Propagate And Calibrate: Real-time Passive Non-line-of-sight Tracking

Yihao Wang[1*] Zhigang Wang[1*] Bin Zhao[1,2†] Dong Wang[1] Mulin Chen[1,2] Xuelong Li[1,2†]
[1]Shanghai AI Laboratory [2]Northwestern Polytechnical University
{wangyihao,wangzhigang,zhaobin,wangdong,chenmulin}@pjlab.org.cn li@nwpu.edu.cn

Abstract

Non-line-of-sight (NLOS) tracking has drawn increasing attention in recent years, due to its ability to detect object motion out of sight. Most previous works on NLOS tracking rely on active illumination, e.g.*, laser, and suffer from high cost and elaborate experimental conditions. Besides, these techniques are still far from practical application due to oversimplified settings. In contrast, we propose a purely passive method to track a person walking in an invisible room by only observing a relay wall, which is more in line with real application scenarios,* e.g.*, security. To excavate imperceptible changes in videos of the relay wall, we introduce difference frames as an essential carrier of temporal-local motion messages. In addition, we propose PAC-Net, which consists of alternating propagation and calibration, making it capable of leveraging both dynamic and static messages on a frame-level granularity. To evaluate the proposed method, we build and publish the first dynamic passive NLOS tracking dataset, NLOS-Track, which fills the vacuum of realistic NLOS datasets. NLOS-Track contains thousands of NLOS video clips and corresponding trajectories. Both real-shot and synthetic data are included. Our codes and dataset will be available on GitHub soon.*

1. Introduction

In contrast to conventional imaging within the direct line-of-sight (LOS), non-line-of-sight (NLOS) imaging aims to tackle an inverse problem, *i.e.*, using indirect signal (*e.g.*, reflection from a visible relay wall) to recover information of invisible areas. To specify, NLOS tracking manages to reconstruct a continuous trajectory in real time when an object or a person is moving in an invisible region, which is sketched in Fig. 1. The ability to track moving objects outside the LOS would enable promising applications, such as autonomous driving, robotic vision, security, medical imaging, post-disaster searching, and rescue operations, *etc*. [2, 14, 17, 27], thus receiving increasing attention in recent years.

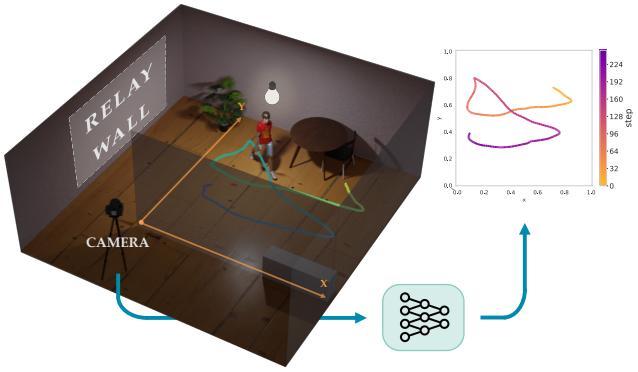


Figure 1. **A schematic of the passive NLOS tracking.** The character is walking in the hidden scene and we can perform real-time tracking by observing and analyzing the relay wall from outside the room with a RGB camera, without any additional equipment.

Existing NLOS tracking techniques mostly rely on active illumination from the detection side [4, 5, 7, 8, 16, 23, 25, 28, 35, 36]. Although introducing denser and finer information, active illumination typically requires expensive equipment (*e.g.*, ultra-fast pulsed laser) and elaborate experimental conditions [29]. These defects cause a gap between active techniques and practical applications. Besides, the oversimplified setting in previous works even expands the gap. Unlike active methods, passive NLOS techniques [1, 3, 5, 18, 19, 29–31, 33, 39, 41] only depend on the feeble diffuse reflection of the hidden region, getting rid of requirements of expensive equipment. So this paper focuses on the low-cost passive NLOS tracking task in realistic scenarios.

We find that most existing NLOS tracking works merely locate the object in each frame independently [4, 5, 7, 8, 19, 23, 28, 36], without considering the position relationship between adjoining moments. This practice directly causes jitters of trajectory, thus resulting in inaccurate tracking (see Sec. 5.3 for more details). In this paper, we consider the significance of making use of motion information and taking advantage of motion continuity prior, which helps achieve more coherent and accurate tracking results.

Furthermore, passive NLOS techniques face the dilemma that the signal-to-noise ratio (SNR) is extremely low [6, 8]. To address this problem, some previous works

*Equal contribution.
†Corresponding author.

conduct background estimation with video's temporal mean and apply background subtraction to every frame [3, 19, 33]. In this way, the difference between frames could be amplified, thus increasing the SNR. However, temporal-mean subtraction inevitably mixes up information from early period. Consequently, it reintroduces extra noise into originally low-SNR signals, which is still a hazard to excavating faint differences between frames.

To address the aforementioned problems, we first introduce *difference frame* to describe motion information. Compared to background estimation and subtraction, a difference frame can be readily obtained by subtracting the previous frame from the current frame. In this way, a difference frame can represent the immediate motion information, and will not introduce noise from other periods. Our experiments show that difference frames do convey essential dynamic messages (see Sec. 5.3 for more details). Additionally, we propose a novel network named PAC-Net (**P**ropagation **A**nd **C**alibration **Net**work), which integrates motion continuity prior into the algorithm. Consisting of two dual modules, Propagation-Cell and Calibration-Cell, PAC-Net maintains a good continuity of trajectory via propagating with difference frames and then calibrating with raw frames in an alternate manner. Our experimental results demonstrate that PAC-Net can achieve centimeter-level precision when tracking a walking person in real time.

We also build NLOS-Track, the first public-accessible video dataset for passive NLOS tracking. It contains realistic scenes to support the proposed task and method, and we expect NLOS-Track to facilitate more NLOS works. In contrast to oversimplified settings in existing NLOS tracking works, NLOS-Track dataset manages to simulate realistic scenarios with humans walking in unknown scenes. The dataset consists of 500 real-shot videos and more than 1,000 synthetic videos, each recording the relay wall when a character walks along the randomly generated trajectory. Paired trajectory ground truth of each video clip is also provided.

