Defense against Adversarial Attacks for both Camera and LiDAR

Elaine Yao
University of British Columbia

1 Introduction

Autonomous Vehicles(AVs) are playing an important role in future transportation. Large companies such as Google, Uber [2] are racing to develop AVs and some of them have already been deployed on the road. AVs rely on different sensors [3] such as cameras, LiDARs, Radars, IMU(Inertial Measurement Unit) and GPS to know the physical environment and react accordingly. Among them, perception sensors including cameras and LiDARs provide the obstacle and traffic sign information to AVs to avoid wrong decisions like collision and violating traffic rules, etc. Therefore, multiple prior works have been studying the security of these perception sensors.

Prior work has shown that AVs are vulnerable to attacks towards camera [10, 12, 33] or LiDAR sensors [5, 8, 28, 35]. Adversaries can change the texture of 2D image [33](e.g., stop sign) or add well-designed adversarial patches [12] to mislead the cameras. They can also inject laser [8] to spoof the LiDAR sensors.

To defend the aforementioned attacks, researchers designed transformation for perceived inputs from camera [19] and LiDAR [32] sensors individually. Multiple Sensor Fusion(MSF) algorithms are also proposed to integrate inputs from cameras and LiDARs to produce correct output based on the unattacked sensors.

However, an important and widely accepted assumption in MSF is there is at least one clean sensor [7].

This assumption is approved to fail at specific situations by Cao et al. [7]. They observed that different shapes of a 3D object can spoof both LiDARs and cameras with the change of point position and pixel valuesc [7]. Also, the generated 3D objects are stealthier and more robust than previous work. As no safe sensor exists under this attack, the MSF algorithm doesn't have reliable sensors to trust. Most recovering-based defenses only work in one kind of sensor attack and thus fail to have good performance under this scenario. Therefore, in the experiments, the car didn't detect the adversarial obstacles and crashed into it.

We observe that defense against attacks to both cameras and LiDARs is urgently needed to secure AVs. Even though some prior works [30, 32, 34] have studied 3D objects adversarial defenses. They are targeting pure neural networks instead of a real AV system. The object detection in a real AV system consisting of input pre-processing(e.g., format transformation, feature generation, etc.), neural network model, post-processing(e.g., clustering, multiple sensor fusion, etc.). Also, static and general 3D objects are considered in prior works rather than moving and traffic-related objects. Thus, their methods can't be directly applied to the attack we're targeting.

Our motivation comes from the previous defenses that remove perturbations from the corrupted inputs [11, 20], or take the majority vote from randomly transformed image [16] against2D images and 3D objects adversarial attacks. The main idea behind these defenses is to do transforms on the perception input and remove the malicious characteristics generated by the attacker.

Therefore, in this work, we aim to smooth the noisy surface of obstacles with this latest 3D point cloud reconstruction network - IF-Defense [30]. IF-Defense [30] aims to recover the surface of 3D objects with the awareness of geometry property and uniform distribution of the points. However, it's only trained and tested in the dataset containing single 3D objects. Its performance in working in real AV perception is unknown. Moreover, directly applying IF-Defense [30] may result in unnecessary computational overhead as not all the 3D perception needs recovery. For example, in a wide open area, the majority of the perception is road and only a small part is the obstacles.

Based on this, we further propose a lightweight segmentation algorithm, aiming to provide a rough location for the areas to be recovered. The intuition for this algorithm is, we observe that due to laser imaging in the LiDAR system, a blank shadow is formed after the obstacle. Usually, the number of points in the obstacle area is much larger than that in road areas. And the number of points in shadow areas is much smaller. We then calculate the Manhattan distance between

obstacle areas and shadow areas to relate the obstacle with the corresponding shadow. A new set of point clouds containing the obstacle and its shadow is treated as the object to be recovered and sent to IF-Defense [30]. Through this, we aim to recover the noisy surface of 3D objects in the point cloud form.

One advantage that our work has is, we are not aimed to recover the direct output for both LiDAR and camera object detection models. We only need to provide at least one or a few correct outputs and let the MSF fuse the two results from LiDAR and camera model to correct detection results. Even though we can't guarantee to recover the outputs, at least some benign inputs are sent to MSF and there are fewer possibilities in the wrong detection.

