SmartMOT: Exploiting the fusion of HD Maps and Multi-Object Tracking for Real-Time Motion Prediction in Intelligent Vehicles applications

Carlos Gómez-Huélamo[1], Luis M. Bergasa[1], Rodrigo Gutiérrez[1], J. Felipe Arango[1], Alejandro Díaz[1],

***Abstract*— Behaviour prediction in multi-agent and dynamic environments is crucial in the context of intelligent vehicles, due to the complex interactions and representations of road participants (such as vehicles, cyclists or pedestrians) and road context information (e.g. traffic lights, lanes and regulatory elements). This paper presents SmartMOT, a simple yet powerful pipeline that fuses the concepts of tracking-by-detection and semantic information of HD maps, in particular using the OpenDrive format specification to describe the road network's logic, to design a real-time and power-efficient Multi-Object Tracking (MOT) pipeline which is then used to predict the future trajectories of the obstacles assuming a CTRV (Constant Turn Rate and Velocity) model. The system pipeline is fed by the monitorized lanes around the ego-vehicle, which are calculated by the planning layer, the ego-vehicle status, that contains its odometry and velocity and the corresponding Bird's Eye View (BEV) detections. Based on some well-defined traffic rules, HD map geometric and semantic information are used in the initial stage of the tracking module, made up by a BEV Kalman Filter and Hungarian algorithm are used for state estimation and data association respectively, to track only the most relevant detections around the ego-vehicle, as well as in the subsequent steps to predict new relevant traffic participants or delete trackers that go outside the monitorized area, helping the perception layer to understand the scene in terms of behavioural use cases to feed the executive layer of the vehicle. First, our system pipeline is described, exploiting the concepts of lightweight Linux containers using Docker to provide the system with isolation, flexibility and portability, and standard communication in robotics using the Robot Operating System (ROS). Second, the system is validated (Qualitative results[1]) in the CARLA simulator fulfilling the requirements of the Euro-NCAP evaluation for Unexpected Vulnerable Road Users (VRU), where a pedestrian suddenly jumps into the road and the vehicle has to avoid collision or reduce the impact velocity as much as possible. Finally, a comparison between our HD map based perception strategy and our previous work with rectangular based approach is carried out, demonstrating how incorporating enriched topological map information increases the reliability of the Autonomous Driving (AD) stack. Code is publicly available https://github.com/Cram3r95/map-filtered-mot as a ROS package.**

keywords: Multi-Object Tracking, Motion Prediction, HD maps, CARLA simulator, EURO-NCAP evaluation

I. INTRODUCTION

In order to achieve a reliable navigation, Autonomous Driving Systems (ADSs) have to perform safe driving behaviours following conventional traffic rules. In that sense, the the perception layer represents one of the most important modules of an Autonomous Driving (AD) stack, responsible of analyzing the online information, also referred as the traffic situation, through the use of a global perception system [1] which involves different on-board sensors as: Light Detection And Ranging (LiDAR), Inertial Measurement Unit (IMU) , RAdio Detection And Ranging (RADAR), Differential-Global Navigation Satellite System (D-GNSS), Wheel odometers or Cameras. Regarding this, one of the most fundamental tasks in perception systems for AD is track the most relevant obstacles (traffic participants) around the vehicle, also known as Multi-Object Tracking. A power-efficient (regarding computational resources) and real-time MOT system is essential for self-driving applications, representing in most cases the preliminary stage before predicting the subsequent future trajectories of these obstacles in the scene, giving the car a valuable reaction time to avoid critical situations or to anticipate its behaviour for the corresponding traffic scenario.

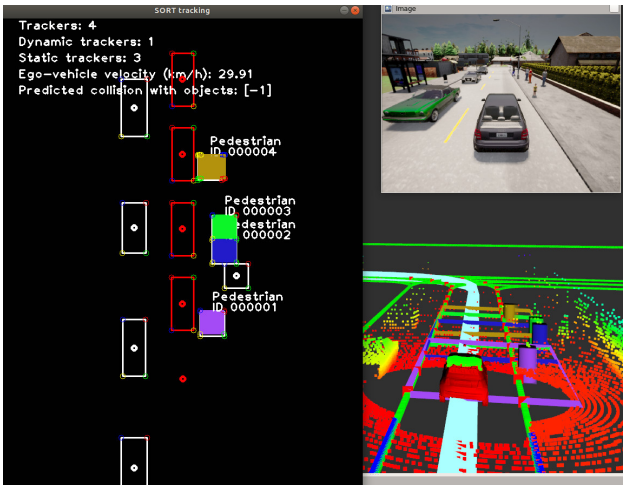

Fig. 1. Simulation example of SmartMOT: On the left, Bird's Eye View (BEV) perspective of the scene with the ego-vehicle prediction in red, tracked objects in colour and non-relevant detections in white. On the right, from top to bottom, RGB camera attached to the vehicle and R-VIZ simulator with the corresponding tracked obstacles and monitorized areas.

