LiDAR-Guided Monocular 3D Object Detection for Long-Range Railway Monitoring

Raul David Dominguez Sanchez[1,2], Xavier Diaz Ortiz[1], Xingcheng Zhou[2], Max Peter Ronecker[1,3], Michael Karner[1], Daniel Watzenig[3] and Alois Knoll[2]

Abstract— **Railway systems, particularly in Germany, require high levels of automation to address legacy infrastructure challenges and increase train traffic safely. A key component of automation is robust long-range perception, essential for early hazard detection, such as obstacles at level crossings or pedestrians on tracks. Unlike automotive systems with braking distances of 70 meters, trains require perception ranges exceeding 1 km. This paper presents an deep-learning-based approach for long-range 3D object detection tailored for autonomous trains. The method relies solely on monocular images, inspired by the Faraway-Frustum approach, and incorporates LiDAR data during training to improve depth estimation. The proposed pipeline consists of four key modules: (1) a modified YOLOv9 for 2.5D object detection, (2) a depth estimation network, and (3-4) dedicated short- and long-range 3D detection heads. Evaluations on the OSDaR23 dataset demonstrate the effectiveness of the approach in detecting objects up to 250 meters. Results highlight its potential for railway automation and outline areas for future improvement.**

I. INTRODUCTION

Railways are a key mode of transport in Europe, facilitating the movement of goods and passengers efficiently. However, the system faces capacity limits and legacy issues [1]. Initiatives such as Digital Schiene Deutschland [2], Shift2Rail [3], and safe.trAIn [4] aim to modernize railway operations and increase automation.

A crucial requirement for autonomous rail systems is a robust perception module capable of detecting objects and locating the train with high precision. This is particularly important for improving safety, as most railway accidents involve track intrusions (61%) or level crossings (24%) [5]. Furthermore, the long braking distances of trains, reaching lengths as high as 1000 meters [6], [7], necessitate early detection of obstacles, significantly beyond the typical 70-meter braking distance for cars [8].

LiDAR provides accurate depth information but is limited in range (250-500 meters), while monocular cameras capture high-resolution images but lack direct depth measurement [9]. The fusion of both sensors has shown significant improvements in 3D object detection [10], [11], [12], [13]. This work proposes a monocular 3D object detection method tailored for railway environments with the following contributions:

Contributions

1) An AI-based 3D object detection approach for autonomous railway vehicles using only monocular images, addressing long-range perception challenges.
2) A novel pipeline integrating a modified YOLOv9 for 2.5D detection with depth estimation module for accurate frustum-based 3D localization.
3) A training strategy leveraging LiDAR point clouds and 3D labels to enhance depth estimation and improve monocular-based 3D detection, reducing the gap with LiDAR-based methods.

II. RELATED WORK

The following provides a brief introduction to 3D object detection, including relevant detector families such as frustum-based methods and the YOLO family, as well as an overview of depth estimation.

A. 3D Object Detection

3D object detection involves identifying objects in a scene and estimating their enclosing 3D bounding boxes. Typically, 8 to 11 parameters are predicted, including the object class, center coordinates, dimensions, and orientation. In autonomous driving, orientation prediction is often simplified to yaw rotation, as objects are assumed to rotate only about the axis perpendicular to the road [14].

Methods for 3D object detection are categorized by:

- **Input data type**: Input data typically includes images or point clouds, which may be further processed into formats like depth maps, voxel grids, or BEV maps. Camera-based methods are divided into monocular [8], [15], stereo [16], and multi-camera [17] approaches. Multimodal methods fuse data from multiple sensors, such as combining LiDAR and monocular images, to improve performance [10], [13].
- **Detection environment**: Detection methods are tailored to either indoor or outdoor scenes. Indoor point clouds are dense and provide rich context, while outdoor point clouds are sparse and influenced by external factors. Recent work explores environment-agnostic methods [18].
- **Detection approach**: Deep learning dominates the field, though classical methods remain relevant for efficiency and simpler parameterization. Hybrid methods combine classical techniques (e.g., preprocessing) with deep learning for improved performance [19].

[1] SETLabs Research GmbH, 80687 Munich, Germany {firstname.lastname}@setlabs.de

[2] Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich, 80333 Munich, Germany firstname.lastname@tum.de

[3] Institute of Visual Computing, Graz University of Technology, 8010 Graz, Austria firstname.lastname@tugraz.at

This classification is not standardized, but it highlights the factors most relevant to this work.

B. Frustum-Based Methods for 3D-Detection

In the context of this paper, a frustum refers to a portion of a 3D point cloud that is pyramid-shaped and is used to narrow down a region of interest in a point cloud.

Frustum-based methods leverage 2D object detections to narrow the 3D search space by projecting 2D bounding boxes into 3D frustums [20]. Faraway-Frustum [21] combines monocular images and LiDAR point clouds to address the challenges of long-range 3D detection. It uses a dual-head architecture: one head employs standard methods like PV-RCNN [22] or F-PointNet [20] for short-range objects, while the second head (FF-Net) processes long-range frustums in BEV format using MobileNet [23].

