CONTEXT-AWARE DATA AUGMENTATION FOR LIDAR 3D OBJECT DETECTION

Xuzhong Hu1∗, Zaipeng Duan1, Xiao Huang2, Ziwen Xu2, Delie Ming1 and Jie Ma†

1School of Huazhong University of Science and Technology, Wuhan, 430074, China
2China Ship Development and Design Center Wuhan, 430064, China

ABSTRACT

For 3D LIDAR object detection, data augmentation is an important module to make full use of precious annotated data. As a widely used data augmentation method, GT-aug effectively improves detection performance by inserting sampled groundtruths into LIDAR frames. However, they are often placed in unreasonable areas, leading to the loss of the semantic information between targets and backgrounds during training. To address this problem, we propose a context-aware data augmentation method (CA-aug), which ensures the proper placement of inserted objects by a simple strategy and produces realistic augmented scenes. CA-aug is lightweight and compatible with other augmentation methods. Experiments conducted on KITTI benchmark show that compared with the GT-aug and the similar method in LIDAR-aug (SOTA), it brings higher accuracy to the existing models especially for the detection of cyclists and pedestrians. We also present an in-depth study of augmentation strategies for the range-view-based (RV-based) models and demonstrate that CA-aug can fully exploit the potential of RV-based networks, boosting the moderate mAP of our test model by 8%.

Index Terms— 3D object detection, LIDAR, data augmentation

1. INTRODUCTION

In the autonomous driving system, 3D object detection is crucial for object tracking, path planning and other functions. Due to the ability to directly obtain accurate spatial geometric information and the continuous reduction of costs, vehicle-mounted LIDAR plays an important role in the scene perception of autonomous driving. Therefore, researches on LIDAR-based 3D detection are gradually emerging and flourishing. Thanks to relative datasets such as KITTI and Waymo, 3D detection models have been developed rapidly with their accuracy and reliability continuously increasing. However, compared with images, labeling 3D training data is much more costly and time-consuming, due to the sparsity of point clouds and more degrees-of-freedom of bounding boxes.

One solution to this problem is to generate more training samples by data simulators such as Carla[4] and Airsim[20]. To deal with the large sim-to-real domain gap, some researchers combine scan data from real-world scenarios with synthetic objects[5, 17]. Recently, Generative Adversarial Networks (GANs) have been employed to generate additional synthetic 3D medical images, thereby enlarging the training dataset[8, 9].

Unlike data simulation, data augmentation does not require much computational cost to produce synthetic samples in advance. It can be integrated into the training process and effectively avoid overfitting. Similar with 2D object detection, basic augmentation methods[7], such as global rotation, mirroring, and translation, are widely used in point clouds. To increase the flexibility of data augmentation, some researchers[2, 14] utilize optimistic strategies to automate relative hyperparameters. Adversarial learning can also be employed to adjust the difficulty of training samples to improve model robustness[16, 13]. To SECOND[22] introduces ground truth augmentation (GT-aug). Firstly, a database is created for the information of all groundtruths. During training, some objects are randomly selected from it and inserted into point cloud scenes. Due to the ability to rectify the data imbalance between positive and negative examples and significantly increase the diversity of training data, GT-aug shows its remarkable effectiveness in various models. Recently, a series of new methods[10, 3, 23, 24] have been proposed. They improve the generalization of networks by changing the distribution pattern of object points, which is equivalent to expanding the groundtruth database.

However, in GT-aug, the sampled objects are added into the current frame with their original positions, which means they are likely to appear in unreasonable areas, such as in a wall or behind a building. It leads to the non-negligible domain gap between augmented scenes and real ones. During training, the semantic information between foregrounds and backgrounds may be ignored since the location distribution of inserted objects have nothing to do with the current context. Worse still, compared with vehicles, the surface points of pedestrians and cyclists are sparser and more irregular in shape. When they appear in improper areas, it is difficult for a detector to
distinguish them from noises. This may mislead networks to learn error information.

In recent years, detection methods[1, 18, 6, 15] based on range view (RV) have attracted the attention of many researchers for easy deployment and fast inference speed. However, RV-based models require maintaining the 2.5D structure of the LIDAR point cloud. As shown in Figure 1(c) when projecting the point cloud augmented by GT-aug into a range image, the occluded points need to be discarded, which may destroy the structure of inserted objects and cause noises during training. For RV-based models, this issue hinders further improvements on the detection accuracy.

