Heterogeneous Data Fusion for Accurate Road User Tracking: A Distributed Multi-Sensor Collaborative Approach

Simone Mentasti*[1] Alessandro Barbiero*[2] Matteo Matteucci[1]

***Abstract*— This work presents the design and validation of a distributed multi-sensor object tracking algorithm designed to integrate heterogeneous sensory data from multiple static acquisition stations. The primary challenge addressed is the accurate tracking of targets in complex urban environments, where occlusions and the dynamic nature of traffic frequently hinder detection and tracking efforts. This challenge is particularly relevant in multimodal exchange areas, where vehicular traffic merges with heavy pedestrian and bicycle flow. We also address the scenario of delayed detection, which can easily occur when data from multiple stations are combined or when intensive data processing is performed. Our algorithm ensures high coverage and accuracy by maintaining dual Extended Kalman Filter states for each object, thus allowing for the assimilation of delayed detections and preserving optimal filter estimates at all times. The results of the proposed pipeline, tested using a digital twin of the Milano Bovisa Campus, demonstrate its efficacy, achieving high tracking precision across various scenarios and sensor combinations. Moreover, the results highlight the advantages of a distributed multi-sensor acquisition system compared to a single central station.**

I. Introduction

The contemporary urban landscape, characterized by its densely populated areas and busy streets, presents unique challenges and opportunities for the transportation sector. In European cities, where the volume of transport per capita has significantly increased in recent years [1], managing the complex dynamics of various road users becomes a critical task [2]. These environments are not only crowded but also comprise a diverse mix of vehicles, pedestrians, cyclists, and other vulnerable road users, making the task of detecting and tracking them more complex and essential. Indeed, recent work has highlighted the importance of accurately identifying and tracking these users to enhance road safety in urban settings [3].

In this intricate urban milieu, the interaction between different road users is often unpredictable and necessitates advanced technological solutions. The vulnerability of pedestrians and cyclists is particularly relevant in multimodal intersections, where interaction between the most vulnerable users and heavy vehicles cannot be avoided [4]. This is the case for areas close to stations, airports, or city centers where pedestrian and bicycle traffic merges with private vehicles and public transportation [5]. In these scenarios, accurate systems are essential for preventing accidents, regulating the traffic flow, and ensuring a harmonious coexistence of vehicles and human road users [6].

Fig. 1. Image of the digital twin of the Bovisa Politecnico Campus used for testing the proposed tracking pipeline. The area, marked by numerous pedestrian crossings and road intersections near the campus entrance, represents the ideal scenario for testing due to its complex urban environment.

Modern systems dynamically evaluate the state of the road infrastructure and handle the traffic to prevent congestion and increase road safety [7]. This high-level infrastructure can also be integrated within heavy vehicle control systems, like buses, to increase the safety of vulnerable road users further, predicting the vehicle trajectory and providing Advanced Driver-Assistance Systems (ADAS) with information on possible collisions or hazards [8].

However, for such systems to function effectively, precise information on the road conditions, the number and types of users, and their trajectories is essential. The diversity of urban traffic necessitates adaptive algorithms that can operate efficiently in varied and dynamic scenarios, integrating different data sources such as cameras, radars, and LiDARs [9] [10] to accurately identify all classes of road users. Furthermore, given the large areas of interest, distributed systems that aggregate and combine data from multiple sources become crucial. These systems ensure comprehensive coverage of the monitored areas, mitigate vehicle occlusions, and track subjects of interest efficiently.

This study addresses the aforementioned challenges by developing a distributed, multi-sensor object-tracking algorithm. The proposed algorithm integrates 2D and 3D detections from various sensors, comprehensively characterizing the traffic environment. This work's core contribution is

*These authors contributed equally.

[1,2]Department of Electronics Information and Bioengineering, Politecnico di Milano, p.zza Leonardo da Vinci 32, Milan, Italy, name.surname@polimi.it[1], name.surname@mail.polimi.it[2]

This paper is supported by "Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile – CNMS)" project funded by the European Union NextGenerationEU program within the PNRR, Mission 4 Component 2 Investment 1.4.

designing a modular and scalable target tracking system tailored for urban scenarios and multimodal exchange nodes, where a dynamic interaction of multiple and heterogeneous agents occurs. The algorithm is developed to combine asynchronous data from diverse sensor types, such as cameras and LiDARs. Multiple factors dictate the usage of LiDARs and cameras: the reduced cost of camera sensors and the higher semantic content provided by images, which can be used for accurate target classification. The system is designed to take different perspectives and observation points within the area of interest as input and also compensate for data delays generated by communication or processing. In particular, raw data can be processed in a distributed way by each monitoring station, and only high-level representations, the detections, are sent to a central processing unit that runs the tracking pipeline. The algorithm was validated using the CARLA simulator [11] within a custom-built digital twin of the Milano Bovisa Politecnico campus. This area serves as an exemplary multimodal exchange, teeming with varied road users, particularly due to the proximity of the railway station and the campus entrance, which generate heavy pedestrian traffic. Additionally, multiple service roads introduce private vehicles and public transportation flows. As illustrated in Fig. 1, the selected area includes various intersections and pedestrian crossings, highlighting the necessity for precise road monitoring and traffic planning.

