Implementation and Latency Assessment of a Prototype for C-ITS Collective Perception

Francesca Pacella
LINKS Foundation
Torino, Italy
francesca.pacella@linksfoundation.com

Edoardo Bonetto
LINKS Foundation
Torino, Italy
edoardo.bonetto@linksfoundation.com

Guido A. Gavilanes Castillo
LINKS Foundation
Torino, Italy
guido.gavilanes@linksfoundation.com

Daniele Brevi
LINKS Foundation
Torino, Italy
daniele.brevi@linksfoundation.com

Riccardo Scopigno
LINKS Foundation
Torino, Italy
riccardo.scopigno@linksfoundation.com

***Abstract*—Nowadays, services in the framework of Cooperative Intelligent Transport Systems (C-ITS) are evolving to tackle more complex scenarios. The State-of-Art C-ITS messages focus on the sharing of information about specific events, hazards or about the position and dynamic status of connected vehicles. New C-ITS services are currently being developed to enable connected road actors to share detailed information gathered from local sensors. The aim is to create a Collective Perception that provides a better knowledge of the surrounding environment at the vehicle-side. In this paper, we focus on a collective perception service that a roadside unit implements to provide information from fixed sensors to connected vehicles. The contribution of this paper is the implementation of a prototype of a collective perception service according to ETSI standards. The prototype includes the development of a platform that processes the sensors' raw data to obtain usable information to share along with a communication facility for sharing the information through specific C-ITS messages. Further contribution concerns the assessment of the end-to-end latency of this service in the developed prototype. The measurements show that the required time to provide the information from the roadside unit to the vehicle is below the threshold recommended in relevant standards. A latency of 250 ms has been estimated for first-time detected objects, while the latency of already tracked objects is of 157 ms. This confirms the viability of the prototype for effectively providing useful information to connected vehicles.**

Keywords— Collective Perception, CAV, C-ITS, V2I, latency, prototype

I. Introduction

A driving vehicle can be defined fully autonomous if it can drive in an environment without requesting the intervention of a human driver. An autonomous vehicle has to decide the route it should follow (*Path Planning*), to determine how to achieve the desired route (*Guidance*), to understand where it is (*Navigation*), and to actuate the proper commands to achieve the desired route (*Control*) [1].

The Guidance of the vehicle also includes the exploitation of the information about obstacles that are received from the vehicle's perception system. Knowing the obstacles present in the surrounding environment is indeed required to refine the trajectory of the vehicle avoiding potential collisions. The identification of obstacles is very challenging since their type, dimension and dynamic status can significantly vary. Further complexity is provided by the unknown behaviors that other road actors (e.g., vehicles, pedestrians, animals) can have.

The perception system of an autonomous driving vehicle is usually made by several sensors, based on different technologies (e.g., camera, LiDAR, RADAR), that can provide different kinds of information for different ranges. The ideal perception system should be able to identify obstacles at 360° at specific distances. However, there are some situations in which a perception system cannot detect obstacles due to physical constraints. For example, obstructions, due to buildings or to vehicles, can limit the range of the perception system being a serious issue for safety.

A possible solution, which can mitigate the possible vulnerabilities of a perception system, is the cooperation among the road actors (e.g., vehicles and road-side infrastructure). The exchange of information using vehicular communication (V2X) can help in the identification of potential obstacles. This is the basis of Cooperative Intelligent Transport Systems (C-ITS). The CAR 2 CAR Communication Consortium identifies three main categories of information that can be exchanged in the context of C-ITS [2]:

- *Awareness Driving*: it concerns the exchange of information about vehicles' dynamics (e.g., position, speed, direction) and of well-defined events (e.g., accident, adverse weather conditions, ...), the exchanged information is only related to connected road actors;
- *Sensing Driving*: road actors share the information provided by their own sensors, in this case the shared information can permit to identify also non-connected road actors;
- *Cooperative Driving*: the interactions among road actors are not limited to the sharing of information, but they can coordinate their maneuvers to ensure enhanced safety.

