Poltergeist: Acoustic Adversarial Machine Learning against Cameras and Computer Vision

Xiaoyu Ji[1], Yushi Cheng[1], Yuepeng Zhang[1], Kai Wang[1], Chen Yan[1], Wenyuan Xu[1†], Kevin Fu[2]
[1]Ubiquitous System Security Lab (USSLAB), Zhejiang University
[2]Security and Privacy Research Group (SPQR), University of Michigan
{xji, yushicheng, ypzhang, eekaiwang, yanchen, wyxu}@zju.edu.cn, kevinfu@umich.edu

Abstract

Autonomous vehicles increasingly exploit computer-vision-based object detection systems to perceive environments and make critical driving decisions. To increase the quality of images, image stabilizers with inertial sensors are added to alleviate image blurring caused by camera jitters. However, such a trend opens a new attack surface. This paper identifies a system-level vulnerability resulting from the combination of the emerging image stabilizer hardware susceptible to acoustic manipulation and the object detection algorithms subject to adversarial examples. By emitting deliberately designed acoustic signals, an adversary can control the output of an inertial sensor, which triggers unnecessary motion compensation and results in a blurred image, even if the camera is stable. The blurred images can then induce object misclassification affecting safety-critical decision making. We model the feasibility of such acoustic manipulation and design an attack framework that can accomplish three types of attacks, i.e., hiding, creating, and altering objects. Evaluation results demonstrate the effectiveness of our attacks against four academic object detectors (YOLO V3/V4/V5 and Fast R-CNN), and one commercial detector (Apollo). We further introduce the concept of `AMpLe` attacks, a new class of system-level security vulnerabilities resulting from a combination of adversarial machine learning and physics-based injection of information-carrying signals into hardware.

I. Introduction

Autonomous vehicles depend on computer-vision-based object detection algorithms to automatically classify objects when cameras capture road images. Correct classification in the midst of a dedicated adversary is important to ensure safe driving decisions. If an adversary can hide, create, or alter the classification results of objects within an image, e.g., failing to detect a pedestrian, the autonomous vehicle could be fooled into making a tragic decision. Since the quality of captured images is critical for robust object detection, modern cameras not only consist of an image sensor, e.g., a CMOS (complementary metal-oxide semiconductor) or CCD (charge-coupled device) sensor, but also an image stabilizer designed to de-blur the images by compensating the jitters of the cameras. The added feature, i.e., image stabilization, improves the image quality in benign scenarios, yet could be exploited by dedicated adversaries. In this paper, we identify a new class of system-level vulnerabilities resulting from the combination of the emergent image stabilization hardware susceptible to acoustic manipulation and the object detection algorithms subject to adversarial examples, and design `Poltergeist` attacks (in short, `PG` attacks) that exploit such vulnerabilities.

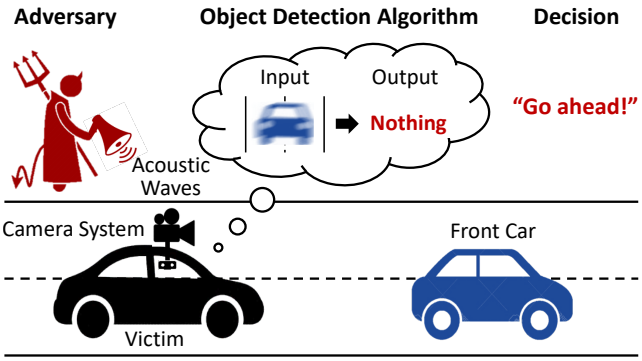


Figure 1: By injecting acoustic signals into the inertial sensors of object-detection systems in autonomous vehicles, an adversary can fool decision making.

Unlike existing work that focused on altering what the *main* sensors (e.g., CMOS sensors) perceive by changing the visual appearance of an object [12], [38], [59], [27] or by projecting lights into the camera [22], our work calls attention to *auxiliary* sensors that are used to assist the *main* ones, e.g., inertial sensors provide motion feedback to the stabilizer for image blur reduction. In light of prior work illustrating that acoustic signals can control the output of accelerometers [43], [48] and gyroscopes [37], [45], we investigate the feasibility of acoustically manipulating the image stabilization process so as to cause misclassification of objects, despite the fact that the camera is stationary. The insight into such threats is essential to expand our ability to secure future devices, as an increasing number of sensors may be added to the feedback control loop to increase the intelligence level of such autonomous systems.

Essentially, `PG` attacks are initiated by controlling the inertial sensors of the stabilizer via resonant acoustic signals, which creates blurred images because of unnecessary stabilization, and finally results in misclassification. To model the validity of acoustic attacks worming their way into classification algorithms of object detection systems, two research questions

†Corresponding author
*Source code and demo: https://github.com/USSLab/PoltergeistAttack

remain unanswered. The first is to quantify the impact of acoustic attacks on the level and patterns of the image blur, regardless of the type of cameras. Without loss of generality, we choose the inertial sensor readings (e.g., acceleration) to quantify the camera motions caused by acoustic manipulation and build a motion blur model to describe the relationship between the sensor readings and the resulted blur patterns. The second is to find an effective blurred image that will lead to successful misclassification. Much work on creating effective adversarial samples requires the machine learning algorithms to be white-box, yet we consider the object detection algorithms as black-box to mimic a real-world attack. To find an effective blurred image that can lead to misclassification, we construct a gradient-free optimization method that can lead to the following three types of attacks:

- **Hiding attacks (HA)** cause an object to become undetected, e.g., make a front car "disappear" (Fig. 1).
- **Creating attacks (CA)** induce a non-existent object, e.g., create a car or a person in the driveway.
- **Altering attacks (AA)** cause an object to be misclassified, e.g., render a person detected as a fire hydrant.

To validate `PG` attacks, we examine the effectiveness of acoustic manipulation on a standard library of roadway images to predict the behavior of existing and future autonomous vehicles, and test the `PG` attacks using standard academic object detectors as well as a standard commercial object detector in autonomous vehicles. In summary, our contributions include the points below:

- To the best of our knowledge, this is the first work to exploit the cameras' auxiliary sensor vulnerabilities via acoustic manipulation to create misclassification in object detection systems.
- We model the limits of `PG` attacks and construct gradient-free algorithms to create adversarial blurry images that can lead to three undesired consequences: hiding objects, creating objects, and altering objects.
- We validate the effectiveness of `PG` attacks with four academic object detectors (YOLO V3/V4/V5 and Fast R-CNN) and one commercial detector (Apollo).

`Poltergeist` attacks serve as the first instance of a broad range of emerging vulnerabilities we call `AMpLe` attacks (injecting **p**hysics into **A**dversarial **M**achine **Le**arning). `AMpLe` attacks combine weaknesses (1) in the physics of hardware [13] (2) and in adversarial machine learning (3) to cause a system-level exploit. With the proliferation of sensors in intelligent cyberphysical systems, we envision that in addition to acoustic signals, future `AMpLe` attacks could leverage signal transmission via ultrasound [55], [49], [51], visible light, infrared, lasers [41], radio [20], magnetic fields, heat, fluid, etc. to manipulate sensor outputs and thus the subsequent machine learning processes (e.g., voice recognition, computer vision). Emerging cyberphysical systems depend on trustworthy data from sensors to make automated decisions. `AMpLe` attacks could cause incorrect, automated decisions with life-critical consequences for closed loop feedback systems (e.g., medical devices [31], autonomous vehicles, factory floors, IoT).

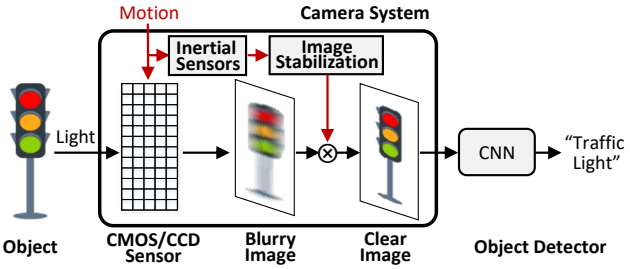


Figure 2: Object-detection systems typically utilize image stabilization with inertial sensors to reduce the blur effect caused by camera motions, and to improve the accuracy of object detection.

II. Background

In this section, we first introduce the object-detection system and the image stabilization system, and then summarize the sensor vulnerabilities that can be used for attacks.

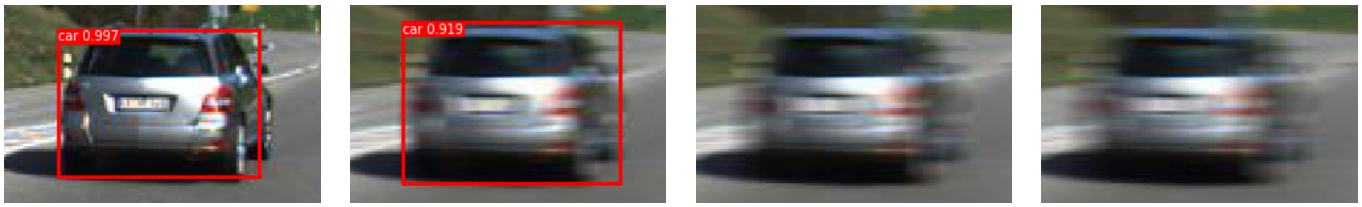
A. Object Detection

Autonomous vehicles rely on computer vision algorithms to detect objects in the environment. As shown in Fig. 2, the object-detection systems work as the following. First, image sensors such as CMOS or CCD sensors convert the lights reflected from physical objects to electrical signals, which are processed and digitized to create digital images. Then, object detectors utilize machine learning algorithms to classify the objects in the images, which will be used for decision making. State-of-the-art object detectors utilize convolutional neural networks (CNNs) for object detection. Two of the most widely used ones are YOLO V3 [33] and Faster R-CNN [15], which are two-stage and one-stage object detectors, respectively.

B. Image Stabilizer

In practice, photos captured by the image sensors can be blurred due to the motions that occur within the exposure duration. The larger the motion, the heavier the blur. To obtain a clear photo, modern camera systems exploit an image stabilizer to reduce the unwanted blur effects. Popular image stabilization techniques include (1) optical image stabilization (OIS) that shifts the camera lens or image sensors physically such that the images can be projected onto the "stationary" imaging plane [6], (2) mechanical image stabilization (MIS) that compensates for camera motions by actuating an external camera stabilizer in an opposite way [4], and (3) digital/electronic image stabilization (DIS or EIS) that eliminates blur patterns through software-based image processing algorithms [47], [26], [11].

Among the aforementioned image stabilization techniques, precise motion estimation is essential. To achieve it, micro-electro-mechanical systems (MEMS) inertial sensors, i.e., accelerometers and gyroscopes, are widely integrated into image stabilizers, and the sensor measurements are fed back to compensate for the blurry images, as shown in Fig. 2. Image stabilizers with inertial sensors are commonly used in various systems, including autonomous vehicles [18], smartphones, sports cameras, etc.