FF-Net handles sparsity in long-range LiDAR data and predicts 3D bounding boxes, including yaw orientation, object center, and dimensions. Frustums are classified as short- or long-range based on their centroid's distance, with thresholds (e.g., 60 m for pedestrians, 75 m for cars) derived from the KITTI dataset [24].

By reducing search space and leveraging multimodal data, frustum-based methods improve detection accuracy and efficiency, making them suitable for autonomous driving applications.

C. The YOLO Object Detector Family and Extensions

The You Only Look Once (YOLO) family of object detectors simplifies 2D detection by combining classification and localization into a single network [25]. YOLOv1 predicts bounding boxes and class probabilities on a 7×7 grid, with non-maximal suppression filtering low-confidence predictions. Subsequent versions introduced significant improvements in accuracy, efficiency, and feature extraction.

YOLOv9 [26] replaced fixed bounding box dimensions with probabilistic distributions, eliminating non-maximum suppression through one-to-one label assignment. Architectural upgrades include Spatial Pyramid Pooling (SPP), Path Aggregation Networks (PANet), Generalized Efficient Layer Aggregation Networks (GELANs), and Programmable Gradient Information (PGI), enhancing multi-scale feature extraction and gradient flow while maintaining real-time performance [27].

YOLO has also been adapted for 2.5D tasks, integrating depth estimation. YOLO-D and Dist-YOLO extend YOLOv3 to directly predict depth in the detection head [28], [29]. Another approach overlays YOLOv5 bounding boxes on depth maps to estimate distances via polynomial regression [30]. While effective for short ranges, these methods require recalibration for new domains.

D. Depth Estimation

Monocular depth estimation converts 2D images into depth maps, enabling point cloud generation. Classical methods relied on geometric cues such as structure-from-motion and multi-view stereo, but deep learning now dominates due to its ability to infer depth from single images. Supervised approaches use ground truth depth, while self-supervised methods exploit geometric consistency. The following methods are relevant to this work. MiDAS [31] trains on multiple datasets for cross-domain generalization using zero-shot transfer. Fine-tuning is required for absolute depth estimation. DenseDepth [32] uses an encoder-decoder with a DenseNet [33] backbone. Its loss function combines L1, gradient, and SSIM terms. Sparse ground truth is addressed with "Colorization using Optimization" [34]. Marigold [35] applies generative diffusion models, treating depth estimation as a denoising process. It improves generalization but increases computational cost.

E. Railway Datasets

Public datasets for autonomous railway perception are limited compared to the automotive domain, where large-scale datasets like KITTI [24], NuScenes [36], and Waymo Open Dataset [37] dominate 3D object detection research. Existing railway datasets vary in focus, covering tasks such as 2D semantic segmentation [38], [39], object detection [40], [41], [42], and 3D perception [43], [44], [45]. However, many are either not publicly available [28], [44], lack mainline railway scenarios, or do not provide long-range annotations.

For long-range 3D object detection, datasets with labeled objects beyond 200 meters are necessary. While some datasets, such as ESRORAD [44] and LRODD [28], include multimodal data, they either prioritize road environments or remain inaccessible.

OSDaR23 [46] addresses the lack of public railway datasets by providing high-quality sensor data for 2D and 3D object detection. It includes six RGB cameras (high/low resolution), three infrared cameras, six LiDAR sensors, a 2D radar, and GNSS/IMU.

The dataset comprises 1534 annotated frames across 21 sequences and 45 subsequences. Annotations, following the ASAM OpenLABEL standard [47], include 2D bounding boxes, polylines for track recognition, 3D cuboids, and per-point semantic segmentation. It is divided into training (70%), validation (15%), and test (15%) sets. Given all of this, OSDaR23 provides the necessary data for our research objectives in long-range 3D object detection railway environments.

III. METHODOLOGY

Given the objectives established for this project, as well as the reviewed literature, the Monocular Faraway-Frustum pipeline (MFF) for 3D object detection was devised (shown in Fig. 1), which leverages LiDAR data during training to enhance object depth estimation, especially for distant objects (in this work distant objects are those beyond 100 meters). In the following subsections the different modules of the developed pipeline will be described.

A. Depth Estimation Module

As can be inferred from the name, the main inspiration for MFF is Faraway-Frustum mainly due to its emphasis on improving long-range detection. Nevertheless, since

