Is the Synthesized Scene in Autonomous Driving Realistic?

Elaine Yao
University of British Columbia

Abstract

For Autonomous Vehicles(AVs), sensor perception is safety-critical. Failures in object detection can cause disasters. Despite various prior works on adversarial 3D physical attacks in AVs, all of them are simulating placing obstacles in the road with synthesized scene. However, the rendering functions to generate the synthesized scenes may not be able to ensure the physical consistency between the obstacle and the background. The real-world sensor perception is much more complicated. In this project, we present the study of authenticity issues of rendering-based synthesized scene in AV systems. We evaluate this by generating different synthesized scene and testing the performance of neural network on them. This allows us to quantify how realistic these scenes are and understand the impact of physical consistency in the scene on the effectiveness of generated adversarial objects.

We design a comprehensive test suite aiming at evaluating the whether the method to integrate an 3D object in the road background is realistic or not. We adopt empirical approaches that address four main design challenges: various impact factors, physical environment consistency, domain-specific metrics and automated pipeline. We evaluate the synthesized scene with our test suite in representative open-source industry-grade AD system object detection models with real-world driving scenarios. We also choose state-of-art adversarial 3D physical attack for evaluation in malicious cases. Our results show that most synthesized scenes are not realistic enough so that the object detection fail to detect the obstacles in it. Such phenomenon can reduce the effectiveness guarantee of generated 3D adversarial attacks in physical world.

1 Introduction

Autonomous Vehicles(AVs) can sense the surrounding environment and move safely with little human input. They are playing an important role in future transportation. Large companies such as Google, Uber [4] are racing to develop AVs and some high level , such as level 4 self-dirving cars have already been deployed on the road. Level 4 is considered to be fully autonomous driving. It can handle complex urban driving situations without driver intervention. A fundamental part in autonomous driving system is perception. It uses sensors [7] 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. Failures in perception can pose a threat to the safety of self-driving. In 2020, a tesla car in autopilot mode collided with an overturned truck as it failed to detect it. 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 [14, 17, 38] or LiDAR sensors [9, 12, 32, 40]. Adversaries can change the texture of 2D image [38](e.g., stop sign) or add well-designed adversarial patches [17] to mislead the cameras. They can also inject laser [12] to spoof the LiDAR sensors.

All of these studies, however, are limited to attacks with synthesized background, i.e., integrating the adversarial object with the road background through rendering functions instead of realistic simulation [11, 29, 32, 38, 40]. In order to simulate the scenario where a 3D vehicle is put in the road, these work will synthesize the attacked-influenced sensor perception, for example, the point clouds by LiDAR and images by camera with respective 3D rendering functions. These rendering techniques provided by computer graphics can simulate the real sensor functions but still lack comprehensive consideration in sensing the environment due to the simplicity of its model. By contrast, sensor perception in physical world can integrate more information such as light condition, realistic texture of adversarial objects, reasonable positions. Although most recent works in adversarial attacks will evaluate their methods in physical world to show the effectiveness and feasibility,

they use the synthesized method to generate the malicious objects for faster speed. The assumption that the synthesized attack-influenced background is realistic should be believed to hold in general [11] and thus examining the effectiveness of this method is an urgent call.

This project presents a study on evaluation of the reality of synthesized backgrounds in AD perception systems today. We test the above rendering-based simulation assumption by evaluating the neural network performance on these integrated sensor perception outputs used in the state-of-art adversarial attack work. This allows us to gain a solid understanding of how much authenticity guarantee the use of synthesized background can provide as a realistic simulation way to generate effective adversarial objects. Specifically, we consider physical 3D objects as the attack vectors for real-life feasibility and examine the performance of object detection neural network models deployed in real AV systems on the synthesized scenes.

Even though previous works have designed perception rendering functions for camera and LiDAR, we find that simply feeding them with different objects and backgrounds won't meet our requirements. First, we need to identify the factors in the synthesized scene that might influence the detection accuracy of the neural network. For example, the object detection model may find it difficult to detect an object which is far from away the sensor. Also, the color of the obstacle is similar to surrounding environment so that it's hidden from the neural network. Second, physical consistency between the obstacle and the driving background should be maximum guaranteed. No matter where we put the obstacle, it should stand on the road instead of floating in the air or hitting the ground. It should also follow the shadow caused by sunlight. Third, to quantify the authenticity of the synthesized scene, we need to come up with domain-specific metrics. The previous works use different metrics to measure whether their adversarial obstacle achieves the goal and lack unified standard, which makes it difficult to provide fair and reasonable metrics. Fourth, we need to develop an automated pipeline for generating different synthesized scene, evaluating the neural network performance in the scene without attack as well as with attack. Manually adjusting the parameters can take a long time and it's hard to do large scale analysis.

