Effect of Post-Processing Methods on CPU-based 3D Object Detection from Camera and LiDAR Data

Nina Masarykova[1], Branislav Fech[1], Marek Galinski[1], Peter Truchly[1]
[1] Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava
Ilkovicova 2, 842 16 Bratislava, Slovakia
nina.masarykova@stuba.sk

***Abstract*—This paper investigates the effect of several post-processing techniques on the accuracy and consistency of a lightweight 3D object detection pipeline using camera and LiDAR data fusion. The baseline camera-LiDAR fusion solution, which we extend in this work, integrates a YOLO-based segmentation with LiDAR point data, without further use of neural networks. This approach focuses on inference suitable for resource-constrained environments. To enhance detection performance, several improvements were introduced: bounding box height calibration using segmentation masks and LiDAR depth data, merging overlapping detections for cyclists, filtering background points from bicycles, and optimizing the center position and orientation of bounding boxes for elongated objects through convex hull analysis and rotation optimization. The evaluation conducted on the View of Delft (VoD) dataset confirmed that these enhancements improved the mean Average Precision (mAP) and Intersection over Union (IoU) for object classes car, cyclist, pedestrian, motorcycle and truck.**

***Keywords*—3D object detection; camera lidar fusion; cpu-based 3D detection; post-processing methods; sensor fusion**

I. Introduction

In order to make roads safer, manufacturers equip vehicles with automated driving solutions from a range of automation levels. One of the most important parts of automated driving solutions is the detection of traffic participants in 3D space. Many solutions rely on the fusion of feature-dense RGB camera images with the LiDAR point cloud, providing valuable depth information, leveraging the complementary strengths of these sensors. Recent research has demonstrated significant advancements in sensor fusion methods [1–3], achieving state-of-the-art results on established multi-sensor datasets such as KITTI [4], nuScenes [5] and more [6], [7]. One of the most frequently used object detection model is You Only Look Once (YOLO) [8]. Although YOLO provides 2D bounding boxes, when combined with depth data, 3D bounding boxes can be obtained. The significant aspect that contributes to the popularity of YOLO-based architecture is undoubtedly its real-time performance and relatively low computational requirements. Compiled smaller sizes of YOLO models can achieve real-time performance even on CPU-only setups. Saucedo et al. [9] proposed lightweight camera-LiDAR fusion for 3D object detection and localization. The method utilizes the YOLOv8 model and Euclidean clustering of the point cloud and achieves 43.2% mIoU. A similar YOLO-LiDAR approach proposed by Kieffer [10] was taken as a baseline for this work.

Deep learning models can effectively produce initial detections. However, studies have highlighted the value of complementing the abilities of deep models with post-processing techniques that refine these outputs. Such techniques are especially important in resource-constrained environments, where the cost of additional processing with a neural network is undesirable.

Both geometric and neural network-based post-processing approaches have been proposed. Chen et al. [11] proposed the post-processing method for bounding box refinement in 2D using Histogram of Oriented Gradients (HOG) features. By extracting the edge information of the object, borders of the boxes were adjusted. DiffuBox [12] applies a diffusion-based refinement on LiDAR point clouds, enhancing 3D detection across domains without retraining the model. For road surface defect detection, Li et al. [13] introduced a clustering and dual threshold box filtering strategy. By combining advantages of Weighted Box Fusion (WBF) and soft Non-Maximum Suppression (NMS), their method outperformed both mentioned.

Our work focuses on evaluating the impact of various post-processing methods on a CPU-based 3D object detection pipeline that utilise YOLO segmentation masks to filter LiDAR points and predict 3D bounding boxes. By applying techniques such as bounding box height calibration, overlapping box merging for cyclists, background points filtering, and rotation optimization, we aim to enhance detection accuracy and consistency, particularly in resource-constrained environments.

II. Method

In this section, we describe the details of the implementation of the 3D object detection pipeline and the post-processing techniques used to improve its accuracy.