Our contributions are mainly in three folds:

- We propose and formulate the purely passive NLOS tracking task, which avoids the use of expensive equipment. Development on this task will allow promising and valuable applications in many fields, such as robotic vision, medical imaging, *etc*.
- We propose a passive NLOS tracking network, PAC-Net, which is capable of utilizing both dynamic and static messages on a frame level. As for dynamic messages, we specially introduce difference frames as clear carriers of motion information, which gets rid of introducing extra noise from other periods.
- We establish the first passive NLOS trajectory tracking dataset, NLOS-Track, which contains thousands of video clips with a variety of scene settings.

2. Related Work

Passive NLOS. Previous passive NLOS imaging techniques mainly focus on reconstructing static information of the invisible scene. Some works leverage pinholes or pinspecks as "accidental cameras" [11, 37] while others rely on occluders (*e.g*., blocking objects or corners). These works make use of shadows and penumbrae cast on the visible wall or floor to extract useful information about the hidden scene [1, 18, 29–31, 39, 41].

As for dynamic NLOS scenario, it was first shown by Bouman *et al*. [3] that obstructions with edges can be exploited as "corner cameras". They reveal the number and trajectories of people moving in an occluded scene with recovered 1-D spatio-temporal videos. Sharma *et al*. [33] presented a deep learning method that reveals the number or activity of people in an unknown room by observing a blank relay wall. Wang *et al*. [38] first proposed a novel method for NLOS moving target reconstruction, which uses an event camera to extract rich dynamic information of the speckle movement.

Compared to existing methods, our technique doesn't introduce any additional structures or special devices. With only a visible blank wall and a conventional RGB camera, we can extract both motion and static information on the frame level and perform tracking in real time.

Active NLOS localization and tracking. Some previous works accomplished NLOS tracking directly through locating the hidden object or person frame-by-frame [4, 5, 7, 8, 23, 28, 36], whereas other methods consider object motion to assist tracking [16, 25, 35]. All methods mentioned above rely on active illumination and most of them rely on time-resolved detection techniques. Methods employing lasers typically take advantage of high flatness of optical experimental platform. In contrast, our method removes the need for any additional illumination, equipment, and special detector.

NLOS datasets. Large-scale, labeled and readily accessible datasets are vital to technique development. However, only few NLOS works provide datasets, and most of them focus on active NLOS imaging. Jarabo *et al*. [21] proposed an effective framework for rendering in transient state, which has been exploited by several following works for data generation [24, 26, 42]. Klein *et al*. [24] released a synthetic data foundation with a few scenes and the first reconstruction benchmark platform for a variety of NLOS imaging tasks, along with task-specific quality metrics. To further expand the data scale and facilitate data-driven methods, a new benchmark dataset for time-resolved NLOS imaging, Z-NLOS, is proposed by Galindo *et al*. [15].

As for passive datasets, Chen *et al*. have presented the first large-scale static passive NLOS dataset [18]. Wang *et al*. [38] created the first event-based NLOS imaging dataset, which explores a novel modal in dynamic NLOS

imaging. In addition to the fact that only a proportion of datasets are available, the lack of realistic dynamic passive NLOS datasets also remains an obstacle to exploring passive NLOS methods. To address this issue, we propose a new dataset, NLOS-Track, which contains both synthetic data and real-shot data.

3. Problem Formulation and Signal Extraction

3.1. NLOS Tracking Problem

Passive NLOS imaging aims to excavate information about a hidden scene through the diffuse reflection of ambient light. This task can be accomplished by observing and analyzing a relay wall. Every slight change in the hidden room could induce an imperceptible variation of the reflection, thus influencing the wall image. The complicated optical system can be formulated as an imaging function $\mathcal{F}$:

$$I = \mathcal{F}(\vec{x}, \Theta), \tag{1}$$

where I denotes the photo of the relay wall, depending on the position $\vec{x}$ of a person and other scene configuration Θ. The scene configuration Θ mainly includes the appearance of the person, the material of the wall, and the illumination condition. The imaging function $\mathcal{F}$ *compresses* the light field within the hidden region and casts it onto the relay wall. According to Eq. (1), a change in the scene (*e.g.*, a person's position) will cause a change of the *shadow* on the relay wall correspondingly. Mathematically, this change can be formulated as the partial derivative of Eq. (1):

$$\frac{\partial I}{\partial \vec{x}} = \frac{\partial \mathcal{F}(\vec{x}, \Theta)}{\partial \vec{x}}. \tag{2}$$

Through the guidance of Eq. (1) and Eq. (2), it is possible to track a person out of LOS by scrutinizing the faint shadow changes. Given a series of discrete observations over time $\{I_0, ..., I_t, ...\}$, *i.e.*, raw frames of a video, NLOS tracking aims to find an inverse imaging function, $\mathcal{F}^{-1}$, which reconstructs the causes $\{x_0, ..., x_t, ...\}$, *i.e.*, the trajectory of the moving objects in real time.

3.2. Difference Frames

Most previous works neglect the motion information in tracking tasks, while it can play an important role in guiding the tracking process. To extract motion information explicitly, we can limit our vision to the vicinity of a single moment t. Along with Eq. (2), we can derive the relation between the relay wall's variation ΔI_t with the person's movement $\Delta\vec{x}_t$ in the form of finite difference:

$$\begin{aligned} \left.\frac{\Delta I}{\Delta \vec{x}}\right|_t &\approx \left.\frac{\partial \mathcal{F}(\vec{x}, \Theta)}{\partial \vec{x}}\right|_t \\ \Longrightarrow I_{t+1} - I_t = \Delta I_t &\approx \left.\frac{\partial \mathcal{F}(\vec{x}, \Theta)}{\partial \vec{x}}\right|_{\vec{x}=\vec{x}_t} \Delta\vec{x}_t \\ &= \mathcal{G}(\vec{x}_t, \Delta\vec{x}_t, \Theta), \end{aligned} \tag{3}$$

