deployed in sensitive locations like hotel rooms or top-secret spaces, as shown in Figure 1. Meanwhile, since people naturally face and look at screens, a behind-screen hidden camera could capture more private details than traditional hidden cameras. Therefore, although this is currently a hypothetical attack, it is necessary to proactively design behind-screen camera detection tools to address this potential threat.

On the other hand, several hidden camera detection methods have been proposed to address the threat of unlawful recording by *traditional* hidden cameras. One common technique involves analyzing wireless traffic generated by the camera's data transfer [11, 19, 27, 31, 46, 49]. However, a camera hidden behind a screen could use wired connections or store videos locally on memory cards, eliminating the need for WiFi or 4G connections. Meanwhile, the screen provides ample space for wired connections or memory storage. More importantly, this method can only detect the presence of hidden cameras but not their exact locations. Other recent studies have explored detecting leakage signals from camera operations, such as thermal analysis [66] or electromagnetic emanation [34], to locate hidden cameras. However, these commonly used detection methods on the market are well-known and effective only when cameras are active. Hackers can easily bypass them by turning off the camera, making it undetectable while the user checks for hidden cameras. Hidden cameras can be remotely controlled or scheduled to activate after a delay, enabling illegal recording—such as one hour after the user enters the room, when they have let their guard down. Many cameras on the market include scheduling functionality that can be exploited for delayed activation [15]. Therefore, the ideal solution would be to detect hidden cameras, whether they are turned ON or OFF.

Recently, an interesting work successfully leveraged Time-of-Flight (ToF) sensors, which are widely available in the commercial market and used in modern smartphones, to locate cameras hidden inside pre-identified suspicious objects [47] based on the camera's high-intensity retro-reflection. The reason behind that is the camera lens must remain exposed to capture light in order to record video. Consequently, a hidden camera has to reflect the light emitted by the ToF sensor. Though interesting, a significant challenge arises when the camera hides *behind* a translucent screen: the glossy surface of the screen also generates high-intensity reflections, which significantly degrades the camera detection performance (cf. Section 6). Thus, it is essential to design new approaches that can precisely detect behind-screen cameras, regardless of whether they are wired or wireless, powered ON or OFF.

Detecting cameras that hide behind screens is challenging. Without the obstruction of a screen, the hidden camera has already only been observable within a limited angle, specifically within a Field-of-View (FoV) cone projected from the hidden camera. However, when the camera is hidden behind the screen, the larger reflections from the screen (the screen size is relatively larger) block the observation of tiny reflections from the camera (the camera size is relatively smaller). This further narrows the already restricted observation angle, making it much more difficult to detect behind-screen hidden cameras. Thus, the *first challenge* posed by the screen reflections is *where is the right position to observe the high-intensity reflection from the hidden camera.*

Furthermore, even if we are in the right position to observe the high-intensity reflection from the hidden camera, the high-intensity reflection from the screen is still there. We can still see high-intensity reflections from the screen and the behind-screen camera simultaneously. Then, the *second challenge* is *how can we accurately distinguish the reflection of hidden behind-screen cameras from the noisy screen reflections.*

The key to overcoming screen reflections lies in our observation that while the screen's reflections shift with the movement of the ToF sensor, the camera's reflections remain stationary. This makes the detection process highly dependent on the sensor's position. Inspired by recent advancements in neural radiance fields, which have proven highly effective in representing 3D scenes, we propose *Neural Infrared Reflectance Field (NIRF)* to address the challenges posed by complex screen reflections. NIRF is designed to learn the intricate optical properties, such as the transmittance and reflectance of objects like the screen and the hidden camera behind it. In NIRF, we can resolve the challenges of *where to find* and *how to distinguish* the hidden behind-screen camera by predicting light reflected from both the screen and the behind-screen camera from any observed positions.

However, although NIRF can capture the optical properties of all objects in the scene, it represents them implicitly, i.e., we cannot explicitly control or isolate any specific object in NIRF, such as the hidden camera. To obtain an image of a particular object in NIRF, we need to render a ray through the implicit scene, allowing it to interact with objects, and then capture the reflected light to form an image—a process known as ray tracing[1]. Therefore, the *third challenge* we face is *how to accurately identify the hidden behind-screen camera in the implicit NIRF scene representation.*

[1] Ray tracing is a computer graphics technique used to simulate the path of light through a scene to calculate shadows, reflections, etc., of objects [1].

To address this challenge, we propose a two-stage detection method. *In the first stage (coarse-grained)*, we observe that there are *double reflections*, from both the screen and the behind-screen hidden camera. Due to the extremely short time difference between the double reflections, they are difficult to distinguish using the direct ToF sensor output. To overcome this, we exploit the phase outputs of our NIRF to detect the subtle depth difference between the screen and behind-screen camera caused by the double reflections. This allows us to search for suspicious camera locations in NIRF. *In the second stage (fine-grained)*, we use ray tracing to pinpoint hidden camera from all suspicious locations. As light rays pass through both the screen and behind-screen camera represented in NIRF, we observe a variation pattern in the transmittance and reflectance. This pattern is consistent regardless of the ray's starting position or direction as long as it passes through both the screen and the hidden camera. This enables us to accurately pinpoint the hidden camera's location and filter out noise from screen reflections.

To the best of our knowledge, we are the first to identify the critical security risks posed by spy cameras hidden behind screens. With all our design components presented above, we implement NIRF using an off-the-shelf ToF sensor and evaluate its performance and robustness on five state-of-the-art full-screen smartphones. Our NIRF achieves over 90% detection accuracy for behind-screen hidden cameras while maintaining a false positive rate below 5%. Furthermore, we conduct a field study by hiding 18 popular spy cameras with different parameters (e.g., lens size ranging from 1 mm to 12 mm) behind a 65-inch translucent screen and recruiting 27 participants to compare our NIRF with three Commercial Off-The-Shelf (COTS) hidden camera detectors and naked-eye detection. In the field study, NIRF achieves an 89.5% behind-screen camera detection rate, significantly outperforming the best COTS hidden camera detector, which only yields a 14.4% detection rate of behind-screen cameras.

2 Background

Spy cameras (hidden behind translucent screen). There is growing concern over the privacy risks posed by hidden spy cameras in locations such as hotels and Airbnb rentals [7, 12, 40, 52]. Spy cameras can be categorized as either (i) inconspicuous cameras, which are stealthily hidden in discreet locations, or (ii) camouflaged cameras, which are disguised as everyday objects like picture frames, smoke detectors, or alarm clocks. The former could be easily detected by existing tools on the market [2–4] or by some recent research efforts [11, 31, 47]. However, the latter, camouflaged cameras, pose a greater challenge due to their *customizable* nature. Hackers can modify their size, shape, and placement to blend seamlessly into ordinary objects, making detection more difficult. Recently, advancements in translucent screen technology have enabled the *integration of cameras behind device screens* [62, 65]. These devices utilize a small translucent screen area to cover the camera, enabling the translucent screen to display content while still allowing light to reach the camera for photography [58]. The screen naturally acts as a cover, making behind-screen camera act as a camouflaged camera. This creates an opportunity for hackers to customize TVs, conference monitors, and other screen-based devices for surveillance by concealing spy cameras behind the screen.

Camera's retro-reflections. Retro-reflection, also known as "cat-eye reflection", occurs when almost all incident light is reflected directly back to its source. In a spy camera composed of a lens and an image sensor, the lens gathers all incident light, while the image sensor acts as a reflective surface to form a retro-reflection system. The retro-reflected light's energy is significantly stronger than other ambient diffuse reflections [29]. These reflections are clearly visible within a limited Field of View (FoV). The best way to detect these retro-reflections is to use a Time-of-Flight (ToF) sensor, as adopted in [47], because the ToF sensor naturally includes a light source that emits infrared light to illuminate the scene and a camera that detects the reflected infrared light.

Neural Radiance Fields (NeRF). It was developed in [37] to model a scene's optical radiance field by training a fully connected feedforward artificial neural network with a series of the scene's images taken from different angles. NeRF treats each image pixel as the result of ray tracing process, which encapsulates the scene features such as object opacity, transmittance, and reflectance of the optical radiance field. Once trained on a few images, NeRF can predict the outcome of ray tracing from any other directions and synthesize a complete image from a given viewpoint. NeRF has achieved remarkable success in view synthesis and 3D model rendering [6], immersive street views [36], facial reconstruction [17], human body modeling [41], and channel modeling [68].

3 When Cameras Hide Behind The Translucent Screen

In this section, we describe what we encounter when detecting behind-screen hidden cameras using their retro-reflections. Although a ToF sensor can still capture the retro-reflections from behind-screen cameras, the reflections from the translucent screen—which 'covers' the camera—challenge identifying the correct reflections we need to detect hidden cameras.

3.1 Reflectivity of the Screen

As a glossy surface, the screen emits a strong specular reflection in response to the light emitted by the ToF sensor. When this strong reflection is captured by the ToF sensor, a complex reflection pattern of the screen is observed. This pattern is influenced by the shape and arrangement of the screen's pixels [8]. As shown in Figure 2, the detected reflection patterns of

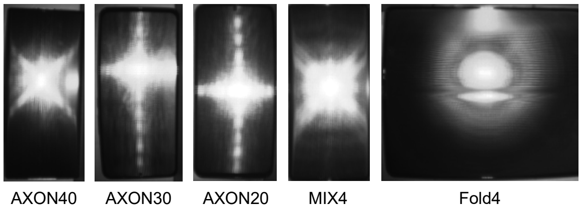


Figure 2: Reflection patterns of various smartphone models.

