Survey on Cooperative Perception in an Automotive Context

Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Rémi Boutteau, Yohan Dupuis

Survey on Cooperative Perception in an Automotive Context

Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Rémi Boutteau, Yohan Dupuis

***Abstract*—The idea of cooperation has been introduced to self-driving cars about a decade ago with the aim to reduce the occlusion caused by other users or the scene. More recently, the research efforts turned toward cooperative infrastructure bringing a new kind of the point of view as well as more processing power. This paper lies in this new field providing a survey that addresses the cooperative environment. We provide an overview of the architectures available to create such a system as well as the challenges introduced by the cooperation. Later, we review the main blocks involved in the perception: localization, object detection & tracking, map generation. Each block is reviewed under the prism of cooperation. We also provide a Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis of the cooperative perception as well as a list of related scenarios alongside experimentations. Finally, we list some related datasets before concluding our paper, underlining the perspectives for further works.**

***Index Terms*—Cooperation, Infrastructure, Vehicle, Localization, Mapping, Object detection, Tracking.**

I. Introduction

THE concept of driverless cars is one of the landmarks of a futuristic world for generations. Already in 1939, General Motors (GM) initiated the first attempt of making this a reality by showcasing a radio piloted car [1]. Since then, the development of this technology has never stopped and is increasingly getting complicated over a wide range of fields such as perception, decision making, and control. After the pioneer works of GM, during the 1980s, Mercedes-Benz showcased the first autonomous car with a vision-controlled robotic van reaching a speed of 63 km/h on streets without traffic. This led to the creation of international projects and challenges such as the Defense Advanced Research Projects Agency (DARPA) Grand Challenge in 2004 consisting of autonomously navigating through the Mojave desert in 142 miles long course [2]. The next step was navigation in an urban environment through normal traffic conditions. In 2007, the DARPA announced the holding of the Urban Challenge that simulates an urban environment with streets, traffic lights, and human-driven vehicles. [3]. We can also note the VisLab Intercontinental Autonomous Challenge (VIAC) challenge in 2010 consisting of driving autonomously through a 13000 km long way from Parma in Italy to Shanghai in China [4]. Nowadays, several companies sell cars with the ability to offer an autonomous driving experience such as Tesla [5] or the Audi A8 [6]. The idea of cooperative vehicles quickly appeared and in 2011 the grand Cooperative Driving Challenge (GCDC) took place in the Netherlands in which vehicles had to perform the best in a platoon [7], [8]. The GCDC has been reiterated in 2016 to perform lane merging, driving in an intersection as well as emergency vehicle handling in a cooperative context [9]. Cooperation between vehicles can be extended to infrastructure and thus led to the project Providentia in Germany [10] consisting in creating a digital twin of a road section generated from the sensors of an infrastructure.

We assume that the purpose of perception is to represent the elements around the ego-vehicle as well as its status in the scene. We distinguish 3 subsections, the localization of the ego-vehicle, the detection and tracking of other users and, finally, the detection and representation of the environment (mapping). Cooperation represents the use of data provided by other agents to perform perception tasks or to refine their results. Cooperation can be performed at three levels of data sharing depending on whether the data is raw (early fusion), preprocessed data (mid fusion), or processed data (late fusion). Fig. 1 represents this pipeline with three steps and three blocks (namely: Localization, Object Detection and Tracking and Map Generation) performing the main perceptive tasks to understand the scene. In the early fusion stage, we represented the raw data fusion. In this stage, the data provided by the sensor at a given timestamp is aggregated and associated with a given transformation between sensors. The raw data come from connected users which perform an early fusion. The raw data from the ego vehicle may also be shared with other users. In the second stage, we note two parallel tasks running. One estimates the vehicle's location in the environment from the sensors and can also benefit from other users' measurements as an aid. The second task performs the detection and tracking of objects in the scene. It can also benefit from the data of connected users to densify the global perception of the environment. Both together perform the heart of the perception outputting feature level data shareable with other users. The last stage aims to build a map, hence giving context to the previously acquired data. It is based on the use of a given prior map and can also be updated cooperatively by connected users. This block diagram tries to briefly showcase the classical scheme of a cooperative Vehicle-to-Everything (V2X) perception pipeline. However, reality offers a broader

A. Caillot (caillot@esigelec.fr) and S. Ouerghi (ouerghi@esigelec.fr) are with Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France.

Y. Dupuis (ydupuis@cesi.fr) is with LINEACT CESI, Paris La Défense, Paris, France

R. Boutteau (remi.boutteau@univ-rouen.fr) is with Normandie Univ, UNIROUEN, UNILEHAVRE, INSA Rouen, LITIS, 76000 Rouen, France.

P. Vasseur (pascal.vasseur@u-picardie.fr) is with Laboratoire MIS, Université de Picardie Jules Verne, UFR des Sciences, Département Informatique, 80000 Amiens, France.

Manuscript received xxxx; revised xxxx

range of architectures with their specificities and a certain amount of challenges when realizing them, which is exposed later in this survey. This paper aims to provide a state of the art of cooperative perception methods for Vehicle-toVehicle (V2V) and Vehicle-to-Vehicle (V2I). We have organized the paper in an order that respects the data flow, divided in six sections. Section II focuses on the creation of cooperative systems from a general point of view. We particularly review the challenges brought by cooperative systems, the possible architectures, and the available communication facilities. We also present a review of frequently used sensors along with their performances in a non-cooperative environment to provide a reference as a basis for comparison. Section III lists the cooperative methods of locating the ego-vehicle in the scene. Section IV, for its part, reviews the methods of detection and tracking of objects in the scene. Section V reviews the role of maps and their usage in a cooperative context. In Section VI, we propose to summarize the cited techniques through a summary table and we propose a SWOT analysis. In section VII we review the scenarios in which cooperation brings real advantages illustrated by experimentations. Finally, We list the datasets available to unlock work perspectives before providing our conclusion in section VIII.

II. Basics of cooperation

The ways of creating cooperative perception systems are multiple and require to assess several types of architecture. Each design has advantages and disadvantages and will deeply affect how the system will react as well as its strengths and its weaknesses. Another unavoidable point of any cooperative system is the communication facilities which define what data can be shared as well as the formats available. These two points will be tackled in this section but we will start by briefly reviewing the results available in the non-cooperative methods based on the same sensors widely used in the cooperative counterpart to get comparison points.

A. Sensing Modalities

Sensors are the basics of any perception system as they allow us to sense ourselves as well as the surrounding environment. Since the sensors we are going to discuss have already been presented in numerous articles, we will rather focus on their performances. In [11], Kuutti et al. brought a survey introducing the sensors and comparing their performances in a positioning context and therefore inspired the following structure.

1) Global Navigation Satellite System: When it comes to knowing our position, the satellite positioning system is the most widely used. Initiated by the United States with the Global Positioning System (GPS), several countries contributed with new satellite constellations. The pure GPS has an error of up to 20 meters [12] but several methods have been used to refine these results. However, since the GPS has an update rate up to 20 Hz, it is often associated with an Inertial Measurement Unit (IMU) bringing a high updating rate [13]. An association of a pure GPS with an IMU showed they could achieve an error of 7.2 meters (Root Mean Square (RMS)) after a path of 408 meters [14].

One of the problems encountered with pure GPS is its first acquisition time. The Assisted GPS (AGPS) brings a solution to this by using the cellular network to download the almanacs and hence reducing the downloading time from the slow satellite connection. However, it does not bring any precision improvement. Unlike the AGPS, the Differential GPS (DGPS) allows a reduction of the error up to 1 to 2 meters in the covered zones [15]. The arrival of the Real-Time Kinematic GPS (RTK) achieved unprecedented performance with an error of a few centimeters range [11]. Similarly to the AGPS, both DGPS and RTK do use a terrestrial infrastructure to download the satellites' almanacs via an internet connection. Although we do not consider Global Navigation Satellite System (GNSS) technology as a cooperative system, it is one as it features several vehicles (terrestrial users and satellites) and infrastructures.

Nowadays, pure GPS had been replaced by the GNSS, currently based on several satellite constellations such as the American GPS, the Chinese BeiDou Navigation System, the Russian Global Navigation Satellite System (GLONASS), the Japanese Quasi-Zenith Satellite System (QZSS) and the European Galileo. Using the Real-time extended (RTX) technology, the Root Mean Square Error (RMSE) achieves a 2.9 cm accuracy [16].

2) Camera: Cameras can be used to detect and track obstacles (pedestrians, cars, animals) as described by Arnold et al. [17]. Formerly, these tasks were mostly based on a geometric approach to the problem, but the machine learning and deep learning methods have taken over the state-of-the-art. Hence, nowadays, most efforts are based on machine learning solutions.

Another field of application for cameras is trajectory estimation, especially with visual odometry [18]. This technique consists in recognizing key points in a frame and then finding them in the following frames to estimate the displacement of the camera. We can note that this method is sensitive to error accumulation over time. This principle is extended in the Simultaneous Mocalization And Mapping (SLAM) algorithms with the difference that the perceived environment is kept to create a map and estimate its position with an accuracy of 75 cm [19]. Nowadays, new methods featuring Deep Learning bring even better results such as DeepSLAM proposed by Li et al. in [20] which gives a mean translation RMSE drift of 5.58% and a mean rotational RMSE drift of $2.47°/100m$ alongside a 100 m to 800 m path. Since visual odometry and SLAM are based on the notion of optical flow, the arrival of event-driven cameras with hardware adaptation offers promising results in both localization and classification.

However, monocular systems pose a limitation on the estimation of the position on the depth axis of the images. One solution is to use two or more cameras to create a stereoscopic vision system and to synchronously search in both cameras for corresponding interest points. Another solution to get the depth information from a monocular system is to use a deep learning algorithm [21]. In addition to these techniques, Camera-Light Detection and Ranging (LiDAR) or Camera-Radar coupling

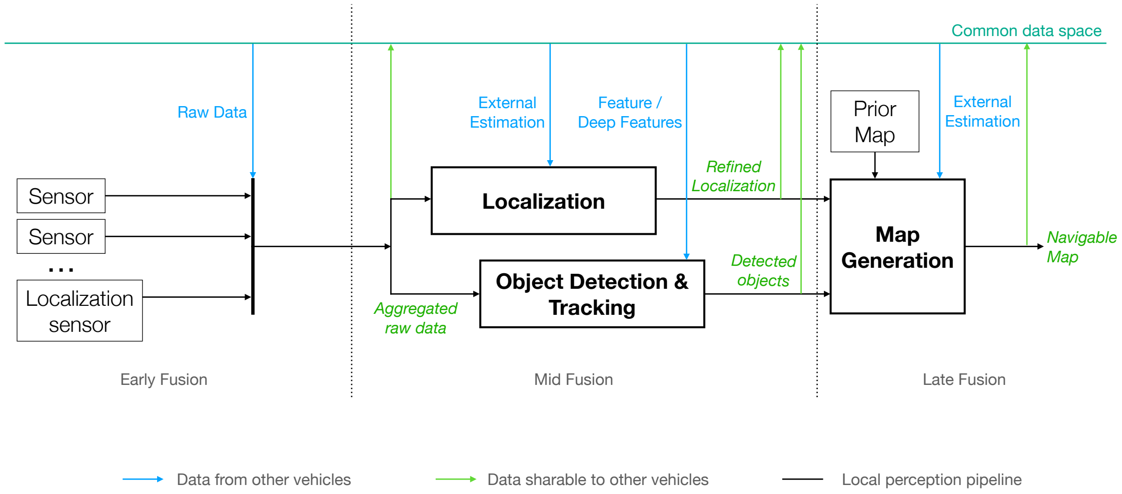

Fig. 1: Block diagram of the minimal perception pipeline in a vehicle (in black). We can distinguish three main stages able to share the locally produced data (in green). Each of them can receive data (in blue) to perform their task cooperatively.

has been extensively investigated in the state of the art.

3) Radio Detection and Ranging (RADAR): Compared to cameras, radars have a lower angular resolution. This characteristic makes them less suitable for the classification of perceived objects. However, their accuracy in distance and speed measurements is much better than cameras and they are therefore used in addition to the latter as in the Providentia project [10].

The concept of visual odometry has been adapted to the radar device. A high-speed rotating radar has allowed a position estimation with an error of 12 meters despite the distortions due to the rapid rotation [22]. Another system using Short Range Radar (SRR) allowed an estimation with an RMS error of 7.3 cm on the lateral axis and 37.7 cm on the longitudinal axis in [23]. In the same way with the SLAM, an experiment allowed a localization with a mean error of 9 cm and a standard deviation of 38 cm [22]. Nevertheless, radars can penetrate certain materials, notably those that compose the ground. Thus, a method based on the mapping of underground terrain has allowed a localization with a precision of 4 cm and is presented in [24]. Despite advantages such as insensitivity to weather conditions, the authors specify that further researches are needed to create robust maps to multi-path effect or to characterize reflections induced by vehicles chassis.