MOT systems aim to estimate the position, orientation and scale of all objects in the field of view of the vehicle over time. While object detection only captures the information of the environment in a single frame, a tracking system, which

[1]Carlos Gómez-Huélamo, Luis M. Bergasa, Rodrigo Gutierrez, J. Felipe Arango and Alejandro Díaz and are with the Electronics Department, University of Alcalá (UAH), Spain. {carlos.gomezh, rodrigo.gutierrezm, juanfelipe.arango, alejandro.diazd}@edu.uah.es, luism.bergasa@uah.es

[1]SmartMOT: https://cutt.ly/uk9ziaq

actually represents the next stage in the perception layer, must consider temporal information, filtering outliers (also referred as false positives) in consecutive detections, being robust to full or partial occlusions. Then, after tracking the most relevant obstacles in the environment (both static and dynamic), the vehicle can use this evolution of the scene over time to infer motion patterns and driving behaviour for trajectory forecasting (last stage of the perception module).

On the other hand, for many years maps have helped human drivers to conduct better decisions throughout the navigation with non-automated vehicles [2]. These maps (digital or physical approach) allow drivers to understanding the relationship between the surrounding environment and their own vehicle, in addition to assisting in routing and navigation tasks. Furthermore, with the advancements in ADs, maps play an even more crucial role: Unlike human drivers, where these abstract maps are complementary to the driving skills, such as experiences, senses and judgement, autonomous vehicles require rather more detailed maps in order to be useful for the other layers in the vehicle, such as the planning or perceptions layers, which can use the most relevant lanes around the ego-vehicle based on traffic-rules to perform better decisions. Maps may provide a trusted baseline where the reliability of the sensor suite cannot be guaranteed. The scope of this paper is to fuse the concepts of tracking-by-detection [3] and semantic information of HD maps, in particular using the OpenDrive [4] format specification to describe the road network's logic, to design a real-time and power-efficient Multi-Object Tracking pipeline which is then used to predict the future trajectories of the obstacles assuming a CTRV (Constant Turn Rate and Velocity) model. To the best of our knowledge, SmartMOT is the first tracking-by-detection pipeline that uses the OpenDrive HD map geometric and semantic information to track and predict the most relevant obstacles of the environment, exploiting the concepts of lightweight Linux containers using Docker to provide the system with flexibility, isolation and portability, and standard communication in robotics using the Robot Operating System (ROS), as a preliminary step before implementing the architecture in our real-world autonomous electric vehicle using an AI embedded system for autonomous machines, such as the NVIDIA AGX Xavier.

The remaining content of this work is organized as follows. The next section presents a review of the tracking-by-detection and HD maps paradigms, covering both concepts from the perception later perspective. Section 3 presents our system pipeline, SmartMOT, illustrating the integration of the HD map information, ego-vehicle status and BEV detections around the vehicle. Section 4 describes the Euro-NCAP based protocol used to validate our architecture in an Unexpected Vulnerable Road User (VRU) scenario, which is adapted to the CARLA [5] simulator. Section 5 shows the qualitative and quantitative results of our architecture in this particular scenario, as well a comparison between our tracking-by-detection module using a rectangular based approach, tracking all objects around the ego-vehicle and predicting the incorporation of the traffic participants in the road using a naive rectangular area, and our proposal based on HD map information. Finally, section 6 deals with the future works and concludes the paper.

II. RELATED WORKS

One of the crucial tasks that Intelligent Vehicles must face during navigation, specially in arbitrarily complex urban scenarios, is to predict the behaviour for moving objects [6] [7], being high-fidely maps widely adopted to provide offline (also known as context) information. Recent learning-based approaches [8] [9] [10] [11], which present the benefit of having probabilistic interpretations of different behaviour hypotheses, requiring to build a representation to encode the trajectory and map information. [8] assumes that detections around the vehicle are provided and focuses its work on behaviour prediction by encoding entity interactions with ConvNets. Intentnet [11] proposes to jointly detect traffic participants (mostly focused on vehicles) and predict their trajectories using raw LiDAR pointcloud and rendered HD map information. PRECOG [12] aims to capture the future stochasiticity by flow-based generative models. Furthermore, MultiPath [9] uses ConvNets as encoder and adopts pre-defined trajectory anchors to regress multiple possible future trajectories. As observed, recent Deep Learning based techniques use relatively complicated filters to predict, in an accurate way, the spatial features of the obstacles in the scene, increasing the complexity and computational cost of the system. On the other hand, traditional methods for behaviour prediction are rule based, where multiple behaviour hypothesis are generated based on constraints from the road maps. Road maps present some clear advantages over sensors: They have "infinite range", so they can extract information even into occluded areas. Second, they do not fail under challenging environmental conditions, such as intense fog or rain. Third, recent HD maps contain highly refined data (in which many hours or days of human verification and preprocessing to reduce noise and uncertainty), quite useful to perform safe navigation. Then, HD maps can be an additional sensor that cannot fail unless the road infrastructure changes, providing meaningful, accurate and useful information in real-time operation. Main uses of maps for autonomous driving are related to the information it can be retrieved: Topological, geometric or semantic information. Regarding AD applications, topological information is mostly focused in the network of roads, useful for the planning layer to traverse the most energy-efficient route. Geometric information is used to accurately representing the geometry of the objects around the vehicle, being distinguished in static (inmmovable objects, permanent obstruction), temporary (physical objects that exist in a location for a limited amount of time) and dynamic (moving vehicles, people or objects). Finally, semantic information includes lane information, road classification and road speed limit, including the relational information, such as where vehicles can and cannot turn, where vehicles must stop and how lanes work together.

