The Invisible Shadow: How Security Cameras Leak Private Activities

Jian Gong *
School of Computer Science and Engineering, Central South University, China
gongjian@csu.edu.cn

Xinyu Zhang
University of California San Diego
United States
xyzhang@ucsd.edu

Ju Ren†
Department of Computer Science and Technology, BNRist, Tsinghua University, China
renju@mail.tsinghua.edu.cn

Yaoxue Zhang
Department of Computer Science and Technology, BNRist, Tsinghua University, China
zhangyx@mail.tsinghua.edu.cn

ABSTRACT

This paper presents a new privacy threat, the Invisible Infrared Shadow Attack (IRSA), which leverages the inconspicuous infrared (IR) light emitted by indoor security cameras, to reveal in-home human activities behind opaque curtains. The key observation is that the in-home IR light source can project invisible shadows on the window curtains, which can be captured by an attacker outside using an IR-capable camera. The major challenge for IRSA lies in the shadow deformation caused by a variety of environmental factors involving the IR source position and curtain shape, which distorts the body contour. A two-stage attack scheme is proposed to circumvent the challenge. Specifically, a DeShaNet model performs accurate shadow keypoint detection through multi-dimension feature fusion. Then a scene constructor maps the 2D shadow keypoints to 3D human skeletons by iteratively reproducing the on-site shadow projection process in a virtual Unity 3D environment. Through comprehensive evaluation, we show that the proposed attack scheme can be successfully launched to recover 3D skeleton of the victims, even under severe shadow deformation. Finally, we propose potential defense mechanisms against the IRSA.

CCS CONCEPTS

• **Security and privacy → Privacy protections**.

KEYWORDS

security camera; infrared light; privacy leakage

ACM Reference Format:
Jian Gong, Xinyu Zhang, Ju Ren, and Yaoxue Zhang. 2021. The Invisible Shadow: How Security Cameras Leak Private Activities. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3460120.3484741

*This work was conducted when the author was a visiting scholar at the University of California San Diego.
†The corresponding author.

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8454-4/21/11…$15.00
https://doi.org/10.1145/3460120.3484741

1 INTRODUCTION

Over the past a few years, smart security cameras have been quickly emerging as a critical element in the smart-home ecosystem. According to market research, around 38% American homes own at least one security camera [29], more than 40% of which are deployed indoor. The market size of security camera has reached $3.6 billion as of 2020, and will surge to $11.6 billion in the next 6 years [29]. The security cameras have been playing a crucial role in monitoring kids, pets, appliance safety, and in protecting against crimes.

Whereas users may trust such devices' promise to keep the video records secure, they may not be aware that these devices are acting as an infrared (IR) light source, which can create a side channel to leak user privacy. In this paper, we investigate a novel attack that makes this possible. Fig. 1 showcases a typical attack scenario. Most security cameras have a night vision mode [1, 5], when they illuminate the target scene with a built-in IR LED to aid low-light video capture. The IR light can easily penetrate thin opaque materials such as window curtains made of cotton and viole which are used to block visible light. Thus, when a home resident stays in between the camera and window, the IR light can project the resident's shadow on an opaque curtain. Though invisible to human eyes, the shadows can be captured by an attacker outside the home using an IR camera. Now that the IR shadows become visible video footage, they can reveal private information such as health/medical conditions, special hobbies, and intimacy between residents. We refer to such an attack as *IR shadow attack* (IRSA).

There are two major challenges to carry out the IRSA in reality. *First*, the IR shadow may be deformed by multiple unpredictable environmental factors, which severely disturb the visual features, making it hard to identify the body contours. These factors include the IR light source's projection angle, the distance between IR source and the curtain, and the irregular curtain surface due to deformation. We refer to these factors as **scene parameters**. One possible solution is to fine-tune and apply existing human body keypoint detection models [11] on the shadow images. However, the scene parameters cause huge diversity on the shadow deformation, which will easily fail existing models. To overcome this challenge, we design a shadow keypoint detection model called De-Shadow Network (*DeShaNet*). DeShaNet incorporates a scene feature fusion module to learn the scene parameters that cause

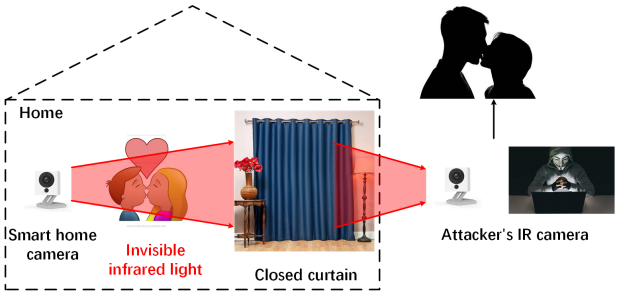

Figure 1: An example showcase of the invisible IR shadow attack. A smart home camera emits IR light and projects the contours of a couple's bodies onto the curtain while they are kissing inside, which are then captured by another smart home camera outside the window. Through the leaked IR shadow, anyone can clearly observe the private activities of the victim couple.

shadow deformation. With an explicit representation of the scene parameters, it recovers the 2D shadow keypoints even under severe deformation. Additionally, it incorporates a *conditional attention module* to increase the detection robustness by automatically evaluating the environment factors and fusing multi-dimensional feature vectors, *e.g.*, the shadow deformation and movement intensity.

Second, the shadow deformation weakens the geometric relation between the 2D shadow keypoint positions and 3D body skeleton in most cases, making it hard to infer the victims' activities merely from the shadow. We thus introduce a *scene constructor* scheme, which consists of a scene parameter estimator (SCE) and a shadow simulator, to explicitly model the scene parameters. The SCE reverse-engineers 4 major scene parameters (the IR source angle/distance, limb length and curtain deformation) by analyzing the deformation characteristics of the shadow. Then, the shadow simulator, which is essentially a 3D scene simulator, tries to iteratively reproduce the same deformed shadow by manipulating a 3D dummy in the corresponding virtual 3D environment. During this process, the accurate 3D skeletons of the victim are derived as a byproduct.

We have implemented the DeShaNet using Pytorch [27] and the scene constructor using Unity 3D [37]. We collect a dataset of 24k video frames to train and test the attack scheme. The dataset covers a variety of realistic situations, including different IR source angles/distances, human subjects, security camera hardware, curtain materials, dark/bright environment, etc. From the experiments, we observe that the DeShaNet framework largely decreases the shadow keypoint detection error compared with existing models. Most importantly, the attack scheme can accurately restore the subject's 3D skeletons with only a few pixels' error even under severe shadow deformation.

To our knowledge, we are the first to propose the concept of IRSA, and reveal the alarming privacy issues of security cameras due to their invisible shadow effect. Our main contributions can be summarized as follows:

- We propose the DeShaNet to detect shadow keypoints, even when the shadows are severely distorted due to curtain deformation.
- We propose a scene constructor design to restore 3D human skeleton from 2D shadow keypoints, in spite of unknown scene parameters such as distance/angle of the IR source relative to the (deformed) curtain.
- We implement the DeShaNet and scene constructor schemes and conduct extensive experiments to validate the feasibility of IRSA. Our evaluation also reveals limitations of the IRSA and hints to possible defense mechanisms.

2 BACKGROUND

2.1 Characteristics of the IR light

The IR light is a kind of electromagnetic wave with a wavelength of 760nm-1000nm, which is imperceptible by human eyes. But the modern photosensitive chips used in cameras have a much wider range of wavelengths, which enables them to capture the IR as well as visible light. Therefore, the IR light is widely used for auxiliary lighting on surveillance cameras, which grants night vision for the cameras without affecting human. However, for normal smartphone cameras, manufacturers usually insert the IR light filter to reduce glare from the IR spectrum [38]. Another important property of the IR light is the penetration ability. Prior work found that the IR camera can see through cloth, such as T-shirt [45] and clothes[44]. All in all, the invisibility and the penetrability provide necessary conditions for the IR shadow attack.

2.2 Attack model and assumptions

In IRSA, we assume the victims are located between the IR source (e.g., the security camera) and a window with curtains. The IR source casts invisible IR light on the victims and projects their body contours on the curtains. The projected IR shadows are then recorded by the IR camera of an attacker outside the victim's window. Due to various scene parameters, e.g, curtain deformation and abnormal IR angle, the recorded shadows are deformed, making it hard to infer the victim's activities directly.

Two requirements are necessary for IRSA to succeed. First, the IR illumination should last long enough, so that the attacker can observe the victim's activities continuously through the invisible shadow and eventually impinge on her privacy. Second, majority of the shadows should be projected on the window curtain, which provides opportunities for the attacker outside to capture the shadows and infer private activities. In what follows, we discuss the likelihood that these requirements are satisfied in practice, and the corresponding victim population.

2.3 Potential victim population

We now discuss the practical vulnerability factors related to IRSA, and empirically estimate the victim population.

Global sales volume of smart cameras. According to a research report from Strategy Analytics [33], the global sales volume of smart cameras was 56 million in 2019, and will increase with an annual growth rate of around 20%. As of 2021, the sales volume is around 80 million. Suppose every household consists of 2 users on average, then around 160 million users will be using smart cameras in their homes.

IR Angle. We analyze the 10 best-selling smart cameras ranked by sales volume on two major online retailers [3, 16]. Their average infrared radiation angle is about 120° (both horizontal and vertical).

