siaNMS: Non-Maximum Suppression with Siamese Networks for Multi-Camera 3D Object Detection

Irene Cortés, Jorge Beltrán, Arturo de la Escalera and Fernando García
Intelligent Systems Laboratory (LSI) Research Group
Universidad Carlos III de Madrid, Leganés, Madrid, Spain
{irecorte, jbeltran, escalera, fegarcia}@ing.uc3m.es

***Abstract*— The rapid development of embedded hardware in autonomous vehicles broadens their computational capabilities, thus bringing the possibility to mount more complete sensor setups able to handle driving scenarios of higher complexity. As a result, new challenges such as multiple detections of the same object have to be addressed. In this work, a siamese network is integrated into the pipeline of a well-known 3D object detector approach to suppress duplicate proposals coming from different cameras via re-identification. Additionally, associations are exploited to enhance the 3D box regression of the object by aggregating their corresponding LiDAR frustums. The experimental evaluation on the nuScenes dataset shows that the proposed method outperforms traditional NMS approaches.**

I. INTRODUCTION

Self-driving vehicles are going to change the future of transportation in terms of safety, efficiency, and pollution. One of the key advantages of this upcoming technology is the potential to reduce the number of road accidents. For this reason, vehicle control will rely on an accurate perception of the environment.

The task of 3D object detection plays a major role in the evolution of the autonomous driving field. However, since supervised deep neural networks have outperformed any pre-existing methods, its development highly depends on the availability of public annotated datasets. In this regard, KITTI [1] has been considered the standard benchmark for many years now. Nonetheless, despite the quality of the annotations and the variety of scenes, it has proved to be insufficient to build robust object detectors for driving scenarios with adverse light or weather conditions.

To tackle this lack of data availability and boost research, many datasets have been released recently. Unlike KITTI, these new collections usually include more complete sensor setups with higher redundancy and are tailored for 360° perception. Thus, we can find nuScenes [2], Waymo [3], Lyft [4], or Argoverse [5], which include information captured from several surrounding cameras, one or more laser scanners, radars, GPS, and whose sizes are several orders of magnitude greater than KITTI's.

The emergence of these new annotated datasets is a big opportunity for progress in the 3D object detection domain, but at the same time, it opens up a set of new challenging tasks, such as processing a vast quantity of data in real-time, or dealing with misalignments among detections coming from different sources of information. Although the first issue may be solved by scaling the hardware, the latter especially affects those methods that take images as input, as they can only work with a limited view of the environment.

Fig. 1. Sample of NuScenes labels. Objects on a single image are colored in orange, while those on two consecutive cameras are shown in yellow.

In this paper, we insert a re-identification module in a popular state-of-the-art 3D object detector, F-PointNets [6], to improve the performance of the box regression of objects on the side of the image. The proposed framework is fed with pairs of 2D proposals from contiguous surrounding cameras mounted on a 360° on-board setup, and provides a similarity estimation so that detections of the same object in different cameras can be matched. In this manner, the siamese network permits not only to suppress multiple detections of the same obstacle in a traditional Non-Maximum Suppresion (NMS) fashion, but also to aggregate the corresponding LiDAR points associated with both occurrences. As a result, a more faithful and complete representation of the object in the spatial modality is created, which can enhance the 3D box estimation performed in the last step of the pipeline.

The rest of this paper is structured as follows. In Section II, a review of the related works is provided. Sections III and IV include the description of the proposed approach and the details of the network design and training, respectively. The experimental results are discussed in Section V. Finally, conclusions and future work are presented in Section VI.

II. RELATED WORK

Perception for autonomous vehicles is widely dominated by deep learning approaches, which usually process LiDAR and cameras data to estimate the 3D position of the surrounding objects. Within these methods, some are focused on object detection using a single sensor, while others make use of data from multiple modalities.