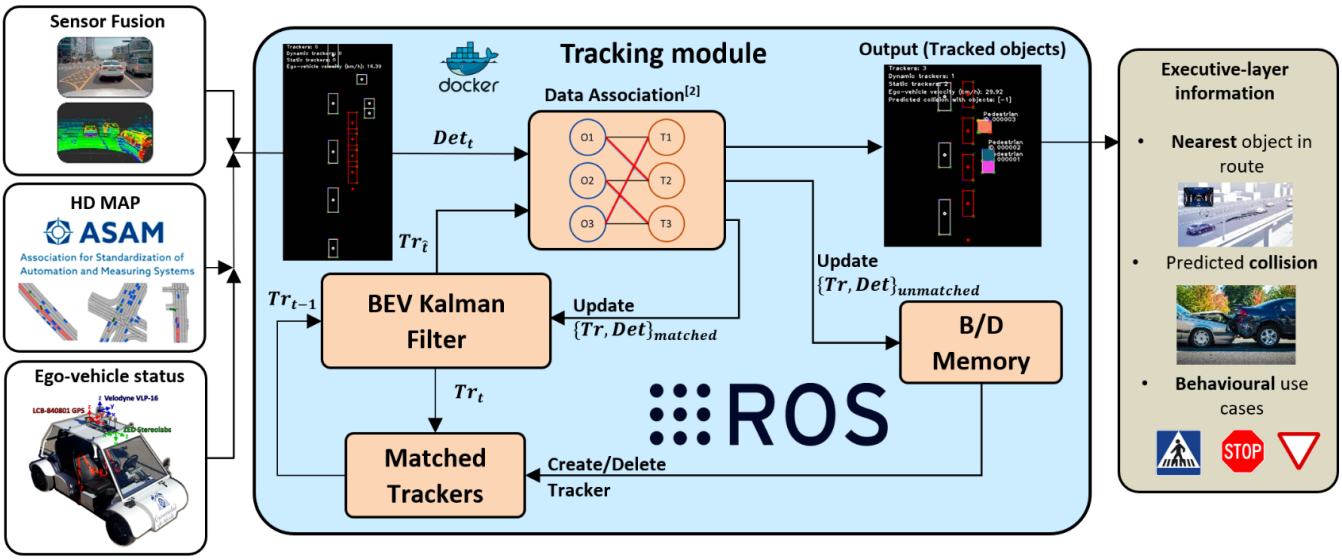

Fig. 2. **BEV MOT system pipeline**: **(1)** Detection module, planning layer and localization layer provide the BEV detected bounding boxes at frame *t* from the raw sensor data, monitorized lanes and ego-vehicle status data respectively using ROS communications, filtering the traffic participants based on the relevant lanes; **(2)** Coordinates are transformed into BEV image plane, so a BEV Kalman filter predicts the state of trajectories in frame *t-1* to current frame $\hat{t}$ throughout the prediction step; **(3)** detections at frame *t* and predicted trajectories at $\hat{t}$ are matched using the Khun-Munkres (a.k.a Hungarian) algorithm; **(4)** matched trajectories are updated based on their corresponding matched detections and the tracker is evaluated again based on its particular monitorized area, to obtain update trajectories at frame *t*; **(5)** Unmatched trajectories and detections are used to delete disappeared trajectories or create new ones respectively; **(6)** Matched predicted trajectories, as well as executive-layer information, are introduced to the system using ROS communications

III. OUR APPROACH

In this work we use [13] as our baseline, in which the MOT problem is approached through a simple yet accurate combination of traditional techniques such as Kalman Filter (KF) [14] and Hungarian algorithm (HA) [15] for state estimation and data association respectively. Nevertheless, though several tracking-by-detection approaches [13] [16] model the state of each obstacle with its 3D position, scale, orientation and their corresponding linear and angular velocity, these approaches usually introduce an unnecessary complexity and computational cost to the system since most traffic scenes can be described in terms of 2D position, angular and linear velocity, apart from the orientation and scale of the resulting bounding box, that is, a Bird's Eye View (BEV), as depicted in Fig. 2. In terms of state estimation, the prediction step is featured by a constant linear and angular velocity, being the unknown accelerations modelled as Gaussian random variables. The input of the update step of the KF is fed with the output of the corresponding sensor suite as observed detections in the BEV space.

In a similar way to the prediction model, the noise associated to the model measurement is featured by Gaussian random variables. Regarding data association between the actual and predicted object detections, we use the 2D Intersection-Over-Union (2D-IoU) in BEV plane instead of using the 3D-IoU version applied in the AB3DMOT baseline [13] and other previous works. The affinity matrix of the HA is then computed using the BEV-IoU between every pair of detection and predicted trajectories. Moreover, we exploit the concepts of standard communication in robotics using the Robot Operating System (ROS) [17] and lightweight Linux containers for consistent software development and deployment using Docker [18]. For more details about the BEV MOT architecture and mathematical expressions, we refer the reader to our previous work [3].