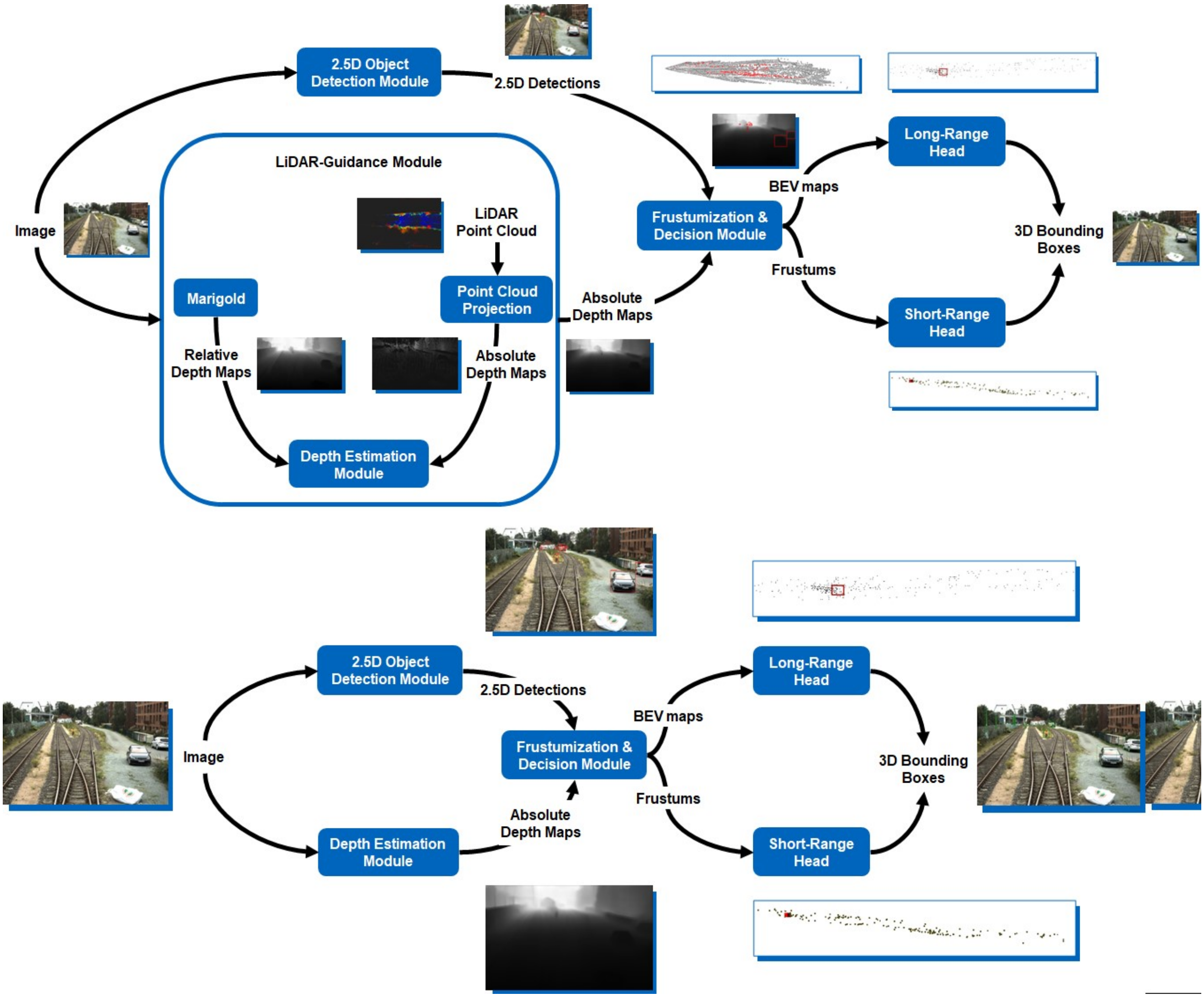

Fig. 1: MFF Pipeline. Top: Pipeline during training, when LiDAR point clouds are used to guide depth estimation. Bottom: Pipeline during testing time, in this case the method works solely using monocular images.

Faraway-Frustum is a multi-modal approach (monocular images and LiDAR point clouds) and the proposed method is monocular-based, it was decided to replace the LiDAR point cloud input with pseudo-clouds generated by back-projecting depth maps. Therefore, a monocular depth estimation module was necessary to estimate depth maps. In monocular depth estimation research, diffusion-based approaches (e.g. Marigold) are currently the state-of-the-art. However, their main disadvantage is their inability of running in real-time. Furthermore, works like MiDAS emphasize the utilization of diverse and large sets of data for training as well as to initially train for relative depth estimation and then fine-tune for absolute depth estimation. As previously mentioned, the mainly utilized dataset in this work was OSDaR23, which does not provide ground truths for relative or absolute depth estimation (the latter can be computed from point clouds). Consequently, to take advantage of the SOTA for depth estimation, to be capable of running the module in real-time, and follow the ”relative then absolute” approach, first, given monocular images, from OSDaR23 and RailSem19 [38] (cropped to have the same ratio as OSDaR23), Marigold is used as the teacher network to guide the outputs of a modified version of DenseDepth. The modifications include a change in the depth loss from L1 to Huber, and a switch in the upsampling method from bilinear interpolation to transposed convolutions. After successful distillation the second step is to train a simple convolutional network that was designated as refinement network. What this network does is map the relative depth to absolute depth. Additionally, in this case the ground truths were inpainted depth maps (using the optimization method described in [34]) generated from OSDaR23’s point clouds.