The choice to employ the CARLA simulator was dictated by the need for precise ground truth data to validate the tracker. Due to the number of objects tracked, it was not feasible to employ real data, and using a simulated environment was preferred. This is also motivated by the focus of this work, which revolves around the tracking pipeline and not the detection. Therefore, having data that is not highly photorealistic is not a limitation for the pipeline validation.

This work is structured as follows: in Section II, we explore the current state of the art in target tracking in autonomous driving and urban scenarios. Next, Section III presents our proposed tracking pipeline, with a particular focus on the employed filtering technique and 2D-3D association strategies. In Section IV, we present the results of our approach on simulated data generated using the Carla simulator to be able to compare the results with accurate ground truth for all the tracked subjects. Finally, Section V draws some final remarks and proposes future directions for this work.

II. Related works

Road user detection and tracking has been a significant research topic for many years, particularly regarding the use of single surveillance cameras to monitor specific road areas such as junctions and roundabouts [12] [13]. This holds especially true in the case of vehicle detection and tracking [14], which was originally performed using geometrical computer vision solutions [15] and has recently transitioned to deep-learning approaches [16]. More recent advances have also introduced the use of multi-camera systems [17] [18] and wide field of view or fisheye cameras [19] [20] to better cover the monitored area and track targets with higher accuracy and for extended durations.

While camera-based detectors and tracking algorithms have been thoroughly analyzed in recent years, leading to efficient and optimized solutions, other sensor classes, such as LiDARs and radars, have only recently gained popularity. The advent of more complex data types has been driven primarily by developments in deep learning technologies for processing heterogeneous data sources and advanced sensor fusion algorithms. At the heart of these advancements is the adoption of Deep Neural Networks (DNNs), which have become a dominant force in multi-object detection and tracking across various driving situations. The integration of data from diverse sensors, particularly the fusion of cameras, LiDAR, RADAR, and GPS, is crucial in tackling the complexities of dynamic and often unpredictable driving environments, enhancing the detection and tracking capabilities of autonomous vehicles, as highlighted in [21] [22].

With the increasing interest in autonomous driving technologies and Advanced Driver-Assistance Systems (ADAS), there has been a focus on enhancing vehicle detection and tracking, especially under adverse weather conditions. Multi-scale deep convolutional neural networks have successfully addressed challenges posed by reduced visibility, such as heavy snow or fog [23]. Furthermore, the efficacy of model-based approaches using laser range finders and Bayes filters in noisy urban settings has been demonstrated by Petrovskaya and Thrun [24], emphasizing the importance of robust and adaptive systems.

Recent advancements in deep learning and the design of integrated models have led to a common approach that simultaneously performs the detection and tracking of targets, leveraging convolutional neural networks and learning-based techniques. End-to-end learning systems, for example, can detect and track objects within the same process without the need for a postprocessing step [25]. These solutions effectively handle complex scenarios with occluded or overlapping objects, leveraging sensors' depth and velocity information for better distinction and tracking accuracy [26]. In contrast, sequential models, popular before the rise of deep learning, offer the advantage of custom detectors for different sensors and a focus on specific tracking aspects. Variants of the Kalman Filter have been widely adopted for this task, both for tracking from moving vehicles and infrastructure [24] [27]. In scenarios involving multiple objects, traditional tracking techniques like Multiple Hypothesis Tracking (MHT) and Joint Probabilistic Data Association Filters (JPDAF) have been widely used for robust tracking of overlapping agents [28]. Due to the increased number of sensors and required processing time, some approaches designed to handle asynchronous and delayed data have been proposed. In particular, new filter architectures tailored to handle significant time delays in the measures [29] and hybrid approaches that combine multiple filter outputs to predict the system state have been used [30].

Finally, advancements in roadside infrastructure, such as camera-based trajectory estimation frameworks [9], play a

pivotal role in generating accurate reference data for autonomous driving applications. This is complemented by recent works exploring correlation filter-based multi-object tracking and radar-camera fusion for 3D tracking [31] [26]. The role of multi-sensor systems in autonomous driving has proven crucial in many scenarios, leading to the current design choices for these vehicles. This multisensory approach is critical in overcoming the limitations of individual sensors, leading to more comprehensive environmental perception, as furthered by Fu et al. [32] with their innovative real-time multi-vehicle tracking frameworks in intelligent vehicular networks.