The core interest of this paper is the incorporation of HD map semantic and geometric information, in addition to the ego-vehicle status, to the system proposed in [3]. As observed in Fig. 2 and Fig. 1, once the most energy-efficient route is calculated, the planning layer calculates the most relevant lanes around the vehicle (also referred as monitorized lanes) to filter the obstacles, for example, not considering the VRUs that are inside the sidewalk far away the road or the vehicles that are located in a lane in which lane change is not allowed, as observed in Fig. 1. For this task, the OpenDrive standard was considered in order to satisfy the requirements of this work. OpenDrive represents a standard that allows to describe with the road map with splines with a high precision, as well as any other significant element described in the standard or even customized by the user, presents a variety of information in terms of metadata, quite useful to identify the roles of the lanes (current, back, lane change is allowed, merging, split, etc.) and an availability of the center of the lanes to facilitate path planning calculation. In our architecture, we make use of the C++ library *libcarla* that facilitate to work with this map standard via a PythonAPI that wraps all the dependencies. Using the concept of monitorized lanes help us to increase the reliability and robustness of the system since it is tracking all objects in the environment, which would escalate the computational cost in an arbitrarily complex urban scenario. Nevertheless, VRUs, like pedestrians or cyclists, are usually

difficult to predict, so we enlarge the monitorized area a certain threshold t to the sidewalk so as to track the closest VRUs to the road, as observed in Fig. 1. With this approach we are able to estimate the velocity of the objects, differentiating between dynamic and static obstacles once the velocity estimation is compensated with the ego-vehicle velocity, provided at the beginning of the workflow. For all dynamic participants, we predict their future trajectory based on their velocity in the short-term, with a minimum period of 3 s. Then, with this hypothesis, if the VRU is getting closer to the road, we estimate its global velocity, then its future trajectory at least 3 seconds ahead, predicting its position in the road using a CTRV (Constant Turn Rate and Velocity) model. Moreover, based on the ego-vehicle status, we also generate a trajectory prediction of our own vehicle, being able to estimate a predicted collision between the ego-vehicle and the traffic participants carrying out the 2D IoU between the corresponding trajectories. If this metric is greater than a given threshold, the perception layer sends a signal to the decision-making layer to suddenly stop the car. Otherwise, if a traffic participants is inside the route but the ego-vehicle forecasted trajectory does not collide with the traffic participant, the velocity is adjusted in a similar way to the well-studied Adaptive Cruise Control (ACC) [19].

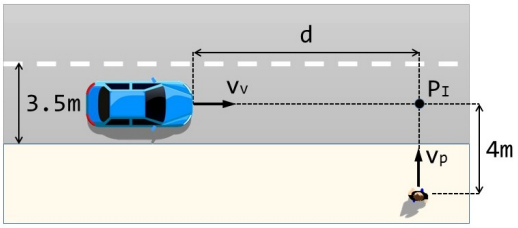

Fig. 3. Car to Pedestrian Nearside Adult (CPNA) scenario

IV. Euro-NCAP based VALIDATION PROTOCOL

A considerable amount of research works and studies, related to pedestrian detection and collision avoidance behavior are present in the literature, where the main objective is to validate the perception and control modules. Nevertheless, as state before our goal to demonstrate how incorporating HD map information helps the whole AD stack to anticipate faster the behaviour of the traffic participants in the corresponding traffic scenarios. Then, common metrics for all frameworks must be used to evaluate the whole architecture, where all modules are integrated. Regarding this, New Car Assessment Programs (NCAPs) protocols are introduced, evaluating the safety of vehicles for different traffic situations and Advanced Driver Assistance Systems, such as Child Occupant Protection (COP), Speed Assist Systems (SAS) or Autonomous Emergency Braking (AEB). Euro-NCAP [20] is introduced in 1997, representing the widely most adopted performance assessment within the scope of the collaboration of European Union countries. China New Car Assessment Program (C-NCAP) [21] is presented (2006) as a research and development benchmark for vehicle manufacturers in Asia, being most of its protocols based on Euro-NCAP. National Highway Transportation Safety Administration (NHTSA), funded in 1970 as an agency of the Department of Transportation of United States, published [22] its guidance documents and regulations on vehicles equipped with ADAS. As observed, these programmes do not present specific protocols in order to evaluate AD stacks, presenting noticeable differences, such as different scenarios, parameters and evaluation metrics. Then, we adop the validation method proposed by [23], which proposes to generalize the VRU v.10.0.3 protocol [24], representing a baseline to compare the performance of different pipelines for the particular situation (both in simulation and real-world) of an Unexpected Vulnerable Road User (VRU) jumping into the road during the navigation, where an Autonomous Emergency Braking (AEB) behaviour must be executed. Fig. 3 illustrates this traffic situation, in which the VRU (a pedestrian in this particular case) starts in the closest sidewalk to the vehicle in a perpendicular position to the vehicle orientation. Once the vehicle starts the navigation and the L2 distance between the ego-vehicle centroid and the VRU centroid is lower than a certain threshold d, the VRU starts its path to unexpectedly cross the road in such a way the ego-vehicle must detect, track and forecast its future trajectory in order to avoid the collision or at least reduce the impact velocity as much as possible. Then, the protocol consists on reproducing the CPNA crash avoidance scenario, with a fixed VRU velocity (v_p) of 5 km/h and a variable ego-vehicle velocity that ranges from 10 km/h to 60 km/h. It is important to note that the threshold d is not fixed, but it is ego-vehicle velocity dependent, that is, the pedestrian must start walking in such a way the impact point (P_I) (Fig. 3) is in the center of the lane for each particular velocity.

