Weakly Supervised 3D Object Detection from Lidar Point Cloud
Supplementary Material

Qinghao Meng[1], ✉Wenguan Wang[2], Tianfei Zhou[3],
Jianbing Shen[3,1], Luc Van Gool[2], and Dengxin Dai[2]

[1]School of Computer Science, Beijing Institute of Technology
[2]ETH Zurich [3]Inception Institute of Artificial Intelligence
https://github.com/hlesmqh/WS3D

In this document, we first give more implementation details of applying our model for 3D pedestrian detection (see §1). Later, in §2, we give some visual results for 3D pedestrian detection on KITTI [1] `val` set. Then, in §3, we compare the annotation results by applying our trained model as annotation tools, working in two annotation modes, *i.e.*, automatic and active. Finally, we discuss some representative failure cases in §4.

1 Weakly Supervised 3D Pedestrian Detection

We specify some modifications for adapting our method for **Pedestrian** class.
Data Preparation. For the KITTI training set which contains a total of $3,712$ scenes, there are only 951 scenes contain pedestrian labels. Considering the small amount of training samples, we use the weakly annotated BEV maps of the 951 scenes to train our Stage-1 model. We randomly choose 515, nearly 25% in $2,257$ samples in those scenes as the training data for our Stage-2 model. Compared with prior fully-supervised algorithms which leverage all the exhaustively annotated 951 scenes with $2,257$ pedestrian samples, we use far less and weak supervision. To reduce futile false negative responses and speeding up the CA-NMS process for better effectiveness, following [2], we set the x, z range of the searching region for pedestrian as $[(-20, 20), (0, 48)]$, respectively.
Pseudo Foreground Groundtruth Generation. For **Car** class, we use an *ellipsoid-shaped* 3D Gaussian distribution for pseudo *soft* foreground groundtruth generation (see Eq. 1). For **Pedestrian** class, we instead directly use a *pillar* (cylinder) to generate pseudo *binary* masks. This is because, compared with vehicles which are typically presented as elongated rectangles on BEV maps, the shapes of human on the BEV maps are more like regular squares. The radius of the pillars are uniformly set as 0.4 m.
Cylindrical 3D Proposal Generation. Considering the small size of pedestrians, we generate the cylindrical proposal with a 1 m radius over (x, z)-plane (4 m radius for vehicle). For each groundtruth, the proposals whose center-distances to it are less than 0.5 m are selected as its training samples.

✉ Corresponding author: *Wenguan Wang* (wenguanwang.ai@gmail.com).

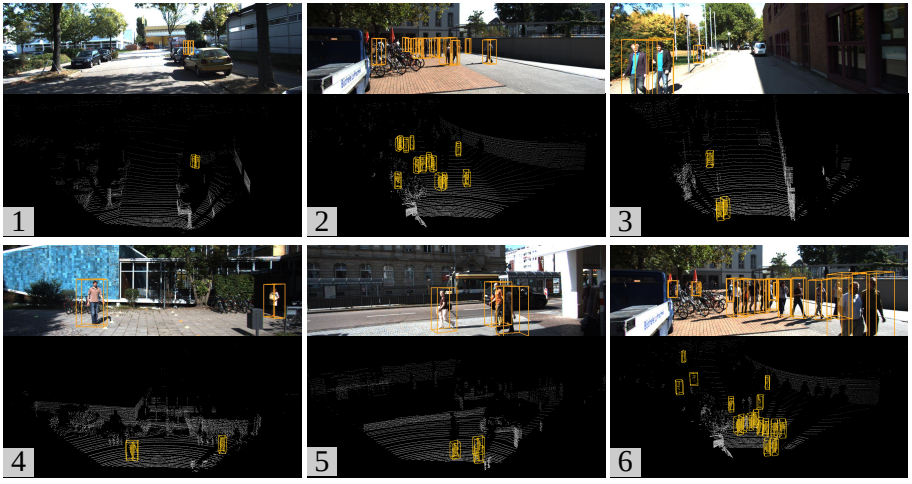

Fig. 1. Qualitative results of 3D object detection (Pedestrian) on KITTI `val` set. Detected 3D bounding boxes on image and point cloud pairs are depicted in yellow.