The possibility to exchange information from sensors is particularly relevant for enhancing safety in autonomous driving. The "*Awareness Driving*" use cases have been already explored and testbeds have been implemented [3], while "*Sensing Driving*" use cases have only been recently approached. The 5G Automotive Association (5GAA) introduced the "*High Definition Sensor Sharing*" use case for improving the perception of the environment by exploiting information from other vehicles [4]. The sharing of processed sensors' information using V2X messages is targeted by the Collective Perception Service (CPS) that is a C-ITS service firstly proposed in [5] and currently under-going ETSI standardization as detailed in Sect. III.

This paper focuses on the sharing of information from road-side fixed sensors to Connected and Automated Vehicles (CAVs). This work deals with this approach since it has been developed within the framework of the European H2020 project ICT4CART whose main goal is to foster autonomous driving to higher levels of automation thanks to the support of an ICT road-side infrastructure.

The main contribution of this paper consists in the development of a complete prototype of a CPS framework. This prototype includes the development of a Perception Processing Platform (3P) and of an ETSI-compliant CPS at the roadside. The 3P framework is a system for processing road-side sensors' data to feed the CPS with the required information. This system exploits data from camera to identify, track and predict the dynamics of selected road actors (i.e., vehicles, bicycles, pedestrians, and animals). Further contribution is related to the assessment of the end-to-end latency of the overall application. The end-to-end latency is computed as the interval between the time instant at which sensors' data are available and the time instant at which the information is processed at the receiving CAV.

The following of the paper is structured as follows. Sect. II introduces related works on collective perception and on tracking solutions, while Sect. III provides some insights about the CPS. In Sect. IV, the 3P platform is detailed. The fine-tuning of the main configuration parameters of the platform is introduced in Sect. V where latency assessment results are also provided. Further works and conclusions are outlined in Sect. VI.

II. Related Works

The concept of collective perception is of particular interest for vehicles or robots that autonomously move in an unknown environment [5],[6]. The sharing of information can help to identify obstacles not perceived by all actors and it can ease the navigation of the environment. Collective perception is currently a trending topic in the C-ITS context. CAVs, infrastructure and V2X-enabled actors can share the information of sensors to enhance the knowledge of every single actor. The exchange of information can be limited among connected vehicles [5] or it can be performed between vehicles and artificial intelligence placed on the infrastructure [7]. Alternatively, as proposed in [8], information sharing can involve several connected actors such as infrastructure, body-worn mobile devices, and vehicles.

Raw data from sensors need to be processed to retrieve information about type, position, and dynamics of objects. In our implementation, the 3P platform provides this information to the CPS exploiting detection and tracking algorithms.

Object detection is a well-studied topic. In this work, we do not aim to introduce new object detection methods, but we exploit existing State-of-Art solutions as basis for the development of the 3P platform. We refer to the surveys [9], [10] for further details about object detection.

A Multiple Object Tracking (MOT) approach is required in the context of Collective Perception since multiple objects need to be followed. The MOT problem can be divided in two main tasks: i) identify the objects to track and ii) follow the objects over time. Several MOT solutions have been proposed in the literature. We refer to [9], [11], [12] for detailed surveys on these solutions. The MOT problem can adopt two different methods for following objects: a computer vision approach or a point-based problem approach. In the first case, visual information from images is used frame by frame for predicting the object position along the whole video (e.g., comparing an appearance model at each frame) as done in [13]. In the second case, only information about the object position is required. For example, [7] proposes a generic interface which enables different types of sensors to be connected for providing object positions that are exploited for tracking the objects using a Labeled-Multi-Bernoulli Filter based method.

The MOT algorithm that we use for developing the 3P platform is based on different solutions that we exploited to create a personalized MOT method that best matches the requirements of our context. The details about the implemented tracking solution are provided in Section IV.

III. C-ITS Collective Perception Service

The specification of the C-ITS CPS indicates which information retrieved from sensors' data can be shared and how. The specification does not include how to retrieve information from raw data' sensors (i.e., object detection and tracking). This aspect is not standardized, and it is considered manufacturer-dependent. The ETSI Technical Specification 103 324, which is currently under drafting at the time of writing this paper, aims to specify the CPS. ETSI already publicly released the Technical Report 103 562 as preliminary document for the description of the CPS [14]. We refer to this document for the details, while in the following of this section we outline main characteristics and features of the CPS.