4) LiDAR: LiDARs (Light Detection And Ranging) can be considered as an intermediary between radar and camera. They provide a list of points in a three-dimension space. These points are extracted from the angle formed by the laser beam and the distance from the sensor and the impact. To get the distance, there are several techniques. The most common one is based on the Time of Flight (ToF) principle, but we can cite other techniques such as the Frequency Modulated Continuous-Wave (FMCW) or the Amplitude Modulated Continuous-Wave (AMCW) [21]. Since the angular resolution is thinner than the radar, we can classify detected objects besides being able to locate them more accurately [25], [26].

In the same way as what we have seen with previous sensors, the principles of visual odometry and SLAM can be adapted to LiDAR sensors. In [27], a GPS, IMU and wheel odometry have been combined within a SLAM framework that allowed a localization estimation error between 10 and 30 cm. An improvement of the SLAM and an implementation of dynamic maps allow an error of 9 cm in a dynamic environment [28]. By projecting the ground on a grid invariant to the laser perspective, a position estimation with an RMS error of 3.3 cm on the longitudinal axis and 1.7 cm on the lateral axis has been performed in [29].

Halfway between cameras and LiDARs, ToF cameras, made of a sensor similar to cameras are based on measuring the ToF taken bay the light to return to the sensor. They provide depth images that can be related to point clouds generated by the LiDARs. By using them in a visual odometry algorithm, Chen et al. were able to estimate the trajectory with an absolute trajectory error (ATE) of 78 cm on a 25-meter path [30].

5) Ultrasonic: The majority of vehicles sold today carry ultrasonic sensors. The drawback of such sensors is that they have a very low angular resolution that requires a too important calculation cost. Also, they are highly sensitive to weather conditions and the Doppler effect when objects are moving fast and have a short-range [11]. These elements make this sensor unsuitable for applications of obstacle localization and classification.

6) Radio Frequency (RF) based methods: Wireless communications are mandatory in a cooperative environment hosting mobile users. However, they can be used as sensors, especially to estimate the position of a receiver. Various sources of radio signals can be used, such as the cellular network or infrastructure made up of anchors, as in the case of Ultra Wide Band (UWB) systems allowing centimeter-scale location [31].

Position estimation methods are generally based on measuring the distance between the transmitter and the receiver. Thus there are four main methods for position estimation :

- Received Signal Strength Indication (RSSI): RSSI based method that consists of measuring the signal strength to measure the distance between the transmitter and the receiver based on the electromagnetic permeability and the diffusion factors of the environment. A distance measurement allows us to position ourselves on a circle surrounding the transmitter base, but, as shown in Fig. 2, it is impossible to know where on this circle. To eliminate the ambiguity, it is necessary to make at least three measurements to find the common intersection of the three circles.

- Time Of Arrival (TOA) and Time Difference Of Arrival (TDOA): These methods that use the transmission delay of a signal between its emission and its reception. Since the speed of an electromagnetic wave is known, it is possible to find the distance between the two devices. In the same way as the RSSI-based method, at least three measurements are necessary to estimate the position of the receiver.

- Angle Of Arrival (AOA):, Unlike the other two methods, AOA method, is based on measuring the angle formed by the direction of the received signal. This angle associated with the position of an anchor forms a straight line on which the vehicle is located. With a second measurement on another anchor, a second straight line is obtained which intersects the first one at the position of the vehicle as illustrated in Fig. 3.

- Fingerprint: This method is based on the specificity of the environment and in particular on its capacity to alter the strength of a signal and to reflect it (multi-path). The aggregated information is compiled into a map allowing us to match the received signals to a position.

Typical setup: The listed sensors succeed to achieve their tasks but also suffer from shortcomings. Therefore, sensor fusion is mandatory to get over the limitations of each one. We already mentioned the fusion between a GNSS receiver and an IMU to improve the localization performance. Similarly, vehicles or infrastructures embed several types of sensors. A usual setup for autonomous cars is constituted of GNSS - IMU to achieve global localization with cameras, laser scanners or RADARs for detection and tracking of elements in the scene or as another source of localization information. Infrastructure also embeds sensors such as cameras and laser scanners or RADARs to locate users as seen in [9], [10].

B. Communication

In the previous section, we have reviewed the most used sensor in an automotive context. In a cooperative context, we want to share the generated data, raw or processed, with other agents with the aim to densify the image of a covered area. Thus, it is mandatory to discuss the communication facilities available, which is the aim of this section. We will focus on the ways to wrap the data they produce and how to share them. Then, we present some of the most widely used communication facilities. We also consider new approaches.

1) Wrapping and sharing the data: To share data, users have to choose a specific network architecture. The most common is the Vehicular Ad-hoc Network (VANET) architecture consisting in connecting every vehicle in the range from each other [37]. In VANET, a channel is common to every vehicle to coordinate the network. The data is shared on different channels and routed by hopping on vehicles between the sender and the receiver. To assess the physical layer's requirement in a VANET network, an amendment of the IEEE 802.11 was added to create Wireless Access in Vehicular Environments (WAVE) (IEEE 802.11p). In Europe, the IEEE 802.11p standard was used to create the ITS-G5 standard [38]. In the same way, two communication protocols are based on these two standards which are respectively the Dedicated Short-Range Communication (DSRC) [39] and the Cooperative-ITS (C-ITS) [38]. Table. II gives an overview of both of the standards and their components compared to the OSI model as given in [38], [39]. We can note the presence of Basic Transport Protocol (BTP) and GeoNetwork which are defined in [38] as well as WAVE Short Message Protocol (WSMP), defined in [39] as facilities to achieve the network and transport layer tasks. The specificity of the GeoNetwork protocol is that it bases itself on the geographical position of the agents to determine the path to follow for the data.

The information shared with DSRC protocol are wrapped in Basic Safety Messages (BSM) [39] which convey information about the emitting vehicle to avoid collisions. Similarly, C-ITS introduces the Cooperative Awareness Messages (CAM) also conveying vehicle information as the BSM but also introduces the Distributed Environment Notification Messages (DENM) which notify hazards on the road and which has a higher priority than the CAM [37]. CAM and DENM messages proposed with C-ITS are used by [9] but the authors also needed to use another type of message, the i-GAME Cooperative Lane Change Message (iCLCM), to indicate to other vehicles their willing to change lane. Authors in [25] used the Signal Phase and Timing (SPaT) messages to anticipate the traffic light changes and used the DSRC's BSM to notify the presence of detected vehicles by the infrastructure. To respond to these new needs, messages such as SPaT but also the messages for road topology data (MAP), for special vehicles (SRM, SSM), for probe vehicle data (PVD, PDM), and in-vehicle information (IVI) are being standardized [38].

Novel network architecture is used by Li et al. in [40]: the Software-Defined Network (SDN). This solution is placed between the VANET and the fully centralized network. The common network is thereby replaced with centralized archi-

Sensor	Given data	Environement's impact	Advantage	Disadvantage	Performances
GNSS	Absolute position	Requires at least 4 satellites in sight of view and is sensitive to the canyoning effect in urban environment.	The system doesn't require an initial position to give a result and can be used in an unknown environment.	The result is outputted once per second and the reliability of the signal depends on the services coverage.	Pure GPS: 20 m Pure GPS + IMU: 7.2 m error GNSS RTX: 2.9 cm
IMU	Relative position	The system is not affected by the environment.	Ability to output a result at a higher frequency than GNSS.	The error accumulate as the time passes and is affected by the precision. The higher the precision is, the higher the price is.	Estimated bellow 7.1 % Relative Error for MPU-9150 [32]
Radar	Distance and relative speed	Affected by weather conditions (mainly rain but also snow, mist).	Long range perception and hardware speed measurement possible	Poor angular resolution making object classification harder	Angular accuracy: 0.5° to 5°; Speed accuracy: $0.2ms^{-1}$; Perception range: up to 250 m; Sampling rate: up to 20 Hz
LiDAR	Point cloud	Affected by weather conditions (mainly fog but also rain).	Compromise between radar and camera allowing a physical measure of the distance but with a lower angular resolution.	The sparseness of the point cloud makes it hard to difficult to sense the texture.	Angular accuracy: 0.03°; ranging accuracy: 10 cm to 2 cm; Perception range: 80 m to 200 m; Sampling rate: up to 100 Hz
Camera	Image	Affected by weather conditions and brightness.	Sense color and textures facilitating segmentation and classifying.	Although it can be estimated, there is no direct depth measurement.	Highly dependent on the sensor and associated optics.
Ultrasonic	Distance from obstacle	Affected by weather conditions	Low cost sensor	Small detection range and high sensitivity to Doppler effect	Maximum range: 6 m

TABLE I: Sensor comparison based on [12], [14], [16], [17], [33]–[36]

Application	Other App. Layer	Safety App. Layer
Pre-Application	CAM / DENM BSM / SPaT / MAP / SRM / SSM	
Transport	TCP / UDP	GeoNetwork / BTP
Network	IPv6	WSMP
Data Link	ITS-G5	
Physical	WAVE	

TABLE II: Representation of the two protocols available in a VANET architecture given through the OSI model [38], [39]. The the C-ITS defined standard are given in green while the DSRC defined standard is given in blue. Both of them provide adapted answers for vehicular communication on the physical layer based on IEEE 802.11p as well as dedicated messages to encapsulate the data between the application layer and the transport layer.

tecture communicating with a controller which manages the interconnections between the road users dynamically.

Another common architecture used nowadays is based on the publisher / subscriber paradigm, mainly supported by the Robot Operating System (ROS) [41] which is frequently used in recent projects [40], [42]–[45]. The structure is based on nodes communicating messages transmitted on topics. Each node can be a publisher or a listener and they can be placed on different devices on the same network. A master program runs and plays the role of a dictionary and is contacted by every node either to inform about the topic they publish on or to know which node to listen to for a specific topic. Messages transiting through topics and are very various and can contain coordinates, images, or point clouds. A new version of ROS (ROS 2) is being developed with some improvements regarding fleets of collaborative robots.

2) Communication facilities: A wide range of communication facilities has been proposed for tackling different needs. We have already mentioned WAVE and ITS-G5 which are based on Wireless Fidelity (Wi-Fi) (IEEE 802.11) but with a given frequency of 5.8 GHz in Europe as well as in Japan and 5.9 GHz in USA [46]. Authors of [47] used the IEEE 802.11p to establish a communication between infrastructure and a vehicle and used DENM to transmit the control messages and the position information. Chen et al. [26] similarly used DSRC, and thus WAVE, to share regions of interest of LiDAR point clouds and indicate sufficient speed. Kim et al. [42] used Wi-Fi IEEE 802.11n and studied the impact of the delay on the position estimation error.

Even if the majority of the current solutions are based on IEEE 802.11 technology and its derivatives, other technologies can be used such as the cellular network. The advantage of it is its wider coverage and the already existing infrastructure [48]. 5G cellular network is particularly promising thanks to its features such as precise localization, high throughput, and low latency. As described in [49], Proviendentia takes the advantage of the 5G network to communicate between the different elements (back-end station, Road Side Unit (RSU), On-Board Unit (OBU)).

Emerging communication technologies are being explored by authors of [40] who used the Millimeter Wave (mmWAVE) [50] band to transmit the point cloud produced by the RSU to the OBU and noted a significant data throughput increase. Another technology studied is the Visible Light Communication (VLC) [51] which consists of using light-emitting diode (LED) arrays (e.g. traffic lights, car lights) to display patterns. VLC allows data rates up to 96 Mb/s but is sensitive to the environment [11]. Finally, UWB which is used for localization is capable of communication [52] with data rates tested up to 250 Mb/s in [53] and up to 1 Gb/s in [54]. However, to our knowledge, UWB is not used for data sharing in the Intelligent Transportation System (ITS) context.

C. Designs and challenges

Until now, we have reviewed the most used sensors used in the automotive context as well as the communication facilities available to share the generated data between agents. However, when several users interact with each other, we have to define the organization of the communication. We distinguish two main approaches: the centralized and the distributed ones. We discuss and compare these approaches in the next lines. Nonetheless, no matter the chosen approach, cooperation brings new challenges. We provide a review of these challenges following the discussion on organization approaches.