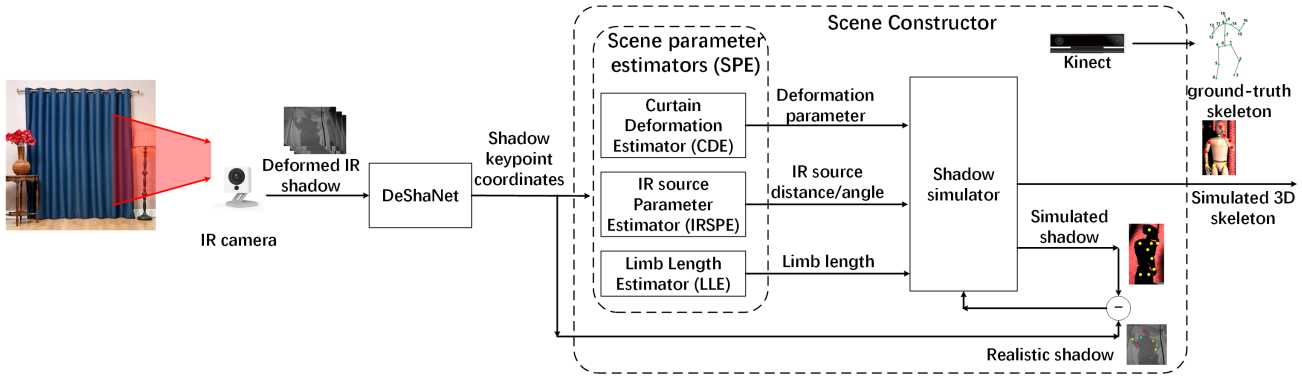

Figure 2: Overall architecture of the invisible IR shadow attack.

As a result, even if the smart cameras are positioned at arbitrary horizontal angles within a windowed room, there is still 1/3 chance of illuminating the window.

IR distance. The median housing area of all countries around the world is about 1000 sq ft [23]. Considering that the most common house type is 2B1B with living room, the maximum distance inside the house (the diagonal distance of a square house) with an area of 1100 sq ft is less than 5 meters, and 70% of the interior of the house with an area of 2200 sq ft is less than 5 meters. On the other hand, the effective distance of the IR light can reach 5 meters, even for the low-profile camera used in our experiments, and similarly for other commodity security cameras [42, 43]. High end smart cameras may have an even longer range. Considering the population distribution and housing area of each country [30], 94.5% of households have a maximum distance of less than 5 meters, which meets the distance requirement of IRSA.

User habits. The smart cameras are typically used for monitoring pets, babies and identifying emergency. So they are usually installed in the main areas of the house, such as bedrooms and living rooms. So it is reasonable to assume that smart cameras can capture a wide range of activities (including privacy sensitive ones), and there is a non-trivial probability that the camera's field of view (FoV) covers a user and part of the window.

Private activity time. People's private activities usually occur after they return home at night, when the light intensity is low and the infrared light tends to be triggered. According to existing tests of smart cameras [42, 43], the IR light can be automatically triggered even under the illumination of typical ceiling lights or desktop lamp, i.e., the ambient environment does not have to be completely dark. On the other hand, many private activities are performed at low ambient illumination, such as masturbating, sexual intercourse, etc. Overall, when the infrared light is triggered, it happens to be the peak time of private activities.

Ratio of curtains/blinds. According to a survey of the global window covering market [12], the ratio of curtains among all kinds of window covers is over 30% in 2018. Therefore, we can assume over 30% of homes use curtains as their window cover.

Based on the above analysis, we can gauge the number of people vulnerable to IRSA. At present, at least 160 million users are engaged with smart cameras at home. Consider the attrition factors, i.e., IR distance (94.5%), IR Angle (33%), curtain/blinds ratio (30%), the potential victim population of IRSA is about $160 * 0.945 * 0.33 * 0.3 =$ 15 million, which is alarming and will grow over time as the smart camera market expands.

2.4 Significance of the attack

How would people react when their activities at home are exposed to others? In principle, the invisible IR shadow can reveal any kinds of activities, after the 3D keypoints are recovered by the proposed framework. According to the survey by Choe et al. [9], 1400 types of in-home activities are considered private, and more than 39% percent of respondents thought exposure of such activities would make them feel **extremely** embarrassed and uncomfortable. We note that some activities are not so obvious but considered private by many people, *e.g.*, eating (indicating unhealthy behavior especially when it lasts long), body twisting (associated with intimate behavior for some people).

The shadow keypoints and 3D skeleton outputs from DeShaNet can potentially leak more information than the activity alone. Existing studies [15, 28, 32, 40] have shown that human shadows reflect walking gestures, which can be used to distinguish different people. With sufficient video footage from public figures, the attacker can train a model to associate the shadow records with the people identities, which poses a more severe privacy threat. Besides the smart security cameras, other popular in-home devices such as Kinect emit IR lights in a similar manner. Many other devices, such as smart display or video call portal, do not have night-vision yet. But they may incorporate this function in the near future and become the vulnerable point for ISRA. Therefore, we believe the ISRA is an alarming issue that should be investigated immediately.

3 SYSTEM OVERVIEW

The proposed IRSA consists of two key steps, as shown in Fig. 2. First, the attacker captures the IR video and feeds it into DeShaNet, which extracts the keypoints of the (deformed) shadow for each video frame. Second, the attacker uses the scene constructor to map the keypoints to a 3D skeleton. More specifically, the scene constructor estimates a set of scene parameters (SPEs) based on the keypoint positions. It then employs a shadow simulator to imitate the realistic shadow in this virtual scene by optimizing the 3D skeleton layout. The final output of the system is the optimized 3D skeleton, which can be used to extract private information such as activity and identity [28, 40].

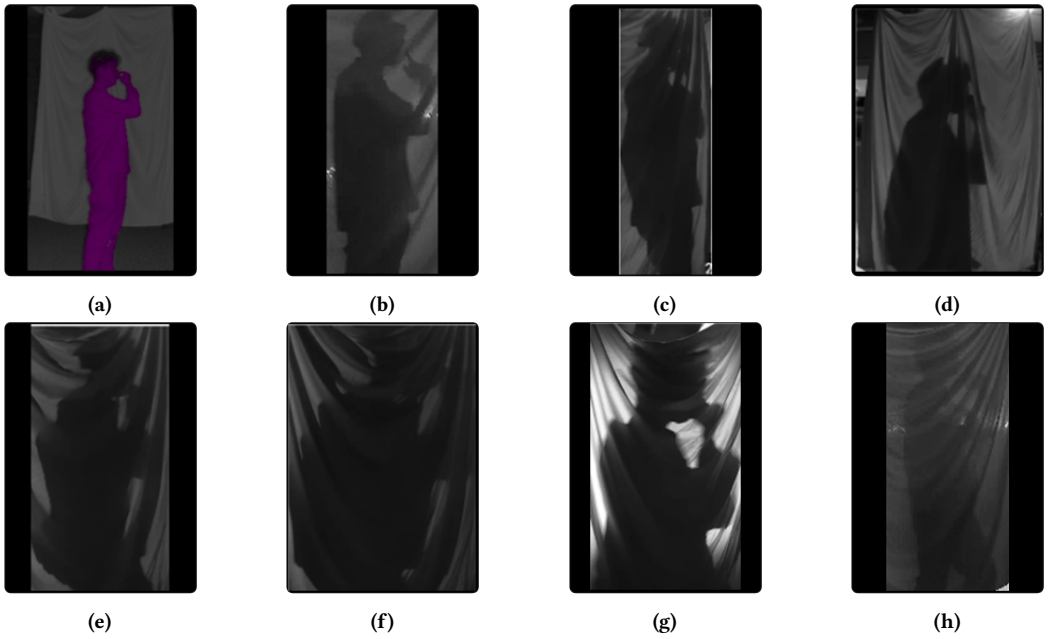

Figure 3: Realistic deformed IR shadows captured by IR cameras under different scene parameters. (a) A person is picking nose in place captured by an interior IR camera. (b) W/o deformation. (c) Curtain deformation: U-shape. (d) Curtain deformation: vertical. (e) IR angle: 30°. (f) IR angle: 60°. (g) IR distance: 1m. (h) IR distance: 5m.

4 DESHANET DESIGN

4.1 Shadow Deformation Caused by Scene Parameters

The shadow deformation can severely distort the IR shadow appearance. We showcase the problem in Fig. 3. We identify 3 major scene parameters causing the shadow deformation: curtain deformation, IR angle and IR distance (i.e., angle/distance of the in-home security camera relative to the curtain surface). In an ideal scenario where the curtain is flat, the IR angle is 0° and distance is short (3m), the attacker-captured IR shadow (Fig. 3(b)) is almost the same as the ground-truth captured by an in-home camera (Fig. 3(a)).

Curtain deformation. Fig. 3(c,d) show two deformed shadows with different curtain deformations: U-shape and vertical. We see that the shadows are deformed obviously, especially on the small body parts (hands and arms). If we can extract the features of the projection surface (i.e., window curtain in this case) by observing the shadow variation, then we may reconstruct the exact positions of the shadow keypoints.

IR angle. Fig. 3(e,f) show two deformed shadows under different IR angles. In general, a larger IR angle stretches the shadow more, and causes the curtain itself to create shadows. For example, the victim's hand can be identified at 30°, but occluded by the curtain's shadow at 60°.

IR distance. Longer IR distance has a much lower shadow contrast and size (Fig. 3(g,h)). Additionally, combined with the curtain deformation, the size variation also changes the shape of the shadow. For example, the hand of the shadow can be clearly seen when the IR source is near but distorted afar.

4.2 Design Motivation and Details

To detect the keypoints under shadow deformation, our DeShaNet solution framework incorporates three sub-modules: 1. An A-LSTM and scene feature fusion module, which can extract the features related to scene parameters, and hence adapt to the scene variations. 2. A trajectory aware module which introduces visually independent features, such as coordinate vectors, to improve the stability under fuzzy shadows with varying deformation. 3. A condition attention module, which improves the detection robustness under dynamic situations. Next we describe each module in detail.