A. Baseline method

As a baseline 3D object detection method, we have chosen *YOLO-LiDAR Fusion: Lidar-camera fusion for 3D object detection in autonomous driving systems* [10]. The model was designed to perform the fusion of data from a RGB camera and a LiDAR. In the pipeline, the RGB image is processed using a YOLO segmentation model, producing segmentation masks of detected objects.

The LiDAR point cloud is first filtered to match the field of view (FOV) of the camera. The LiDAR points are projected into the camera plane and further filtered using segmentation

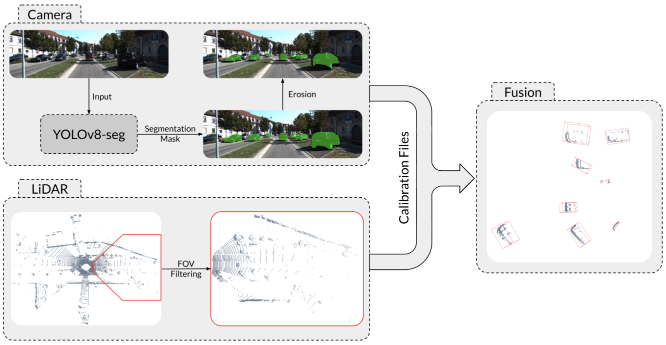

Figure 1. YOLO-LiDAR fusion model architecture, reprint from [10]

masks obtained in the previous step. Only points present within the segmentation masks were preserved, effectively isolating points belonging to detected objects. To remove noisy points caused by the reflectance properties of certain parts of the objects, the DBSCAN clustering algorithm is used.

Using the filtered points for each segmented instance an oriented bounding box is generated using principal component analysis (PCA) and approximation to the minimal bounding box. The graphical representation of the original architecture can be seen in Figure 1.

When tested on a laptop-grade CPU, the whole pipeline was able to process 7 frames per second (FPS), which could be further increased by using compiled version of the YOLO model. These characteristics suggest that the YOLO-LiDAR Fusion model could be suitable for resource-constrained environments where a model with high computational requirements would not be usable.

B. *Post-processing methods*

In following subsections the post-processing methods that were used to refine predicted 3D bounding boxes by the baseline method are described.

1) Bounding Box Height Calibration: The first enhancement was focused on correcting the height of the bounding boxes. For each detected object, the segmentation mask was first eroded in order to reduce the influence of noisy or ambiguous pixels near the edges. From the cleaned mask, the minimum and maximum y coordinates of the polygon were extracted, providing the vertical height of the object in the image pixels.

The intrinsic parameters of the camera—specifically the vertical focal length f_y were used in conjunction with an estimated average depth of the object obtained from the LiDAR point cloud to convert this height into real-world measurements. This depth was calculated as the average coordinate x of all 3D LiDAR points projected inside the eroded mask, under the assumption that x points in the camera's forward direction.

The real-world height in meters was then calculated using the pinhole camera model as stated in Equation 1:

$$h_{\text{seg}} = h_{\text{pix}} \cdot \left(\frac{d_{\text{avg}}}{f_y} \right) \tag{1}$$

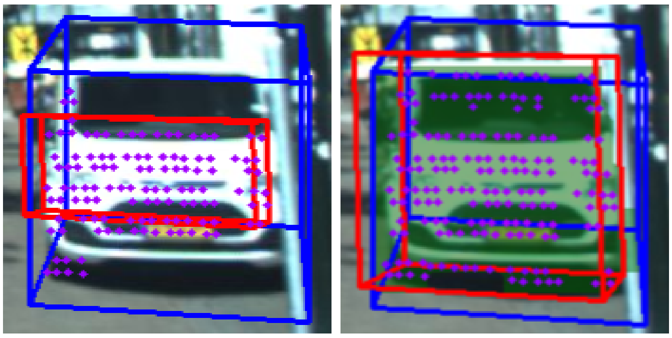

Figure 2. Comparison of bounding boxes for car: the image on the left shows the original bounding box for car (red), while the image on the right shows the bounding box with fixed height of the box taken from segmentation mask (red). Blue boxes represent the ground truth annotation.