(a) Car detected without any motion blur (confidence score 0.997)

(b) Car detected (0.919) after linear motion blur (slight, horizontal)

(c) Nothing detected after linear motion blur (medium, horizontal)

(d) Nothing detected after linear motion blur (heavy, horizontal)

Figure 3: A car cannot be correctly detected under increasing linear motion blur.

(a) Nothing detected for the original image without any motion blur

(b) Person detected (0.902) after linear motion blur (slight, horizontal)

(c) Boat detected (0.894) after linear motion blur (heavy, inclined)

(d) Car detected (0.851) after linear motion blur (heavy, horizontal)

Figure 4: The region with no interested objects can be incorrectly detected as a person (b), a boat (c), and a car (d) under different linear motion blur.

(a) Car detected without any motion blur (confidence score 0.979)

(b) Car is misclassified as bus (0.99) after linear motion blur (slight, vertical)

(c) Car is misclassified as bottle (0.439) after rotational motion blur (slight, anticlockwise)

(d) Car is misclassified as person (0.969) after rotational motion blur (heavy, anticlockwise)

Figure 5: A car can be incorrectly detected as a bus (b), a bottle (c), and a person (d) under different motion blur.

C. Inertial Sensor Vulnerability to Acoustic Signals

MEMS inertial sensors, e.g., accelerometers and gyroscopes, are known to be vulnerable to resonant acoustic injection attacks [37], [43], [48], [45]. Both the MEMS accelerometers and gyroscopes rely on sensing masses to measure the inertial stimuli. In particular, the sensing masses move as they are exposed to stimuli, and their displacements are mapped to measurable capacitance changes. In addition to regular motion stimuli, the sensing mass can be influenced by acoustic signals at the frequencies close to the natural frequency of the mechanical structure of the mass, i.e., the sensing mass is forced into resonance at the same frequency as the sound pressure waves. As a result, the sensor can output a controllable value according to the injected resonant acoustic signal even if the sensor is stationary. This vulnerability allows an attacker to manipulate the sensor outputs by injecting carefully crafted acoustic signals. Much work [43], [45] has demonstrated the feasibility of fine-grained control over a MEMS inertial sensor's output. For instance, Trippel *et al.* [43] proposed the output biasing attack, which provides fine-grained accelerometer output control, and Tu *et al.* [45] showed similar attacks against gyroscopes. In this paper, we utilize similar attacks to launch `PG` attacks.

D. Remark

In summary, object-detection systems rely on inertial sensors to create photos free from motion blur. While the inertial sensors provide motion feedback for the image stabilizer, they also inevitably introduce acoustic injection vulnerabilities, whereby an attacker can manipulate the sensor outputs and the motion compensation process. Thus, photos produced after unnecessary compensation may cause the object detection algorithm to misclassify the objects.

III. Preliminary Analysis

The key of `Poltergeist` attacks involves two blocks that (1) control the outputs of inertial sensors via acoustic signals, and (2) create a blurry image that can lead to misclassification, subject to the motion compensation constraints. Since the acoustic manipulation is validated via prior work, in this section we conduct a preliminary analysis to investigate the feasibility of fooling object detectors by emulating the motion compensation guided by the output of inertial sensors. Since linear acceleration and rotation are the most dominant motions measured by inertial sensors, we generate blurry images with two common blur filters in Photoshop [1], i.e., the linear and rotational motion blur filters, and test the misclassification results on a representative academic object detector, i.e., YOLO V3. The images for motion compensation are selected from an autonomous driving dataset BDD100K [53]. We adjust the blur parameters randomly yet in three categories, e.g., slight, media, heavy linear acceleration or rotational motion, and we show a few representative results in Fig. 3-4, whereby a detected object is marked as a rectangle with the confidence

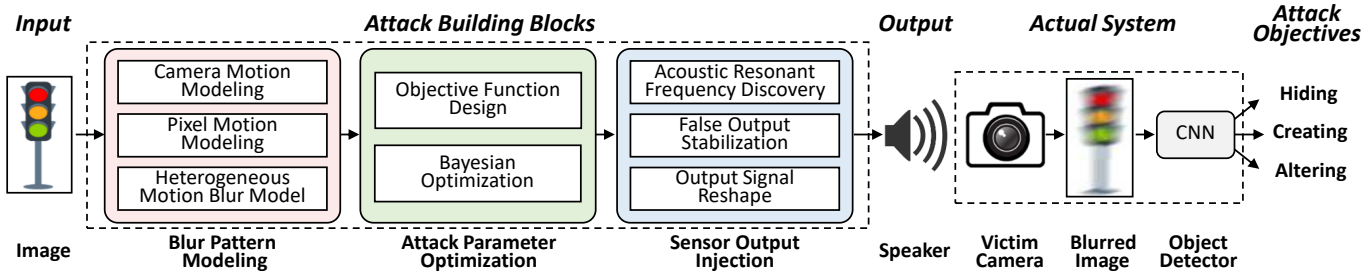


Figure 6: PG attacks: The adversary first uses an image of the target object to generate feasible attack parameters with blur pattern modeling and attack parameter optimization. Then, the adversary manipulates the sensor outputs according to the calculated parameters via acoustic signals to deceive the object detector, which may lead to hide, create, or alter objects.

scores and the type of predicted classes. The preliminary results illustrate the following key insights.

> ***Observation:*** *The blur caused by unnecessary motion compensation can change the outline, the size, and even the color of an existing object or an image region without any objects, which may lead to hiding, altering an existing object, or creating a non-existing object.*

Hiding objects. As shown in Fig. 3, a *car* that was detected with a high confidence score in the original image becomes less likely to be classified correctly after being blurred with a linear motion. As the degree of the blur increases, the detection confidence drops until the object is unable to be detected at all. Our hypothesis is that the unnecessary motion compensation blurs the object outline and its color, which are critical in extracting features for object detection.

Creating objects. As shown in Fig. 4, the original image region where nothing interesting has been detected can be detected with a *person*, a *boat*, or a *car* under various linear motion blurs, even though the blurred images are meaningless to a human.

Altering objects. As illustrated in Fig. 5, the *car* that is correctly classified with a high confidence score in the original image is misclassified as other classes and even counterintuitive ones, e.g., a *bus*, a *bottle*, and a *person* under the linear and rotational motion compensation. We observe that for the latter two cases, the non-linear rotational motion compensation has played a dominant role, since such motion blurring may have changed the shape of the red car into a person's head.

The aforementioned three types of misclassification cases demonstrate the feasibility of deceiving object detectors using the blurring effect and encourage us to further investigate `Poltergeist` attacks via unnecessary motion compensation. The blurring effect is essentially the addition of pixels at multiple locations determined by the motion displacement, e.g., the color of the pixel in a blurred image is the sum of the one at its origin and the one at distance. This motivates us to model the blurring effect from the pixel perspective, and quantify its impact on the object detection. In Sec. V, we quantitatively model the blurring processes in terms of creating images that lead to misclassification, and we refer to such images as *adversarial blurry images* hereafter.

IV. Threat Model

In this paper, we consider three types of attacks:

- **Hiding attacks (HA)**, where the goal of the adversary is to let the object detector fail to identify an object that is of interest to the systems.
- **Creating attacks (CA)**, where the adversary blurs the images such that the object detector detects a non-existing object as if it were physically present.
- **Altering attacks (AA)**, where the adversary blurs the image such that an object is misclassified to another object.

In addition, we assume the adversary has the following capabilities to achieve the aforementioned attacks:

Black-box Object Detector. We assume that the adversary has no prior knowledge of the object detection algorithm, including but not limited to its architecture, parameters, etc. However, the adversary can obtain the classification results and their confidence scores for each detected object, which are default outputs for various object detectors, e.g., YOLO V3, Faster R-CNN.

Camera and Sensor Awareness. The adversary can acquire and analyze a camera of the same model as the one used in the targeted system, from which she can learn the information of the camera parameters, e.g., the camera focal length, the camera exposure time, and the physical locations as well as parameters of the inertial sensors.

Acoustic Attack Capability. We assume the adversary can launch acoustic injection attacks towards the inertial sensors in the target system. She may transmit acoustic signals by (1) setting up a speaker or an ultrasonic transducer array along the roadside, (2) attaching speakers to the surface of a target system, or (3) controlling a compromised on-board speaker in the target system, e.g., a speaker in the car.

V. Attack Design

A. Overview

To generate adversarial blurry images via acoustic signals, it is important to address the following two challenges:

- **Challenge 1:** How to quantify the impact of acoustic manipulation upon the patterns and levels of the image blur?

- **Challenge 2:** How to optimize the blur patterns for an effective and efficient attack against black-box object detectors?

To have a scene misclassified, i.e., achieving HA, CA, and AA attacks, PG incorporates three key modules as shown in Fig. 6. The **Blur Pattern Modeling** module models the relationship between the sensor readings, i.e., the outputs of the accelerometer and gyroscope, and the intended blur patterns. The **Attack Parameter Optimization** module formulates the attack as an optimization problem to find the feasible attacks under different blur patterns, and derives the feasible solutions in terms of sensor readings, i.e., accelerations and angular velocities. The **Sensor Output Injection** module generates an elaborate attack signal according to the expected sensor readings, and transmits it to the sensor module of a camera system to launch PG attacks. In the following sections, we present our attack building blocks in detail.

B. Blur Pattern Modeling

To generate adversarial blur patterns via acoustic signals, we first model the relationship between the manipulation of sensor outputs and the resulted blur patterns.

1) From Sensor Outputs to Compensatory Camera Motions: To quantify the impact of acoustic manipulation, we first model the false camera motion (FCM) caused by false sensor outputs. Without loss of generality, the FCM has up to six DOFs (degree-of-freedom), i.e., x-axis, y-axis, z-axis caused by the accelerometer, and roll, pitch, yaw caused by the gyroscope. For simplicity, we assume the x-y plane is the imaging projection plane and FCMs in all the six DOFs share the same original point, i.e., the physical center of the camera. In this way, the FCMs can be denoted as $\vec{M}_f = \{\vec{a}_x, \vec{a}_y, \vec{a}_z, \vec{\omega}_r, \vec{\omega}_p, \vec{\omega}_y\}$, where $\vec{a}_x$, $\vec{a}_y$, and $\vec{a}_z$ are the false sensor outputs of the x-axis, y-axis, and z-axis of the accelerometer respectively, and $\vec{\omega}_r$, $\vec{\omega}_p$, and $\vec{\omega}_y$ are the false sensor outputs of the roll, pitch, and yaw DOFs of the gyroscope, respectively. As the compensatory camera motion (CCM) is equivalent but inverse to the FCM, we then express the compensatory camera motion under acoustic injection attack as: $\vec{M}_c = \{-\vec{a}_x, -\vec{a}_y, -\vec{a}_z, -\vec{\omega}_r, -\vec{\omega}_p, -\vec{\omega}_y\}$.