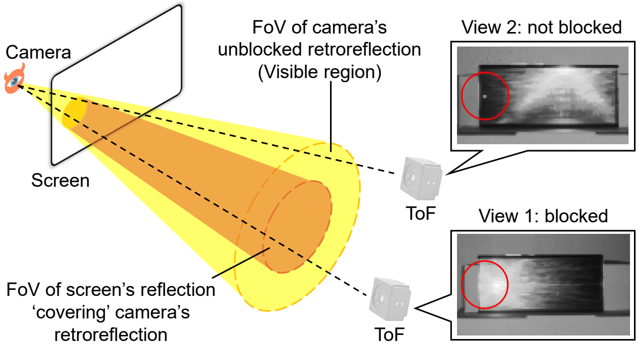

Camera
Screen
FoV of camera's unblocked retroreflection (Visible region)
View 2: not blocked
ToF
View 1: blocked
ToF
FoV of screen's reflection 'covering' camera's retroreflection

Figure 3: Illustration of the screen's effect on the camera's optical reflection FoV. The orange area is the screen's specular reflection FoV, while the yellow area is the behind-screen camera's reflection FoV. The screen's 'covering' on the camera greatly reduces the camera's original reflection FoV.

different smartphone screens vary. The retro-reflection from the hidden camera is usually very small, typically occupying no more than 9 pixels [47]. The small, intense reflections at the edges of the reflection patterns can produce spurious reflections, leading to potential misjudgments.

3.2 Effect on the Reflection Field-of-View

The strong specular screen reflection also significantly affects the reflection Field-of-View (FoV) of behind-screen cameras. Without the 'covering' of the screen, the camera's reflections can be captured by the ToF sensor within a cone-shaped area. However, for behind-screen cameras (the screen 'covers' the cameras), there is a cone centered around the 0° angle where specular reflections dominate, making a camera's reflections indistinguishable. As shown in Figure 3, the screen's specular reflection completely obscures the camera's retro-reflection in the orange area, making it invisible. Only in the yellow area–the cone's side lobes, called *visible region*–can the camera's tiny reflection be detected. Finding the visible region in various real-life scenarios is challenging without prior knowledge of the camera's location.

3.3 Key Observations for Camera Detection

In the presence of screen reflections, it is crucial to accurately identify which reflection originates from the behind-screen hidden camera rather than from the screen itself. Below, we present our key observations that can be exploited to identify the correct reflection from the behind-screen hidden camera.

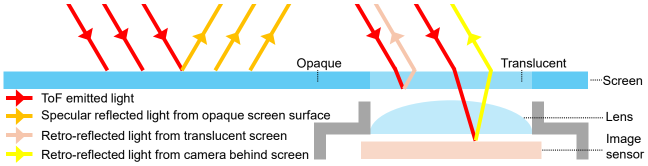

Opaque
Translucent
Screen
ToF emitted light
Specular reflected light from opaque screen surface
Retro-reflected light from translucent screen
Retro-reflected light from camera behind screen
Lens
Image sensor

Figure 4: Double reflections: the reflections from different parts of the screen and the behind-screen hidden camera.

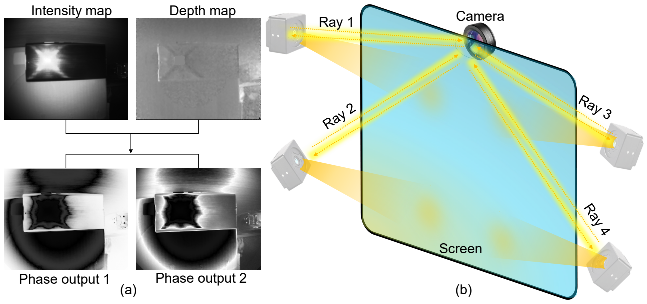


Figure 5: Observations: (a) The phase output can be used to observe the depth difference caused by double reflections; (b) As the ToF sensor moves, the position of the behind-screen camera's retro-reflections *remains static* relative to the screen, while the screen's specular reflections *change dynamically*.

Observation 1: Double reflections. Except for the specular screen reflection, the ToF sensor also detects double reflections from the unique sandwich structure of the translucent screen and the hidden camera behind it, as shown in Figure 4. One portion of the incident light is retro-reflected from the surface of the translucent screen, while the other portion passes through the screen and reaches the hidden camera. This light is then converged by its lens and fully reflected back from its image sensor surface to the ToF sensor. These double reflections allow the ToF sensor to detect tiny depth differences in the final depth map, representing the distance between the screen and the hidden camera. However, these subtle differences are often hard to observe in the ToF's depth map. Inspired by ToF-based texture recovery work [60, 61], we can utilize the phase output of the ToF to reveal clear depth differences, as shown in Figure 5(a).

Observation 2: Retro-reflection consistency. When the ToF sensor moves relative to the screen, the specular reflection from the screen shifts, while the retro-reflection from the hidden camera remains fixed, as shown in Figure 5(b). This retro-reflection consistency helps identify the camera's visible region during ToF movement. In the *visible region (cf. Section 3.2)*, regardless of the ToF sensor's movement, the retro-reflection of the hidden camera stays constant. To effectively track and verify this retro-reflection consistency, users must capture these reflections from multiple perspectives when they temporarily appear within visible regions. However, it is challenging since users may not know when the ToF is within the camera's visible region. In real-world scenarios, the size of the visible regions is significantly affected by the screen's reflectivity, as well as the angle and distance of the ToF sensor relative to the screen.

Recently, many works on advanced Neural Radiance Fields (NeRF) have relied on RGB and depth images to enhance scene reconstruction performance[5, 13, 32]. Instead of using both RGB and depth images, we propose using only phase data captured by a ToF sensor to address the aforementioned camera detection-specific challenges: the infrared light emitted by a ToF sensor is used to construct a *Neural Infrared Reflectance Fields (NIRF)*, which generates a sufficiently accurate observation of the target screen surfaces. NIRF learns optical properties, such as transmittance and reflectance, of objects like the screen and the hidden camera behind it. Through our NIRF, we can predict the infrared light reflected by the screen and the behind-screen camera, as captured by the ToF sensor at any position and viewpoint, even unseen ones. This capability will enable us to infer the positions of behind-screen hidden cameras based on the observation we have presented in this subsection.

4 Modelling The NIRF

To model the Neural Infrared Reflectance Fields (NIRF), we first discretize the screen scene into a finite number of small 3D voxels. Given the ToF sensor's position and direction (i.e., shooting angle), these voxels are used to re-render the images that would be captured from the corresponding position and direction. A ToF sensor is combined with an IR light source and an IR camera. and movements of the ToF sensor will cause changes in lighting conditions. For this, we then derive in Section 4.1 the scene's appearance in response to allocating a point light source with the ToF sensor using the inverse square law of light propagation. Meanwhile, the ToF sensor has multiple successive windows for receiving reflected light and uses the phase shift of the reflected light to calculate the distance. As mentioned in Observation 1, we need to utilize phase outputs from these windows to learn rich-detailed optical properties. Therefore, we derive the phase output of ToF in Section 4.2 and design a new neural phase rendering procedure that models phase outputs captured by a ToF sensor in Section 4.3. Lastly, since ToF changes its position over time during the actual camera detection process, we integrate a time variable to facilitate better training of the time-dependent network (Section 4.4).

4.1 Infrared Light Source

A ToF sensor responds only to the light from its own IR point source and not to any ambient illumination. For an IR light source in the ToF sensor positioned at $\mathbf{o}$, each voxel is illuminated from a single direction. Since the IR camera that acts as the receiver is collocated with the IR light source in the ToF sensor, as shown in Figure 6(a), the incident radiance from the incident direction $\mathbf{w}_i$ at a position $\mathbf{o}_s = \mathbf{o} + \mathbf{w}_i s$ (i.e., a voxel) along the ray that meets the inverse square law of light fall-off is given by

$$L_i(\mathbf{o}_s, \mathbf{w}_i) = \frac{I}{\|\mathbf{o}_s - \mathbf{o}\|^2}\delta\left(\frac{\mathbf{o}_s - \mathbf{o}}{\|\mathbf{o}_s - \mathbf{o}\|^2} - \mathbf{w}_i\right), \qquad (1)$$

where I represents the emitted intensity of the IR light source, and $\delta(\cdot)$ is the Dirac distribution describing light coming from only a single direction.

4.2 ToF Phase Output

The IR light source sends a modulated light signal into the environment, and a ToF sensor measures the time it takes for the light to reflect back. The measurement of the ToF sensor can be expressed as $h(t) = \int_0^T f(\tau - t) g(\tau) d\tau$, where $h(t)$ represents a convolution process that describes the ToF sensor's response to the light it emits. Here, $f(t) = \frac{1}{2}\sin(2\pi f t) + \frac{1}{2}$ is the modulated temporally-varying intensity function of the light source with modulation frequency f, and $g(t) = \sin(2\pi f t + \phi)$ is the ToF exposure response function. To measure the phase shift between the incident light and its response in order to calculate the time-of-flight, $f(t)$ and $g(t)$ are periodic functions with the same period T. Given the constant speed of light c, the temporal information is determined by the traveled path length s and derived as $t = s/c$. After capturing different phase offsets $\{0, \pi/2, \pi, 3\pi/2\}$, the measurements are:

$$h_{\text{ToF}}(s) = (h(0) - h(\pi)) - i\left(h\left(\frac{\pi}{2}\right) - h\left(\frac{3\pi}{2}\right)\right) = \exp\left(\frac{i2\pi f s}{c}\right). \qquad (2)$$

Therefore, the response at every pixel is the phase output 1 (the phase offset from 0 to π) and phase output 2 (the phase offset from $\pi/2$ to $3\pi/2$) of a voxel in the scene.