1) Centralized approach: The cooperative approach makes it possible to overcome the problems of non-cooperative approaches such as extending the horizon line. As an example, multiple points of view can be used to reduce the effects of obstructions while densifying the areas covered.

The centralized aspect of this approach concerns the processing of the acquired data. In this mode of operation, users share their acquisitions to a single point, for example, a roadside processing unit. This server is in charge of processing the data and extracting useful information from it, which are then shared with users. The Providentia project is based on this approach. Data acquired by the sensors placed on gantries on a section of highway are transmitted to a roadside processing unit, which creates a digital twin of the section of road accessible to all [10]. Similarly, Lv et al. based their work on a centralized approach in which four LiDARs monitor an intersection and transmit their data to an RSU. Users are detected, located and their information is then relayed to other users [25]. The main disadvantage of this solution is that the efficiency of this architecture relies mainly on the processing power of the processing unit. In [55], Shi et al. proposed a solution to the throughput drop of the service model of [56] by introducing the cluster-based VANET which consists of linking sub vehicular network into a larger one. Data collected by each vehicle of a sub-network are filtered and stored on a server to be broadcast under request which resulted in lower network usage and hence reduced energy consumption.

2) Distributed approach: In contrast to the centralized approach, the data acquired by the users are directly transmitted to all vehicles simultaneously. Therefore the processing of these data is done onboard for each vehicle. A typical case of decentralized management is presented in [9] by Xu et al. through their participation in the GCDC of 2016. Each vehicle was broadcasting its state and its maneuver intentions which allowed the event anticipation and improved the car control. However, the system used connected cars which are in range with each other limiting the size of the network. Li et al. proposed the use of the SDN in [40] to optimize the network usage and set up mmWAVE communication to increase the throughput allowing them to share raw LiDAR point clouds. To solve the problem of disconnection in a sparse fleet, Zheng et al. proposed in [57] the use of the cellular infrastructure to create a heterogeneous network. The common point of these applications is that data of every vehicle is processed onboard on each vehicle. However, the coverage quality depends on the size of the user fleet [58].

3) A comparison: As stated before, both centralized and distributed approaches have several advantages and weaknesses, as shown in table III. We can observe that the distributed approach is the most common because, nowadays, the majority of cooperative applications are based on V2V approaches. However, applications based on centralized approaches are increasingly present today, especially within projects such as MEC-View [59], [60] and Providentia [10].

4) Challenges of cooperation: As we have seen, the architecture of a cooperative system dramatically impacts the efficiency of a cooperative system. However, this is not the only challenging part of cooperative systems. As well as for non-cooperative systems, difficulties brought by the type's diversity from the data acquired from the sensors exists as well as the one from the calibration of the acquisition hardware. But, to them, we have to add other challenges such as the synchronization between the actors, the extreme difference of point of view, or the matching of the receiving data with the locally acquired one.

a) Multi-modality: Multi-modality is the one we are the most aware of since it appeared in the early time of robotic perception. Some projects avoided this problem such as Lv et al. [25] who decided to solely use LiDARs as well Chen et al. in [26] and Li et al. in [40].

Another project trend is to use the different sensors for an application to merge the results to improve the reliability or to enriches the properties of a detected item. As explained by Hinz et al. [49], the Providentia project uses cameras and radars to sense the environment. The choice of multi-modality has been made to answer different needs which are the detection and the classification, performed by the cameras, and the distance and speed measurement, performed by the radars. Later in this paper, we assess the way of merging the streams of data given by the sensors with three different approaches: early fusion, late fusion, and deep fusion.

Another way to solve the multi-modality and calibration challenges is to process the data locally for each sensor and share the output in the form of messages. The above-cited project of Lv et al. [25] uses this principle to share the detected vehicles facilitating the broadcasting with smaller data. However, data association is a challenging topic that must be performed afterward during the aggregation step. The GCDC 2016 offers an answer based on the choice that users broadcast their states and intentions only avoiding duplicate data and assignment tasks. Xu et al. used in [9] a LiDAR to perceive vehicle in front of the ego-vehicle on both lanes by looking for clusters of points. As reported by the authors,

	Distributed	Centralised
Advantages	• Reliable in case of failure of an element • Available everywhere	• More data aggregated • Global view of the scene • More computing power
Disadvantages	• Limited computing power • Network less optimised (duplicated data) • Synchronisation	• Synchronisation • Converging network (Possible bottleneck effect) • Latency between the sensor and the received information

TABLE III: Advantages and disadvantages observed between distributed and centralized architectures.

these clusters could be associated with the messages sent by other vehicles on the map with their coordinates.

b) Calibration: Calibration is the other most known challenge in the perception pipeline. The calibration in a cooperative environment aims to determine the transformation between the sensors to be able to merge acquired data from several views at least at a given frame. If this task can already be challenging on a single agent, it becomes more laborious in a multi-mobile user environment. In this situation, the transformation matrix between sensors constantly changes as the vehicle moves in the scene, featuring long baselines. Moreover, synchronization is arduous due to the absence of a physical triggering line.

Fortunately, to calibrate an infrastructure, manual measures can be sufficient and remain simple to conduct. Lv et al. [25] calibrated their infrastructure by measuring the distance between the four sensors placed at each corner of the intersections.

Similarly, it is possible to semi-automate the calibration process in the same way as the vision calibration with a chessboard. Krammer et al. describe in [44] the calibration procedure of the Providentia project: cameras have been intrinsically calibrated using a chessboard and the radar was calibrated by using the built-in tools based on the vanishing point method. We can note that for a cooperative project, the baselines encountered in the scenes are much wider than the ones met locally on a vehicle. Thus, a similar application might bring an answer to calibrate infrastructures with a large baseline and very different point of view: the Motion Capture (Mo-Cap) systems. Yang et al. give in [61] an example of calibration with multiple Microsoft Kinects v2 synchronized through Network Time Protocol (NTP). The system uses a calibration wand to fix the common origin between the cameras similarly to several other commercial systems (e.g. Vicon, OptiTrack). Unfortunately, the use of a calibration wand will encounter laser scanners or radars limits: their low angular resolution. In [62], Xia et al. propose a state of art for global calibration of non-overlapping cameras. Some of the presented methods could apply to cooperative roadside infrastructures such as the methods based on Structure From Motion (SFM) or the visual measuring instruments consisting in locating landmarks with a known position in the sensor data to recover the position of the sensors.

However, none of these methods helps when mobile acquisition platforms appear. Nowadays, the most widely used method, in this case, relies on absolute coordinates and hence relying on the pose estimation performance assessed in the Perception part.

c) Synchronization: Synchronization is another major challenge to consider. In a cooperative context, calibration relies on the synchronization of the elements to determine the transformation between the sensors, especially with the mobile sensors. There are multiple sources of desynchronization such as an offset between the clocks or the communication delays. Although clocks are synchronized, we cannot ensure their acquisition are triggered at the same moment which adds uncertainty at the moment to merge the acquired data. Similarly, different sampling rates require interpolation between acquired or predicted data, also adding uncertainty.

In a local system such as a car or an infrastructure, physical lines can be used to trigger and thus synchronize the sensors together. However, this solution cannot be used in a cooperative context since some users are mobile.

In [42], the authors showed that the delay induced by the communication can significantly affect the position estimation and thus estimate delays between the users to match the timestamps of acquisition to reduce the delay's impact.

Another solution can be found by using the NTP to synchronize the users. This is the solution given by Yang et al. in [61]. As we mentioned earlier, they use the NTP protocol to synchronize their Kinect to perform their acquisitions. Nevertheless, while adapting the NTP to the automotive network seems to be a reasonable solution, it brings the question of which user provides the clock. A natural answer could be to use the infrastructure's clock but we know by experience that they are not always accurate (e.g. clock provided by the Radio Data System (RDS) data from the local radios). Another answer is to use the GPS timestamps and the triggering signal they provide with a Coordinated Universal Time (UTC) format offering a basic accuracy of $2\mu s$ [63], widely used nowadays.

Movement-based synchronization can also be an answer but highly depends on the calibration stage and requires an overlapping area in the acquired data.

d) Point of views: Point of views can be extremely different in a scene featuring infrastructure and mobile users. Thus ask the question on the fundamentally different looking of a single object which can even be considered as non-overlapping data. An example could be a sensor observing the front left corner of a car and another sensor observing the right back corner of the same car. The Mo-Cap systems can bring an answer to this section here as well by trying to match the perceived object with a skeleton or a bounding box and fitting them together.

Another question comes with the mix of mobile and static users. In [64] Merriaux et al. show that LiDAR scans are affected by the movement and demonstrate that the rectification of the point cloud brings better results at the merging step. To our knowledge, there is no study of a fixed laser scanner with moving objects but we can suppose that some alteration can be caused on the scanned moving structures.

e) Perception matching: Perception matching between objects sensed by others and shared to the ego vehicle and the object sensed by the local sensor is a typical challenge of a cooperative system and is rarely assessed in the works we have seen. A basic solution is to match the object with their positions as in [9] but the noise induced by the sensors can lead to errors. Similarly, we can use features describing a vehicle. The position can indeed be a feature and we can add to them more features. This is what Kim et al. do in [42] by using the speed of the vehicle as the key feature to match the shared data with the perceived objects.

With a more mathematical approach, Miller et al. propose in [65] a solution based on the bipartite graphs which are based on the graph theory. However, the limitation of the bipartite graphs seems that the data can be associated with only two participants. Thus, it can perfectly fit with a centralized architecture with each participant fitting their observation with the one stored by the infrastructure.

III. Localization

As we have seen earlier, some non-cooperative methods manage to reach the constraints of $0.3m$ given for in-lane autonomous navigation [11], [66] in optimal conditions. However, non-cooperative approach is limited by sensor capabilities such as the GPS coverage density significantly affecting its performance as well as weather and light conditions affecting optical sensors such as cameras and LiDARs. Indeed, the multiplication of the estimations makes it possible to eliminate the outliers as highlighted in [67]. Moreover, cooperative systems allow the extension of covered areas and fields of view, which again increases the reliability and precision of the estimations [42], [68], [69]. The other interest of cooperation in a localization context lies in the reduction of costs. The improvement of the accuracy and reliability of a sensor is generally proportional to its price. However, they can be improved by multiplying the number of sensors distributed over other users or infrastructures hence reducing the cost of each vehicle [68].

The cooperation can be implemented at several levels of estimation from the lowest level by sharing raw sensor results to a higher level by sharing estimated coordinates. In the first case, the objective is rather to extend the coverage of services either because they are inaccessible (e.g. GPS in a tunnel) or because the vehicle is not sufficiently equipped.

A. Low-level cooperative position estimation

One of the most commonly used examples of cooperative position estimation today is GNSS. GNSS uses the multilateration technique to estimate the position of a point by measuring the distance between it and several anchors as explained for TDOA or TOA algorithms (Fig. 2). Here, GPS satellites are used as anchors with their known positions since, in addition to transmitting the time of transmission allowing to estimate the distance between the satellite and the receiver, they also transmit their orbital parameters (almanacs) allowing to recalculate their position depending on the date.

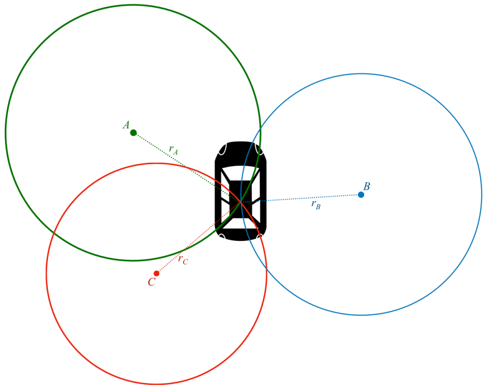

Fig. 2: Illustration of the multilateration principle. A, B and C represent users or infrastructure points with known locations. The multilateration allow to find the location of the vehicle from the distances r_A, r_B and r_C and the positions of A, B and C.

1) Multilateration: Because of the effectiveness of multilateration, this method has been adapted to other sensors from other vehicles or infrastructures. In particular, Rohani et al. in [70] made a simulation with a GPS and a measurement of the distance between vehicles, obtaining an error ranging from 3.3 m to 6.75 m depending on the quality of communication with other vehicles. The maximum error corresponds to the error of the GPS alone which shows that, in this case, the cooperation only adds a better accuracy to the GPS but does not degrade it in case of bad conditions.

To reduce the impact of poor communication, it is possible to apply weight on distance measurements. This is notably what Ahammed et al. propose in [71] by applying weight on the measurements depending on the distance between the two entities leading to an average error of 2.38 m on a fleet of 10 vehicles. Similarly, Altoaimy et al. in [72] apply weight on position estimates using the signal to noise ratio (SNR) on the communication used to estimate the distance between entities. The simulation of this scenario leads to errors of 85 cm with 20 vehicles and 25 cm with 200 vehicles.