Regarding the evaluation metrics, a score for each test is calculated based on the velocity reduction of the vehicle, as following:

- For a vehicle velocity v_v less than or equal to 40km/h:
  - If the vehicle stops without collision, the highest score is achieved:

$$score_{test} = score_{max} \tag{1}$$

  - Otherwise, if the vehicle collides, its score is defined as follows:

$$score_{test} = \frac{v_{test} - v_{impact}}{v_{test}} \cdot score_{max} \tag{2}$$

- For v_v higher than 40km/h:
  - If the vehicle is able to reduce its speed in at least 20 km/h, the highest score is achieve:

$$v_{impact} \leq v_{test} - 20 \rightarrow score_{test} = score_{max} \tag{3}$$

  - Otherwise, if the vehicle collides at a velocity greater than the velocity under test less a threshold of 20 km/h, no score is achieve:

$$v_{impact} > v_{test} - 20 \rightarrow score_{test} = 0 \tag{4}$$

Finally, the final score of a particular pipeline is given by the arithmetic mean of the results obtained in each CPNA crash avoidance test for different weather conditions. For further details about the validation protocol, we refer the reader to [23].

V. EXPERIMENTAL RESULTS

In this section we obtain some interesting both qualitative and quantitative results, evaluating our AD stack [19] in the CPNA crash avoidance scenario using two different perception layer strategies. On the one hand, we implement the perception module stated by [3] which tracks all objects in the environment regardless their topological information and considers a naive velocity dependent rectangular monitorized area in front of the vehicle to determine the distance to the nearest object in the route as well as to predict the collision. On the other hand, we use SmartMOT to track and predict the future trajectories of only the most relevant obstacles around the vehicle, that is, those in which the human in manual driver should pay attention throughout the route, such as VRUs close to the road, vehicles in intersections and lanes where the lane change maneuver is allowed, etc. Qualitative results may be found in the following play list SmartMOT [2], where the SmartMOT performance is illustrated. Regarding urban environment complexity, in order to validate a whole AD architecture the system must be tested in countless environments and scenarios, which would escalate the cost and development time exponentially with a physical approach. Considering this, the use of photo-realistic simulation (virtual development and validation testing) and an appropriate design of the driving scenarios are the current keys to build safe and robust AV. In our work we propose the use of CARLA (CAR Learning to Act) [5] as the best open-source simulator to reach our goals, taking even more importance when analyzing the behaviours the vehicle can face in these complex traffic scenarios. One of the best advantages of CARLA is the possibility to create ad-hoc urban layouts, useful to validate the navigation architecture in challenging driving scenarios. This code can be downloaded from the ScenarioRunner repository, associated to the CARLA GitHub. The ScenarioRunner is a module that allows the user to define and execute traffic scenarios for the CARLA simulator. In the present case, we define several scenarios according to the CPNA crash avoidance traffic situation, modifying the velocity of the ego-vehicle and the presence of other traffic participants. All test were carried out in a PC desktop (Intel Core i7-9700k, 32GB RAM with CUDA-based NVIDIA GeForce RTX 2080 Ti 11GB VRAM), using the version 0.9.10.1 version of CARLA as well as the corresponding ROS Bridge, responsible of communicating the CARLA environment with our ROS-based architecture, and ScenarioRunner modules. In particular, we make use of the OpenScenario standard, supported by ScenarioRunner, where both the VRU and ego-vehicle features can be modified to accomplish the Euro-NCAP requirements. Due to size constraint of this paper, we do not validate the performance of our architecture for different weather conditions but only in daytime conditions. In order to appreciate the behaviour of the vehicle during navigation, we incorporate a very illustrative temporal diagram (Fig. 4), representing a powerful manner to qualitatively validate how the architecture behaves in an end-to-end manner, since we can observe how the car behaves considering the different actions and events [19] provided by the executive layer, which is actually the output of the whole architecture before sending commands to the motor. As observed, the ego-vehicle starts far away from the adversary and starts its navigation. At second 22 a pedestrian that is in the sidewalk is detected, so tracking-by-detection and subsequent motion prediction must be carried as fast as possible to avoid collision, since the scenario is designed in such a way that the pedestrian must start walking in such a way the impact point (P_I) (Fig. 3) is in the center of the lane for each particular velocity. After that, our prediction module intersects the ego-vehicle forecasted trajectory and the pedestrian forecasted trajectory. If the Intersection over Union (IoU) is greater than a threshold (in this case, 0.01), a *predictedcollision* flag is activated and the low-level (reactive) control, which always runs in the background of the decision-making layer, performs an emergency break until the car is stopped in front of the obstacle. Navigation is resumed once the obstacle leaves the driving lane. Table I compares the performance of the architecture by implementing [3] and a rectangular monitorized lane to retrieve the nearest object in route and predict collision against our proposal, where it can be appreciated that for velocities greater than 40 km/h, using HD map semantic and geometric information gives the car a valuable reaction time to anticipate the VRU behaviour and avoid the collision, achieving the highest score.