Towards this end, we design an automatic and comprehensive synthesized scene test suites, which addresses the challenges above and thus provides evaluation for the authenticity of these rendering methods. Through preliminary experiments, we choose different driving backgrounds, 3D obstacle properties(including the color, shape and texture) and the interaction between the background and the obstacle, e.g., the relative position, as the impact factors and serve them as the parameters to adjust. The attackers assumed in the previous work can just place an object on the road as simulated in the synthesized scene. To systematically generate realistic scene, we adopt camera imaging theory to adjust the height of the object so that it's standing on the road. Light condition is considered to comply with the driving background. Also, we start with normal obstacle which can be obtained from life easily, e.g., a common chair. Under these test settings, we address design challenge 3 by considering the correctness of bounding box of detected object, object class and its corresponding confidence score. We extract these by parsing the output of object detection neural network. Also, we use these as building blocks to compute the overall scores for authenticity of the scene. In the end, we developed automated pipelines for selecting different factors and evaluating the detection performance under benign and malicious cases.

We evaluate the scene synthesizing method in MSF-ADV [11] and choose the image object detection neural network model in Autoware.AI [2], which is representative for current AD systems. We also choose the attack in MSF-ADV [11] to generate adversarial 3D objects which can both fool the camera and LiDAR object detection models. We select 3 shapes of chairs from McGill 3D Shape Benchmark [5] and evaluate each on 5 real-world driving scenarios from the KITTI dataset [22]. 60 different scenes are synthesized and evaluated. Our results show that the benign obstacle in the synthesized scene fail to be detected in all the test settings. We also find that for the attack strategy generating the adversarial object, if the benign object fails to be detected in the first place, it's also hard to generate effective adversarial obstacles. What's more, in this situation, it's hard to provide guarantee that the generated adversarial object is effective in physical world.

In summary, this work makes the following contributions:

1. We study the authenticity of synthesized scenes in AD perception systems. We successfully design a comprehensive test suite aiming at evaluating the whether the method to integrate an 3D object in the road background is realistic or not.

2. We adopt empirical approaches that address four main design challenges: various impact factors, physical environment consistency, domain-specific metrics and automated pipeline.

3. We evaluate the synthesized scene with our test suite in representative open-source industry-grade AD system object detection models with real-world driving scenarios. We also choose state-of-art adversarial 3D physical attack for evaluation in malicious cases. Our results show that most synthesized scenes are not realistic enough so that the object detection fail to detect the obstacles in it. Such phenomenon can reduce the effectiveness guarantee of generated 3D adversarial attacks in physical world.

While rendering the obstacle into the road background is a general way of generating adversarial 3D obstacles, prior works lack the realistic validation of the synthesized scene. In

this project, we try to evaluate it by measuring the neural networks performance under different settings. We hope that our findings can inspire more future related research to validate their rendering process in AD perception when designing the 3D adversarial obstacles.

2 Background

2.1 AV Perception System

In the state-of-art AV systems, perception plays an important role in ensuring the safety. The sensors are used to detect the obstacles and measure the velocity or distance in real time. Typical systems in high-level, such as Level 4 [1] AVs adopt both LiDAR and camera for visual perception. LiDAR [15] can detect the ranges by shooting an object with a laser and measure the distance by getting the time for the light to be reflected back to the receiver. Compared with RADAR, LiDAR is much more accurate in resolution. Thus it's used to reconstruct exact 3D models of objects in autonomous systems. However, it's difficult to get the texture-related information such as the color [19]. On the other hand, camera images are good at providing shape and texture information, but lack depth and distance information due to its 2D imaging. In order to compensate the weaknesses and utilize the strength in each sensor, most AV systems will adopt Multi-Sensor Fusion(MSF) design, in which it will fuse the sensor reading from both LiDAR and camera.

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

In this project, we focus on the camera perception part. 3D objects are sensed by the camera in the form of 2D images. When a random obstacle in put in the middle of the road, a synthesized image with obstacle and road background is generated. This image will then be fed into the object detection neural network for results.

2.2 Synthesized Scene

In order to synthesize the obstacle with the road background, many prior works [11] use Neural 3D Mesh Renderer(NMR) for camera rendering [23]. NMR provides a way to generate a 2D image from the 3D world. It can transfer rendering gradient with consideration of texture, lighting, camera and the object shapes. Given the camera pose, light condition, relative position between the camera and the overall background, NMR will provide the image output of this background with certain camera setting.