Our work extends the state of the art by focusing on asynchronous multisensory and distributed scenarios, where multiple acquisition systems record and share data of a crucial road area from different points of view. This approach guarantees high coverage and accurate tracking of all targets, even in scenarios where occlusions mask the objects. Moreover, the proposed approach is designed to seamlessly fuse different types of sources, enabling the tracking of road users even if they are visible only from one of the sensors.

III. Tracking algorithm

The tracking algorithm is designed with two primary objectives. First, it must be capable of combining heterogeneous sensory data, such as images from cameras and point clouds from LiDARs, which provide 2D and 3D information, respectively. Second, it should effectively merge data coming from different monitoring stations. This is necessary because multimodal areas are often too extensive to be effectively covered by a single acquisition station (i.e., a set of cameras and LiDARs mounted at a single central point). Therefore, data from different stations must be efficiently combined, enabling the tracking of the position and state of users across various sensors and systems while considering possible transmission or processing delays.

For this work, we consider three asynchronous acquisition stations, each equipped with cameras and LiDARs, but the approach can be easily generalized to more complex setups. Moreover, since this work's goal is the tracking pipeline's design, we do not focus on the raw data from the sensors but rather on the output of a detection or segmentation algorithm. Specifically, the input to the image-based tracking pipeline branch consists of 2D bounding boxes, while we assume a detector can provide 3D bounding boxes from the LiDAR. In Section IV, we perform various experiments, employing either the ground truth with realistic noise or the output of a detection algorithm. But, for the design of the tracking pipeline, our focus remains solely on the processed data. A high-level representation of the complete tracking pipeline, described in the next sections, is depicted in Fig. 2.

A. Tracker architecture

The first step of the pipeline, $detection\ analysis$, processes the asynchronous data from the sensors' detectors. Specifically, detections are grouped into two classes, $detections_{low}$ and $detections_{high}$, based on the confidence score of the detector. This categorization is particularly useful for objects at a distance or partially occluded, which are still worth tracking, although the detector might struggle to consistently detect them. Objects detected in the previous step are also divided into two categories at this phase: objects seen for the first time, which need to be confirmed as real targets, and objects that already have an active tracker $unconfirmed\ tracks$ and $confirmed\ tracks$, respectively.

The next step involves data association, $first\ association$. The state of tracked and lost objects (i.e., objects that have not been seen for a few frames) is predicted using an Extended Kalman Filter (EKF). For camera data, tracked objects are projected back to the image plane, while for LiDAR data, association is performed directly in 3D. To perform the association, we compute the Intersection over Union (IoU) between the $confirmed\ tracks$ and the $detections_{high}$. The IoU value is calculated for all possible pairs, generating a scoring matrix used to solve the linear assignment problem and find the minimum cost function. Then, only matches exceeding a fixed IoU threshold are considered. We employ the Jonker-Volgenant algorithm [33] for this purpose. The output of this phase includes $matched\ objects$, a list of $unmatched\ tracks$, and $unmatched\ detections_{high}$.

The third step (i.e., $second\ association$) performs the same operations as the second but with the $detections_{low}$ objects and the confirmed tracks that have not yet been assigned (i.e., the $unmatched\ tracks$). In this case, the threshold for the Intersection over Union (IoU) is stricter to prevent matching tracks with false detections. From this steps, we produce a second list of associations. Objects that have not been matched are then considered lost and are kept in the filter for a fixed amount of time before being removed.

Finally, in the $third\ association$, we consider the $unconfirmed\ tracks$; for these, we attempt to match them with the remaining $unmatched\ detections_{high}$. This step is performed last since we initially match the detections only with tracks that have been observed multiple times. Therefore, we know they belong to real objects, and only after with new objects that have been seen only once and might be false detections. The association is conducted in a similar manner to the previous two steps but with stricter IoU.

If some $detections_{high}$ remain unmatched at this stage, we initialize a new tracker for these objects, and they are placed into the pool of $unconfirmed\ tracks$. We consider only the $detections_{high}$ to minimize the risk of creating a high number of false positives, which could lead to an excessive number of trackers and potentially overload the system.

The final step consists of cleaning up the tracks. We also review the list of already lost objects, and if they have not been matched for an extended period, they are removed. Additionally, we check the Intersection over Union (IoU) between tracks to identify and remove possible duplicates.