TABLE I

COMPARISON OF OUR TWO DIFFERENT PERCEPTION STRATEGIES IN THE CAR TO PEDESTRIAN NEARSIDE ADULT (CPNA) SCENARIO. WE BOLD THE BEST SCORE IN **BLACK**

CPNA					
v_{test}	$score_{max}$	**Rectangular area + [3]**		**SmartMOT**	
		v_{impact}	*score*	v_{impact}	*score*
10 km/h	1.00	0.0 km/h	1.00	0.0 km/h	1.00
20 km/h	1.00	0.0 km/h	1.00	0.0 km/h	1.00
30 km/h	2.00	0.0 km/h	2.00	0.0 km/h	2.00
40 km/h	3.00	0.0 km/h	3.00	0.0 km/h	3.00
50 km/h	2.00	23.82 km/h	2.00	0.0 km/h	2.00
60 km/h	1.00	44.23 km/h	0.00	0.0 km/h	1.00
Total	10.00		9.00		**10.0**

Fig 5 shows different analysis of the CPNA crash avoidance scenario with variable ego-vehicle and the incorporation of other traffic participants in the scenario (5(c) 5(d)). T_0 corresponds with the moment the vehicle either stops or collides, and crosses **x** represent the moment in which the system sends a predicted collision signal to the executive layer, so it is coherent that crosses in tests where the ego-vehicle collides with the VRU are shifted to the right (prediction collision

[2]SmartMOT: https://cutt.ly/uk9ziaq

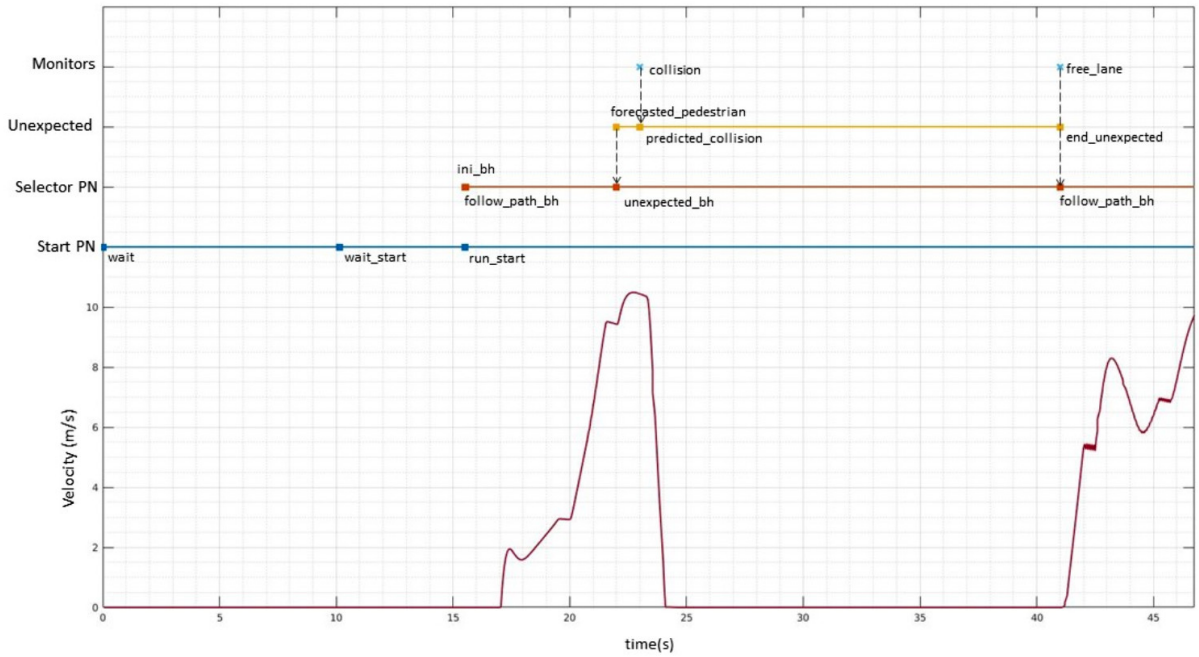

Fig. 4. Unexpected Vulnerable Road User (VRU) temporal diagram. At the top, the events produced by our monitors and map manager modules. In the middle, the selector, and start (background) PNs of our decision-making layer. At the bottom, the velocity of the car throughout the navigation

