

A Multi-Modal Perception Pipeline for Object Detection and Tracking in Autonomous Racing

Davide Malvezzi¹, Michele Pestarino¹, Vittoria Cavicchioli², Valentina La Gamba²,
Silvia Severi², Fabio Bagni², Luca Bartoli², Massimiliano Bosi², Francesco Gatti²,
Micaela Verucchi², Ayoub Raji¹, Marko Bertogna^{1,2}

Abstract—Object detection and tracking are fundamental components of perception systems for autonomous driving. Achieving robust performance under adverse conditions such as limited visibility, sensor noise, and failures remains an open challenge, particularly in autonomous racing, where vehicles operate at very high speeds, experience strong vibrations, and interact under small safety margins. This paper presents a multi-modal late-fusion perception pipeline for object detection and tracking in the autonomous racing domain. The proposed system extends previous work by exploiting all onboard sensors through a late-fusion approach and a dedicated multi-object tracking framework. Independent detections from cameras, LiDARs, and RADARs are combined to provide timely and robust state estimates of surrounding vehicles. The tracking method explicitly compensates for detection delays and embeds in its model prior knowledge of vehicle dynamics and track layout. Experimental evaluation on real-world data across diverse critical scenarios, representative of challenging edge cases also in urban driving, confirms the effectiveness of the proposed pipeline and its suitability to support safe and adaptive planning decisions.

I. INTRODUCTION

Autonomous racing represents one of the most extreme operational domains for perception systems. At speeds approaching 80 m/s, even small delays can translate into meter-scale state estimation errors: a latency of 100 ms corresponds to nearly 8 meters of spatial displacement. In such conditions, asynchronous multi-sensor measurements, detection delays, and high relative velocities significantly amplify tracking inaccuracies and directly impact collision avoidance and overtaking safety. The latest research trends focus on reducing inference time, improving Bird-Eyed-View (BEV) representations and operators' performance, and achieving more accurate depth estimation from cameras [1]. Another important goal in this domain is to achieve robust performance under adverse weather conditions, limited visibility, and sensor failures. These are scenarios highly relevant in the motor-sport domain, and in this regard, the increase in research on Autonomous Racing can be exploited as an experimental and learning field for autonomous driving [2].

¹University of Modena and Reggio Emilia, Modena, Italy {davide.malvezzi, michele.pestarino, ayoub.raji, marko.bertogna}@unimore.it

²HiPeRT Srl, Modena, Italy {vittoria.cavicchioli, valentina.lagamba, silvia.severi, fabio.bagni, luca.bartoli, massimiliano.bosi, francesco.gatti, micaela.verucchi, marko.bertogna}@hipert.it



Fig. 1. UNIMORE Racing's Dallara EAV-24 competing in the 2025 A2RL at the Yas Marina F1 Circuit. Left: detections from four cameras (pink). Right: Bird's-Eye View showing LiDAR detection (blue) and RADAR detections (green), together with the estimated distance and velocity of the opponent.

In particular, the birth of recent international competitions, such as the Indy Autonomous Challenge (IAC) in 2021 and the Abu Dhabi Autonomous Racing League (A2RL) in 2024, helped to deploy and test the state-of-the-art solutions in full-scale autonomous race cars running at speeds up to 80 m/s, with a high differential velocity between different agents, high vibration and noise that stress the sensors, and in conditions where latency and delay are extremely important. In this paper, we present a perception pipeline for object detection and tracking in the autonomous racing domain, deployed by the UNIMORE Racing team during the 2025 A2RL event at the Yas Marina Circuit (Fig. 1). The proposed solution takes advantage of all sensors onboard through a late fusion approach and an ad-hoc object tracking process that explicitly accounts for delay compensation. Section II provides an extensive overview of the state-of-the-art in perception-related research, highlighting the existing gaps in the literature that this work aims to address. Section III describes the hardware platform of the experimental vehicle and presents an overview of the proposed perception pipeline. The detection processes adopted for each sensor modality are detailed in Section IV, while Section V explains how the detections of different sensors are fused together. The object tracking methodology is presented in Section VI. Section VII reports the experimental evaluation, discussing both the performance and limitations of the proposed approach in three different scenarios typical of racing competitions. Finally, Section VIII concludes the paper and outlines directions for future work.

