3D Object Detection with Temporal Information

[1] *Tsung-Lin, Tsou* (鄒宗霖) *Chiou-Shann Fuh* (傅楸善)

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
r10922081@ntu.edu.tw fuh@csie.ntu.edu.tw

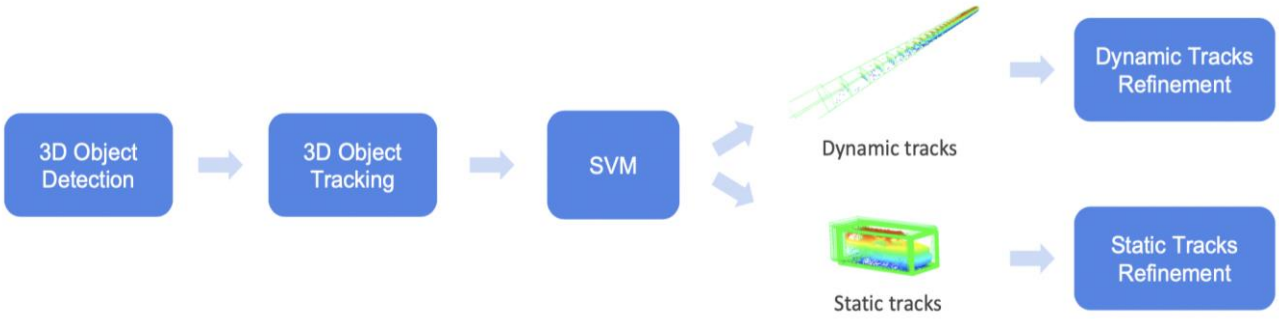

Fig. 1. The overall pipeline of 3DAL [1].

ABSTRACT

There have been many studies on 3D object detection in recent years. In particular, 3D object detection from LiDAR point clouds has become a practical solution, especially in autonomous driving applications. However, we observe that current 3D object detectors have the problem that the longer the distance between object and sensor, the poorer the outcome. Besides, the objects closer to the sensor have the denser point clouds, while the objects further from the sensor have the sparser point clouds. This makes it more difficult for 3D object detector to identify the distant objects.

Therefore, our work is to follow the CVPR 2021 paper 3DAL [1] and improve existing 3D object detector by using temporal information obtained from 3D object tracking. Specifically, while most of the 3D object detector focuses on single-frame input, real-time, and onboard scenario, 3DAL focuses on offboard usage. There are many offboard use cases of perception largely underexplored, such as using machines to automatically generate high-quality 3D labels, fusing multi-frame information and aggregating LiDAR point clouds to get more complete data and improve performance.

Since the code of 3DAL is not available, our main contribution in this work is to use comparable components to reproduce it and prove the validity of temporal information via visualization.

We evaluate 3DAL on the Waymo Open Dataset (WOD): large-scale autonomous driving benchmark containing more than 1,000 LiDAR (Light Detection And Ranging) scan sequences with 3D annotations for every frame. 3DAL pipeline dramatically lifts the perception quality compared with existing 3D object detectors that are designed for the real-time, onboard use cases. We also conduct ablation experiments such as static tracks refinement and dynamic tracks refinement, and provide detection visualization results with and without temporal information.

Keywords: *LiDAR-based 3D object detection, 3D object detection, 3D object tracking, temporal information, point clouds visualization, Waymo Open Dataset (WOD).*

Table 1. Performance (AP: Average Precision) of current 3D object detector CenterPoint [3] on Waymo Open Dataset validation set, breaking down by range [0, 30), [30, 50), [50, +∞) meters. We can see that the longer the distance between object and sensor, the poorer the outcome, which is even more obvious for [50, +∞) meters.

	Vehicle	Pedestrian	Cyclist
(0, 30)	0.902	0.773	0.799
[30, 50)	0.698	0.696	0.671
[50, +∞)	0.441	0.574	0.535