Although these results are not accurate enough for standalone navigation, it is important to note that they were obtained using GPS only as a base. Therefore, the use of other technologies may lead to better results, such as the work by Del Peral-Rosando in [73] using a TDOA algorithm on 5G cellular network antennas estimating the position of the receiver with an error between 20 cm and 25cm.

2) Triangulation: In the same way, as for multilateration, triangulation makes possible the estimation of the position of a receiver in an environment equipped with anchors. However,

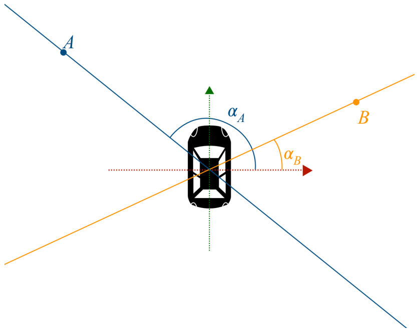

Fig. 3: Illustration of the triangulation principle. A and B are users with a known location and are detected by the ego-vehicle. The angles of detection α_A and α_B form two lines intersecting at the position of the ego-vehicle.

where multilateration uses the measurement of the distance between the receiver and the anchors, triangulation uses the angle of incidence of the signal emitted by the anchors. Triangulation is therefore the principle on which the AOA approaches are based, as illustrated in Fig. 3.

However, the authors of [74] note that the multilateration approach obtained better results at the middle of the network but that the triangulation approach became more efficient at the edges of the network. The authors, therefore, propose the implementation of hybrid TDOA and AOA systems. Nevertheless, triangulation-based approaches in the context of cooperative vehicle localization are still rare today.

3) Geometric: Compared to the two previous approaches, the geometric approach is one of the most direct methods. It consists of positioning the users in the local coordinate system of an observer (vehicle or infrastructure) having its global position known. To locate the user in the local coordinate system, several sensors can be used such as cameras, RADARs or LiDARs.

In particular, Einseider et al. have implemented an alternative positioning system for underground parking lots [47]. The detection and localization of vehicles are done via cameras placed at known positions. To estimate the position of vehicles in the fields of view of the cameras, the images are segmented into zones of 1 meter. The device set up by the authors allows detecting a vehicle at 20 m with a maximum error of 80cm. This methodology takes advantage of the geometric topology and the small distances of the scene but does not apply to larger baselines. To overcome this problem of scenes with large distances of the Providentia project, RADARs have been added to the cameras. This device allows the detection and localization of vehicles over distances up to 200 m with a longitudinal RMSE of 3.27 m and a lateral RMSE of 0.53 m [44].

With a smaller scene, Lv et al. proposed an approach based on LiDARs at the four corners of an intersection. Vehicles are identified in point clouds by clustering points having a distance between them below a fixed threshold. The position extracted from this point cloud corresponds to the nearest point of the laser scanner. In [69], Héry et al. suggest a solution to extract the position of the vehicle from these point clouds. They distinguish two types of clusters, those shaped like an L common in lateral detection and those shaped like a C for longitudinal detection. In addition to these two types of clusters, two cooperation formulations are presented. The first one corresponds to the one where the ego-vehicle is equipped with sensors estimating the pose of a vehicle with a known position. The second one corresponds to the formulation where the ego vehicle has its position estimated by the other vehicle having its position known and being equipped with the sensors. Héry et al. observe that the second formulation, corresponding to the case where the ego-vehicle position is estimated by the vehicle knowing its position and being equipped with sensors to estimate the pose between the two vehicles, obtains better results than the first formulation. This lies in the case of infrastructures where the positions of the sensors are precisely known. Besides, L-shaped clusters provide, in both formulations, results with better accuracy and consistency, underlining the importance of the structural perception of the vehicle.

B. High-level cooperative position estimation

As we have shown, low-level-oriented approaches are much closer to the hardware. In the case of high-level approaches, they use estimates of already established positions as a basis for refining them. One of the most popular methods for position estimation applications is based on the Extended Kalman Filter (EKF). This approach has been chosen by Miller et al. in [65] to enrich the position estimate obtained by a GNSS system with position estimates from other vehicles and integrating these data through the use of an EKF with a resulting standard deviation of 0.02m.

However, despite their efficiency in terms of calculation cost, EKFs are only applicable to locally linear signals with noise following a Gaussian distribution. In other words, the error of position estimation must be contained in a Gaussian distribution and thus won't allow jumps (which can appear in urban canyoning conditions). Outside these conditions, they are no longer efficient and other methods such as particle filters are preferred. This is what Huang and Wu have chosen in [75] by proposing a cooperative framework based on this approach and the Interacting Multiple Model (IMM) adapted to cooperation. The authors simulate the use of the simple Particle Filter (PF) and obtain an RMS error of 0.2146 m/m traveled on the x-axis and 0.2135 m/m on the y-axis whereas with the IMM-PF filter they obtain 0.1249 m/m and 0.1193 m/m on the x-axis and y-axis respectively.

While Miller et al. [65] use an approach based on graph theory and in particular bipartite graphs to associate perceived vehicles with the one from the real world, Gulati et al. use bipartite graphs in the form of factor graphs for localization [68], [76]. In [68], the authors present a formulation of the co-operative localization problem by setting constraints between vehicles according to their distance to correct measurement errors and obtain better results than those obtained using an EKF. The authors reiterate in [76] by integrating data from infrastructures and exceed the previous results.

The use of the high-level approach based on optimization and filtering methods brings several advantages. The first one is that this approach is compatible with low-level approaches. Indeed, low-level approaches give as output a position estimation whereas high-level approaches take as input position estimates to refine them. Consequently, the high-level approaches operate as a brick placed to improve those used for position estimation. However, this advantage of easy integration into an existing system underlines a major drawback: high-level approaches require basic components to obtain a first position estimation and therefore cannot be used alone. Another advantage of using this approach is that the processing and size of the data required are reduced significantly facilitating the communication between users. This is indeed the observation of Gulati et al. in [68], [76] via the use of factor graphs.

We could distinguish 3 methods mainly used: EKF, Particle Filter, and Graph-based methods. Thus we introduced an example of each to understand the available methods with their advantages as well as their limitations. However, Gao et al. gather a lot more of these methods in a cooperative context in their book [67] diving into mathematical details as well as the diversity of variations of each method which is beyond the scope of this paper.

C. Conclusion

In Table IV, we note that several methods solely offer a better precision compared to the non-cooperative methods. Generally, cooperative localization optimizes the output of the standalone position estimation methods, refining the estimation through extra data usage. However, a poor quality localization ability of an agent might dramatically affect the overall results. It also offers an alternative source of localization for GNSS denied environments, especially from well-located measure points such as infrastructures.

IV. Object detection and tracking

To navigate in an environment, it is necessary to be able to detect obstacles in the scene and to track them. Today, most approaches are based on non-cooperative detection algorithms. This is mainly due to the limitations of communication methods. In this section, we will review the different approaches to perform detection in a cooperative context and the available tracking methods.

A. Detection and classification

The first step before classifying objects is the extraction of areas of interest from the data produced by the sensors. Here, we want to isolate the mobile objects from the background. This is what Lv et al. do in [25] where after subtracting the background, group the remaining points into clusters. These clusters are delimited by batches of points having a distance to each other below a threshold set beforehand. Another strategy was adopted by Chen et al. in [26] where the shared data correspond to areas of interest depending on the position of the vehicles such as the part of the scene scanned by two vehicles. The more precise extraction of objects is performed during the detection phase.

The trend of point cloud raw data sharing is very recent. This is because communication between users has been limited for a long time. For instance, the majority of cooperative systems perform the detection and classification of objects in a scene locally. The extracted data is often enriched before being shared. A typical example is the Providentia project [44] where cameras provide a video stream sent into a neural network based on the You Only Look Once Version 3 (YOLOv3) architecture to detect vehicles in the images. The data of the vehicles thus classified are enriched thanks to RADAR sensors allowing a better estimation of their position in the scene. Similarly, Lv et al. [25] based their solution on the same concept where the user's characteristics are locally extracted and where the classification, using the random forest algorithm, is done locally. The corresponding data are then centralized to facilitate user tracking.

Nowadays, the majority of detection and classification methods are based on algorithms based on a neural network-oriented architecture. Grigorescu et al. in [77] and Arnold et al. in [17] provides an overview of methods used to detect and classify other users in a non-cooperative manner. Although the details of these methods are beyond the scope of this paper, Arnold et al. offer a review of data fusion methods, thus providing insight into the problem of multi-modality and the management of several streams. Based on the work of Chen et al. [78], the authors raise 3 fusion schemes :

Early Fusion:
: The data streams are merged and formatted before passing through the neural network. As an example, color data can be added to point clouds from cameras. The disadvantage of this solution is that it is not robust to stream failure.

Late fusion:
: This is the classic scheme we have seen: the data are processed locally and separately for each modality and then the results are merged only at the end. Although it does not benefit as much from the cooperation in terms of classification, it offers the best performance.

Mid fusion:
: Also named as deep fusion, the raw data are sent to the neural network, which will handle the association of the data by itself. Although it is sensitive to the absence of modality, it takes full advantage of cooperation and offers better results than the previous methods. It is with this scheme that the work of Chen et al. [26] can be associated.

These three approaches were initially formulated for the local processing on the vehicle but can easily be extrapolated into a cooperative context. Therefore, we can associate these three strategies to the notion of a stream that can contain point clouds, images, or the characteristics of the detected users from any source. However, this extrapolation has a cost in terms of complexity because the sensors have to be calibrated dynamically from each other. Since the systems are independent of each other, the extrinsic parameters between the sensors are composed of translation, rotation, and time-

Paper	Category	Methodology	Metrics	Results	Simulation / Experiment	Cooperative style	Notes
[70]	Multilateration (Low-Level)	GPS Multilateration	Error	3.3 m to 6.75 m	Simulation	V2V	
[71]	Multilateration (Low-Level)	GPS weighted Multilateration (based on distance)	Average error	2.38 m	Simulation	V2V	With a fleet of 10 vehicles.
[72]	Multilateration (Low-Level)	GPS weighted Multilateration (based on SNR)	Error	85 cm to 25 cm	Simulation	V2V	With a fleet of 20 vehicles and another of 200.
[73]	Multilateration (Low-Level)	TDOA with 5G antennas	Error	20 cm to 25 cm	Simulation	V2I	
[47]	Geometric (Low-Level)	Image segmentation	Error	80 cm	Experimental	V2I	At 20 m
[69]	Geometric (Low-Level)	Sensor fusion known → unknown	Mean error	x: 11 cm, y: 36 cm, h: 39 cm	Experimental	V2V	
[69]	Geometric (Low-Level)	Sensor fusion unknown → known	Mean error	x: 27 cm, y: 116 cm, h: 124 cm	Experimental	V2V	
[75]	Optimisation (High-Level)	Particle Filter	RMSE	x: 0.2146, y: 0.2135 m/m traveled	Simulation	V2V	
[75]	Optimisation (High-Level)	IMM-PF	RMSE	x: 0.1249, y: 0.1193 m/m traveled	Simulation	V2V	
[65]	Optimisation (High-Level)	EKF based optimisation	Standard deviation	0.02 m	Both	V2V	
[68]	Optimisation (High-Level)	Factor Graph	combined RMSE	See original publication for graph	Simulation	V2I	Improvement compared to EKF
[76]	Optimisation (High-Level)	Factor Graph	decrease RMSE	10.54 %	Simulation	V2I	Compared to EKF with 4 vehicles for 1000 iterations

TABLE IV: Recapitulative table of the reviewed localization works.

shift parameters.

The authors of [26] however proposed an extrapolation of the deep fusion scheme in [79] in which the raw data from a laser scanner start being processed in a neural network. The authors tried using the feature at a different level: the voxel feature level and the spatial feature level. The first one shares a 3D grid containing the result of the VoxelNET neural network while the other one shares a higher-level feature from the fusion of spatial features maps. While the first one generates a large amount of data, the spatial feature level is sparser, thus lighter, facilitating the exchanges in a bandwidth-limited environment. Similarly, Marvasti et al. in [80] propose a method to share deep features from an intermediary layer of a neural network. However, such an approach brings the question of standardization of the perception pipeline among every user especially on the evolution of the neural network in charge of detection as well as the diversity of models from the different suppliers.