B. 2.5D Object Detection Module

In Faraway-Frustum a 2D object detector is used to identify regions of interest in point clouds (frustums). Based on the centroid of a frustum, it is decided whether this cloud should be processed with a short-range detection head or a long-range detection head. Even though the same steps could be followed in MFF, due to the fact that pseudo-clouds are not as accurate as LiDAR point clouds and that they tend to include more noise, it was decided to do a weighted sum instead. The summed components are a frustum's centroid and an estimation of the distance of the object in such frustum. To do so, our method replaces the 2D object detector with a 2.5D object detector, that is a 2D object detector that also estimates the distance to the camera for the predicted objects. Works like [28], [29] modified YOLO to also predict distance, based on such works the following modifications were applied to YOLOv9 (the latest version of the YOLO detectors at the time) to create a 2.5D object detection module:

- **Distance Head**: Besides the box regression head and the class head an additional head was added comprised of a simple fully convolutional network and producing normalized values which would represent distances between 0 and 250 meters.
- **Huber Loss**: The usually used losses for distance estimation are either an L1 loss or L2 loss. When used individually, the losses did not produce estimations as accurate as expected and tended to overfit to the distance range with the most examples. It was decided to combine them using Huber Loss which resulted in a better performance.
- **Multiple Datasets**: Since OSDaR23 is a relatively small dataset, it was difficult to reach a good performance using solely this dataset. In works like [28], they integrate data from KITTI to boost the performance of the 2.5D object detector. Therefore, a similar approach was followed by training this module with OSDaR23 and KITTI. Additionally, KITTI+OSDaR23 training was used as a pre-training step, after which the model was fine-tuned only using OSDaR23.

C. Frustumization & Decision Module

The next block in the pipeline is the frustumization and decision block, which takes as input a depth map and 2.5D object predictions, based on these it generates frustums and estimates the center of the frustums (using the aforementioned weighted sum), to then transform these clouds from the LiDAR frame to the frustum frame and feed them to the corresponding head. In the case of the long-range head, the frustums are splatted into Bird's Eye View (BEV) representation before being fed to the head together with their respective class priors.

D. Detection Heads

Similarly to Faraway-Frustum, the long-range head in MFF is comprised of a feature encoder and fully connected layers. However, in contrast with the original implementation, MFF utilized various sets of fully connected layers to increase the capacity of the module to represent more complex functions and better detect the classes of interest. On the other hand, for the short-range head and with the objective of leveraging SOTA point-cloud-based 3D object detectors (since they are usually developed to work in a short-range detection environment), Faraway-Frustum utilizes PV-RCNN, with that in mind, it was decided to evaluate the performance of the pipeline with different heads, being these: PointRCNN, PointPillars, and Part-A^2. Both modules were only trained with the OSDaR23 dataset.

IV. Evaluation

Since each module was trained separately, we first evaluate their individual performance before assessing their integration into the full pipeline. Individual evaluations use the OSDaR23 validation set, while full pipeline evaluations use its test split, ensuring design choices are guided by performance gains while maintaining independent assessment.

The detected classes include person, road vehicle, buffer stop, catenary pole, and signal pole. Fig. 2 illustrates the distribution of classes in the dataset, showing the number of points inside each 3D bounding box instance in the clouds vs. the distance from the object across different splits

The distributions of the classes (number of points within the 3D bounding box of an object vs. x-axis distance from the camera to that same object) across the splits are shown in Fig. 2. Analyzing these distributions was essential for understanding the model's behavior and performance. Furthermore, the dataset labels were filtered to include only those with corresponding 2D-3D pairs, as these were the only ones relevant to this study.

A. Individual Module Evaluation

1) 2.5D Object Detection Module: The validation results of the 2.5D object detection model are shown in Table I. Detection quality was measured using mean average precision at an IoU threshold of 0.5 (mAP@0.5), while mean absolute error (MAE) evaluated distance prediction accuracy. The best-performing classes were person and road vehicle, likely due to their higher representation in training, particularly from KITTI pre-training. In contrast, poles were harder to detect due to their thin geometry, leading to lower mAP. MAE initially appears low within 0–250 meters (full range), but error analysis across distance ranges shows a significant increase at greater distances. This can be attributed to an imbalance in label distribution, with more short-range than long-range samples, causing the model to overfit to closer distances. Additionally, KITTI validation samples were included during pre-training, improving performance at shorter ranges.

To validate the implementation, the module was compared against Dist-YOLO [29] and YOLOv9, retrained using only KITTI (2D detections only). As shown in Table II, the models exhibit comparable performance. The 2D object detection results align closely with the original implementation.