4.3 Neural Phase Rendering

Given the ToF pose, a phase output is generated by tracing rays through the object and computing the reflectance observed along the ray as follows:

$$\begin{aligned} L(\mathbf{o}, \mathbf{w}_e) &= \int_{s_n}^{s_f} \sigma(\mathbf{o}_s) L_e(\mathbf{o}_s, \mathbf{w}_e) h_{\text{ToF}}(2\|\mathbf{o}_s - \mathbf{o}\|) ds \\ &= \int_{s_n}^{s_f} \sigma(\mathbf{o}_s) \underbrace{\frac{T_r(\mathbf{o}, \mathbf{o}_s)^2 b(\mathbf{o}, \mathbf{w}_e) I \exp\left(\frac{i4\pi f\|\mathbf{o}_s - \mathbf{o}\|}{c}\right)}{\|\mathbf{o}_s - \mathbf{o}\|^2}}_{L_r(\mathbf{o}_s, \mathbf{w}_e)} ds, \end{aligned} \qquad (3)$$

where $L_e(\mathbf{o}_s, \mathbf{w}_e) = \int_{\mathcal{S}} b(\mathbf{o}, \mathbf{w}_i, \mathbf{w}_e) L_i(\mathbf{o}_s, \mathbf{w}_i) d\mathbf{w}_i$ represents the IR light radiated from a voxel in a direction $\mathbf{w}_e$. Here, $\mathcal{S}$ is the unit sphere of incident directions. $b(\mathbf{o}_s, \mathbf{w}_i, \mathbf{w}_e)$ is the bidirectional reflectance distribution function [1, 9] that describes how much light is reflected into each outgoing direction $\mathbf{w}_e$ from each incoming direction $\mathbf{w}_i$ at voxel $\mathbf{o}_s$. In the case of retro-reflection from a translucent screen and behind-screen camera, where $\mathbf{w}_e = -\mathbf{w}_i$, as shown in Figure 6(a). $L_r(\mathbf{o}_s, \mathbf{w}_e)$ represents the reflectance from voxel $\mathbf{o}_s$ in

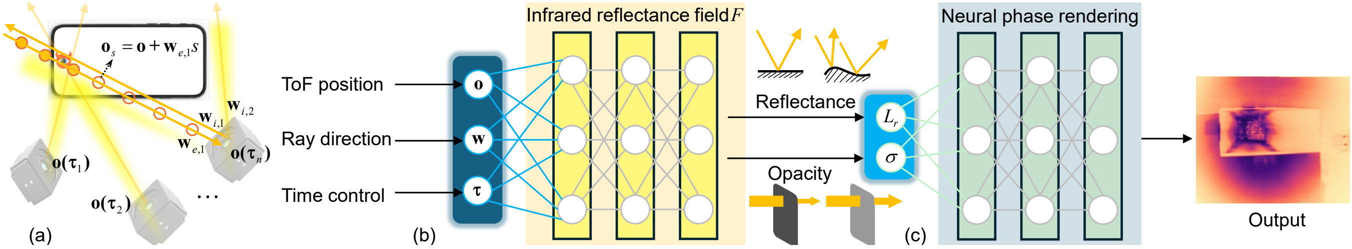


Figure 6: An overview of the representation and rendering procedure of our proposed NIRF (Neural Infrared Reflectance Field): (a) Sampling opacity and reflectance at various points along the ToF rays in §4.1; (b) Feeding ToF position, ray direction, and time control into a network to learn the infrared reflectance fields for scene representation, producing the reflectance and opacity distribution along the ray in §4.2 and §4.3; (c) Ray tracing through the learned field generates a phase output in §4.4.

direction $\mathbf{w}_e$. $T_r(\mathbf{o}, \mathbf{o}_s) = \exp\left(-\int_{s_n}^{s} \sigma(\mathbf{o}_t)dt\right)$ describes the accumulated transmittance for light prograting from position $\mathbf{o}$ to $\mathbf{o}_s$ for near and far bounds $t \in [s_n, s_f]$. As shown in Figure 6(b), L_r characterizes the visual appearance of different materials (e.g., shiny or matte). $\sigma(\mathbf{o}_s)$ is the opacity function, which controls the opacity at voxel $\mathbf{o}_s$, and large values of σ represent opaque regions, while small values represent transparent ones. NIRF using phases provides a better way to model the complex reflectance response in scenarios involving screens and cameras hidden behind them, as mentioned in Section 3.3.

4.4 Training the NIRF Network

Following NeRF, we build an MLP-based (i.e., fully connected feedforward artificial neural network) NIRF network, denoted as F, to learn the complex reflectance L_r and opacity σ of objects (e.g., screens and behind-screen hidden cameras) within the entire infrared reflectance field. The developed network F is a function of ToF sensor position $\mathbf{o}$ and ray direction $\mathbf{w}_e$, and it outputs both the opacity $\sigma(\mathbf{o}_s)$ and the reflectance $L_r(\mathbf{o}_s, \mathbf{w}_e)$ of a voxel $\mathbf{o}_s$ in light ray passing through $\mathbf{o}$ in direction $\mathbf{w}_e$. These two important optical properties functions can be used to render phase outputs of a scene from any given ToF position and direction, as shown in Figure 6(c). Further, we follow [33] to add a temporal variable τ which is optimized per frame to predict a position- and time-dependent blending weight $\mathbf{o}(\tau)$ to blend the NIRF network's outputs $L(\mathbf{o}, \mathbf{w}_e)$. This allows us to control the network's output to vary with time. Given a set of phase measurements for a scene captured by a ToF sensor at different time instances, we use NIRF to sample a set of ToF rays, render the phase output, and minimize the L2 loss between the rendered and measured phases. To train our NIRF network F, we must obtain the ToF sensor poses (i.e., ToF sensor position and ray direction). Typically, input poses for training are recovered using multiple captured RGB images [48]. However, this method cannot be directly applied to ToF sensor poses, as any recovered ToF poses outside the depth measurement range $[s_n, s_f]$ cannot be used in the training process. Given that the ToF phase output contains depth information that can assist in determining the ToF pose [16, 59], we optimize ToF poses from scratch within the training loop.

5 Detecting Behind-Screen Camera

5.1 Screen Detection

The learned NIRF in Section 4 is an implicit neural representation of a scene containing screens and behind-screen cameras. As a result, we cannot explicitly control the position of the screen within the model. However, devices equipped with screens can be installed in various environments and recorded by the ToF sensor from different angles or distances by users in real-life scenarios. To address this, we detect the screen within the phase output of the NIRF using the Segment Anything Model (SAM) [45]. SAM is a promptable model based on zero-shot learning, meaning it can predict masks for objects belonging to classes that were not seen during training. As illustrated in Figure 7(a), screen masks are generated for images captured by a ToF sensor at different positions and angles. At the start of the screen detection stage, the user indicates that the screen in the scene needs to be detected, and this selection is input into SAM as prompts. In the subsequent stages, the screen masks corresponding to these objects are tracked, and the presence of cameras hidden behind the screen is detected within the masked area.

5.2 Coarse-Grained Camera Detection

As discussed in Section 3.2, the FoV of a hidden camera behind the screen is limited and can often only be observed within a narrow range, named as visible regions. Therefore, we need to conduct a coarse-grained search in obtained NIRF to locate the suspicious positions that may contain the hidden camera. We use the time variable τ to control the movement of the ToF sensor until it is positioned within a visible region.

How to know ToF sensor is in the visible region? This can be determined by the slight depth difference between the screen and the camera behind it (as discussed in Observation 1 in Section 3.3). As illustrated in Figure 7(b), there is a phase difference between phase output 1 and phase output 2 on the

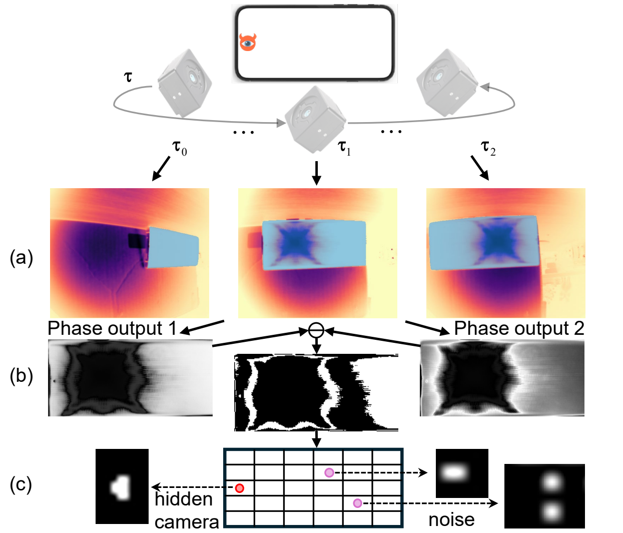


Figure 7: The process of detecting behind-screen cameras: (a) Screen detection is performed using SAM across different times in §5.1; (b) The obtained screen output is converted into successive ToF phase outputs, and the phase differences are used to detect small blobs that exhibit the reflection characteristics of the hidden camera in coarse-grained detection stage (§5.2); (c) All detected blobs are mapped to the standard reflection grid, and the fine-grained detection (§5.3) stage is applied to eliminate false positives caused by noise.

screen, caused by the depth difference between the screen and the behind-screen camera. Given the extremely small size of the translucent screen region and the small size of the hidden camera, the detected blobs are typically very small. This small size feature further aids in identifying the visible regions. Using these observations, we superimpose the difference between phase output 1 and 2, generated at each time instance τ, to identify the suspicious location of the behind-screen camera on the screen.