The CPS specification defines the new C-ITS message named Collective Perception Message (CPM), for sharing perception information. The CPM contains information of the ITS-Station generating it, the available sensors at that ITS-Station and the objects detected. Further information that can be optionally provided in the CPMs are the free space areas detected. The CPMs are generated by the CPS periodically providing information of newly detected objects or of moving objects. The CPS broadcasts the CPMs generated and processes the received CPMs. The CPM is basically an object-based representation of an environmental perception model. The object-related information to be inserted into a CPM concerns the type, the position, and the dynamics (e.g., speed and heading) of an object. Additional information can also be provided such as the volume occupancy and the acceleration.

IV. Perception Processing Platform

The 3P platform provides information about objects to the CPS. Exploiting sensors' raw data, the 3P platform identifies objects, determines their position, and computes their dynamics. Our implementation is based on a camera as the main sensor for the perception of road objects. The processing of the information must be accurate and fast since this information is used by CAVs to decide their maneuvers. Not precise or old information can negatively impact their decisions. The 3P platform is made of two main components: the *Object Detection* (OD) unit and the *Multiple Object Tracking* (MOT) unit.

A. Object Detection unit

The OD unit takes as input the image frames from the camera to detect objects in the field of view of the camera. The output of the OD unit for each frame consists of a list of the identified objects, specifying their types and their position in the frame (i.e., a bounding box of the portion of the frame in which the object is to be recognized). The OD unit is based on

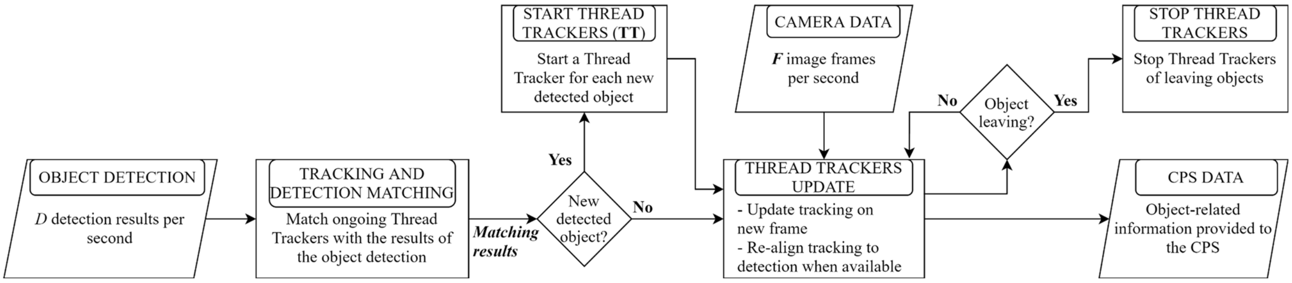
START THREAD TRACKERS (TT)
Start a Thread Tracker for each new detected object
CAMERA DATA
F image frames per second
STOP THREAD TRACKERS
Stop Thread Trackers of leaving objects
No
Object leaving?
Yes
OBJECT DETECTION
D detection results per second
TRACKING AND DETECTION MATCHING
Match ongoing Thread Trackers with the results of the object detection
Matching results
Yes
New detected object?
No
THREAD TRACKERS UPDATE
- Update tracking on new frame
- Re-align tracking to detection when available
CPS DATA
Object-related information provided to the CPS

Fig. 1. Inner structure of the MOT unit.

the open-source neural network framework Darknet and the You Only Look Once v3 (YOLOv3) that is a real-time high-performant object detections system based on deep-learning [15].

B. Multiple Object Tracking unit

The MOT unit is based on a scalable multi-threading approach like the one proposed in [16]. In detail, the MOT unit launches a program thread for tracking each object. We refer to this tracking thread with the name of Thread Tracker. A certain Thread Tracker tracks the assigned object as it was the only one in the frame. We selected this approach since it guarantees the scalability to easily follow all identified objects whose number can continuously change at each frame.

The tracking algorithm performed by each Thread Tracker is the Kernelized Correlation Filter (KCF) [17]. The selected KCF algorithm is an on-line and computer-vision based solution. The on-line feature is a mandatory choice since on-line tracking can provide results at each frame as envisaged for the CPS that requires input information with low latency. The KCF algorithm is a computer-vision approach since it applies a correlation filter on the image in the surrounding of the last known position of the object. The correlation filter technique is performed in the frequency domain allowing the tracking algorithm to easily achieve high frame per second rate. Detailed explanations about correlation filter tracking can be found in [18].