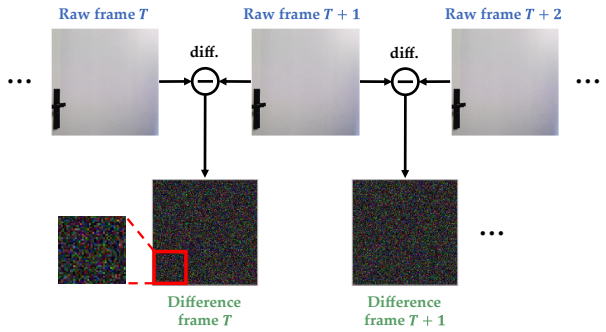


Figure 2. **Visualization of raw frames and difference frames.** The "diff." is the abbreviation for "difference", which means subtracting the previous frame from the current frame. We visualize difference frames after taking absolute value and normalizing to $[0, 1]$ for higher contrast.

where ΔI_t is the *difference frame*, which is obtained by subtracting the previous frame from the current frame in the video (Fig. 2), and $\mathcal{G}$ denotes the imaging function of difference frame. As shown in Eq. (3), the person's movement $\Delta\vec{x}_t$ directly influences the difference frame ΔI_t. Given a combination of position and motion $(\vec{x}, \Delta\vec{x})$, $\mathcal{G}$ maps the tuple to a corresponding difference frame. Therefore, through excavating difference frames, we can further leverage dynamic motion information beyond static positions, which significantly benefits the NLOS tracking task. See Sec. 5.3 for more details.

Furthermore, the temporally local nature of difference frames enables them to provide "clean" motion information. In comparison, although background frame estimation and subtraction [3, 19, 33] can increase SNR, this practice inevitably introduces information of other time to every single frame, thus making static information "dirty".

4. Tracking Method

When a person is walking in the hidden room, there is a live streaming video coming from the camera shooting at the relay wall. This raw frame stream $\{I_t\}$ incorporates a series of discrete static position information. We can readily separate the difference frame stream $\{\Delta I_t\}$ from the raw frame stream, which contains the dynamic motion information instead. To exploit both motion and position information conveyed by two streams, we propose a concise dual architecture, PAC-Net (**P**ropagation **A**nd **C**alibration **Net**work). Instead of using two-branch architectures [34] to process two streams separately, PAC-Net integrates the motion continuity prior to its workflow with a specially designed alternating recurrent architecture.

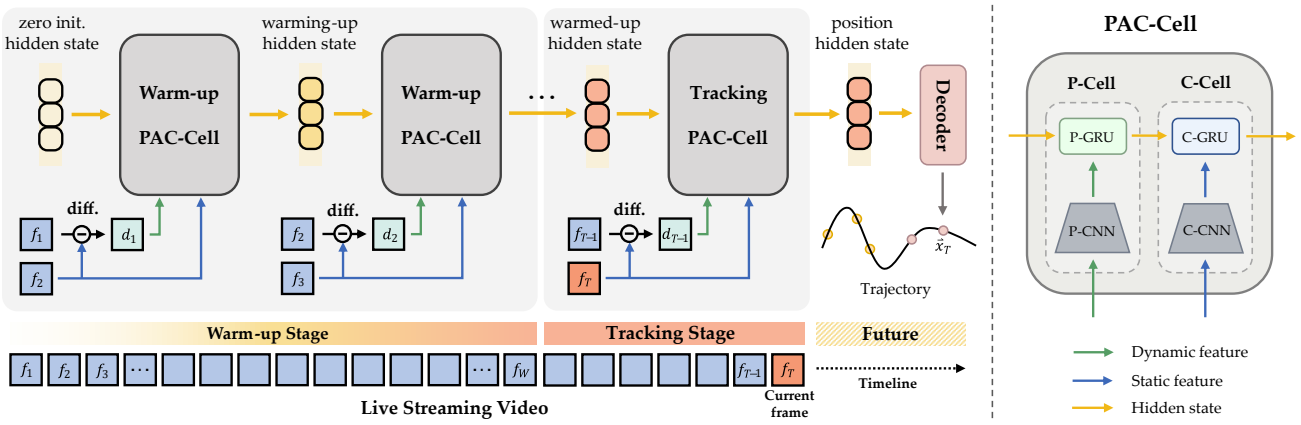


Figure 3. **Visualization of tracking pipeline with PAC-Net.** There are two stages in the whole pipeline, *Warm-up Stage* and *Tracking Stage*. Specifically, given a live streaming video, the Warm-up Stage leverages the first W-th frames to "warm up" the hidden state $\mathbf{h}$ using the Warm-up PAC-Cell. Then the Tracking PAC-Cell infers the trajectory in the Tracking Stage. Both two cells take in dynamic feature (dark green arrow) and static feature (dark blue arrow) and updates the hidden state (light yellow arrow) alternately. Note that PAC-Net can process every incoming frame in an online manner. (Best viewed in color.)

4.1. PAC-Net

The architecture of PAC-Net and the corresponding real-time tracking pipeline are illustrated in Fig. 3. The design notion of PAC-Net is to process difference and raw stream in an alternate manner, namely *propagate* and *calibrate*. Based on this, we design a dual architecture using the main component, *PAC-Cell*. Within a PAC-Cell, there are two symmetric cells, Propagation-Cell and Calibration-Cell [1]. Since difference frames convey significant dynamic motion information as shown in Eq. (3), P-Cell first *propagates* the hidden state between two observations with a difference frame ΔI_{T-1}. Then the newly incoming raw frame I_T introduces absolute position messages, allowing C-Cell to *calibrates* the hidden state to a more accurate one. The whole procedure is as below:

$$\begin{aligned} \text{Propagate}: &\quad \tilde{\mathbf{h}}_T = \text{P-Cell}\left(\mathbf{h}_{T-1}, \Delta I_{T-1}\right), \\ \text{Calibrate}: &\quad \mathbf{h}_T = \text{C-Cell}\left(\tilde{\mathbf{h}}_T, I_T\right). \end{aligned} \tag{4}$$

In this paper, we use GRU cell [10] as a recurrent cell and ResNet-18 [20] as a feature extractor, which allow real-time inference with low computational cost. The decoder in Fig. 3 is a two-layer Multilayer Perceptron (MLP). In fact, PAC-Net is a framework-level architecture for reconstruction tasks with temporally dense observations. The aforementioned components could be selected accordingly in other tasks.

At each time step T, the current position $\vec{x}_T$ can be decoded from the hidden state $\mathbf{h_T}$. So on and so forth, the trajectory of a moving subject can be tracked in real time.