B. Tracking

The aim of tracking users is to follow them as long as possible in the scene. Several methods are available to tackle tracking tasks, enumerated in [81] by Datondji et al., such as region-based, contour-based, feature-based, or model-based methods. Datondji et al. also list two types of tracking algorithms: matching-based and Bayesian-based algorithms. However, this can be a challenging task because of several parameters such as occlusions, change of perception (e.g. appearance, distortion, etc.), or environment changes (e.g. lighting, color change, weather change, etc.). Cooperation brings an answer to these difficulties by bringing various points of view.

In [82], authors underline that localization tasks and tracking tasks can be bounded in Simultaneous Localization And Tracking (SLAT). Authors propose to use a localization method based on the footprints of radio transceivers based on the Omnipresent Signals of Opportunity (SOOP) method. Connected vehicles seek targets by exterminating the radio reflection of illuminated targets. They propose a SLAT method based on the derivation of Fisher Information Matrix (FIM) to locate users and use a hybrid distributed algorithm based on Belief Propagation (BP) to track them and obtain better results than EKF based methods. The tracking method used is based on the region matching method. Similarly, Miller et al. used in [65] a region matching method based on bipartite graphs to track vehicles in a V2V context.

In Providentia project [44], [49], authors based their tracking methods using Gaussian Mixture Probability Hypothesis Density (GMPHD). Similarly, Chen et al. in [83] used a GMPHD based method to extract the tracks of multiple vehicles. The authors perform a SLAT using a Bayes inference-based algorithm optimizing relative pose estimation and fusing the matched tracks using fast covariance intersection based on information theory (IT-FCI). These methods are based on region methods alongside Bayesian-based algorithms. An answer to the resolution of complex scenes is provided by Huang and Wu in [75]. The authors rely on cooperation and on a method using particle filters to locate vehicles more precisely, thus reducing ambiguities when the vehicles are very close to each other.

Kim et al. in [42], [43] uses the speed of the vehicles as a feature to identify the user and to track them, thus performing a feature-based tracking with a matching algorithm. Lv et al. in [25] used the corner of the detected car the closest to the sensor and applied the Global Nearest Neighbor (GNN) [84] method to track the vehicles. This approach lies in the use of a contour-based method with a matching algorithm.

In [85], authors propose a set of metrics available for tracking tasks performance evaluation which is nowadays frequently used. However, in a cooperative context, we have not found works that bring a quantitative evaluation of their tracking methods. This is mainly caused by the fact that, in cooperative works, a tracking task is just a tool but not at the center of the research efforts.

C. Conclusion

The multiplication of the points of view offers a significant advantage to overcome the limitations of the sensors or to reduce the effects of the changes of the scene condition. In Table V, we provide a summary of the solutions given for user detection. Tracking on the other hand seems to be put aside since the cooperative tracking methods used are often only a means to obtain other results on other parts of the perception pipeline. We also observe that the field of detection and tracking in a cooperative domain benefits from very little research effort. We believe that this lack of experimentation in a cooperative context is due to the bandwidth requirements in communication as well as the sensitivity to desynchronization and pose estimation errors.

V. Map generation

In the previous sections, we have reviewed the ego-localization methods as well as the detection and tracking methods of agents in a given scene. Ego-localization and detected and tracked objects can be merged in a map. Thus, the map can be built cooperatively by aggregating information from multiple agents. Nowadays, commercial solutions are available to bring a cooperative aspect to the maps available in navigation aids. This is notably the case for crowdsourcing-based solutions such as TomTom, HERE, Waze, etc.

Hence, it is clear that the goal of cooperative mapping used today is to optimize routes and adapt vehicle navigation by anticipating the different events on the user's route. These objectives can be taken further, in particular, to predict trajectories in real-time thanks to lower latency and better accuracy of shared data.

In this section, we review the use of maps in a cooperative context and the different formats available.

A. Geometric maps

Geometric maps are made up of vector elements describing the environment. This method is used in applications such as OpenStreetMaps. However, in a cooperative context, data from services like the one mentioned above are not precise enough, which has led to the creation of maps with better accuracy. In [86], the authors present Enhanced Maps (Emap) that provide lane level accuracy maps. To achieve this goal, Bétaille et al. propose to add a set of circles and clothoids to the traditional vertices. Also in view to improve map accuracy, Bender et al. present in [87] the lanelets. The lanelets take the form of vertices representing the left and right sides of a traffic lane. These vertices also have an enhanced topological role by representing the links between places and the distance between them.

The use of geometrical maps in a cooperative application has a supporting role in which the information shared between users is integrated. Xu et al., in their review of their participation in the 2016 GCDC [9], had to recreate a high-definition map to enrich the OpenStreetMap plots before using it. Thanks to these high-definition maps, it has become possible to precisely place elements in real-time such as other users or danger zones to be avoided and thus to navigate in a context of cooperative driving in several scenarios that we will present later. Similarly, in the Providentia project, the autobahn section has been modeled beforehand with great precision, creating a digital twin of the scene [44]. Here, the infrastructure shares the position of each of the detected vehicles to generate a dynamic map. Finally, the team of Lv et al. [25] didn't use maps but has rather relied on sharing information in real-time that can be used to enrich geometric maps such as the position of vehicles, pedestrians or even information on the status of traffic lights.

Through these applications, a global pattern emerges: the shared information is used to enrich the map rather than to modify it in depth. Cooperative geometric maps are therefore made up of a succession of layers. The base layer represents the terrain and is almost invariable. It can be created from national institutes or directly extracted from sensors. Then, the higher the layer level is, the shorter the lifespan of the elements of this layer is. This layer organization has been formalized under the name Local Dynamic Maps (LDM) by European Telecommunications Standards Institute (ETSI) [88] and takes the format of layers with varying validity periods and offers an implementation framework. The LDM is thus defined as 4 layers :

- Type 1: Static data (Roads, applied speeds, infrastructure-setc.)
- Type 2: Long term transient data (Work zone, temporary speed change)
- Type 3: Medium-term transient data (weather situation, parked vehicles, traffic jams, etc.)
- Type 4: short term transient data (vehicles on the road, traffic lights, etc.)

Each layer is updated with a frequency depending on the duration of validity of the information. Typically, the Type 4 Layer is updated in real-time.

Paper	Category	Methodology	Metrics	Results	Simulation / Experiment	Cooperative style	Notes
[25]	Raw data based detection	Random Forest	Detection Rate	95.5 %	Experimental	V2I	No data given for the tracking performances
[26], [79]	Raw data fusion based detection	CNN based network	Average Precision	Near detection: 77.46 %, far detection 71.42 %	Experimental (Datasets)	V2V	With KITTI
[79]	Voxel feature fusion detection	CNN based network	Average Precision	Near detection: 77.46 %, far detection 58.27 %	Experimental (Datasets)	V2V	With KITTI
[79]	Spatial feature fusion detection	CNN based network	Average Precision	Near detection: 50 %, far detection 57.14 %	Experimental (Datasets)	V2V	With KITTI
[80]	Deep feature fusion detection	CNN based network	Detection of undetected vehicle	Up to 30 %	Simulation (CARLA)	V2V	Detection of undetected vehicle by non cooperative algorithm
[47]	Geometric Tracking	Image segmentation	Error	80 cm	Experimental	V2I	At 20 m
[44]	Geometric Tracking	Sensor fusion	RMSE	lat: 3.27 m lon: 0.53 m	Experimental	V2I	At up to 200 m

TABLE V: Recapitulative table of the reviewed cooperative detection and tracking works.

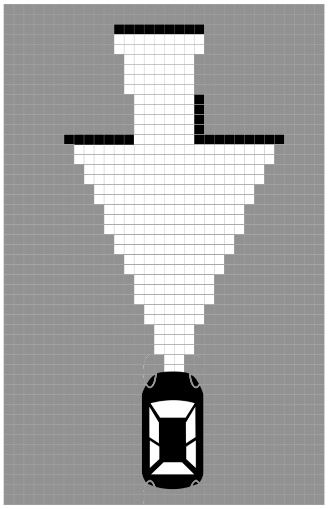

Fig. 4: Occupancy grid example. Grey boxes represent a 50 % probability of occupancy if the area is unknown. The white boxes correspond to the zones identified as free and the black boxes correspond to the zones occupied by an obstacle.

B. Volumetric maps

Volumetric maps are, unlike geometric maps, atomic elements representing the presence or absence of an obstacle that form a grid with squares contiguous to each other or scattered arbitrarily. The advantage of volumetric maps lies in the fact that they can be easily created from sensor data and therefore represent the immediate environment at the time of data acquisition. Occupancy grids fall into volumetric maps category forming a 2-dimensional grid, or matrix, similar to an image [89]. Indeed, a greyscale image can be taken where each pixel corresponds to an area of the environment and where the greyscale represents the probability that the area corresponding to the pixel contains an obstacle as illustrated in Fig. 4. These maps have the advantage that they can be combined very easily. The authors of [90] have thus shown that they were able to associate the maps of several robots to obtain a complete map of the environment. The goal of associating them is to find the transformation matrix between perception systems. In the case of 2D occupation grids, the transformation matrix $T_{x,y,\theta}$ contains three parameters: translation on the x-axis, translation on the y-axis and rotation by an angle θ. Hence the authors seek a matrix $T_{x,y,\theta}$ that maximizes the similarity between two overlapping maps also called a point registration algorithm.

Kim et al. propose in [42], [43] to enrich their map by taking pictures with cameras positioned on several vehicles. The images captured in this way are distorted to be laid on the ground, providing a satellite view of the scene. To obtain this result, they applied the Inverse Perspective Mapping (IPM) method. When the camera acquires an image, the scene is projected onto the sensor plane. TheIPM is based on the inverse principle: the 2D points of the sensor plane (stored in an image) are projected back into a 3D space, assuming that each of the points is on a flat surface (e.g. the road). The authors of [42], [43] set the plane to $Z = 0$ and used other sensors to remove the points that do not belong to this plane. Thus, by knowing the position of the different vehicles and, by extension, the position of the cameras, it is possible to obtain a map enriched with a satellite view cooperatively.

Although this type of map has the advantage of being simple to use and share, it has the disadvantage of becoming heavier with the size of the environment being explored invariably, whether the areas are interesting or not. To overcome this problem, the notion of quadtree can be introduced. The quadtree divides the map into coarse blocks which, if they contain useful details, can be subdivided into sub-blocks which, in the same way, can be divided into sub-sub-blocks.

The authors of [91] used the quadtree-based method to store a grid of occupancy generated by LiDAR type sensors. Although they note that the method is more computationally intensive, it shows its advantage by dividing up to 10.9 the storage required for an equal area and accuracy. However, to the best of our knowledge, there are no methods for merging maps in quadtree format.

Until now, we have mainly been talking about two-dimensional maps, both geometric and volumetric maps. However, three-dimensional maps are becoming more and more popular thanks to sensors that provide information in three dimensions rather than just on one plane. 3D maps play an active role in navigation, especially in complex environments [92], and provide additional elements that make it easier to combine several maps.

Similarly, the 2D occupancy grids are also available in a 3D version consisting of voxels (volumetric elements). However, just as 2D maps tend to be too large, 3D maps are even more affected by this problem due to the additional axis. The answer to this problem is similar to that of two-axis maps: the octree. Hornung et al. present in [93] the OctoMap framework allowing the management of maps and their updates based on a probabilistic approach. Unlike the quadtree maps, the octree maps have benefited from a better interest in the context of cooperation. We can notably mention Drwiega's work in [94] proposing a method for associating several maps in the Octree format. To estimate the transformation matrix between the respective coordinate reference of the two maps, the author translates the Octree map into a point cloud and then applies the Iterative Closest Point (ICP) algorithm to it.

This brings us to maps based on point clouds. The volumetric maps we have seen so far represent the first steps in navigation in the context of mobile robotics that can be cooperative. However, in the context of the autonomous vehicle, point cloud-based maps are more widespread. Point cloud-based maps have the advantage of representing each impact (and therefore obstacle) in Cartesian coordinates as well as the raw data from laser scanners like sensors [95]. These maps contain both fixed elements (the background, Type 1 in the LDM reference frame) and highly dynamic elements (Type 4 in the LDM reference frame). As a result, the static part of the map is occluded by the dynamic elements of the scene. One solution to reduce the impact of occlusion is cooperation, where the map can be generated by several sensors offering several points of view. This is notably what Bosch's teams chose in the MEC-view project [59], [60] where a set of cameras and LiDARs were placed on lampposts to generate an High Definition (HD) map and offer a view free of blind spots to autonomous vehicles updating in real-time. In [25], Lv et al. proposed a solution to extract the background from the raw scans by aggregating several frames and then applying thresholding to the voxels resulting from the rasterization of the accumulated point clouds.