In this work, we aim to answer the following research questions:

1. Is it possible to apply transforms on noised 3D objects to recover the detection output of the AV system?

2. Will the transforms degrade the accuracy when it's applied to clean inputs?

3. Can the adversaries alter their attack accordingly to avoid this model-specific defense?

1.1 Threat Model

The attacker is assumed to know the details of the MSF algorithm in the victim system. Most adversarial attacks [5, 10, 12, 28, 33] on camera or LiDAR sensors in the AVs are white-box attacks and this assumption holds for many prior works. The adversaries are also able to profile the road environment they're targeting, and generate obstacles using 3D printing.

1.2 Challenges

C1: How to find useful transforms for both camera and LiDAR perception? Prior works have studied the transformations in 3D objects for camera or LiDAR perception [20, 22, 24]. However, due to the different perception theories in camera and LiDAR, it might be hard to find a common transform for both of them. And this is needed in the attacks towards both sensors.

C2: How to avoid the deterioration of performance on clean objects? Applying random transform on all inputs will decrease the accuracy rate on clean inputs [12], as it may remove some important properties in real-life objects. Therefore, the transformation that we're designing should try to recover the corrupted inputs while remain the important properties in clean inputs.

C3: How to avoid the attacker from altering the attacks with the knowledge of the transform? Transformation is usually effective to a certain series of characteristics in the inputs. If a white-box attacker knows the transforms applied in the system, he may alter the optimization objectives to prevent the transform from removing the adversarial parts. Is there a way to decrease the success rate that the attacker can design a similar attack?

2 Background

2.1 Point cloud

Point cloud refers to a set of data points in space to represent a 3D object produced by a 3D scanner. As 3D data can provide a better understanding of the shape and geometric information in the surrounding environment [17], AV systems are usually equipped with LiDAR sensors to generate the 3D point cloud. However, unlike 2D images, the 3D point cloud is highly unstructured and difficult to interpret. For example, traditional 2D image filtering techniques like mean filtering [25] and median filtering [18] can't be applied on the 3D point cloud. And previous 2D image neural networks are also not applicable. This makes the 3D point cloud smoothing and object reconstruction more difficult.

2.2 LiDAR perception in AV

ADD FIGURE! Figure 1 shows the perception module in common AV system [1]. 3D objects are first perceived by LiDAR and camera to generate point clouds(LiDAR) and frames of images(camera). These sensor data then go through a pre-processing unit to extract some aggregated features and ROI(Region of Interest). Pre-processed data will be fed into the LiDAR perception network and camera perception network respectively in the MSF algorithm unit. The MSF algorithm will fuse the outputs of two perception networks and give the detection output.

In this project, we plan to add point cloud recovery between the LiDAR rendering part and pre-processing part, given the fact that the MSF algorithm can produce correct results with at least one correct sensor output.

3 Approach

3.1 Overview

Adversarial 3D objects usually have noisy surfaces to mislead the detection networks. Thus, surface denoising is needed in the pre-processing unit. Directly applying the object reconstruction network like IF-Defense [30] will result in high computational overhead and low performance. Thus, we design a lightweight segmentation algorithm to first crop the potential areas containing obstacles based on laser imaging theory, and then apply IF-Defense [30] to recover the surface of the 3D object.