Choice of feature extraction backbone. The state-of-the-art video keypoint detection models, such as 3D mask R-CNN, do not fit our scenario because their region proposal network does not support global image feature, which is essential to solve the shadow deformation problem. The global image feature refers the high-level image feature with acceptance field covering the overall image. In contrast, our DeShaNet architecture (Fig. 4) aims to capture global image features, which contain rich information related with the scene parameters.

Specifically, we use pretrained convolutional neural network (CNN) stacks from the Resnet-50 [14] to extract global image features F_v. Since the features of IR shadows are very different from RGB images, the pretrained CNN backbone needs to be fine-tuned on a large number of IR shadow images. To reduce the amount of new training data needed, we freeze the parameters in the bottom layers of the pretrained CNN backbone and fine-tune the parameters in the top layers. To find the best balance between generalization ability and training data requirements, we try different combinations of frozen layers and fine tuning layers, and empirically choose the combination (freezing the first 3 layers and fine-tuning the rest) that achieves best performance when tested on real data.

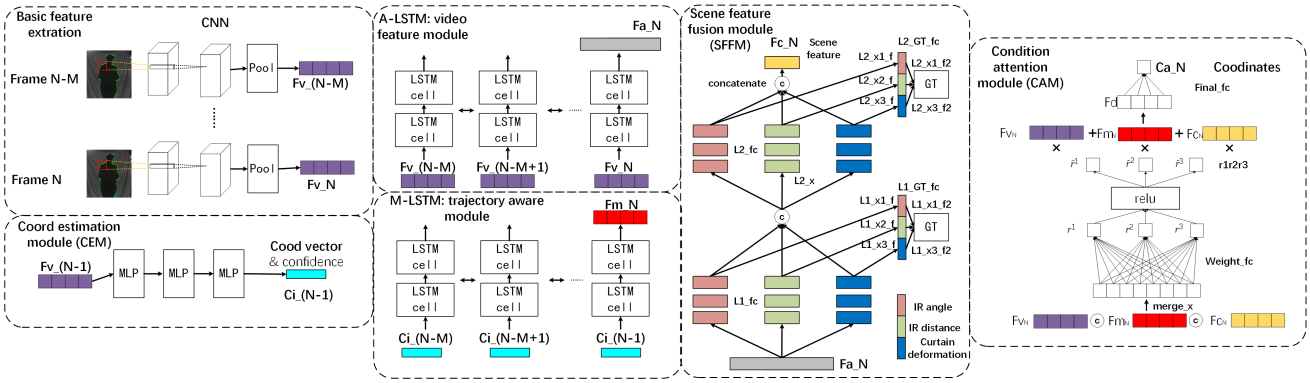

Figure 4: Overall architecture of the DeShaNet.

Scene feature fusion module (SFFM). We design the SFFM to extract the correlations between the three major scene parameters and dynamic shadow features (Section 4.1). SFFM builds on the video feature extraction module (A-LSTM). The visual feature F_a output by the A-LSTM implicitly contains information of the scene parameters. We thus feed F_a into 3 MLP branches, each of which learns to predict one scene parameter, as shown in Fig. 4. All the parameters (IR angle/distance and curtain deformation level) are normalized.

Since the shadow deformation is a product generated by all scene parameters, knowing one of them can help estimate others. Therefore, the SFFM adopts a two-stage architecture. In the first stage, the output features are trained to learn the scene parameters. Then these outputs are fused by concatenation and fed into the second stage, which is trained to reach higher accuracy. The final output features are fused again to produce the scene features F_c, which are then fed into the condition attention module for feature fusion. In our implementation, the first stage comprises 2 layers of MLP with kernel size of 64, and the second has 2 layers of MLP with kernel size of 96.

Trajectory aware module. To deal with some extreme cases when the shadow parts are deformed severely or merged together, our trajectory aware module leverages the motion continuity so that the keypoints of deformed shadows can be inferred from historical keypoint trajectory explicitly. This module consists of a coordinate estimation module (CEM) and an M-LSTM. The CEM predicts the keypoint coordinates from historical images C_i^{N-M}-C_i^{N-1} by CNN stacks and MLP layers. It comprises 3 layers of MLPs with hidden size of 64, 96 and 18. The next step is to predict the coordinate feature F_m of the current image from these coordinates, which is essentially a sequence to sequence learning problem [35]. Therefore, it is natural to use the LSTM for this task, which excels at modeling temporal information from long sequences. This LSTM model (referred to as M-LSTM) comprises two layers of LSTM cells with hidden layer size of 96.

Condition attention module (CAM). Three feature vectors (F_v, F_m and F_c) are involved for the final keypoint coordinates prediction. However, these feature vectors have completely different physical meanings and may become less reliable under specific situations. Specifically, the visual feature vector F_v will be less reliable when the shadows become fuzzy due to high dynamic movement or severe occlusion. When the shadow movement speed becomes relatively slow, the trajectory feature vector F_m does not contain much useful information. The scene feature vector F_c should have less impact when the scene parameters do not cause much shadow deformation. To fuse these highly heterogeneous feature vectors, we custom build an attention module called CAM. The CAM comprises of a feed-forward network to calculate the fusion weights $\bar{r}^1$, $\bar{r}^2$ and $\bar{r}^3$, which are then multiplied with the 3 feature vectors and fed into an MLP layer to predict the final coordinates C_a^N. The size of the feed-forward network is the sum of the 3 feature vectors and the size of the MLP layer is the same as one of the feature vectors. The C_a^N contains the normalized 2D (x, y) coordinates for 9 keypoints.

5 SCENE CONSTRUCTOR DESIGN

DeShaNet can recover the 2D keypoint positions from the shadow, but these still need to be converted to a 3D skeleton to enable human activity recognition. Unlike classical 3D skeleton detection tasks in computer vision, restoring 3D skeletons from the 2D deformed shadow images is essentially an undetermined problem. Our scene constructor framework aims to overcome this hindrance by filling in environmental information. It estimates the scene parameters by modeling and simulating the shadow projection process in a virtual 3D environment. The 3D skeleton of the victim is derived as a byproduct of this process. The overall architecture of the scene constructor is shown in Fig. 5.

5.1 Design principle of the scene parameter estimators

The scene constructor consists of 3 scene parameter estimators which we detail below. To ease the exposition, we summarize the related math symbols in Table 1.

IR Source Parameter Estimator (IRSPE). Intuitively, the IR distance affects the shadow size and the IR angle causes horizontal stretching. Therefore, by analyzing the shadow distortion, we can infer the IR distance/angle. As shown in Fig. 6, we denote the IR distance as h, and denote the horizontal distance between the shadow edge and the IR source as x. Further, we denote the angle of the IR source relative to the edge of shadow as θ. Through simple geometries, we have:

$$x = h * tan(\theta). \tag{1}$$

Let the width of the shadow be Δx, then we have:

$$x + \Delta x = h * tan(\theta + \Delta\theta). \tag{2}$$

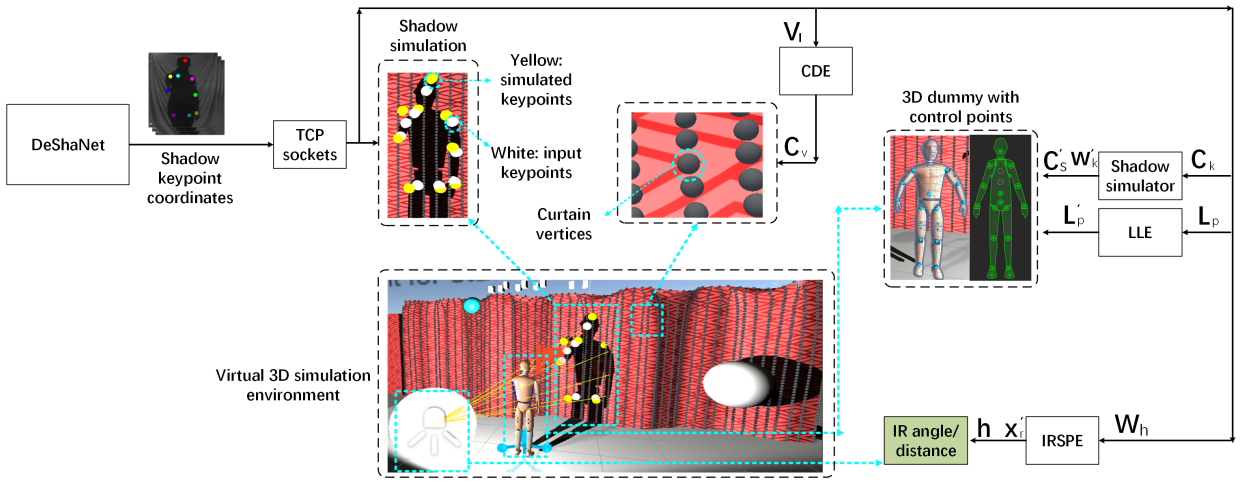

Figure 5: Overall architecture of the scene constructor.

Table 1: Annotations and abbreviations in this paper.