To solve the above problems, we propose a context-aware 3D data augmentation CA-aug. As shown in Figure 1(b)(d), a simple yet efficient module is configured to choose proper poses of groundtruths while avoiding the excessive occlusion by background points. Note that we are not the first to attempt to do so. LIDAR-aug[5] has proposed the similar idea by constructing "Validmap". However, we further consider LIDAR characteristics and distribution pattern of object points. Experiments demonstrate that our method is superior to GT-aug and LIDAR-aug in detection accuracy improvement. We also apply CA-aug to a RV-based model, which previous researches on data augmentation always ignore and validate its ability to significantly improve the detection performance.

In CA-aug, we simply place sampled objects where laser beams can reach. Specifically, we divide the current training frame into ground points \( P_g \) and obstacle points \( P_o \), which can be obtained by a simple BEV-based segmentation algorithm with no need for a pretrained network. As shown in Figure 2(a), they are further projected into a range image. In each column on it, we call the area from the last row to the nearest obstacle point (red) "Validspace", where sampled target points can appear. Since the ground points (blue) above the obstacle points in the image are ignored, the "Validspace" is continuous and can be easily described by a vector \( V \). As shown in Figure 3(a), it also covers empty zones between scan lines or in the distance compared to "Validmap"[5].

As shown in 2(c), to preserve the data structure of LIDAR, we fix the distance and the observation angle \( \alpha \) of the target and rotate it by the vertical z-axis of the origin to the selected position. Additionally, we allow a small proportion of object points to appear outside the "Validspace", which means the sampled objects can be placed behind some thin obstacles such as poles. As shown in 2(b), a vector termed "Rangebin" \( V_r \) is designed to describe the point distribution of the sampled target. With the two vectors and a collision avoidance module, rational positions can be quickly determined.

2. METHOD

2.1. Overview

Our proposed framework is illustrated in Figure 2. Firstly, the points and annotated information of various targets is collected and stored in the GT database. During training, targets are randomly sampled from it and inserted into the scene point cloud. Intuitively, in driving scene, these samples such as vehicles should be placed on the road. However, in 3D object detection datasets, road points are not labeled. In [19], a 3D segmentation model is applied to estimate road and sidewalk areas. However, it requires extra semantic segmentation datasets for model training, which is a waste of time and computing resources.

In CA-aug, we simply place sampled objects where laser beams can reach.
Fig. 3. (a) The red grids represent the "Validmap" of the scene. (b) The green rays represent the "Validspace" of the scene.

Fig. 4. Different types of augmentation for RV-based models. The gray points are deleted and the purple points are retained. (a) Directly project all points into the range image; (b) Discard the target if it loses too many points during projection; (c) In the range image, the background points at the same pixels as the target points are deleted; (d) Move the target to another area to avoid excessive occlusion.

Where \( f = f_{\text{up}} - f_{\text{down}} \) is the vertical field of view, \( r = \sqrt{x^2 + y^2 + z^2} \) is the range of point.

Both "Validspace" \( V \) and "Rangebin" \( V_r \) can also be computed from the range view representation:

\[
V[j] = \begin{cases} 
\min(r_p), & p \in P^o_j \\
\inf, & P^o_j = \phi
\end{cases} \quad (3)
\]

\[
V_r[j] = \text{num}(P^o_j) \quad (4)
\]

Where \( P^o_j \) and \( P^o_j \) are the obstacle points and the objects points in the \( j \)th column.

2.3. Location Check

Noted that both "Validspace" and "Rangebin" can be calculated offline. During training, they are selected for Location Check.

Suppose that the start column of the sampled target is \( j \) in the range image. The ratio \( r_j \) of object points in "Validspace" can be estimated:

\[
M_j = (V[j : j + l_g] < r_{\text{box}} + l_{\text{box}}/2) \quad (5)
\]

\[
r_j = M_j \cdot V_r/n_g \quad (6)
\]

Where \( l_g \) is the length of \( V_r \), \( n_g \) is the number of object points and \( l_{\text{box}} \) and \( r_{\text{box}} \) are the length and the distance of the 3D box. The vector \( M_j \) consists of 0 and 1, which indicates whether a column of object points are in "Validspace". When \( r_j \) exceeds the threshold \( \alpha \), the angle corresponding to column \( j \) is considered valid. We can easily get all these columns \( C \) via an operation similar with correlation:

\[
M = [M_0; M_1; \ldots; M_{W-l_g}] \quad (7)
\]

\[
r = M \cdot V_r/n_g \quad (8)
\]

\[
C = \text{where}(r > \alpha) \quad (9)
\]