The MOT unit implements a partial detection-based approach. The detection information provided by the OD unit is used to start the tracking of new objects and to realign the already tracked objects. The KCF algorithm uses its own tracking predictions to update the correlation filter to better follow the object, but this approach accumulates errors resulting in a drift in the object tracking. This issue can be solved by re-initializing the KFC algorithm with the information received from the OD unit. A fully detection-based tracking approach was not selected since detection times may be too high due to the deep-learning based approach for the object detection. Thus, we preferred to provide low latency information to the CPS taking care to reduce as much as possible the inaccuracies by periodically realigning detection and tracking results.

The inner structure of the MOT unit is shown in Fig. 1. The MOT unit receives in input the image frames from the camera and the object detection information from the OD unit. The output of the MOT consists in tracking information (e.g., position of the tracked objects) to the CPS. The main modules of the MOT unit are described in the following subsections.

1) Start Thread Tracker

This module launches a new Thread Tracker for each newly detected object. The notification of a new object is received from the *Tracking and Detection Matching* module that provides the information about the bounding box coordinates of the object to track. The Thread Tracker initializes the tracking filter of the KCF algorithm extracting color-based features of the bounding box area. At the start of the MOT unit, since no object is tracked, this module launches a Thread Tracker for each detected object.

2) Tracking and Detection Matching

This module receives as input the information of the detected objects from the OD unit, and it matches them with the currently tracked ones. In case that this module identifies a newly detected object, it sends the object information to the *Start Thread Tracker* module, otherwise it provides the matching and the detection information to the *Thread Trackers Update* module.

The matching between detected and tracked objects is based on a score computed on geometrical features of their bounding boxes. This score is the normalized and weighted sum of several metrics (e.g., the distance between the centers of the bounding boxes, the ratio of the dimensions, etc.). The optimal matching solution is determined by applying the Hungarian algorithm, which is widely used in the tracking field [18], [19]. It is a combinatorial optimization algorithm that solves the problem in polynomial time finding an optimal assignment. This matching procedure aims to minimize incorrect assignments, but these may still happen. For example, if a tracked object does not have its corresponding detection and a new object is detected enough close to it.

3) Thread Trackers Update

Each Thread Tracker receives in input the frames from the camera, and it updates the position of the tracked object based on the output of the KCF algorithm. If the algorithm cannot provide the new object position with an adequate confidence level (e.g., due to an occlusion), the Thread Tracker updates the position of the tracked object considering its estimated trajectory based on previous frames.

Each Thread Tracker receives also in input the information about the matching with detected objects from the *Tracking and Detection Matching* module. If a detected object is assigned to the specific tracked object, the Thread Tracker use this information to refine the position of bounding box and to reinitialize the tracking filter of the KCF algorithm.

4) Stop Thread Tracker

The Thread Tracker defines a tracked object as leaving the scene considering these two conditions: 1) the bounding box of the tracked object is close to the borders of the image frame and its estimated trajectory is going outwards the frame; 2) the tracked object does not match with any detection for n consecutive detections provided by the OD module. In this work, the value of n has been fixed to 5. If at least one

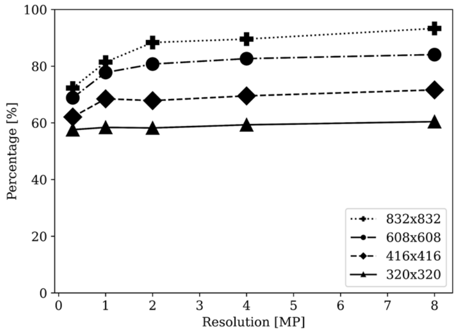

Fig. 2. Percentage of average number of objects detected for each input layer dimension and image resolution with respect to the ground truth.

condition is satisfied the *Stop Thread Tracker* module terminates the thread and free the associated resources.