Fig. 2 overviews the image synthesizing process in MSF-ADV [11]. It first chooses the background from the target road, and the obstacle that it intends to put on the road, for example, a brown chair. The 3D chair is presented in the form of point cloud, which is a set of data points in space. Camera parameters and light condition will be set to rendering this chair to a 2D image. Then the relative location of the chair in the background is set. Original pixels in the background will be masked by the pixels in the 2D chair image to simulate how chair is put in the middle of the road. Before feeding the synthesized scene into detection neural networks, some pre-processing steps such as data transformation, utilizing the Region of Interest(RoI) filter to clear unrelated input parts and collecting aggregated features are performed. These pre-processing processes can reduce the input size fed into the neural network and greatly improve the inference speed.

In this project, we will use this as the target pipeline for generating synthesized scene and will evaluate the authenticity of the synthesized output.

2.3 Adversarial 3D Object Attacks

Prior works find that it's easy to fool models with deceptive data, which causes the malfunction in the neural network model. This kind of adversarial attacks are also studied in the context of physical world [16,18,37]. In the AV systems, some prior works proposed physical adversarial attacks by placing obstacles in the air or on the road [16, 18, 37]. Some are targeting at fooling the camera-based perception neural network [16, 18, 37] while others are focusing on LiDAR-based perception neural network [12,30]. Recently, considering that MSF design is widely used in AD systems, MSF-ADV [11] is proposed to attack both LiDAR and camera sensors with a 3D adversarial object.

MSF-ADV [11] treats the process of generating the adversarial attacks as an optimization problem and Fig. 3 provides an overview of the process to generate adversarial obstacles. It first picks a normal 3D object and apply 3D transformations including rotation, position shifting to get different angles of the object. This is to improve the robustness of the obstacle in various kinds of environment. Then it will generate point cloud and image of this 3D object through ray-tracing [6] and NMR [23] to simulate the perception output of LiDAR and camera sensors. These two sensor outputs will be integrated with the road background and pre-processed before being sent to perspective perception neural networks and MSF algorithm. The attacker is assumed to be able to perturb the shape and position of the 3D object. Also, adversarial loss function is designed to cause the malicious object not to be detected by MSF algorithm as well as keep the obstacle stealthy. After obtaining the generated malicious object, the attacker can just 3D print it and place it in the road according to the parameters

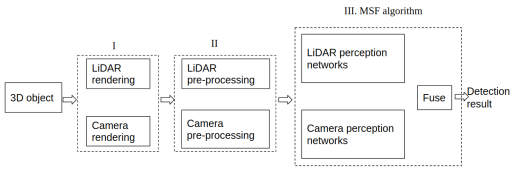


Figure 1: Perception module in AV systems

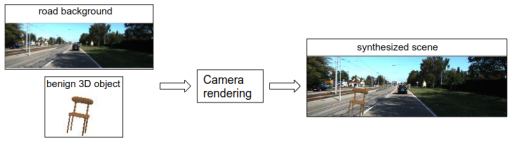


Figure 2: Synthesized scene through camera rendering

in the optimization process.

In this project, we will use MSF-ADV [11] as the target adversarial attack in the evaluation part. MSF-ADV [11] uses the same method in previous section to generate synthesized scene. Therefore, we want to see whether the authenticity of the synthesized scene will influence the effectiveness of the generated malicious object.

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 [34] 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 [34] to recover the surface of the 3D object.

3.2 Characteristics of adversarial 3D objects

Adversarial 3D objects are generated by adding, removing, and modifying the 3D points. These perturbations, however,

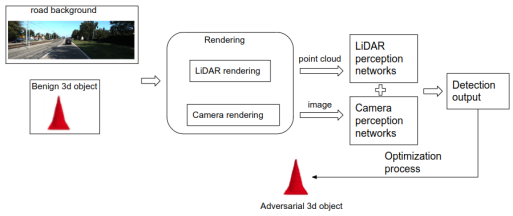
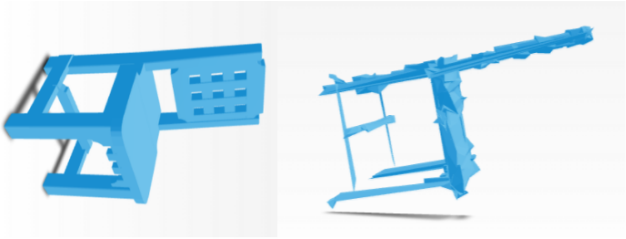


Figure 3: Pipeline for generating adversarial 3D object attacks for both Lidar and Camera



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

Figure 4: Benign chair and the malicious version

will always lead to a rough object surface, violating the geometrical features in Figure 4(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.

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 [33], L0 [31], MLS [10]. 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 [21] and IF-Defense [34]. 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 [34], 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 [34].