II. RELATED WORK

The literature on perception in autonomous racing can be broadly categorized into three categories: creation of multi-modal datasets, single-modality detection pipelines, and partial multi-sensor fusion approaches. In [3], the first-ever open dataset for high-speed autonomous racing is presented with more than 6.5 hours of data in ROS2 and nuScenes format, with some benchmarks on object detection and tracking. In particular, the authors compared the average precision of LiDAR detections using BEV labels and 3D bounding boxes with PointPillars [4] and Voxel-R-CNN [5], achieving acceptable precision at a maximum range of 60 m. A brief presentation of RADAR-based detection and tracking is given, focusing on noise reduction and filtering, while for the cameras, a solution based on YOLOv5 is evaluated, with detections achieving distances over 100 meters. Despite the fact that all available sensors have been tested for the detection task, no sensor fusion or multi-modal detection approaches are presented. Another dataset is presented in [6], which focuses on providing annotated multi-camera images for detection on road courses, in particular, lane detection, proposing a Generative Adversarial Network (GAN) method and comparing it to current state-of-the-art solutions. The most recent dataset in the autonomous racing domain is BETTY [7], a large-scale, multi-modal dataset collected from multiple autonomous racing vehicles. It targets a variety of autonomous driving tasks and comprises more than 13 hours of recorded data. However, no benchmark of the state-of-the-art method for object detection and tracking is provided on the presented data. The most relevant works on object detection and tracking for autonomous racing are presented by researchers who took part in the IAC and A2RL events. Team PoliMOVE presented a solution based on LiDAR sensors in which a tracking-by-detecting approach is used, following classical algorithms for point cloud segmentation with ground removal and clustering, pose estimation exploiting knowledge of track layout, and object tracking via a variable-step Extended Kalman Filter (EKF) [8]. In [9], team KAIST described their solution based on LiDAR clustering, with ground filtering on the 3D point clouds via height distribution, fused and tracked, including the RADARs detections via an Interacting Multiple Model Filter employing three models: constant velocity, constant turn rate and velocity, and random motion. Their results showed the best accuracy and recall using LiDAR-only clustering and tracking in a detection range lower than 50 meters, while in a mid-range distance (50-100 m), exploiting both sensors brought better performance, and finally, at distances greater than 100 m only the RADAR-based detections were valuable with a low recall due to the limited range of the LiDAR sensors. Another work from the KAIST team presents MonoDINO-DETR, a monocular 3D object detection framework that employs a Vision Foundation Model (DINOv2) as the backbone to extract global features for depth estimation [10]. The reported results demonstrate competitive performance. However, inference time may limit its applicability in strict real-time

scenarios, and the relatively small size of the training and validation datasets suggests that further evaluation on larger benchmarks would be beneficial to fully assess its generalization capability. The object tracking method presented by the TUM Autonomous Motorsport team in [11] is also based on the fusion of LiDAR and RADAR detections. Particular emphasis is placed on compensating for sensor-specific detection delays, as at high speeds, the apparent position of an object can shift by several meters if such delays are not accounted for. In addition, a plausibility check is applied to filter out detections that lie outside the drivable area of the track. Finally, Team MIT-PITT presented their autonomous racing stack in [12]. Their perception module relies solely on cameras and LiDAR, with RADAR sensors not being utilized. YOLOv5 is used to detect race cars in camera images, and the 3D object pose is recovered by exploiting prior knowledge of the shape and size of the vehicles. However, the pose estimation error increases proportionally with the real-world distance between the camera and the target vehicle. For LiDAR data, PointPillars is used to perform object detection on point clouds. Object association and tracking are then carried out using AB3DMOT [13]. Notably, no multi-modal sensor fusion is applied prior to object association and tracking. In these works, not all available sensor modalities have been exploited simultaneously, due to factors such as computational limitations, slow overall pipeline execution relative to the demands of high-speed racing, and elevated false positive rates. In addition, critical components such as delay compensation have been implemented only by team TUM, and none of the existing tracking methods incorporate racing-specific knowledge. To address these gaps, we propose a unified, delay-aware, multi-modal perception architecture that integrates cameras, LiDARs, and RADARs within a consistent tracking framework explicitly designed for extreme-speed racing scenarios. We validate the proposed approach through experimental evaluations that highlight both its performance and limitations in three scenarios representative of typical racing conditions.

III. SYSTEM OVERVIEW

The Dallara EAV-24 is equipped with a comprehensive array of sensors, including a Vectornav VN-310 dual-antenna GNSS units and inertial measurement solutions. In addition to these, the vehicle features three Innovusion Falcon K LiDAR sensors, seven Sony IMX728 cameras, and four ZF ProWave RADAR units. The sensor setup on the vehicle is shown in Fig. 2.

The object tracking pipeline combines detections from individual sensors using a late fusion approach. Late fusion enables the integration of heterogeneous sensor modalities, providing increased flexibility. Changes can be introduced at the fusion stage without requiring a redesign of the entire system. Moreover, late fusion improves the robustness of the overall pipeline, as in the event of a sensor failure, the remaining sensors can continue to provide object detections, thereby ensuring continuity of the tracking. However, improper temporal synchronization or spatial alignment of

the data sources in the late fusion pipeline can lead to inconsistencies in the final detection results, causing incorrect detection associations. Therefore, all sensors are temporally synchronized using the Precision Time Protocol (PTP) [14], sharing the clock with the main computation system. Furthermore, an offline multi-sensor calibration procedure is carried out to accurately estimate the relative poses of all sensors mounted on the race car. First, the intrinsic parameters of the camera and the distortion coefficients of the lens are obtained using a checkerboard-based calibration method [15]. Subsequently, the extrinsic pose between cameras and LiDARs with overlapping field-of-view is computed using an ArUco-based calibration approach [16]. Finally, the position and orientation of the RADARs are estimated using a velocity-based calibration following the method proposed by [17].

A schematic representation of the proposed object detection and tracking pipeline is shown in Fig. 3. The process begins by extracting object detections from each sensor modality, with a dedicated detection method applied to each. In Fig. 1, examples of the resulting detections are illustrated. Using the sensor calibration information, the fusion module associates detections from sensors with overlapping fields of view, ensuring that observations of the same physical object are correctly matched. The fused detections are then used to update a multi-object tracking module, which continuously estimates the position and velocity of each object. Finally, the tracked objects are provided as input to the planning module, which uses the estimated states to plan overtaking maneuvers [18].