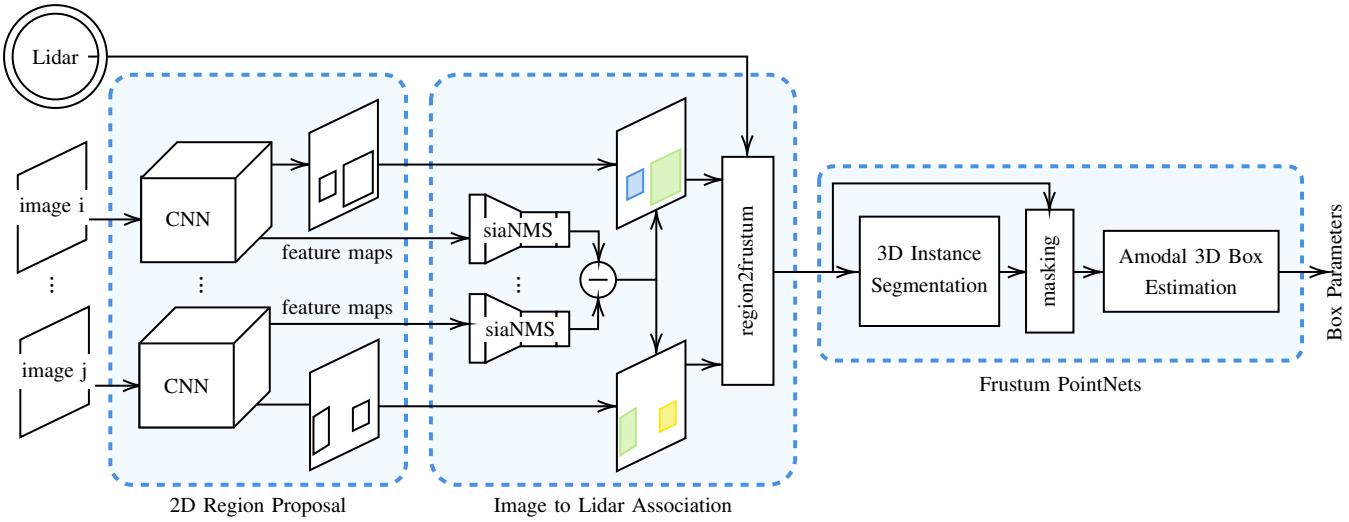

Fig. 2. General system overview. The images for all the cameras are processed through the 2D CNN object detector, which proposes 2D regions and provides the feature maps for each proposal. Those features are then introduced to the siaNMS, which will determine which detections correspond to the same object. Taking this into consideration, the frustum areas for each detection are computed, and the point cloud input for the F-PointNets is calculated. Finally, the 3D box parameters are estimated as proposed by Qi *et al.* in [6].

LiDAR 3D detection. The 3D point cloud captured by laser scanners provides reliable/precise geometry and reflection information in the long-range, typically covering 360° in the horizontal field of view (HFOV). On the contrary, the unstructured nature of LiDAR data caused by its uneven sparsity makes it hard to be processed efficiently. As a result, a discretization of the cloud is often performed before feeding the networks, either in the form of a voxelization [7], [8], [9], or a Bird Eye's View projection [10], [11].

Monocular 3D detection. Other methods take advantage of the appearance information provided by camera sensors to perform 3D object detection. Although this kind of modality is well structured and contains rich and dense features, it suffers from the lack of spatial data, a limited HFOV, and weak robustness against light changes. In order to deal with RGB inputs, most approaches propose two-stage solutions [12], [13]. First, 2D proposals are computed using a CNN-based detector, and their estimated depth information is then used to obtain the 3D bounding boxes. However, alternative approaches are presenting networks able to infer the final detections while skipping an explicit prediction of a depth map [14], [15].

Multi-modal fusion 3D detection. Lately, a range of deep networks that process both camera and LiDAR data has been presented. The main motivation is the possibility to combine complementary information sources to enhance the learned representation of the objects and increase the robustness of the model against adverse conditions. Nevertheless, how these modalities are fused effectively is yet a matter of research. Currently, two distinct lines are followed. On the one hand, a variety of strategies to perform fusion at the feature level [16], [17], [18] have been introduced. On the other hand, some works divide the process into two steps: performing detection in the image space, and later regressing the 3D box using a subset of the LiDAR modality [6], [19].

Despite the different degrees of maturity of each of these approaches, those using images as input present difficulties when integrated into multiple-cameras setups. First, due to the limited HFOV of these sensors, the accuracy of the detection of objects falling on the side of the image is impaired by truncations. Second, a single object may be detected twice when located within the overlapping area of contiguous cameras.