2.4. Collision Avoidance

The detailed process of Location Check is described in Algorithm 1. Noted that directly adding objects into the current frame with valid angles may lead to a collision problem, which means that the 3D bounding boxes of the samples may intersect. Therefore, a collision avoidance algorithm needs to be introduced. Suppose that there remain \( k \) BEV boxes \( b_1, \ldots, b_k \) in the scene, when adding the next box \( b_{k+1} \), we determine if a collision occurs according to whether its four sides intersect with any side of boxes \( b_1, \ldots, b_k \). It's faster than calculating the IoU among all boxes[5].

Algorithm 1: Location Check

3. EXPERIMENT

3.1. Experiment Setup

Dataset We trained and evaluated all models on KITTI, one of the most popular datasets of 3D object detection. It contains 7481 training frames and 7518 testing frames. By convention, the training frames are further divided into the training split (3712 samples) and the validation split (3769 samples). We test our method on val split. There are three categories of targets: cars, pedestrians, and cyclists. For each, three difficulty levels (easy, moderate, hard) are involved. We apply KITTI’s evaluation benchmark - 3D average precision(AP) calculated with 40 recall positions to evaluate the detection results.

Models To demonstrate the universality of CA-aug, experiments are conducted on four models[12, 15, 21, 22]. For the RV-based model, we remove the RCNN module of RangeRCNN[15] and name this one-stage detector Rangetest. All the models are implemented with default parameters from their original papers.
 augmentation Besides GT-aug, we compare our CA-aug with LIDAR-aug. Original hyparameters in [5] are applied and no CAD models are introduced to extend the GT database. [11] presents three strategies to maintain the 2.5D structure of augmented point clouds. As visualized in 4, we apply them in the RV-based detector Rangetest. For “Culling”, we remove objects that remain less than 4 or lose 75% surface points. For CA-aug, the pixel length of the BEV map $\delta$ and the height threshold $\sigma$ are set to 0.25m and 0.4m. We set $\alpha = 0.8$ to allow sampled objects appearing behind thin obstacles. When training Rangetest, the “Culling” strategy is utilized to filter out sparsely objects.

3.2. Generalization

As shown in Table 1, our method achieves higher moderate mAP than GT-aug for all models. For the detection of cyclists and pedestrians, it outperforms GT-aug by a large margin mainly because it reduces noises during training. However, CA-aug doesn’t provide SECOND and PointPillar with better performance of car detection, since it makes them pay more attention to the other two classes. For the heavy model PV-RCNN, CA-aug brings an overall AP improvement, which indicates its ability to reduce overfitting. Additionally, our method overcomes the limitation of the data augmentation for RV-based detectors and greatly improves the detection performance of Rangetest.

3.3. Comparison

In Table 2 and Table 3, we report the results of comparative experiments on 3D car detection. As can be observed, CA-aug achieves the best performance.

Although ensuring placement rationality, Lidar-aug doesn’t fix the distances and observation angles of sampled targets, which breaks geometric characteristics of scanning lines. Putting dense targets in the distance or sparse ones in near places could mislead the network into learning the information that the density of foreground points is independent of their ranges. However, “Validmap” method is suitable for the placement of CAD models, because appropriate point patterns can be generated on the surface according to their ranges. CA-aug takes advantage of the radial symmetry of LIDAR points and moves objects through rotation. So the above issue doesn’t occur.

Compared to “Naive”, both “Culling” and “Drilling” alleviate the loss of surface points, enabling the network to learn more information from relatively complete geometric structures. By moving objects to reasonable places, CA-aug not only controls the degree of occlusion, but also guides the model to explore more semantic information between targets and backgrounds.

4. CONCLUSION

We have represented CA-aug to solve the problem of the irrational object placement in GT-aug. It creates more realistic augmented 3D scenes to help detection models learn the semantic relationships between targets and environments. Our method is plug-and-play and compatible with other augmentations. The experiments demonstrate its effectiveness and generalization. We believe that CA-aug is also applicable to other tasks in autonomous driving, such as 3D object tracking and 3D semantic segmentation.