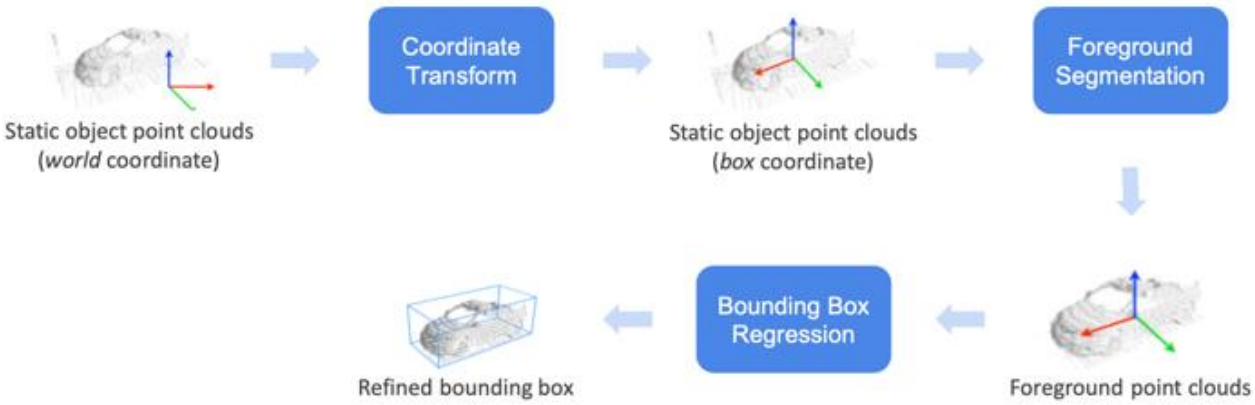

Fig. 4. The model architecture of static tracks refinement.

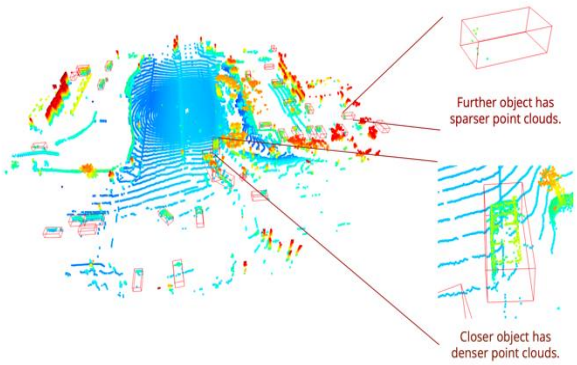

Fig. 2. The visualization of point clouds with different distances between object and sensor. We can see that the objects closer to the sensor have the denser point clouds, while the objects further from the sensor have the sparser point clouds.

1. INTRODUCTION

With the advancement of 3D deep learning and strong application demands, there have been many studies on 3D object detection [2, 3, 4, 5, 6, 7, 8] in recent years. In particular, 3D object detection from LiDAR point clouds has become a practical solution, especially in autonomous driving applications. However, from Table 1, we can observe current 3D object detectors have the problem that the longer the distance between object and sensor, the poorer the outcome. In addition, the objects closer to the sensor have the denser point clouds, while the objects farther from the sensor have the sparser point clouds in Figure 2. This makes it more difficult for 3D object detector to identify the distant objects.

Therefore, our work is to follow the CVPR 2021 paper 3DAL [1] and improve existing 3D object detector by using temporal information obtained from 3D object tracking. Specifically, while most of the 3D object detector focuses on single-frame input, real-time, and onboard scenario, 3DAL focuses on offboard usage. There are many offboard use cases of perception that are largely underexplored, such as using machines to automatically generate high-quality 3D labels, which fusing multi-frame information and aggregating LiDAR point clouds to get more complete data and improve performance in Figure 3.

We evaluate 3DAL [1] on the Waymo Open Dataset (WOD): a large-scale autonomous driving benchmark containing more than 1,000 LiDAR scan sequences with 3D annotations for every frame. The dataset includes a total number of 1150 sequences with 798 for training, 202 for validation, and 150 for testing. Each LiDAR sequence lasts around 20 seconds with a sampling frequency at 10Hz. For our experiments, we evaluate 3D object detection metrics for vehicles and pedestrians.

Finally, the 3D object detector with 3DAL [1] pipeline dramatically lifts the perception quality compared with existing 3D object detectors that are designed for the real-time, onboard use. We also conduct ablation experiments such as static tracks refinement and dynamic tracks refinement, and provide detection visualization results with and without temporal information.

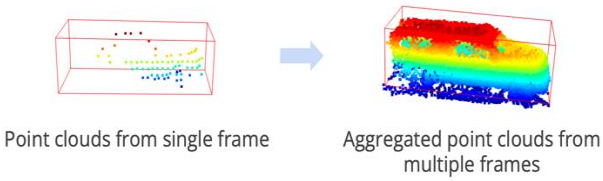

Fig. 3. The visualization of point clouds from different numbers of frames. We can see that after fusing multi-frame information and aggregating LiDAR point clouds, the data become more complete.