Training. For **Car** class, we use Adam optimizer with an initial learning rate 0.002 and weight decay 0.0001. In Stage-1, we train the network for 8K iterations with batch-size 25. In Stage-2, the whole training process takes 50K iterations with batch-size 800. For **Pedestrian** class, we use the same parameters to train Stage-1 model and reduce the training process to 20K iterations in Stage-2.

2 Qualitative Results on KITTI `val` Set (Pedestrian)

In Fig. 1, we visualize representative outputs of our model on KITTI `val` set for **Pedestrian** class. As seen, for simple cases of non-occluded objects in reasonable distance which we got enough number of points, our model outputs remarkably accurate 3D bounding boxes (like subfigures 3, 4 and 5). Second, we are surprised to find that our model can even correctly predict some highly occluded ones (subfigure 1) and works well in several crowded scenes (subfigures 2 and 6). This proves that our proposed detector not only handles well vehicles, but also adapts to other challenging classes in autonomous driving scenes, under less and easily acquired supervision.

3 Annotation Results on KITTI `val` Set (Car)

Due to our specific network architecture and weakly supervised learning protocol, our model, once trained, can be applied as an annotation tool, which allows automatic and active annotation modes, to improve annotation efficiency. In Fig. 2, we present some annotation results generated from automatic and active modes. It can be observed that in most cases our model with automatic mode

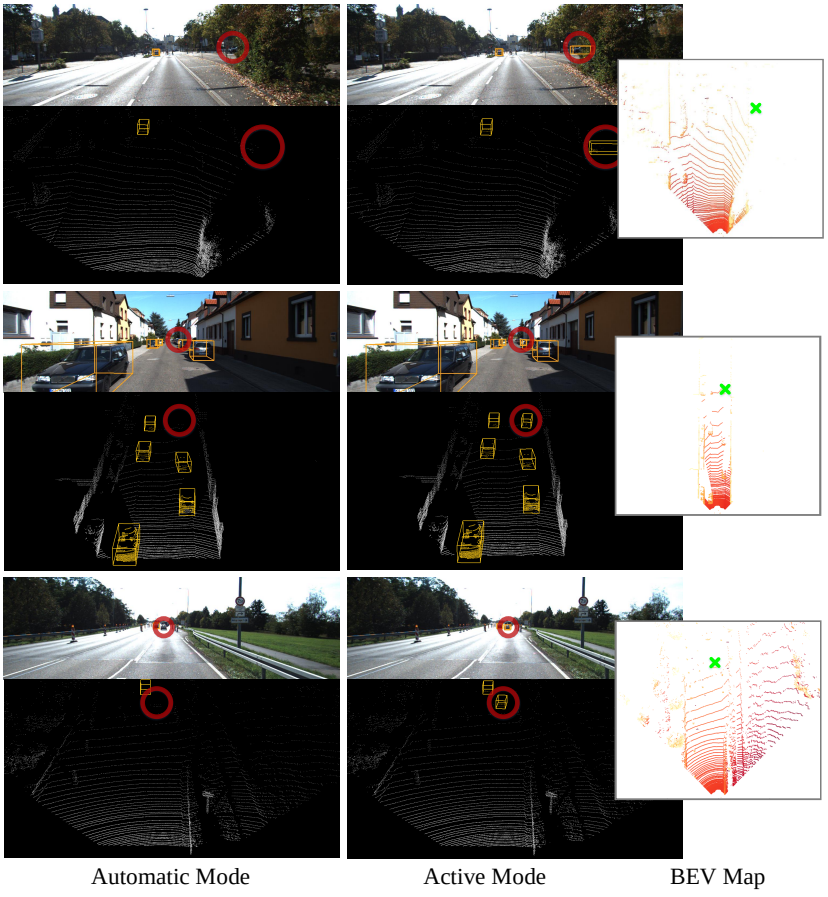

Fig. 2. Annotation results for 3D object detection (Car) on KITTI `val` set. labeled 3D bounding boxes on image and point cloud pairs are depicted in yellow. The improved annotations are highlighted by red circles. Zoom-in for details.