Acknowledgement This work was supported by National Natural Science Foundation of China. U1913602

**Table 1. The performance of GT-aug[22] and CA-aug on the KITTI validation set. When training Rangetest, the “Culling” strategy is utilized.**

| Model     | Method         | Car (IoU = 0.7) | Pedestrian (IoU = 0.5) | Cyclist (IoU = 0.5) | mAP   |
|-----------|----------------|-----------------|------------------------|---------------------|-------|
|           | Easy           | Moderate        | Hard                   | Easy                | Moderate | Hard |       |
| PointPillar| GT-aug         | 87.76           | 78.26                  | 75.34               | 55.49  | 50.27 | 45.63 |
|           | CA-aug         | 87.61(-0.15)    | 78.26(+0.00)          | 75.45(+0.09)        | 58.32(2.83) | 52.49(+2.22) | 48.24(+2.61) |
|           |                |                 |                        |                     | 81.81(+2.60) | 82.46(+1.73) | 58.89(+1.54) |
| SECOND    | GT-aug         | 90.87           | 84.95                  | 78.91               | 56.88  | 52.96 | 48.22 |
|           | CA-aug         | 90.55(-0.22)    | 81.47(-0.48)          | 78.60(-0.31)        | 58.91(+2.03) | 53.93(+0.97) | 49.29(+1.07) |
|           |                |                 |                        |                     | 82.40(+4.09) | 64.09 | 59.69 |
|           |                |                 |                        |                     | 66.34 |       |       |
| PVR-CNN   | GT-aug         | 91.76           | 84.62                  | 82.45               | 63.63  | 56.81 | 51.14 |
|           | CA-aug         | 92.25(+0.49)    | 84.93(+0.31)          | 82.64(+0.19)        | 66.00(+2.37) | 59.77(+2.96) | 55.41(+3.28) |
|           |                |                 |                        |                     | 88.78 | 70.84 | 66.58 |
|           |                |                 |                        |                     | 70.75 |       |       |
| Rangetest | GT-aug         | 86.66           | 77.44                  | 72.74               | 47.76  | 39.55 | 35.11 |
|           | CA-aug         | 89.89(+1.33)    | 80.52(+3.08)          | 75.64(+3.10)        | 52.91(+5.15) | 42.22(+2.67) | 35.63(+0.52) |
|           |                |                 |                        |                     | 77.49(+6.24) | 58.66(+7.83) | 54.09(+7.65) |
|           |                |                 |                        |                     | 60.46(+4.52) |       |       |

**Table 2. The 3D AP performance of PointPillar is reported. The “baseline” means that no objects are inserted.**

| Method     | Car (IoU = 0.7) | Easy | Moderate | Hard |
|------------|-----------------|------|----------|------|
| Baseline   | 88.08           | 74.85| 70.55    |      |
| GT-aug[22]| 87.80           | 78.36| 75.41    |      |
| Lidar-aug[3]| 87.75       | 78.24| 75.35    |      |
| 3D-VField[13]| 87.05     | 77.13| 75.55    |      |
| CA-aug(ours)| 88.82       | 78.66| 75.75    |      |

**Table 3. Comparison of CA-aug and the other three strategies[11].**

| Method     | Car (IoU = 0.7) | Easy | Moderate | Hard |
|------------|-----------------|------|----------|------|
| Baseline   | 85.14           | 73.45| 68.64    |      |
| Naive      | 88.47           | 77.09| 72.92    |      |
| Culling    | 89.15           | 78.28| 73.32    |      |
| Drilling   | 88.84           | 78.17| 73.27    |      |
| CA-aug(ours)| 90.33       | 81.04| 76.06    |      |

**Table 4. The ablation study on CA-aug using Rangetest model.**

| Baseline Collision Avoidance Object Rotation Culling | Car (IoU = 0.7) |
|-----------------------------------------------------|-----------------|
| ✓ ◯ ◯ ◯                                             | 88.29           |
| ✓ ◯ - ◯                                            | 89.79           |
| ✓ ◯ - ◯                                            | 89.44           |
| ✓ ◯ - ◯                                            | 90.33           |
| ✓ ◯ - ◯                                            | 81.04           |
| ✓ ◯ - ◯                                            | 76.06           |
5. REFERENCES

[1] Yuning Chai, Pei Sun, Jiquan Ngiam, Weiuye Wang, Benjamin Caine, Vijay Vasudevan, Xiao Zhang, and Drago Anguelov. To the point: Efficient 3D object detection in the range image with graph convolution kernels. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15995–16004, 2021.