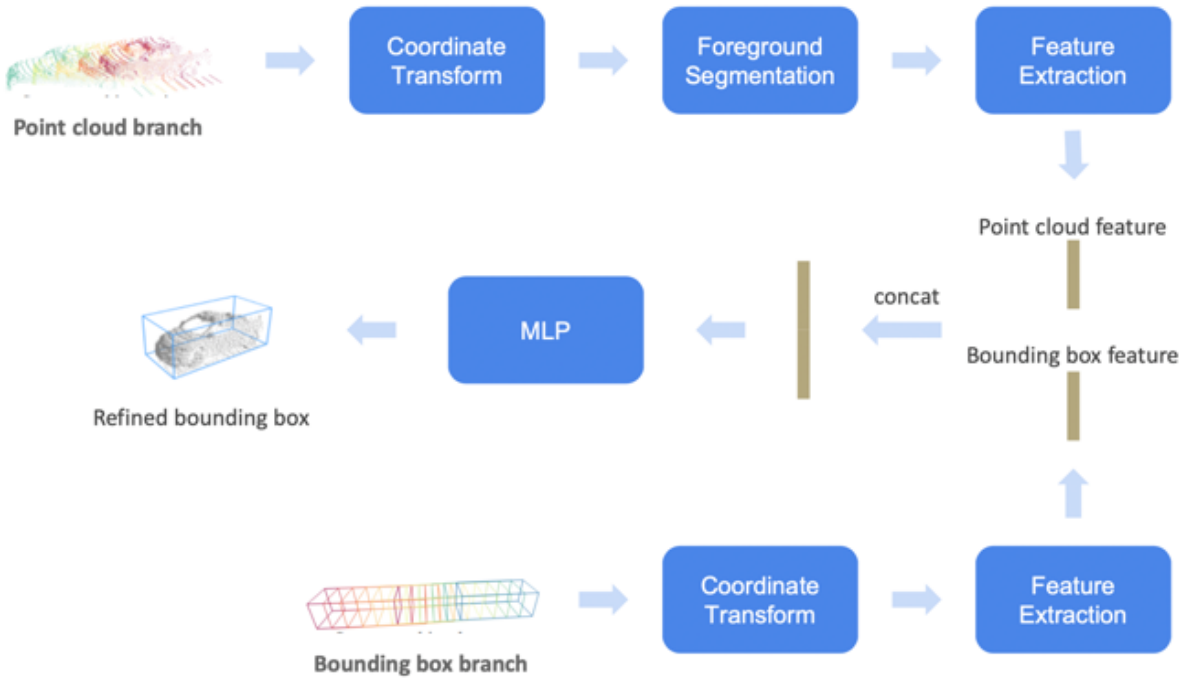

Fig. 5. The model architecture of dynamic tracks refinement.

2. METHODOLOGY

Our method is mainly based on 3DAL [1]. Since the code of 3DAL is not available, our main contribution in this project is to use comparable components to reproduce it and prove the validity of temporal information via visualization. First, 3DAL uses 3D object detection module to obtain the initial 3D bounding boxes. Then, 3D object tracking module is applied to obtain temporal information to densify the LiDAR point clouds. Finally, it uses the more complete data, the densified LiDAR point clouds, to refine the 3D bounding boxes. We will elaborate each component of 3DAL in the following subsections: 3D object detection, 3D object tracking, motion state classification, static tracks refinement, and dynamic tracks refinement in Figure 1.

2.1. 3D Object Detection and 3D Object Tracking

In this section, we simply use the current 3D object detector CenterPoint [3] to obtain static tracks and dynamic tracks in Figure 1. First, CenterPoint voxelizes the LiDAR point clouds into small voxels across 3D space. Then, voxel feature extraction and 3D feature extraction are applied to obtain 2D BEV pseudo image feature map. Finally, it uses the 2D BEV pseudo image feature map to predict 3D bounding boxes, and it can be divided into two stages. In the first stage, it predicts the centers of 3D bounding boxes and regresses 3D information, which includes the velocity being used in 3D object tracking. In the second stage, it predicts the confidence scores and refines the estimates.

Given tracked 3D bounding boxes after 3D object detection and 3D object tracking for an object, we can extract the object's LiDAR point clouds from the temporal sequence. To extract data in the tracks, we first transform all 3D bounding boxes and LiDAR point clouds to the World Coordinate System (WCS) through the known sensor poses. Then we crop the LiDAR point clouds which belong to the object. In 3DAL, they use the term object track data to refer to such 4D (3D spatial information and 1D temporal information) object information.