In this paper, we consider creation of adversarial blur patterns via CCMs from two sets of DOFs: (1) in the imaging projection plane, which correspond to motions in the x-axis, y-axis, and roll DOFs, and (2) towards (backwards) the imaging projection plane, which refer to motions in the z-axis DOF. The other two DOFs are not exploited since they require additional pixel information out of the captured scene, which may not always be available in real-world attacks.

2) From Compensatory Camera Motions to Pixel Motions: The compensation gives rise to pixel motions in the finally formed image. In this subsection, we derive the pixel motions under CCMs along different DOFs, and the complete deduction can be referred to in Appendix. A.

Pixel Motions caused by x-axis and y-axis CCMs. The x-axis and y-axis CCMs introduce horizontal or vertical camera displacements respectively, which will then change the optical path and result in pixel motions in the same DOF but in an opposite direction, since the formed image is reverse to the scene in the camera. Thus, a linear compensation in the x/y-axis DOF causes a linear pixel motion in the formed image expressed as $\frac{f}{2u}\vec{a}_x T^2$ or $\frac{f}{2u}\vec{a}_y T^2$, where $-\vec{a}_x(-\vec{a}_y)$ are the acceleration along the x and y axes, f is the camera focal length, u is the object distance, and T is the camera exposure time. We refer to the pixel motions along the x-axis and y-axis DOFs as *linear motion.*

Table I: Summary of 3 Types of Motion Blur.

No.	Sensor Output	Blur Type	Blur Parameter
1	$\{\vec{a}_x, \vec{a}_y\}$	Linear	$\vec{L}_{xy} = \frac{f}{2u}(\vec{a}_x + \vec{a}_y)T^2$ $\alpha = \arccos(\frac{\vec{a}_x \cdot \vec{a}_y}{\lvert\vec{a}_x\rvert\lvert\vec{a}_y\rvert})$
2	$\vec{a}_z$	Radial	$p = \frac{\vec{a}_z T^2}{2u}$
3	$\vec{\omega}_r$	Rotational	$\beta = \omega_r T$

Pixel Motions caused by the z-axis CCMs. The z-axis CCM, however, gives camera displacements towards or away from the scene, and changes imaging distances and therefore pixel motions backwards or towards the image center [58]. For each pixel in the output image, a CCM in the z-axis DOF with an acceleration of $-\vec{a}_z$ in fact gives rise to a pixel motion of $\frac{\vec{a}_z T^2}{2u} r_o$ towards the image center, where r_o refers to the pixel distance to the image center. We refer to the pixel motions towards or away from the image center as *radial motion.*

Pixel Motions caused by the roll CCMs. In contrast to CCMs in the x-axis, y-axis, and z-axis DOFs that change the optical path, CCMs in the roll DOF only rotate the image sensor. As a result, a CCM in the roll DOF with an angular velocity of $-\vec{\omega}_r$ gives a pixel angular velocity of $\vec{\omega}_r$. For each pixel in the output image, the pixel motion is then determined by its distance to the rotation center r_c, the angular velocity $\vec{\omega}_r$, and the exposure time T as $\omega_r T r_c$. We call this type of pixel motion *rotational motion.*

As the output image represents an integration of all pixel motions during the camera exposure time, each type of motions introduces a specific blur pattern into the output images.

3) From Pixel Motion to Blur Patterns: Based on the three kinds of pixel motions along different DOFs, we can categorize the blur patterns into three types shown in Tab. I.

(1) Linear Motion Blur is the blur pattern caused by linear pixel motions, as shown in Fig. 13(a). It is specified by the camera focal length f, the camera exposure time T, the scene distance u, and the x-axis and y-axis accelerations $\vec{a}_x$ and $\vec{a}_y$. Linear motion blur is formulated as $\{\vec{L}_{xy}, \alpha\}$, where $\vec{L}_{xy} = \frac{f}{2u}(\vec{a}_x + \vec{a}_y)T^2$ and $\alpha = \arccos(\frac{\vec{a}_x \cdot \vec{a}_y}{\lvert\vec{a}_x\rvert\lvert\vec{a}_y\rvert})$, $\vec{L}_{xy}$ is the identical pixel displacement of each pixel in the image and α is the angle between $\vec{L}_{xy}$ and the y-axis. A representative example is shown in Fig. 13(a) in Appendix. B.

(2) Radial Motion Blur is the blur pattern caused by radial pixel motions along rays towards or away from the image center, as shown in Fig. 13(b) in Appendix. B. It is specified by the image center I, the exposure time T, the scene distance u, and the z-axis acceleration $\vec{a}_z$. We denote radial motion blur as $p = \frac{\vec{a}_z T^2}{2u}$.

(3) Rotational Motion Blur is the blur pattern caused by rotational pixel motions along an arc, as shown in Fig. 13(c) in Appendix. B. It is specified by the rotation center C, the exposure time T, and the angular velocity $\vec{\omega}_r$ in the roll DOF. We use a rotation angle to express the rotational motion blur as $\beta = \omega_r T$.

4) Heterogeneous Motion Blur Model: Considering that pixel motions can simultaneously occur along multiple DOFs, resulting in heterogeneous blur patterns as shown in Fig. 13(d) in Appendix. B, we build a heterogeneous motion blur model for the final output image to describe the blur patterns caused by compensatory camera motions in the four aforementioned DOFs.

Denote B as the final blurred image and X as the originally unblurred one. For each pixel $B(i,j)$ located in row i and column j in the blurred image B, we have:

$$B(i,j) = \frac{1}{T}\int_{-T}^{0} X(i+u(t), j+v(t))dt \tag{1}$$

where $u(t)$ and $v(t)$ are the pixel motion functions in the x-axis and y-axis, respectively, $I = (o_0, o_1)$ is the center of image X and $C = (c_0, c_1)$ is the rotational center.

By discretization, $B(i,j)$ can be estimated by:

$$\begin{aligned} B(i,j) &= \frac{1}{n+1}\sum_{k=-n}^{0} X(i+u(k\cdot\frac{T}{n}), j+v(k\cdot\frac{T}{n})) \\ &= \frac{1}{n+1}\sum_{k=-n}^{0} X(i'(k), j'(k)) \end{aligned} \tag{2}$$

From Equ. (1) and Equ. (2), we can observe that the blur patterns depend on the camera exposure time and the pixel motions in the image plane. The key to model the heterogeneous motion blur is to resolve the pixel location $(i'(k), j'(k))$ at each discrete time k during the exposure time, which can be retrospectively calculated with the final pixel location (i,j) and the pixel motion functions in the x-axis and y-axis as:

$$\begin{aligned} [i'(k), j'(k)]^T &= H(i,j,u(k),v(k)) \\ &= [u(k), v(k)]^T + [i,j]^T \end{aligned} \tag{3}$$

where $u(k)$ and $v(k)$ are the pixel motion displacements in the x-axis and y-axis at the time k, respectively. Based on the three types of motion blur patterns analyzed above, we can further derive $u(k)$ and $v(k)$ as:

$$\begin{aligned} [u(k), v(k)]^T &= \begin{bmatrix} \cos\alpha & \cos(\frac{k}{n}\beta+\gamma) & \cos\delta \\ \sin\alpha & \sin(\frac{k}{n}\beta+\gamma) & \sin\delta \end{bmatrix} \begin{bmatrix} \frac{kf|\vec{a}_x+\vec{a}_y|T^2}{2nu r_c} \\ \frac{k|\vec{a}_z|T^2 r_o}{2nu} \end{bmatrix} \\ \gamma &= \arctan(\frac{j-c_1}{i-c_0}), \quad r_c = \|(i,j),(c_0,c_1)\|_2 \\ \delta &= \arctan(\frac{j-o_1}{i-o_0}), \quad r_o = \|(i,j),(o_0,o_1)\|_2 \end{aligned} \tag{4}$$

where γ is the angle between the y-axis and the radius r_c specified by (c_0, c_1) and (i,j), δ is the angle between the y-axis and the radius specified by (o_0, o_1) and (i,j). Together with Equ. (2), Equ. (3), and Equ. (4), which form the heterogeneous motion blur model, we can obtain $B(i,j)$ and thus the entire blurred image B.

C. Gradient-Free Attack Parameter Optimization

With the heterogeneous motion blur model, we are able to simulate the blur effects via changing the four parameters, i.e., $\vec{a}_x, \vec{a}_y, \vec{a}_z, \vec{\omega}_r$ of accelerometer and gyroscope. To improve the attack effectiveness, we design objective functions specific to hiding, creation, and alteration attacks (HA, CA, and AA).

As mentioned in Sec. II, to implement practical adversarial attacks, we consider the object detector to be black-box since we cannot always obtain the network frameworks and parameters in real-world attacks. For a black-box object detector, given an input X, it makes several most possible predictions $Y = f(X)$. Each prediction $Y_i \in Y$ can be represented as:

$$Y_i = (B_i, S_i^B, C_i, S_i^C) \tag{5}$$

where B_i and C_i are the bounding box and class of the prediction, and S_i^B and S_i^C are the corresponding confidence scores. An adversarial example in our case refers to a blurred image B determined by attack parameters $\{\vec{a}_x, \vec{a}_y, \vec{a}_z, \vec{\omega}_r\}$, i.e., $B = X' = X + \Delta$. To implement an effective attack, we try to resolve optimized adversarial examples (attack parameters) with objective functions specifically designed for each attack.

1) Objective Functions: To optimize HA, CA, and AA, we consider the following three factors: (1) the product of the bounding box confidence score S_i^B and the class confidence score S_i^C that determines whether the object can be detected, i.e., the product should be larger than a threshold for the object to be detected, (2) the intersection over union U_{ij} that indicates the overlapping degree of the bounding box B_i and the bounding box B_j, and (3) the magnitude of Δ that represents the degree of blur effects. The former two factors determine the attack success rate while the last one presents the attack cost. We take all of them into consideration to strike the balance of attack success rate and cost.

For HA, to hide the prediction of a particular object of interest Y_i, the product of S_i^B and S_i^C should be less than the threshold that determines whether the object can be detected, and the magnitude of Δ that represents the attack cost should be minimized to ease the burden of acoustic injection attacks. Therefore, the objective function for HA can be given as:

$$\boxed{\begin{aligned} \textbf{HA:} \quad & \underset{\vec{a}_x,\vec{a}_y,\vec{a}_z,\vec{\omega}_r}{\operatorname{argmin}} && w_1 S_i^B S_i^C \ + \ w_2||\Delta||_p \\ & \text{s.t.} && |\vec{a}_x + \vec{a}_y + \vec{a}_z| < \xi_1 \\ & && |\vec{\omega}_r| < \xi_2 \end{aligned}} \tag{6}$$

where w_1 and w_2 are the weights that balance the attack success rate and cost, and ξ_1 and ξ_2 are the physical attack capability restrictions of accelerometers and gyroscopes.