signal was given in time). Left column tracks all objects around the vehicle and adopts a geometric monitorized area to estimate the nearest distance and predicted collision, whilst right column uses HD map information to help in the Multi-Object Tracking and motion prediction tasks, monitorizing only the most relevant traffic participants around the vehicle that is, SmartMOT. Using HD map information is able to avoid collision until a ego-vehicle velocity of 80 km/h, where SmartMOT is not able to send a signal of predicted collision (output of the system, as shown in Fig. 2) in time, colliding at a velocity of 39.78 km/h. Nevertheless, this velocity at the moment of collision is even lower that the impact velocity (44.23 km/h) when testing the system under 60 km/h condition not using HD map in the MOT stage, illustrating how incorporating additional semantic and geometric map information helps the vehicle to react faster or at least mitigate the effect of collision. Moreover, we simulate both perception strategies using the most common velocities in urban scenarios, which range from 30 to 50 km/h, including 5(c) 5(d) static adversaries (in particular, vehicles and pedestrians) which do not actually in the traffic scenario to fulfill the particular requirements stated by [23] protocol. As expected, tracking all objects around the ego-vehicle and using the rectangular monitorized area suffers when the number of traffic participants is increased around the ego-vehicle, whilst SmartMOT holds this exponential increase by analyzing the objects and their corresponding role as relevant obstacles considering the information provided by the HD map, avoiding the collision in all situations.

VI. CONCLUSIONS AND FUTURE WORKS

This work proposes SmartMOT, a simple yet powerful pipeline that fuses the concepts of tracking-by-detection and information of HD maps, in particular using the OpenDrive standard, to design a real-time and power-efficient Multi-Object Tracking (MOT) pipeline used to track and predict the future trajectories of only the most relevant obstacles around the ego-vehicle, considering their role according to the semantic information provided by the map. This end-to-end pipeline is integrated with ROS for standard communications in robotics and Docker to provide the system with flexibility and isolation. Then, an end-to-end validation of our ROS-based fully-autonomous driving architecture is carried out, obtaining a specific score using the Car to Pedestrian Nearside Adult (CPNA) scenario, testing our proposal against our previous work in terms of tracking and motion prediction, illustrating how the incorporation of HD map information gives the vehicle a valuable time to anticipate the Vulnerable Road User (VRU) behaviour or at least mitigate the collision. We hope that our distributed pipeline can serve as a solid baseline on which others can build on to advance the state-of-the-art in fusing perception data and map information to perform real-time motion prediction in arbitrarily complex urban scenarios. As future works, we plan to incorporate Deep Learning in the MOT and motion prediction stages regarding the paradigm of Multi-Agent interaction, integrated with an enhanced monitorized area and regulatory elements around the vehicle, in order to validate our proposal in more challenging situations to improve the reliability, effectiveness and robustness of our system as a preliminary stage before implementing it in our real autonomous electric car.

