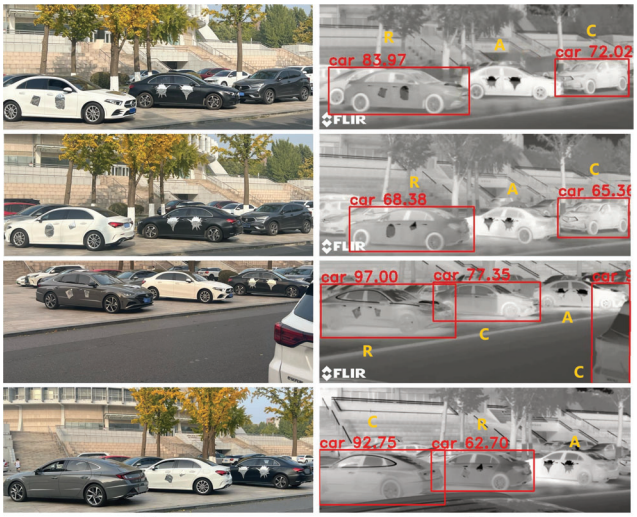
Infrared Adversarial Car Stickers

Xiaopei Zhu[1] Yuqiu Liu[2] Zhanhao Hu[3] Jianmin Li[4] Xiaolin Hu[4,5,6*]
[1]School of Integrated Circuits, Tsinghua University, Beijing, China
[2]Department of Technology, Beijing Forestry University, Beijing, China
[3]Department of Electrical Engineering and Computer Sciences, UC Berkeley, California, USA
[4]Department of Computer Science and Technology, Institute for Artificial Intelligence, BNRist, Tsinghua University, Beijing, China
[5]THBI, IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing, China
[6]Chinese Institute for Brain Research (CIBR), Beijing, China
{zxp18}@mails.tsinghua.edu.cn
{liuyuqiu99, zhanhaohu.cs}@gmail.com
{lijianmin, xlhu}@mail.tsinghua.edu.cn

Abstract

Infrared physical adversarial examples are of great significance for studying the security of infrared AI systems that are widely used in our lives such as autonomous driving. Previous infrared physical attacks mainly focused on 2D infrared pedestrian detection which may not fully manifest its destructiveness to AI systems. In this work, we propose a physical attack method against infrared detectors based on 3D modeling, which is applied to a real car. The goal is to design a set of infrared adversarial stickers to make cars invisible to infrared detectors at various viewing angles, distances, and scenes. We build a 3D infrared car model with real infrared characteristics and propose an infrared adversarial pattern generation method based on 3D mesh shadow. We propose a 3D control points-based mesh smoothing algorithm and use a set of smoothness loss functions to enhance the smoothness of adversarial meshes and facilitate the sticker implementation. Besides, We designed the aluminum stickers and conducted physical experiments on two real Mercedes-Benz A200L cars. Our adversarial stickers hid the cars from Faster RCNN, an object detector, at various viewing angles, distances, and scenes. The attack success rate (ASR) was 91.49% for real cars. In comparison, the ASRs of random stickers and no sticker were only 6.21% and 0.66%, respectively. In addition, the ASRs of the designed stickers against six unseen object detectors such as YOLOv3 and Deformable DETR were between 73.35%-95.80%, showing good transferability of the attack performance across detectors.



(a) (b)

Figure 1. Infrared attack effect on real cars. (a) Visible light view of real cars. (b) Infrared view of real cars. C: clean car. R: car with random shape stickers. A: car with adversarial stickers. The numbers above the bounding boxes are object confidence scores (%) with 0.6 threshold. Our adversarial stickers hid the car from Faster RCNN at various viewing angles, distances and scenes. In comparison, the clean car and the car with random shape stickers were detected at the same situation.

1. Introduction

It is well known that deep learning models can be misled by carefully designed perturbations to the input which is called *adversarial example*, and the perturbation process is called *adversarial attack*. Adversarial attacks can be di-

*Corresponding author.

vided into two categories, digital attacks [4, 8, 9, 16, 24, 26] which assume that the attackers can directly modify the model input in the digital world, and physical attacks [11, 12, 27, 28, 35, 37] which assume that the attackers can only modify the object or scene in the physical world. Physical attacks have attracted much attention because of their importance of assessing the security of real-world AI systems.

One type of physical attack is called *infrared physical attack* [30–32, 35, 37]. Infrared imaging is widely used in our daily lives, such as body temperature monitoring and autonomous driving. Since infrared cameras can function normally at night, they are becoming more and more important in autonomous driving systems; so is their safety.

Previous infrared physical attacks [30–32, 35, 37] mainly focused on infrared pedestrians, and only two works [31, 32] conducted experiments on model cars. There is currently a lack of infrared attack research on real cars. The reasons might be as follows. Compared with pedestrians with constant body temperature, the temperature distributions of real cars are more uneven (e.g., the temperature of the engine is much higher than that of other places), so their infrared characteristics are more complex than pedestrians. Compared with model cars without engines, real cars' structures and materials are very different from those of model cars, so their infrared characteristics are also quite different. Besides, real car attacks require designing and manufacturing adversarial patterns on the entire 3D car surface, which poses a great challenge to physical experiments. But we believe that for the safety of autonomous driving cars, physical real car attack is worth in-depth investigation.