For CA, to create a prediction of an object of interest that is not present physically in the image (denoted as Y_o), the class of the created prediction C_o shall be the targeted class T, the product of S_o^B and S_o^C should be larger than the threshold, the sum of intersections over union of the created box B_o with any $B_i \in Y$ shall be minimized to ensure that it is created rather than transformed from existing objects, and the magnitude of Δ should be minimized to reduce the attack cost. Therefore,

the objective function for targeted CA can be given as:

$$\textbf{CA:} \quad \underset{\vec{a}_x,\vec{a}_y,\vec{a}_z,\vec{\omega}_r}{\operatorname{argmin}} \quad -w_3 \frac{S_o^B S_o^C|_{C_o=T}}{\sum_{i=1}^{m} U_{oi}} + w_4||\Delta||_p \quad \text{s.t.} \quad |\vec{a}_x + \vec{a}_y + \vec{a}_z| < \xi_1, \; |\vec{\omega}_r| < \xi_2 \tag{7}$$

Similarly, w_3 and w_4 are the weights that balance the attack success rate and cost for CA.

For AA, to alter the prediction of an object of interest in the image, i.e., from Y_i to Y_i', the class of the altered prediction C_i' shall be the targeted class T, the product of $S_i^{B'}$ and $S_i^{C'}$ should be larger than the threshold, the intersection over union $U_{ii'}$ of the altered box B_i' and the benign box B_i shall be maximized to guarantee that it is transformed from the object of interest, and the magnitude of Δ should be minimized to reduce the attack cost. Thus, we give the objective function for targeted AA as follows:

$$\textbf{AA:} \quad \underset{\vec{a}_x,\vec{a}_y,\vec{a}_z,\vec{\omega}_r}{\operatorname{argmin}} \quad -w_5 U_{ii'} S_o^B S_o^C|_{C_o=T} + w_6||\Delta||_p \quad \text{s.t.} \quad |\vec{a}_x + \vec{a}_y + \vec{a}_z| < \xi_1, \; |\vec{\omega}_r| < \xi_2 \tag{8}$$

where w_5 and w_6 are the weights for AA.

In summary, to optimize PG attacks, we take the attack success rate, object location, and attack cost into comprehensive consideration, and design objective functions specific to each type of attack.

2) Bayesian Optimization: To optimize the designed objective functions, we employ Bayesian Optimization [23], a sequential design strategy for global optimization of black-box functions that does not assume any functional forms. The reason to choose Bayesian Optimization is that we regard object detectors as black-box and thus only prediction outputs are used in objective functions. As a result, the objective functions can be considered as black-box as well, and common derivative-based optimization methods such as gradient descent are not applicable.

We use the implementation from [28] in this paper, which works by constructing a posterior distribution of functions that best describes the target function. Since the target function is unknown to the algorithm, Bayesian Optimization first constructs a prior distribution over the target function using a Gaussian Process [5]. An exploration strategy, e.g., Upper Confidence Bound [10] or Expected Improvement [19], is then used to determine the next point to be explored. As the number of observations grows, the algorithm becomes more certain of which regions in parameter space to explore and the prior distribution is iteratively updated to form the posterior distribution over the target function, either until it converges or the iterations end.

D. Launching Sensor Output Manipulation Attack

With the optimized attack parameters, i.e., the desirable sensor outputs, we then inject crafted acoustic signals into the targeted camera system. To achieve it, we resort to the output biasing attack proposed in [43], which utilizes the sampling deficiencies at the analog-to-digital converter (ADC) and gives an adversary control over the inertial sensor's output for several seconds.

Manipulating a false sensor output via the output biasing attack has three steps: (1) finding the acoustic resonant frequency of the target sensor by frequency sweep, (2) stabilizing fluctuating false outputs into constant outputs by shifting the acoustic resonant frequency to induce a direct current alias at the ADC, and (3) reshaping the desired output signal by modulating it on top of the acoustic resonant frequency. The details of these steps can be found in [43].

The sensor outputs under PG attacks are the linear superposition of (1) the actual motions of the sensor, (2) the induced false motions caused by the output biasing attack, and (3) the false motions caused by ambient noises at other frequencies. Among the three components, the actual motions of the camera (sensor), if they exist, will be correctly compensated by the image stabilization and thus result in no blur. The false motions caused by both PG attacks and the ambient noises, on the contrary, will cause unnecessary compensation and lead to undesired blur. Considering that the strength of the ambient noises is far less than PG acoustic signals, PG attacks dominate the sensor outputs and the blur patterns in the final images, even when the sensor is in a moving vehicle or in a noisy environment.

VI. Evaluation

In this section, we evaluate PG attacks against object-detection systems. We consider two sets of evaluations in this paper: (1) simulated attack evaluation, where adversarial blurry images are generated by our blur model with public autonomous driving image datasets as input, and (2) real-world evaluation, where the blurred images are captured by a commercial camera product with an image stabilization system, i.e., a smartphone in a moving vehicle under the acoustic signal injection attacks. In both evaluations, the blurry images are fed into the CNN algorithms for object detection.

We use the attack success rate (SR) as the metric, which is the ratio of the number of successful attacks against an object detector over the total number of conducted attacks. In summary, we highlight the key result of PG attacks as follows:

- For the simulation evaluation, HA can achieve an overall SR of 100%, scenario-targeted CA can achieve an overall SR of 87.9%, and scenario-targeted AA can achieve an overall SR of 95.1% against the four academic object detectors YOLO V3/V4/V5 and Fast R-CNN, and the commercial one, i.e., YOLO 3D used in Baidu Apollo [3].
- For the real-world evaluation, PG attacks towards a Samsung S20 smartphone on a moving vehicle at four typical scenes demonstrate an average success rate of 98.3% for hiding attacks, 43.7% for creation attacks, and 43.1% for altering attacks.
- PG attacks are robust across various scenes, weathers, time periods of a day, and camera resolutions.

Table II: Summary of used datasets BDD100K and KITTI.

Dataset	Resolution	# of Images	Classes of Interest	# of Objects of Interest Detected				
				YOLO V3 [52]	YOLO V4 [2]	YOLO V5 [46]	Faster R-CNN [34]	Apollo [3]
BDD100K	1080×720	200	person, car, truck, bus, traffic light, stop sign	741	1531	1708	1125	993
KITTI [14]	1242×375	200		651	1425	1543	1059	904

A. Experimental Setup

Object Detectors. We evaluate PG attacks using four academic object detectors YOLO V3/V4/V5 [52], [2], [46] and Faster R-CNN [8], and one commercial object detector YOLO 3D used in Apollo [3]. The former four are representative models of one-stage and two-stage detectors for general object detection while the commercial YOLO 3D is a customized detector designed for autonomous driving. The backbone networks used for the pre-trained models YOLO V3/V4/V5 and Faster R-CNN are Darknet-53 and ResNet-101, respectively. The four academic detectors are all trained on the Common Objects in Context (COCO) dataset [9] and Apollo is trained on an unrevealed backbone network and dataset.

Classes of Interest. Given the real-world situations of autonomous driving, we consider 6 representative classes of interest in this paper: *person, car, truck, bus, traffic light,* and *stop sign*. The other classes are thus regarded as classes of uninterest. Objects belonging to classes of interest and classes of uninterest are then called objects of interest and objects of uninterest, respectively. For the aforementioned detectors, YOLO V3/V4/V5 and Faster R-CNN support the detection of all 6 classes of interest while Apollo does not support the detection of *traffic light* and *stop sign*.

Fine-grained Attack Forms. We implement 3 fine-grained attack forms in this paper: (1) untargeted, (2) scenario-targeted, and (3) targeted. For HA, we implement the targeted form, i.e., "one to none", which hides an object of interest in the scene. For CA, we implement all three attack forms, where untargeted CA ("none to any"), scenario-targeted CA ("none to a set"), and targeted CA ("none to one") create an object of any classes, any classes of interest, and a specific class, respectively. Similarly, for AA we have untargeted AA ("one to any") that alters an object of interest into an object of any other classes, scenario-targeted AA ("one to a set") that alters an object of interest into any objects of uninterest or an object of uninterest into any objects of interest, and targeted AA ("one to one") that alters an object of interest into another specific object of interest. We introduce scenario-targeted CA and AA since for autonomous driving, creating an object of any classes of interest, e.g., *person, car, etc*, or altering an object of uninterest into any objects of interest, e.g., *fire hydrant* to *person, car, etc*, may result in similar attack consequences such as improper stops, respectively. Similarly, altering an object of interest into any objects of uninterest, e.g., *car* to *bird, bottle, etc*, may have similar impacts such as resulting in car collisions. We envision this attack form reveals the practical impacts of PG attacks in autonomous driving, i.e., the capabilities to affect decisions.

Computing Platform. We implement the aforementioned object detectors in our lab with a server equipped with an Intel Xeon Gold 6139 CPU @2.30 GHz, a GeForce RTX 2080 Ti GPU, and 128 GB physical memory, which is also used to optimize attack parameters as well as generate adversarial blurry images.

B. Simulation Evaluation

In the simulation evaluation, we use adversarial blurry images generated by our model to spoof object detectors.

1) Datasets: We use two autonomous driving datasets BDD100K [53] and KITTI [14] in the simulation evaluation. BDD100K is the largest and most diverse open driving dataset so far for computer vision research, which covers different scenes, weather conditions, and times of day. KITTI is another widely-used dataset in mobile robotics and autonomous driving research, which captures real-world traffic situations with many static and dynamic objects in diverse scenarios. For both datasets, we randomly select 200 images for evaluation. The numbers of objects of interest detected in the selected BDD100K and KITTI images by each object detector are summarized in Tab. II. The performance variations between the detectors are caused by the following reasons: (1) Various object detectors perform differently in detecting small objects [32], and (2) Apollo does not support the detection of *traffic light* and *stop sign*.

2) Attack Effectiveness: In this section, we evaluate the effectiveness of hiding, creating, and altering attacks, respectively. Illustrations of hiding, creating, and altering attacks on BDD100K and KITTI images are shown in Appendix. B.

Hiding Attacks. The results summarized in Tab. III demonstrate the overall attack success rates for hiding attacks. For any target object detector, HA can achieve an overall success rate (SR) of 100% towards objects of interest in both BDD100K and KITTI images. Thus, HA shows a good performance against both academic and commercial object detectors.

Due to its high success rate, HA can pose a severe threat to object detectors, especially in autonomous vehicles. For instance, HA can hide any object of interest, e.g., *person*, *car*, or *traffic light*, on the road with 100% SR, which may lead to unintended operations of autonomous vehicles, resulting in severe consequences such as hitting the person or car, or driving through a red light.

Creating Attacks. The results summarized in Tab. IV demonstrate the effectiveness of creating attacks. For YOLO V3, untargeted CA can achieve overall SRs of 69.5% and 80.0% for BDD100K and KITTI images, scenario-targeted CA can achieve overall SRs of 68.5% and 77.0%, and targeted CA can achieve overall SRs of 16.6% and 19.7%. For YOLO V4, the overall SRs are 93.0% and 91.5% for untargeted CA, 88.5% and 85.0% for scenario-targeted CA, and 34.3% and

Table III: Effectiveness of Hiding Attacks.