In this regard, most approaches have opted for greedy NMS algorithms, where all candidate detections are compared to each other to suppress duplicates by computing the IoU of their corresponding boxes and preserving the one with the highest score. However, due to the processing time required to compute the IoU between rotated boxes, an approximation of the method is usually chosen and axis-aligned detections are considered instead, leading to a loss of accuracy.

Alternatively, other methods with similar purposes have been developed within related research fields such as multi-object tracking. For instance, siamese [20] or triplet [21] networks aim to identify multiple occurrences of the same object over time by computing a feature vector in the image space and estimating their similarity by reducing the distance for positive pairs, while increasing it for negative matches.

III. PROPOSED APPROACH

In this paper, we embed a re-identification module into the popular F-PointNets [6] detector, where the camera image is used to obtain object proposals and a further PointNet ensemble estimates the 3D boxes from the corresponding frustum clouds.

Although it is among the state-of-the-art multi-modal fusion approaches, its performance decreases when integrated in multi-camera setups, such as those designed for self-driving perception, as detection of objects truncated by the

HFOV of the camera leads to incomplete LiDAR frustums, damaging the final 3D outcome.

To address this issue, a siamese network is used to associate 2D proposals representing the same object in contiguous cameras. A set of fully-connected layers generate an embedding for every proposal inside the overlapping area of the cameras, and a similarity distance is computed. Then, those pairs whose distance falls below a given threshold α are paired. To reduce the processing time of this stage, feature vectors from an intermediate layer of the 2D detector are used as inputs. Finally, matched bounding boxes are used to extract point cloud frustums from LiDAR, which are added together.

An overview of the whole pipeline is shown in Fig. 2. As can be seen, the proposed re-identification method is placed between the image detection and the 3D estimation stages.

A. Feature map extraction from image detection network

As for the 2D detector, a tuned version of the widespread Faster R-CNN [22] framework is selected. For the backbone, a ResNet-50 [23] with Feature Pyramid Networks (FPN) [24] is used. Weights from a model pre-trained on COCO [25] with an additional instance segmentation branch are used, since image detection benefits from the multi-task learning [26].

As mentioned before, the siamese network's input vectors are taken from an intermediate feature map generated during the 2D detector inference. Therefore, after the final regression of the 2D boxes is computed by the fully connected (FC) layers of the framework, they are scaled down appropriately to extract their corresponding feature map by applying the ROI Align operation over the output of the fourth ResNet block *res4*. Due to the nature of this pooling layer, an fixed-sized feature vector is obtained for every detection, which can be fed into the re-identification network.

B. Object re-identification

In order to determine which detections correspond to the same obstacle, we consider all detections on the overlapping region of two contiguous cameras as candidates. Then, the object re-identification network processes the extracted features and obtains an embedding for each detection, and the L2 distances between all possible combinations are obtained. If the distance is higher than a defined threshold dis_{thr}, the pair is dismissed. The remaining matches are sorted by distance and the ones with the smallest distances are chosen, following the Hungarian Method.

C. Multi-view frustum aggregation

Finally, to obtain the point cloud inputs to the F-PointNets, the three-dimensional region for each frustum is computed as in [6]. For all the non-paired detections, their frustum is determined from the 2D bounding box and through the camera projection matrix. In the case of paired detections, these steps are followed: first, the frustum region for each detection of the object is calculated; then, if both frustum overlap, the LiDAR points composing the union of both regions are selected as inputs. If they don't, the match is considered a false positive and it is dismissed.

Before feeding frustums into F-Pointnets, they have to be rotated so that their central axis is orthogonal to the image plane. To do so, the center of the 2D bounding box is lifted to its corresponding 3D line, which is used as axis. For multi-view detections, the central axis is calculated as $\overline{or\, p_m}$, where or is the origin of the camera, and p_m is the middle point between p_l and p_r, being

$$p_l = \mathcal{F}_1 \cap \mathcal{C}_d, \tag{1}$$

$$p_r = \mathcal{F}_2 \cap \mathcal{C}_d, \tag{2}$$

where $\mathcal{F}_i$ is the frustum region each of the detections and the $\mathcal{C}_d$ is the circumference defined by the maximum detection distance. An example in *Bird's Eye View* can be seen in Fig. 3.