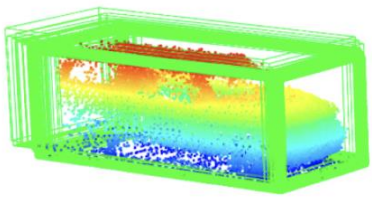

Fig. 8. Visualization of a static track. Since the vehicle is in a static state for a certain amount of time, its LiDAR point clouds can be aligned in the World Coordinate System (WSC). (Green boxes: the 3D bounding boxes belong to the vehicle. Colored LiDAR point clouds: the LiDAR point clouds belong to the vehicle.)

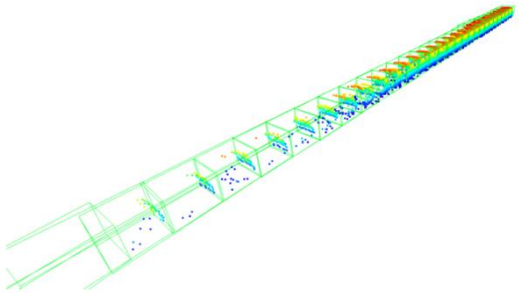

Fig. 9. Visualization of dynamic track. Since the vehicle is not in a static state for a certain amount of time, its LiDAR point clouds cannot be aligned in the World Coordinate System (WSC). (Green boxes: the 3D bounding boxes belong to the vehicle. Colored LiDAR point clouds: the LiDAR point clouds belong to the vehicle.)

2.2. Motion State Classification

Thus in 3DAL that in reality, numerous objects are in a static state for a certain amount of time. For example, motionless cars or buildings in a city do not move within minutes, hours or even days. In terms of offboard 3D object detection, it is preferred to assign a single 3D bounding box to a static object rather than separate bounding boxes in different frames to avoid jittering.

As a result, we resort to different approaches for static and dynamic objects and use the Support Vector Machine (SVM) to classify an object's motion state (static or not) before tracks refinement. Although it is difficult to predict whether an object is static or not from just a few frames due to the detection noise, it is rather easy if all data in the tracks are used.

In this section, we use some heuristic features, i.e., begin-to-end distance of a track, variance of center of a track, and a Support Vector Machine (SVM) to classify the object tracks obtained by 3D object detection and 3D object tracking into stack tracks or dynamic tracks. Albeit the model is simple, the accuracy of classification can be as high as 97 percent. (Note: All components in this section are implemented by ourselves.)

2.3. Static Tracks Refinement

In this section, the static tracks refinement model takes the merged LiDAR point clouds from different frames and predicts a single 3D bounding box. Moreover, the box can then be transformed to each frame through the known sensor poses.

More specifically, for the static tracks classified by Support Vector Machine (SVM), we use static tracks refinement model in Figure 4, to regress the final 3D bounding boxes in static tracks. And we only need to predict a single box for each static track and it can be transformed to each frame within the static track through the known sensor poses.

First, we transform the LiDAR point clouds belong to the object to a box coordinate before the per-object 3D bounding box refinement, so that the LiDAR point clouds are more aligned across objects. In the box coordinate, the +X axis is the box heading direction, the origin is the center of the 3D bounding box. Then, the LiDAR point clouds belong to the object are passed through a PointNet-based foreground segmentation network to segment the foreground. Finally, the foreground LiDAR point clouds belong to the object will be regressed by a PointNet-based bounding box regression network. (Note: All components in this section are implemented by ourselves.)

2.4. Dynamic Tracks Refinement

In this section, the dynamic tracks refinement model needs to predict different 3D bounding boxes for each frame. Due to the sequence input, the model design space is much larger than the static one. More specifically, for the dynamic tracks classified by Support Vector Machine (SVM), we use dynamic tracks refinement model in Figure 5, to regress the final 3D bounding boxes in dynamic tracks. We leverage both the point cloud sequence and the bounding box sequence without aligning object points to a keyframe explicitly as in 3DAL [1] to predict different 3D bounding boxes for each frame within the dynamic track. For the point cloud branch, we first transform the LiDAR point clouds belong to the object to a box coordinate and segment the foreground similar to the static one. Then we use a PointNet-based model to extract point cloud feature. For the bounding box branch, we use similar method to extract bounding box feature. Finally, the features are concatenated to form the joint feature which will be passed through a PointNet-based box regression network to predict the final 3D bounding boxes. (Note: All components in this section are implemented by ourselves.)