The aforementioned two infrared model car attacks [31, 32] are only effective within limited viewing angles (e.g., horizontal angles -30° $-$ 30° and pitch angle 0°). But we want to implement a *full-angle* attack (the horizontal angles 0° $-$ 360°, and pitch angle 0° $-$ 90°), so the attack angles cover an entire hemisphere surface, which is a challenging task. We also notice that these methods [31, 32] are both case-by-case attacks, which needs to optimize an adversarial pattern for each image[1], while our goal is to achieve a universal attack which uses the same adversarial pattern to attack detectors at various viewing angles, distances and scenes.

Towards this goal, we propose an infrared physical attack method applied to a real car based on 3D modeling. We aim to design a set of infrared adversarial stickers to make cars invisible to infrared detectors at various viewing angles (*full-angle*), distances and scenes. Since most current 3D car models are visible-light models, and there is a lack of 3D infrared car models, we build a 3D infrared car model with real infrared characteristics at various viewing angles. For the generation of infrared adversarial pattern for stickers, we propose to optimize a 3D adversarial mesh at first, then project the shadow of 3D adversarial mesh to obtain a 2D adversarial pattern, and finally attach the 2D adversarial pattern to the car surface. The motivation of this mesh shadow attack (MSA) method is that we hope to find a better solution in a higher-dimensonal 3D space instead of directly optimizing the 2D adversarial pattern. To improve the smoothness of adversarial patterns and facilitate sticker implementation, we propose a 3D control points-based mesh smoothing algorithm and use a set of smoothing loss functions.

For the physical implementation of infrared adversarial patterns, we use an aluminum film which modifies the surface emissivity of the object instead of altering the surface temperature used by previous works [30–32, 35, 37]. Like many car stickers, the adversarial car stickers can be easily attached on the car surface. The stickers are only 0.08mm thick and take up almost no space.

To assess the safety of infrared detection in real autonomous driving scenes, we used two real Mercedes-Benz A200L cars. Physical experiments show that our infrared adversarial stickers made the real cars hide from the infrared detector Faster RCNN at various viewing angles, various distances, and multiple scenes, with an attack success rate (ASR) of 91.49%. To the best our knowledge, this is the first 3D multi-view physical infrared vehicle attack, and also the first infrared attack conducted on real cars.

2. Related Work

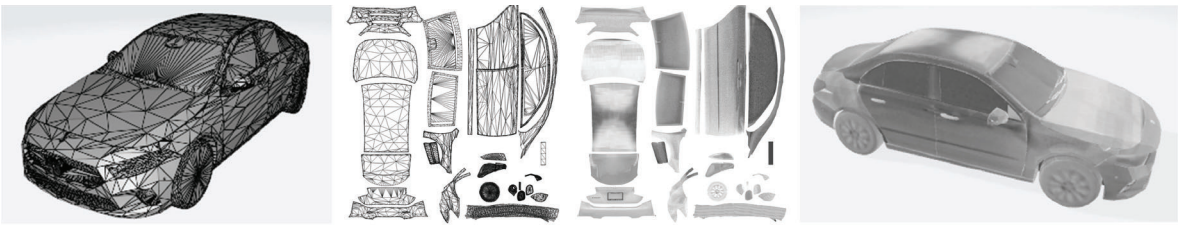
2.1. Visible Light Physical Adversarial Attacks

Huang et al. [13] proposed a universal physical camouflage (UPC) attack for object detectors. Zhang et al. [34] proposed a vehicle camouflage for physical adversarial attack on object detectors in the wild. Wang et al. [29] proposed the Dual attention suppression (DAS) attack to generate adversarial vehicle camouflage in the physical world. Suryanto et al. [25] generated the physical adversarial camouflage by using a differentiable transformation network. Wang et al. [28] proposed a 3D full-coverage vehicle camouflage for physical adversarial attack (FCA). It is worth noting that all above works are proposed for visible light images.

2.2. Infrared Physical Adversarial Attacks

Zhu et al. [35] proposed a bulb-based board to fool infrared pedestrian detectors in the physical world. Zhu at al. [37] proposed an infrared invisible clothing to hide from infrared pedestrian detectors in the physical world. Wei et al. [30] proposed the HotCold blocks to attack the infrared pedestrian detectors. Wei et al. [32] proposed a physical adversarial infrared patch (AIP) based on a points-clustering algorithm. Wei et al. [31] proposed a unified adversar-

[1]We found this by checking and running their official codes.



(a) (b) (c) (d)

Figure 2. Construction and optimization of real infrared car texture mapping. (a) Car mesh model. (b) Reorganized faces map. (c) Infrared car texture map collected from real world. (d) Rendered infrared car model.

ial patch (UAP) for physical attacks based on a boundary-limited shape optimization algorithm. It is worth noting that all above methods are proposed for infrared pedestrian or model car attacks. There is currently a lack of research on infrared real vehicle attacks in the physical world.

3. Car Sticker Attack Method

Our method consist of several steps. First, we build a 3D infrared car model based on real infrared characteristics. Next, we use the infrared adversarial pattern generation method based on 3D mesh shadow. To make the 3D adversarial mesh smoother, we use a 3D control points-based mesh smoothing algorithm and use a set of smoothness losses. Then we apply adversarial patterns to 3D infrared car model and optimize the adversarial patterns. Finally, we introduce the physical implementation method of infrared adversarial car stickers.