IV. DETECTIONS

A. Camera Detections

Camera-based object detection is performed using the YOLOv4 neural network [19]. The model is initially pre-trained on the open-source DeepDrive BDD100k dataset [20] to recognize road objects such as cars, bikes, and pedestrians. The model is then specifically fine-tuned using a custom dataset, which includes data collected from online

videos of human-driven open-wheel race cars and nearly 600 manually labeled images from previous test sessions. During the inference process, the model receives a batch of N_{cameras} images and for each camera j outputs a set of 2D detections ($BB_{\text{cam}_j}^{2D}$). Each detection $bb_{ji} \in BB_{\text{cam}_j}^{2D}$ is a bounding-box represented as a tuple $(x_{ji}, y_{ji}, w_{ji}, h_{ji})$, where (x_{ji}, y_{ji}) denotes the coordinates of the top-left corner and (w_{ji}, h_{ji}) represent the width and height of the bounding-box, all in pixel coordinates relative to the corresponding image. The detection timestamp is set to the same value as the timestamp of the image in which the detection occurred. The inference process is optimized using tkDNN [21], a custom framework designed to enhance performance on NVIDIA GPUs using TensorRT technology. This optimization enables the model to achieve an average inference time of 4.5 ms for a batch of 7 input images on a NVIDIA RTX 6000 Ada GPU.

B. LiDAR Detections

LiDAR object detection is performed using the PointPillars model [4], which is initially pre-trained on the KITTI dataset [22] with the car label only. To adapt the model to specific race scenarios, a custom dataset has been created by manually labeling nearly 25000 LiDAR point clouds from previous race events. During inference, the model receives a single point cloud composed of three LiDAR scans merged together with the respective extrinsic parameters and outputs a set of 3D detections (BB_{lid}^{3D}). Each detection i is a 3D bounding-box represented as a tuple (p_i, θ_i, s_i) , where p_i denotes the 3D position of the bounding-box within the point cloud, θ_i represents its orientation relative to the z-axis, and s_i defines the dimensions of the box. Since LiDAR points are detected at different time instances due to the sensor's operating mode, the detection timestamp is calculated as the average of all contributing LiDAR points, which yields a more precise temporal reference for the tracker's delay-compensation mechanism. The inference process is optimized using tkDNN [21], enabling the model to achieve an average inference time of 8.4 ms on an NVIDIA RTX 6000 Ada GPU.

C. RADAR Detections

RADAR-based object detection is performed using a filtering and clustering approach. First, the RADAR point cloud is filtered by evaluating the RADAR Cross-Section (RCS), Signal-to-Noise Ratio (SNR), and the existence probability assigned to each detected point by the sensor. This step helps to eliminate outliers caused by reflections and ghosting effects. Next, using vehicle localization data, points outside the track boundaries are discarded. After filtering, a simple clustering algorithm is applied to the remaining points. Clusters are formed by imposing two constraints: the distance between all points within a cluster must not exceed a maximum distance Δd_{max} , and the velocity difference between points must not be greater than Δv_{max} . Only clusters with at least N_{min} points are considered valid. This entire processing pipeline is highly efficient, taking less than 3 ms for each RADAR point cloud.

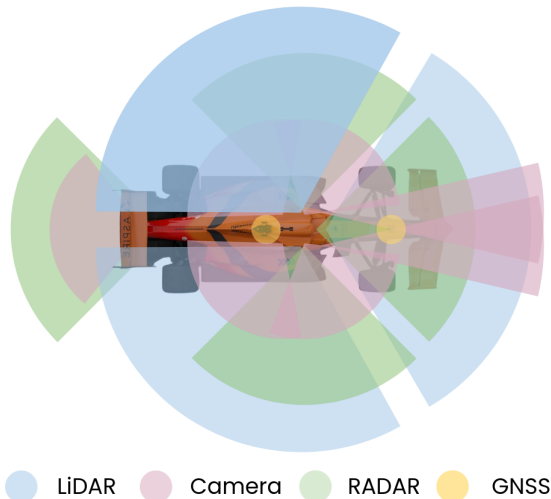


Fig. 2. Sensor setup for the Dallara EAV-24.

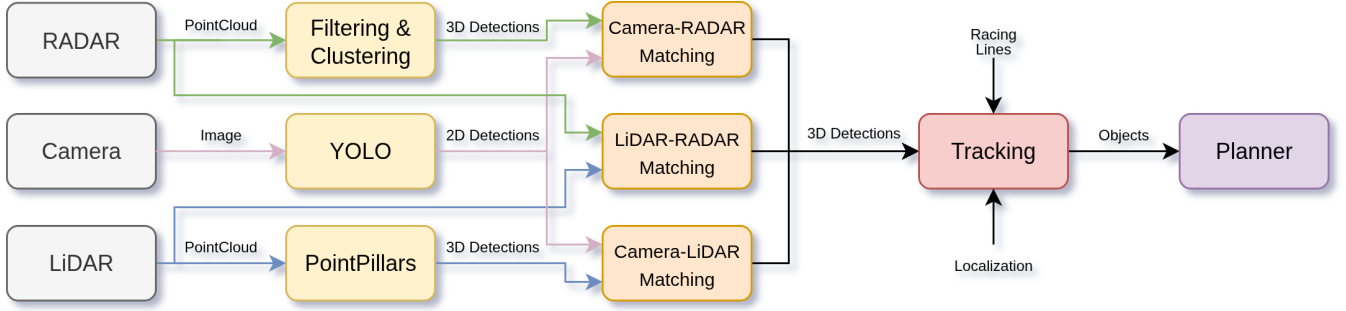


Fig. 3. In the proposed detection and tracking pipeline, object detections are extracted for each sensor modality and subsequently fused together. The fused detections are used as measurement updates by a multi-object tracker, whose outputs are finally provided to the planning module to compute evasive maneuvers.