In addition to the dynamic track refinement model in 3DAL [1], we try to align or register LiDAR point clouds with respect to a keyframe (e.g. the current frame) to obtain more complete data for 3D bounding box estimation. Instead of feeding the whole track into the network, we can try to estimate the velocity of each object in the 3D object detection section and map all the LiDAR point clouds in other frames to the keyframe. By solving the alignment error, we may get a better 3D object detection in dynamic tracks.

Table 2. The performance (2D Intersection Over Union, 3D Intersection Over Union, 3D accuracy) of static tracks refinement. (CenterPoint: the detection results without using temporal information. Ours: the detection results with using temporal information.)

	2D IOU	3D IOU	3D accuracy
CenterPoint	0.858	0.768	0.774
Ours	0.869	0.785	0.835
3DAL	-	-	0.823

Table 3. The performance (2D Intersection Over Union, 3D Intersection Over Union, 3D accuracy) of dynamic tracks refinement. (CenterPoint: the detection results without using temporal information. Ours: the detection results with using temporal information.)

	2D IOU	3D IOU	3D accuracy
CenterPoint	0.764	0.703	0.774
Ours	0.780	0.717	0.858
3DAL	-	-	0.857

3. EXPERIMENT

We evaluate our method on Waymo Open Dataset (WOD): a large-scale autonomous driving benchmark containing more than 1,000 LiDAR scan sequences with 3D annotations for every frame. The dataset includes a total number of 1,150 sequences with 798 for training, 202 for validation, and 150 for testing. Each LiDAR sequence lasts around 20 seconds with a sampling frequency at 10Hz. For our experiments, we evaluate 3D object detection metrics for vehicles and pedestrians.

Finally, the 3D object detector with 3DAL [1] pipeline dramatically lifts the perception quality compared with existing 3D object detectors that are designed for the real-time, onboard use. We also conduct ablation experiments such as static tracks refinement and dynamic tracks refinement, and provide detection visualization results with and without temporal information.

In this section, we show the results of the three experiments. The performance (2D Intersection Over Union, 3D Intersection Over Union, 3D accuracy) of static tracks refinement is shown in Table 2. The performance (2D Intersection Over Union, 3D Intersection Over Union, 3D accuracy) of dynamic tracks refinement is shown in Table 3. And the performance (mAP: mean Average Precision) on Waymo Open Dataset validation set of 3DAL and the method implemented by ourselves is shown in Table 4.

3.1. Experimental Results

As shown in Tables 2 and 3, the method implemented by ourselves (the detection results with using temporal information) expectedly outperforms CenterPoint (the detection results without using temporal information) in static tracks refinement and dynamic tracks refinement given that it uses additional temporal information.

Furthermore, we also get better accuracy than 3DAL [1] by about 1% and 0.1% on static tracks refinement and dynamic tracks refinement, respectively. This is because the 3D object detector we used in section 2.1 is able to yield better performance results than the detector used in 3DAL [1].

However, in Table 4, due to the use of Test-Time Augmentation (TTA) in 3DAL [1], their overall performance on Waymo Open Dataset (WOD) validation set is better than ours. (test-time augmentation used in 3DAL: by rotating the point cloud around Z-axis by 10 different angles i.e. [0, $\pm 1/8\pi$, $\pm 1/4\pi$, $\pm 3/4\pi$, $\pm 7/8\pi$, π], and ensembling the 3D bounding boxes with weighted box fusion.)

In 3DAL [1], although Test Time Augmentation (TTA) can be parallelized across multiple devices for faster execution, it may lead to excessive computational complexity and memory usage. We decide not to conduct test time augmentation for the following two reasons: First, we aim to validate whether temporal information is helpful to 3D object detection, instead of using augmentation to increase accuracy. Second, we want to save the large amount of computational complexity and memory that Test Time Augmentation (TTA) will cost.

3.2. Alignment experiment

In addition to the dynamic track refinement model mentioned in 3DAL [1], we try to align or register LiDAR point clouds with respect to a keyframe (e.g. the current frame) to obtain a more complete data for 3D bounding box estimation. Instead of feeding the whole track into the network, we can try to estimate the velocity of each object in the 3D object detection section and map all the LiDAR point clouds in the other frames to the keyframe. By solving the alignment error, we may get a better 3D object detection in dynamic tracks.