[1] Hereafter referred to as P-Cell and C-Cell.

Some tracking results are demonstrated in Fig. 4 and Fig. 6. Note red squares in Fig. 4 highlight the difference between trajectories after propagation and calibration.

Note that the alternating recurrent workflow plays an indispensable role to allow the hidden state $\mathbf{h}$ to propagate stably between observations. Formally, the process of a recurrent neural network (RNN) updating the hidden state could be reformulated as the discretized first-order method for integrating ordinary differential equations (ODEs) [9, 13]:

$$\begin{aligned} \frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta) &\Longrightarrow \mathbf{h}_t = \mathbf{h}_{t-1} + f\left(\mathbf{h}_{t-1}, \theta_{t-1}\right) \\ &\Longrightarrow \mathbf{h}_t = \tilde{f}\left(\mathbf{h}_{t-1}, x_t\right), \end{aligned} \tag{5}$$

where x_t is the newly incoming observation and θ represents the parameters. It is natural to consider taking advantage of this intrinsic nature of RNNs to perform tracking. However, our experiments expose the defect of the first-order method (See Sec. 5.3 for more details) that the tracking results either maintain poor continuity or diverge over time. In contrast, the alternating workflow allows PAC-Net to integrate dynamic and static information. This behavior is similar to predictor-corrector methods for solving ODEs, which can ensure numerical convergence via alternately predicting and correcting. Such methods can be formulated as follows:

$$\begin{aligned} \tilde{\mathbf{h}}_{t,0} &= \mathbf{h}_{t-1} + f\left(\mathbf{h}_{t-1}, \theta_{t-1}\right), \\ \tilde{\mathbf{h}}_{t,n} &= \mathbf{h}_{t-1} + \tilde{f}\left(\mathbf{h}_{t-1}, \tilde{\mathbf{h}}_{t,n-1}, \theta_{t-1}\right), \end{aligned} \tag{6}$$

where $n = 1, 2, ..., N$ and N denotes the total number of corrections. After predicting with motion information of difference frames, PAC-Net performs one time of

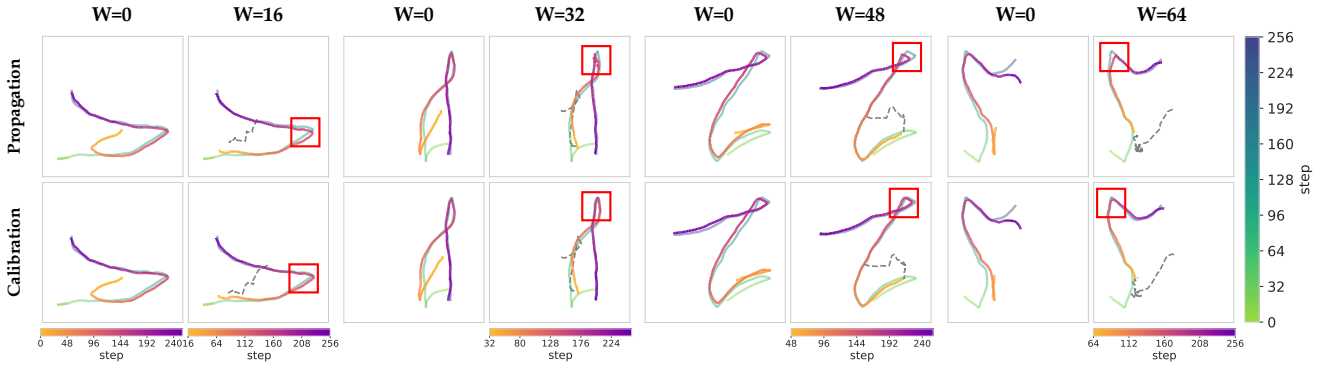


Figure 4. **Visualization of tracking results with PAC-Net on synthetic data.** Two rows indicate trajectories decoded from P-Cell and C-Cell respectively, called P-trajectory and C-trajectory. Each column indicates different warm-up steps W. Grey dashed lines represent the course of the warm-up. Two color gradients for mapping step numbers are applied to ground truths (green) and tracking results respectively. For neat visualization, trajectories are normalized with the room size so that each sub-figure is a square. All columns titled $W = 0$ share the same color bar as the first column. Red squares denote where trajectories are refined by C-Cell. (Best viewed in color.)

correction with static information of raw frames. This way, PAC-Net can perform tracking coherently and accurately without divergence.

The tracking procedure allows end-to-end training of the model via minimizing the loss with respect to ground truth trajectory. The loss function is composed of position error $loss_x$ and velocity(displacement) error $loss_v$ with an adjustment parameter α_v, controlling the weight of $loss_v$. Both of them follow a Mean-Square Error (MSE) fashion:

$$\begin{aligned} Loss &= loss_x + \alpha_v \cdot loss_v \\ &= MSE(\{\tilde{\vec{x}}_t\}, \{\vec{x}_t\}) + \alpha_v \cdot MSE(\{\Delta\tilde{\vec{x}}_t\}, \{\Delta\vec{x}_t\}), \end{aligned} \tag{7}$$

where α_v is set to 500 in all experiments to align the orders of magnitude of $loss_x$ and $loss_v$. The velocity loss ensures the model learns a reasonable representation from difference frames and accelerates the convergence of trajectory continuity. Please refer to the supplementary material for more details about model training.

4.2. Warm-up

Since we have no knowledge about the hidden scene before the video stream comes in, we apply a zero-initialization to the hidden state $\mathbf{h}$ in GRU cell. Although the principle is different, we observe a similar phenomenon as described in keyhole imaging [28] – The tracking trajectories deviate from the ground truth during early steps and gradually converge over time (visualized in columns titled $W = 0$ in Fig. 4). This is reasonable because a "well-defined" hidden state $\mathbf{h}$ doesn't come from nowhere. The convergence over time may indicate gradually gaining knowledge of the unknown room since there are various room sizes and wall materials in our dataset.