V. DETECTIONS FUSION

For each possible pair of detections from different sources with overlapping field-of-view, a matching process is carried out. The first step involves temporal matching, where pairs of detections are created based on their timestamps, ensuring that the time difference between them is less than $\Delta T_{\max}$. Then the detection are merged based on their respective sources.

A. Camera-LiDAR Matching

For each 2D bounding-box $bb_{ji} \in BB_{\text{cam}_j}^{2D}$, the LiDAR point clouds with overlapping field-of-view are selected. These point clouds are filtered by removing the ground plane. For each point, its normal vector is computed by fitting a local plane using its neighboring points. The points are then filtered based on the norm value along the vertical axis. Additionally, points above a certain height threshold (e.g., 3 meters) and points belonging to the ego-vehicle are discarded. The remaining LiDAR points are then projected onto the camera image using the transformation matrix $\mathbf{T}_{\text{lidar}_k}^{\text{cam}_j}$, the intrinsic parameters of the camera $\mathbf{K}_j$, and the distortion parameters $\mathbf{d}_j$

$$\mathbf{p}_{\text{cam}_j} = \text{distort}(\mathbf{K}_j \mathbf{T}_{\text{lidar}_k}^{\text{cam}_j} \mathbf{p}_{\text{lidar}_k}, \mathbf{d}_j)$$

Only LiDAR points $\mathbf{p}_{\text{lidar}_k}$ whose corresponding image points $\mathbf{p}_{\text{cam}_j}$ fall inside the bounding-box bb_{ji} are considered. Finally, a distance histogram is constructed using the points range value, and the bin with the highest probability is selected to represent the object's distance. The final 3D detection is obtained by projecting the bb_{ji} into world coordinates using the selected distance. The merged detection timestamp is recomputed as the mean timestamp of all contributing points.

B. Camera-RADAR Matching

The same process used in the *Camera-LiDAR Matching* is applied between each 2D bounding-box $bb_{ji} \in BB_{\text{cam}_j}^{2D}$ and each cluster detected in the RADAR point cloud, with the exception of the ground removal step. The merged detection includes additional information on the object's radial velocity obtained by the RADAR points. The detection timestamp is

set to that of the RADAR point cloud, as it is the source that provides the distance information.

C. LiDAR-RADAR Matching

For each 3D bounding-box $bb_i \in BB_{\text{lid}}^{3D}$, the set of RADAR points within the bounding-box is identified. The merged detection is then created by augmenting the original bb_i with information on the object's radial velocity obtained by the RADAR points.

VI. OBJECT TRACKING

The object detection pipeline described earlier provides observations that are subsequently used as input to a multi-object tracker. For each tracked object, its state is estimated through an Extended Kalman Filter (EKF). Each object is also associated with a precomputed racing line that represents a trajectory it is most likely to follow along the track. The racing line encodes the position and heading of the reference path, as well as a speed profile describing the maximum achievable velocity of the race car at each point along the trajectory. Accordingly, the EKF state vector is expressed in Frenet coordinates relative to the associated racing line. The state is defined as $[s, d, v_s, v_d]$, where s denotes the position of the object along the racing line, d is the displacement of the object along the normal of the racing line, while v_s and v_d represent longitudinal and lateral velocities, respectively. The orientation of the object is inferred from the matched racing line using the curvilinear position s of the object. This choice is motivated by the fact that PointPillars is the only detector that provides direct measurements of vehicle orientation. Consequently, in the presence of one or more LiDAR failures, alternative orientation corrections are not available. The process model is derived from a modified constant-velocity formulation that introduces control inputs $[U_{v_s}, U_{v_d}]$ and constants $[\tau_{v_s}, \tau_{v_d}]$. This allows us to introduce first-order dynamics into the velocity-related state variables, as

shown below

$$\dot{s} = v_s \quad (1)$$

$$\dot{d} = v_d \quad (2)$$

$$\dot{v}_s = \frac{U_{v_s} - v_s}{\tau_{v_s}} \quad (3)$$

$$\dot{v}_d = \frac{U_{v_d} - v_d}{\tau_{v_d}} \quad (4)$$

Unlike a standard constant-velocity model, this formulation allows the predicted velocities to converge smoothly toward the reference values $[U_{v_s}, U_{v_d}]$. These references are derived from the speed profile associated with the selected racing line, thereby embedding prior knowledge of the expected vehicle behavior at different track segments into the prediction step. For example, the model naturally captures deceleration when approaching a corner and acceleration during corner exit, leading to more realistic and robust state predictions in high-speed racing scenarios. The constants $[\tau_{v_s}, \tau_{v_d}]$ define how fast the state converges to the reference values.