3.4 Lightweight segmentation algorithm

Figure 5 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 [34] 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 5 to get the image of 2D points. Then we divide the 2D projection into multiple cells illustrated in Figure 6(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 6(b). Algorithm 1 shows how we get the potential segmentation areas.

From the observation in Figure 5, 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 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 [34] is a framework to recover the corrupted surface of the point cloud based on the implicit function [8]. 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 [34] is implemented with ONet [26] and ConvONet [27] network. However, the dataset that IF-defense [34] uses is ShapeNet [13], which is a set of single clean 3D objects. It hasn't been tested on AV scenarios such as the KITTI [22] dataset.

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 [34] 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.

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*

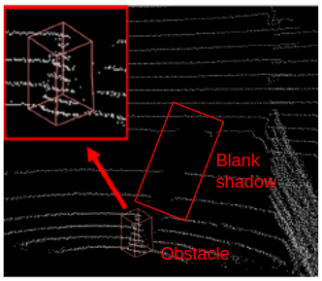


Figure 5: LiDAR perception. Modified from Figure 10 in [11]

4 Experiments

4.1 Experiment setup

For this project, we'll use Baidu Apollo [3] open-source AD system. It's a widely-used industry-grade system equipped with the typical MSF algorithm. We'll also use LGSVL simulator [28], 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 [11] provided by Cao *et al.*. After generating adversarial objects with the tool mentioned, we'll test our AV-based IF-defense [34] 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 Discussions

5.1 Limitations of our Experiments

5.2 Future work

5.3 Challenges

6 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 [14, 17, 38] 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 [11]. The Lidar-based attack methods [9, 12, 32, 40] propose to spoof the LiDAR with injecting laser [12], finding vulnerable LiDAR detection locations [40] or changing the shape of the 3D objects [9]. 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 [11]. 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 [34, 36, 39] aims to detect and recover the corrupted objects before they're sent to the detection algorithm. The authors reconstruct the objects with implicit functions [34] or denoising and upsampling [39]. 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 [20, 24, 25, 35] 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] The 6 levels of vehicle autonomy explained. https://www.synopsys.com/automotive/autonomous-driving-levels.html. Accessed: 2022-03-05.

[2] Autoware.ai. https://www.autoware.org/. Accessed: 2022-03-05.

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

[4] 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.

[5] Mcgill 3d shape benchmark. http://www.cim.mcgill.ca/~shape/benchMark/. Accessed: 2022-03-05.

[6] Ray tracing. https://computationalthinking.mit.edu/Fall20/lecture14/. Accessed: 2022-03-05.

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

[8] 1 implicit function theorems. 2019.

[9] 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.

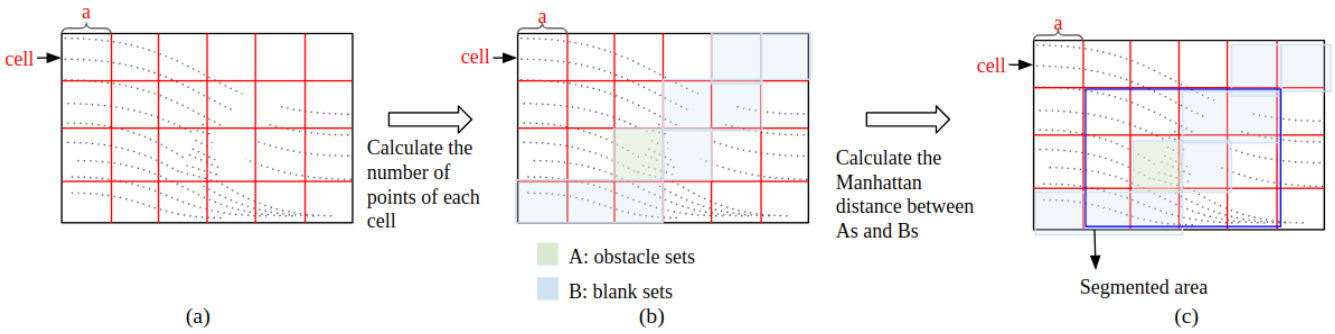


Figure 6: Segmentation algorithm

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] Ronald T. H. Collis. Lidar. *Encyclopedic Dictionary of Archaeology*, 2021.

[16] Object Detectors and In The. Camou: Learning a vehicle camouflage. 2018.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[23] Hiroharu Kato, Y. Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[24] 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.

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

[26] 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.

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

[28] 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.

[29] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. *ArXiv*, abs/2006.16974, 2020.

[30] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. *ArXiv*, abs/2006.16974, 2020.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. Seeing isn't believing: Practical adversarial attack against object detectors. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[38] 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.

[39] 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.

[40] 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.