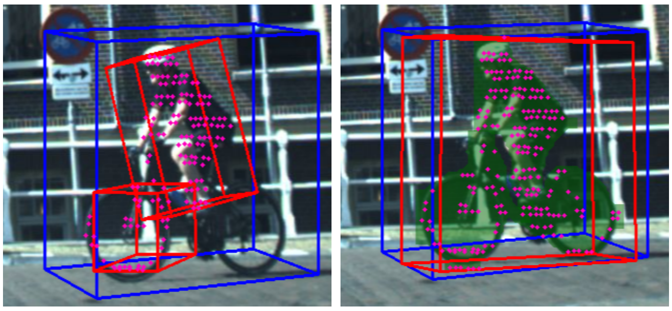

Figure 3. Comparison of bounding boxes for bicycle: the image on the left shows the original separate bounding boxes for bicycle and pedestrian (red), while the image on the right shows the united bounding box encapsulating both the bicycle and pedestrian (red). The blue boxes represent ground truth annotation.

where:
- h_{seg} is the estimated real-world height in meters,
- h_{pix} is the pixel height of the segmentation mask,
- d_{avg} is the average depth of the object,
- f_y is the vertical focal length of the camera.

Subsequently, the computed height was passed into the 3D bounding box generation module and used to directly define the vertical size of the box. Finally, the entire bounding box was vertically shifted so that its base aligns with the lowest LiDAR point within the object. Significant improvements in vertical accuracy and consistency were observed, particularly for cars, as can be seen in Figure 2 and motorcycles.

2) Handling Cyclists and Overlapping Classes: The pre-trained YOLO segmentation model does not include a dedicated *cyclist* class and detects the rider as a *pedestrian* and the bicycle as a *bicycle* instead.

A merging strategy based on the detection of overlapping bounding boxes was implemented. Pairs of pedestrian-bicycle boxes that overlapped in the bird's-eye view (BEV) were considered as *cyclist* and merged by fully enclosing both boxes.

This strategy significantly improved the consistency of 3D box generation for cycling-related objects, as illustrated in Figure 3.

3) Background Filtering for Bicycles: The LiDAR points often pass through gaps in the frame of the bicycle and the wheels, capturing background objects such as walls. This caused the bounding boxes to include unnecessary background points, leading to incorrect depth of the predicted bounding boxed.

To address this, a percentile-based background filtering mechanism inspired by the interquartile range (IQR) method was applied. The points were filtered using the X-axis values of the 3D LiDAR points. Unlike the standard IQR method, we adopted a modified approach in which the lower bound was set at the 0th percentile and the upper bound at the 60th percentile. This ensured that all nearby points were retained, while points significantly further from the object, typically associated with background surfaces, were excluded.
To determine the filtering bounds, percentile spread multiplier of 0.8 was employed, analogous to how margins are computed in interquartile-based filtering. LiDAR points with X values that fell outside this extended range were classified as background noise and excluded. The multiplier of 0.8 was selected based on empirical evaluation, striking a balance between removing distant irrelevant points and preserving the essential ones.

4) Center and Rotation Optimization: Several enhancements were introduced to improve the determination of the center and orientation of the bounding boxes, especially for elongated objects such as trucks. Initially, in the baseline method the center of the bounding box was calculated based on the geometric centroid of the LiDAR points of the object. The baseline bounding box generation function aims to minimize the volume of the enclosing box, which occasionally results in orientations that are misaligned. The goal was to produce bounding boxes that more accurately reflect the object's shape and heading in the ground plane. The following procedure was used:

- **Convex Hull Calculation:** The 3D LiDAR points of each detected object were projected onto the ground plane (X-Y axes), and a convex hull was computed to approximate the object's footprint from a top-down view.
- **Furthest Points Identification and Center Estimation:** From the convex hull, the two most distant points were identified. The midpoint between these points was used as the center of the bounding box. This heuristic better represents the spatial extent of elongated objects than the geometric centroid, which may be biased by uneven point distributions.
- **Rotation Optimization:** To align the bounding box with the object's principal axis, we searched for the optimal yaw angle θ around the previously computed center. The cost function $\mathcal{L}(\theta)$ (Equation 2) was defined as the average distance between all filtered LiDAR points $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\} \subset \mathbb{R}^2$ and their nearest edge on the bounding box base $\mathcal{B}$ rotated by the angle θ. To evaluate this efficiently, a fully vectorized implementation was used to compute point-to-edge distances.