A. Objects Life-cycle

The tracking module is responsible for updating active objects, discarding outdated ones, and creating new objects when necessary. For each new detection, its timestamp is used to identify the two ego-vehicle states that immediately precede and follow it in time. The ego-vehicle state at the detection timestamp is then obtained via linear interpolation between these two states. The detection is then transformed into the global frame using the interpolated ego-vehicle pose. Subsequently, if the detection timestamp is older than the current time, the delay compensation mechanism is applied to ensure the temporal consistency of the tracking state despite asynchronous and latency-affected detections. This is critical in high-speed racing scenarios, where even small timing mismatches can lead to meter-scale estimation errors. To implement this, the states of all tracked objects are reverted to the corresponding past instant using a maintained history of states and corrections for each object. A Munkres [23] association process is executed based on the Euclidean distance to associate the current detection with the reverted tracked objects. If the detection matches an existing object, it is inserted into its correction history at the right timing. After the association, all object states are reported to the current time, re-applying their correction history. Otherwise, if no object matches the detection, a new object is created. The position is initialized with the detection position. The object velocity is initialized to the ego-vehicle velocity, as under normal conditions the two vehicles are expected to have similar speeds. In addition, the initial covariance is set high to describe the state uncertainty. The new object is considered tracked only after a minimum number of corrections (e.g. 3 to 5) are applied, ensuring robustness against sporadic false positives and stable convergence to its true state. Active objects are removed based on the time since the last correction or if duplication is detected. More precisely, duplication occurs when the positions of

two objects are too close to each other, in terms of squared Euclidean distance, considering their N most recent states. In this case, the most recent object is discarded.

B. Racing Line Matching

The tracking module relies on a set of precomputed racing lines, each associated with different speed profiles. These racing lines provide prior knowledge of conventional trajectories for the track. When a newly detected object is initialized, a default racing line is assigned to it, as a reference trajectory is required to express the object's state in Frenet coordinates.

As the object state evolves over time, a different racing line from the predefined set may better match the actual trajectory of the object and provide more accurate information about its heading and speed profile. Consequently, a racing line matching procedure is performed at each state update. Specifically, the cumulative Euclidean distance is computed between each candidate racing line and the last N estimated states of the tracked object. The racing line that minimizes this cumulative distance is selected, and the object state is subsequently recomputed in the Frenet coordinate frame associated with the selected racing line.

VII. EXPERIMENTAL RESULTS

During the 2025 A2RL event, multiple controlled test sessions were conducted with several autonomous race cars operating simultaneously on the track. On these occasions, sensor data has been collected onboard the race cars, and high-quality localization data for each vehicle was shared among test participants after the completion of each session. The localization data contains the position, heading, and longitudinal velocity of each vehicle on the track. In this work, the shared localization data is used as ground truth, providing a reliable benchmark to assess the performance of the proposed tracking system. For evaluation purposes, both estimated and ground truth trajectories are transformed into Frenet coordinates with respect to the track center line. This representation enables the decomposition of the tracking error into longitudinal and lateral components. In this way, longitudinal error reflects discrepancies in the vehicle's position along the direction of motion, while lateral error represents deviations perpendicular to the track. This approach offers a more meaningful analysis of the tracking performance compared to traditional Euclidean error, as it takes into account the specific dynamics of the race cars. Evaluating the longitudinal error is particularly interesting in situations where the tracked object presents significant changes in speed due to hard braking or accelerating. The longitudinal velocity error and the orientation error are also computed to assess the accuracy of the estimation. To gain a more comprehensive understanding of the performance of the tracking system, the calculated errors are correlated with the distance between the ego-vehicle and the tracked object. This analysis enables the identification of patterns in the tracking error under various conditions on the track. In general, larger errors may be observed when the ego-vehicle

is far from the opponent, highlighting situations where the system is less accurate. Analyzing how the tracking error varies under different race conditions helps to understand the strengths and limitations of the tracking pipeline and to identify specific scenarios where the detection and tracking system needs improvement or where its performance may be critical and requires accurate trajectory tracking.

This section demonstrates the necessity of a multi-modal pipeline by highlighting the complementary contributions of each modality to tracking accuracy through an ablation study. We further analyze the performance and limitations of the proposed method in critical racing scenarios that also reflect challenging urban edge cases. Specifically, we consider high relative speed overtaking, multi-object tracking under occlusions, and side-by-side overtaking.

A. Ablation Study

The ablation study is conducted on a 315-second sequence in which another vehicle is chased at high speed. Table I reports the contribution of each sensing modality to the overall tracking performance and highlights how different sensors affect specific error components. The LiDAR-RADAR configuration achieves the lowest longitudinal RMSE, low lateral, and velocity errors, confirming the complementary nature of LiDAR spatial precision and RADAR Doppler information. When only camera and RADAR are used, performance degrades noticeably, with clear increases in both longitudinal and lateral errors. This highlights the critical role of LiDAR in providing accurate 3D detection, especially in the depth direction, where monocular camera estimates are inherently less accurate at long range. Using only camera and LiDAR yields intermediate performance. In this configuration, the velocity RMSE increases significantly, demonstrating the importance of RADAR Doppler measurements for accurate velocity estimation. Although the full multi-modal configuration exhibits a slightly higher longitudinal RMSE, it achieves the best overall balance across metrics. In particular, it provides the lowest velocity error, the highest lateral accuracy, and the longest tracking range and duration. The increase in longitudinal error is due to the extended tracking range: beyond the effective distance of LiDAR, only camera and RADAR detections are available. Even if these modalities are inherently less precise, they enable earlier object detection, improving overall tracking continuity. Overall, this ablation study highlights the complementary roles of each sensor: cameras enhance tracking continuity and range, RADAR improves motion estimation, and LiDAR provides precise 3D position information. These results validate the

design of the proposed late-fusion architecture and confirm the importance of a unified multi-modal pipeline to fully exploit the heterogeneous information available from each sensing modality.