3.1. Building a 3D Infrared Car Model

To simulate the infrared car attack realistically, we build a 3D model based on the infrared characteristics of a real car. It is worth noting that our method can be applied to any target car, and we chose Mercedes-Benz A200L in our experiments, simply because one of the authors have this car. Figure 2(a) shows the car mesh model M_{car}. Next, we need to create a "skin" for the car model based on infrared photos taken by an infrared camera FLIR T560. However, the infrared photos captured by the camera are all 2D images, and the challenge is how to "paste" these 2D infrared images onto the 3D car mesh model. First, we flatten all the faces of 3D car mesh onto a 2D plane called *faces map*. After that, we use MAYA software to rearrange these faces to divide different areas, such as roof, doors, etc., as shown in Figure 2(b). This process facilitates the alignment of real infrared car images with the 3D car mesh.

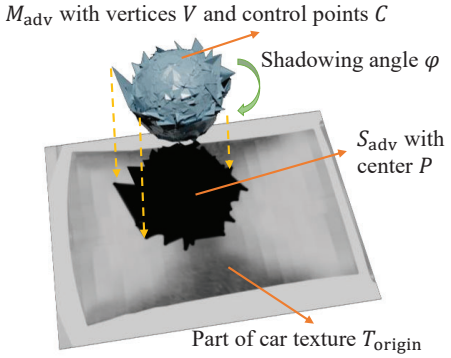


Figure 3. Schematic diagram of mesh shadow method.

Subsequently, we crop the infrared images into different parts based on the *faces map* (Figure 2(b)) and paste the cropped images onto the 2D *faces map*, and then we get the infrared *texture map* of the car, as shown in Figure 2(c). See *Supplementary Material (SM)* for how these infrared photos are taken, cropped and pasted. This process establishes a correspondence between the real infrared car images and the 3D car mesh. The rendered infrared car model is shown in Figure 2(d), which is built for a real car with engine running.

3.2. Generation of 2D Shadow Based on 3D Mesh

We aim to design infrared stickers with adversarial patterns to hide the cars from infrared detectors at various viewing angles (*full-angle*), distances, and scenes. We propose a 3D adversarial mesh shadow attack (MSA) method to generate the 2D infrared adversarial patterns for stickers. The motivation of MSA method is that we hope to find a better solution in a higher-dimensonal 3D space instead of directly optimizing the 2D adversarial pattern. The core of MSA method is to optimize a 3D adversarial mesh M_{adv} at first, then project the shadow of 3D adversarial mesh to obtain a 2D adversarial pattern S_{adv}, and finally attach the 2D adversarial pattern S_{adv} to the car surface. Note that the 3D adversarial mesh M_{adv} is different from the 3D car mesh M_{car} in Section 3.1. The optimization variables include the 3D mesh vertices coordinates V, the mesh shadowing angle φ, and the center point position P when pasting the 2D pattern onto the car's texture map T_{origin} (Figure 3).

The shadowing operation refers to rendering the mesh

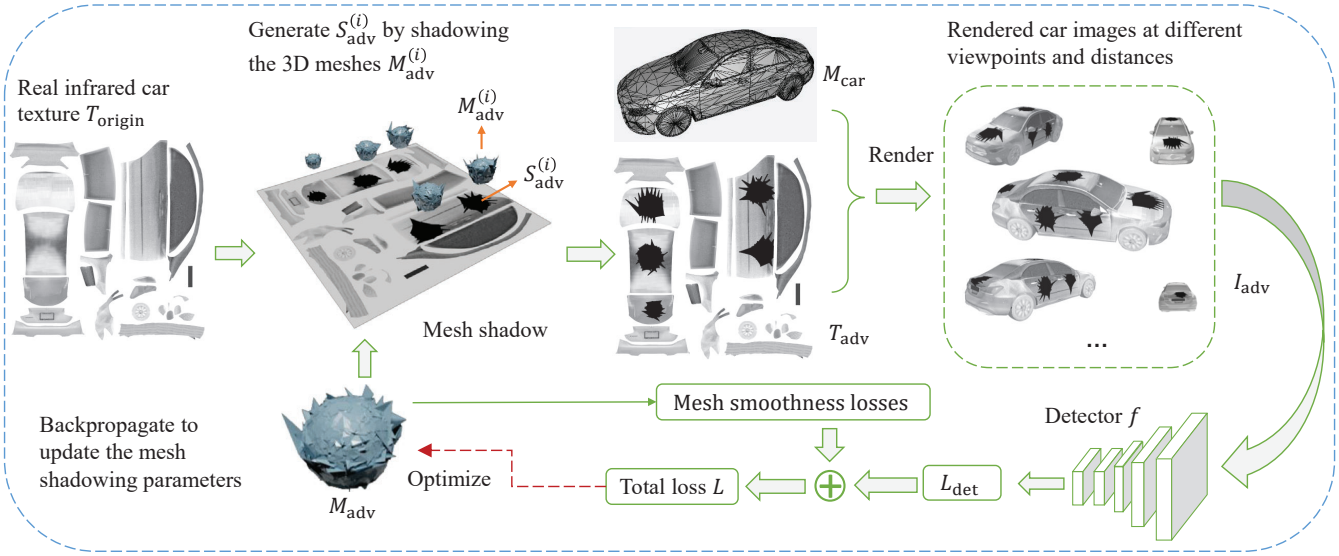


Figure 4. The overall pipeline of the proposed method.