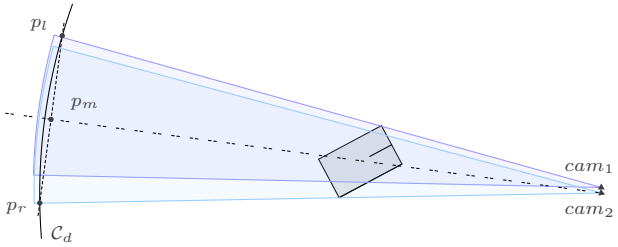

Fig. 3. Multi-frustum axis calculation.

IV. Network description and training

The objective of the siaNMS network is to learn an embedding that transforms the input feature maps into a n-dimensional Euclidean space, and it is trained such that the squared L2 distances between the embeddings of different detections are correlated to the detections similarity. The detailed network architecture is shown in Fig. 4.

A. Loss function definition

For every input feature map x, the output embedding is represented by $f(x) \in \mathbb{R}^d$. We want to ensure that the embedding of a specific detection $f(x_i^r)$ (reference) is closer than a threshold α to the embeddings of all detections $f(x_i^p)$ (positives) of the same object, and that the embedding of any other object x_i^n (negatives) is further away than a threshold β. To achieve this goal, we have used a Double Margin Contrastive Loss [27]. Thus, we want

$$\left\| f\left(x_i^r\right) - f\left(x_i^p\right) \right\|_2 < \alpha, \tag{3}$$

$$\left\| f\left(x_i^r\right) - f\left(x_i^n\right) \right\|_2 > \beta, \tag{4}$$

$$\forall \left(f\left(x_i^r\right), f\left(x_i^p\right), f\left(x_i^n\right) \right) \in \mathcal{P}. \tag{5}$$

where α and β are two constant margins and $\mathcal{P}$ is the set of all possible image pairs in the training set. The loss that is being minimized is then:

$$\begin{aligned} \mathcal{L} = \frac{1}{2} \sum_i^N \Big[& \max\left(\left\| f\left(x_i^r\right) - f\left(x_i^p\right) \right\|_2 - \alpha, 0 \right)^2 + \\ & \max\left(\beta - \left\| f\left(x_i^r\right) - f\left(x_i^n\right) \right\|_2, 0 \right)^2 \Big]. \end{aligned} \tag{6}$$

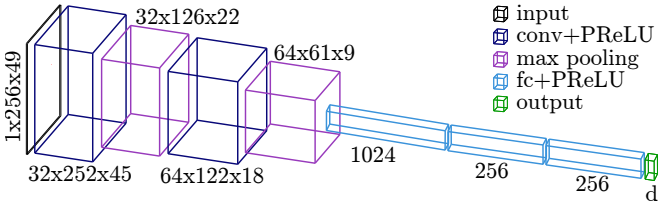

Fig. 4. Siamese Embedding Network. Each encoder is composed by two groups of convolutional and max pooling layers followed by a set of fully connected layers.

B. Dataset preparation

As the dataset used for assessing the proposed method, nuScenes, is divided into scenes with different driving conditions (e.g. weather, lightning and road kind), an *ad hoc* strategy has been followed to build the training and validation splits. Every scene in the dataset contains a sequence of frames, composed in turn by a set of sensor readings, synchronized by timestamp. Additionally, 3D box labels for objects in the scene are provided, where each object is considered an *instance* with annotations for every occurrence along the set of frames. To train the siamese network, only instances whose 3D box projects inside two consecutive cameras are considered.

In order to prepare the training data, two separate processes are followed. First, an offline computation of the network inputs is performed. Second, an online pairing method is applied during the training phase to select the reference and candidate detections.

At the offline step, feature maps of these objects have to be obtained so that they can be used as inputs for the re-identification network. As described in III-A, 2D detection is performed on all the images in the training set. Then, labels are associated with the predicted boxes by computing the IoU in the image plane. To avoid decreasing the training set size, the projection of the 3D ground-truth labels are used for all false negatives.

While training, an Online Hard Example Mining approach [28] is applied in order to form the reference-candidate pairs $\{x_i^r, x_i^c\}$. Concretely, the following steps are used:

1) At every iteration, N reference instances from the training set are taken. For each one, a random occurrence x_i^r is picked.
2) The positive candidate is given by the corresponding bounding box of the same object in the contiguous camera, x_i^p.
3) To obtain the negative pair, x_i^n, another instance appearing in the same scene is randomly selected. Then, every occurrence in the scene of that object in the overlapping area of the contiguous camera is fed into the network, and the corresponding losses are computed. Afterward, the sample with the highest loss value is picked to perform the backward propagation.