In the same way, as for occupation grids (2D or 3D), the key point allowing the cooperation and thus the association of point cloud maps is the estimation of the transformation matrix between the respective referential of each point cloud. As explained by Yang et al. in [96], [97], the point set registration algorithms are particularly suitable for this task. Indeed, their objective is to find the transformation matrix minimizing the distances between a set of points located on overlapping acquisition parts. Note that sensors, and thus point clouds, by convention, are measured in metric systems which implies that scaling is generally not necessary (if it is required, it would be specified by the manufacturer). Thus, the desired transformation is then qualified as rigid in which the transformation matrix is composed only of the translation matrix and the rotation matrix. The most popular algorithm in mobile robotics is the ICP algorithm that looks for the minimum distance between corresponding points in the two-point clouds by using the method of least squares. However, this method is particularly sensitive to outliers. Another challenge appears with the lengthening of the baseline which is the increase of the disparity of the points. To overcome this problem, Wu et al. propose in [98] a semi-automatic solution to merge sparse point clouds called PA-ICP. PA-ICP is based on the recognition of corners which must be paired with their corresponding corners in every point cloud. Finally, in the context of the autonomous vehicle, it is vital to know the confidence index of the generated map and thus the quality of the point cloud association. Yang et al. propose in [96] TEASER, a point set registration algorithm capable of indicating its confidence index and being robust to outliers.

As we have written, maps based on point clouds contain both static and dynamic elements. The dynamic elements can therefore be extracted from the latter to be processed to recognize their role in the scene and track them if necessary.

C. Conclusion

To conclude this section about mapping, we can observe that cooperative mapping serves the enrichment of the context in which vehicles moves. Thanks to the larger memory available on the infrastructure it is possible to store and thus share heavy HD maps. As we will see in the next section, the multiplication of the point of view reduces the occlusions and improve the reliability of the detection and tracking. These detected objects can be placed on the map following the LDM model and then shared with the connected vehicles to help the trajectory planning stage.

VI. Review and summary

In the previous sections, we have reviewed the three main blocks of the perception pipeline in a cooperative context: localization, mapping, and object detection and tracking. In addition to this, we reviewed the architectures available for cooperative systems along with their advantages and drawbacks. We also observed the challenges brought by cooperative solutions as well as the available network facilities. This information allows us to establish a SWOT and thus obtain a clear view of the state of the art of cooperative perception and more particularly of those using an infrastructure. This SWOT is available in Table VI.

Through these sections, we have also reviewed several solutions that make use of cooperation for certain blocks of perception. For the sake of clarity, Table VII provides a review of them.

VII. Cooperative perception in real life

So far, we have reviewed the available data to perform perception, the methods to share them as well as the different approaches and challenges related to cooperation. We also

Strenghts: • More precise localisation in environment with GNSS • Localisation possible in GNSS denied environment • No drift in Localisation • V2X Mapping • Better reliability • Cost reduction • Less occlusion • Real-Time update (solely depending on the transfer latency and computation time) • Larger field of view • Detection of unconnected User	Weaknesses: • Similar precision of pose estimation with non cooperative system • Dependent to the number of users • Computation expensive • High throughput required • Latency
Opportunities: • Raw sensor data fusion • Various point of view of the scene • V2I Map generation • V2I Object management • Infrastructure always available and calibrated • Further Trajectory planning • Anticipation of dangers • Infrastructure offers more storage and can delete duplicate parts allowing storing HD maps • Better Bird Eyes View map creation • Existing matching methods	Threats: • Higher cost for the infrastructure • Lack of normalisation between constructors • Consistency of the accuracy of the pose estimation between the sensors • Detection and classification accuracy of each participant • Synchronisation between participants • Data association of a single object with a very different point of view. • Missing stream or data management • Data management between mobile and fixed users

TABLE VI: Cooperative Perception - SWOT

reviewed three main tasks of perception using cooperation namely, ego-localization, detection and tracking, and, finally, map generation. This section aims to assess the scenarios in which cooperative perception proposes a significant impact as well as the related experimentations. We will close this section with a presentation of datasets.

A. scenarios & Experiments

The cooperative perception responds to safety issues and more specifically those related to the lack of visibility in blind spots. This lack of visibility can be caused by the structure of the scene or by other users. We can take the example of pedestrians wanting to cross the road but being hidden by parked vehicles or even vehicles appearing in an intersection and being hidden by buildings. It is on this last example that the point cloud sharing project of Li et al. is based [40]. The authors' work focuses on the SDN network structure for connected vehicles as well as the use of mmWAVE wireless communication links offering higher data rates than networks in the 2.4 GHz frequency bands. In this network, there are several infrastructures equipped with laser scanners that allow the visualization of areas hidden by buildings thanks to the fusion of LiDAR point clouds covering the trajectory of the connected vehicle. In this way, they can reduce the effects of blind spots and detect other users that were previously undetectable.

Li et al. also addressed the overtaking scenario in which it can sometimes be difficult to know whether a vehicle is coming into the opposite lane since the view is occluded by the vehicle we wish to overtake. This is also one of the scenarios that motivated the work of Kim et al. in [43]. In this paper, the authors use cameras placed on several vehicles to create a see-through visualization system. To merge the images, the authors project these pixels onto the ground to create a birds-eye view map. This map can then be back-projected according to a camera model to visualize what is behind the vehicle.

The 2016 edition of the GCDC was an opportunity to explore several other scenarios as well as challenging several teams. In this case, Xu et al. [9] presented these scenarios and their comments about their experience. Three scenarios are presented:

Zipper merge:

This case corresponds more generally to the insertion of a vehicle into a lane and is encountered in several situations such as when a traffic lane becomes inaccessible (e.g. for maintenance).

Crossing at intersection:
: Here, a vehicle wants to cross an intersection with as little disturbance to the traffic situation as possible.

Emergency vehicle yielding:
: This situation corresponds to the arrival of an emergency vehicle and which therefore has the priority. The vehicles on the scene must leave a passageway between the traffic lanes to allow them to circulate.

During the experimentation phase, the vehicles transmitted their status (position, speed, wheel angle, etc.) and could make requests involving a change in vehicle behavior. For example, when inserting into a lane, the vehicle behind changes its speed to leave sufficient space for the requesting vehicle to change lanes. During this challenge, the vehicles do not cooperate on the perception axis but rather on the vehicle control axis. However, the authors note in their remarks the weaknesses of the perception implemented caused by the lack of a multi-sensor based perception methods.

The project Proviendentia [44] aims to bring cooperative perception to the motorways. This project is composed of cameras and radars placed on gantry bridges on a section of the motorway. Vehicles are detected and classified using machine learning algorithms, and their positions are estimated using the data provided by the radars. A digital twin of the road section is created from this data and is accessible in real-time.

MEC-View is a similar project implemented by Bosch [59], [60] where LiDARs and cameras are placed on lampposts at an intersection to cover blind spots caused by other vehicles. Similarly, Lv et al. in [25] equipped an intersection with 4 LiDARs sensors to track vehicles and detect obstacles to inform users. The problem of intersections is particularly extrapolated to roundabouts, which are more frequent on the European continent.

Another issue, raised by Kim et al. [43] as a limitation to their system resides in the roads forming parabola peaks. Under these conditions, the topology of the terrain reduces the driver's field of view to the sensors.

B. Datasets

The increasing interest in the cooperative vehicle initiated the sharing of some datasets. However, they tackle specific contexts such as communication or infrastructure perception.

a) Ko-PER [99]: This dataset proposes a context of a cooperative infrastructure. It is made of sequences monitoring an intersection with 14 laser scanners (4 for the road, 2 for the sidewalk, and 8 for the egresses) and 8 monochromatic cameras (only two are available in the dataset due to personal data protection purposes). Laser scanners are synchronized and operate at $12.5Hz$ while the cameras operate at $25Hz$ in phase with the laser scanners. Raw data from the scanners and undistorted images from the cameras are available alongside reference data of selected vehicles and object labels.

b) Warringal [100]: The authors propose a dataset gathering communication interaction between vehicles of a fleet of 13 elements for 3 years. The data proposed are the state of the vehicle, the list of each communication and their length, the signal strength of each communication (e.g. RSSI or antenna used by each vehicle), and the map.

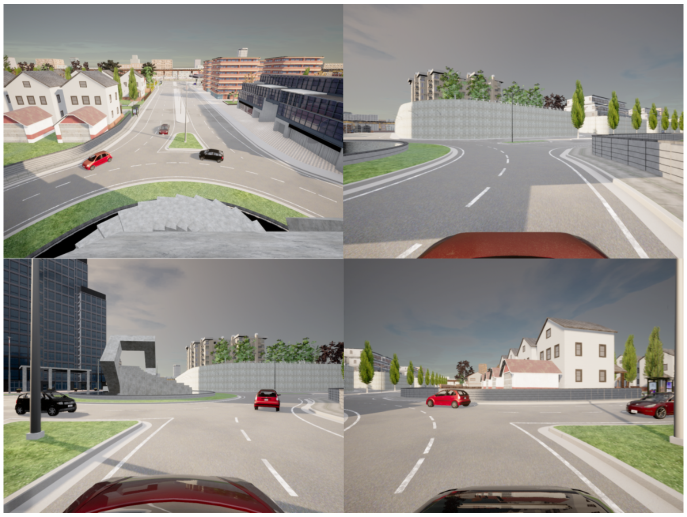

Fig. 5: Synchronous video frames from each camera of our multi-agent dataset made with CARLA.

c) T&J [26]: This dataset has been created to complement KITTI's dataset [101] by adding a cooperative dimension. For the learning and evaluation phase of their Sparse Point-cloud Object Detection (SPOD) algorithm, the authors needed a dataset offering overlapping acquisitions from several points of view. The latter is composed of images from multiple cameras, radar data as well as point clouds from LiDARs. As with the KITTI dataset, this data is linked to a GPS and an IMU but offers simultaneous views from different positions.

A lack of cooperation: We could have presented other datasets such as KITTI's or more recently the Waymo Open Dataset or INTERACTION by Zhan et al. [102]. However, we can note the absence of cooperation and dataset representing the scenarios we presented despite the interest and the projects responding to its problems. We can also note that the datasets presented deal either only with infrastructure or V2V cooperation. However, simulators can bring an answer to this lack by allowing the acquisition of data from several points of view synchronously. Moreover, they solve the problem of the ground truth definition as well as the calibration challenges. CAR Learning to Act (CARLA) [103] is one of them providing several sensors such as cameras, depth cameras, LiDAR (simulated ray cast), IMU and RADAR. In July 2018, version 0.9.0 introduced the multi-client multi-agent support offering cooperative vehicles perspective. Fig. 5 showcases the possibilities offered by CARLA with synchronized image acquisition from vehicles and infrastructure at a roundabout. Other simulators are available such as Deepdrive [104], LGSVL Simulator [105] or AirSim [106]. However, CARLA remains the most popular nowadays.

VIII. Conclusion and perspectives

This paper was an opportunity for us to review the different stages of a perception pipeline under a cooperative context and its associated challenges.

A large amount of work tackling the localization problem has been accomplished. We reviewed a wide range of solutions introducing different paradigms, improving the pose estimation, or offering an alternative reference point in a GNSS denied environment. We also noted that the cooperative localization topic is very active as witnesses the amount of recent literature. However, we perceived a strong contrast concerning the available literature amount on cooperative detection and tracking. Even if some projects employ local perception systems merging the detected user's information, the topic remains sparse in raw data sharing. With the abilities offered by the new communication infrastructures, we believe that a whole new panel of innovation becomes reachable featuring algorithms fed with both feature level data and raw data.

We also reviewed the usage of maps in a cooperative context which, similarly to localization, is currently an active topic. In this field, cooperation also makes it possible to overcome the limits of the distance of sensors, allowing better anticipation of trajectories and possible adjustments.

More generally, we witnessed a difference in the cooperative scheme between V2V and V2I architecture. In V2V, vehicles communicate evenly with each other whereas, in V2I, the privileged approach is unidirectional from the infrastructure to the connected agents. We believe that bidirectional cooperation could be beneficial in bringing an ”in the scene” point of view, thus adding details helping to understand the scene. This bidirectional scheme may provide new opportunities for dynamic calibration, reinforcement learning [107] or as an arbitrator in case some agents share erroneous data.