M_{adv} to a dark area while retaining the mesh contour (Figure 3), and the dark area has a uniform grayscale value within the contour. The grayscale value is consistent with the infrared characteristics of the sticker we use. Let Ω denote this operation. If we want the car to have adversarial effect at various viewing angles, we need to optimize N adversarial meshes to generate N adversarial shadows on different places of the car (Figure 4):

$$S_{\text{adv}}^{(i)} = \Omega(M_{\text{adv}}^{(i)}, \varphi^{(i)}), i = 1, 2, ...N. \tag{1}$$

3.3. 3D Control Points-Based Mesh Smoothing

If we directly optimize the vertices coordinates V of the mesh M_{adv}, many "peaks" will appear on the mesh surface, which will make the shadow shape very complex and will be difficult for the physical implementation of the adversarial shadow S_{adv}. Inspired by the Gaussian smoothing kernel and spline interpolation method, we propose a smoothing control algorithm for 3D mesh vertices. Its core idea is to use some 3D control points C as anchor points, and the coordinate offsets of mesh vertices V are expressed as the weighted average of the coordinate offsets of C. The calculation details are described in *SM*.

We use Θ to denote the above transformation, and $C^{(i)}$ to denote the control points set of $M_{\text{adv}}^{(i)}$, then

$$V^{(i)} = \Theta(C^{(i)}), i = 1, 2, ...N. \tag{2}$$

Since the control points $C^{(i)}$ determine the vertices coordinates $V^{(i)}$, the optimization variables changes from $\left(V^{(i)}, P^{(i)}, \varphi^{(i)}\right)$ to $\left(C^{(i)}, P^{(i)}, \varphi^{(i)}\right), i = 1, 2, ...N$.

3.4. Mesh Smoothness Loss Functions

To further enhance the smoothness of 3D adversarial mesh M_{adv} and 2D adversarial pattern S_{adv}, we use a set of loss functions including: *mesh normal consistency loss*, *mesh edge loss*, *chamfer distance loss*, and *Laplacian smoothing loss*. During the optimization process of 3D mesh M_{adv} , these functions guide the generation of a smoother adversarial mesh. A smoother 3D mesh M_{adv} results in a smoother 2D shadow pattern S_{adv}, which is beneficial for manufacturing physical stickers based on the 2D shadow pattern.

The *mesh normal consistency loss* computes the normal consistency for each pair of neighboring faces, and minimizing it encourages the mesh surface to be smoother. We suppose that mesh M_{adv} has a total of F faces. Let n_i, n_j $(1 \leq i, j \leq F)$ represent the normal vector of any two adjacent faces, then this loss function is described as:

$$L_{\text{norm}} = \text{Average}\left(1 - \cos(n_i, n_j)\right). \tag{3}$$

Average is calculated between any pair of adjacent faces.

The *mesh edge loss* computes mesh edge length regularization loss, and minimizing it encourages a reduction in the average edge length of the adversarial mesh. Suppose M_{adv} has a total of M edges, l_k $(k = 1, 2, ..., M)$ represents the length of each edge, and this loss function is described as

$$L_{\text{edge}} = \left(\sum_{k=1}^{M} l_k\right)/M. \tag{4}$$

The *chamfer distance loss* computes chamfer distance [6] between the sampling points of adversarial mesh and a standard sphere. Reducing chamfer distance loss encourages the adversarial mesh M_{adv} to approximate a sphere.

We denote the point clouds sampled by adversarial mesh M_{adv} and the standard spherical mesh M_{sphere} by S_1 and S_2, respectively. The chamfer distance loss is described as

$$L_{\text{chamfer}} = \sum_{p \in S_1} \min_{q \in S_2} \|p - q\|_2^2 + \sum_{q \in S_2} \min_{p \in S_1} \|q - p\|_2^2 . \tag{5}$$

The *Laplacian smoothing loss* L_{Laplace} computes the Laplacian smoothing objective for the adversarial mesh. We define this function as Laplace, and its calculation details are introduced in [18], so

$$L_{\text{Laplace}} = \text{Laplace}\left(M_{\text{adv}}\right). \tag{6}$$

3.5. Applying the 2D Shadow to 3D Car Model

We apply the adversarial shadow S_{adv} to 3D infrared car model by changing its texture map T_{origin}. This simulates the process we paste the adversarial stickers to the car surface in the physical world. To facilitate physical implementation, we simulate to paste the stickers onto the door, roof, front hood and rear of the car, which have wide ranges of flat area and are easy to paste. In each area, we paste one or two adversarial shadow patterns (Figure 4). To simulate real-world perturbations, such as errors in cutting the adversarial stickers, variations in surface temperature on the adversarial stickers, and errors in the pasting positions, we introduce random perturbations to the vertex coordinates V of the adversarial mesh M_{adv}, random noise to the pattern S_{adv}, random changes in grayscale values of S_{adv}, and random perturbations in the position P during the pasting of S_{adv}. This approach, known as the Expectation Over Transformation (EOT) algorithm [2], enhances the robustness of our algorithm in real-world scenes.