Black-box Detector	Overall Attack Success Rate			
	BDD100K		KITTI	
YOLO V3/V4/V5	100% (Avg.)	person (100%), car (100%), truck (100%), bus (100%), traffic light (100%), stop sign (100%)	100% (Avg.)	person (100%), car (100%), truck (100%), bus (100%), traffic light (100%), stop sign (100%)
Faster R-CNN	100% (Avg.)	person (100%), car (100%), truck (100%), bus (100%), traffic light (100%), stop sign (100%)	100% (Avg.)	person (100%), car (100%), truck (100%), bus (100%), traffic light (100%), stop sign (100%)
Apollo	100% (Avg.)	person (100%), car (100%), truck (100%), bus (100%)	100% (Avg.)	person (100%), car (100%), truck (100%), bus (100%)

Table IV: Effectiveness of Creating Attacks.

Black-box Detector	Creating Attack	Overall Attack Success Rate			
		BDD100K		KITTI	
YOLO V3	Untargeted†	69.5%		80.0%	
	Scenario-targeted‡	68.5%		77.0%	
	Targeted§	16.6% (Avg.)	person (12.0%), car (57.5%), truck (8.5%), bus (7.0%), traffic light (13.5%), stop sign (1.0%)	19.7% (Avg.)	person (31.0%), car (58.0%), truck (8.5%), bus (7.0%), traffic light (10.5%), stop sign (3.0%)
YOLO V4	Untargeted	93.0%		91.5%	
	Scenario-targeted	88.5%		85.0%	
	Targeted	34.3% (Avg.)	person (42.5%), car (83.5%), truck (30.0%), bus (12.5%), traffic light (34.5%), stop sign (2.5%)	31.6% (Avg.)	person (52.5%), car (72.5%), truck (31.5%), bus (10.0%), traffic light (22.5%), stop sign (0.5%)
YOLO V5	Untargeted	97.5%		96.5%	
	Scenario-targeted	96.0%		95.0%	
	Targeted	37.7% (Avg.)	person (57.5%), car (90.5%), truck (23.5%), bus (14.0%), traffic light (37.5%), stop sign (3.0%)	39.8% (Avg.)	person (71.0%), car (87.0%), truck (25.5%), bus (9.5%), traffic light (40.5%), stop sign (5.5%)
Faster R-CNN	Untargeted	97.4%		97.9%	
	Scenario-targeted	95.9%		96.9%	
	Targeted	37.9% (Avg.)	person (65.0%), car (88.7%), truck (19.6%), bus (30.9%), traffic light (20.1%), stop sign (3.1%)	40.9% (Avg.)	person (88.7%), car (80.4%), truck (12.4%), bus (31.4%), traffic light (16.0%), stop sign (16.5%)
Apollo	Untargeted	91.2%		96.0%	
	Targeted	40.2% (Avg.)	person (47.4%), car (79.9%), truck (18.0%), bus (15.5%)	46.2% (Avg.)	person (67.7%), car (83.8%), truck (15.2%), bus (18.2%)

† Untargeted: none to any ‡ Scenario-targeted: none to a set §Targeted: none to one

31.6% for targeted CA. For YOLO V5, the overall SRs are 97.5% and 96.5% for untargeted CA, 96.0% and 95.0% for scenario-targeted CA, and 37.7% and 39.8% for targeted CA. For Faster R-CNN, the overall SRs are 97.4% and 97.9% for untargeted CA, 95.9% and 96.9% for scenario-targeted CA, and 37.9% and 40.9% for targeted CA. For Apollo, since it hardly detects any object of uninterest, scenario-targeted CA is basically equal to untargeted CA. Thus, we conduct the untargeted and targeted CA only. For untargeted CA, it can achieve overall SRs of 91.2% and 96.0% for BDD100K and KITTI images while for targeted CA, it can achieve overall SRs of 40.2% and 46.2%. Note that higher overall SRs of targeted CA here do not indicate that Apollo is more vulnerable. The reason for the high overall SRs is that Apollo does not detect *traffic light* and *stop sign*, which are more difficult to create. Among five detectors, YOLO V5 and Faster R-CNN are most vulnerable to CA. For both BDD100K and KITTI images, the Top 3 objects that are most likely to be injected are *car, person,* and *truck*.

Targeted CA is difficult in our case since we try to create a non-existent object by manipulating its surrounding pixels without modifying or adding any physical objects or lights like prior works [57], [22]. Nevertheless, CA, especially scenario-targeted CA, which we assume is more practical, can also pose severe threats to autonomous driving by creating an object of interest, e.g., *person, car, etc*, on the road with high successful rates, which can lead to malicious driving behaviors such as emergency brakes or detours.

Altering Attacks. The results summarized in Tab. V demonstrate the effectiveness of altering attacks. When against YOLO V3, untargeted AA can achieve overall SRs of 91.8% and 98.7% for BDD100K and KITTI images, scenario-targeted AA can achieve overall SRs of 82.2% and 96.9%, and targeted AA can achieve overall SRs of 23.7% and 19.8%. For YOLO V4, the overall SRs are 98.1% and 97.2% for untargeted AA, 97.9% and 95.6% for scenario-targeted AA, and 32.3% and 28.3% for targeted AA. For YOLO V5, the overall SRs are 99.6% and 99.3% for untargeted AA, 98.2% and 97.1% for scenario-targeted AA, and 34.1% and 32.4% for targeted AA. For Faster R-CNN, the overall SRs are 98.0% and 99.4% for untargeted AA, 95.5% and 97.2% for scenario-targeted AA, and 20.5% and 30.6% for targeted AA. For Apollo, since it hardly detects any object of uninterest, scenario-targeted AA is basically equal to HA or scenario-targeted CA. Therefore, we conduct the untargeted and targeted AA for Apollo only, and the overall SRs achieved are 67.0% and 73.0% for untargeted

Table V: Effectiveness of Altering Attacks.

Black-box Detector	Altering Attack	Overall Attack Success Rate			
		BDD100K		KITTI	
YOLO V3	Untargeted†	91.8%		98.7%	
	Scenario-targeted‡	82.2% (Avg.)	OOI*→ OOU** (82.1%), OOU → OOI (75%)	96.9% (Avg.)	OOI → OOU (96.8%), OOU → OOI (100%)
	Targeted§	23.7% (Avg.)	Top 5: bus → car (100%), stop sign → car (100%), truck → car (96.7%), bus → truck (88.9%), traffic light → car (77.8%)	19.8% (Avg.)	Top 5: bus → car (100%), truck → car (92.9%), traffic light → car (84.2%), bus → person (83.3%), bus → truck (66.7%)
YOLO V4	Untargeted	98.1%		97.2%	
	Scenario-targeted	97.9% (Avg.)	OOI → OOU (97.8%), OOU → OOI (100%)	95.6% (Avg.)	OOI → OOU (95.5%), OOU → OOI (97.3%)
	Targeted	32.3% (Avg.)	Top 5: bus → car (100%), truck → car (97.8%), car → person (95.6%), person → car (90.1%), car → truck (73.2%)	28.3% (Avg.)	Top 5: bus → person (100%), truck → car (96.5%), bus → car (95.9%), car → person (82.3%), car → truck (77.9%),
YOLO V5	Untargeted	99.6%		99.3%	
	Scenario-targeted	98.2% (Avg.)	OOI → OOU (98.1%), OOU → OOI (100%)	97.1% (Avg.)	OOI → OOU (96.9%), OOU → OOI (99.6%)
	Targeted	34.1% (Avg.)	Top 5: truck → car (97.8%), bus → car (97.2%), traffic light → car (90.3%), person → car (89.2%), person → truck (76.2%)	32.4% (Avg.)	Top 5: bus → person (100%), bus → car (100%), truck → car (92.1%) bus → truck (85.2%) person → car (81.1%)
Faster R-CNN	Untargeted	98.0%		99.4%	
	Scenario-targeted	95.5% (Avg.)	OOI → OOU (95.3%), OOU → OOI (100%)	97.2% (Avg.)	OOI → OOU (96.9%), OOU → OOI (100%)
	Targeted	20.5% (Avg.)	Top 5: truck → car (94.2%), bus → car (92.9%), person → car (75.9%) stop sign → person (75.0%), person → bus (70.1%)	30.6% (Avg.)	Top 5: bus → person (100%), car → person (97.6%), truck → car (97.4%) stop sign → person (95.7%), truck → person (92.3%)
Apollo	Untargeted	67.0%		73.1%	
	Targeted	16.6% (Avg.)	Top 5: truck → car (76.0%), person → car (75.0%), bus → car (68.4%), bus → truck (26.3%), person → truck (25.8%)	18.3% (Avg.)	Top 5: truck → car (75.0%), person → car (70.2%), bus → car (66.7%), truck → bus (25.0%), bus → truck (25.0%)

† Untargeted: one to any ‡ Scenario-targeted: one to a set §Targeted: one to one
* OOI: object of interest ** OOU: object of uninterest

AA, and 16.6% and 18.3% for targeted AA. Among five detectors, YOLO V5 and Faster R-CNN are most vulnerable to AA. For BDD100K images, the object most likely to be altered into is *car* when against five target detectors. For KITTI images, it will be *car* when against YOLO V3/V4/V5 and Apollo, and *person* when against Faster R-CNN.

Similar to targeted CA, targeted AA is difficult but both untargeted AA and scenario-targeted AA achieve great performance against these detectors. Specifically, scenario-targeted AA can alter objects of interest into objects of uninterest or vice verse with high success rates. The former is similar to HA and can render autonomous vehicles unresponsive to foreground people or cars. The latter is similar to CA and can deceive autonomous vehicles into taking unnecessary actions such as speed cuts or emergency brakes. Both two cases are likely to cause severe traffic accidents.

3) Attack Robustness: In addition to the attack effectiveness, we evaluate the attack robustness of PG attacks across different scenes, weathers, times of day, and camera resolutions, since autonomous driving systems usually take images outdoors and thus may suffer from those impacts, if any.

For this set of experiments, we use the BDD100K dataset, and classify the selected images according to their own annotations. For the object detector, we use Faster R-CNN.

Scenes. For various scenes, the present objects and backgrounds may have variations, e.g., the highway is likely to have more cars while the residential street is likely to have more people, which may have impacts on the attack performance. Based on the actual annotations, the selected BDD100K images can be classified into 3 typical scenes: (1) City street (104 images), (2) Highway (61 images), and (3) Residential street (35 images). To investigate the attack robustness across different scenes, we evaluate the attack performance of HA, CA, and AA against Faster R-CNN for each scene, respectively.

The results shown in Fig. 7 demonstrate that the performances of HA, CA, and AA show no obvious discrepancy across various scenes. It is because PG attacks do not rely on any physical objects or lights, which enables the applicability of PG attacks in numerous autonomous driving scenes.