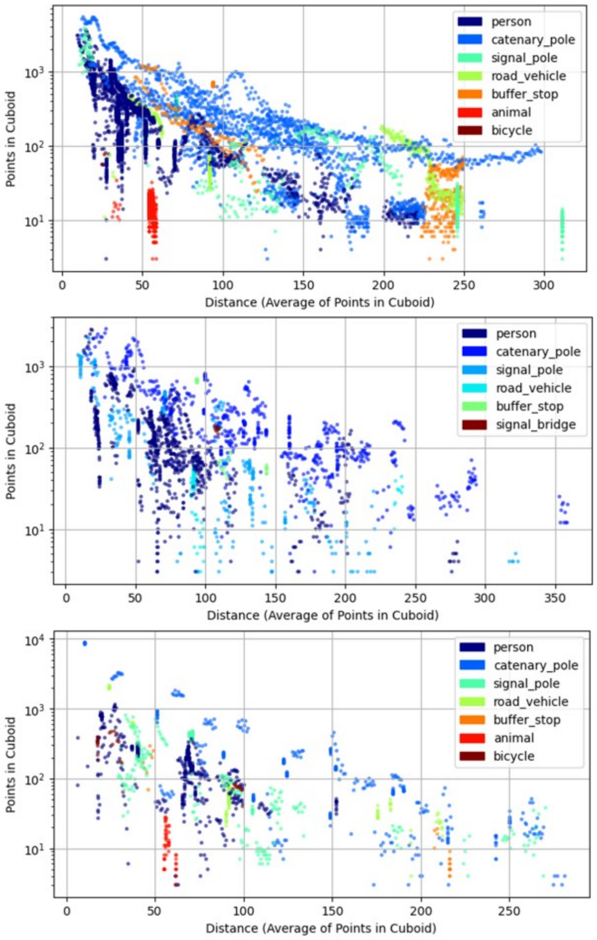

Fig. 2: Number of point cloud points within the 3D bounding box of an object, plotted against the x-axis distance from the camera to that same object. Top: Training split, Middle: Validation split, Bottom: Test split.

However, the slightly higher MAE of our module compared to Dist-YOLO is likely due to differences in dataset splits. While YOLOv9 and our module were trained using the frequently used train/val splits mentioned in [48], Dist-YOLO employed a custom split with most data allocated for training and only a few frames for validation.

2) Depth Estimation Module: The depth estimation module achieved a relative depth estimation error of 0.0335, demonstrating accurate results and successful distillation (Fig. 3). During fine-tuning for absolute depth estimation, a MAE of 11.51 meters was obtained. This is notable given the diverse depth range of inpainted OSDaR23 depth maps, which extended up to 500 m, requiring significant adaptation by the module (Fig. 4).

TABLE I: Validation results of the 2.5D object detection module. P: pre-training, F: fine-tuning, K: KITTI, O: OSDaR23, A: 0-50, B: 50-100, C: 100-150, D: 150-200, E: 200-250, FR: Full Range.

Stage	Dataset	mAP@0.5	MAE ranges (m)					
			A	B	C	D	E	FR
P	K, O	52.0	9.09	19.37	23.78	77.41	136.68	15.09
F	K	50.0	9.99	10.08	18.33	49.21	75.37	17.10

TABLE II: Performance comparison between YOLOv9, Dist-YOLO, and the 2.5D object detection module from MFF on the KITTI validation dataset.

Model	mAP@0.5	MAE (m)
YOLOv9 (retrained)	89.9	-
Dist-YOLO (W) [29]	77.1	10.22
MFF (Ours)	86.3	14.18

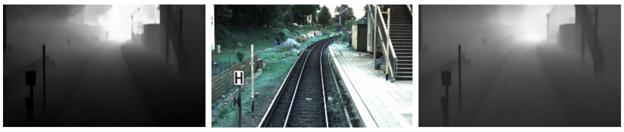

Fig. 3: Depth estimation module prediction example (relative depth). Left: Ground truth, Center: Input, and Right: Prediction.

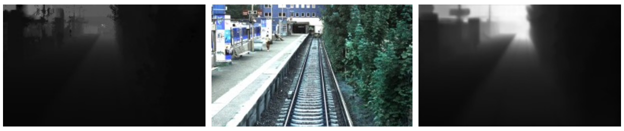

Fig. 4: Depth estimation module prediction example (absolute depth). Left: Ground truth, Center: Input, and Right: Prediction.

3) Long Range Head: To train the long-range head, only OSDaR23 data was used. Using OSDaR23-based depth maps and ground truth (GT) 2D object labels, frustums were generated, filtered by object distance, and then projected into BEV maps. As shown in Table III, the best-detected class was buffer stop, likely due to the limited number of examples, which were highly similar between the training and validation sets. The next best-performing classes were person and catenary pole, which had the most training samples and were well-distributed across the long-range, improving detection robustness. In contrast, signal poles, similar to the 2.5D detection case, were difficult to detect due to their small and thin structure. Notably, road vehicles in the validation set appeared only in the 200–250 m range, but the head failed to detect them. The highest MAE in long-range detection was 6.88 m, which is reasonable given detections extended up to 250 m.

To validate the module's implementation, it was tested on the KITTI dataset (Car class) and compared to its original implementation. As shown in Table IV, the module outperforms the original long-range head, primarily due to the increased capacity of the long-range head in MFF.

4) Short Range Head: The short-range head was evaluated using three LiDAR-based 3D object detectors: PointR-CNN, PointPillars, and Part-A^2. These models were trained on short-range frustums from the OSDaR23 dataset, with

TABLE III: Validation results of the long-range head. MAE is computed at different ranges. N/A: Not applicable because there where no objects of that class at that range, N/D: No detections even when there were objects of that class in a specific range.

Class	mAP@0.1		MAE (m)		
	BEV	3D	100-150	150-200	200-250
Person	45.14	41.07	1.11	0.96	1.05
Road Vehicle	N/D	N/D	N/A	N/A	N/D
Buffer Stop	99.8	99.1	0.65	N/A	N/A
Catenary Pole	50.86	49.13	2.22	2.85	5.27
Signal Pole	2.79	1.18	3.91	5.05	6.88

results shown in Table V. All models exhibited similar behavior. Buffer stop, road vehicle, and catenary pole achieved high detection performance due to their large, distinguishable geometry. In contrast, person and signal pole were consistently difficult to detect, likely due to noise from inpainted depth maps and the small size of these objects. Among the evaluated models, PointPillars achieved the best overall performance and was integrated into the full pipeline.