Finally, although cooperative perception is currently an active topic, we noted the absence of datasets featuring multiple points of view, from different actors, in a scene. These datasets are a real key point in cooperative perception since they are mandatory to bring novel cooperative solutions. However, their creation requires solving the abovementioned challenges such as calibration.

Ref	Type	Tasks	Sensors	Communic.	Architect.	Method	Comment
[25]	V2I	Localization, Mapping, Classification & Tracking	LiDAR	DSRC, SPaT, BSM	Centralized	Geometric Relative Localization, Background filtered, PC cluster, Random Forest	The lanes are detected by aggregating vehicles paths and vehicles are tracked with the closest point of their corresponding cluster. The infrastructure does not merge the Point Clouds and transmit the information to the users via Bluetooth. The pose of the vehicle is determined in relative coordinates and converted to absolute coordinates.
[26]	V2V	Detection & classification	LiDAR	DSRC (ROI)	Distributed	CNN	SPOD is based on CNN. A dataset has been created.
[9]	V2X	Localization	LiDAR, GNSS-RTK	ETSI C-ITS (CAM, DENM, iCLCM)	Distributed	Control	Absolute coordinates are transmitted by messages by each vehicle.
[40]	V2I	Mapping	LiDAR	mmWave, ROS	Mixed	LDM	Share Point clouds through mmWave links.
[10], [44]	V2I	Detection, classification & tracking, Localisation	Camera, radar	4G, 5G, Optic Fibre	Centralised	YOLOv3, Tracking via radar, GM-PHD	The radar help to determine the position of the users in the absolutes coordinates.
[42]	V2V	Mapping, vehicle matching	Odometry, LiDAR, camera, DGPS	IEEE802,11n (WiFi)	Distributed	IPM	RAW data are shared between vehicles for mapping. Feature-based object matching (speed of the vehicles). Maps are merged using the coordinates given in the messages.
[43]	V2V	Tracking, Mapping	Odometry, LiDAR, camera, DGPS	IEEE 802,11gn (WiFi), 3G, 4G, ROS	Distributed	Mapping: IPM, ICP, CSM	The position of tracked users are given into relative to the ego-vehicle coordinates.
[45]	V2X	Tracking	Camera, LiDAR	IEEE 802.11bgn (WiFi), ROS	Distributed	GM-PHD Filter, EKF, Sequential Monte Carlo	The relative poses are estimated
[68], [76]	V2I	Localisation	Range detector, Odometry, GPS (all simulated)	Simulated	Distributed	Factor graph (High Level)	The absolute positions are directly processed.
[47]	V2I	Tracking	Camera	IEEE 802.11g (WiFi), IEEE 802.11p (WAVE) with DENM messages	Centralised	Geometric (Low level)	The map and position of the user are transmitted from the infrastructure. The position is given in absolute coordinates of the car park space.
[69]	V2V	Localisation	LiDAR, GNSS RTK	Not given	Distributed	Geometric (low level)	The relative pose is extracted from the LiDAR's data and is used to compute the absolute pose.
[70]	V2V	Localisation	GPS, Range sensor	Not given	Distributed	Bayesian (High level)	The estimated position is given in absolute coordinates.
[65]	V2V	Localisation, Tracking	GPS RTK, camera, radar, LiDAR	DSRC	Distributed	EKF, Bipartite graphs (High level)	The localization is given in absolute coordinates. The bipartite graphs are used to match users to the detected ones.
[59], [60]	V2I	Localization, detection and tracking	Camera, LiDAR	4G, 5G	Centralised	Not given	
[108]	V2V	Localization and tracking	Radar	DSRC	Distributed	GMPHD	The estimated position is given in absolute coordinates.

TABLE VII: Summary of the experimentation and methods reviewed along the paper underlining their conditions of realization, the methods used and their results.

References

[1] K. Bimbraw, "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," in 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 1. IEEE, 2015, pp. 191–198.

[2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann et al., "Stanley: The robot that won the darpa grand challenge," Journal of field Robotics, vol. 23, no. 9, pp. 661–692, 2006.

[3] C. Urmson, J. A. Bagnell, C. Baker, M. Hebert, A. Kelly, R. Rajkumar, P. E. Rybski, S. Scherer, R. Simmons, S. Singh et al., "Tartan racing: A multi-modal approach to the darpa urban challenge," 2007.

[4] A. Broggi, P. Cerri, M. Felisa, M. C. Laghi, L. Mazzei, and P. P. Porta, "The vislab intercontinental autonomous challenge: an extensive test for a platoon of intelligent vehicles," International Journal of Vehicle Autonomous Systems, vol. 10, no. 3, pp. 147–164, 2012. [Online]. Available: https://www.inderscienceonline.com/doi/pdf/10.1504/IJVAS.2012.051250

[5] M. Dikmen and C. M. Burns, "Autonomous driving in the real world: Experiences with tesla autopilot and summon," in Proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications, 2016, pp. 225–228.

[6] V. Shreyas, S. N. Bharadwaj, S. Srinidhi, K. Ankith, and A. Rajendra, "Self-driving cars: An overview of various autonomous driving systems," in Advances in Data and Information Sciences. Springer, 2020, pp. 361–371.

[7] R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. S. Tabar, N. V. Irukulapati, C. Englund, P. Falcone et al., "Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge," IEEE transactions on intelligent transportation systems, vol. 13, no. 3, pp. 994–1007, 2012.

[8] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson, and J. Didoff, "The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles," IEEE Wireless Communications, vol. 23, no. 4, pp. 146–152, 2016.

[9] P. Xu, G. Dherbomez, E. Héry, A. Abidli, and P. Bonnifait, "System architecture of a driverless electric car in the grand cooperative driving challenge," IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 1, pp. 47–59, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01703415/document

[10] F. GMBH, "Providentia," ONLINE : http://testfeld-a9.de/, accessed 15.05.2020. [Online]. Available: http://testfeld-a9.de/

[11] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 829–846, 2018.

[12] H.-S. Tan and J. Huang, "Dgps-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 4, pp. 415–428, 2006.

[13] G. Liao, J. Zhao, C. Cui, H. Long, J. Zhu, and A. Djerida, "Dynamic attitude estimation improvement for low-cost mems imu by integrating low-cost gps," arXiv preprint arXiv:2008.10469, 2020.

[14] F. Zhang, H. Stähle, G. Chen, C. C. C. Simon, C. Buckl, and A. Knoll, "A sensor fusion approach for localization with cumulative error elimination," in IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). IEEE, 2012, pp. 1–6.

[15] I. Skog and P. Handel, "In-car positioning and navigation technologies—a survey," IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 1, pp. 4–21, 2009.

[16] A. Ochalek, W. Niewiem, E. Puniach, and P. Ćwikakala, "Accuracy evaluation of real-time gnss precision positioning with rtx trimble technology," Civil and environmental engineering reports, 2018.

[17] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3782–3795, 2019.

[18] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," IEEE robotics & automation magazine, vol. 18, no. 4, pp. 80–92, 2011.

[19] C. Li, B. Dai, and T. Wu, "Vision-based precision vehicle localization in urban environments," in Chinese Automation Congress. IEEE, 2013, pp. 599–604.

[20] R. Li, S. Wang, and D. Gu, "Deepslam: A robust monocular slam system with unsupervised deep learning," IEEE Transactions on Industrial Electronics, 2020.

[21] R. Garg, N. Wadhwa, S. Ansari, and J. T. Barron, "Learning single camera depth estimation using dual-pixels," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 7628–7637.

[22] D. Vivet, F. Gérossier, P. Checchin, L. Trassoudaine, and R. Chapuis, "Mobile ground-based radar sensor for localization and mapping: An evaluation of two approaches," International Journal of Advanced Robotic Systems, vol. 10, no. 8, p. 307, 2013.

[23] E. Ward and J. Folkesson, "Vehicle localization with low cost radar sensors," in IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016, pp. 864–870.

[24] M. Cornick, J. Koechling, B. Stanley, and B. Zhang, "Localizing ground penetrating radar: A step toward robust autonomous ground vehicle localization," Journal of field robotics, vol. 33, no. 1, pp. 82–102, 2016.

[25] B. Lv, H. Xu, J. Wu, Y. Tian, Y. Zhang, Y. Zheng, C. Yuan, and S. Tian, "Lidar-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities," IEEE Access, vol. 7, pp. 79 895–79 907, 2019.

[26] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019, pp. 514–524.

[27] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in Robotics: science and systems, vol. 4, no. Citeseer. Citeseer, 2007, p. 1.

[28] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in IEEE International Conference on Robotics and Automation. IEEE, 2010, pp. 4372–4378.

[29] J. Castorena and S. Agarwal, "Ground-edge-based lidar localization without a reflectivity calibration for autonomous driving," IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 344–351, 2017.

[30] S. Chen, C.-W. Chang, and C.-Y. Wen, "Perception in the dark—development of a tof visual inertial odometry system," Sensors, vol. 20, no. 5, p. 1263, 2020.

[31] M. Delamare, R. Boutteau, X. Savatier, and N. Iriart, "Static and dynamic evaluation of an uwb localization system for industrial applications," Sci, vol. 2, no. 1, p. 7, 2020.

[32] J.-S. B. Valencia, M. R. Garcia, and J.-P. V. Ceballos, "A simple method to estimate the trajectory of a low cost mobile robotic platform using an imu," International Journal on Interactive Design and Manufacturing (IJIDeM), vol. 11, no. 4, pp. 823–828, 2017.

[33] F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," Sensors, vol. 17, no. 2, p. 271, 2017.

[34] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car," IEEE vehicular technology magazine, vol. 14, no. 2, pp. 103–111, 2019.

[35] M. Bernas, B. Płaczek, W. Korski, P. Loska, J. Smyła, and P. Szymała, "A survey and comparison of low-cost sensing technologies for road traffic monitoring," Sensors, vol. 18, no. 10, p. 3243, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3243/pdf

[36] K. G. Panda, D. Agrawal, A. Nshimiyimana, and A. Hossain, "Effects of environment on accuracy of ultrasonic sensor operates in millimetre range," Perspectives in Science, vol. 8, pp. 574–576, 2016.

[37] T. ETSI, "102 862 v1. 1.1 (2011-12) intelligent transport systems (its)," Performance Evaluation of Self-Organizing TDMA as Medium Access Control Method Applied to ITS, 2011.

[38] A. Festag, "Cooperative intelligent transport systems standards in europe," IEEE communications magazine, vol. 52, no. 12, pp. 166–172, 2014.

[39] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," Proceedings of the IEEE, vol. 99, no. 7, pp. 1162–1182, 2011.

[40] Z. Li, T. Yu, R. Fukatsu, G. K. Tran, and K. Sakaguchi, "Proof-of-concept of a sdn based mmwave v2x network for safe automated driving," in IEEE Global Communications Conference (GLOBECOM). IEEE, 2019, pp. 1–6.

[41] 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. Kobe, Japan, 2009, p. 5.

[42] S.-W. Kim, Z. J. Chong, B. Qin, X. Shen, Z. Cheng, W. Liu, and M. H. Ang, "Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results," in IEEE/RSJ International

Conference on Intelligent Robots and Systems. IEEE, 2013, pp. 5059–5066.

[43] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 663–680, 2014.

[44] D. G. Annkathrin Krämmer*, Christoph Schöller* and A. Knoll, "Providentia - a large scale sensing system for the assistance of autonomous vehicles," in Robotics: Science and Systems (RSS), Workshop on Scene and Situation Understanding for Autonomous Driving, 2019. [Online]. Available: https://sites.google.com/view/uad2019/accepted-posters

[45] M. Vasic, "Cooperative perception algorithms for networked intelligent vehicles," EPFL, Tech. Rep., 2017.

[46] B. Bilgin and V. Gungor, "Performance comparison of ieee 802.11 p and ieee 802.11 b for vehicle-to-vehicle communications in highway, rural, and urban areas," International Journal of Vehicular Technology, vol. 2013, 2013.

[47] J. Einsiedler, O. Sawade, B. Schäufele, M. Witzke, and I. Radusch, "Indoor micro navigation utilizing local infrastructure-based positioning," in IEEE Intelligent Vehicles Symposium. IEEE, 2012, pp. 993–998. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6232262

[48] G. Association et al., "Cellular-vehicle to everything (c-v2x)[internet]."

[49] G. Hinz, M. Buechel, F. Diehl, M. Schellmann, and A. Knoll, "Designing a far-reaching view for highway traffic scenarios with 5g-based intelligent infrastructure," in 8. Tagung Fahrerassistenz, 2017. [Online]. Available: https://mediatum.ub.tum.de/doc/1421303/file.pdf

[50] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges," Wireless networks, vol. 21, no. 8, pp. 2657–2676, 2015.