Symbols	Description
$C_k, C_k^{'}$	Shadow keypoint coordinates from DeShaNet/simulation, where k refers the keypoint id.
W_h	Input sequence of the head width on the shadow
h	Distance between the IR source and the curtain
x_r	horizontal distance between the IR source and the right edge of the curtain
$L_p, L_p^{'}$	Relative limb length from input/simulation, where p refers the index of limb pairs
$v_l, v_l^{'}$	Local shadow speed from input/simulation
C_v	Curtain vertex coordinates
w_k	Rotation angle of all 3D skeleton joints, where k refers to keypoint id
w_a	Rotational angle of the overall skeleton
C_s	3D skeleton position coordinates

Then we subtract equation (2) by (1):

$$\Delta x = h * (tan(\theta + \Delta\theta) - tan(\theta)) \approx h * (1/cos(\theta)^2). \quad (3)$$

Equation (3) shows the relationship between the shadow width, the IR source distance and the IR source angle. Since the shadow width Δx varies at different locations, we can infer the h and θ by sampling the Δx when the victim moves across multiple locations. However, estimating h and θ simultaneously is an undetermined problem. Our IRSPE adopts an iterative simulation driven solution with the following high level work flow. It simulates a massive number of distance and angle settings of the IR source. For each simulation sample, the shadow is projected on a virtual curtain and its width is compared with the realistic shadow width. Then, we calculate the difference between the simulated shadow width and the realistic shadow width. Finally, the IR source angle/distance with the smallest difference is regarded as the optimal estimation.

In practice, the shadow width can also be impacted by curtain deformation, the distance variation between the victim and the curtain, and body postures. We thus introduce the following mechanisms to counteract these factors.

Limb length estimator (LLE). We introduce an LLE scheme to reduce the skeleton estimation error due to the limb length variation across people. Intuitively, the 3D limb length is proportional to the 2D projected limb length on the shadow. However, two major factors can weaken this correlation: *1. Body posture.* The varying distance between two keypoints on the shadow reaches maximum when their connecting line is parallel to the window curtain. At this time, the distance between all pairs of keypoints have exactly the same deformation. Therefore, we can use the maximum 2D distance of all pairs of keypoints to approximate the 3D limb length. *2. The residual errors of the DeShaNet keypoint output.* It is well known that the keypoint detection errors of DL models follow a Gaussian distribution [18]. The *2D* projected limb length can be calculated from the distance between keypoint locations. Therefore, the true value of limb length should lies in the top-N 2D distance instead of top-1. We then take the median of the top-N distances as the estimated 2D limb length.

For match score calculation, we use the ratio between the absolute limb length and an anchor length as metric. Since the distance between the head and the shoulder is relatively stable under different body postures and viewpoints, we use it as the the *anchor length.*

Curtain deformation estimator (CDE). The curtain deformation stretches the shadow, making the shadowed body parts merge together or changing their shapes. To overcome this issue, our CDE scheme explicitly reconstructs the deformed surface of the curtain in a virtual environment, and reproduces the same deformation effect as that observed by the attacker's camera. As the victim moves across locations, the movement of the observed shadow exhibits different levels of fluctuations due to curtain deformation. A wrinkled curtain surface will fluctuate the shadow moving speed more than a smooth curtain. Fig. 7 clearly showcases this relationship.

To simplify the explanation of CDE, we define the local moving speed of keypoint l as v_l, local deformation angle θ_l, mean angle between curtain and attacker θ_m, mean moving speed between curtain and attacker V_m. Intuitively, when v_l decreases, θ_l becomes larger. During the aforementioned simulation process, the θ_l can be altered by modifying the coordinates of the vertices of the virtual curtain. We iteratively adjust θ_l to make the distribution of all

simulated speeds v_s most close to the distribution of v_l. This in turn leads to a curtain deformation closest to reality.

5.2 The shadow simulator

The shadow simulator aims to simulate shadows and make their keypoint coordinates match the input shadow keypoint coordinates which are derived from the DeShaNet. It achieves this by placing and modifying virtual components in a Unity 3D environment, including the IR light source, body skeleton model and window curtain. Among these components, the IR light source, the window curtain and the skeleton limb length are estimated by the aforementioned 3 scene parameter estimators, respectively. The shadow simulator mainly aims to derive the parameters of a 3D dummy skeleton, including: *(i)* Skeleton rotational angles: the rotational angles of the overall skeleton (w_a) and of all keypoints (w_k). *(ii)* Skeleton position C_s: the (x, y) coordinates of the overall skeleton.

The implementation of the shadow simulator follows 2 steps: parameter generation and match score calculation.

1. Parameter generation: For each parameter, the shadow simulator exhaustively tries all possible values in empirically predefined scopes and intervals (listed in Table 2).

2. Match score calculation: After each parameter is updated, the shadow is refreshed accordingly. We then calculate the match score, defined as the difference between the keypoint coordinates of the simulated shadow $C_k^{'}$ and that of the input shadow C_k from DeShaNet: $S = \sum ||C_k - C_k^{'}||_2$. The parameter set with the lowest match score will be used as the optimal estimation.

To reduce the huge search space caused by possible combination of skeleton parameters, the shadow simulator groups the parameters according to each parameter's impacts on others, and updates them sequentially in descending order. The *impact* of a skeleton parameter is determined by its number of leaf nodes. For example, the rotational angle of the overall skeleton w_a affects all the skeleton keypoint coordinates, so it has 9 leaf nodes. On the other hand, the rotational angle of the wrist w_7/w_1 has the lowest impacts because it does not affect other skeleton parameters, i.e., it has 0 leaf nodes. We list all the parameters according to their impacts in descending order as follows: C_s, w_a, w_2, w_3, w_4,w_5, w_6,w_7,w_1,w_8,w_9, where the keypoint indices from 1 to 9 are head, l-shoulder, r-shoulder, l-elbow, r-elbow, l-wrist, r-wrist, l-thigh, r-thigh, respectively.

6 SYSTEM IMPLEMENTATION

Dataset. We create a realistic indoor scene to perform the IRSA and collect data, as shown in Fig. 8. We recruit different subjects and conduct various activities between the IR source and the curtain. To simplify the ground-truth data collection, we use Kinect v2 as IR source, which is equipped with a similar IR emitter as typical security cameras (Sec. 7.3). We then use a commercial software, Brekel Body v2 [6], which is designed for Kinect, to derive the ground-truth body skeletons from Kinect videos. The IR shadow is captured by a smart home camera (Wyze [5]) on the other side of the curtain. The curtain can be manually adjusted to arbitrary shapes to simulate different curtain patterns by clamps and fixtures on the wall.

We collect over 40 groups of data, each of which records a 1-2 minutes IR shadow video and the corresponding 3D skeleton ground-truth, along with a visible body movement video (captured by the Kinect). We manually label the shadow keypoints for each image in the IR videos by comparing it against the body movement video. The IR video records the shadow of the subject performing multiple kinds of activities, including armpit stretching, nose picking, body twisting, eating (for a long time), bottom uplifting (simulating sex behavior), dancing, walking, running. Among these activities, the first 5 are considered privacy-concerning by over 39% people according to the existing survey [9]. Additionally, the dataset covers a variety of realistic situations including different IR distances/angles and curtain deformation.

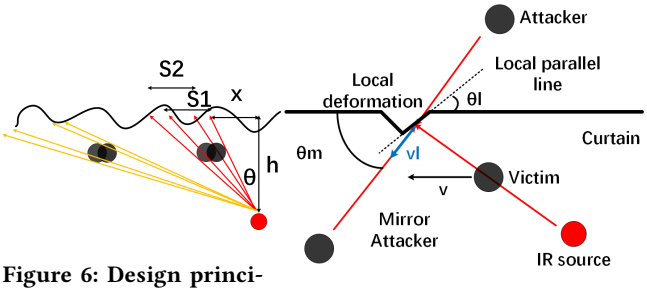

Figure 6: Design principle of the IRSPE.

Figure 7: Design principle of the CDE.

Table 2: Empirical scopes and intervals of all parameters.

	w_k	C_s	w_a	h	x_r	C_v
Scope	-120° ~120°	-5m~5m	-180° ~180°	1m~5m	-5m~5m	-0.1m ~0.1m
Interval	1°	0.1m	1°	0.1m	0.1m	0.01m

Model implementation. The DeShaNet is implemented using Pytorch [27] and trained on a GPU server. The scene constructor is implemented in Unity 3D [37]. The virtual curtain is implemented using Obi cloth [25], which provides fine grained control and simulation of the curtain surface. The 3D dummy is derived from the PuppetMaster [36].

Baseline methods. We choose the 3D Mask R-CNN [11] as our baseline for keypoint detection. We fine-tune the original 3D Mask R-CNN implementation [10] on our dataset by fine-tuning the keypoint detection head branch and the classification head. The tube proposal network and CNN backbone remain unchanged.

7 EVALUATION

7.1 Evaluation of the DeShaNet

We first evaluate the proposed DeShaNet by varying one scene parameter while fixing others to the default. By default, the curtain deformation is u-shape, IR distance is 2.6m, IR angle 0° and subject ID is 1. The datasets are split into training set and testing set. All the models are trained on the same training set and evaluated on the testing set with the same training epoch. The evaluation metric in this section is 2D detection error in pixel. Considering the average field of view is about 3.6*2 meters and the image size is 384*216, a pixel corresponds to 9.3 mm. The exact conversion factors may change according to the actual environment.

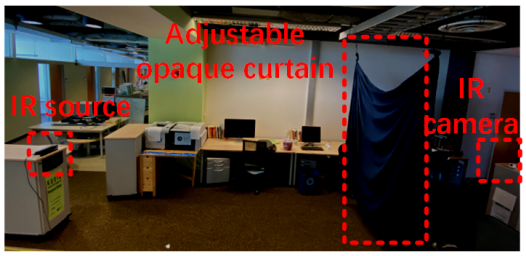

Figure 8: Experiment scene for the IRSA (low illumination).

Figure 9: Experiment scene for the IRSA (day time).