A natural idea is that we could disentangle a few early steps as *Warm-up Stage* from the original tracking procedure. Therefore, we construct two independent PAC-Cells in PAC-Net. The first one is called *Warm-up PAC-Cell*, which is responsible for "pulling" the hidden state $\mathbf{h}$ from zero initialization to a reasonable distribution, literally, "warming up" the model. This way, another PAC-Cell, *Tracking PAC-Cell*, could concentrate on accurately tracking by encoding each subsequent frame into a more accurate embedding. In Fig. 4 we visualize the course of warm-up with grey dashed lines. The Warm-up Stage could be regarded as an indirect way to find an appropriate initial hidden state $\mathbf{h}$ for the Tracking Stage. Note if there is a Warm-up Stage, *i.e.*, $W > 0$, we only use the inferred trajectory in Tracking Stage to supervise the model training. We compare different steps to perform warm-up and report the results in Sec. 5.3.

5. Experiments

In this section, we first introduce our proposed dataset, NLOS-Track, then some quality metrics for evaluating the tracking results, and finally verify the effectiveness of our proposed method with experimental studies.

5.1. NLOS-Track Dataset

Compared to existing datasets for passive NLOS imaging (Tab. 1), NLOS-Track is focused on fitting realistic dynamic scenes. Therefore, rather than including tons of static photographs or using unrealistic objects as tracking targets (*e.g.*, humanoid dolls or cardboard mannequins), NLOS-Track contains more than one thousand clips of videos that shoot at a blank relay wall while a real person is walking around the hidden room. We also render rich realistic scenes that are diverse in room size, walking character, wall material, and lighting condition.

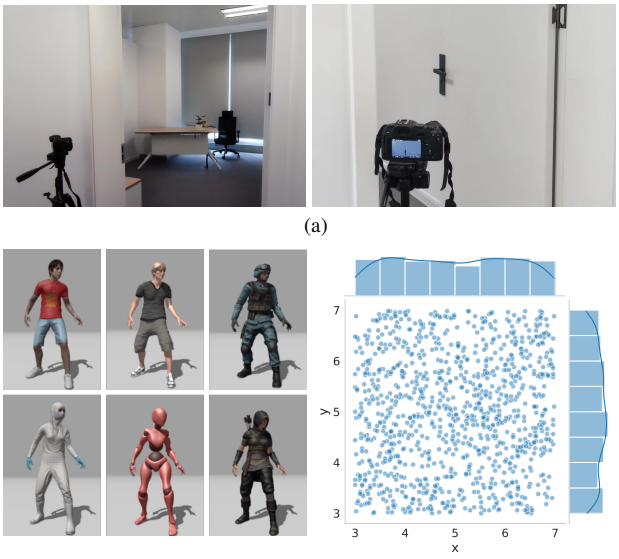


Figure 5. (a) A setup scenario with a camera observing the wall from outside the room. Left: A panorama. Right: View from the camera side. (b) Left: Part of the characters we use in synthetic data. Right: Room size (in meters) distribution and each dot represents a room.

5.1.1 Real-shot data collection

We use a consumer-grade micro SLR camera (Canon EOS RP) to capture videos of the relay wall at 25 FPS. To obtain the ground truth of the walking trajectory, we stick a USB camera (HIKVISION E14a) to the ceiling, which records the whole process of people walking from a top view at 25 FPS as well. From the top viewed videos we are allowed to use Aruco codes to locate the walking person's coordinate frame by frame at a sub-centimeter precision. Post-processing of recorded videos includes manually aligning the wall-shooting video stream and coordinate stream and cropping them into video clips of 250 frames. We ask 3 subjects (each change 3 different clothes) to walk at various speeds to generalize the dataset. Please consult supplementary materials for details about the real-shot dataset.

5.1.2 Rendering setup

In order to reduce the gap between the synthetic data and the photo-realistic data, we use the Cycles render engine in Blender [12], which is a physically-based path tracer and provides excellent performance in rendering realistic images. All 3D human-like characters and skeleton models for walking animation are acquired from Mixamo, a free animation platform of Adobe. After the character is imported and the walking animation is bound to the character[2], we use the Cycles render engine to conduct steady-state rendering of the relay wall frame by frame. All video clips have 320 frames each, at 30 FPS, and a resolution of 256×256 pixels. The videos are firstly rendered into `.png` sequences with 8-bit RGB color depth and then pre-processed into `.npy` files for IO efficiency.

On an NVIDIA A100 graphics card, the rendering speed is about 0.8 seconds per frame at a resolution of 256×256 pixels. In total, we spend about 70 hours rendering the full synthetic dataset of 1000 video clips.

[2]Refer to supplementary materials for more details about our random trajectory generation strategy.

Dataset	Modal	Data Source	Setup	Size
Platform [24]	Transient	Synthetic	Static & Dynamic	/
Z-NLOS [15]	Transient	Synthetic	Static	300 measurements
NLOS-Passive [18]	Steady	Real-shot	Static	3,200,000 images
NLOS-ES [38]	Event	Real-shot	Dynamic	4,180 images
NLOS-Track (Ours)	Steady	Real-shot & Synthetic	Dynamic	~1,500 videos (445,000 frames)

Table 1. **A brief summary of existing NLOS datasets.** Transient and Steady under the Modal column denotes time-resolved transient-state and conventional RGB steady-state respectively. "/" denotes difficulty to report the data size due to mixed data types.

5.1.3 Other considerations on generalization

We randomize several settings before rendering each clip of the video for generalization purposes. The room sizes (in meters) are sampled from a uniform distribution $U(3, 7)$, as shown in Fig. 5b. Besides, the character is randomly selected among more than 20 characters with a variety of outfits. We also change the position and luminosity of light sources and the camera position. Four different floor styles are applied randomly. The texture and roughness of the photographed wall are randomized as well. All these settings are recorded in a `.yaml` file to guarantee reproducibility.

Besides, we simulate the real-world noise in synthetic data to make them more realistic. We first compute the mean and standard deviation of real data and synthetic data at the pixel level and then aligned their statistics by adding noise to the rendering results.

5.2. Metrics

As suggested by Klein *et al*. [24], the root-mean-square (RMS) error could be used to evaluate the position inference and tracking quality by directly computing the Euclidean distance between tracking and GT trajectory. Since RMS error is positively correlated with the loss function (MSE) we use, we refer to other three metrics for a comprehensive evaluation – area between two curves (Area) [22], dynamic time warping (DTW) [32], and partial curve mapping (PCM) [40]. All three metrics are based on similarity measures between two curves. To specify, metrics reported in Tab. 2 are only computed in the Tracking Stage and are normalized by trajectory length for fairness.