B. Full Pipeline Evaluation

After assembling the full pipeline, it was evaluated on the OSDaR23 test set. The 2.5D object detection module achieved the results shown in Table VI, while the depth estimation module reached an MAE of 12.85 m. Performance closely matches validation results, which is expected given the significant overlap between training and test sequences. As previously noted, road vehicle and person, the most frequent classes, performed best, suggesting that a larger dataset could improve performance for other classes. The buffer stop MAE is the main concern, as its distribution differs between validation and test sets. In validation, buffer stops are mostly within 150 m, whereas in testing, they extend beyond 200 m, leading to higher errors.

Although the depth estimation module maintained similar performance, analyzing the absolute per-pixel difference between predicted and ground truth depth maps (Fig. 5) reveals key insights. At short range, scene structures are consistent, enabling accurate depth estimation. However, at long range, scene variations increase—some extend up to 300 m, others to 500 m—making depth estimation more challenging due to these dynamic changes.

TABLE IV: Performance comparison between the long-range head utilized in Faraway-Frustum and MFF. The utlized dataset was KITTI and the evaluated class was Car.

Method	3DmAP@0.1
Faraway-Frustum [21]	46.90
MFF (ours)	60.61

TABLE V: Validation results of different LiDAR-based 3D object detectors with OSDaR23. Based on its performance, the ★ detector was chosen to be utilized in the full pipeline.

mAP@0.5	PointRCNN	PointPillars ★	Part-A^2
BEV	23.624	31.51	23.89
3D	21.02	27.454	21.55

TABLE VI: Performance of the 2.5D object detection module on the OSDaR23 test set.

Class	mAP@0.5	MAE (m)
Person	76.72	10.02
Road Vehicle	96.81	26.01
Buffer Stop	26.95	119.75
Catenary Pole	44.43	4.69
Signal Pole	58.76	4.77
All	60.73	15.76

It can be concluded that the error carried from the 2.5D predictions and depth maps can end up affecting the quality of the frustums, which at the same time ends up hindering 3D object predictions. Nevertheless, it has been shown that the pipeline works and is promising since the results obtained from individual tests demonstrate that with better distance estimations as well as less noisy and more accurate depth maps, the pipeline would be capable of producing accurate results.

As shown in Fig. 6, the full pipeline successfully produces 3D object detections. The top image demonstrates accurate detections of persons, vehicles, and a pole. However, in the bottom image, while both short- and long-range detections are present, some poles remain undetected. This occurs because the first image contains classes that the 2.5D detection module handles well, whereas the second image consists of objects the module struggles to detect.

These results indicate that errors from 2.5D predictions and depth maps degrade frustum quality, ultimately hindering 3D object predictions. However, the pipeline remains effective, and improving depth estimation and reducing noise would potentially further enhance its performance.

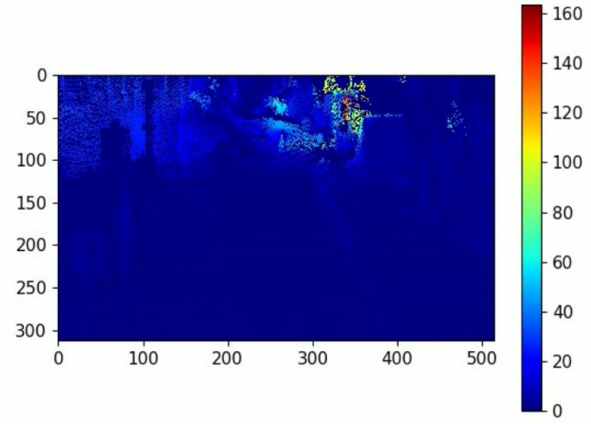

Fig. 5: Heatmap exemplifying the error distribution in the predicted depth maps. Error increases considerably in very distant regions.

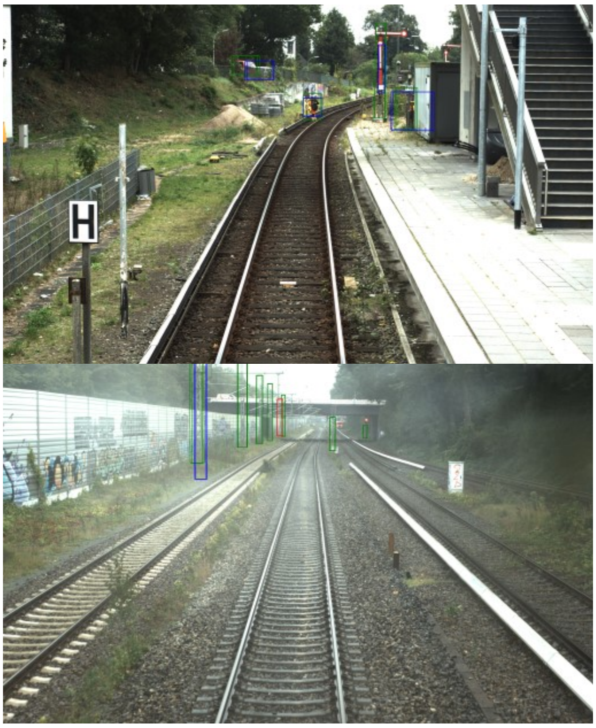

Fig. 6: Examples of detections made by MFF, green boxes represent GTs,blue boxes represent short-range 3D object predictions and red boxes represent long-range 3D object predictions.