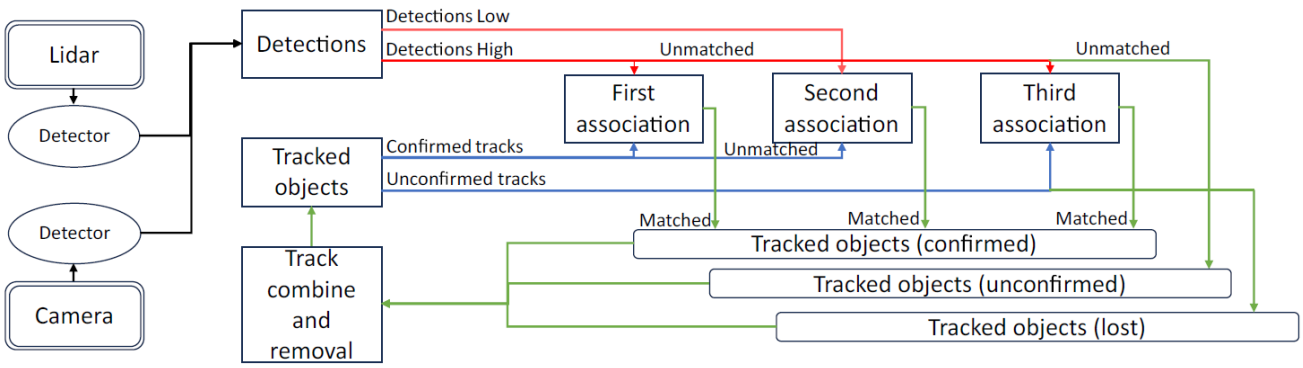

Fig. 2. Schema of the tracking algorithm: At the top, new detections (red) from the LiDAR or camera are divided into two streams and combined through consecutive associations with the tracked obstacles (blue) at the bottom. Gradually, data are matched and saved as confirmed tracks. Detections that are not matched are saved as unconfirmed tracks, and tracks that are not matched are categorized as lost tracks.

B. Extended Kalman Filter

As mentioned in the previous section, to track targets over time, we utilize an Extended Kalman Filter that combines 2D and 3D detections. Specifically, to fully represent each tracked element, we define the state of the target as $(x, y, \theta, l_{ratio}, d_{ratio}, h, v, \omega)$. For improved numerical stability, as suggested in [34], we represent the length and width of the bounding boxes as ratios with the height; $l_{ratio} = \frac{l}{h}$ and $d_{ratio} = \frac{d}{h}$. Targets are then described with the position of their center, the heading, the speed v, and the angular velocity ω. The state equation is computed using the Runge-Kutta integration model, which is shown in 1.

$$\begin{cases} x^{(t)} & = x^{(t\text{-}1)} + dt \cdot v^{(t\text{-}1)} \cos(\theta^{(t\text{-}1)} + \frac{\omega^{(t\text{-}1)} \cdot dt}{2}) \\ y^{(t)} & = y^{(t\text{-}1)} + dt \cdot v^{(t\text{-}1)} \sin(\theta^{(t\text{-}1)} + \frac{\omega^{(t\text{-}1)} \cdot dt}{2}) \\ \theta^{(t)} & = \theta^{(t\text{-}1)} + \omega^{(t\text{-}1)} \cdot dt \\ l_{ratio}^{(t)} & = l_{ratio}^{(t\text{-}1)} \\ d_{ratio}^{(t)} & = d_{ratio}^{(t\text{-}1)} \\ h^{(t)} & = h^{(t\text{-}1)} \\ v^{(t)} & = v^{(t\text{-}1)} \\ \omega^{(t)} & = \omega^{(t\text{-}1)} \end{cases} \quad (1)$$

Given this representation, the 3D measurement matrix becomes trivial, as shown in Equation 2. Conversely, the measurement matrix for 2D detections must take into account the camera model to perform the 3D-2D projection and is analyzed in detail in Section III-C.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (2)$$

Due to the nature of the data and the time required to process them, some delays might occur in detection. This is particularly true when deep-learning-based models are applied to the data and computations have to be performed directly on the infrastructure, where limited computational power is available. This can cause detections to arrive out of order and with significant delay. In these scenarios, it is important to fuse this information, even if it refers to a previous time, because it still carries relevant information. Moreover, some sensors might require heavier processing, and fusing only the latest data could lead to consistently discarding specific sensory data.

To maintain all the possible information coming from detections and ensure the filter estimate is optimal at all times, we constantly keep two parallel EKF states for each object. When a detection arrives, it can instantly update the filter and provide a new updated state. If, after that event, a delayed detection arrives, we can use the second EKF state, left at a previous timestamp, to integrate the late-coming detection and subsequently update the filter with the last saved detection. In this way, we can achieve an optimal estimate without discarding delayed detections. A visual representation of this process can be seen in Figure 3.