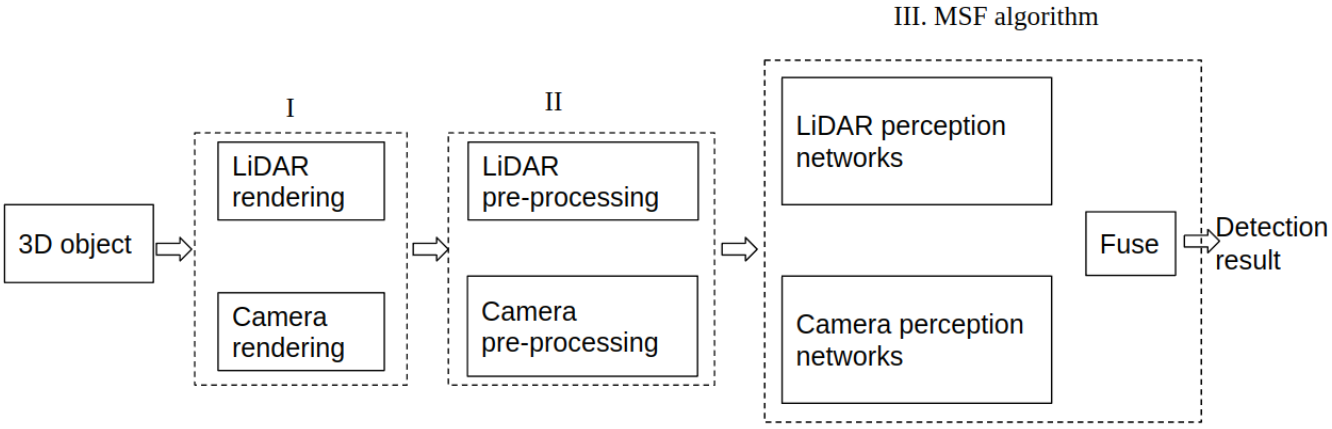
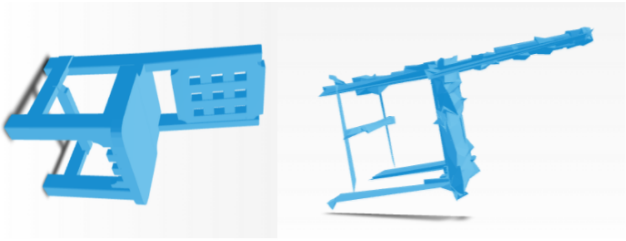


Figure 1: Perception module in AV systems

3.2 Characteristics of adversarial 3D objects

Adversarial 3D objects are generated by adding, removing, and modifying the 3D points. These perturbations, however, will always lead to a rough object surface, violating the geometrical features in Figure 2(b). When the distance between the vehicle and the adversarial obstacle is larger than the brake distance, it is easy for the vehicle to mistake this glitchy surface as noise and ignore the overall shape of the obstacle. In this case, it fails to detect the obstacle and crashes into it. Therefore, it's important to smooth the noise on the surface and let the out-of-bound points lie on the surface.



(a) Benign chair (b) Malicious chair generated by MSF-ADV[6]

Figure 2: Benign chair and the malicious version

3.3 Recovering methods

For 3D point clouds, there are broadly two types of methods to recover the distorted surface of 3D adversarial objects. One is traditional filtering based methods such as VG [29], L0 [27], MLS [6]. They perform well in removing overall noise in the 3D perception but fail to recover the broken surfaces. The other is deep neural network based object reconstruction methods such as DUP-Net [14] and IF-Defense [30]. They aim to recover the broken surface and local part removal attack.

However, they are only evaluated on a single 3D object instead of the outdoor scene perceived by autonomous cars. It is difficult to apply them directly in AV perception because 1) it will induce large unnecessary computation overhead as the whole road condition is fed as the input, and 2) the object reconstruction methods are mostly trained on single object datasets instead of real AV outdoor datasets.

Therefore, in this defense, we first design our own lightweight segmentation algorithm in real AV perception, and then apply the latest object reconstruction method - IF-Defense [30], to remove the noise on the surface.

In the following parts, we first introduce our lightweight segmentation algorithm and then introduce the segmentation-based IF-Defense [30].

3.4 Lightweight segmentation algorithm

Figure 3 is an example of projected LiDAR perception of a traffic cone in the middle of the road. These wave-like curves are the result of laser scanning in an open area. An obstacle will prevent the laser from scanning the area behind it and thus a blank area is formed behind the obstacle. We can therefore use these white areas to segment the potential areas containing the obstacle. After that, applying IF-Defense [30] only in these segmented areas will reduce the computation workload.

To segment these areas, we first project the 3D object point cloud with a front view like Figure 3 to get the image of 2D points. Then we divide the 2D projection into multiple cells illustrated in Figure 4(a). The cell is a square with the edge length a to help us calculate the points' distribution. We then calculate the total amount of points in the cell and store it, like in Figure 4(b). Algorithm 1 shows how we get the potential segmentation areas.