Reflection grid. To facilitate the coarse-grained search and localization of the visible region, we propose a reflection grid, which tracks the position of reflections. As shown in Figure 7(c), this grid is based on the screen detected in Section 5.1. The aspect ratio of the screen should match that of the grid. The screen mask generated via SAM, with the largest squareness, will determine the grid's aspect ratio. This square screen mask, with a length U and width V, is divided into $M \times N$ cells, where M is the number of rows and N is the number of columns. The dimensions of each cell are $U/M \times V/N$, and each cell is denoted as $c_{i,j}$ where i is the row and j is the column. However, a challenge arises in mapping the detected screen mask to this grid. Since the output of NIRF is generated from various views, the shape of the detected screen mask may be distorted and deviate from the grid's rectangular structure. To address this, we employ homography decomposition [35], denoted as M_h. M_h describes the spatial relationship between multiple views of the same object and is used to map the screen detected from different viewing angles to a standard reflection grid.

We consider a reflection to be detected only when all pixels of the detected reflection blob fall within the cell. This approach is necessary because the retro-reflection from the translucent screen and the camera behind it is very small. This method helps avoid large noise caused by specular reflections from the screen. Once the reflections have been localized, the location of a reflection found on the screen, $c_{i,j} = 1, \text{size}(M_h L) < \frac{U}{M} \times \frac{V}{N}$, can be translated to a location inside the cell using the computed transformation matrix.

5.3 Fine-Grained Camera Detection

After identifying all suspicious camera positions on the screen, there are still a lot of false positives arise from the leakage of the strong screen reflections, as shown in Figure 7(c). Therefore, we need a fine-grained camera detection stage to remove all noises and find the true camera hidden behind the screen. This stage is based on two key observations:

- *Consistent retro-reflection position across different ToF positions:* The distinctive behavior of a retro-reflection from the translucent screen and camera behind it is that the reflections remain in the same location on the screen. These reflections should be found within the same cell in the reflection grid, regardless of how the ToF sensor's position is moved within the NIRF.
- *Consistent optical properties across ray tracing in different directions:* Even when ToF sensors at different locations emit rays at varying shooting angles through the translucent screen and the camera behind it, consistent optical properties, such as transmittance and reflectance, should be observed. Specifically, the accumulated transmittance of the rays will first decrease slightly when passing through the translucent screen and then decrease further when passing through the screen and reaching the camera. From a reflectance standpoint, when the rays pass through the translucent screen, a small peak will be detected due to the screen's retro-reflection, and when the rays reach the camera, a larger peak will be detected due to the stronger retro-reflection of the hidden camera.

Based on the first observation, we generate multiple camera poses around the searched time τ from the coarse-grained detection stage, as shown in Figure 8(a). From each camera pose, we emit light rays from all suspicious areas to sample reflection points along the ray, as shown in Figure 8(b). Using the second observation, we look for consistency in the optical properties of rays generated from different directions. As shown in Figure 8(c), for transmittance, we expect two drops: one after encountering the screen and another after passing through the behind-screen camera. For reflectance, we expect a small peak from the screen reflection, followed

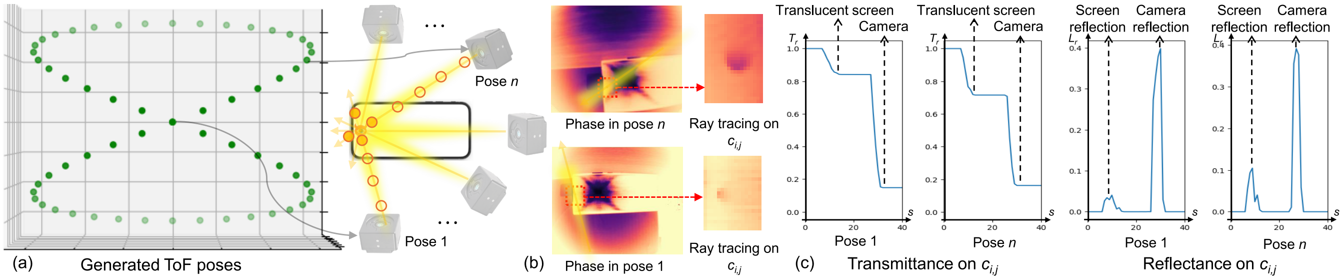


Figure 8: Multi-view ray tracing process: (a) The ToF sensor generates multiple ToF poses, representing multi-view observations within the visible region of the behind-screen camera; (b) Ray tracing is performed on the suspicious cell, where a sampled ray is emitted from the ToF sensor and passes through the suspicious cell; (c) The transmittance and reflectance calculated at the sampling points along the ray's path are used to verify the existence of the behind-screen camera. Here, s represents the ray distance—smaller s indicates a sampling point closer to the ToF light source, while larger s indicates a sampling point farther away. As the ray passes through the translucent screen and reaches the behind-screen camera, significant changes in transmittance and reflectance can be observed.

by a larger peak from the behind-screen hidden camera reflection. If these patterns are observed, it indicates that the ray from this pose has successfully detected the spy camera.

To confirm that a suspicious cell indeed contains a hidden camera and not a random error, we rely on the fact that there will always be a visible region where the ToF sensor can fully capture the camera's retro-reflections, as the ToF sensor is inside the behind-screen camera's FoV. For the set of frames captured within the visible region of the behind-screen camera, the corresponding reflection cell should be consistent across rays generated from all poses in multiple time slots τ, showing the patterns of the same reflectance and transmittance. Therefore, we conclude that a camera is hidden in this cell if more than half of the times corresponding to the cell identified in Section 5.2 show reflection consistency.

6 Performance Evaluation

6.1 Evaluation Setup

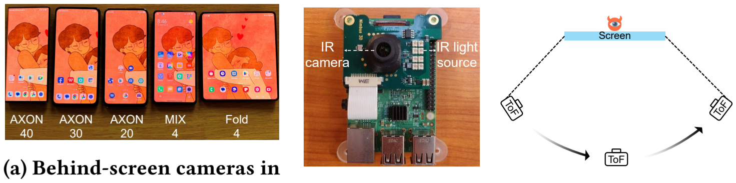
The used behind-screen hidden cameras. We use existing behind-screen cameras on *five full-screen smartphones* (ZTE AXON20/30/40, Xiaomi MIX4, and Samsung Fold4) as our targeted behind-screen hidden cameras, as shown in Figure 9(a). All these smartphones feature a translucent screen region that conceals the front camera, making it difficult to detect. Each translucent screen has a high resolution of 400 pixels per inch. All the cameras can capture high-resolution images exceeding 10 megapixels (MP) and have wide-angle lenses offering a FoV between 60 and 90 degrees.

Time-of-Flight (ToF) sensor. We select the Pieye Nimbus 3D ToF sensor [43] as the hidden screen camera detector. As shown in Figure 9(b), this sensor has a resolution of 352×288, comparable to the ToF sensors found in modern smartphones. Both the high dynamic range mode and exposure time on this ToF sensor are set to be automatic.

Implementation. We follow the setup of previous work [47] to let the ToF sensor move around 60 cm away from the screen but within a 120-degree FoV range in front of the screen to record phase data, as shown in Figure 10(c). We extract 120 frames from the front-scanned recordings at intervals of 3 frames, where 100 frames are randomly selected for training, and the remaining 20 frames are for testing the NIRF model.[2] After capturing the image, the Pieye Nimbus 3D ToF sensor uploads scanned images to the server, where an NVIDIA A10 GPU is used for training and detecting the behind-screen camera. 64 points are sampled on each ray with a configuration similar to NeRF [37]. The Adam optimizer is adapted to optimize the weights of NIRF with a learning rate of 0.001. After 5000 iterations, we halve the learning rate. Other hyper-parameters remain at default values (e.g., $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 0.9$).

Metrics. We use two metrics in the evaluation: (1) *Detection rate*: A successful detection of the behind-screen camera refers to when a method accurately identifies the presence of a hidden camera and correctly pinpoints its location, matching the actual position of the behind-screen camera. This is also known as a True Positive. The detection rate refers to the percentage of successful detections within a given set of measurements, which ideally should be 100%; (2) *False positive*: A false positive arises when the system incorrectly identifies a location on the screen as containing a camera, but this location does not correspond to the actual position of the behind-screen camera. The false positive rate refers to the percentage of false positives among all outputs generated by the system, with an ideal target of 0%. This metric is particularly critical in hidden camera detection, as false alarms not only waste the user's time and effort but also undermine the credibility of the detection method.

[2] We observed that using fewer training inputs requires more iterations to reach the same rendering performance but may miss finer details, which is also aligned with recent studies [22]. Given that the training loop for NIRF is nearly the same time regardless of the number of training images, we opted for 100 training frames to ensure a comprehensive NIRF.