V. Implementation

A Road Side Unit (RSU) hosts the 3P platform, the CPS, the ITS-G5 communication stack, and associated services. The RSU implements the connection with the camera and it manages the transmission of C-ITS messages with a ITS-G5 modem. All the software modules and related algorithms are implemented on the RSU. The RSU is based on a Nvidia Jetson AGX Xavier board that is a high-end embedded board offering "*a 512-core Volta GPU with 64 Tensor Cores, a 8-core Carmel ARM v8.2 64-bit CPU, and a 32GB 256-Bit LPDDR4x memory*". The vehicle is equipped with an On-Board Unit (OBU) based on an ARM embedded board. The OBU and the RSU are running the ITS communication stack developed by LINKS Foundation. The OBU processes the received C-ITS messages and provides information to relevant applications, such as a Human-Machine Interface (HMI) or the autonomous driving module.

The image frames are provided by an IP camera in the same local network of the Xavier board using Real Time Streaming Protocol (RTSP). The information among the different modules of the 3P platform is exchanged through a Message Queue Telemetry Transport (MQTT) message broker that runs on the Xavier board.

VI. Results

In the first subsection, we introduce the tuning of the main parameters of the 3P platform. The goal is to find the best trade-off between accuracy and latency. The tuning has been done considering an offline recorded video to test the different combinations of parameters in the same situation.

In the second subsection, we provide the assessment of the end-to-end latency that corresponds to the interval between the instant at which a frame is provided to the 3P platform and the instant at which the information is provided to the HMI or Advanced Driving Assistance Systems (ADAS) on the receiving vehicle. The end-to-end latency has been measured considering a live video.

A. Sensitivity analysis of 3P platform's parameters

The parameters of the OD unit are the size of the input layer of the YOLOv3 neural network and the resolution of the received images. The MOT unit has only the video resolution as input configuration parameter. Following sizes of input layers have been tested: 320x320, 416x416, 608x608, and 832x832. The offline video is considered with the following resolutions 0.5, 1, 2, 4, and 8 MP for the OD unit and 0.3, 0.5, and 1 MP for tracking operations in the MOT unit. The offline video lasts 10 seconds with a frame rate of 25 fps. The ground truth of object detection has been generated by running the YOLOv3 network with a 1600x1600 input layer and using an image resolution of 8 MP. The tracking ground truth has been created performing offline the tracking with the detection information of the ground truth received at each frame.

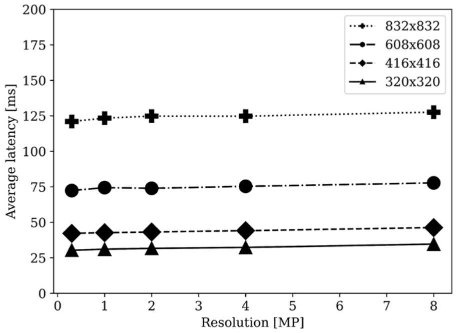

Fig. 3. Average processing time of the object detection for each input layer size and image resolution.

Fig. 2 shows the percentage of the average number of detected objects in all the video for each possible combination of resolution and input layer size with respect to the ground truth. The average time employed by YOLOv3 to process one frame is depicted in Fig. 3 for the same set of cases.

The increase of the image resolution is more impacting on the percentage of detected objects when the size of the neural network is larger. The impact of the resolution on the detection time is instead negligible. Considering these results, the image resolution for the object detection shall be at least 2 MP. Higher image resolutions can also be used since they do not impact the detection latency, while they slightly improve the accuracy of the object detection.

In Fig. 2 it is possible to notice that the percentage of detected objects increases with the size of the neural network. This behavior is expected due to the working principle of YOLOv3 [15]. The percentage gap with respect to the ground truth is mainly due to vehicles in the background of the scene that are identified only when they get closer to the camera. We estimated that the 608x608 and the 832x832 neural networks can detect objects approximately at 400 and 500 meters. Both distances can be considered sufficient to not impair the effectiveness of the CPS even at high speeds.

The higher accuracy of larger input layers has the drawback of a significant increase of the processing time required by the object detection as shown in Fig. 3. This time directly impacts on the performance of the MOT since it delays the instant at which the MOT unit can exploit the detection information to reduce the tracking errors. However, the tracking performance is also conditioned by the object detection accuracy. A trade-off between these two aspects needs to be found. It is necessary to evaluate if the tracking can provide better results if it receives less accurate detection information but sooner or if it receives more accurate detections after a longer time period.