From the observation in Figure 3, obstacles usually have higher point density followed by a blank shadow-like area. Therefore, our goal is to find dense cells which are also accompanied by a series of sparse cells. First, in Line 7, we

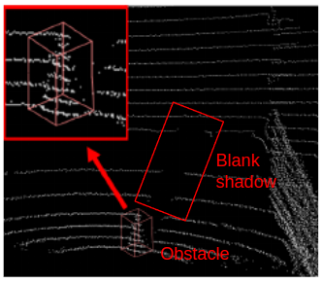


Figure 3: LiDAR perception. Modified from Figure 10 in [7]

sort the array of the number of points in each cell, choose the top $c\%$ dense cell numbers as our obstacle set A in Line 8, and choose the least $d\%$ dense cell as our blank area set B in Line 9. In Line 10, we select one cell i from A, calculate the Manhattan distance between the i and each cell in set B according to Eq. 1.

In Eq. 1, $A(i)_x$ represents the x coordinate of i_{th} cell in A, i.e., the obstacle set. $A(i)_y$ represents the y coordinate of i_{th} cell in A. $B(i)_x$ represents the x coordinate of i_{th} cell in B, i.e., the blank area set. $B(i)_y$ represents the y coordinate of i_{th} cell in B. And the Manhattan distance d_{ij} is the sum of the absolute differences of the coordinates of two points. We choose this metric because it's fast to calculate and also represents how far two points are from each other.

Then in Line 15, we sort the array of Manhattan distance D in ascending order. We further calculate the Manhattan distance among the top $k\%$ cells in D in Line 18, to measure the distances among blank area cells. In Line 19, we add all the Manhattan distances in D and compare it with a threshold. If the sum is lower than a threshold h, we think these cells are close to each other and might be shadows of the obstacle. Then we choose the highest and lowest x and y coordinates of the points and serve this as the segmentation boundary. Algorithm 1 ends when all the obstacles in set A are iterated. We note that parameters in the algorithm like c, d, k, and *threshold* have to be decided by experiment and profiling.

$$d_{ij} = |A(i)_x - B(j)_x| + |A(i)_y - B(j)_y| \tag{1}$$

3.5 Segmentation-based IF-defense

IF-defense [30] is a framework to recover the corrupted surface of the point cloud based on the implicit function [4]. It aims to optimize the shape of 3D objects to follow the geometry property and realize the uniform distribution of 3D points. It uses geometry-aware and distribution-aware loss functions to encourage the optimized points to lie on the surface as well as distribute more evenly. IF-defense [30] is implemented with ONet [23] and ConvONet [24] network. However, the dataset that IF-defense [30] uses is ShapeNet [9], which is a set of single clean 3D objects. It hasn't been tested on AV scenarios such as the KITTI [15] dataset.

Algorithm 1 Object segmentation algorithm

1: *num_Array*: array of the number of points in each cell
2: A: obstacle sets
3: B: blank area sets
4: D: Manhattan distance array
5: *sum_Dis*: Sum of difference in Manhattan distance
6: *bound_Dict*: Boundaries of each segmentation
7: Sort num_Array in descending order
8: $A \leftarrow$ Top c% elements in num_Array
9: $B \leftarrow$ Least d% elements in num_Array
10: **while** A is not fully checked **do**
11: **while** B is not fully checked **do**
12: $d_{ij} = |A(i)_x - B(j)_x| + |A(i)_y - B(j)_y|$
13: add d_{ij} into D
14: **end while**
15: Sort D in ascending order
16: $D \leftarrow$ Top $k\%$ D
17: **while** D is not fully checked **do**
18: $diff = |D(i)_x - D(i+1)_x| + |D(i)_y - D(i+1)_y|$
19: sum_Dis += $diff$
20: **end while**
21: **if** $sum\_Dis < threshold$ **then**
22: $bound\_Dict[i] \leftarrow$ smallest x coordinate in D
23: $bound\_Dict[i] \leftarrow$ largest x coordinate in D
24: $bound\_Dict[i] \leftarrow$ smallest y coordinate in D
25: $bound\_Dict[i] \leftarrow$ largest y coordinate in D
26: **end if**
27: **end while**
28: return *bound_Dict*

One big challenge of applying it on point clouds in AV outdoor perception is that there are multiple objects existing in the scene, like roads, buildings, vehicles, and pedestrians. Since we aim at removing the obstacle noises fed into the object detection network, recovering the whole point cloud including the road is a waste of resources. Also, it may not perform well in AV outdoor perception due to the different training dataset.