However, the alignment can be a harder problem especially for occluded or faraway objects with fewer points. Although we do have conducted the experiment that tries to align LiDAR point clouds across frames, the result of the experiment fails to satisfy our expectation. Hence, this experiment will be further designed in our future work.

Table 4. Performance (mAP: mean Average Precision) on Waymo Open Dataset validation set of 3DAL and the method implemented by ourselves

	Detection	Temporal	TTA	Veh	Ped
Ours	o			0.767	0.790
Ours	o	o		0.789	0.812
3DAL	o			0.746	0.780
3DAL	o	o	o	0.845	0.829

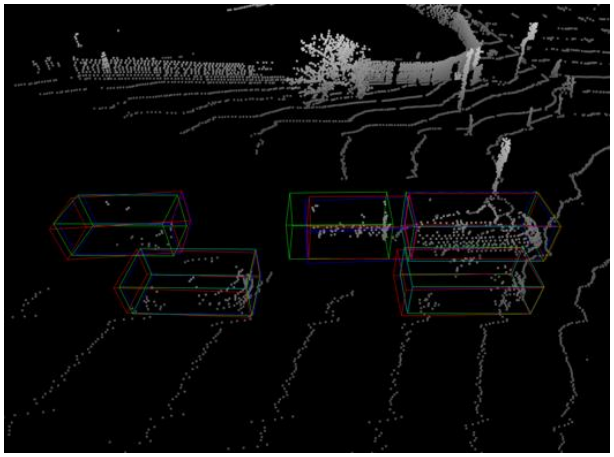

Fig. 6. Visualization of detection results with and without temporal information. (Red: ground truth, Green: 3D bounding boxes without using temporal information, Blue: 3D bounding boxes with using temporal information.)

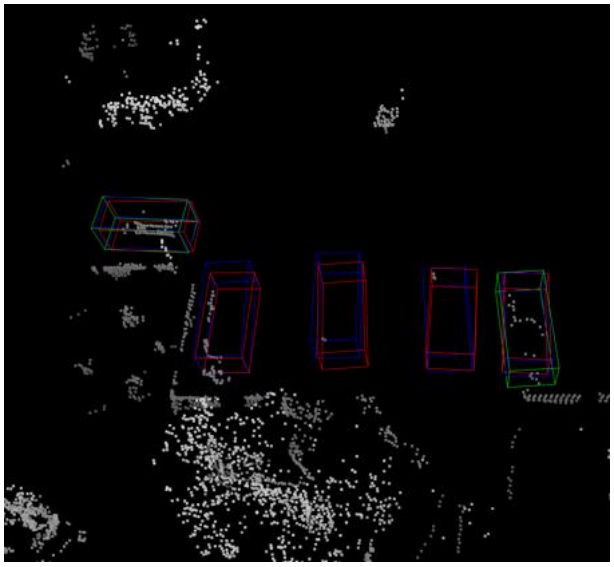

Fig. 7. Visualization of detection results with and without temporal information. (Red: ground truth, Green: 3D bounding boxes without using temporal information, Blue: 3D bounding boxes with using temporal information.)

3.3. Visualization

We can validate our thought (temporal information is helpful to 3D object detection) by visualizing the detection results. In Figures 6 and 7, we can see that for the objects with sparse LiDAR point clouds, the method that utilizes the temporal information has more accurate detection results than the method that does not use it. Besides, in Figure 7, the method that utilizes the temporal information is the only way to detect the objects with even sparser LiDAR point clouds. It is because the aggregated LiDAR point clouds provide more complete data.

4. CONCLUSION

We demonstrate that considering LiDAR point clouds in multiple frames do help improve the 3D object detection, especially for the objects farther from the sensor with sparse point cloud in single frame. Another interesting finding is that we find that dynamic objects have higher 3D accuracy in comparison with static objects, this is because they are closer to the sensor than static ones.

By replacing some components in 3DAL [1], it is shown that our performance is better than the original paper in accuracy for static tracks refinement and dynamic tracks refinement, which means that leveraging adjacent frame information does help 3D object detection and it is not specific to certain model structure.

Finally, the 3D object detector with 3DAL pipeline dramatically lifts the perception quality compared with existing 3D object detectors that are designed for the real-time, onboard use. But we decided not to conduct test time augmentation for the reason that the main purpose of our project is to validate whether temporal information is helpful to 3D object detection, instead of using augmentation to increase accuracy.