Next, we need to paste the adversarial patterns S_{adv} onto the original car texture map T_{origin}. Let Γ denote the pasting operation. We establish a Cartesian coordinate system with the center point of T_{origin} as the origin. We use $P^{(i)}$ to represent the coordinates of the pasting positions for N adversarial shadows $S_{\text{adv}}^{(i)}, i = 1, 2, ...N$. The texture map after pasting the adversarial shadows is

$$T_{\text{adv}} = \Gamma\left(S_{\text{adv}}^{(i)}, P^{(i)}, T_{\text{origin}}\right), i = 1, 2, ...N. \tag{7}$$

We use the differentiable renderer Pytorch3D [20] to render the adversarial texture map T_{adv} onto the surface of the car mesh M_{car}, resulting in the rendered infrared car images with adversarial patterns, denoted as I_{adv}. Let Ψ denote the rendering operation with parameters θ which include the rendering distances and angles. Mathematically, this process can be expressed as:

$$I_{\text{adv}} = \Psi\left(M_{\text{car}}, T_{\text{adv}}, \theta\right). \tag{8}$$

3.6. Optimization of 3D Mesh Shadow Attack

After we get the rendered infrared adversarial images I_{adv}, we input them into the target detector f. The output of the target detector typically includes object confidence f_{obj}, class confidence f_{cls}, and bounding box f_{bbox}. Since our goal is to create a stealthy attack, meaning that our adversarial texture T_{adv} should make the infrared car hide from the detector, we try to lower the object confidence $f_{\text{obj}}\left(I_{\text{adv}}\right)$ as much as possible. Therefore, the detection loss is defined as:

$$L_{\text{det}} = f_{\text{obj}}\left(I_{\text{adv}}\right). \tag{9}$$

The overall loss function is defined as follows:

$$\begin{aligned} L = L_{\text{det}} + w_1 \cdot L_{\text{norm}} + w_2 \cdot L_{\text{edge}} \\ + w_3 \cdot L_{\text{chamfer}} + w_4 \cdot L_{\text{Laplace}}. \end{aligned} \tag{10}$$

Here, w_1, w_2, w_3, and w_4 are weights of different losses, which are determined empirically. We use the backpropagation algorithm according to the loss function L to update the optimization variables $C^{(i)}, P^{(i)}, \varphi^{(i)}, i = 1, 2, \ldots, N$. The overall pipeline is illustrated in Figure 4.

3.7. Physical Implementation Method

We use aluminum films to make adversarial car stickers. Instead of altering the surface temperature of an object, this approach focuses on modifying the surface emissivity of the object, which is different from previous works [30–32, 35, 37]. Aluminum typically has an emissivity around 0.1, while the surface of a car, typically made of steel, has an emissivity around 0.8, resulting in different infrared characteristics. We utilize an ultra-thin (only 0.08mm) aluminum film, which can be easily attached on the surface of a car like many car stickers. We only need around 13 mins to make a sticker, and the cost of a sticker is only around 0.2 USD. The implementation process of adversarial stickers is shown in *Supplementary Video 1*.

4. Experiments

4.1. Dataset

We used the FLIR_ADAS_1_3 [7] infrared dataset released by FLIR company for infrared autonomous driving scenes. It contains 10,228 real infrared photos collected in streets and highways of Santa Barbara, USA. The infrared camera is FLIR Tau2. The training set contains 7160 images, and the test set contains 3068 images. We used this dataset to finetune the target detector.

4.2. Target Detectors

We initially chose the classic two-stage object detector: Faster R-CNN [22], as our primary target detector. We used the pre-trained Faster RCNN model provided by the torchvison library [17] and then finetuned it on the

Table 1. ASRs (%) of cars with different textures against different detectors.

Texture \ Detector	Faster	RetinaNet	Cascade	Libra	SSD	YOLOv3	Deformable
Origin	2.10	4.49	18.86	16.47	15.27	25.15	12.28
Random shape	18.26	17.07	23.95	28.74	45.81	50.00	23.65
AIP	14.97	15.27	32.93	27.54	23.65	38.32	27.54
UAP	20.06	31.44	38.02	33.23	36.83	39.82	24.25
Ours	**96.31**	**86.83**	**73.35**	**79.04**	**95.80**	**86.52**	**83.83**

FLIR_ADAS_1_3 dataset. The average precision (AP) for car class of the finetuned model on the training set was 0.96, and the AP on the test set was 0.92. After attacking Faster RCNN in a white-box setting, we transfered our attack method to other unseen detectors such as YOLOv3 [21], Deformable DETR [36], etc. which were provided by mmdetection library [5] in a black-box setting.

4.3. Evaluation Metrics

In our experiments, we used the attack success rate (ASR) as the evaluation metrics of the attack methods. The ASR was defined as the ratio of the number of cars which were not detected to the total number of cars. We set the confidence threshold of target detectors to 0.6 and the IOU threshold between the prediction box and ground truth to 0.5, similar to previous works [31, 32, 35, 37]. The ASR was calculated based on the average value of sample points collected from various distances, horizontal angles, and pitch angles with the sampling method described in Section 4.4.

4.4. Attack Faster RCNN in the Digital World

We optimized $N = 5$ adversarial shadow patterns to simulate 5 adversarial stickers to be pasted on the car surface.

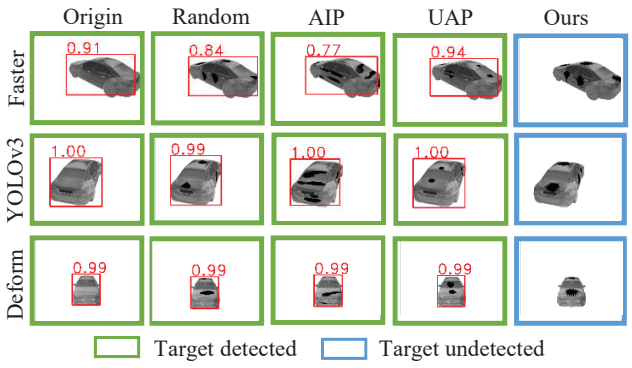


Figure 5. Examples of detection results of different detectors for target cars with different textures. The numbers above the red bounding boxes are the object confidence scores, with a threshold of 0.6. The results of other detectors are shown in *SM*.