Target classes are also tracked and updated according to the predicted values. To prevent sudden class switches due to possible false positives, classes are updated using Equation 3, where $conf_i$ represents the target's class confidence and $conf_d$ is the detection's confidence. The class is altered, and new priors are applied to modify the object's shape only when the confidence falls below 0.5.

$$\text{conf}_{i+1} = \begin{cases} \frac{\text{conf}_i \cdot 2 + \text{conf}_d}{3} & \text{if detected class} = \text{tracked class} \\ \text{conf}_i - \frac{\text{conf}_d}{3} & \text{if detected class} \neq \text{tracked class} \end{cases} \quad (3)$$

C. 2D detections

To perform data association between the 2D detections and the 3D tracked objects, it is necessary to reproject the 3D objects onto the image plane. This process guarantees an easier and more accurate association between the detections and the tracks than performing a 2D to 3D projection. To

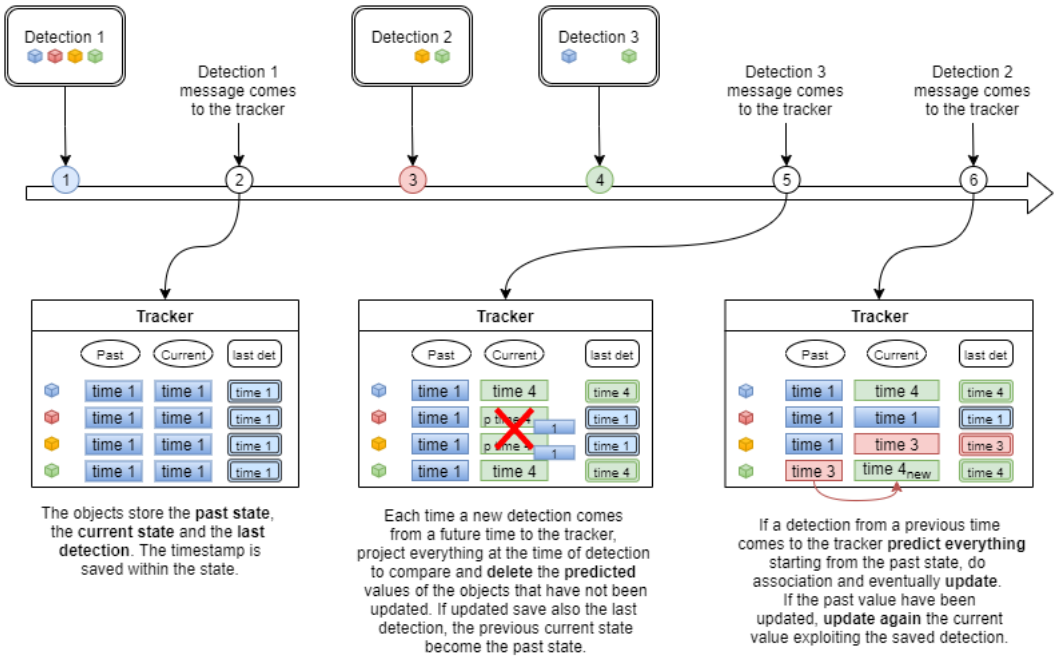

Fig. 3. Tracker's update routine. The update routine is designed to preserve maximum information and prevent the integration of the same measurement twice within the Kalman Filter. Moreover, the tracker can integrate past information to prevent information loss, particularly when data are received with significant delay due to processing or network issues.

achieve this, the tracked obstacle must be represented in an efficient way that can be integrated into the Kalman filter and allow comparison with the 2D detections on the image plane. Directly projecting the 3D bounding box is sub-optimal and introduces excessive nonlinearities. For this reason, when data association and tracking are performed on the image plane, the 3D targets are represented so that they always reproject onto ellipses on the 2D image, making the process more robust. This approach allows the pose estimation problem to be efficiently solved using the underlying closed-form projection equation. In particular, our approach uses the ellipsoid-ellipse projection within the measurement matrix of the EKF to fuse the 2D detections in order to improve the state of the ellipsoid-based tracked objects.

Several steps are required to integrate the ellipsoids into the filter, starting with the 3D tracked bounding box. In particular, we first define the $View\ Matrix\ V$ as the $4x4$ matrix that transforms points from the $world\ frame$ to the $camera\ frame$.

$$\mathrm{V} = \begin{bmatrix} \mathrm{rot}_m & \mathbf{p} \\ \mathbf{0}_3 & 1 \end{bmatrix} \tag{4}$$

Where rot_m is a $3x3$ rotation matrix and p is the translation vector, representing the position of the camera relative to the world frame. Next, to convert the 3D bounding box into an ellipsoid, we consider the three components of the bounding box: position, size, and orientation. The size of the bounding box (i.e., height h, width w, length l) is used to compute the matrix Q, which has as its diagonal the semi-axes a, b, c of the ellipsoid:

$$Q = \begin{bmatrix} a^2 & 0 & 0 & 0 \\ 0 & b^2 & 0 & 0 \\ 0 & 0 & c^2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{5}$$

The rotation matrix R is computed as an affine transformation derived from the $3x3$ rotation matrix obtained from the bounding box heading. This approach is based on the general assumption in 3D point cloud detection that the roll and pitch angles are zero, and only the yaw angle is relevant for the vehicle state.

$$R = \begin{bmatrix} R^T & 0 \\ 0 & 1 \end{bmatrix} \tag{6}$$

Finally, the translation matrix T is initialized using the translation vector, which is computed from the center of the bounding box, as shown in Equation 7.