V. CONCLUSIONS

This work explored the challenges and advancements in achieving fully autonomous trains, with a focus on long-range 3D object detection. The focus on three main objectives: developing an AI-based 3D object detection system using only monocular images, enabling long-range detection, and improving accuracy through LiDAR guidance. It introduces a modular detection method combining a 2.5D object detector, depth estimation, and a decision mechanism for short- and long-range detection. Experimental results show that while the approach is effective, limitations in data distribution, depth accuracy, and backbone module performance affect precision. Future work suggests real-time performance optimization, end-to-end training, synthetic data integration, noise reduction in depth maps, and testing other 2.5D object detection models to enhance accuracy and robustness. Despite these challenges, the approach establishes a solid foundation for future research and contributes to advancing autonomous railway perception.

ACKNOWLEDGMENT

This work has received funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant agreement 19I21039A.

REFERENCES

[1] C. Borghini, “Trains: The backbone of mobility,” *Pais Positivo*, 2021, originally published in Portuguese in the April edition of the Pais Positivo magazine, issue no 144.

[2] “What is digitale schiene deutschland?” https://digitale-schiene-deutschland.de/en/What-is-Digitale-Schiene-Deutschland.

[3] “About shift2rail,” https://rail-research.europa.eu/about-shift2rail/.

[4] safe.trAIn, “safe.train: Rethinking mobility.” [Online]. Available: https://safetrain-projekt.de/en/

[5] (2023) Railway safety statistics in the EU. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Railway_safety_statistics_in_the_EU

[6] D. Ristić-Durrant, M. Franke, and K. Michels, “A Review of Vision-Based On-Board Obstacle Detection and Distance Estimation in Railways,” *Sensors*, vol. 21, no. 10, p. 3452, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/10/3452

[7] E. H. Assaf, C. von Einem, C. Cadena, R. Siegwart, and F. Tschopp, “High-Precision Low-Cost Gimballing Platform for Long-Range Railway Obstacle Detection,” *Sensors*, vol. 22, no. 2, p. 474, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/2/474

[8] Z. Yang, Z. Yu, C. Choy, R. Wang, A. Anandkumar, and J. M. Alvarez, “Improving distant 3d object detection using 2d box supervision,” 2024. [Online]. Available: https://arxiv.org/abs/2403.09230

[9] Y. Wang, “Lidar on its way out? camera’s market size from 76% to 79% by 2033,” *IDTechEx research*, 2023. [Online]. Available: https://www.idtechex.com/en/research-article/lidar-on-its-way-out-cameras-market-size-from-76-to-79-by-2033/28865

[10] H. Wu, C. Wen, S. Shi, X. Li, and C. Wang, “Virtual Sparse Convolution for Multimodal 3D Object Detection,” in *CVPR 2023*, 2023, pp. 21 653–21 662.

[11] X. Li, T. Ma, Y. Hou, B. Shi, Y. Yang, Y. Liu, X. Wu, Q. Chen, Y. Li, Y. Qiao, and L. He, “LoGoNet: Towards Accurate 3D Object Detection With Local-to-Global Cross-Modal Fusion,” in *CVPR 2023*, 2023, pp. 17 524–17 534.

[12] Y. Li, X. Qi, Y. Chen, L. Wang, Z. Li, J. Sun, and J. Jia, “Voxel Field Fusion for 3D Object Detection,” in *CVPR 2022*, 2022, pp. 1120–1129.

[13] X. Li, B. Shi, Y. Hou, X. Wu, T. Ma, Y. Li, and L. He, “Homogeneous Multi-modal Feature Fusion and Interaction for 3D Object Detection,” in *ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 691–707.

[14] I. Oleksiienko and A. Iosifidis, “Chapter 13 - 3D object detection and tracking,” in *Deep Learning for Robot Perception and Cognition*, A. Iosifidis and A. Tefas, Eds. Academic Press, 2022, pp. 313–340.

[15] R. Zhang, H. Qiu, T. Wang, Z. Guo, Z. Cui, Y. Qiao, H. Li, and P. Gao, “MonoDETR: Depth-guided Transformer for Monocular 3D Object Detection,” in *ICCV 2023*, 2023, pp. 9155–9166.

[16] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving,” Feb. 2020.

[17] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, “BEVDepth: Acquisition of Reliable Depth for Multi-View 3D Object Detection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, pp. 1477–1485, June 2023.

[18] Z. Li, X. Xu, S. Lim, and H. Zhao, “Towards unified 3d object detection via algorithm and data unification,” 2024. [Online]. Available: https://arxiv.org/abs/2402.18573

[19] M. Mahmeen, R. D. Dominguez Sanchez, M. Friebe, M. Pech, and S. Haider, “Collision avoidance route planning for autonomous medical devices using multiple depth cameras,” *IEEE Access*, vol. 10, pp. 29 903–29 915, 2022.