Table 3: Shadow keypoint detection error comparison on different levels of curtain deformation (pixels).

model	w/o	vertical	U-shape	random	average
3D Mask R-CNN	5.13	8.64	10.22	12.75	9.18
vanilla-CNN	7.25	10.17	12.86	12.95	10.8
DeShaNet -CNN	5.04	6.13	7.52	7.99	6.67
DeShaNet	4.52	5.25	7.09	7.28	6.03

Different curtain deformation. In this experiment, both the training data and the testing data include 4 kinds of curtain deformation: vertical deformation, random deformation, U-shape deformation and no deformation. The results in Table 3 show that DeShaNet has the lowest error on all kinds of deformations, which is 32% lower than 3D Mask R-CNN on average. This indicates the design of SFFM and CAM in DeShaNet eliminate the performance bottleneck of the 3D Mask R-CNN on shadow keypoint detection. Additionally, the error of DeShaNet w/ deformation is only 2.02 larger than w/o deformation, the increase rate of which is 62% smaller than the 3D Mask R-CNN. This indicates the deformation largely impacts the detection, and DeShaNet can solve the problem very well. Although different curtain deformations have different visual impacts, DeShaNet shows consistent performance, indicating that it achieves good generalization ability.

On the other hand, to verify the performance boost by the SFFM and CAM, we implement two extra baseline models : DeShaNet -CNN and the vanilla-CNN. These two are essentially the CNN parts of DeShaNet (CNN backbone + CEM). The difference is the DeShaNet -CNN is trained together with the overall DeShaNet and the vanilla-CNN is trained separately. The results show that DeShaNet -CNN outperforms the vanilla-CNN on all kinds of curtain deformation, with 38% lower error on average. It confirms that the SFFM and CAM components of DeShaNet can indeed boost the overall performance of the CNN part through joint training. We illustrate several examples under severe curtain deformation in Fig. 10 (a)-(d).

Different IR source distance/angle. We now vary the IR distance and angle, and summarize the results in Table 4. When the IR distance is within a specific range (< 4.2m), the maximum error of DeShaNet (7.09) is only 17% larger than the smallest error (5.83), indicating that the IR distance does not impact the keypoint estimation in a noticeable way. We refer to this distance threshold as the *margin distance*. When the IR distance reaches the margin distance (4.2m), the detection error increases greatly to 8.2, which is 40% higher than the smallest error. This is mainly due to the attenuation of the IR light strength over a long distance.

On the other hand, the IR angle has a much larger impacts than the distance. The results in Table 5 show that the detection error of DeShaNet increases along with angle. When the angle exceeds 60°, the error reaches the maximum of 10.25, which is 44% larger than the smallest error at 0°, indicating that the model becomes unreliable at extreme large IR angles. Additionally, DeShaNet outperforms the 3D Mask R-CNN across all angles, with an average error reduction of 28%. We note that there are error dips from 0 to 15 degrees. This is because the errors within 0-30 degrees do not increase by the angle. The dips are mainly caused by normal model errors. We illustrate examples under extreme IR angles and long IR distances in Fig. 10 (e)-(l).

Different subjects/limb length. We evaluate the generalization ability of the DeShaNet across different subjects. We use a single subject's data to train the DeShaNet model, and test it over 4 other subjects. All the scene parameters are set to default values. Table 6 summarizes the results. We observe that the average error of DeShaNet on the testing subjects 2-5 is 7.20 pixels (translating to around 5.6 cm), which is only 1.6% higher than training/testing on subject 1. Such a strong generalization capability comes from the SFF and M-LSTM modules, both of which are insensitive to the individual subject's features. Additionally, by inspecting the raw video data, we find that the moving speed of subjects leads to the slight different results. Subject 2 is obviously slower and has the lowest estimation error. We illustrate examples of different subjects in Fig. 10 (m)-(p).

7.2 Evaluation of the Scene Constructor

In this section, we evaluate the performance of the scene constructor, including its 3 SPEs and the shadow simulator. When evaluating each SPE, we disable the other 2 SPEs and set corresponding scene parameters to the default values, i.e., the IR distance is 5m and angle 0°; The height of the virtual dummy is 1.7m and the curtain is flat without deformation.

Evaluation of IRSPE. We first test whether the IRSPE can accurately estimate the IR source position. The testing data includes 5 different IR distance and 5 different angles The shadow keypoints of videos are first detected by the DeShaNet and then converted to anchor width sequences by the contour detection algorithm[26]. The IRSPE essentially estimates the IR source position from the anchor width sequences. The results in Table 7 show that when the IR angle is relatively small (< 60°), the average IR position error is 0.075, which is 7× smaller compared with large IR angles (> 60°). The shadow stretching error of small angles is only around 0.27, showing that the IRSPE restores the human activity precisely. In contrast, the shadow stretching errors at large angles are 1.6× larger, so IRSPE cannot precisely restore the human skeleton in such

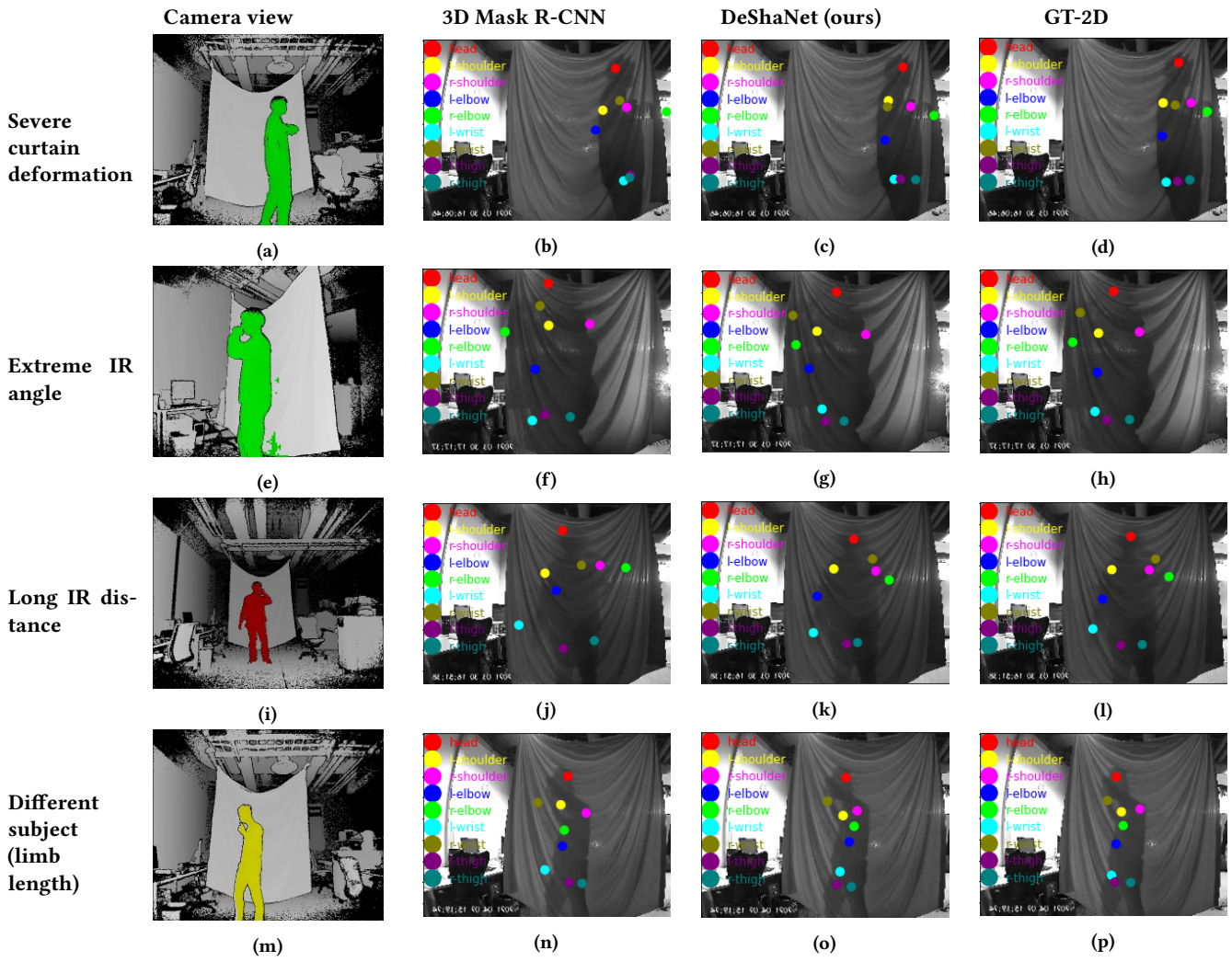

Figure 10: Example detection results of DeShaNet under severe deformation by various scene parameters. (a)-(d) Shadow keypoint detection results under severe curtain deformation. (e)-(h) Shadow keypoint detection results under extreme IR angle. (i)-(l) Shadow keypoint detection results under extreme IR angle. (m)-(p) Shadow keypoint detection results under extreme IR angle.

Table 4: Shadow keypoint detection error comparison on different IR source distances (pixels).

model	1.4m	2.0m	2.6m	3.4m	4.2m	average
3D Mask R-CNN	8.35	8.86	10.22	9.93	11.25	9.72
DeShaNet -CNN	6.25	7.04	7.52	7.35	9.45	7.52
DeShaNet	5.83	6.39	7.09	6.89	8.20	6.88

Table 5: Shadow keypoint detection error comparison on different IR source angles (pixels).

model	0°	15°	30°	45°	60°	average
3D Mask R-CNN	10.22	10.17	10.25	11.86	12.30	10.96
DeShaNet -CNN	7.52	7.03	7.96	8.32	11.56	8.47
DeShaNet	7.09	6.76	7.58	7.68	10.25	7.87

cases. The underlying reason is the detection error of DeShaNet and error of IRSPE both increases with angle.