[2] Shuyang Cheng, Zhaoci Leng, Ekin Dogus Cubuk, Barret Zoph, Chunyan Bai, Jiquan Ngiam, Yang Song, Benjamin Caine, Vijay Vasudevan, Congcong Li, Quoc V. Le, Jonathon Shlens, and Dragomir Anguelov. Improving 3D object detection through progressive population based augmentation. In ECCV, 2020.

[3] Jaeseok Choi, Yeji Song, and Nojun Kwak. Part-Aware Data Augmentation for 3D Object Detection in Point Cloud. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3391–3397, 2021.

[4] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. ArXiv, abs/1711.03938, 2017.

[5] Jin Fang, Xinxin Zuo, Dingfu Zhou, Sheng Jin, Sen Wang, and Liangjun Zhang. LiDAR-Aug: A General Rendering-based Augmentation Framework for 3D Object Detection. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4708–4718, 2021.

[6] Jiaqi Gu, Zhiyu Xiang, Pan Zhao, Tingming Bai, Lingxuan Wang, and Zhiyuan Zhang. CVFNet: Real-time 3D Object Detection by Learning Cross View Features. ArXiv, abs/2203.06585, 2022.

[7] Martin Hahner, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Quantifying Data Augmentation for LiDAR based 3D Object Detection. ArXiv, abs/2004.01643, 2020.

[8] Maryam Hammami, Denis Friboulet, and Razmig Kéchichian. Data augmentation for multi-organ detection in medical images. In 2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA), pages 1–6. IEEE, 2020.

[9] Changhee Han, Yoshiro Kitamura, Akira Kudo, Akimichi Ichinose, Leonardo Rundo, Yujiro Furukawa, Kazuki Umemoto, Yuanzhong Li, and Hideki Nakayama. Synthesizing diverse lung nodules wherever massively: 3d multi-conditional gan-based ct image augmentation for object detection. In 2019 International Conference on 3D Vision (3DV), pages 729–737. IEEE, 2019.

[10] Jordan S. K. Hu and Steven L. Waslander. Pattern-Aware Data Augmentation for LiDAR 3D Object Detection. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pages 2703–2710, 2021.

[11] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What You See is What You Get: Exploiting Visibility for 3D Object Detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10998–11006, 2020.

[12] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12689–12697, 2019.

[13] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 3d-vfield: Adversarial augmentation of point clouds for domain generalization in 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17295–17304, 2022.

[14] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. PointAugment: An Auto-Augmentation Framework for Point Cloud Classification. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6377–6386, 2020.

[15] Zhidong Liang, Ming Zhang, Zehan Zhang, Xian Zhao, and Shiliang Pu. RangeRCNN: Towards Fast and Accurate 3D Object Detection with Range Image Representation, March 2021.

[16] Wenxin Ma, Jian Chen, Qing Du, and Wei Jia. PointDrop: Improving Object Detection from Sparse Point Clouds via Adversarial Data Augmentation. 2020 25th International Conference on Pattern Recognition (ICPR), pages 10004–10009, 2021.

[17] Sivalaban Manivasagam, Shenlong Wang, K. Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Binh Yang, Wei-Chiu Ma, and Raquel Urtasun. LiDARsim: Realistic LiDAR Simulation by Leveraging the Real World. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11164–11173, 2020.

[18] Gregory P. Meyer, Ankita Gajanani Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12669–12678, 2019.

[19] Petr Sebek, \vs Simon Pokorný, Patrik Vacek, and Tomá\vs Svoboda. Real3D-Aug: Point Cloud Augmentation by Placing Real Objects with Occlusion Handling for 3D Detection and Segmentation. ArXiv, abs/2206.07634, 2022.

[20] S. Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In FSR, 2017.

[21] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10526–10535, 2020.

[22] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedding 3D Object Detectors. Sensors, 18(10):3337, October 2018.

[23] Yifan Zhang, Qijian Zhang, Zhiyu Zhu, Junhui Hou, and Yixuan Yuan. GLENet: Boosting 3D Object Detectors with Generative Label Uncertainty Estimation. ArXiv, abs/2207.02466, 2022.

[24] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14489–14498, 2021.