This evaluation is performed considering the MOT unit to receive input detection information with different delays depending on the size of the neural network used. In the specific, considering the computed object detection latencies

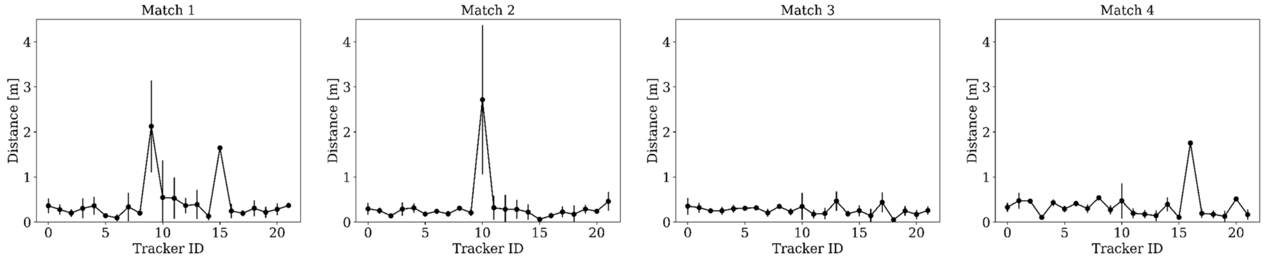

Fig. 4. Average distance error for the tracking results of the tracked objects (Tracker ID) for different object detections obtained with neural network of size 320x320, 416x416, 608x608, and 832x832 for respectively "Match 1", "Match 2", "Match 3" and "Match 4".

shown in Fig. 3 and that the time between two consecutive frames is of 40 ms in the offline video (i.e., the frame rate is 25 fps), the delay in the communication of detection information between the OD unit and the MOT unit results to be of 1, 2, 3 and 4 frames for respectively 320x320, 416x416, 608x608, and 832x832 neural networks. The object detection is performed keeping fixed the image resolution to 4 MP, while the MOT unit uses a 1 MP resolution of the video.

The tracking results are compared with respect to the tracking ground truth. The comparison has been done by measuring the pixel distance between the tracking position and the tracking ground truth position for each object. To provide more understandable results, we convert the pixel distance to meter distance. The conversion is done considering that the bounding box of the object is around 1.8 meters since the object is a car that is seen from an almost frontal visual perspective. It was not possible to perform a georeferentiation of the objects since the camera position and its calibration parameters are not available for the offline video.

Fig. 4 introduces the results of the comparison for tracking performed considering the previously illustrated object detection configurations. "*Match 1*" provides the results for tracking with object detection based on 320x320 neural network, "*Match 2*", "*Match 3*" and "*Match 4*" for respectively 416x416, 608x608, and 832x832 neural networks. The four plots report the average distance error of a tracked object with respect to the tracking ground truth for 22 different objects tracked along the video. The average error is below one meter in all cases. The peaks that it is possible to notice are vehicles whose identifier is not correctly assigned to the correct object during the matching algorithm phase. This is due to low-quality detection information (e.g., duplicated detected objects) or to lately received detection input providing outdated information (e.g., detected object located in an old position too different from the actual one). The average distance errors for the different cases are 0.44 m for "*Match 1*", 0.36 m for "*Match 2*", 0.26 for "*Match 3*", and 0.37 m for "*Match 4*". These results indicate that the smaller neural networks provide object detection with low accuracy impairing the performance of the tracking algorithm, while the high delay of the largest neural network makes the performance of the tracking to degrade since information even if accurate is received too late. The performed analysis points out that the neural network size of 608x608 should be selected. This configuration is a good compromise between object detection latency and accuracy resulting in more accurate tracking performance.

Another parameter that impacts both accuracy and latency of the tracking algorithm is the resolution of the video used by the MOT unit. The tracking algorithm is indeed based on a KCF filter that adopts a computer vision approach that performs pixel-level operations. The analysis of this parameter has been done considering that the object detection configuration has been kept fixed with a neural network of 608x608 and an image resolution of 4 MP. We report the average latency per object since the KCF filter initialization has to be performed for each object making the tracking time to be dependent on the number of the objects in a frame. The average latency for 1 MP resolution is of 2.4 ms per object. Decreasing the image resolution, the objects are characterized by a smaller number of pixels reducing the time need by the MOT unit to perform the tracking. A video resolution of 0.5 MP has a tracking latency of 1.6 ms and a resolution of 0.3 MP has a 1.2 ms latency.