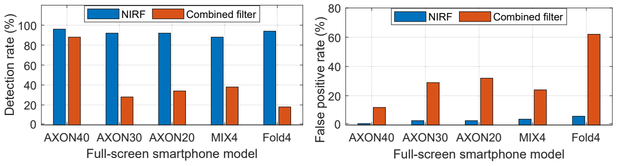


(a) Behind-screen cameras in five commercial smartphones are used in evaluation

(b) The Nimbus ToF sensor module

(c) Top view of ToF sensor scan range

Figure 9: Setup: behind-screen cameras and the ToF sensor.



(a) Detection rate

(b) False positive rate

Figure 10: Performance under different smartphones.

We compare the performance of two solutions: (1) *Our NIRF:* As detailed in the previous sections, our solution leverages infrared-based ToF sensor captures to construct a neural infrared reflectance field. By designing both coarse-grained and fine-grained stages, we can detect the camera hidden behind the translucent screen based on the principle of ray tracing within the constructed neural infrared reflectance field; (2) *Combined filter:* This approach also uses the ToF output but applies only the conventional combined filter scheme to reduce noise from screen reflections [47].

6.2 Preliminary Evaluation

We test NIRF across different smartphone models to assess its performance, using five different smartphones (see Figure 9(a)). As illustrated in Figure 10(a), the detection rate of our NIRF exceeds 90%. The combined filter solution, however, produces varying results depending on the specific reflection patterns of each screen. The detection rate is above 80% on the AXON40, but it is lower on the other four smartphones, with the Fold4 achieving a detection rate of less than 20%. Also, our NIRF demonstrates an extremely low false positive rate across different devices, thanks to the verification of the screen's reflective properties and the behind-screen camera using the neural infrared reflectance field (as shown in Figure 10(b)). On the other hand, the combined filter generally results in a false positive rate exceeding 10%. This occurs because there is significant noise from various reflection patterns, and a simple combined filter cannot effectively eliminate all noise from these complex patterns. As shown in Figure 2, specular reflection patterns from the ZTE AXON30 and AXON20 have strong, wide vertical components that interfere more than those from other smartphones, thus often leading to false identification as camera reflections. Meanwhile, Samsung Fold4 and Xiaomi MIX4 exhibit larger and stronger reflection patterns compared to ZTE AXON40, resulting in higher false positives.

6.3 Ablation Study

In this section, we evaluate the effectiveness of two designed stages, i.e., the first coarse-grained stage (S1), and the second fine-grained stage (S2), for behind-screen camera detection based on AXON40 smartphone. The results are shown in Figure 11. We observe that the performance of NIRF degrades when either S1 or S2 is removed, indicating that both components contribute significantly to camera detection. Specifically, without S1, the detection rate drops significantly to below 20% because most observations in the NIRF field fail to detect the camera's presence due to the limited visibility of the behind-screen camera. Without S2, although the camera's presence may still be detected, the output is accompanied by a large number of false positives (>80%) caused by screen reflections, which can severely affect user judgment.

6.4 Robustness Evaluation

Next, we evaluate NIRF's robustness performance in detecting behind-screen hidden cameras under various conditions.

Impact of background. Considering that the screens are often placed in different environments where the background wall may be built with various materials, we evaluate the performance of NIRF under different wall materials like paper, wood, steel, cement, and glass, as shown in Figure 14. The evaluation results are presented in Figure 12(a). We can observe that the hidden camera detection rate of our NIRF remains above 80% across different wall materials. In contrast, the combined filter method shows a detection rate of less than 40% under cement and glass walls due to the more complex reflections these materials introduce. Furthermore, the false positive rate of our NIRF stays below 5% across all wall materials, while the combined filter method results in a false positive rate exceeding 10%, as shown in Figure 12(b).

Impact of ambient light. We also evaluate the robustness of NIRF under different ambient light conditions: *(1) Darkness* (light intensity: ≈5 lux), *(2) Low light* (≈150 lux), *(3) Medium light* (≈550 lux), and *(4) Bright light* (on a sunny day, ≈2800 lux). The evaluation results are shown in Figure 13. Our NIRF can successfully detect the behind-screen hidden camera with almost a 100% detection rate (Figure 13(a)) regardless of the ambient light conditions, and the false positive rate is below 3% (Figure 13(b)). This is mainly because NIRF exploits infrared light for camera detection, which is resilient to the changes in ambient light. As a comparison, the combined filter shows unstable filtering performance, leading to degraded detection under certain ambient lighting conditions, and the false positive rate can reach up to 20%.

Impact of screen protector. We further evaluate the robustness performance of NIRF when a screen uses different

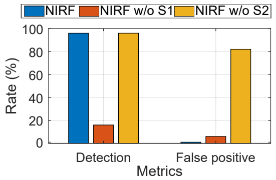
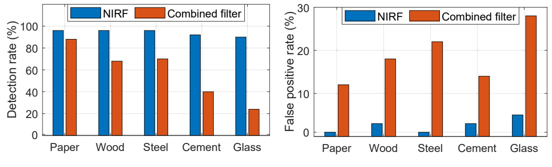
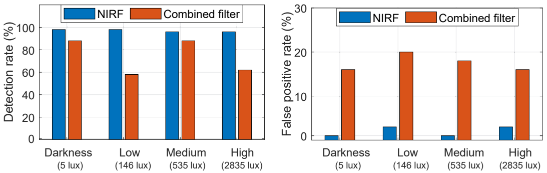


Figure 11: Ablation study.



(a) Detection rate (b) False positive rate

Figure 12: Performance vs. background differences.



(a) Detection rate (b) False positive rate

Figure 13: Performance vs. ambient light conditions.

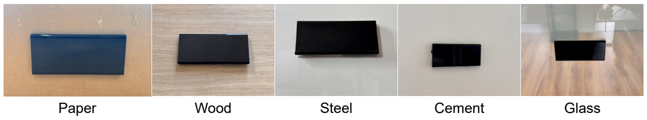
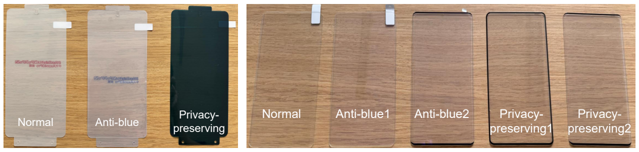


Figure 14: Settings of screen installation backgrounds.

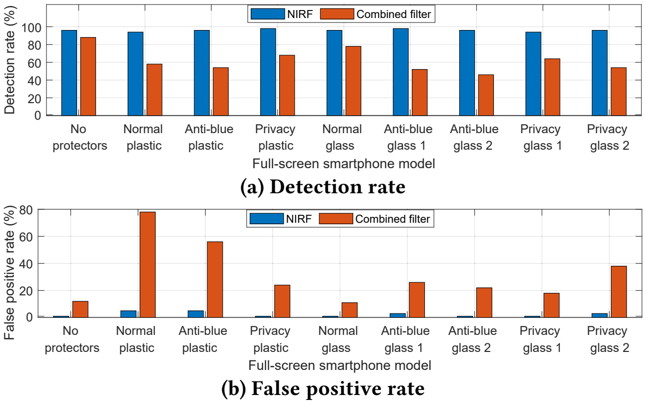


(a) Plastic protectors (b) Glass protectors

Figure 15: Settings of different screen protectors.

protectors, as shown in Figure 15. We consider three representative types of screen protectors: *(1) Normal*; *(2) Anti-blue*, which reduces the blue light to protect users' eyes; and *(3) Privacy-preserving*, which obscures the screen's displayed contents when viewed from angles exceeding a certain field of view. We also consider screen protectors made from different materials – *plastic and glass* – as they have different light reflection properties. The evaluation results are given in Figure 16. We can easily observe that despite the changes in the screen's reflection patterns caused by different screen protectors, our NIRF can reliably detect the behind-screen hidden camera with high detection rates and low false positive rates. However, the compared combined filter method struggles with interference caused by various reflection patterns introduced by the screen protector, leading to much lower detection rates and much higher false positive rates. Meanwhile, the unstable performance arises because filter-based methods cannot consistently isolate camera retroreflection from different screen reflection patterns. High-frequency noise at the screen reflection's wide edges increases false positives, further contributing to instability.

Impact of screen content. We also evaluate the performance of NIRF in a very challenging scenario when the screen displays different contents. We consider five scenarios, as shown in Figure 17 and described below: *(1) Screen displaying videos*, and *(2) Static screen content with different colors* displayed on the screen where the camera is hidden behind. The screen's brightness is set to 100% to maximize the potential interference. The evaluation results are shown in Figure 18, demonstrating the superior robust performance of NIRF over the combined filter method in detecting hidden behind-screen cameras under different screen contents.



(a) Detection rate

(b) False positive rate

Figure 16: Performance under different screen protectors.

7 Field Study: Evaluation with a 65-inch Translucent Screen

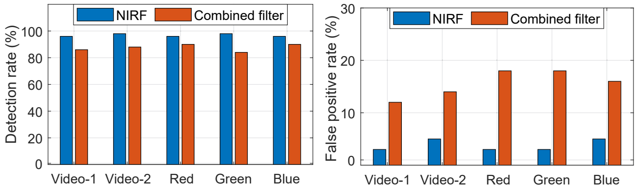
To further evaluate NIRF's performance with real-world settings, we recruit 27 participants to conduct a field study using a 65-inch translucent screen and 18 commodity spy cameras.