The hyper-parameters are detailed in *SM*. After optimization, we obtained the adversarial shadow patterns (T_{adv} in Figure 4), and the rendered car with adversarial shadow patterns (I_{adv} in Figure 4).

After that, we evaluated the attack effectiveness of the adversarial shadow patterns. For a fair comparison, we employed the original car pattern (without any sticker) and random shape pattern as control patterns. The figures of these patterns are shown in *SM*. These patterns were rendered onto the same car model, and the resulting images were input into Faster R-CNN. The results, as shown in Table 1, indicate that our adversarial shadow patterns achieved an ASR of 96.31% for Faster R-CNN in the digital world. In comparison, the ASRs for the original car pattern and random shape pattern were only 2.10% and 18.26%, respectively. This demonstrates the effectiveness of our attack method. Figure 5 shows typical examples.

We then analyze the ASR of our method at various (*full-angle*) viewing angles and distances. The horizontal angle $azim$ ranged from 0 to 360 degrees, and we sampled it every 20 degrees. The pitch angle $elev$ ranged from 0 to 90 degrees, and we sampled it every 6 degrees. The distance $dist$ ranged from 1 to 8 meters, and we sampled it every 1 meter. Figure 6(b-d) show the ASRs with respect to one variable (e.g., $dist$) by taking the average of ASRs over all combinations of values of the other two variables (e.g., $elev$ and $azim$). The results indicate that our approach achieves successful attacks at various (*full-angle*) viewing angles and various distances. In contrast, many previous works [30–32, 35] were effective only within limited viewing angles (e.g., horizontal angle from -30 to 30 degrees and pitch angle 0 degree) and shorter distances (e.g. 3 to 6 meters). Note that there is a decrease in ASR at 2 meters and 0(or 180)-degree horizontal angle. This suggests that Faster RCNN is more robust in these specific scenes, potentially due to the distribution of training images. Nevertheless, for the majority of viewing angles and distances, the ASRs of our method consistently exceeded 80%.

4.5. Ablation Study

To evaluate the effectiveness of the 3D control points-based mesh smoothing algorithm (CMS) and a set of smoothing

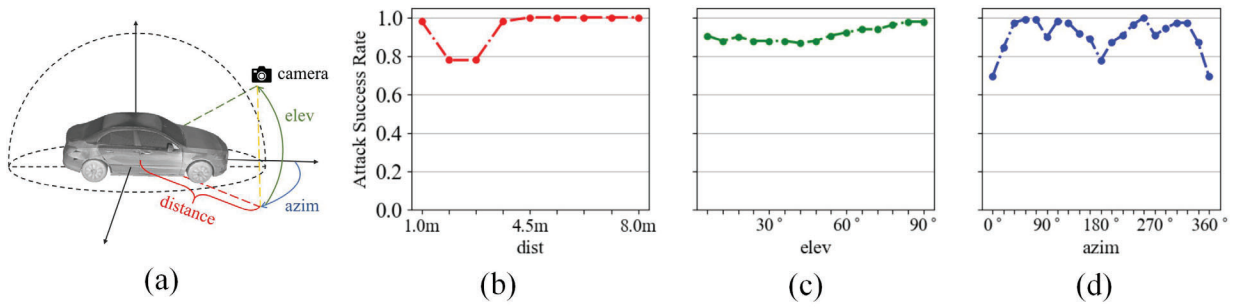


Figure 6. Full-angle ASRs at diffrent (a) parameters including (b) distances, (c) pitch angles, and (d) horizontal angles. See text for details.

losses (SMLS), we performed ablation experiments. We conducted a subjective evaluation on the smoothness score of the 3D adversarial mesh and 2D adversarial pattern and we also evaluated the physical implementation time of 2D adversarial patterns under different settings. The experimental settings and results are detailed in *SM*. The results indicate that both CMS and SMLS improved the smoothness of adversarial meshes and patterns, and their combination was better. Besides, these methods effectively reduced the physical implementation time of adversarial patterns.

4.6. Exploring the Interpretability of the Attack

To gain deeper insights into our attack methods, we utilized the GradCAM [23] technique to analyze the changes in network attention maps before and after the attack. See *SM* for more details.

4.7. Comparison with 2D Optimization Methods

We extended the previous 2D infrared model car attack methods [31, 32] to our 3D car model. We generated adversarial car textures on our car model based on their papers and codes, which are shown in *SM*. Following the settings in Section 4.4, we evaluated the attack performance of the different methods. The statistics of ASRs are presented in Table 1, with typical examples shown in Figure 5.

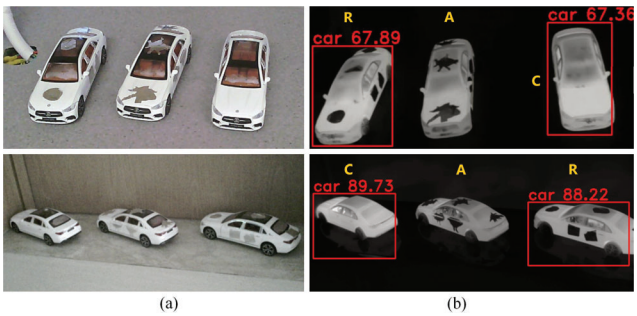


Figure 7. Infrared attack effect on model cars. (a) Visible light view of model cars. (b) Infrared view of model cars. C: clean car. R: car with random shape stickers. A: car with adversarial stickers.