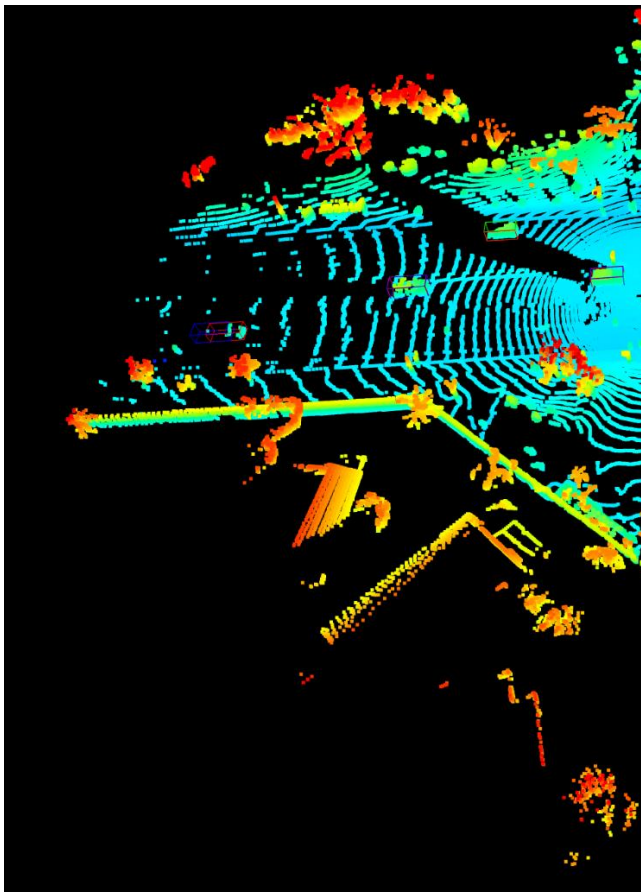

Fig. 8. Visualization of LiDAR point clouds and detection results in a single Waymo Open Dataset validation frame. (by the well-known visualization tools Open3d)

4. FUTURE WORK

In addition to the dynamic track refinement model mentioned in 3DAL [1], we try to align or register LiDAR point clouds with respect to a keyframe (e.g. the current frame) to obtain a more complete data for 3D bounding box estimation. Instead of feeding the whole track into the network, we can try to estimate the velocity of each object in the 3D object detection section and map all the LiDAR point clouds in the other frames to the keyframe. By solving the alignment error, we may get a better 3D object detection in dynamic tracks.

However, the alignment can be a harder problem especially for occluded or faraway objects with fewer points. Although we do have conducted the experiment that tries to align LiDAR point clouds across frames, the result of the experiment fails to satisfy our expectation. Hence, this experiment will be further designed in our future work.

Besides, neither 3DAL [1] nor the method implemented by ourselves cannot be applied to another domain. More specifically, the model trained on the Waymo Open Dataset (WOD) cannot be applied to another dataset like KITTI dataset, nuScenes dataset, Lyft dataset to name a few. This is because there are some domain shifts among these datasets [9, 10] (e.g., different number of LiDAR beam ways, different LiDAR beam angles, different number of LiDAR point clouds per object or per frame, different weather condition…), but we have not considered any of them yet.

Furthermore, the other problem occurs in the training of 3DAL [1] is that it needs a large amount of 3D labels. Unfortunately, manually annotating the 3D labels is laborious and costly. But labeling weak labels like 2D bounding boxes can be 3-16 times faster than 3D labels [11]. Thus, a valuable future work of this project is exploring how to use weak labels like 2D bounding boxes along with sparse LiDAR point clouds to perform 3D object detection.

5. FAILURE CASES

Although using temporal information obtained from 3D object tracking is extremely effective to improve existing 3D object detector, failure cases still exist. For example, in Figures 9 and 10, while the number of LiDAR point

clouds belonging to the object is already significant in the single frame, adding temporal information from multi-frame only induces more noisy LiDAR points, which leads to lower accuracy of 3D bounding boxes.