Thus, we plan to first use the lightweight segmentation algorithm to get the potential recovery areas. Then we apply the IF-defense [30] to recover the selected areas. At last, we replace the originally selected point clouds with the recovered point clouds and feed the combined output into the pre-processing unit in the AV perception module. In this way, we hope to recover the LiDAR detection output before sending it to the sensor fusion part.

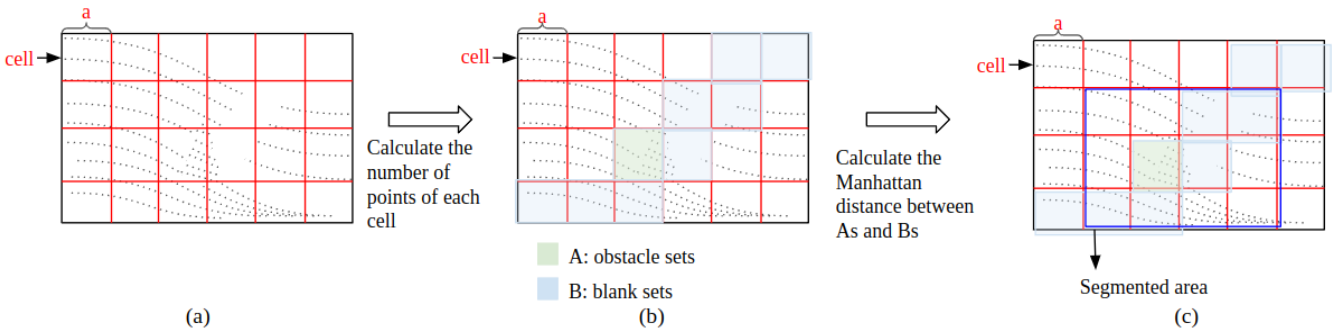


Figure 4: Segmentation algorithm

4 Experiments

4.1 Experiment setup

For this project, we'll use Baidu Apollo [1] open-source AD system. It's a widely-used industry-grade system equipped with the typical MSF algorithm. We'll also use LGSVL simulator [26], an open-source simulator providing a virtual environment to test AV systems. To reproduce the attacks on camera and LiDAR, we use the Github repository [7] provided by Cao *et al.*. After generating adversarial objects with the tool mentioned, we'll test our AV-based IF-defense [30] by processing the rendered 3D point cloud before feeding it into the pre-processing part in Figure

As for the evaluation metric, we will measure 1) the detection accuracy of the adversarial obstacles , 2) the detection accuracy and false-positive of the overall obstacles, including the benign ones and adversarial ones.

5 Related Work

Adversarial camera and LiDAR-based attacks AVs Attacks in perception sensors can be divided into two categories, camera-based attack, and object-based attack. The camera-based attack methods [10, 12, 33] propose to hide the objects to be detected by adding adversarial patches. The attacker can apply different interference methods to enhance the robustness so that the objects won't be detected from varying observation angles and distances. This camera-based attack aims to change the texture of the object [7]. The Lidar-based attack methods [5, 8, 28, 35] propose to spoof the LiDAR with injecting laser [8], finding vulnerable LiDAR detection locations [35] or changing the shape of the 3D objects [5]. This kind of attack can fool the LiDAR object detection mechanism, but it's hard to spoof cameras as it aims to change the shapes instead of the texture of the object [7]. In these works, to mislead the neural network, some outstanding patterns are generated to cause the model to have a tendency towards specific outputs.

Defense towards the adversarial camera and LiDAR-based attacks in AVs Defenses against these adversarial perception attacks also fall into two types. One kind of defense [30, 32, 34] aims to detect and recover the corrupted objects before they're sent to the detection algorithm. The authors reconstruct the objects with implicit functions [30] or denoising and upsampling [34]. Although these methods can achieve a good recovering rate, they focus on either camera-based attacks or object-based attack. The other kind of defense aims to fuse multiple sensors [13, 21, 22, 31] to avoid the spoofed sensor guiding the detection output. These Multiple Sensor Fusion (MSF) algorithms integrate the image and LiDAR feature map strategically to rely on the unattacked sensors.