On the other hand, the results in Table 8 show that the error of the IRSPE increases slightly with distance when it is within the margin value (< 4.2m). The IR position error and the stretching error are only 0.06 and 0.14. When the IR distance exceeds the margin value, the position error and stretching increase obviously to 0.65 and 0.33. This is mainly due to the low IR illumination intensity beyond the margin distance.

Evaluation of LLE. We further verify whether the LLE can accurately estimate the relative limb length of different subjects. Recall the shadow keypoints are detected by DeShaNet first, and then converted to shadow limb length, which is used by LLE to estimate the 3D relative limb length. The results in Table 9 show that the maximum relative limb length error is 0.034 (subject 3),

Table 6: Shadow keypoint detection error comparison on different subjects/limb length (pixels).

model	s1 1.85m	s2 1.73m	s3 1.75m	s4 1.70m	s5 1.80m	average
3D Mask R-CNN	10.22	10.56	9.89	10.13	9.35	10.03
DeShaNet -CNN	7.52	7.41	8.36	7.72	7.85	7.76
DeShaNet	7.09	6.88	7.56	7.12	7.26	7.18

Table 7: IR source position estimation error of the IRSPE on different angles.

metric	0°	15°	30°	45°	60°	average
IR position error (m)	0.06	0.04	0.08	0.12	0.56	0.17
Shadow stretching error (m)	0.15	0.21	0.29	0.45	0.71	0.36

Table 8: IR source position estimation error of the IRSPE on different distance.

metric	1.4m	2.0m	2.6m	3.4m	4.2m	average
IR position error (m)	0.01	0.04	0.06	0.13	0.65	0.17
Shadow stretching error (m)	0.08	0.13	0.15	0.21	0.33	0.18

which is only 6.2% larger than the lowest error (subject 1). It is worth noting that subject 3 is not the tallest among all ubjects, which indicates that the relative limb length error is not strongly related with the subject's height. We also find that the absolute limb length error of subject 3 is 0.024m, which is only 0.005m higher than the subject 1, indicating that LLE is robust to the limb length variation.

On the other hand, we find that the limb length error is mainly affected by moving speed. By inspecting the original video footage, we find that the movement speeds of subject 3 and 5 are relatively high, resulting in high keypoint detection errors of the DeShaNet (refer to Table 6), which in turn introduces noise on the input limb length.

Verifying the CDE. We further evaluate whether the CDE can accurately restore the curtain deformation. Since it is hard to quantify curtain deformation, we instead use the local moving speed v_l as an indirect metric to evaluate the effectiveness of the CDE. Our testing data includes 4 different curtain deformation patterns. The shadow keypoints of videos are first detected by the DeShaNet and then converted to local moving speed v_l. The CDE then estimates the vertex coordinates of the curtain from v_l. We use the flat-curtain (i.e., curtain deformation estimation is disabled) as a baseline. The results in Table 10 show that complex curtain patterns tend to produce higher moving speed error. The highest error (0.344) occurs under random deformation pattern, which is 3× larger than the flat curtain baseline (w/o). Compared with the baseline, the CDE shows 3.5× lower moving speed error and 2.8× lower keypoint error, indicating that CDE can faithfully restore the realistic curtain surface.

Table 9: Evaluation of the LLE on different subjects.

metric	s1 1.85m	s2 1.73m	s3 1.75m	s4 1.70m	s5 1.80m	average
relative limb length error	0.032	0.029	0.034	0.033	0.033	0.032
absolute limb length error (m)	0.019	0.017	0.024	0.022	0.023	0.021

Table 10: Evaluation of the CDE on different curtain deformation patterns.

metric	w/o	vertical	U-shape	random	average
moving speed error (m/s) w/o CDE	0.337	0.638	0.872	1.344	0.797
2D keypoint error (m) w/o CDE	0.027	0.032	0.037	0.041	0.034
moving speed error (m/s) /CDE	0.083	0.215	0.258	0.344	0.225
2D keypoint error (m) /CDE	0.008	0.011	0.013	0.018	0.012

Evaluation of the system assembly. We now evaluate the scene constructor with all SPEs enabled in more comprehensive situations. The final target of the scene constructor is to restore the 3D skeleton from the keypoint predictions of the DeShaNet. Multiple factors that could affect the restoration accuracy, including the occlusion of activity and the shadow deformation caused by scene parameters. Therefore, according to the intensity of occlusion and deformation, the testing data are divided into 5 groups: severe occlusion (oc-h), weak occlusion (oc-l), low deformation (de-l), medium deformation (de-m) and high deformation (de-h). The detailed categorization information is listed in Table 11. We use two metrics to evaluate the performance: S_{2D} and S_{3D}, which represents the shadow keypoint errors and the 3D skeleton errors, respectively. We then test the scene constructor on the groups of data and the results are shown in Table 12.

When only enabling IRSPE, LLE or CDE, the average S_{2D} are 25%, 8% and 17% lower than the baseline, and the average S_{3D} are 18% ,6% and 12% lower than the baseline, respectively. It indicates that all the design components play a crucial role in improving the estimation accuracy. It is worth noting that the IRSPE shows the largest improvement because the IR angles have the greatest impacts on the shadow keypoint detection. We obtain best results when enabling all SPEs together, with 36% lower S_{2D} and 26% lower S_{3D} than the baseline. It indicates that the 3 SPEs are all necessary and complementary to each other. We illustrate examples of shadow simulation effects and 3D skeleton estimation by different combinations of SPEs in Fig. 11-15.

7.3 Evaluation of other aspects

Comparison of different IR devices. Except from smart home security cameras, there are various other in-home devices that emit IR light, e.g., Kinect, smartphones and mobile lidars. These devices also have the potential of causing the IRSA. Table 13 summarizes the major characteristics of representative devices. Here the IR patterns refer to the patterns of the IR illumination, e.g., solid areas

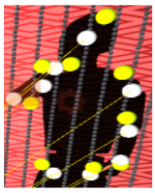

Figure 11: W/o IRSPE, LLE, CDE.

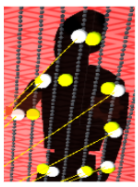

Figure 12: W/ IRSPE, w/o LLE, CDE.

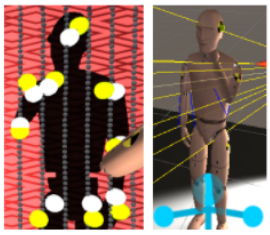

Figure 13: W/ IRSPE, LLE, w/o CDE.

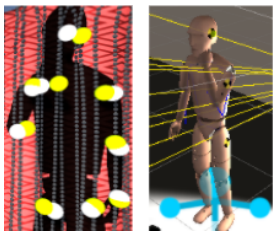

Figure 14: W/ IRSPE, LLE CDE.

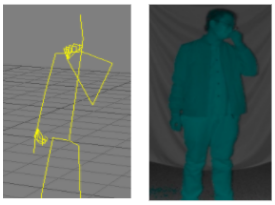

Figure 15: 3D skeleton ground-truth and realistic body activities under IR camera view.

Table 11: Testing data groups definition.

data	description
oc-l	bottom uplifting , body twisting
oc-h	nose picking, armpit stretching, eating (for a long time)
de-l	curtain deformation (w/o), IR distance (1-2.5mm), IR angles (0°)
de-m	curtain deformation (U-shape), IR distance (2.5-3.5m), IR angles (over 15-45°)
de-h	curtain deformation (random, vertical), IR distance (over 3.5m), IR angles (over 45°)

and spot patterns as shown in Fig. 16-17. Through the IR patterns, the attacker can infer what kind of devices victims are using, which may help estimate their activities more accurately. Both IR light patterns can project valid shadows and penetrate curtains.

Attack during daytime. Although the IRSA most happens at night due to the environmental illumination, we find that it can also be performed during daytime. For example, the IR light of smartphone cameras can be triggered at daytime when the indoor illumination is low, e.g., window curtains are closed, which provides opportunity for IRSA. We deploy such an attack scenario during daytime, as shown in Fig.9. The attacker's camera is placed beside the window under strong daylight which overwhelms the IR shadow, as shown in Fig.18. However, we find that the attacker could easily use an IR lens filter [2] to circumvent this challenge. The IR lens filter is low cost (<$8) and commonly used for photography and placed in front of the camera lens, which filters out the visible lights and only leaves the IR light with specific wavelength pass through. As shown in Fig. 19, by placing an 850nm IR lens filter, the shadow can be observed again in spite of the sunlight.

Curtain material and thickness. We evaluate how the curtain material and thickness would affect the IRSA. We test two materials, one made of 100% cotton and is opaque under normal indoor illumination, as shown in Fig. 8. The second is made of voile [4], which is half transparent under normal indoor illumination. During the test, we put some objects between the IR source and the curtain and observe the IR shadow on the other side of the curtain. Finally, we increase the distance between the IR source and the curtain until the shadow cannot be observed. Table 14 shows that the IR light can penetrate multiple layers of curtain of both materials at a reasonably long distance. Multiple layers of curtain are harder penetrate. Additionally, the voile curtain is easier to be penetrated than cotton, thus facing a higher privacy risk under IRSA.