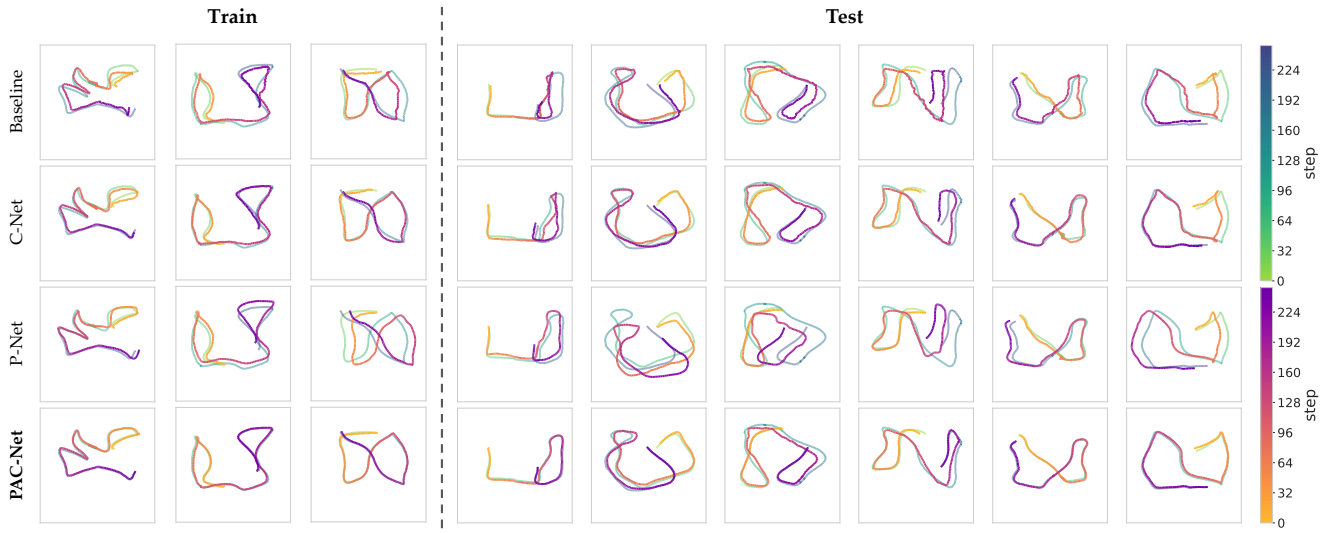


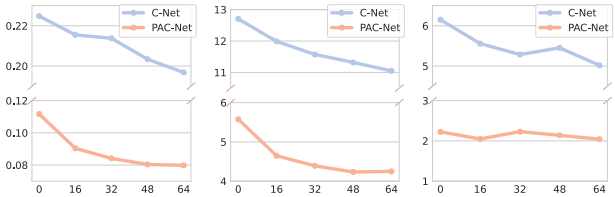
Figure 6. **Visualization of tracking results with different models on the real-shot dataset.** Rows indicate different methods and columns indicate different trajectories sampled from the Train and Test set. Color gradients share the same meaning as in Fig. 4. (Best viewed in color.)

5.3. Results

In this section, we describe the experimental results of our proposed method, along with the baseline method and ablation study. It is worth mentioning that we don't compare different choices of components (*e.g.*, various feature extractors, different RNN cells, dimension of decoder) in experiments because our goal is to demonstrate the effectiveness of PAC-Net's architecture, not to improve the performance.

PAC-Net. To verify our design notion of PAC-Net, we illustrate trajectory output by P-Cell and C-Cell separately in two rows of Fig. 4, which are called P-trajectory and C-trajectory. P-Cell first uses motion information conveyed by difference frames to step forward, which forms P-trajectory. Then under collaboration with C-Cell, P-trajectory is refined into the more accurate C-trajectory with position information introduced by raw frames. In Fig. 4 we use red squares to denote sudden changes of velocity, where C-Cell plays a clear calibrating role. Numerical metrics reported in Tab. 2 show that PAC-Net not only achieves a centimeter-level precision in Tracking Stage but also demonstrates robustness on both real-shot data and synthetic data, thus outperforming other methods.

To validate the effectiveness of our method, we construct two degenerated models by removing P-Cell or C-Cell from PAC-Cell, called C-Net and P-Net respectively. Each of the two degenerated models only takes in raw frames or difference frames. We also evaluate a CNN-based baseline model, fed with raw frames. Note we use a ResNet-34 and a two-layer GRU in degenerated models and a ResNet-50 as CNN baseline to align the parameter number.



(a) Area (↓) (b) DTW (↓) (c) PCM (↓)

Figure 7. **Comparison of different warm-up steps on synthetic data.** Both C-Net and PAC-Net have better tracking performance as warm-up steps increase with a gradual saturation.

Baseline. There are obvious jitters in reconstructed trajectories using vanilla CNN, which is shown in the first row of Fig. 6. Poor statistic metrics reported in Tab. 2 prove the necessity of taking advantage of recurrent structure to maintain the trajectory continuity.

C-Net (without dynamic motion information). Although C-Net stabilizes the trajectory to some extent via recurrent structure, as shown in the second row of Fig. 6, there are still jitters and discontinuities in trajectories. This observation indicates that it is not sufficient to only use position information in passive NLOS tracking.

P-Net (without static position information). P-Net has the same structure as C-Net, but takes as input difference frames and reconstructs displacement (velocity) between observations instead. To compute metrics using motion information independently, we provide the true initial position so P-Net can accumulate velocities into a complete trajectory. As shown in the third row of Fig. 6, P-Net is capable of maintaining better smoothness and stability than C-Net, but there is an obvious overall translation with respect to ground truth.