B. High Relative Speed Difference Overtaking

A critical scenario in racing occurs when a vehicle suddenly stops on the track due to a technical failure, and another vehicle approaches it at high-speed from behind. This represents one of the worst-case situations, as the very high relative speed difference between the two vehicles drastically reduces the available reaction time. Consequently, it is essential to detect and begin tracking such objects as early as possible, while also obtaining a fast-converging and accurate velocity estimate. Reliable velocity estimation enables the planning module to accurately predict the time-to-collision with the stopped vehicle and to devise an appropriate evasive maneuver to avoid the collision. Multiple controlled real-world tests were conducted with a stationary vehicle on the track. A specific case is presented in Fig. 4. The stopped vehicle is first detected at a distance of approximately 80 m while the ego-vehicle is approaching from behind at nearly 60 m/s and accelerating. At this initial stage, a new object is created as it does not match any previously tracked object. Its velocity is initialized to the ego-vehicle speed. The object begins to be actively considered 150 ms after the first detection, once the minimum number of matched detections is reached. At this point, the object is 68.81 m from the ego-vehicle, with an estimated velocity of 6.64 m/s. Although the estimated speed of the object is not actually zero, the large relative speed difference between the ego-vehicle and the object provides critical information for the planning module to safely prepare the overtake maneuver. After 350 ms from the initial detection, the time-to-collision (TTC) reaches 1 s, representing the remaining time before a potential collision if no evasive action is taken. At this moment, the object is 57.35 m from the ego-vehicle, and

TABLE I
SENSOR MODALITIES ABLATION STUDY RESULTS

Modalities			Longitudinal	Lateral	Velocity	Distance	Time
Camera	LiDAR	RADAR	RMSE [m]	RMSE [m]	RMSE [m/s]	Max [m]	[s]
×	✓	✓	1.17	0.41	0.95	73.05	308.6
✓	×	✓	1.72	0.53	1.13	95.15	305.7
✓	✓	×	1.54	0.42	1.43	76.91	306.4
✓	✓	✓	1.40	0.41	0.77	96.0	313.1

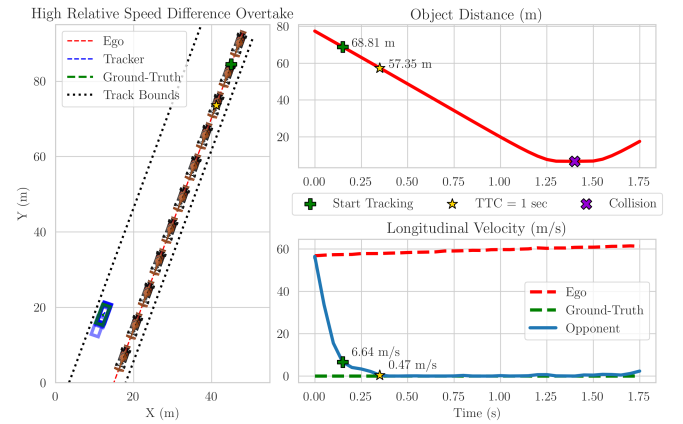


Fig. 4. The object is initially detected at a distance of approximately 80 m and begins to be actively tracked once a sufficient number of detections are associated with it. At the start of tracking, the object is 68.81 m away, moving with an estimated velocity of 6.64 m/s. At the distance of 57.35 m, the TTC is reduced to 1 second, and the estimated velocity converged to nearly 0 m/s.

its estimated velocity has converged to almost 0 m/s. The maximum tracking errors occur at the moment the object is first initialized, with a longitudinal error of 1.03 m, a lateral error of 0.25 m, and an orientation error of 0.90 degrees. After initialization, the estimate errors rapidly drop to almost zero, and the state remains stable. This analysis demonstrates that the proposed tracking pipeline can detect stopped objects with high relative speed differences early, providing the planning module with sufficient time and accurate information to compute a safe and effective evasive trajectory.

C. Multi-object Tracking Under Occlusions

In racing competitions, vehicles tend to follow similar trajectories to achieve optimal lap times. As a consequence, they are often longitudinally aligned, and the closest opponent is likely to occlude the visibility of other vehicles from the perspective of the ego-vehicle. In such conditions, when the ego-vehicle initiates an overtaking maneuver, a previously occluded opponent may suddenly appear in front of it. The detection and tracking pipeline must therefore minimize the latency required to initialize and stabilize the tracking of newly visible opponents in order to react safely. In the scenario shown in Fig. 5, the ego-vehicle follows the blue vehicle, which suddenly brakes with a deceleration of 7 m/s^2 . In response, the planning module initiates an overtaking maneuver. As soon as the ego-vehicle moves laterally, a previously occluded and stopped vehicle (orange) enters the field of view. This vehicle, which had not been tracked before, is first detected at a distance of approximately 52 m, and its estimated velocity rapidly converges to 0 m/s within the following two iterations. As in the previous scenario, the maximum tracking errors occur at the time of object initialization, with a longitudinal error of 0.29 m and a lateral error of 0.83 m. In contrast, the orientation error remains practically negligible throughout the tracking interval. This scenario demonstrates that the proposed detection