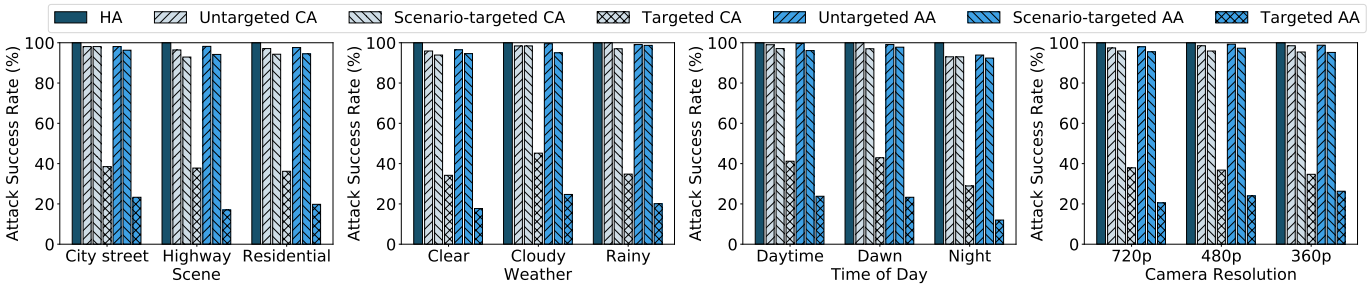


Figure 7: Attack robustness on various scenes, weathers, time of day, and camera resolutions.

Weathers. Autonomous driving systems usually take images outdoors, which may be affected by weathers. To investigate, we classify the selected BDD100K images based on their weather conditions: (1) Clear (103 images), (2) Cloudy (64 images), and (3) Rainy (33 images). From the results shown in Fig. 7, we can find that attacks towards images taken in various weathers may show different performances but the differences are slight. Since the dataset used to train the object detector usually contains images of various weathers, we believe that PG attacks are likely to be robust across various weathers.

Times of Day. Similar to weathers, images taken outdoors may be affected by light conditions. To investigate, we classify the selected BDD100K images based on the time they are taken: (1) Daytime (103 images), (2) Dawn (64 images), and (3) Night (33 images). From the results shown in Fig. 7, we observe that the attack success rates of CA and AA are slightly lower at night. This is because images taken at night have more dark pixels in both objects and backgrounds. As a result, the new pixels created by CA and AA are likely to be unitary in color, which decreases the attack performances.

Camera Resolutions. Another factor that may affect the attack performance is the camera resolution since autonomous vehicles may use cameras of various resolutions to take images. To study its impact, we re-sample the BDD100K images from (1) 720p, to (2) 480p, and (3) 360p to simulate cameras of different resolutions. The results shown in Fig. 7 demonstrate that HA, CA, and AA show similar performances across different camera resolutions. This suggests that PG attacks are likely to be applicable to object-detection systems with various camera resolutions.

C. Real-world Attack Evaluation

In the real-world evaluation, we target a smartphone on a moving vehicle and conduct PG attacks towards it inside the vehicle via acoustic signals.

1) Setup: The target system, the attack devices, and the attack methodology for the real-world evaluation are as follows:

Target System. We target a Samsung S20 smartphone mounted on an Audi Q3 car that serves as its computer vision. The target smartphone uses an accelerometer and a gyroscope for motion feedback and compensates the images with both OIS and EIS.

Attack Devices. We launch PG attacks towards the target smartphone inside the car. The used attack devices include a Rigol DG5072 Arbitrary Waveform Generator [35] for acoustic signal generation, a Fostex FT17H Horn Super Tweeter

Figure 8: Experimental setups. An ultrasonic speaker beams designed acoustic signals to the target smartphone in a moving vehicle to launch PG attacks.

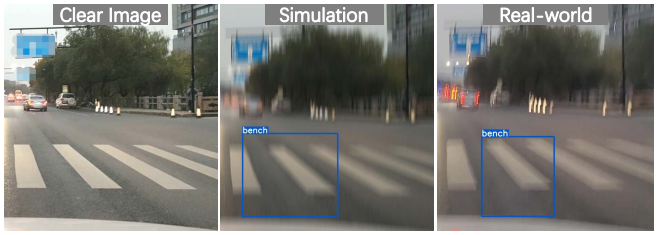


Figure 9: An illustration of an original image, its simulated and real-world adversarial blurry images.

speaker for acoustic signal beaming, an audio amplifier used before the speaker for increasing the volume of the ultrasound, an uninterrupted power supply used as the main power source, and a DC power supply used as the power source of the audio amplifier, as shown in Fig. 8. We set up these devices inside the car and place the speaker towards the target smartphone at a distance of around 10 cm and a power of 8-10 W to launch PG attacks.

Attack Methodology. During the experiments, we drive the vehicle around in the city with an average speed of 20-30 km/h (for safety reasons) and launch three types of PG attacks (i.e., hiding, creating, altering) at four representative scenes: (1) city lane, (2) city crossroad, (3) tunnel, and (4) campus road. In each attack, we first take a clear image of the scene and generate optimized attack parameters with the remote server. Then, we induce the desired sensor outputs by emitting the acoustic signals derived by optimized attack parameters. Once the acoustic signals become stable, a 5-second video (amounts to 150 images) is recorded with the victim smartphone for each attack. In total, we collect 12 videos (1800 images) for

Table VI: Results of real-world attacks.

Attacks	Scenes							
	City Lane		City Crossroad		Tunnel		Campus Road	
	Goal	SR	Goal	SR	Goal	SR	Goal	SR
Hiding	hide a "person"	98.1%	hide a "car"	100%	hide a "car"	100%	hide a "car"	95.2%
Creating	create a "truck"	17.1%	create a "bus"	75.7%	create a "truck"	43.9%	create a "person"	37.9%
Altering	alter a "car" into a "bus"	81.4%	alter a "car" into a "boat"	54.4%	alter a "traffic light" into a "person"	15.0%	alter a "car" into a "person"	21.7%

the real-world evaluation, which are fed into the Faster R-CNN object detector for performance evaluation.

2) Attack Effectiveness: An illustration of a clear image and its simulated and real-world adversarial blurry images is shown in Fig. 9, from which we can observe that both blurry images show similar patterns and both are recognized mistakenly with a non-existing bench on the pavement. It suggests that the simulated images are representative of the ones created in the presence of real attacks, even in a moving vehicle. To report the effectiveness of the real-world PG attacks quantitatively, we calculate the SR as the number of images that are successfully attacked over the total number of images collected during each attack. The results shown in Tab. VI demonstrate that all three types of PG attacks are feasible even in a moving vehicle.

Impact of Ambient Audios. As analyzed in Sec. V-D, audios or noises other than those from PG attacks have little impact on the sensor outputs since they cannot force the sensor into resonance and thus the caused motions are subtle. To validate it experimentally, we measure sensor outputs in the presence of PG attacks and when emitting ambient audios at various strength against the target smartphone. The ambient audios include (1) white noises, (2) people talking, and (3) sine waves of various frequencies played by a speaker. From the results shown in Fig. 10, we observe that even when played at the equal volume, none of the ambient audios are able to manipulate the sensor outputs while PG attacks can induce false sensor outputs at a much larger scale. Thus, PG attacks would be effective even in a noisy environment.

Impact of Attack Distances. The strengths of the modulated acoustic signals received by the sensors depend on the signal transmitting power and the distance between the speaker and the target smartphone. To evaluate the trade-offs between the power levels and attack ranges, we conduct experiments to investigate the powers required to induce the same sensor output for various speaker-smartphone distances. During the experiments, our goal is to induce the gyroscope output to be 0.5 rad/s, and we vary the speaker-smartphone distance from 10 cm to 120 cm. The results shown in Fig. 11 demonstrate that a larger attack power level is needed to induce the same sensor output at a longer distance. Particularly, an attack power of 10 W suffices to launch an attack from 1.1 m away, which can be achieved by adversaries with modest budgets.

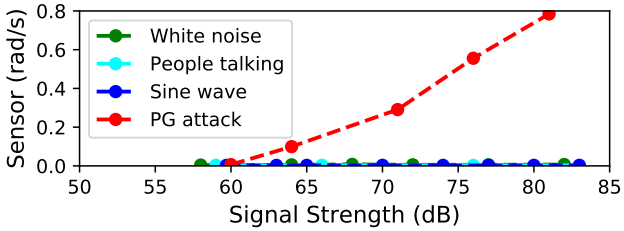


Figure 10: Compared with ambient audios, PG attacks demonstrate significant sensor output manipulation capabilities.

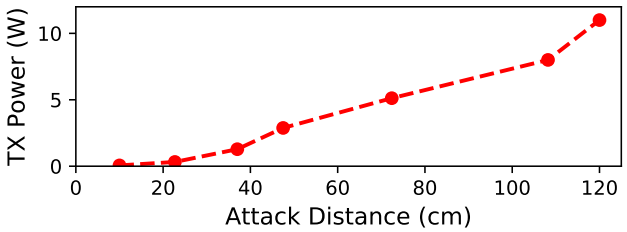


Figure 11: A larger attack power level is needed to induce the same sensor output at a longer distance.

VII. Discussion

A. Countermeasures

PG attacks exploit the vulnerabilities of MEMS inertial sensors embedded in image stabilization system and mislead the object detection algorithms to ultimately affect the decisions of systems such as autonomous vehicles. In this section, we provide several potential defense mechanisms by increasing the difficulty of launching our attacks.

MEMS Inertial Sensors Safeguarding. PG attacks employ MEMS inertial sensors' vulnerabilities to acoustic injection attacks as the attack entrances, which are enabled by two attack surfaces: (1) susceptibility of the micro inertial sensing structure to resonant acoustic signals, and (2) incapability of signal processing/ conditioning circuits to handle out-of-band analog signals properly. For the former, acoustic isolation can be employed by surrounding sensors with microfibrous metallic fabric [39] or MEMS fabricated acoustic metamaterial [54]. For the latter, a secure low-pass filter can be designed to eliminate out-of-band analog signals, which suppresses the adversary's capability of controlling sensor outputs via signal aliasing [43]. In addition, a microphone can be employed to detect acoustic injection attacks and alert the system to the possible existence of our attacks.

Image Stabilization Techniques. Another exploited vulnerability is that image stabilization techniques conduct motion compensation based on unreliable inertial sensor readings. We envision it can be mitigated by adding an additional digital image stabilization that de-blurs images with the pixel information only after the existing one, which serves as a second barrier to blur images and thus our attacks.

Object Detection Algorithms. From the aspect of object detection algorithms, possible defense mechanisms include: (1) modifying the input images to disturb or even remove

adversarial blur patterns via a guided de-noiser such as [16], [30], [56], which may mitigate the threats but affect the detection efficiency and accuracy, and (2) improving detection models by raising the detection criterion or incorporating adversarial training, which may increase the difficulty of our attacks but impair models' generalization abilities.

Sensor Fusion Techniques. Another complementary defense approach is to exploit sensor fusion for decision making. Autonomous vehicles can employ multiple types of sensors, e.g., LiDARs, radars combined with cameras to perceive the environment. It can increase the attack overhead in terms of cost and time by requiring the adversary to target multiple sensors simultaneously.