The results indicated that the ASR of our method (96.31%) outperformed the ASRs of two alternative methods (14.97% and 20.06%) for Faster RCNN in the digital world. The reasons may be as follows. The two 2D optimization methods based on boundary optimization [31] or points clustering [32] are subject to various constraints. These constraints are used to ensure that, for example, the boundary curves do not have overlaps [31], and the adversarial patterns do not split into multiple pieces [32]. With more constraints, the feasible solution space becomes smaller. However, our 3D mesh shadow approach does not have such constraints, and we can explore a larger solution space, leading to better results.

4.8. Attack Transferability

We tested the effectiveness of our adversarial car texture (T_{adv} in Figure 4) optimized for Faster RCNN against other unseen detectors, including RetinaNet [14], Cascade RCNN [3], Libra RCNN [19], SSD [15], YOLOv3 [21] and Deformable DETR [36]. It is worth noting that these experiments were performed in a black-box setting, which is a more challenging but practically significant scene for real-world applications. The results are shown in Table 1. The ASRs of our method against unseen models reached 73.35%-95.80%. It indicates that our method performed well in a black-box setting and had good attack transferability to unseen models including not only CNN-based models but also transformer-based models. Besides, the transferability of our method is stronger than not only two simple baselines but also two infrared attack methods [31, 32].

4.9. Physical Attacks on Model Cars

We initially conducted physical experiments on three same 1:24 scale Mercedes-Benz model cars (Figure 7(a)). We crafted the adversarial aluminum stickers according to the optimized patterns, scaled to match the size of the model car, and applied them to the model car. In addition, we created randomly cut-shaped stickers as a control. We heated the model cars with hot water to simulate the real car with engine running. We used a rotating turntable to conve-

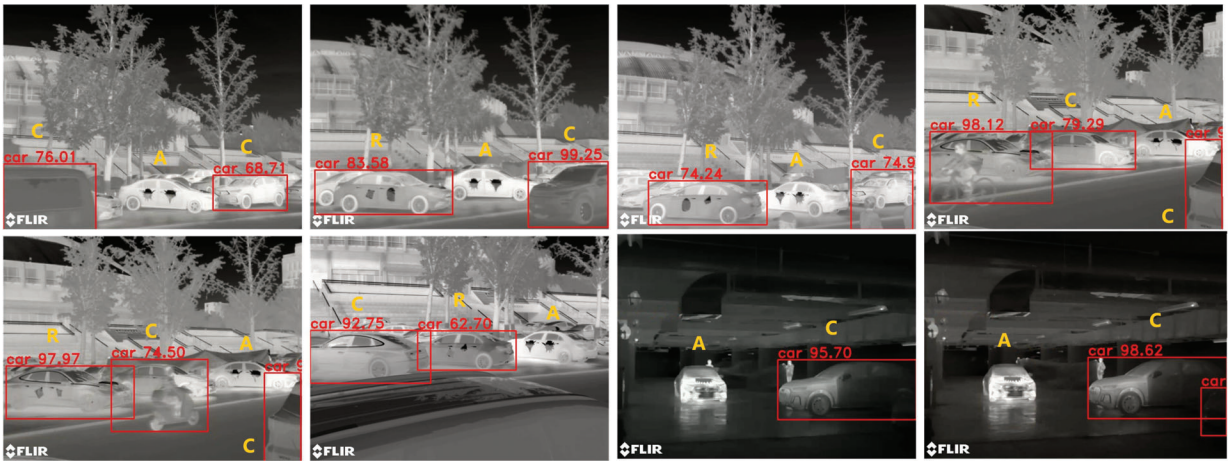


Figure 8. Examples of infrared real car attacks. C: clean car. R: car with random shape stickers. A: car with adversarial stickers.

niently assess the attack effectiveness over the entire 0-360 horizontal degree and 0-90 pitch angle range. The infrared camera was FLIR T560. We utilized the same Faster RCNN detector as in Section 4.4. The detection results indicated that our adversarial stickers achieved an ASR of 84.86% on the 1:24 scale car model in the physical world. In comparison, the clean car and the car with random patterns had ASRs of only 19.08% and 35.37%, respectively. Figure 7 provides specific examples from the physical world experiments. A demo video for physical model car attacks is shown in *Supplementary Video 2*.

4.10. Physical Attacks on Real Cars

We conducted physical experiments on two real Mercedes-Benz A200L cars (one black one white). It was a sunny day with a temperature of about 25°C. For a fair comparison, we pasted the the adversarial stickers, randomly cut-shaped stickers or nothing on the same car successively. We recorded 30 videos (each video is around 2 minutes) and sampled 3688 infrared images of these cars from various angles and distances in both ground and underground parking lots using a FLIR T560 camera. The height of the camera tripod can be adjusted from 1m to 2m. We sent these images to Faster RCNN.

The results indicated that our adversarial stickers achieved an ASR of 91.49% on the real cars with the engines running, while the random shape stickers and no sticker had ASRs of only 6.21% and 0.66%, respectively. When the engines were off for one hour, the ASR of adversarial stickers dropped a little to 88.42%. The reason might be that the infrared patterns of adversarial stickers with engines off were not as clear as the patterns with engines running. However, the ASR of adversarial stickers were still better than ASR of random shape stickers (5.72%) and no sticker (1.86%) when the engines were off. Figure 1 and Figure 8 show some examples. There were a few other cars that passed by or were parked when we were recording videos and therefore appeared in our photos, which were also detected. See *Supplementary Video 3* for the demo video.