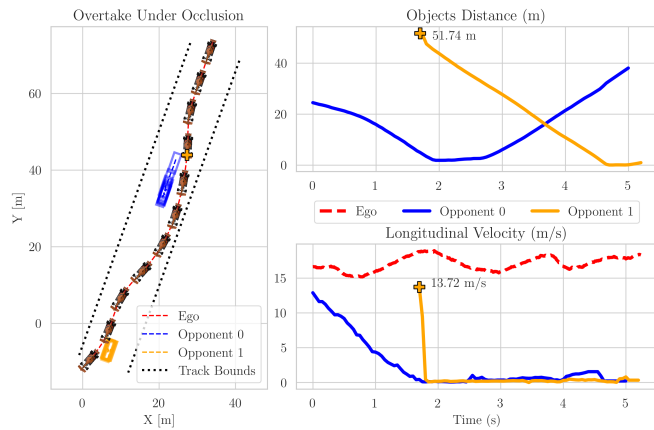


Fig. 5. In this scenario, the blue opponent occludes the visibility of the orange opponent. As soon as the ego-vehicle initiates the overtaking maneuver, the orange vehicle becomes visible at a distance of 51.74 m ahead, providing sufficient time for the planning module to adapt the overtaking trajectory.

and tracking pipeline can immediately initialize the tracking for newly visible objects, providing sufficient time for the planning module to adapt the overtaking trajectory.

D. Side-by-side Overtaking

Side-by-side overtaking represents one of the most critical situations in racing, as the vehicles operate at extremely close distances and near their handling limits. Ensuring the safety of both race cars and the successful execution of the overtaking maneuver requires high tracking accuracy. In this case study, we evaluate a side-by-side overtaking scenario in which our race car overtakes another vehicle, reaching a minimum lateral distance of 1.75 m. The analysis is shown in Fig. 6. On the left, the ego-vehicle maneuver is shown in red and the opponent's estimated trajectory in blue overlapped with the actual opponent ground-truth trajectory in green. On the right, the tracking error with respect to the ground path is shown. The error line is colored based on the minimum distance between the ego-vehicle and the opponent ground-truth position. In addition, the three colored sections indicate the relative position of the opponent vehicle with respect to the ego-vehicle: in front, side-by-side, and behind, respectively. The longitudinal exhibits a clear bias that depends on the relative position of the opponent vehicle. When the opponent is in front, the error is predominantly negative, it oscillates around zero when the vehicles are laterally aligned, while it increases during the overtaking phase and remains positive once the opponent is consistently behind the ego-vehicle. This behavior is mainly attributed to the tracker correction being dominated by the fusion of camera detections with the other sensors. Our camera-based detections lack semantic awareness of which specific part of the opponent vehicle is actually being observed. As a result, when the opponent is ahead, the tracker mainly observes the rear of the vehicle and underestimates the longitudinal position. Conversely, when the opponent is behind, the tracker mainly observes the front part of the vehicle, leading to an overestimation of the longitudinal position. The lateral error is also affected

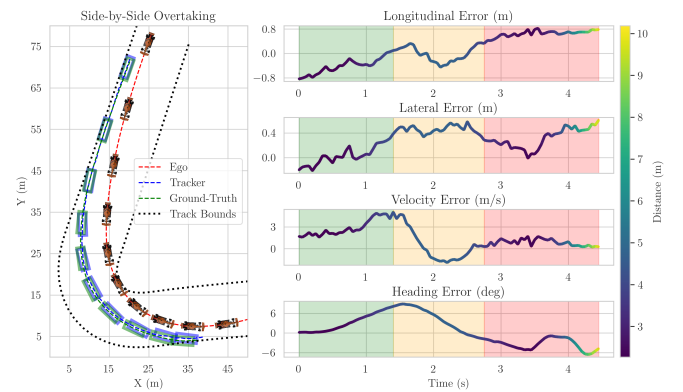


Fig. 6. The side-by-side overtake maneuver is illustrated on the left, while the corresponding tracking errors are shown on the right. The error are correlated to the minimum distance between the two vehicles. The three colored sections indicate the relative position of the opponent vehicle with respect to the ego-vehicle: in front, side-by-side, and behind, respectively.

by a similar bias, but only when the vehicles are laterally aligned. This bias results in a maximum longitudinal error up to 0.84 m and a maximum lateral error up to 0.61 m. Additionally, the lateral error observed when the opponent vehicle is behind is influenced by the reduced number of measurement updates. In this configuration, rear visibility is provided only by the rear camera and the rear RADAR, as the LiDARs are occluded by the rear wing. The estimated longitudinal velocity initially exhibits an overshoot, reaching a peak error of 5 m/s, and fails to fully capture the opponent's abrupt braking maneuver. It then gradually converges to the correct value in approximately 1.75 seconds. Finally, the heading error is largely driven by the racing line selected from the set provided to the tracker. The chosen racing line assumes a tighter cornering trajectory, whereas the opponent vehicle instead reacted to the ego-vehicle taking a wider turn. This mismatch results in a positive increase in heading error, up to 9 degrees. As following the taken trajectory would have led the opponent outside the track boundaries, the opponent steered left earlier than expected, causing the heading error to shift toward negative values. This analysis clearly highlights that estimation accuracy strongly depends on the relative position of the tracked object with respect to the ego-vehicle, as well as on the need to incorporate a more complete semantic understanding of vehicle geometry into the detection process. Despite the limitations discussed, the achieved estimation accuracy was sufficient to successfully complete the side-by-side overtaking maneuver.