Datasets	Methods		Metrics				
	Type	Model	RMS$_x$($\times 10^{-2}$)	RMS$_v$($\times 10^{-3}$)	Area (↓)	DTW (↓)	PCM (↓)
Real-shot	CNN	ResNet-50	2.60	2.94	0.01184	0.8559	0.8171
	ConvRNN	C-Net	1.65	2.24	0.00644	0.4617	0.4819
		C-Net + Warm-up	1.55	2.17	0.00601	0.4371	0.4392
		P-Net	4.03	2.87	0.01137	0.9727	1.211
	Ours	PAC-Net	1.46	**1.17**	**0.004347**	0.3367	0.3313
		PAC-Net + Warm-up	**1.37**	1.28	0.004388	**0.3348**	**0.3027**
Synthetic	CNN	ResNet-50	22.0	3.90	0.2807	21.891	316.806
	ConvRNN	C-Net	15.5	3.31	0.2248	12.703	6.151
		C-Net + Warm-up	15.1	3.16	0.2138	11.572	5.286
		P-Net	11.3	2.17	0.1251	5.774	2.826
	Ours	PAC-Net	9.70	2.21	0.1117	5.576	**2.225**
		PAC-Net + Warm-up	**8.78**	**1.79**	**0.08413**	**4.391**	2.268

Table 2. **Evaluation metrics of different models on different datasets.** In metric columns with a down arrow ↓, the lower metric denotes the better performance of the corresponding model. Note that P-Net doesn't have a warm-up stage because the initial position is given. Models with "+ Warm-up" use warm-up steps $W = 32$. **Bold** denotes the best-performing models on each metric.

Different Warm-up Steps. To evaluate the contribution of the Warm-up Stage, we evaluate PAC-Net and C-Net with different warm-up steps. Note that a longer Warm-up Stage doesn't introduce extra computational cost during inference, but only delays the beginning of the Tracking Stage. In Fig. 4 we demonstrate some tracking results with different warm-up steps. With more steps to warm up, the trajectory will be more accurate when the tracking stage begins. We also observe a gradual saturation of the model's performance as warm-up steps increase. This trend can be clearly seen with statistic metrics shown in Fig. 7 and warm-up course (grey dashed lines) demonstrated in Fig. 4. With a warm-up step $W \approx 48$, the Warm-up PAC-Cell has taken in sufficient frames to warm up the hidden state $\mathbf{h}$ thus providing a reasonable initialization for the following Tracking Stage.

Real-time Inference. We run our model on a laptop with 8-core AMD Ryzen 7 5800H CPU and an Nvidia GeForce RTX 3060 laptop GPU. We achieve an inference speed of approximately 900 frames per second and therefore adequate for real-time inference. The single-scene inference speed on one card of NVIDIA A100 is about 5000 FPS.

6. Limitations and Future Work

Now our work is limited to 2D indoor tracking. However, our proposed method is also compatible with 3D tracking in the wild, which remains not validated due to the difficulty of collecting and labeling 3D data. It is prospective to extend our method to 3D scenes, which will meet more diversified real needs in fields like autonomous driving and security. Apart from that, our pipeline is now applied to single-object tracking tasks. It is natural to consider extending our method to multi-object tracking. The analysis of this problem will be difficult because the complexity of motion information doesn't accumulate linearly with the number of objects. Besides, developing self-supervised or semi-supervised techniques on NLOS tracking is also a promising perspective, which helps the generalization of models.

7. Conclusion

In this paper, we formally propose and formulate the task of real-time passive NLOS tracking. We introduce difference frame which conveys clean motion information and helps to reconstruct a continuous and smooth trajectory. In addition, we propose a novel and concise network architecture, PAC-Net, which is capable of maintaining good continuity and stability via processing raw frames and difference frames alternately. Note that PAC-Net is a framework-level architecture, and thus can be applied to other reconstruction tasks with dense observation in time series. We mitigate the dilemma of initialization of a recurrent-fashion network to some extent by disentangling an independent Warm-up Stage. To evaluate our proposed method and facilitate data-driven techniques, we establish the first passive NLOS tracking dataset, NLOS-Track, which contains about 1500 videos of realistic scenes. Both real-shot data and synthetic data are included to generalize the dataset.

Acknowledgement. This research is supported in part by Shanghai AI Laboratory and National Natural Science Foundation of China under Grant 62106183.

References

[1] Manel Baradad, Vickie Ye, Adam B Yedidia, Frédo Durand, William T Freeman, Gregory W Wornell, and Antonio Torralba. Inferring light fields from shadows. In *CVPR*, pages 6267–6275, 2018. 1, 2

[2] Paulo V. K. Borges, Ash Tews, and Dave Haddon. Pedestrian detection in industrial environments: Seeing around corners. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4231–4232, 2012. 1

[3] Katherine L Bouman, Vickie Ye, Adam B Yedidia, Frédo Durand, Gregory W Wornell, Antonio Torralba, and William T Freeman. Turning corners into cameras: Principles and methods. In *ICCV*, pages 2270–2278, 2017. 1, 2, 3

[4] James Brooks and Daniele Faccio. A single-shot non-line-of-sight range-finder. *Sensors*, 19(21):4820, 2019. 1, 2

[5] Yanpeng Cao, Rui Liang, Jiangxin Yang, Yanlong Cao, Zewei He, Jian Chen, and Xin Li. Computational framework for steady-state nlos localization under changing ambient illumination conditions. *Optics Express*, 30(2):2438–2452, 2022. 1, 2

[6] Piergiorgio Caramazza, Alessandro Boccolini, Daniel Buschek, Matthias Hullin, Catherine F. Higham, Robert Henderson, Roderick Murray-Smith, and Daniele Faccio. Neural network identification of people hidden from view with a single-pixel, single-photon detector. *Scientific Reports*, 8(1), 2018. 1

[7] Susan Chan, Ryan E Warburton, Genevieve Gariepy, Yoann Altmann, Stephen McLaughlin, Jonathan Leach, and D Faccio. Fast tracking of hidden objects with single-pixel detectors. *Electronics Letters*, 53(15):1005–1008, 2017. 1, 2

[8] Susan Chan, Ryan E. Warburton, Genevieve Gariepy, Jonathan Leach, and Daniele Faccio. Non-line-of-sight tracking of people at long range. *Opt. Express*, 25(9):10109–10117, May 2017. 1, 2

[9] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *NeurIPS*, 31, 2018. 4

[10] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. *Empirical Methods in Natural Language Processing*, 2014. 4

[11] Adam Lloyd Cohen. Anti-pinhole imaging. *Optica Acta: International Journal of Optics*, 29(1):63–67, 1982. 2