4.11. Adversarial Defense

We tested five adversarial defense methods to defend our attack methods in the digital world, including adversarial training [8], PixelMask [1], Bit squeezing [33], JPEG compression [10] and Total variation minimization [10]. Experiment details for these methods are in *SM*. The results show that although these methods had a certain defense effect, the ASR of our method still reached 88.83%-94.81% after adding defense, which shows that our method is a powerful attack method.

5. Conclusion

We propose infrared adversarial stickers to hide a real car from infrared detectors at various viewing angles, distances, and scenes in the physical world. We build a 3D infrared car model with real infrared characteristics and propose a 3D mesh shadow method for the generation of infrared adversarial pattern. To make the 3D adversarial mesh smoother, we propose a 3D control points-based smoothing algorithm and use a set of smoothness loss functions. Our adversarial stickers enabled two real cars to evade Faster RCNN at various viewing angles, distances and scenes. Besides, our method has strong attack transferability against multiple unseen detectors in a black-box setting.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Nos. 61734004, U2341228, U19B2034).

References

[1] Akshay Agarwal, Mayank Vatsa, Richa Singh, and Nalini Ratha. Cognitive data augmentation for adversarial defense via pixel masking. *Pattern Recognition Letters*, 146:244–251, 2021. 8

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018. 5

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019. 7

[4] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017. 2

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 4

[7] FLIR. Free flir thermal dataset for algorithm training. [EB/OL]. https://www.flir.com/oem/adas/adas-dataset-form/ Accessed Nov. 12, 2021. 5

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Int. Conf. Learn. Represent.*, 2015. 2, 8

[9] Jindong Gu, Hengshuang Zhao, Volker Tresp, and Philip HS Torr. Segpgd: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness. In *European Conference on Computer Vision*, pages 308–325. Springer, 2022. 2

[10] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *Int. Conf. Learn. Represent.*, 2018. 8

[11] Zhanhao Hu, Siyuan Huang, Xiaopei Zhu, Xiaolin Hu, Fuchun Sun, and Bo Zhang. Adversarial texture for fooling person detectors in the physical world. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[12] Zhanhao Hu, Wenda Chu, Xiaopei Zhu, Hui Zhang, Bo Zhang, and Xiaolin Hu. Physically realizable natural-looking clothing textures evade person detectors via 3d modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16975–16984, 2023. 2

[13] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L. Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[14] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Int. Conf. Comput. Vis.*, 2017. 7

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Eur. Conf. Comput. Vis.*, 2016. 7

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Int. Conf. Learn. Represent.*, 2018. 2

[17] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016. 5

[18] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389, 2006. 5

[19] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 7

[20] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[21] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 6, 7

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6), 2016. 5

[23] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 7

[24] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 23(5):828–841, 2019. 2

[25] Naufal Suryanto, Yongsu Kim, Hyoeun Kang, Harashta Tatimma Larasati, Youngyeo Yun, Thi-Thu-Huong Le, Hunmin Yang, Se-Yoon Oh, and Howon Kim. Dta: Physical camouflage attacks using differentiable transformation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15305–15314, 2022. 2

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Int. Conf. Learn. Represent.*, 2014. 2

[27] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, 2019. 2

[28] Donghua Wang, Tingsong Jiang, Jialiang Sun, Weien Zhou, Zhiqiang Gong, Xiaoya Zhang, Wen Yao, and Xiaoqian Chen. Fca: Learning a 3d full-coverage vehicle camouflage for multi-view physical adversarial attack. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2414–2422, 2022. 2

[29] Jiakai Wang, Aishan Liu, Zixin Yin, Shunchang Liu, Shiyu Tang, and Xianglong Liu. Dual attention suppression attack: Generate adversarial camouflage in physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8565–8574, 2021. 2

[30] Hui Wei, Zhixiang Wang, Xuemei Jia, Yinqiang Zheng, Hao Tang, Shin'ichi Satoh, and Zheng Wang. Hotcold block: Fooling thermal infrared detectors with a novel wearable design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15233–15241, 2023. 2, 5, 6

[31] Xingxing Wei, Yao Huang, Yitong Sun, and Jie Yu. Unified adversarial patch for cross-modal attacks in the physical world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4445–4454, 2023. 2, 6, 7

[32] Xingxing Wei, Jie Yu, and Yao Huang. Physically adversarial infrared patches with learnable shapes and locations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12334–12342, 2023. 2, 5, 6, 7

[33] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS*, 2018. 8

[34] Yang Zhang, PD Hassan Foroosh, and Boqing Gong. Camou: Learning a vehicle camouflage for physical adversarial attack on object detections in the wild. *ICLR*, 2019. 2

[35] Xiaopei Zhu, Xiao Li, Jianmin Li, Zheyao Wang, and Xiaolin Hu. Fooling thermal infrared pedestrian detectors in real world using small bulbs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3616–3624, 2021. 2, 5, 6

[36] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *Int. Conf. Learn. Represent.*, 2021. 6, 7

[37] Xiaopei Zhu, Zhanhao Hu, Siyuan Huang, Jianmin Li, and Xiaolin Hu. Infrared invisible clothing: Hiding from infrared detectors at multiple angles in real world. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2, 5, 6