VIII. CONCLUSION

The proposed architecture shows that combining multi-modal sensing with delay-aware tracking and dynamics-informed priors is key for reliable perception in extreme-speed environments. A late-fusion scheme improves the robustness to sensor failures, while the tracker compensates for detection delays and incorporates prior knowledge of the vehicle and track. Real-world racing experiments validate the effectiveness of the pipeline for safe high-speed planning. Future work will address the observed limitations by incorporating semantic-aware vehicle geometry models, enabling the tracker to explicitly reason about which portion of the opponent vehicle is being observed and reducing estimation biases.

REFERENCES

- [1] J. Mao, S. Shi, X. Wang *et al.*, "3d object detection for autonomous driving: A comprehensive survey," *Int. J. Comput. Vis.*, vol. 131, pp. 1909–1963, 2023.
- [2] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 458–488, 2022.
- [3] A. Kulkarni *et al.*, "Racecar - the dataset for high-speed autonomous racing," in *Proc. 2023 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023. [Online]. Available: <http://dx.doi.org/10.1109/IROS55552.2023.10342053>
- [4] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 12 689–12 697.
- [5] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," *arXiv:2012.15712*, 2020.
- [6] S. Ghosh, Y.-H. Chen, C.-H. Huang, A. S. M. M. Jameel, C. C. Ho, A. E. Gamal, and S. Labi, "A racing dataset and baseline model for track detection in autonomous racing," 2025. [Online]. Available: <https://arxiv.org/abs/2502.14068>
- [7] M. Nye, A. Raji, A. Saba, E. Erlich, R. Exley, A. Goyal, A. Matros, R. Misra, M. Sivaprakasam, M. Bertogna, D. Ramanan, and S. Scherer, "Betty dataset: A multi-modal dataset for full-stack autonomy," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 2453–2460.
- [8] M. Cellina, M. Corno, and S. M. Savaresi, "Lidar-based vehicle detection and tracking for autonomous racing," *arXiv preprint*, vol. arXiv:2501.14502, 2025. [Online]. Available: <https://arxiv.org/abs/2501.14502>
- [9] C. Jung *et al.*, "An autonomous racing system: Design, implementation, and analysis; team kaist at the iac," *Field Robotics*, vol. 3, pp. 766–800, 2023.
- [10] J. Kim, S. Moon, S. Nah, and D. H. Shim, "Monodino-detr: Depth-enhanced monocular 3d object detection using a vision foundation model," 2025. [Online]. Available: <https://arxiv.org/abs/2502.00315>
- [11] P. Karle, F. Fent, S. Huch, F. Sauerbeck, and M. Lienkamp, "Multi-modal sensor fusion and object tracking for autonomous racing," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3871–3883, 2023.
- [12] A. Saba *et al.*, "Fast and modular autonomy software for autonomous racing vehicles," *Field Robotics*, vol. 4, pp. 1–45, 2024.
- [13] X. Weng, J. Wang, D. Held, and K. Kitani, "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics," *IROS*, 2020.
- [14] S. T. Watt, S. Achanta, H. Abubakari, E. Sagen, Z. Korkmaz, and H. Ahmed, "Understanding and applying precision time protocol," in *2015 Saudi Arabia Smart Grid (SASG)*, 2015, pp. 1–7.
- [15] W. Burger, "Zhang's camera calibration algorithm: In-depth tutorial and implementation," University of Applied Sciences Upper Austria, School of Informatics, Communications and Media, Dept. of DigitalMedia, Hagenberg, Austria, Tech. Rep. HGB16-05, 05 2016. [Online]. Available: https://www.researchgate.net/publication/303233579_Zhangs_Camera_Calibration_Algorithm_In-Depth_Tutorial_and_Implementation
- [16] S. Chen, H. Zhang, Z. Yang, Y. Yang, and K. Wang, "Aruco code-based calibration method for lidar and camera integration," in *2024 IEEE 25th China Conference on System Simulation Technology and its Application (CCSSTA)*, 2024, pp. 674–678.
- [17] C. Doer and G. F. Trommer, "Radar inertial odometry with online calibration," in *2020 European Navigation Conference (ENC)*, 2020, pp. 1–10.
- [18] A. Toschi, F. Prignoli, and M. Bertogna, "Modular decision-making and drivable areas for multi-agent autonomous racing," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 12 435–12 441.
- [19] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. [Online]. Available: <https://arxiv.org/abs/2004.10934>
- [20] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," *CoRR*, vol. abs/1805.04687, 2018. [Online]. Available: <http://arxiv.org/abs/1805.04687>
- [21] M. Verucchi *et al.*, "A systematic assessment of embedded neural networks for object detection," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2020, pp. 937–944.
- [22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.
- [23] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. [Online]. Available: <https://doi.org/10.1137/0105003>