7.1 Evaluation Setup

Used cameras and placement. We select the most popular 18 spy cameras that are available on Amazon. These cameras are widely used in surveillance systems [18], popular products [23], or DIY spy cameras [21]. These selected cameras are shown in Figure 19; they have different attributes such as optical properties, connectivity, and data storage capacity. Specifically, ① and ② are wireless cameras, while other cameras only have wired connections. In the evaluation, we randomly hide these cameras *behind* a 65-inch translucent screen built by VideowindoW [57] that has great potential to be used as TV, conference monitors, and displays in smart home applications [53]. The placement of the 18 spy cameras behind the translucent screen is shown in Figure 20. With each camera hidden behind the screen, human eyes cannot locate them when standing before the screen.

Baselines: Commercial Off-The-Shelf (COTS) camera detectors. We choose three COTS detectors as the baselines: a K18 detector, a T1 detector, and an X17 detector, as shown in Figure 20. They all are equipped with red LEDs to perform camera detection (referring as "lighting mode"). In addition, T1 and K18 have one more antenna and feature a Received Signal Strength Indicator (RSSI) mode. Detectors in the lighting mode detect reflections from cameras; in RSSI mode,

Figure 17: Settings of screen contents: (a-b) Displaying Video-1 and Video-2; (c-e) Static Red, Green, and Blue.



(a) Detection rate

(b) False positive rate

Figure 18: Performance under different screen contents.

detectors keep beeping when the RSSI exceeds a certain threshold at a specific distance from the cameras.

Pipeline of the Field Study. We conduct a field study to assess the effectiveness of our NIRF and the three COTS hidden camera detectors. We recruited 27 participants from our university (16 females and 11 males, aged between 22 and 31) through our university mailing lists, social networks, and advertising boards in our department building[3]. The field study was conducted in a laboratory setting. Participants visited the laboratory either during the daytime or at night. We controlled the amount of natural light by opening or closing windows and adjusted the intensity of artificial ceiling lighting based on each participant's preference during the field study. After the participants are trained on how to use these detectors and our NIRF, they are asked to find the spy cameras hidden behind the screen. Emulating the typical process of detecting hidden cameras in unfamiliar areas such as hotel rooms, the experiment is divided into five stages:

Stage 1: Once entering the area, participants are given 5 min to inspect the entire screen with their naked eyes and asked if they believe any spy cameras are hidden behind. If they do, they point out the suspected camera locations. All cameras are active for half the time and inactive for the other half.

Stage 2: We deactivate all 18 spy cameras. Participants are taught how to use the detectors for the task of *"Find suspicious hidden cameras behind the 65-inch translucent TV within 15 min."* They are unaware of the number of hidden spy cameras placed in the room. The experiment begins once the participant is familiar with the detection process. Participants use the X17, T1, and K18 detectors in order of increasing functional complexity, with a maximum of 5 min per device. After using each detector, they report if they detect hidden cameras behind the screen; if so, annotate the cameras' locations.

Stage 3: We activate cameras ③-⑱ and alternately activate the wireless cameras ① and ②, ensuring each group (camera ① active, camera ② active, both cameras active) appears nine times throughout the entire experiment. Each participant reenters the room and uses the detectors to inspect the screen for 15 minutes. After using each detector, they report any detected hidden spy cameras and annotate their locations.

Stage 4: With all 18 cameras either activated or deactivated, participants use our designed NIRF to build a neural reflectance field to detect the hidden-behind-screen cameras.

Stage 5: Participants finally describe their use experience and rate the utility of the detectors and our NIRF. All the recordings of the behind-screen cameras are permanently deleted after the experiment.

7.2 Evaluation Results

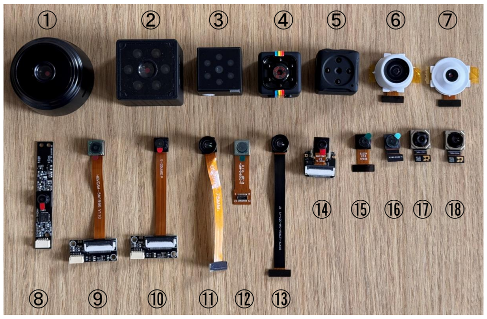
The evaluation metrics we use include the detection rate, calculated as the average detection rate, the number of correct detections for each camera, false positives, and utility.

Average detection rate. The results are shown in Figure 21. When using only the naked eyes, the cameras hidden behind the screen cannot be detected by all participants, regardless of whether the cameras are ON or OFF. When the cameras are OFF, the average detection rates of X17, T1, and K18 are extremely low, with K18 achieving the highest rate at 9.7%. This low detection rate is due to the screen's strong reflective characteristics and low light transmittance, obscuring the weak reflections from the cameras behind the screen when using the COTS detectors. When the cameras are ON, the average detection rate of T1 increases from 0 to 6%, while K18's rate slightly improves to 14.4%. Despite this increase, the overall average detection rate remains low, primarily because only cameras ① and ② have wireless connections.

We also observe an interesting phenomenon: *the screen significantly attenuates the wireless signals, making it difficult for the detectors to detect changes in the RSSI even when the antenna touches the front of the screen.* Despite the screen being only one centimeter thick, this short distance is enough for the screen to attenuate the wireless signals significantly. However, when the detectors are moved behind the screen and closer to cameras ① or ②, it continuously beeps, making these cameras easier to locate. As a comparison, our NIRF effectively detects spy cameras regardless of their connectivity type and ON/OFF states. We further illustrate the correct detections for each hidden spy camera in Figure 22. While the camera parameters and behind-screen positions do influence NIRF's performance, the overall average detection rate is 89.5%, which is substantially higher than those of the COTS detectors and naked-eye observation.

We further explore the relationship between the number of activated wireless cameras and the average detection rate. The evaluation results are shown in Figure 23. When only one wireless camera is activated, the average detection rates of K18 and T1 stay below 20%, though K18 performs better due to its higher sensitivity. When both wireless cameras

[3] The user study is approved by our Institutional Review Board (IRB).



Spy camera ID	①	②	③	④	⑤	⑥	⑦	⑧	⑨
Resolution	2MP	2MP	12MP	12MP	2MP	2MP	2MP	2MP	5MP
Lens diameter (mm)	10	9	4	3	1	12	3	1	1
Field of view (°)	140	150	140	150	NA	160	65	63	60
Connectivity	Wireless	Wireless	Wire	Wire	Wire	Wire	Wire	Wire	Wire
Storage (TF card)	128G	128G	32G	32G	32G	64G	64G	64G	64G
Spy camera ID	⑩	⑪	⑫	⑬	⑭	⑮	⑯	⑰	⑱
Resolution	5MP	2MP	5MP	5MP	2MP	2MP	8MP	12MP	12MP
Lens diameter (mm)	1	10	1	9	1	1	1	3	5
Field of view (°)	60	160	63	160	78	78	62	75	120
Connectivity	Wire	Wire	Wire	Wire	Wire	Wire	Wire	Wire	Wire
Storage (TF card)	64G	64G	64G	64G	64G	64G	64G	64G	64G

Figure 19: The 18 off-the-shelf spy cameras used in the experiments: (left) cameras; (right) their parameters.

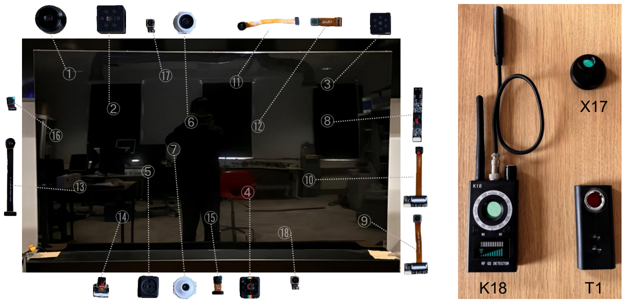


Figure 20: Setup for the user study: (left) locations *(front view)* of the spy cameras placed behind the translucent TV; (right) baseline off-the-shelf camera detectors.

① and ② are activated simultaneously, the increased signal activity further boosts the average detection rate, with K18's detection rate rising to 66.7%. Notably, our NIRF is not affected by the number of the activated cameras, consistently achieving an average detection rate above 80%.

False positive. The results are shown in Table 1. Based on our post-experiment analysis, false positives observed by naked eyes mainly come from dust on the screen and the reflection of the screen to the indoor environment, which leads to misjudgment of some areas of the screen as the location of hidden cameras. For COTS detectors, false positives mainly come from the RSSI mode of T1 and K18 to perceive wireless signals over the air. K18 has a higher adjustable range of signal sensitivity, which produces more false positives. For example, K18 keeps beeping when close to some regular objects due to the nearby electronics like the participant's own smartphone. Most participants said they could sense the presence of the camera using K18 but could not determine its exact location and could only make a rough guess. Such coarse-grained indicators lead to more false positives and downgrade the utility.

Utility. After the experiments, we also survey the participants' use experience using a *5-point Likert scale (1: not helpful at all; 5: absolutely helpful).* The X17 detector has an overall score of 1.21, T1 has an overall score of 1.78, and K18 has an overall score of 3.04; our NIRF scores 4.85. This shows NIRF is 'helpful' in detecting behind-screen hidden cameras.

Table 1: The False Positive (FP) of different methods.