[51] S. Arai, Y. Shiraki, T. Yamazato, H. Okada, T. Fujii, and T. Yendo, "Multiple led arrays acquisition for image-sensor-based i2v-vlc using block matching," in IEEE 11th Consumer Communications and Networking Conference (CCNC). IEEE, 2014, pp. 605–610.

[52] S. M.-S. SADOUGH, "A tutorial on ultra wideband modulation and detection schemes," Shahid Beheshti University, Faculty of Electrical and Computer Eng., 2009.

[53] P. Gunturi, N. W. Emanetoglu, and D. E. Kotecki, "A 250-mb/s data rate ir-uwb transmitter using current-reused technique," IEEE Transactions on Microwave Theory and Techniques, vol. 65, no. 11, pp. 4255–4265, 2017.

[54] N.-S. Kim and J. M. Rabaey, "A high data-rate energy-efficient triple-channel uwb-based cognitive radio," IEEE Journal of Solid-State Circuits, vol. 51, no. 4, pp. 809–820, 2016.

[55] Y. Shi, X.-H. Peng, and G. Bai, "Cooperative v2x for cluster-based vehicular networks," International Journal on Advances in Networks and Services Volume 12, Number 3 & 4, 2019, 2019.

[56] M. Minea, "Cellular—sensorless v2i—based traffic information and communications infrastructure: Case study for high class motorways," in 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). IEEE, 2017, pp. 1–6.

[57] B. Zheng, P. Wang, F. Liu, and C. Wang, "Cooperative data delivery in sparse cellular-vanet networks," in 6th International Conference on Digital Home (ICDH). IEEE, 2016, pp. 128–132.

[58] P. E. Carnelli, M. Sooriyabandara, and R. E. Wilson, "Large-scale vanet simulations and performance analysis using real taxi trace and city map data," in IEEE Vehicular Networking Conference (VNC). IEEE, 2018, pp. 1–8.

[59] Bosch, "Conduite automatisée : comment les voitures et les infrastructures communiquent en milieu urbain," Website, Jul. 2020, accessed 2020-07-29. [Online]. Available: https://www.bosch.fr/actualites/2020/conduite%2Dautomatisee%2Dcomment%2Dles%2Dvoitures%2Det%2Dles%2Dinfrastructures%2Dcommuniquent%2Den%2Dmilieu%2Durbain/

[60] M. Gabb, H. Digel, T. Müller, and R.-W. Henn, "Infrastructure-supported perception and track-level fusion using edge computing," in IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 1739–1745.

[61] B. Yang, H. Dong, and A. El Saddik, "Development of a self-calibrated motion capture system by nonlinear trilateration of multiple kinects v2," IEEE Sensors Journal, vol. 17, no. 8, pp. 2481–2491, 2017.

[62] R. Xia, M. Hu, J. Zhao, S. Chen, Y. Chen, and S. Fu, "Global calibration of non-overlapping cameras: State of the art," Optik, vol. 158, pp. 951–961, 2018.

[63] L. Wang, J. Fernandez, J. Burgett, R. W. Conners, and Y. Liu, "An evaluation of network time protocol for clock synchronization in wide area measurements," in IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008, pp. 1–5.

[64] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "Correction de nuages de points lidar embarqué sur véhicule pour la reconstruction d'environnement 3D vaste," in Reconnaissance de Formes et Intelligence Artificielle (RFIA), Clermont-Ferrand, France, Jun. 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01906323

[65] A. Miller, K. Rim, P. Chopra, P. Kelkar, and M. Likhachev, "Cooperative perception and localization for cooperative driving," in IEEE International Conference on Robotics and Automation, Jul. 2020.

[66] F. Ghallabi, "Precise self-localization of autonomous vehicles using lidar sensors and highly accurate digital maps on highway roads," Ph.D. dissertation, Université Paris sciences et lettres, 2020.

[67] C. Gao, G. Zhao, and H. Fourati, Cooperative Localization and Navigation: Theory, Research, and Practice. CRC Press, 2019.

[68] D. Gulati, F. Zhang, D. Clarke, and A. Knoll, "Vehicle infrastructure cooperative localization using factor graphs," in IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016, pp. 1085–1090.

[69] E. Héry, P. Xu, and P. Bonnifait, "Pose and covariance matrix propagation issues in cooperative localization with lidar perception," in IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019, pp. 1219–1224.

[70] M. Rohani, D. Gingras, V. Vigneron, and D. Gruyer, "A new decentralized bayesian approach for cooperative vehicle localization based on fusion of gps and vanet based inter-vehicle distance measurement," IEEE Intelligent transportation systems magazine, vol. 7, no. 2, pp. 85–95, 2015.

[71] F. Ahammed, J. Taheri, A. Y. Zomaya, and M. Ott, "Vloci: Using distance measurements to improve the accuracy of location coordinates in gps-equipped vanets," in International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services. Springer, 2010, pp. 149–161.

[72] L. Altoaimy and I. Mahgoub, "Fuzzy logic based localization for vehicular ad hoc networks," in IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS). IEEE, 2014, pp. 121–128.

[73] J. A. del Peral-Rosado, J. A. López-Salcedo, S. Kim, and G. Seco-Granados, "Feasibility study of 5g-based localization for assisted driving," in International Conference on Localization and GNSS (ICL-GNSS). IEEE, 2016, pp. 1–6.

[74] O. Hassan, I. Adly, and K. Shehata, "Vehicle localization system based on ir-uwb for v2i applications," in 8th International Conference on Computer Engineering & Systems (ICCES). IEEE, 2013, pp. 133–137.

[75] C. Huang and X. Wu, "Cooperative vehicle tracking using particle filter integrated with interacting multiple models," in ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6.

[76] D. Gulati, F. Zhang, D. Malovetz, D. Clarke, G. Hinz, and A. Knoll, "Graph based vehicle infrastructure cooperative localization," in 20th International Conference on Information Fusion (Fusion). IEEE, 2017, pp. 1–6.

[77] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," Journal of Field Robotics, vol. 37, no. 3, pp. 362–386, 2020.

[78] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1907–1915.

[79] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, 2019, pp. 88–100.

[80] E. E. Marvasti, A. Raftari, A. E. Marvasti, Y. P. Fallah, R. Guo, and H. Lu, "Cooperative lidar object detection via feature sharing in deep networks," arXiv preprint arXiv:2002.08440, 2020.

[81] 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.

[82] Y. Wang, Y. Wu, and Y. Shen, "Cooperative tracking by multi-agent systems using signals of opportunity," vol. 68, no. 1, pp. 93–105, 2020, conference Name: IEEE Transactions on Communications.

[83] X. Chen, J. Ji, and Y. Wang, "Robust cooperative multi-vehicle tracking with inaccurate self-localization based on on-board sensors and inter-vehicle communication," vol. 20, no. 11, p. 3212, 2020, publisher: Multidisciplinary Digital Publishing Institute.

[84] C. Bo, H. Lu, and D. Wang, "Weighted generalized nearest neighbor for hyperspectral image classification," IEEE Access, vol. 5, pp. 1496–1509, 2017.

[85] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," EURASIP Journal on Image and Video Processing, vol. 2008, pp. 1–10, 2008.

[86] D. Bétaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 4, pp. 786–798, 2010.

[87] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in IEEE Intelligent Vehicles Symposium Proceedings. IEEE, 2014, pp. 420–425.

[88] T. ETSI, "102 863 (v1. 1.1): Intelligent transport systems (its); vehicular communications; basic set of applications; local dynamic map (ldm); rationale for and guidance on standardization," Technical report, ETSI, Tech. Rep., 2011.

[89] T. Sebastian, B. Wolfram, and F. Dieter, "Probabilistic robotics," 2005.

[90] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," Proceedings of the IEEE, vol. 94, no. 7, pp. 1384–1397, 2006.

[91] R. Jungnickel, M. Köhler, and F. Korf, "Efficient automotive grid maps using a sensor ray based refinement process," in IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016, pp. 668–675.

[92] P. Merriaux, R. Rossi, R. Boutteau, V. Vauchey, L. Qin, P. Chanuc, F. Rigaud, F. Roger, B. Decoux, and X. Savatier, "The vikings autonomous inspection robot: Competing in the argos challenge," IEEE Robotics & Automation Magazine, vol. 26, no. 1, pp. 21–34, 2018.

[93] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," Autonomous Robots, 2013, software available at http://octomap.github.com. [Online]. Available: http://octomap.github.com

[94] M. Drwiega, "Features matching based merging of 3d maps in multi-robot systems," in 24th International Conference on Methods and Models in Automation and Robotics (MMAR). IEEE, 2019, pp. 663–668.

[95] M. Soilán, A. Sánchez-Rodríguez, P. del Río-Barral, C. Perez-Collazo, P. Arias, and B. Riveiro, "Review of laser scanning technologies and their applications for road and railway infrastructure monitoring," Infrastructures, vol. 4, no. 4, p. 58, 2019.

[96] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," arXiv preprint arXiv:2001.07715, 2020.

[97] H. Yang and L. Carlone, "A polynomial-time solution for robust registration with extreme outlier rates," arXiv preprint arXiv:1903.08588, 2019.

[98] J. Wu, H. Xu, and W. Liu, "Points registration for roadside lidar sensors," Transportation Research Record, p. 0361198119843855, 2019.

[99] E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer, "The ko-per intersection laserscanner and video dataset," in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014, pp. 1900–1901.

[100] J. Ward, S. Worrall, G. Agamennoni, and E. Nebot, "The warrigal dataset: Multi-vehicle trajectories and v2v communications," IEEE Intelligent Transportation Systems Magazine, vol. 6, no. 3, pp. 109–117, 2014.

[101] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 3354–3361.

[102] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle et al., "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," arXiv preprint arXiv:1910.03088, 2019. [Online]. Available: https://arxiv.org/pdf/1910.03088.pdf;

[103] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in Proceedings of the 1st Annual Conference on Robot Learning, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 1–16. [Online]. Available: http://proceedings.mlr.press/v78/dosovitskiy17a.html

[104] C. Quiter and M. Ernst, "deepdrive/deepdrive: 2.0," 2018.

[105] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020, pp. 1–6.

[106] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in Field and Service Robotics, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[107] W. Liu and Y. Shoji, "Edge-assisted vehicle mobility prediction to support v2x communications," IEEE Transactions on Vehicular Technology, vol. 68, no. 10, pp. 10 227–10 238, 2019.

[108] X. Chen, J. Ji, and Y. Wang, "Robust cooperative multi-vehicle tracking with inaccurate self-localization based on on-board sensors and inter-vehicle communication," Sensors, vol. 20, no. 11, p. 3212, 2020.

Antoine CAILLOT received a Master of Engineering in Embedded Systems for Autonomous Vehicles from ESIGELEC (France, 2019). He is working toward a PhD in Computer Science from the University of Rouen and is affiliated with IRSEEM Lab. His research interests include cooperative perception and intelligent transportation systems.

Safa OUERGHI received her Master degree from SUP'COM (Higher School of Communication of Tunis) in Information and communication Technology in 2012. She received her PHD from SUP'COM in collaboration with ESIGELEC in 2018 for her work related to vision-based vehicle localization. From 2020, she is an Associate Professor at the ESIGELEC engineering school and a researcher in the IRSEEM research institute. Her research interests are perception, localization and computer vision dedicated to robotics and autonomous vehicles.

Pascal VASSEUR is full professor at the Université de Picardie Jules Verne (France) and is a member of the MIS laboratory. His research interests are computer vision and image processing and their applications to intelligent transportation, mobile and aerial robots.

Rémi BOUTTEAU received his engineering degree from the IMT Lille Douai and his MSc degree in Computer Science from the University of Lille in 2006. In 2010, he received his PhD degree from the University of Rouen Normandy for works related to Computer Vision (catadioptric sensors, 3D reconstruction, Structure-from-Motion). From 2009 to 2020, he was an Associate Professor at the ESIGELEC engineering school and a researcher in the IRSEEM research institute. Since 2020, he is a Full Professor at University of Rouen Normandy within the STI team (Intelligent Transportation System) at the LITIS Lab (IT Laboratory, Information Processing and Systems). His research interests are perception, localization and computer vision dedicated to autonomous vehicles.

Yohan DUPUIS received the MSc in Electrical Engineering from Union Graduate College, NY and a MEng from ESIGELEC, France, in 2009. In 2012, he earned a PhD in Computer Science from Université de Rouen. He is now Research Director at LINEACT CESI. His research interests focus on perception for ground vehicle-infrastructure interaction understanding.