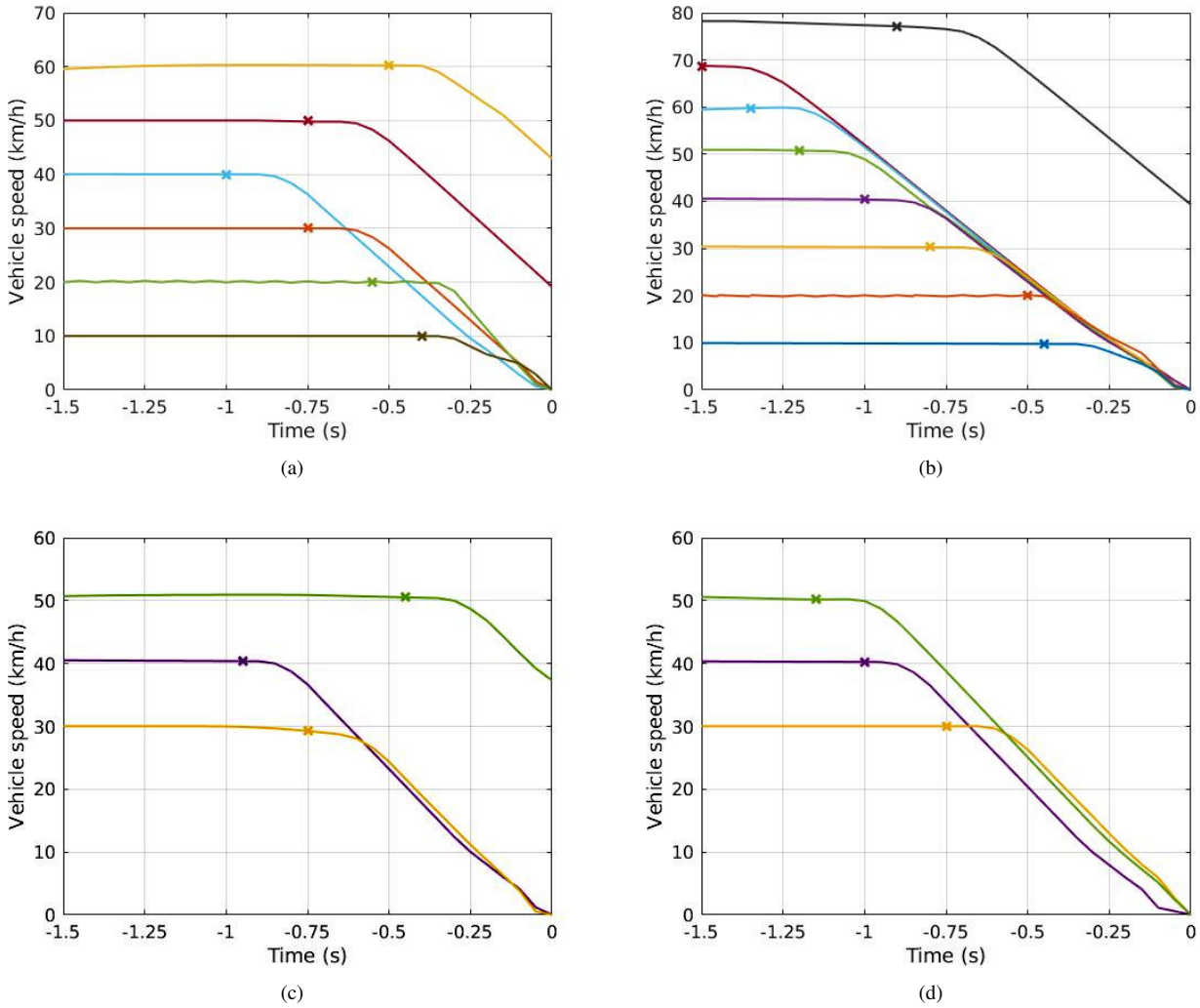

Fig. 5. Analysis of the Car to Pedestrian Nearside Adult (CPNA) crash avoidance scenario with variable ego-vehicle velocity. Left column (a,c) adopts a rectangular monitorized area to estimate the nearest distance and predicted collision, Right column (b,d) uses HD map information for this purpose. On the other hand, first row shows the scenario without additional traffic participants, second row analyzes the crash avoidance scenario including additional traffic participants to the road, monitorized sidewalk area and non-relevant sidewalk area. Crosses in the lines represent the moment in which the system sends a predicted collision signal to the executive layer

ACKNOWLEDGMENT

This work has been funded in part from the Spanish MICINN/FEDER through the Techs4AgeCar project (RTI2018-099263-B-C21) and from the RoboCity2030-DIH-CM project (P2018/NMT- 4331), funded by Programas de actividades I+D (CAM) and cofunded by EU Structural Funds.

REFERENCES

[1] C. Gómez-Huelamo, L. M. Bergasa, R. Barea, E. López-Guillén, F. Arango, and P. Sánchez, "Simulating use cases for the uah autonomous electric car," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2305–2311, IEEE, 2019.

[2] K. Wong, Y. Gu, and S. Kamijo, "Mapping for autonomous driving: Opportunities and challenges," *IEEE Intelligent Transportation Systems Magazine*, 2020.

[3] C. Gómez-Huélamo, J. Del Egido, L. M. Bergasa, R. Barea, M. Ocana, F. Arango, and R. Gutiérrez-Moreno, "Real-time bird's eye view multi-object tracking system based on fast encoders for object detection," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2020.

[4] M. Dupuis, M. Strobl, and H. Grezlikowski, "Opendrive 2010 and beyond–status and future of the de facto standard for the description of road networks," in *Proc. of the Driving Simulation Conference Europe*, pp. 231–242, 2010.

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[6] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8748–8757, 2019.

[7] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1929–1934, IEEE, 2019.

[8] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8454–8462, 2019.

[9] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[10] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11525–11533, 2020.

[11] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Conference on Robot Learning*, pp. 947–956, PMLR, 2018.

[12] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2821–2830, 2019.

[13] X. Weng and K. Kitani, "A baseline for 3d multi-object tracking," *arXiv preprint arXiv:1907.03961*, 2019.

[14] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[15] H. W. Kuhn and B. Yaw, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart*, pp. 83–97, 1955.

[16] H.-k. Chiu, A. Prioletti, J. Li, and J. Bohg, "Probabilistic 3d multi-object tracking for autonomous driving," *arXiv preprint arXiv:2001.05673*, 2020.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[18] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.

[19] C. Gómez-Huélamo, J. Del Egido, L. M. Bergasa, R. Barea, E. López-Guillén, F. Arango, J. Araluce, and J. López, "Train here, drive there: Simulating real-world use cases with fully-autonomous driving architecture in carla simulator," in *Workshop of Physical Agents*, pp. 44–59, Springer, 2020.

[20] M. R. van Ratingen, "The euro ncap safety rating," in *Karosseriebautage Hamburg 2017* (A. Piskun, ed.), (Wiesbaden), pp. 11–20, Springer Fachmedien Wiesbaden, 2017.

[21] K. Guo, Y. Yan, J. Shi, R. Guo, and Y. Liu, "An investigation into c-ncap aeb system assessment protocol," in *SAE Technical Paper*, SAE International, 09 2017.

[22] Takács, D. A. Drexler, P. Galambos, I. J. Rudas, and T. Haidegger, "Assessment and standardization of autonomous vehicles," in *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*, pp. 000185–000192, 2018.

[23] R. Moreno, F. Arango, C. Gómez-Huélamo, L. M. Bergasa, R. Barea, and J. Araluce, "Validation method for unexpected pedestrian behaviour in an autonomous vehicle," in *2021 IEEE Intelligent Vehicles Symposium (IV): In submission*, IEEE, 2021.

[24] "Euro ncap assessment protocol - vru - v10.0.3." https://cdn.euroncap.com/media/58230/euro-ncap-assessment-protocol-vru-v1003.pdf. Accessed: 2021-02-10.