7.4 Case study: recognizing private activities using the recovered 3D keypoints

In this section, we demonstrate how the 3D keypoints derived from DeShaNet can be used as input to existing skeleton based activity recognition algorithms and consequently impinge on user privacy. We adopt a representative algorithm, extremely randomized trees (ERT) [13], which follows 3 stages. (i) Data preprocessing. The coordinates of all the 3D keypoints are first normalized to a unified coordinate system with a predefined origin: $C_{origin} = (C_{l\_shoulder} + C_{r\_shoulder} + C_{head})/3$. Then a Savitzky–Golay filter with a 5-point cubic polynomial is applied to all the 3D skeletons to remove noise: $C_i^{'} = (-3 * C_{i-2} + 12 * C_{i-1} + 17 * C_i + 12 * C_{i+1} - 3 * C_{i+2})/35$, where C_i denotes the coordinates at frame i and $C_i^{'}$ is the filtered result. (ii) Spatio-temporal feature encoding. For spatial encoding, the 3D keypoint coordinates from the same frame are encoded into two matrices using Minkowski distances and cosine distance, respectively. For temporal feature encoding, each coordinate is encoded by two scalars $J_{i,max}$ and $J_{i,min}$, which are calculated by the difference between current coordinates and the maximum/minimum coordinates respectively. Then, each frame is further encoded by a vector with length $2 * N$, where N denotes the keypoint number (9 in our setup) of each frame . (iii) Random forest learning by the extremely randomized trees algorithm. The randomized trees perform frame level classification based on the spatial feature and the temporal feature. The final classification is derived by averaging the results from all trees. The total number of trees is 40 and the maximum depth is 20.

Since the source code of the ERT [13] is not publicly available, we implement it following [13] based on the *scikit-learn* python library. We have validated our reproduction of ERT on the Microsoft MSR action 3D dataset [22] adopted in [13] and got consistent accuracy (82.1% vs 80.9%), which verifies the correctness of our implementation. In order to evaluate the accuracy of activity recognition, we divide the dataset we collected (Section 6) into 3 categories, for training, validation and testing, respectively. The training set and validation set contain the same categories of activities but are orthogonal to each other: eating, running, walking, dancing and stretching exercise. For subset, we further divide it into multiple subsets, one subject each. The testing set contains 3 different activities (i.e., nose picking, body twisting and bottom uplifting) from the training and validation set, and is used to test the model generalization. For each entry in the dataset, we have converted it to 3D skeletons using the DeShaNet and scene constructor. Table 15 summarizes the results. We see that the average activity recognition accuracy is 87.9% and 83.4%, on the validation and testing set

Table 12: Evaluation of the scene constructor on different occlusion and deformation.

model	oc-l		oc-h		de-l		de-m		de-h		average	
	S_{2D}	S_{3D}	S_{2D}	S_{3D}	S_{2D}	S_{3D}	S_{2D}	S_{3D}	S_{2D}	S_{3D}	S_{2D}	S_{3D}
w/o IRSPE, LLE, CDE	0.198	0.145	0.213	0.158	0.201	0.137	0.219	0.146	0.258	0.161	0.217	0.149
w/ CDE, w/o IRSPE, LLE	0.153	0.114	0.186	0.145	0.163	0.113	0.186	0.132	0.215	0.148	0.180	0.130
w/ LLE, w/o IRSPE, CDE	0.165	0.127	0.208	0.151	0.181	0.122	0.208	0.139	0.234	0.156	0.199	0.139
w/ IRSPE, w/o LLE, CDE	0.127	0.096	0.173	0.138	0.148	0.106	0.175	0.129	0.186	0.139	0.161	0.121
w/ IRSPE, LLE, CDE	0.107	0.076	0.158	0.131	0.113	0.083	0.149	0.122	0.167	0.133	0.138	0.109

Figure 16: IR light patterns of mobile Lidar (Intel L515).

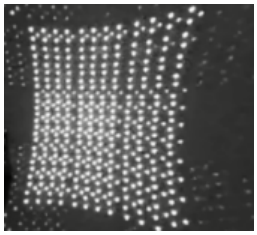

Figure 17: IR light patterns of iPhone 11 Pro.

Figure 18: Camera view at day time without using IR filter.

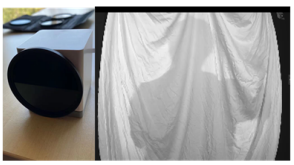

Figure 19: Camera view at day time using IR filter (850nm).

Table 13: Comparisons of different IR devices.

device	IR pattern	maximum distance	trigger type
Kinect	solid	~ 7m	when used
iPhone 11 Pro	sparse spots	~ 4.5m	when used
Intel L515	dense spots	~ 5m	when used
smart home camera (Wyze)	solid	~ 6m	dark illumination

Table 14: Maximum penetration distance of different curtain materials and thickness. Cot-1 refers 1 layer curtain of cotton material. Voi-1 refers 1 layer curtain of voile material.

cot-1	cot-2	voi-1	voi-4	voi-8
7m	3m	>10m	5m	2m

respectively, which is consistent with [13]. Note that the keypoints in [13] where obtained through the Kinect 3D camera. The result implies that the 3D keypoints generated by our DeShaNet and scene constructor are sufficiently accurate for recognizing fine-grained activities that involve body/limb movements.

7.5 Generalization to strongly private activities

In this section, we show that the DeShaNet keypoint generator works for both generic activities and private concerning activities. We divide our dataset into weak privacy activities and real privacy activities. The former include eating, running, walking, dancing, stretching exercise and body twisting. The latter include nose picking and bottom uplifting (simulating sex behavior). Based on the user study in [9], over 24% of people think that the exposure of these two kinds of activities are extremely private. Then we test the 2D keypoint detection errors under different scene factors: curtain

Table 15: 3D skeleton based activity classification accuracy using extremely randomized trees algorithm [13].

	Training	Validation	Testing
Subject1	91.2%	87.9%	83.4%
Subject2	93.6%	88.7%	84.1%
Subject3	90.4%	87.2%	82.6%
Average	91.7%	87.9%	83.4%

deformation, IR distance, IR angle and subjects. Other experimental configurations are consistent with Sec. 7.1.

Table 16 summarizes the results, where the weak and real activities are denoted by "we" and "re", respectively. We see that the 2D keypoint detection errors of weak and real privacy activities are similar under different experimental setups. The average error of weak privacy activities is 6.97, which is only 2.4% lower than real privacy activities. It indicates that our model can generalize to real privacy-concerning activities. The underlying reason is simple. Both the weak and real privacy activities share similar body movement, which mainly focus on the upper body and limbs, such as hands, shoulders, arms and legs. Therefore, the two categories of activities do not have essential differences with respect to the DeShaNet keypoint generation, which ultimately results in similar recognition accuracy.

8 RELATED WORK

Privacy threats for smart homes. Privacy at home has always been a concern for many people, although most people are unaware of the potential sources of threats. Choe et al. [9] conducted a survey which revealed over 1400 private behaviors/activities that people do not want to be exposed at home. Zheng et al. [47] investigated people's awareness of smart home devices' capabilities, and found that most people failed to pay attention to the potential security/privacy threats. As the smart home ecosystem evolves, new privacy threats begin to emerge, often relying on novel techniques. For instance,

Table 16: Comparison of weak and real privacy-concerning activities (pixels).

model	deformation		IR disatance		IR angle		subject		average	
	we	re	we	re	we	re	we	re	we	re
3D Mask R-CNN	9.18	9.03	9.72	9.83	10.96	10.67	10.22	10.88	10.02	10.1
DeShaNet-CNN	6.67	6.52	7.52	7.42	8.47	8.52	7.52	7.35	7.55	7.45
DeShaNet	6.03	6.46	6.88	6.92	7.87	8.21	7.09	7.33	6.97	7.14

LiShield [48] addresses the privacy leakage due to unauthorized cameras, by using a smart LED to corrupt the camera image sensor. Sami et al. [31] use the lidar on sweeping robots to detect tiny vibration of objects caused by speech and in turn decode the speech. However, such attacks require hacking into smart home devices. Xu et al. [41] showed that TV illumination projected on window curtains can expose the TV content that people are watching. In contrast, the IRSA attacker does not need to access any devices in the subject's home, but can still reveal the subject's physical activities at home, thus posing a greater threat.

Shadow detection based applications. In computer vision applications, shadows are usually regarded as image noise, so previous related work mainly studied how to remove shadows from images. Zheng et al. [46] proposed a distraction-aware shadow detection scheme to remove ambiguous shadows where the visual appearances of shadow and non-shadow regions are similar. Wang et al. [39] further use generative adversarial networks (GAN) to accurately remove shadows. Recently, visible light shadow has also been leveraged in visual sensing applications. For instance, Li et al. [19] realized sparse body skeleton detection (5 joints in total) through shadows projected on the floor. In addition, they also used ordinary table lamp shadows to identify hand poses [20]. Meanwhile, Nguyen et al. [24] used ceiling light shadows for coarse-grained human occupancy detection. In contrast, the proposed IRSA needs to accurately reconstruct 3D body keypoints from shadows, and faces a unique challenge of shadow deformation.

Video keypoint detection. Keypoint detection has always been an active research branch in computer vision. Early solutions [7] focused on real-time multi-person keypoint detection. The 3D Mask R-CNN model [11] represents the state-of-the-art in terms of detection accuracy. Various aspects of the keypoint detection tasks have been further explored, such as solving severe occlusion [8] and deformation [34]. These solutions mainly leverage prior knowledge of the human body structure. More recently, an unsupervised keypoint detection scheme [17] was proposed to eliminate the need for labeled data. In addition, Mehta et al. [21] propose to predict 3D skeletons from RGB videos directly. However, existing keypoint detection schemes are all based on RGB videos, which cannot be directly applied to shadow keypoint detection in IRSA. This is because the prior knowledge of human body structures is not as informative for shadows, especially when the projection surface (e.g., window curtains) severely deforms the shadows.