[20] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum PointNets for 3D Object Detection From RGB-D Data,” in *CVPR 2018*, 2018, pp. 918–927.

[21] H. Zhang, D. Yang, E. Yurtsever, K. A. Redmill, and Ü. Özgüner, “Faraway-Frustum: Dealing with Lidar Sparsity for 3D Object Detection using Fusion,” Mar. 2021.

[22] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, “Pv-rcnn: Point-voxel feature set abstraction for 3d object detection,” 2021. [Online]. Available: https://arxiv.org/abs/1912.13192

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

CONFIDENTIAL. Limited circulation. For review only.

[24] 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.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[26] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," 2024. [Online]. Available: https://arxiv.org/abs/2402.13616

[27] C.-Y. Wang and H.-Y. M. Liao, "Yolov1 to yolov10: The fastest and most accurate real-time object detection systems," 2024. [Online]. Available: https://arxiv.org/abs/2408.09332

[28] M. A. Haseeb, "Machine learning techniques for autonomous multi-sensor long-range environmental perception system," Ph.D. dissertation, Universität Bremen, 2021.

[29] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-yolo: Fast object detection with distance estimation," *Applied Sciences*, vol. 12, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1354

[30] A. Masoumian, D. G. Marei, S. Abdulwahab, J. Cristiano, D. Puig, and H. A. Rashwan, *Absolute Distance Prediction Based on Deep Learning Object Detection and Monocular Depth Estimation Models*. IOS Press, Oct. 2021. [Online]. Available: http://dx.doi.org/10.3233/FAIA210151

[31] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2020. [Online]. Available: https://arxiv.org/abs/1907.01341

[32] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv e-prints*, vol. abs/1812.11941, 2018. [Online]. Available: https://arxiv.org/abs/1812.11941

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018. [Online]. Available: https://arxiv.org/abs/1608.06993

[34] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 689–694.

[35] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, "Repurposing diffusion-based image generators for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[36] 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," in *CVPR*, 2020.

[37] J. Mei, A. Z. Zhu, X. Yan, H. Yan, S. Qiao, Y. Zhu, L.-C. Chen, H. Kretzschmar, and D. Anguelov, "Waymo open dataset: Panoramic video panoptic segmentation," 2022. [Online]. Available: https://arxiv.org/abs/2206.07704

[38] O. Zendel, M. Murschitz, M. Zeilinger, D. Steininger, S. Abbasi, and C. Beleznai, "RailSem19: A Dataset for Semantic Rail Scene Understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[39] Y. Wang, L. Wang, Y. H. Hu, and J. Qiu, "Railnet: A segmentation network for railroad detection," *IEEE Access*, vol. 7, pp. 143 772–143 779, 2019.

[40] J. Harb, N. Rébéna, R. Chosidow, G. Roblin, R. Potarusov, and H. Hajri, "Frsign: A large-scale traffic light dataset for autonomous trains," 2020. [Online]. Available: https://arxiv.org/abs/2002.05665

[41] L. Guan, L. Jia, Z. Xie, and C. Yin, "A lightweight framework for obstacle detection in the railway image based on fast region proposal and improved yolo-tiny network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–16, 2022.

[42] A. Zouaoui, A. Mahtani, M. A. Hadded, S. Ambellouis, J. Boonaert, and H. Wannous, "Railset: A unique dataset for railway anomaly detection," in *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS)*, vol. Five, 2022, pp. 1–6.

[43] S. Liu, C. Li, T. Yuwen, Z. Wan, and Y. Luo, "A lightweight lidar-camera sensing method of obstacles detection and classification for autonomous rail rapid transit," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23 043–23 058, 2022.

[44] R. Khemmar, A. Mauri, C. Dulompont, J. Gajula, V. Vauchey, M. Haddad, and R. Boutteau, "Road and Railway Smart Mobility: A High-Definition Ground Truth Hybrid Dataset," *Sensors*, vol. 22, no. 10, p. 3922, Jan. 2022.

[45] M. Neri and F. Battisti, "3D Object Detection on Synthetic Point Clouds for Railway Applications," in *2022 10th European Workshop on Visual Information Processing (EUVIP)*, 2022, pp. 1–6.

[46] R. Tilly, P. Neumaier, K. Schwalbe, P. Klasek, R. Tagiew, P. Denzler, T. Klockau, M. Boekhoff, and M. Köppel, "Open sensor data for rail 2023," 2023. [Online]. Available: https://data.fid-move.de/dataset/3d7e7406-639f-49f6-bbca-caac511b4032

[47] Advanced Data Controls Corp., Annotell AB, Ansys Inc., Deepen AI, Deutsches Zentrum für Luft- und Raumfahrt e. V., Five, iASYS Technology Solutions Pvt. Ltd., LiangDao GmbH, Peak Solution GmbH, SAIC Motor Corporation Ltd., Tata Consultancy Services Pvt. Ltd, understandAI GmbH, Vicomtech, WMG University of Warwick, "Asam openlabel," Associationation for Standardization of Automation and Measuring Systems, Standard 1.0.0, 2021. [Online]. Available: https://www.asam.net/standards/detail/openlabel/

[48] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6526–6534.