can obtain high-quality annotation results. In addition, our model allows human annotators to place extra clicks on the centers of desired objects, thus the inferior or missing predictions can be corrected. In the active mode, with the weak supervision provided by human annotators, better proposals can be generated around the click points and thus leading to improved predictions.

4 Failure Cases on KITTI `val` Set (Car&Pedestrian)

Though our predictions for cars are particularly accurate, there are still common failure modes, summarized in Fig. 3. The *first* type of common mistakes are caused by the heavy occlusions, such as the vehicle in subfigure 1, highlighted by the red circle, is predicted with wrong height. We think leveraging more

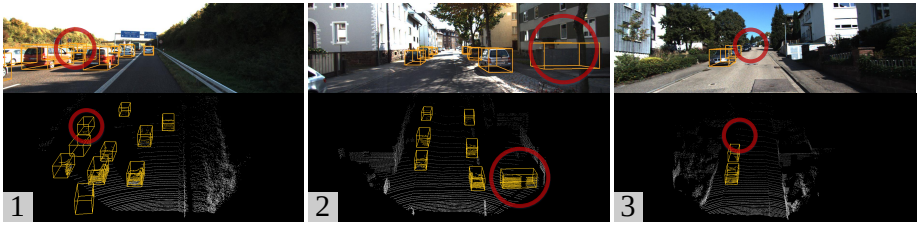

Fig. 3. Failure cases of 3D car detection on KITTI `val` set. Predicted 3D bounding boxes on image and point cloud pairs are depicted in yellow. The inaccurate predictions are highlighted by red circles. Zoom-in for details.

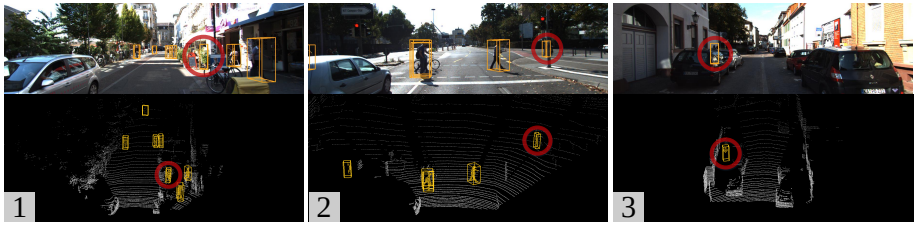

Fig. 4. Failure cases of 3D pedestrian detection on KITTI `val` set. Predicted 3D bounding boxes on image and point cloud pairs are depicted in yellow. The inaccurate predictions are highlighted by red circles. Zoom-in for details.

contextual information may be helpful. The *second* type of challenge is caused by some background objects, like the large box in subfigure 2, which has a similar shape of vehicle. Our model is easily confused, as these background objects look very like vehicles in the point cloud. *Third*, for some challenging cases where the foreground points are extremely sparse, our model is hard to make accurate predictions. Subfigure 3 shows a typical example, where the points of the highlighted vehicles are very few due to the occlusion of hillside. The last two problem can be partially mitigated by considering extra appearance information from camera images. Detecting pedestrians is more challenging and leads to similar lapses. As we can see in Fig. 4, the model is occasionally confused by cylindrical obstacles such as the plant in subfigure 1 and the pole in subfigure 2, which are false positives. In subfigure 3, the two pedestrians are very close and highly occluded, making our output mix them together. Above challenges also indicate possible directions for our future efforts.

References

1. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR. (2012) 1
2. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR. (2019) 1