Camera state	Camera OFF					Camera ON				
Method	Eyes	X17	T1	K18	NIRF	Eyes	X17	T1	K18	NIRF
No. of FP	4	3	0	16	2	4	0	9	24	2

8 Discussions

Transferability. Our system works well when NIRF is trained on different smartphones, environments, and backgrounds, as shown in our evaluation. This is because the SAM model generates screen masks that focus solely on the screen, thereby eliminating the influence of the broader environment and background. Furthermore, transferring a single NIRF across different scenes remains a substantial challenge. Potential solutions include pre-training on large, scene-diverse datasets, employing incremental learning to dynamically adapt to environmental variations, and utilizing transfer learning to improve model reusability across different environments. We will explore these directions in our future work.

Curved/Larger screens. Curved screens may further complicate the reflection pattern of the screen, and our solution may need some adjustments to adapt. One potential solution might be to optimize the reflectance field that NIRF needs to learn. For potentially larger screens (e.g., much larger than 65-inches) in the future, we may need other large-scene modeling techniques that split the screen into sub-regions and use distinct subnetworks to model and composite them, such as Block-NeRF [50]. Since these screens are not widely available, we leave them for future research.

Lightweight deployment. NIRF can be deployed on mobile devices by utilizing their ToF sensors. The phase inputs for NIRF are indirectly derived from depth and intensity maps, which can be obtained through dedicated ToF APIs, such as the ARCore Depth API [67] on Google-certified models (e.g., the Samsung S20 Ultra), or through the Camera2 API on Android smartphones. In addition, many lightweight RGB-based NeRF models have been launched recently that could be directly deployed on mobile devices [10, 38]. One typical example is Instant-NGP [38] that can train an RGB-based NeRF in five seconds. These advancements will further expand and facilitate NIRF applications on mobile devices.

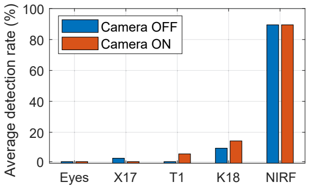


Figure 21: Detection rate vs. different detectors.

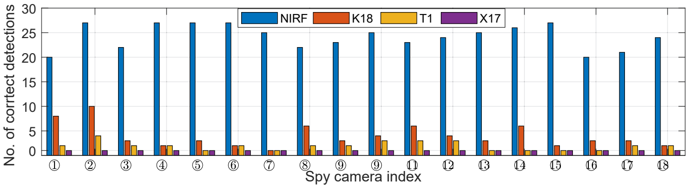


Figure 22: Number of correct detections of each behind-screen hidden camera when the cameras are ON.

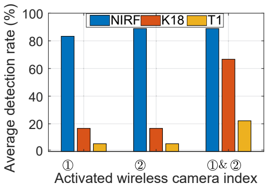


Figure 23: Detection rate vs. activated wireless cameras.

9 Related Work

Detection via RF signals relies on wireless spy cameras transmitting information via WiFi and exposing their wireless traffic patterns [11, 19, 27, 31, 46, 49]. However, this method often fails to localize spy cameras and has a high false positive rate, frequently triggered by surrounding IoT devices or WiFi routers. It is fundamentally limited to detecting only wireless spy cameras. Notably, a survey found that a large number of spy cameras on the market do not support WiFi connectivity, instead saving videos locally [66]. Meanwhile, for cameras hiding behind a screen, the vast space behind the screen provides extremely convenient conditions for the wired connection of the hidden spy camera and the use of large TF card storage. Another approach involves detecting electromagnetic leakage signals from hidden spy cameras. CamRadar [34] detects electromagnetic leakage from a camera's clock signal. However, this approach requires the camera to be active to emit leaked signals, and the presence of a screen significantly attenuates them. As discussed in Section 7, even when a COTS camera detector is placed directly in front of the screen, it could fail to detect the signals emitted from wireless cameras.

Detection via thermal emission. Thermal cameras have been used to detect hidden spy cameras by capturing their thermal emissions [20, 66]. HeatDeCam [66] scans the area to identify these thermal features. However, this method also requires the camera to be active to have thermal emission. In practice, hackers can remotely control hidden cameras, either through wired connections (such as via the control line of a TV or monitor) or by using a timing function (e.g., activating the camera a set number of minutes after the user turns on the room lights or TV). This strategy prevents users from detecting hidden cameras when they initially inspect their environment, such as once entering a hotel room. By activating the camera only after the user has checked the unfamiliar environment and become less cautious, the risk of exposure to hidden cameras is significantly reduced.

Detection via optical reflections focuses on spotting reflections from hidden camera lenses [47, 56]. LAPD [47] used ToF sensors to detect camera lenses by identifying high-intensity reflections. However, our work deals with translucent screens and cameras hidden behind it, which pose new challenges. LAPD's combined filter struggles with the complex reflection patterns of screen covers, complicating the detection. Our approach leverages the unique reflective properties of the screen and camera behind it, using ray tracing based on a neural infrared reflectance field to detect hidden cameras behind the screen effectively.

Through-screen computing was proposed in [64]. This concept refers to the processing of light signals for various computing purposes such as communication, sensing, and imaging, where light originates from the physical world, passes through the translucent screen, and reaches the behind-screen optical sensors. Among these applications, through-screen visible light communication and sensing systems were proposed to overcome challenges and improve light-based connectivity [30, 63, 65]. Meanwhile, screen perturbation was also proposed for future devices equipped with translucent screens to enhance visual privacy protection [62]. However, NIRF focuses on a new branch of through-screen computing, aiming to address the risk of behind-screen cameras potentially functioning as hidden surveillance cameras.

10 Conclusion

In this work, we studied how to detect the cameras hiding behind translucent screens. By analyzing the reflection responses from both the screen and the behind-screen hidden cameras in physical environments, we constructed Neural Infrared Reflectance Fields (NIRF) to capture their transmittance and reflectance properties. Building upon this, we designed and implemented a NIRF-based detection method to identify the hidden cameras behind screens, testing with five smartphones and a 65-inch translucent screen. Our method achieved high detection rates across different conditions. We believe NIRF can raise more awareness of behind-screen cameras and stimulate many follow-up studies to detect them.

Acknowledgments

We would like to thank M. Zuniga for his support regarding the 65-inch translucent screen, as well as the shepherd and anonymous reviewers for their valuable suggestions. This work was supported in part by SURF Research Cloud grants EINF-5830 and EINF-9562.

References

[1] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. 2019. *Real-time rendering.* AK Peters/crc Press.

[2] Amazon. 2024. Begonia Hidden Camera Detector Wants Detector RF Wireless Portable Bed Mitt Detector RF Wireless Infrared Detector Strong Anti-Spy GPS Tracker Detector for Home Office Hotel. (2024).

[3] Amazon. 2024. Camera Detector, Mini USB Charging Spy Camera Finder Device with Large Receiving Frequency, For Hotel, Bathroom, Office(T1). (2024).

[4] Amazon. 2024. Wantsen RF Wireless Detector Hidden Camera Detector for GSM Tracker, Eavesdropping Devices, Wireless Cameras, Bed Bug Finder, Infrared Detector, Strong Magnetic Detector. (2024).

[5] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. 2021. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS).*

[6] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR).*

[7] BBC. 2020. Stacey Dooley Investigates: 'My daughter was tormented by spycam sex crime'. *https://www.bbc.co.uk/bbcthree/article/63de169c-dfbb-4b22-9da3-b5f3a55628a2* (2020).

[8] Pierre Boher, Thierry Leroux, Thibault Bignon, and Véronique Collomb-Patton. 2016. Multispectral BRDF measurements on anisotropic samples: application to metallic surfaces and OLED displays. *Electronic Imaging* (2016).

[9] Pierre Boher, Thierry Leroux, Thomas Muller, and Philippe Porral. 2019. Accurate physico-realistic ray tracing simulation of displays. *Electronic Imaging* (2019).

[10] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. 2023. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR).*

[11] Yushi Cheng, Xiaoyu Ji, Tianyang Lu, and Wenyuan Xu. 2018. Dewicam: Detecting hidden wireless cameras via smartphones. In *Proceedings of the ACM Asia Conference on Computer and Communications Cecurity (ASIACCS).*

[12] CNN. 2024. 'So creepy': Inside CNN's investigation of Airbnb's hidden camera problem. *https://edition.cnn.com/2024/07/09/business/video/airbnb-hidden-cameras-investigation-digvid* (2024).

[13] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. 2022. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR).*

[14] Statista Research Department. 2022. Number of spycam related crimes in South Korea from 2011 to 2021. https://www.statista.com/statistics/1133121/south-korea-number-of-spycam-crimes/.

[15] Foscam. 2024. How to set up schedule recordings for cameras that were added to a Foscam NVR? *https://www.foscam.com/faqs/view.html?id=76* (2024).

[16] Yang Fu, Ishan Misra, and Xiaolong Wang. 2023. MonoNeRF: Learning generalizable NeRFs from monocular videos without camera poses. In *Proceedings of International Conference on Machine Learning (ICML).*

[17] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR).*

[18] hackster.io. 2021. ESP32-CAM Video Surveillance Smart Camera. *https://www.hackster.io/KDPA/esp32-cam-video-surveillance-smart-camera-7f9a63* (2021).