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \text{dist}\left(\mathbf{p}_i, \mathcal{B}(\theta)\right) \tag{2}$$

- **Optimization Procedure:** The optimal rotation angle was determined by local minimization of the scalar function of a variable, which performs continuous optimization of the yaw angle. This approach improved alignment accuracy for long objects and reduced computational overhead compared to brute-force search over discrete angles.

This strategy significantly improved the alignment of bounding boxes for trucks and other long vehicles, as illustrated in Figure 4. By accurately aligning the bounding box with the true orientation of the object, the model was able to produce more realistic and stable bounding box estimates.

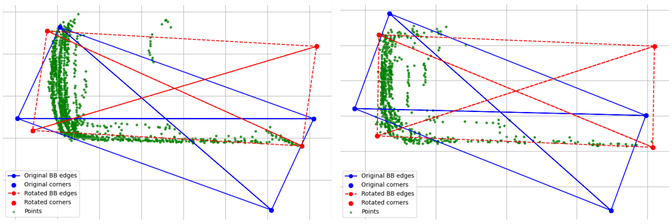

Figure 4. Demonstration of bounding boxes rotation refinement in birds-eye-view: the blue boxes show the original rotation, while the red boxes show the boxes after correction based on the distance to LiDAR points (green).

III. Evaluation

A. Dataset

For the purposes of evaluation, the View of Delft (VoD) [14] dataset was used. This dataset was captured in an urban environment and provides multimodal sensory data: high-resolution images, detailed lidar point clouds, and 4D radar measurements. The combination of these sensors makes the dataset suitable for experimenting with various sensor fusions and 3D object detection as a whole. The data was collected in various locations across the city of Delft in Netherlands, covering a wide range of scenarios such as residential areas, intersections, open spaces as well as narrow and dense streets. It contains 8600 annotated frames at 10 Hz with more than 120 000 annotations in camera FoV. Of the 13 classes provided in the dataset annotations, only cars, pedestrians, cyclists, trucks, and motorcycles were used for evaluation of described post-processing methods. The quantity of annotated bounding boxes present in the dataset can be seen in Table I

TABLE I. Number of annotated bounding boxes per object class in the dataset

	Car	**Cyclist**	**Pedestrian**	**Motorcycle**	**Truck**
Count	19,899	25,443	19,892	571	219

B. YOLO model

The pre-trained version of the YOLOv8 segmentation model was used. The model weights provided by Ultralytics [15] result from training on the Common Objects in Contex (COCO-Seg) dataset. [16]. The *small* size of the model consists of

11.8M parameters and achieves mean average precision (mAP) of 37.8% across 80 present classes.

C. Results

The impact of the proposed post-processing methods was evaluated by comparing detection performance before and after their application. Table II presents the absolute improvements in detection accuracy across five object categories (car, cyclist, pedestrian, motorcycle, truck). Evaluation metrics include mean Average Precision (mAP) at intersection over union (IoU) thresholds of 0.5 and 0.25, and mean IoU between predicted and ground truth 3D bounding boxes.

TABLE II. Absolute improvement between the baseline implementation described in Section II-A and the pipeline enriched with post-processing techniques. Values shown are in percentage points (% pts.).

Metric	**Car** (% pts.)	**Cyclist** (% pts.)	**Pedestrian** (% pts.)	**Motorcycle** (% pts.)	**Truck** (% pts.)
AP@0.5	8.06	2.10	8.19	5.78	9.59
AP@0.25	7.91	8.28	4.87	11.04	15.53
Avg IoU	5.57	2.85	3.34	4.86	6.39

The results demonstrate that post-processing techniques can substantially enhance the accuracy and consistency of 3D object detection when applied after an initial neural segmentation and point cloud fusion stage. Among all evaluated object classes, truck and motorcycle detections benefited the most from the applied enhancements.