9 DISCUSSION

Defensing mechanisms against the IRSA. A straightforward method to prevent the IRSA is to ensure the curtain and window fall outside the security camera's field of view, so that no IR shadow can be projected towards the curtain surface. However, not all the ordinary users would be aware of IRSA, so it is highly desirable to prevent it from the source, i.e., security cameras and other IR devices. One potential solution is to require that the IR light source emit special light patterns, instead of the simple solid or dot patterns. The IR source can periodically project random light patterns which are known only to the legitimate camera (often co-located with the light source). Each pattern only covers parts of the field-of-view, and different patterns are complementary to each other in space. Then the legitimate camera assembles all the image frames within one period to reconstruct a complete frame. From the attacker's view, it is infeasible to acquire complete shadows because only a small parts of the shadow are created each time.

System Limitations. Although we have extensively evaluated the IRSA over a variety of situations, there still exist some limitations. First, the current attack system is only applicable on a single subject, as the DeShaNet only supports single person shadow detection. This limitation can potentially be solved by fusing the tube proposal module of the 3D Mask R-CNN with DeShaNet. Second, the keypoint coverage is low. Currently, there are only 9 keypoints in total, which may not be enough for higher precision activity detection, such as finger motion. A straightforward solution is increasing the keypoint quantity in the DeShaNet. However, we think the essential problem is the difficulty of detecting the finger from the severely deformed shadow, which we leave for future exploration.

10 CONCLUSION

We have demonstrated the IRSA, a new privacy leakage threat caused by common smart home camera devices with a night vision mode. We have studied various environmental factors that may hinder the attack, including the curtain deformation, IR distance/angles and limb length. We further propose the DeShaNet and scene constructor to recover the subtle 3D skeletons from deformed IR shadows, which reveal the victim's behaviors in a more delicate way. We hope that this study can draw people's attention on the invisible IR side channel that security camera (or other IR light sources such as Kinect) leaks, which can cause severe privacy issues. In addition, we believe the manufacturers of indoor security cameras need to act immediately to install the defense mechanisms to thwart IRSA.

ACKNOWLEDGMENTS

This research was supported in part by the National Natural Science Foundation of China under Grant No. 62122095, 62072472 and U19A2067, Natural Science Foundation of Hunan Province, China under Grant No. 2020JJ2050, 111 Project under Grant No. B18059, and the Young Talents Plan of Hunan Province of China under Grant No. 2019RS2001.

REFERENCES

[1] Amazon. 2021. 360 AC1C camera. (2021). https://www.amazon.com/360-Security-Recognition-Detection-Activity/dp/B089W4PKRW/

[2] Amazon. 2021. IR filter. (2021). https://www.amazon.com/gp/product/B015XMSWIQ/

[3] Amazon. 2021. Top selling smart cameras on Amazon. https://www.amazon.com/s?k=smart+cameras&s=review-rank&qid=1625164398&ref=$sr\_st\_review-rank$. (2021).

[4] Amazon. 2021. Voile curtain. (2021). https://www.amazon.com/gp/product/B0155EB71Q/

[5] Amazon. 2021. Wyze camera. (2021). https://www.amazon.com/Wyze-Indoor-Wireless-Detection-Assistant/dp/B076H3SRXG/

[6] Brekel. 2021. Brekel Body v2. (2021). https://brekel.com/body_v2/

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7291–7299.

[8] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. 2019. Occlusion-aware networks for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 723–732.

[9] Eun Kyoung Choe, Sunny Consolvo, Jaeyeon Jung, Beverly Harrison, and Julie A Kientz. 2011. Living in a glass house: a survey of private moments in the home. In *Proceedings of the 13th international conference on Ubiquitous computing*. 41–44.

[10] Rohit Girdhar. 2018. 3D Mask R-CNN. (2018). https://rohitgirdhar.github.io/DetectAndTrack/

[11] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. 2018. Detect-and-track: Efficient pose estimation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 350–359.

[12] Grandviewresearch. 2019. Window Covering Market Size. https://www.grandviewresearch.com/industry-analysis/window-covering-market. (2019).

[13] Y. Hbali, S. Hbali, L. Ballihi, and M. Sadgal. 2017. Skeleton-based human activity recognition for elderly monitoring systems. *IET Computer Vision* 12, 1 (2017), 16–26.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[15] Yumi Iwashita, Adrian Stoica, and Ryo Kurazume. 2010. People identification using shadow dynamics. In *IEEE International Conference on Image Processing*.

[16] JD. 2021. Top selling smart cameras on JD.COM. https://search.jd.com/Search?keyword=smartcamera. (2021).

[17] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. 2019. Unsupervised keypoint learning for guiding class-conditional video prediction. *arXiv preprint arXiv:1910.02027* (2019).

[18] Jia Li, Wen Su, and Zengfu Wang. 2020. Simple pose: Rethinking and improving a bottom-up approach for multi-person pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11354–11361.

[19] Tianxing Li, Chuankai An, Zhao Tian, Andrew T Campbell, and Xia Zhou. 2015. Human sensing using visible light communication. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 331–344.

[20] Tianxing Li, Xi Xiong, Yifei Xie, George Hito, Xing-Dong Yang, and Xia Zhou. 2017. Reconstructing hand poses using visible light. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–20.

[21] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2020. XNect: Real-time multi-person 3D motion capture with a single RGB camera. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 82–1.

[22] Microsoft. 2009. MSR Action 3D dataset. https://www.microsoft.com/en-us/download/details.aspx?id=52315. (2009).

[23] Msn. 2020. Average house size around the world. https://www.msn.com/en-in/lifestyle/smart-living/how-big-is-the-average-house-size-around-the-world/ar-AAdKEhh. (2020).

[24] Viet Nguyen, Mohamed Ibrahim, Siddharth Rupavatharam, Minitha Jawahar, Marco Gruteser, and Richard Howard. 2018. Eyelight: Light-and-shadow-based occupancy estimation and room activity recognition. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 351–359.

[25] Obi. 2021. Obi cloth. (2021). https://assetstore.unity.com/packages/tools/physics/obi-cloth-81333

[26] OpenCV. 2021. opencv contour. (2021). https://docs.opencv.org/master/dd/d49/tutorial_py_contour_features.html

[27] Pytorch. 2020. Pytorch website. (2020). https://pytorch.org/

[28] M.W. Rahman and M.L. Gavrilova. 2017. Kinect gait skeletal joint feature-based person identification. In *IEEE International Conference on Cognitive Informatics & Cognitive Computing*.

[29] Grand View Research. 2021. Smart Home Security Cameras Market Size. https://www.grandviewresearch.com/industry-analysis/smart-home-security-camera-market. (2021).

[30] Worldpopulation review. 2021. World population. https://worldpopulationreview.com/. (2021).

[31] Sriram Sami, Yimin Dai, Sean Rui Xiang Tan, Nirupam Roy, and Jun Han. 2020. Spying with your robot vacuum cleaner: eavesdropping via lidar sensors. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 354–367.

[32] Makoto Shinzaki, Yumi Iwashita, Ryo Kurazume, and Koichi Ogawara. 2015. Gait-Based Person Identification Method Using Shadow Biometrics for Robustness to Changes in the Walking Direction. In *IEEE Winter Conference on Applications of Computer Vision*.

[33] Strategyanalytics. 2019. Smart home surveillance camera market forecast and analysis. https://www.strategyanalytics.com/access-services/devices/connected-home/smart-home/reports/report-detail/2019-smart-home-surveillance-camera-market-forecast-and-analysis. (2019).

[34] Masanori Suganuma, Xing Liu, and Takayuki Okatani. 2019. Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9039–9048.

[35] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215* (2014).

[36] Unity. 2021. PuppetMaster. (2021). https://assetstore.unity.com/packages/tools/physics/puppetmaster-48977

[37] Unity. 2021. Unity 3D. (2021). https://unity.com/

[38] Edward J Wang, William Li, Junyi Zhu, Rajneil Rana, and Shwetak N Patel. 2017. Noninvasive hemoglobin measurement using unmodified smartphone camera and white flash. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2333–2336.

[39] Jifeng Wang, Xiang Li, and Jian Yang. 2018. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1788–1797.

[40] C. C. Wei, L. H. Tsai, H. P. Chou, and S. C. Chang. 2020. *Person Identification by Walking Gesture Using Skeleton Sequences*. Advanced Concepts for Intelligent Vision Systems.

[41] Yi Xu, Jan-Michael Frahm, and Fabian Monrose. 2014. Watching the watchers: Automatically inferring tv content from outdoor light effusions. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 418–428.

[42] Youtube. 2019. Night vision trigger test 1 of different smart cameras. https://www.youtube.com/watch?v=hx5k4dIbT3Y. (2019).

[43] Youtube. 2019. Night vision trigger test 2 of different smart cameras. https://www.youtube.com/watch?v=hx5k4dIbT3Y. (2019).

[44] Youtube. 2021. How to see through material with a Night Vision Camcorder. (2021). https://www.youtube.com/watch?v=RdtJlHVDcmM

[45] Youtube. 2021. Infra-X-Vision. (2021). https://www.youtube.com/watch?v=9DilYUiu4AQ

[46] Quanlong Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. 2019. Distraction-aware shadow detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5167–5176.

[47] Serena Zheng, Noah Apthorpe, Marshini Chetty, and Nick Feamster. 2018. User perceptions of smart home IoT privacy. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–20.

[48] Shilin Zhu, Chi Zhang, and Xinyu Zhang. 2017. Automating Visual Privacy Protection Using a Smart LED. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*.