[19] Yan He, Qiuye He, Song Fang, and Yao Liu. 2021. Motioncompass: pinpointing wireless camera via motion-activated traffic. In *Proceedings of the ACM Mobile Systems, Applications, and Services (MobiSys).*

[20] InfiRay. 2024. Can a thermal camera detect hidden cameras? *https://www.infiray.com/can-a-thermal-camera-detect-hidden-cameras.html* (2024).

[21] AUTODESK instructable. 2020. A Super Easy Security Camera With the ESP32 CAM. *https://www.hackster.io/KDPA/esp32-cam-video-surveillance-smart-camera-7f9a63* (2020).

[22] Ajay Jain, Matthew Tancik, and Pieter Abbeel. 2021. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR).*

[23] KUUS. 2024. Buy Mini Spy Camera? Smallest Hidden Spy Cam. *https://kuus.shop/en/mini-spy-camera-eng/?srsltid=AfmBOooH60gZ598tA4WSCw8O3AU3WoSRjQQQTgBwHdv8kSjsVwUKu1h4* (2024).

[24] LankeeM9. 2020. Cameras under screens are the biggest threat to privacy right now. *https://www.reddit.com/r/privacy/comments/gd0sev/cameras_under_screens_are_the_biggest_threat_to/* (2020).

[25] PA Legal. 2021. The Effects of Under-Display Cameras on Privacy. *https://thepalaw.com/technology-law/under-display-cameras-and-what-they-mean-for-privacy/* (2021).

[26] LetsGoDigital. 2021. Samsung Under Panel Camera for smartphones and TVs. *https://nl.letsgodigital.org/smartphones/samsung-under-panel-camera/* (2021).

[27] Zhijing Li, Zhujun Xiao, Yanzi Zhu, Irene Pattarachanyakul, Ben Y Zhao, and Haitao Zheng. 2018. Adversarial localization against wireless cameras. In *Proceedings of the ACM MobiSys Workshop.*

[28] Courtney Linder. 2020. There Might Be Secret Surveillance Equipment in Your Vacation Rental. https://www.popularmechanics.com/technology/security/a32983077/vacation-rental-surveillance-spying/.

[29] Chun Liu, Changming Zhao, Haiyang Zhang, Zilong Zhang, Shuyuan Gao, and Yunshi Wang. 2019. Analysis of Mini-Camera's Cat-Eye Retro-Reflection for Characterization of Diffraction Rings and Arrayed Spots. *IEEE Photonics Journal* (2019).

[30] Hao Liu, Hanting Ye, Jie Yang, and Qing Wang. 2021. Through-Screen Visible Light Sensing Empowered by Embedded Deep Learning. In *Proceedings of the ACM SenSys Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT).*

[31] Tian Liu, Ziyu Liu, Jun Huang, Rui Tan, and Zhen Tan. 2018. Detecting wireless spy cameras via stimulating and probing. In *Proceedings of the ACM Mobile Systems, Applications, and Services (MobiSys).*

[32] Xinyang Liu, Yijin Li, Yanbin Teng, Hujun Bao, Guofeng Zhang, Yinda Zhang, and Zhaopeng Cui. 2023. Multi-modal neural radiance field for monocular dense slam with a light-weight tof sensor. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

[33] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. 2023. Robust dynamic radiance fields. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR).*

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

[35] Ezio Malis and Manuel Vargas Villanueva. 2007. Deeper understanding of the homography decomposition for vision-based control. (2007).

[36] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR).*

[37] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing

scenes as neural radiance fields for view synthesis. *Commun. ACM* (2021).

[38] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)* (2022).

[39] Notebookcheck. 2022. Review of ZTE Axon 40 Ultra smartphone - invisible selfie camera, three 64MP cameras and flagship performance. *https://www.notebookcheck.net/Review-of-ZTE-Axon-40-Ultra-smartphone-invisible-selfie-camera-three-64MP-cameras-and-flagship-performance.638777.0.html* (2022).

[40] The Voice of America. 2019. 1,600 Hotel Guests Secretly Filmed, Livestreamed in S. Korea. *https://www.voanews.com/a/hotel-guests-secretly-filmed-live-streamed-in-s-korea/4840662.html* (2019).

[41] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*.

[42] PhotoBuff. 2022. This selfie camera is completely INVISIBLE. *https://www.facebook.com/phonebuff/videos/this-selfie-camera-is-completely-invisible/1458256984607823/* (2022).

[43] pieye. 2023. Pieye Nimbus 3D ToF Camera Module. *https://pieye.org/* (2023).

[44] South China Morning Post. 2019. As South Korea confronts 'spycam porn' epidemic, pressure mounts on shops that sell hidden devices. *https://www.scmp.com/news/asia/east-asia/article/3003613/south-korea-confronts-spycam-porn-epidemic-pressure-mounts* (2019).

[45] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714* (2024). https://arxiv.org/abs/2408.00714

[46] Muhammad Salman, Nguyen Dao, Uichin Lee, and Youngtae Noh. 2022. CSI: DeSpy: enabling effortless spy camera detection via passive sensing of user activities and bitrate variations. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*.

[47] Sriram Sami, Sean Rui Xiang Tan, Bangjie Sun, and Jun Han. 2021. LAPD: Hidden Spy Camera Detection using Smartphone Time-of-Flight Sensors. In *Proceedings of the ACM Embedded Networked Sensor Systems (SenSys)*.

[48] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*.

[49] Akash Deep Singh, Luis Garcia, Joseph Noor, and Mani Srivastava. 2021. I always feel like somebody's sensing me! A framework to detect, identify, and localize clandestine wireless sensors. In *Proceedings of the USENIX Security Symposium (USENIX Security)*.

[50] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. 2022. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*.

[51] TechDaily. 2022. ZTE Axon 40 Ultra Unboxing - The Best Hidden Selfie Camera Yet! *https://www.youtube.com/watch?v=cX2CLEy7xLs* (2022).

[52] Time. 2019. Australian Police Charge Tourist Over Spycam Case at a Bondi Beach Hostel. *https://time.com/5558651/hong-kong-tourist-hidden-camera-australia-hostel/* (2019).

[53] Tech Times. 2022. CES 2022 The 30 Best and Coolest Pieces of Tech Showcased in Las Vegas. *https://www.techtimes.com/articles/270506/20220113/ces-2022-smart-home-technology-smart-home-devices-agritech-products-agritechnology-autonomous-vehicles-smart-glasses-ar-vr-headset.htm* (2022).

[54] The Straits Times. 2019. South Korea spycam crimes put hidden camera industry under scrutiny. *https://www.straitstimes.com/asia/east-asia/south-korea-spycam-crimes-put-hidden-camera-industry-under-scrutiny* (2019).

[55] Digital Trends. 2025. Bezel-Free Beauty: Lenovo's Yoga Slim 9i Hides a Camera Under the Display. *https://www.youtube.com/watch?v=q3OVs3W0JwM* (2025).

[56] Khai N Truong, Shwetak N Patel, Jay W Summet, and Gregory D Abowd. 2005. Preventing camera recording by designing a capture-resistant environment. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*.

[57] VideowindoW. 2024. transforming windows into screens. *https://www.videowindow.eu/* (2024).

[58] Zhibin Wang, Yilu Chang, Qi Wang, Yingjie Zhang, Jacky Qiu, and Michael Helander. 2020. 55-1: Invited Paper: Self-Assembled Cathode Patterning in AMOLED for Under-Display Camera. In *SID Symposium Digest of Technical Papers*.

[59] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. 2021. NeRF−: Neural Radiance Fields Without Known Camera Parameters. *arXiv preprint arXiv:2102.07064* (2021).

[60] Zhiyuan Xie, Xiaomin Ouyang, Xiaoming Liu, and Guoliang Xing. 2021. UltraDepth: Exposing high-resolution texture from depth cameras. In *Proceedings of the ACM Embedded Networked Sensor Systems (SenSys)*.

[61] Zhiyuan Xie, Xiaomin Ouyang, Li Pan, Wenrui Lu, Guoliang Xing, and Xiaoming Liu. 2023. Mozart: A Mobile ToF System for Sensing in the Dark through Phase Manipulation. In *Proceedings of the ACM Mobile Systems, Applications, and Services (MobiSys)*.

[62] Hanting Ye, Guohao Lan, Jinyuan Jia, and Qing Wang. 2023. Screen Perturbation: Adversarial Attack and Defense on Under-Screen Camera. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*.

[63] Hanting Ye and Qing Wang. 2021. SpiderWeb: Enabling Through-Screen Visible Light Communication. In *Proceedings of the ACM Embedded Networked Sensor Systems (SenSys)*.

[64] Hanting Ye and Qing Wang. 2024. Vision Paper: Computing behind Transparent Screen. In *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN)*.

[65] Hanting Ye, Jie Xiong, and Qing Wang. 2023. When VLC Meets Under-Screen Camera. In *Proceedings of the ACM Mobile Systems, Applications, and Services (MobiSys)*.

[66] Zhiyuan Yu, Zhuohang Li, Yuanhaur Chang, Skylar Fong, Jian Liu, and Ning Zhang. 2022. HeatDeCam: detecting hidden spy cameras via thermal emissions. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*.

[67] Yunfan Zhang, Tim Scargill, Ashutosh Vaishnav, Gopika Premsankar, Mario Di Francesco, and Maria Gorlatova. 2022. Indepth: Real-time depth inpainting for mobile augmented reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* (2022).

[68] Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. 2023. Nerf2: Neural radio-frequency radiance fields. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*.