C. Training details

The proposed network has been trained for 25 epochs with a batch size of 8. An Adam optimizer is used with an initial learning rate of 0.0001, which decays by a factor of 0.1 every 8 epochs. Regarding the loss function parameters, the lower margin value $\alpha = 1$, while the upper threshold $\beta = 3$. Moreover, a $50 : 50$ ratio of positive and negative samples has been enforced via the online hard negative pair selection method described above.

Besides, the original size of the input images (1600x900) of the 2D detector is scaled down so that the largest size is equal to 1333px. No data augmentation techniques are used.

Different output dimensions for the object embeddings ($d = \{5, 10, 20, 50, 100, 200, 500, 1000\}$) have been tested. According to our validation tests, the 100-dimensional encoding provides the best performance. This can be explained because the amount of parameters is correlated to the capability of representing the appearance of an object. However, from a certain amount on, the network starts to overfit and is unable to generalize properly for examples not included in the training set.

V. RESULTS

The presented approach is evaluated using nuScenes detection benchmark [2]. Concretely, our method falls within the *Open Track* due to the use of both camera and LiDAR information as input. Since the main contribution of this paper is the introduction of a re-identification module in the perception pipeline as an alternative to traditional NMS, instead of the ten classes included in the official detection challenge, the three classes (Car, Pedestrian, and Cyclist) from the original model in F-PointNets paper are used.

A. Evaluation metrics

The used metrics are defined in the nuScenes detection benchmark [2]:

- Average Precision (AP) [%]: a true positive (TP) is defined if the 2D center distance between a detection and a label is smaller than a threshold. This metric is calculated for thresholds of $\{0.5, 1, 2, 4\}$ meters, and then the average for each class is calculated.

The remaining metrics are only calculated for TP detections:

- Average Translation Error (ATE) [m]: Average distance between detection and label centers.
- Average Scale Error (ASE) [%]: The IoU after aligning centers and orientation between detection and label 3D boxes is calculated. ASE is defined as 1 - IoU.
- Average Orientation Error (AOE) [rad]: Yaw angle difference between detection and label boxes in radians.

Other metrics such as Average Velocity Error (AVE) and Average Attribute Error (AAE) are not taken into account as they do not apply to the purpose of this paper.

B. Experimental setup

To evaluate the performance of the proposed approach, results over the nuScenes validation set are provided. For a fair analysis, we consider the same pipeline for 3D detections only with variations in the method to address the grouping of detections between cameras. As shown in Table I, a comparison between three approaches has been considered:

Fig. 5. Results on nuScenes validation set. From left to right: 3D detections using Vanilla, Axis-NMS and siaNMS approaches.

TABLE I
COMPARISON OF THE 3D CAR DETECTION PERFORMANCE ON THE NUSCENES VALIDATION SET IN DIFFERENT REGIONS OF INTEREST

Areas	Vanilla				Axis NMS				siaNMS			
	AP	ATE	ASE	AOE	AP	ATE	ASE	AOE	AP	ATE	ASE	AOE
All	45.5	0.331	18.4	0.371	49.1	0.330	18.4	0.367	**51.1**	**0.320**	**18.2**	**0.350**
Overlap	37.9	0.317	18.6	0.312	46.6	0.312	18.6	0.306	**49.0**	**0.287**	**18.0**	**0.253**

1) A vanilla version of the F-PointNets is used for every camera. Afterward, all the object detections are transformed to the global frame.
2) Detections from the previous solution are filtered using a traditional Greedy NMS to suppress duplicates, with an IoU threshold of 0.3. An axis-aligned NMS approach in Bird's Eye View is selected over a rotated one due to its suitability for real-time applications.
3) The proposed siamese network is used to remove multiple detections of the same object in contiguous cameras. This process is only applied to cars, as they are by far the most frequently truncated class among the considered ones, due to its dimensions.

To better understand the impact of the proposed siamese network, the same evaluation is performed taking into account only the regions of overlap between cameras.