IV. Conclusion

This work demonstrated how a set of lightweight, mathematically and geometrically driven post-processing methods can improve the performance of a YOLO-LiDAR 3D object detection pipeline without additional neural computation. The evaluation of the View of Delft dataset showed consistent performance improvements in key detection metrics, especially for object classes trucks and cyclists. The results validate the importance of geometry-aware post-processing as a complement to deep segmentation models, particularly in scenarios where inference must remain computationally undemanding.
Future work could focus on the correction of bounding box width, particularly in scenarios where the point cloud is incomplete due to occlusion. In such cases, the 3D LiDAR data often fail to capture dimensions of the object sufficiently, resulting in overly narrow or collapsed bounding boxes. This issue is especially common for cars and trucks when viewed from side angles or when partially obstructed. To address this, class-specific heuristics based on known dimensions or typical aspect ratios could be introduced as priors during post-processing stage.

Acknowledgment

This project was Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00014 and by APVV-23-0519 project “Legal and Technical Challenges of Smart Mobility to Increase Road Traffic Safety”. The authors acknowledge the use of generative AI tools for language editing and grammar refinement during the preparation of this manuscript. These tools were used to improve clarity and readability, but all content, ideas, and interpretations are solely those of the authors.

References

[1] Y. Li *et al.*, “Deepfusion: Enhancing lidar and camera fusion for 3d object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[2] X. Bai *et al.*, “Transfusion: Robust lidar-camera fusion for 3d object detection with transformers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[3] Z. Liu *et al.*, “Bevfusion: Bird’s-eye-view representation for efficient lidar and camera fusion,” *IEEE Robotics and Automation Letters*, 2022.

[4] 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.

[5] 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.

[6] M. Galinski, T. Milesich, M. Janeba, J. Danko, P. Lehoczkỳ, L. Magdolen, L. Šoltés, and A. Tomčala, “Camera and telemetry data from slovak roads in various light and weather conditions,” *Scientific Data*, vol. 11, no. 1, p. 1450, 2024.

[7] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, “V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10914–10921, 2022.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[9] M. A. Saucedo, N. Stathoulopoulos, V. Sumathy, C. Kanellakis, and G. Nikolakopoulos, “Box3d: Lightweight camera-lidar fusion for 3d object detection and localization,” in *2024 32nd Mediterranean Conference on Control and Automation (MED)*, pp. 101–106, IEEE, 2024.

[10] T. Kieffer, “Lidar-camera fusion for 3d object detection in autonomous driving systems,” Master’s thesis, University of Luxembourg, August 2024.

[11] X. Chen, Z. Zhang, M. Li, and D. Li, “Border-oriented post-processing refinement on detected vehicle bounding box for ADAS,” in *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)* (H. Yu and J. Dong, eds.), vol. 10615 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, p. 106150B, Apr. 2018.

[12] X. Chen, Z. Liu, K. Z. Luo, S. Datta, A. Polavaram, Y. Wang, Y. You, B. Li, M. Pavone, W.-L. Chao, M. Campbell, B. Hariharan, and K. Q. Weinberger, “Diffubox: Refining 3d object detection with point diffusion,” in *Advances in Neural Information Processing Systems* (A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds.), vol. 37, pp. 103681–103705, Curran Associates, Inc., 2024.

[13] Z. Li, Z. Bai, Y. Chen, X. Zhang, Y. Gu, and Y. Qiao, “Robust bounding box refinement for accurate road surface defect detection,” in *2022 5th International Conference on Data Science and Information Technology (DSIT)*, pp. 1–6, 2022.

[14] A. Pálffy *et al.*, “View of delft: A multi-sensor dataset for autonomous vehicle perception,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[15] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics yolov8,” 2023.

[16] 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 *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pp. 740–755, Springer, 2014.