$$\mathrm{center} = \begin{bmatrix} cx \\ cy \\ cz \end{bmatrix} \quad T = \begin{bmatrix} I_3 & \mathrm{center} \\ \mathbf{0} & 1 \end{bmatrix} \tag{7}$$

The dual ellipsoid can then be computed as a combination of the three elements as follows:

$$Q_{fixed\_frame} = T * R^T * Q * R * T^T \tag{8}$$

Finally, using the camera view matrix, it is possible to reproject in the camera frame:

$$Q_{camera\_frame} = V * Q_{fixed\_frame} * V^T \tag{9}$$

From this ellipsoid, we can compute the $3x3$ projected ellipsoid matrix on the image plane using the intrinsic camera calibration, represented by the camera matrix P:

$$C = P * Q_{camera\_frame} * P^T \tag{10}$$

As a final step, it is possible to extract from the C matrix the ellipse parameters (u, v, a, b, ψ), representing the center, the semi-axes, and the orientation with respect to the x axis.

In the 2D scenario, the measurement matrix H is then defined as follows:

$$X_{ellipse} = H * X_{State}^T \tag{11}$$

where we have previously defined $X_{ellipse} = (u, v, a, b, \psi)$ and $X_{state} = (x, y, \theta, l, d, h, v, w)$. The H matrix can be computed by combining all the previously described operations.

The opposite process is instead employed when using 2D detections to initialize new tracks. To initialize a 3D object from a 2D detection, we use the average dimensions of object classes along with the geometry derived from the view and projection matrices. This process involves leveraging the intrinsic and extrinsic calibration of the camera to perform a bird's eye view projection. Under the assumption of a flat surface, this projection is used to retrieve the initial position of the tracked object.

IV. Experimental Results

We employed the Carla simulator to validate our proposed pipeline, primarily due to the availability of accurate ground truth for all tracked targets, which is not available with real-recorded data. Indeed, while real data are preferable, obtaining precise positions via RTK-GNSS for a high number of subjects is challenging, making a simulated environment more suitable. This choice is also motivated by the focus of this work on the tracking pipeline rather than detection, meaning that slightly less realistic images do not compromise the validity of the data.

The tests were conducted in a digital twin of the area near the Bovisa Politecnico Campus, fully reconstructed inside Carla for use as a virtual test environment. Figure 4 illustrates the simulation scenario, depicting the Milano Bovisa Politecnico campus and the positions of three sensor stations monitoring intersections and pedestrian crosswalks. Since this digital twin reconstruction simulates the real setup of the campus, each of the three stations is equipped with realistic sensors. In particular, a 360° LiDAR, simulating an Ouster OS-Dome, and two sets of cameras, one with a wide 120° field of view and one with a narrow 32° field of view. Our experiments used various scenarios, ranging from a simple single-sensor acquisition station to multiple multi-sensor stations cooperating to track targets. We performed tests using both state-of-the-art detectors and the ground truth combined with realistic noise to simulate inaccurate detections. In particular, for the noise simulation, we employed a 15% missed detection rate, $0.15m$ error in positioning, and $0.3m$ error in the bounding box size estimate.

Fig. 4. Image of the Carla simulation scenario, digital recreation of the Milano Bovisa Politecnico campus. The red circles represent the position of the tree sensor station, acquiring and monitoring the intersection and the pedestrian crosswalks.

The first test scenario, $V1$, involved only three LiDARs mounted at different locations, covering the area characterized by intersections and multiple pedestrian crossings depicted in Fig. 4. A second, more complex setup, $V2$, expanded the three-LiDAR system of $V1$ with four RGB cameras processed using Yolov8 [35] and three segmentation cameras, simulating different camera types like thermal or sensors with embedded AI. The final configuration, $V3$, consisted of a single station equipped with a LiDAR, a camera, and a segmentation camera to prove the effectiveness of the multi-station setup.

The experimental results, presented in Table I, underscore the accuracy of our solution, which consistently achieves high precision in tracking targets across various stations and exhibits a high True Positive rate. Remarkably, the system compensates for the 15% simulated missed detection rate by accurately tracking targets, resulting in a low missed rate across all configurations. Moreover, each configuration demonstrates a low Mean Object Tracking Precision (MOTP) value, with the lidar-only setup achieving the lowest at only 0.14, indicating high precision in object positioning relative to the ground truth.

A key metric of interest is the High-level Object Tracking Accuracy (HOTA) [36], which evaluates the tracking algorithm's capabilities by integrating detection precision, association, and localization into a singular value. It addresses the assignment problem by exploring the full range of matching thresholds and computing the integral across this spectrum. The results particularly favor the Lidar-only solution due to its reduced localization error, whereas camera usage might result in less precise positioning in certain scenarios, thereby affecting the metric negatively. However, camera-based systems excel in object description and characterization, thanks to the rich semantic content of images and facilitating easier integration into existing infrastructure. Additionally, the impact of errors introduced by image data is minimal.