B. Limitation

`PG` attacks still have the following limitations at present. First, as the first attempt, even though we successfully launch attack signals towards smartphone cameras, we have not conducted end-to-end attacks towards on-board cameras on real autonomous vehicles. Second, in the current simulation model, we assume the image stabilization hardware conducts motion compensation ideally with a motion equivalent but reserve to the camera motion. However, in practice the image stabilization algorithms can be more complicated. Incorporating a more realistic image stabilization model may help further improve the attack effectiveness. Third, `PG` attacks mainly focus on disturbing object detection results on a single image at present though the real-world experiments have demonstrated the possibility of continuous attacks. However, a more effective continuous attack requires further investigations and methodology improvements. We designate the aforementioned issues as our future work.

VIII. Related Work

Adversarial Attacks against Computer Vision Systems. The vulnerabilities of computer vision systems have been actively investigated with the recent rise of face recognition, autonomous vehicles and surveillance systems. Existing attacks can be classified into two categories based on the targeted component: (1) attacks on the camera hardware, and (2) attacks on the object detector or classifier. Attacks in the first category aim to make a camera capture malicious images that may deceive both human eyes and classifiers. For example, strong lights can blind a camera and cause denial-of-service [29], [44], [50], and a fake road sign projected onto a wall may be recognized as a real one [27]. Attacks in the second category target at compromising object classifiers or detectors via adversarial images, which can deceive computer vision without being noticed by humans. Earlier work in this field mainly focused on generating adversarial images in the digital domain [42], [25], [7], [24], [40], e.g., by adding optimized noises directly to the images. However, digital attacks may not be practical when the target is a real-world computer vision system such as autonomous vehicles, which requires the attacker to inject adversarial noises via a camera from the physical world. Recent studies have demonstrated the feasibility of physical adversarial attacks, e.g., by attaching physical stickers or patterns to the object of interest [12], [38], [57], [36] or on the camera lenses [21], or by projecting light to the object of interest [59] or the camera [22]. Most of the existing attacks require an adversary to either modify an object's visual appearance or project visible lights, which may be noticed by alert users. In this work, we launch physical adversarial attacks against object detection via acoustic injection on the image stabilization system, which at ultrasonic frequencies can be totally imperceptible to users.

Acoustic Injection Attacks against Inertial Sensors. A wide range of control systems depend on the timely feedback of MEMS inertial sensors to make critical decisions [45], which however can be threatened by acoustic injection attacks at resonant frequencies. Son et al. [37] first presented acoustic attacks on MEMS gyroscopes, which can cause denial-of-service of the sensor and make a drone crash. Trippel et al. [43] proposed output biasing and output control attacks that can achieve elaborate control over the output of MEMS accelerometers using modulated sounds. Wang et al. [48] developed a sonic gun and demonstrated the impact of acoustic attacks on various smart devices such as virtual reality devices, drones, and self-balancing vehicles. Tu et al. [45] devised side-swing and switching attacks to manipulate the output of MEMS gyroscopes and accelerometers. Our work is inspired by the aforementioned studies. We use acoustic signals to control the output of inertial sensors, which will provide a false feedback to the image stabilization system and induce adversarial blur patterns in the captured images that can deceive the object detectors.

IX. Conclusion

Our paper identifies a new class of system-level vulnerabilities resulting from the combination of the emergent image stabilization hardware susceptible to acoustic manipulation and the object detection algorithms of machine learning subject to adversarial examples. Our `Poltergeist` attacks exploit such vulnerabilities to hide, create, or alter object detection results. Evaluation results demonstrate the effectiveness of `Poltergeist` attacks against four academic object detectors YOLO V3/V4/V5 and Fast R-CNN, and one commercial detector Apollo. While it's clear that there exist pathways to cause computer vision systems to fail with acoustic injection, it's not clear what products today are at risk. Rather than focus on today's nascent autonomous vehicle technology, we model the limits in simulation to understand how to better prevent future yet unimagined autonomous vehicles from being susceptible to acoustic attacks on image stabilization systems.

Acknowledgments

We thank the anonymous reviewers, Yan Tong and Himaja Motheram for their valuable comments. This work is supported by China NSFC Grant 62071428, 61941120, 61925109, in part by a gift from Analog Devices Inc. and NSF CNS-2031077. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSFC or NSF.

REFERENCES

[1] Adobe Support, "Use the blur gallery in photoshop," 2017, https://helpx.adobe.com/photoshop/using/blur-gallery.html.

[2] H.-Y. M. L. Alexey Bochkovskiy, Chien-Yao Wang, "Yolov4: Optimal speed and accuracy of object detection," *arXiv*, 2020.

[3] ApolloAuto, 2017, https://github.com/ApolloAuto/apollo.

[4] D. Bereska, K. Daniec, S. Fras, K. Jedrasiak, M. Malinowski, and A. Nawrat, "System for multi-axial mechanical stabilization of digital camera," in *Vision Based Systemsfor UAV Applications*. Springer, 2013, pp. 177–189.

[5] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS'08)*, 2008, pp. 153–160.

[6] B. Cardani, "Optical image stabilization for digital cameras," *IEEE Control Systems Magazine*, vol. 26, no. 2, pp. 21–22, 2006.

[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 38th IEEE Symposium on Security and Privacy (S&P'17)*. IEEE, 2017, pp. 39–57.

[8] X. Chen and A. Gupta, "An implementation of faster rcnn with study for region sampling," *arXiv preprint arXiv:1702.02138*, 2017.

[9] Common Objects in Context Dataset, 2018, https://cocodataset.org/.

[10] E. Contal, D. Buffoni, A. Robicquet, and N. Vayatis, "Parallel gaussian process optimization with upper confidence bound and pure exploration," in *Proceedings of the 2013 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD'13)*. Springer, 2013, pp. 225–240.

[11] S. Erturk, "Digital image stabilization with sub-image phase correlation based global motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1320–1325, 2003.

[12] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018, pp. 1625–1634.

[13] K. Fu and W. Xu, "Risks of trusting the physics of sensors," *Communications of the ACM*, vol. 61, no. 2, pp. 20–23, 2018.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[15] R. Girshick, "Fast r-cnn," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV'15)*, 2015, pp. 1440–1448.

[16] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi, "From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 2319–2328.

[17] M. Gu, *Advanced optical imaging theory*. Springer Science & Business Media, 2000, vol. 75.

[18] Honeywell, "Inertial measurement unit," 2020, https://www.inised.de/en/sensors/imu/.

[19] J. P. Kleijnen, W. Van Beers, and I. Van Nieuwenhuyse, "Expected improvement in efficient global optimization through bootstrapped kriging," *Journal of global optimization*, vol. 54, no. 1, pp. 59–73, 2012.

[20] D. F. Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu, "Ghost talk: Mitigating emi signal injection attacks against analog sensors," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy (S&P'13)*. IEEE, 2013, pp. 145–159.

[21] J. B. Li, F. R. Schmidt, and J. Z. Kolter, "Adversarial camera stickers: A physical camera attack on deep learning classifier," *arXiv preprint arXiv:1904.00759*, vol. 2, no. 2, 2019.

[22] Y. Man, M. Li, and R. Gerdes, "Ghostimage: Remote perception attacks against camera-based image classification systems," in *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID'20)*, 2020.

[23] J. Mockus, *Bayesian approach to global optimization: theory and applications*. Springer Science & Business Media, 2012, vol. 37.

[24] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 1765–1773.

[25] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 2574–2582.

[26] C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization," in *Proceedings of 13th International Conference on Pattern Recognition (ICPR'96)*, vol. 3. IEEE, 1996, pp. 284–288.

[27] B. Nassi, D. Nassi, R. Ben-Netanel, Y. Mirsky, O. Drokin, and Y. Elovici, "Phantom of the adas: Phantom attacks on driver-assistance systems." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 85, 2020.

[28] F. Nogueira, "Bayesian Optimization: Open source constrained global optimization tool for Python," 2014, https://github.com/fmfn/BayesianOptimization.

[29] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[30] S. Ramakrishnan, S. Pachori, A. Gangopadhyay, and S. Raman, "Deep generative filter for motion deblurring," in *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW'17)*, 2017, pp. 2993–3000.

[31] B. Ransford, D. B. Kramer, D. Foo Kune, J. Auto de Medeiros, C. Yan, W. Xu, T. Crawford, and K. Fu, "Cybersecurity and medical devices: a practical guide for cardiac electrophysiologists," *Pacing and Clinical Electrophysiology*, vol. 40, no. 8, pp. 913–917, 2017.

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 779–788.

[33] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems (NPIS'15)*, 2015, pp. 91–99.

[35] Rigol DG5072, 2020, https://www.batronix.com/shop/waveform-generator/Rigol-DG5072.html.

[36] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*, 2016, pp. 1528–1540.

[37] Y. Son, H. Shin, D. Kim, Y. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking drones with intentional sound noise on gyroscopic sensors," in *Proceedings of the 24th USENIX Conference on Security Symposium (USENIX Security'15)*, 2015, pp. 881–896.

[38] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramer, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," in *Proceedings of the 12th USENIX Workshop on Offensive Technologies (WOOT'18)*, 2018.

[39] P. Soobramaney, G. Flowers, and R. Dean, "Mitigation of the effects of high levels of high-frequency noise on mems gyroscopes using microfibrous cloth," in *Proceedings of the 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC-CIE'15)*, vol. 57113. American Society of Mechanical Engineers, 2015, p. V004T09A014.

[40] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[41] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu, "Light commands: Laser-based audio injection attacks on voice-controllable systems," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security'20)*, 2020, pp. 2631–2648.

[42] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[43] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks," in *Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P'17)*. IEEE, 2017, pp. 3–18.

[44] K. N. Truong, S. N. Patel, J. W. Summet, and G. D. Abowd, "Preventing camera recording by designing a capture-resistant environment," in *Proceedings of the 7th International Conference on Ubiquitous Computing (UbiComp'05)*. Springer, 2005, pp. 73–86.

[45] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors," in *Proceedings of the 27th USENIX Conference on Security Symposium (USENIX Security'18)*, 2018, pp. 1545–1562.

[46] Ultralytics, "Yolo v5," 2020, https://github.com/ultralytics/yolov5.