References

[1] Baidu apollo. https://github.com/ApolloAuto/apollo. Accessed: 2022-03-05.

[2] Companies are racing to make self-driving cars. but why? https://www.washingtonpost.com/outlook/2022/02/04/self-driving-cars-why. Accessed: 2022-03-05.

[3] Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles.

[4] 1 implicit function theorems. 2019.

[5] Mazen Abdelfattah, Kaiwen Yuan, Z. Jane Wang, and Rabab Kreidieh Ward. Towards universal physical attacks on cascaded camera-lidar 3d object detection models. In *ICIP*, 2021.

[6] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Point set surfaces. *Proceedings of the Conference on Visualization*, 35-36:21–28, 2001.

[7] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security

of multi-sensor fusion based perception in autonomous driving under physical-world attacks. Proceedings - IEEE Symposium on Security and Privacy, May 2021.

[8] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Wonseok Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[9] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[10] Shang-Tse Chen and Jason Martin. Physical adversarial attack on object detectors (extended abstract). 2018.

[11] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of jpg compression on adversarial images. *ArXiv*, abs/1608.00853, 2016.

[12] Kevin Eykholt, I. Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, Tadayoshi Kohno, and Dawn Xiaodong Song. Physical adversarial examples for object detectors. *ArXiv*, abs/1807.07769, 2018.

[13] Davi Frossard and Raquel Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642, 2018.

[14] Hongxing Gao, Wei Tao, Dongchao Wen, Junjie Liu, Tse-Wei Chen, Kinya Osa, and Masami Kato. Dupnet: Towards very tiny quantized cnn with improved accuracy for face detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 168–177, 2019.

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231 – 1237, 2013.

[16] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. *ArXiv*, abs/1711.00117, 2018.

[17] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:4338–4364, 2021.

[18] Thomas S. Huang, G Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:13–18, 1979.

[19] Connie Khor Li Kou, Hwee Kuan Lee, Ee-Chien Chang, and Teck Khim Ng. Enhancing transformation-based defenses against adversarial attacks with a distribution classifier. In *ICLR*, 2020.

[20] Bin Liang, Hongcheng Li, Miaoqiang Su, Xirong Li, Wenchang Shi, and Xiaofeng Wang. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. *IEEE Transactions on Dependable and Secure Computing*, 18:72–85, 2021.

[21] Ming Liang, Binh Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7337–7345, 2019.

[22] Ming Liang, Binh Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018.

[23] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, 2019.

[24] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *ArXiv*, abs/2003.04618, 2020.

[25] S. Rakshit, A. Ghosh, and B. Uma Shankar. Fast mean filtering technique (fmft). *Pattern Recognit.*, 40:890–897, 2007.

[26] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Martins Mozeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, Eugene Agafonov, Tae Hyung Kim, Eric Sterner, Keunhae Ushiroda, Michael Reyes, Dmitry Zelenkovsky, and Seonman Kim. Lgsvl simulator: A high fidelity simulator for autonomous driving. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020.

[27] Yujing Sun, Scott Schaefer, and Wenping Wang. Denoising point sets via l0 minimization. *Comput. Aided Geom. Des.*, 35-36:2–15, 2015.

[28] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Binh Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples

for lidar object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13713–13722, 2020.

[29] Liying Wang, Yan Xu, and Yu Li. A voxel-based 3d building detection algorithm for airborne lidar point clouds. *Journal of the Indian Society of Remote Sensing*, 47:349–358, 2018.

[30] Ziyi Wu, Yueqi Duan, He Wang, Qingnan Fan, and Leonidas J. Guibas. If-defense: 3d adversarial point cloud defense via implicit function based restoration. *ArXiv*, abs/2010.05272, 2020.

[31] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018.

[32] Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *ArXiv*, abs/1902.10899, 2019.

[33] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[34] Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1961–1970, 2019.

[35] Yi Zhu, Chenglin Miao, T. Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. Can we use arbitrary objects to attack lidar perception in autonomous driving? *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.