The multi-station setup emerged as the superior approach,

TABLE I

EVALUATION METRICS OF THE PROPOSED PIPELINE ON MULTIPLE CARLA RUNS

Metric	V1	V2	V3
True Positive	20157	343716	47559
False Positive	315	45427	8218
Missed	89	6177	11098
Association Mismatch	3852	65586	1953
Total Objects	20246	349893	58657
Detection Accuracy (detA)	0.9804	0.8695	0.7112
Localization Accuracy (locA)	0.8552	0.7840	0.7751
Association Accuracy (assA)	0.8702	0.7668	0.7829
Detection Recall (DetRe)	0.9956	0.9823	0.8108
Detection Precision (DetPr)	0.9846	0.8833	0.8527
Detection F1 Score (DetF1)	0.9901	0.9302	0.8312
Association Recall (AssRe)	0.8081	0.7226	0.8494
Association Precision (AssPr)	0.9965	0.9796	0.9919
Association F1 Score (AssF1)	0.8925	0.8317	0.9152
Mean Object Tracking Precision (MOTP)	0.1448	0.2160	0.2249
Mean Object Tracking Accuracy (MOTA)	0.7898	0.6651	0.6374
High-Level Object Tracking Accuracy (HOTA)	0.8055	0.6617	0.5949

outperforming the single-station configuration by detecting and tracking a greater number of objects with enhanced accuracy. This can be attributed to the multi-station system's broader field of view and data redundancy, which are instrumental in filtering false detections and tracking partially occluded subjects.

The evaluation of the proposed algorithm focuses on comparisons across various configuration setups, reflecting our innovative approach in the distributed asynchronous multi-station obstacle detection and tracking scenario. This emphasis is due to the peculiarity of our study; while much of the existing research in this field is dedicated to autonomous vehicles equipped with moving sensors, our work distinguishes itself by concentrating on scenarios involving multiple static stations. This makes it unfeasible to compare the proposed approach with the most common autonomous vehicle pipelines. Indeed, this distinction sets our research apart and addresses this new challenging topic.

V. CONCLUSIONS

In conclusion, the proposed tracking algorithm exhibits high accuracy and robustness in complex, dynamic urban environments when combining data from multiple and heterogeneous sources. Moreover, our algorithm seamlessly integrates 2D and 3D detections, exploiting the ellipsoid representation to fuse the two data types. The experiments highlight the benefits of a multi-sensor approach, showing that single-station setups are less effective in the tracking task, and multi-station systems significantly enhance tracking performance by offering diverse perspectives and mitigating occlusions. Furthermore, the proposed solution addresses the common issue of delayed detections in distributed systems by providing a multi-filter architecture that integrates outdated data into the filter. This work lays the groundwork for further research into distributed multi-station tracking systems. In particular, the validation of the algorithm in a real, controlled environment at the actual Bovisa Campus could further validate the solution's effectiveness, even in the absence of complete ground truth coverage for all tracked subjects.

REFERENCES

[1] E. Cigu, D. T. Agheorghiesei, A. F. Gavriluță, and E. Toader, "Transport infrastructure development, public performance and long-run economic growth: a case study for the eu-28 countries," *Sustainability*, vol. 11, no. 1, p. 67, 2018.

[2] R. Vaiana, G. Perri, T. Iuele, and V. Gallelli, "A comprehensive approach combining regulatory procedures and accident data analysis for road safety management based on the european directive 2019/1936/ec," *Safety*, vol. 7, no. 1, p. 6, 2021.

[3] M. Garcia-Venegas, D. A. Mercado-Ravell, L. A. Pinedo-Sanchez, and C. A. Carballo-Monsivais, "On the safety of vulnerable road users by cyclist detection and tracking," *Machine Vision and Applications*, vol. 32, no. 5, p. 109, 2021.

[4] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier, "Tracking all road users at multimodal urban traffic intersections," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 11, pp. 3241–3251, 2016.

[5] Y. Ni, M. Wang, J. Sun, and K. Li, "Evaluation of pedestrian safety at intersections: A theoretical framework based on pedestrian-vehicle interaction patterns," *Accident Analysis & Prevention*, vol. 96, pp. 118–129, 2016.

[6] A. Martin, *Factors influencing pedestrian safety: a literature review*. TRL Wokingham, Berks, 2006, no. PPR241.

[7] A. Atta, S. Abbas, M. A. Khan, G. Ahmed, and U. Farooq, "An adaptive approach: Smart traffic congestion control system," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 9, pp. 1012–1019, 2020.

[8] M. Bersani, G. Ding, S. Mentasti, S. Arrigoni, M. Vignati, E. Sabbioni, D. Tarsitano, and F. Cheli, "An i2v communication network for driver assistance in public transport," in *2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*. IEEE, 2020, pp. 1–6.