[47] F. Vella, A. Castorina, M. Mancuso, and G. Messina, "Digital image stabilization by adaptive block motion vectors filtering," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 796–801, 2002.
[48] Z. Wang, K. Wang, B. Yang, S. Li, and A. Pan, "Sonic gun to smart devices: Your devices lose control under ultrasound/sound," *Black Hat USA*, pp. 1–50, 2017.
[49] W. Xu, C. Yan, W. Jia, X. Ji, and J. Liu, "Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5015–5029, 2018.
[50] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, no. 8, p. 109, 2016.
[51] C. Yan, G. Zhang, X. Ji, T. Zhang, T. Zhang, and W. Xu, "The feasibility of injecting inaudible voice commands to voice assistants," *IEEE Transactions on Dependable and Secure Computing*, 2019.
[52] YOLO V3, 2018, https://pjreddie.com/darknet/yolo/.
[53] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, 2020, pp. 2636–2645.
[54] W. N. Yunker, C. B. Stevens, G. T. Flowers, and R. N. Dean, "Sound attenuation using microelectromechanical systems fabricated acoustic metamaterials," *Journal of Applied Physics*, vol. 113, no. 2, p. 024906, 2013.
[55] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*, 2017, pp. 103–117.
[56] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. Lau, and M.-H. Yang, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018, pp. 2521–2529.
[57] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't believing: Towards more robust adversarial attack against real world object detectors," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*, 2019, pp. 1989–2004.
[58] S. Zheng, L. Xu, and J. Jia, "Forward motion deblurring," in *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV'13)*, 2013, pp. 1465–1472.
[59] Z. Zhou, D. Tang, X. Wang, W. Han, X. Liu, and K. Zhang, "Invisible mask: Practical attacks on face recognition with infrared," *arXiv preprint arXiv:1803.04683*, 2018.

APPENDIX

A. Pixel Motions Caused by CCMs in the x-axis, y-axis and z-axis DOFs

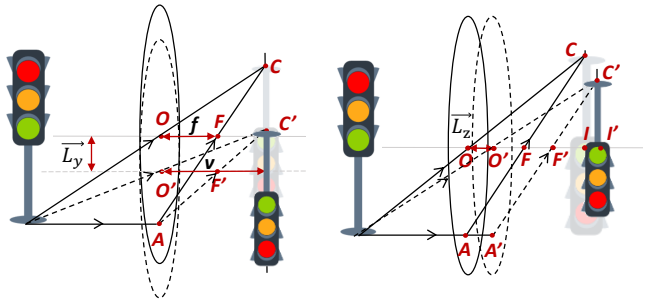
The x-axis and y-axis DOFs. CCMs in the x-axis or the y-axis DOF give linear pixel motions in the same DOF but in the opposite direction. Take the y-axis for an instance. When taking a photo towards an object such as the traffic light shown in Fig. 12(a), the y-axis acceleration $-\vec{a_y}$ introduces a camera displacement of

$$\overline{OO'} = \overline{FF'} = \vec{L}_y = -\frac{1}{2}\vec{a}_y T^2 \tag{9}$$

during the camera exposure time T, where O and O' are the original and moved camera centers, and F and F' are the original and moved camera focuses, respectively. For an arbitrary pixel C in the image, the camera displacement $\vec{L}_y$ introduces a relative pixel displacement of $\overline{CC'} - \overline{FF'}$. Based on the Optical Imaging Theory [17], we have

$$\overline{OO'} // \overline{FF'} // \overline{CC'} \tag{10}$$

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \tag{11}$$



(a) CCMs in the y-axis DOF. (b) CCMs in the z-axis DOF.

Figure 12: Illustrations of CCMs in the y-axis and z-axis DOFs.

where f is the camera focal length, u is the object distance, and v is the image distance. As a result, we have

$$\triangle AFF' \backsim \triangle ACC' \tag{12}$$

$$\frac{\overline{FF'}}{\overline{CC'}} = \frac{\overline{AF}}{\overline{AC}} = \frac{f}{v} \tag{13}$$

Then, $\overline{CC'} - \overline{FF'}$ can be calculated as

$$\overline{CC'} - \overline{FF'} = (\frac{v}{f} - 1)\overline{FF'} = \frac{1}{\frac{u}{f} - 1}\overline{FF'} \tag{14}$$

Since the object distance is usually much larger than the focal length, i.e., $u \gg f$, we can assume that $\overline{CC'} - \overline{FF'} \approx \frac{f}{u}\vec{L}_y$. As the formed image is reverse to the scene (camera), the camera displacement $\vec{L}_y$ in fact gives rise to a pixel motion of $-\frac{f}{u}\vec{L}_y$ for every pixel in the output photo. CCMs motions in the x-axis DOF obey the same rule. As a result, for a CCM in the x-axis and y-axis DOFs with accelerations of $\{-\vec{a}_x, -\vec{a}_y\}$, we have a pixel motion of $\{\frac{f}{2u}\vec{a}_x T^2, \frac{f}{2u}\vec{a}_y T^2\}$.

The z-axis DOF. CCMs in the z-axis DOF give pixel motions backwards or towards the image center [58]. As shown in Fig. 12(b), the z-axis acceleration $-\vec{v_z}$ introduces a camera displacement of

$$\overline{OO'} = \overline{FF'} = \vec{L}_z = -\frac{1}{2}\vec{a}_z T^2 \tag{15}$$

which results in a changed image distance (from $\overline{OI}$ to $\overline{O'I'}$), and pixel motions towards the image center (e.g., from C to C'). For each pixel in the image, the pixel motion caused by CCMs in the z-axis DOF depends on the camera displacement $\vec{L}_z$ and its distance to the image center I [16]. For an arbitrary pixel C, its pixel displacement can be given as

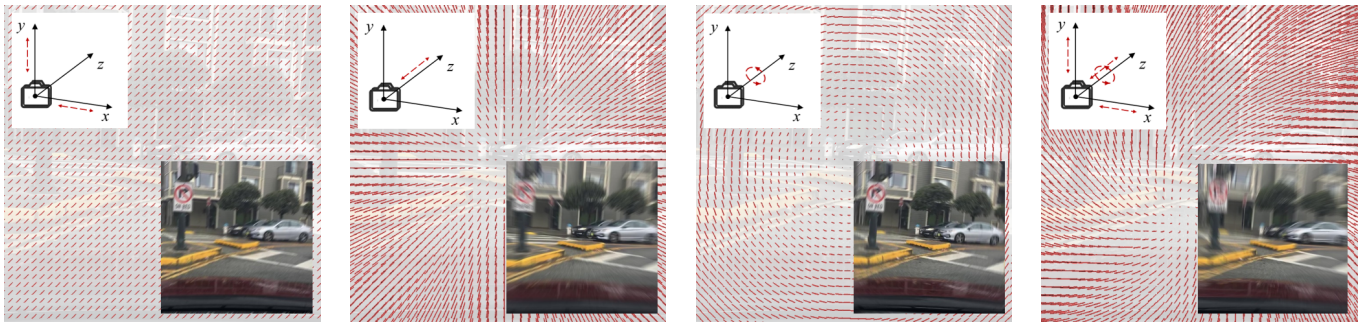
$$\overline{I'C'} - \overline{IC} = (\frac{\overline{I'C'}}{\overline{IC}} - 1)\overline{IC} \tag{16}$$

Since $\triangle F'I'C' \backsim \triangle FIC$ and $\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$, we have

$$\frac{\overline{I'C'}}{\overline{IC}} - 1 = \frac{\overline{F'I'}}{\overline{FI}} - 1 = \frac{v' - f}{v - f} - 1 = \frac{u - u'}{u - f} \tag{17}$$

Since $u - u' = \vec{L}_z$ and $u \gg f$, we can assume $\overline{I'C'} - \overline{IC} \approx \frac{\vec{L}_z}{u}\overline{IC}$. Thus, a CCM in the z-axis DOF with an acceleration of $-\vec{a}_z$ in fact gives rise to a pixel motion of $\frac{\vec{a}_z T^2}{2u} r_o$ towards the image center, where r_o refers to the pixel distance to the image center.

B. Illustrations of Linear, Radial, Rotational, and Heterogeneous Motion Blur



(a) Linear motion blur caused by pixel motions in the x-axis and y-axis DOFs ($\vec{L}_{xy}$=15 pixels, $\alpha = 45°$).

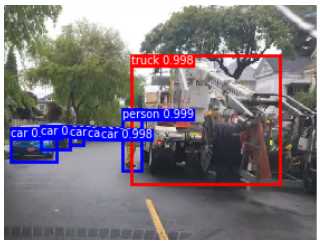
(b) Radial motion blur caused by pixel motions in the z-axis DOF ($p = 0.1$).

(c) Rotational motion blur caused by pixel motions in the roll DOF (C= image center, $\beta = 3°$).

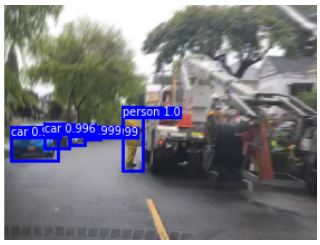
(d) Heterogeneous motion blur caused by pixel motions in (a)-(c).

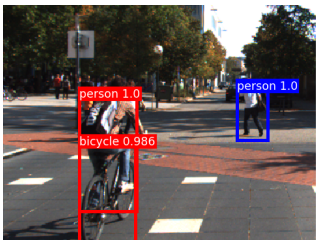
Figure 13: Illustrations of linear (a), radial (b), rotational (c), and heterogeneous (d) motion blur.

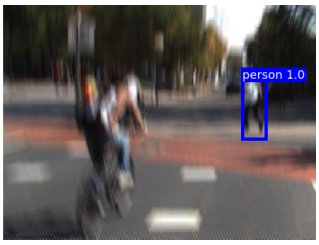
C. Illustrations of Hiding, Creation, and Alteration Attacks



(a) The *truck* in the clear image (left) is hidden after blurring (right).

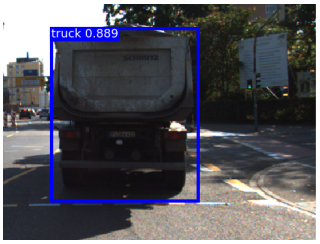


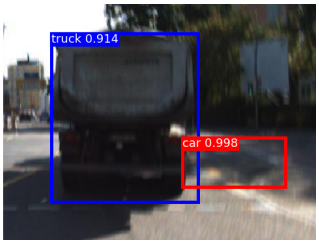




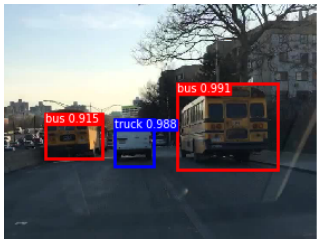
(b) The *person* and *bicycle* in the clear image (left) is hidden after blurring (right).

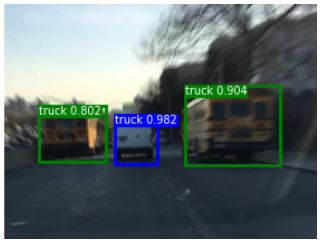
(c) A *person* is created with a high confidence score after blurring.



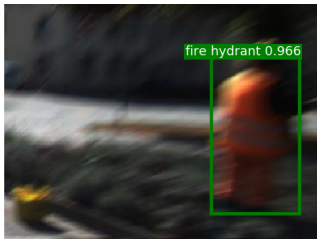


(d) A *car* is created with slight blur.





(e) Two *buses* in the clear image (left) are altered into *trucks* after blurring (right).



(f) The *person* in the clear image (left) is altered into a *fire hydrant* after blurring (right).

Figure 14: Illustrations of hiding (a-b), creation (c-d), and alteration (e-f) attacks on BDD100K (left) and KITTI (right) images.