[12] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 6

[13] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *NeurIPS*, 32, 2019. 4

[14] Daniele Faccio, Andreas Velten, and Gordon Wetzstein. Non-line-of-sight imaging. *Nature Reviews Physics*, 2(6):318–327, 2020. 1

[15] Miguel Galindo, Julio Marco, Matthew O'Toole, Gordon Wetzstein, Diego Gutierrez, and Adrian Jarabo. A dataset for benchmarking time-resolved non-line-of-sight imaging. In *ACM SIGGRAPH 2019 Posters*, pages 1–2. Association for Computing Machinery, 2019. 2, 6

[16] Genevieve Gariepy, Francesco Tonolini, Robert Henderson, Jonathan Leach, and Daniele Faccio. Detection and tracking of moving objects hidden from view. *Nature Photonics*, 10(1):23–26, 2016. 1, 2

[17] Ruixu Geng, Yang Hu, Yan Chen, et al. Recent advances on non-line-of-sight imaging: Conventional physical models, deep learning, and new scenes. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2021. 1

[18] Ruixu Geng, Yang Hu, Zhi Lu, Cong Yu, Houqiang Li, Hengyu Zhang, and Yan Chen. Passive non-line-of-sight imaging using optimal transport. *IEEE TIP*, 31:110–124, 2022. 1, 2, 6

[19] JinHui He, ShuKong Wu, Ran Wei, and YuNing Zhang. Non-line-of-sight imaging and tracking of moving objects based on deep learning. *Optics Express*, 30(10):16758–16772, 2022. 1, 2, 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[21] Adrian Jarabo, Julio Marco, Adolfo Munoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. A framework for transient rendering. *ACM TOG*, 33(6):1–10, 2014. 2

[22] Charles F Jekel, Gerhard Venter, Martin P Venter, Nielen Stander, and Raphael T Haftka. Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. *International Journal of Material Forming*, 12(3):355–378, 2019. 6

[23] Jonathan Klein, Martin Laurenzis, and Matthias Hullin. Transient imaging for real-time tracking around a corner. In *Electro-Optical Remote Sensing X*, volume 9988, page 998802. SPIE, 2016. 1, 2

[24] Jonathan Klein, Martin Laurenzis, Dominik L Michels, and Matthias B Hullin. A quantitative platform for non-line-of-sight imaging problems. *BMVC*, page 104, 2018. 2, 6

[25] Jonathan Klein, Christoph Peters, Jaime Martín, Martin Laurenzis, and Matthias B Hullin. Tracking objects outside the line of sight using 2d intensity images. *Scientific reports*, 6(1):1–9, 2016. 1, 2

[26] Xiaochun Liu, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten. Non-line-of-sight imaging using phasor-field virtual wave optics. *Nature*, 572(7771):620–623, 2019. 2

[27] Tomohiro Maeda, Guy Satat, Tristan Swedish, Lagnojita Sinha, and Ramesh Raskar. Recent advances in imaging around corners. *arXiv preprint arXiv:1910.05613*, 2019. 1

[28] Christopher A Metzler, David B Lindell, and Gordon Wetzstein. Keyhole imaging: non-line-of-sight imaging and tracking of moving objects along a single optical path. *IEEE Transactions on Computational Imaging*, 7:1–12, 2020. 1, 2, 5

[29] Charles Saunders, John Murray-Bruce, and Vivek K Goyal. Computational periscopy with an ordinary digital camera. *Nature*, 565(7740):472–475, 2019. 1, 2

[30] Sheila W Seidel, Yanting Ma, John Murray-Bruce, Charles Saunders, William T Freeman, C Yu Christopher, and Vivek K Goyal. Corner occluder computational periscopy: Estimating a hidden scene from a single photograph. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–9. IEEE, 2019. 1, 2

[31] Sheila W. Seidel, John Murray-Bruce, Yanting Ma, Christopher Yu, William T. Freeman, and Vivek K Goyal. Two-dimensional non-line-of-sight scene estimation from a single edge occluder. *IEEE Transactions on Computational Imaging*, 7:58–72, 2021. 1, 2

[32] Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40, 2008. 6

[33] Prafull Sharma, Miika Aittala, Yoav Y. Schechner, Antonio Torralba, Gregory W. Wornell, William T. Freeman, and Frédo Durand. What you can learn by staring at a blank wall. In *ICCV*, pages 2330–2339, 2021. 1, 2, 3

[34] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NeurIPS*, 27, 2014. 3

[35] Brandon M Smith, Matthew O'Toole, and Mohit Gupta. Tracking multiple objects outside the line of sight using speckle imaging. In *CVPR*, pages 6258–6266, 2018. 1, 2

[36] Matthew Tancik, Guy Satat, and Ramesh Raskar. Flash photography for data-driven hidden scene recovery. *arXiv preprint arXiv:1810.11710*, 2018. 1, 2

[37] Antonio Torralba and William T Freeman. Accidental pinhole and pinspeck cameras: Revealing the scene outside the picture. In *CVPR*, pages 374–381, 2012. 2

[38] Conghe Wang, Yutong He, Xia Wang, Honghao Huang, Changda Yan, Xin Zhang, and Hongwei Chen. Passive non-line-of-sight imaging for moving targets with an event camera. *arXiv preprint arXiv:2209.13300*, 2022. 2, 6

[39] Yangyang Wang, Yaqin Zhang, Meiyu Huang, Zhao Chen, Yi Jia, Yudong Weng, Lin Xiao, and Xueshuang Xiang. Accurate but fragile passive non-line-of-sight recognition. *Communications Physics*, 4(1):1–9, 2021. 1, 2

[40] Katharina Witowski and Nielen Stander. Parameter identification of hysteretic models using partial curve mapping. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 5580, 2012. 6

[41] Adam B Yedidia, Manel Baradad, Christos Thrampoulidis, William T Freeman, and Gregory W Wornell. Using unknown occluders to recover hidden scenes. In *CVPR*, pages 12231–12239, 2019. 1, 2

[42] Dayu Zhu and Wenshan Cai. Fast non-line-of-sight imaging with two-step deep remapping. *ACS Photonics*, 9(6):2046–2055, 2022. 2