[9] T. Fleck, S. Ochs, M. R. Zofka, and J. M. Zollner, "Robust tracking of reference trajectories for autonomous driving in intelligent roadside infrastructure," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1337–1342.

[10] S. Masi, P. Xu, P. Bonnifait, and S.-S. Ieng, "Augmented perception with cooperative roadside vision systems for autonomous driving in complex scenarios," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1140–1146.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[12] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 10, pp. 2681–2698, 2016.

[13] K. Abdulrahim and R. A. Salam, "Traffic surveillance: A review of vision based vehicle detection, recognition and tracking," *International journal of applied engineering research*, vol. 11, no. 1, pp. 713–726, 2016.

[14] S. Sri Jamiya and P. Esther Rani, "A survey on vehicle detection and tracking algorithms in real time video surveillance," *International journal of scientific & technology research*, 2019.

[15] K. Sage and S. Young, "Security applications of computer vision," *IEEE aerospace and electronic systems magazine*, vol. 14, no. 4, pp. 19–29, 1999.

[16] H. Kim, "Multiple vehicle tracking and classification system with a convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 3, pp. 1603–1614, 2022.

[17] P. Ren, K. Lu, Y. Yang, Y. Yang, G. Sun, W. Wang, G. Wang, J. Cao, Z. Zhao, and W. Liu, "Multi-camera vehicle tracking system based on spatial-temporal filtering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4213–4219.

[18] R. Rios-Cabrera, T. Tuytelaars, and L. Van Gool, "Efficient multi-camera vehicle detection, tracking, and identification in a tunnel surveillance application," *Computer Vision and Image Understanding*, vol. 116, no. 6, pp. 742–753, 2012.

[19] W. Wang, T. Gee, J. Price, and H. Qi, "Real time multi-vehicle tracking and counting at intersections from a fisheye camera," in *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 17–24.

[20] S.-T. Kim, M. Fan, S.-W. Jung, and S.-J. Ko, "External vehicle positioning system using multiple fish-eye surveillance cameras for indoor parking lots," *IEEE Systems Journal*, vol. 15, no. 4, pp. 5107–5118, 2020.

[21] R. Ravindran, M. J. Santora, and M. M. Jamali, "Multi-object detection and tracking, based on dnn, for autonomous vehicles: A review," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 5668–5677, 2020.

[22] A. Kim, A. Ošep, and L. Leal-Taixé, "Eagermot: 3d multi-object tracking via sensor fusion," in *2021 IEEE International conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 315–11 321.

[23] M. Hassaballah, M. A. Kenk, K. Muhammad, and S. Minaee, "Vehicle detection and tracking in adverse weather using a deep learning framework," *IEEE transactions on intelligent transportation systems*, vol. 22, no. 7, pp. 4230–4242, 2020.

[24] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[25] D. Zhao, H. Fu, L. Xiao, T. Wu, and B. Dai, "Multi-object tracking with correlation filter for autonomous vehicle," *Sensors*, vol. 18, no. 7, p. 2004, 2018.

[26] R. Nabati, L. Harris, and H. Qi, "Cftrack: Center-based radar and camera fusion for 3d multi-object tracking," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 243–248.

[27] O. Masoud and N. P. Papanikolopoulos, "A novel method for tracking and counting pedestrians in real-time using a single camera," *IEEE transactions on vehicular technology*, vol. 50, no. 5, pp. 1267–1278, 2001.

[28] F. Camara, N. Bellotto, S. Cosar, D. Nathanael, M. Althoff, J. Wu, J. Ruenz, A. Dietrich, and C. W. Fox, "Pedestrian models for autonomous driving part i: low-level models, from sensing to tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6131–6151, 2020.

[29] M. S. Mahmoud and M. F. Emzir, "State estimation with asynchronous multi-rate multi-smart sensors," *Information Sciences*, vol. 196, pp. 15–27, 2012.

[30] H. Lee, S. Kang, and S. Han, "Real-time optimal state estimation scheme with delayed and periodic measurements," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5970–5978, 2017.

[31] T. Chen, R. Wang, B. Dai, D. Liu, and J. Song, "Likelihood-field-model-based dynamic vehicle detection and tracking for self-driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3142–3158, 2016.

[32] H. Fu, J. Guan, F. Jing, C. Wang, and H. Ma, "A real-time multi-vehicle tracking framework in intelligent vehicular networks," *China Communications*, vol. 18, no. 6, pp. 89–99, 2021.

[33] D. B. Malkoff, "Evaluation of the jonker-volgenant-castanon (jvc) assignment algorithm for track association," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068. SPIE, 1997, pp. 228–239.

[34] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *European Conference on Computer Vision*. Springer, 2022, pp. 1–21.

[35] 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*, 2016, pp. 779–788.

[36] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.