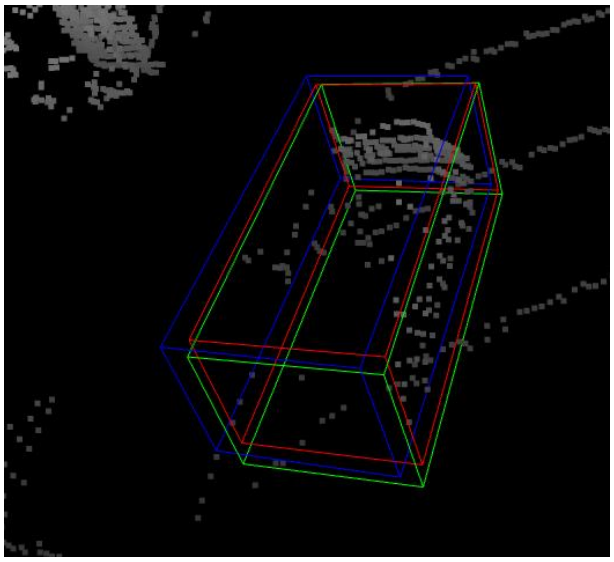

Fig. 9. Visualization of detection results with and without temporal information. (Red: ground truth, Green: 3D bounding boxes without using temporal information, Blue: 3D bounding boxes with using temporal information.)

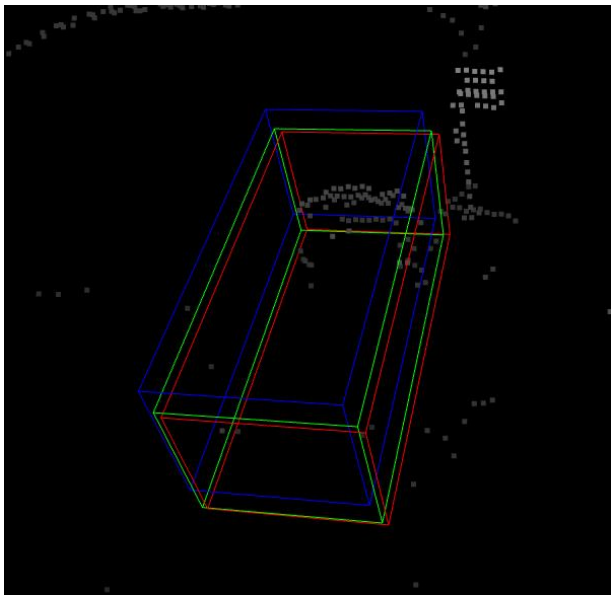

Fig. 10. Visualization of detection results with and without temporal information. (Red: ground truth, Green: 3D bounding boxes without using temporal information, Blue: 3D bounding boxes with using temporal information.)

REFERENCES

[1] C. R. Qi, Y. Zhou, M. Najibi, P. Sun, K. Vo, B. Y. Deng, and D. Anguelov, "Offboard 3D Object Detection from Point Cloud Sequences," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, pp. 6134-6144, 2021.

[2] S. S. Shi, C. X. Guo, L. Jiang, Z. Wang, J. P. Shi, X. G. Wang, and H. S. Li, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," P*roceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, pp. 10529-10538, 2020.

[3] T. W. Yin, X. Y. Zhou, and P. Krahenbuhl, "Center-Based 3D Object Detection and Tracking," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, pp. 11784-11793, 2021.

[4] Y. H. Hu, Z. Z. Ding, R. Z. Ge, W. X. Shao, L. Huang, K. Li, and Q. Liu, "AFDetV2: Rethinking the Necessity of the Second Stage for Object Detection from Point Clouds," *AAAI*, Virtual, 2022.

[5] Y. Zhou, and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud based 3D Object Detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, pp. 4490-4499, 2018.

[6] A. H. Lang, S. Vora, H. Caesar, L. B. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, pp. 12697-12705, 2019.

[7] S. S. Shi, X. G. Wang, and H. S. Li, "PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, pp. 770-779, 2019.

[8] Z. T. Yang, Y. N. Sun, S. Liu, and J. Y. Jia, "3DSSD: Point-based 3D Single Stage Object Detector," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, pp. 11040-11048, 2020.

[9] Y. Wang, X. Y. Chen, Y. R. You, L. E. Li, B. Hariharan, M. Campbell, K. Q. Weinberger, and W. L. Chao, "Train in Germany Test in the USA: Making 3D Object Detectors Generalize," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, pp. 11713-11723, 2020.

[10] J. H. Yang, S. S. Shi, Z. Wang, H. S. Li, and X. J. Qi, "ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, pp. 10368-10378, 2021.

[11] Y. Wei, S. Su, J. W. Lu, and J. Zhou, "FGR: Frustum-Aware Geometric Reasoning for Weakly Supervised 3D Vehicle Detection," *IEEE International Conference on Robotics and Automation (ICRA)*, Virtual, pp. 4348-4354, 2021.