C. Discussion

As can be seen in the results shown in Table I, the presented approach outperforms traditional NMS in all evaluated metrics even though the input feature maps have not been optimized for re-identification tasks. The one that benefited the most is the AP since a large number of redundant objects are removed. Nonetheless, all other metrics are also improved to some extent, due to the fact that having more complete point clouds for truncated obstacles results advantageous for the quality of the 3D box regression. All these effects are magnified when the analysis is performed isolating the overlap areas, as there is were most duplicate detections are found.

As shown in Fig. 5, cases where either the object is truncated in both images or the detection on one of the cameras leads to a wrong 3D box are better resolved by the siaNMS. In such situations, the 3D estimations usually present a greater misalignment due to the incomplete LiDAR input, and may not have sufficient overlap for the NMS to be able to suppress it. These examples are the most benefited from the multi-view frustum aggregation feature of the proposed approach, as can be observed in the first four rows of Fig. 5.

Despite the above, we have detected cases in which the re-identification method works worse than the traditional NMS, see last row of Fig. 5. These are cases in which the object appears with very different perspectives in both images, e.g. when the obstacle is very close to the camera, or the light conditions are not adequate, such as reflections, overexposure, etc. Hence, both methods might be used together as they can provide a complementary behavior in edge case scenarios.

VI. CONCLUSIONS

In this paper, an effective alternative to NMS for suppressing duplicate detections of the same object in multi-camera setups has been presented. To this end, a siamese network has been added between the detection and the 3D box regression stages of a top-performing 3D object detector. Moreover, the association of 2D detection boxes has been exploited to obtain a more reliable representation of the objects in the LiDAR space, improving the quality of the inputs of the subsequent stage in the pipeline.

The proposed work has been evaluated in a challenging 360° object detection benchmark, proving its capability to cope with complex scenarios. According to the experimental results, the embedded re-identification network outperforms the traditional NMS method in average precision and reduces the translation, size and orientation errors thank to the aggregation of 3D frustums from matched image detections.

In future work, several models will be trained to permit the re-identification of other kinds of classes, paying special attention to bulky objects, which are more prone to be truncated by camera images. Besides, the siamese layers will be integrated into the perception pipeline creating an end-to-end deep neural network. Thus, following a multi-task learning strategy, the encoder will be able to compute feature maps that are best suited for both 2D detection and re-identification purposes.

ACKNOWLEDGMENT

Research supported by the Spanish Government through the CICYT projects (TRA2016-78886-C3-1-R and RTI2018-096036-B-C21), Universidad Carlos III of Madrid (PEAVAUTO-CM-UC3M) and the Comunidad de Madrid (SEGVAUTO-4.0-CM P2018/EMT-4362). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[3] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: An open dataset benchmark," *arXiv preprint arXiv:1912.04838*, 2019.

[4] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet, "Lyft level 5 av dataset 2019," urlhttps://level5.lyft.com/dataset/, 2019.

[5] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 918–927.

[7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[8] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Std: Sparse-to-dense 3d object detector for point cloud," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1951–1960.

[10] J. Beltran, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "Birdnet: a 3d object detection framework from lidar information," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3517–3523.

[11] B. Yang, M. Liang, and R. Urtasun, "Hdnet: Exploiting hd maps for 3d object detection," in *Conference on Robot Learning*, 2018, pp. 146–155.

[12] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 867–11 876.

[13] G. Brazil and X. Liu, "M3d-rpn: Monocular 3d region proposal network for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9287–9296.

[14] W. Bao, B. Xu, and Z. Chen, "Monofenet: Monocular 3d object detection with feature enhancement networks," *IEEE Transactions on Image Processing*, 2019.

[15] A. Simonelli, S. R. Bulò, L. Porzi, E. Ricci, and P. Kontschieder, "Single-stage monocular 3d object detection with virtual cameras," *arXiv preprint arXiv:1912.08035*, 2019.

[16] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *arXiv preprint arXiv:1712.02294*, 2017.

[17] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 641–656.

[18] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7345–7353.

[19] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection," in *IROS*. IEEE, 2019.

[20] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 34–39.

[21] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[27] M. J. Gómez-Silva, J. M. Armingol, and A. de la Escalera, "Deep parts similarity learning for person re-identification." in *VISIGRAPP (5: VISAPP)*, 2018, pp. 419–428.

[28] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.