The impact of the different image resolutions on the tracking accuracy has been evaluated comparing the results of tracking with 1 MP with those provided by using lower resolutions. Using the 0.5MP video the tracking accuracy is characterized by an average distance error of about 6.5 cm and standard deviation of 3 cm with respect to the tracking results obtained with the 1 MP. Considering tracking with a 0.3 MP video resolution the average distance error becomes 13.5 cm with a standard deviation of 15 cm. This significantly higher standard deviation means that tracking is not able to correctly follow the cars due to not correct matching during the update phases. Thus, the tracking algorithm can be run also receiving in input a 0.5 MP video lowering its latency and only slightly worsening its accuracy.

B. Assessment of the End-to-End Latency

The measurement of the end-to-end latency has been done considering a live video from the IP camera Vivotek IB9387-HT. The OD unit has been configured to use a 608x608 neural network and a 5 MP video stream with a frame rate of 30 fps. The MOT unit receives in input a 0.5 MP video stream with a frame rate of 30 fps. The latency measurements have been based on 500 consecutive frames.

The end-to-end latency is made by the following contributions: 1) the object detection latency, 2) the tracking latency, 3) the latency introduced by the CPS to create CPM messages, 4) the latency for coding, transmission at the RSU side and for the decoding at the OBU side. The latency contribution of the object detection should be considered just for objects that are detected the first time. After the initial detection, information about objects is directly updated by the MOT unit reducing the overall latency.

The measured object detection latency has an average value of 93 ms. The higher detection time is mainly due to the different image frame proportions with respect to the offline video used in the sensitivity analysis. However, this latency allows to provide detection information to the MOT unit with three frames of delay that guarantees the best tracking accuracy as found in the sensitivity analysis.

The tracking latency of the live video is on average 30 ms. In the specific, about 30 objects have been tracked corresponding to an average latency per object of 1 ms. The small difference with respect to the offline video is due to the different scenes that are analyzed. In the live video, objects have smaller dimensions with respect to the overall image frame size due to the different visual perspective. This reduces the time employed by the KCF filter lowering the latency introduced by the MOT unit. The tracking information provided by the MOT unit is received by the CPS that processes it for preparing the CPM messages that are then coded and sent using an ETSI standard compliant ITS communication stack. The CPS takes on average 1.25 ms since the reception of the tracking information to prepare the CPMs to be sent. The encoding/decoding operations and the transmission operations at the RSU and the OBU side account for a total amount of 25 ms on average.

The overall latency is then around 150 ms for the first-time detected object, while the information about objects that were already detected in previous frames can be provided to connected vehicles in about 57 ms. These values do not consider the intergeneration time between two CPMs that needs to be observed as defined in the ETSI standard. The minimum value of intergeneration time is equal to 100 ms. Considering the worst case in which the information is provided to the CPS just after a CPM is sent, the previous values should be incremented of 100 ms becoming 250 ms for first-time detected objects and 157 ms for already known objects. The obtained latency values indicate that the information provided by the CPS can be exploited by the receiving vehicles since standard threshold value for collision risk warning applications is 300 ms as defined in the standard ETSI TS 101 539-3. Higher values are not acceptable since the received information may be too old to allow prompt response from AD systems or human driver to take actions.

VII. Conclusions and Future Works

In this paper, we presented a prototype of a roadside unit implementing a Collective Perception Service that shares information from road sensors to vehicles. The prototype includes a Perception Processing Platform (3P), which is used to process data from sensors, and the C-ITS service that broadcasts the information retrieved from the platform to the connected vehicles using standard C-ITS messages.

We performed an assessment of the end-to-end latency for evaluating the effectiveness of the implemented service to provide information to connected vehicles. The results show that our implementation can satisfy the latency requirements indicated in relevant standards. Information about first-time detected objects can be provided to vehicles in 250 ms, while 157 ms is the time needed for already detected objects.

Future works will concern the improvement of the 3P platform. One objective is to include a LiDAR sensor that can provide accurate distance measurements of the detected objects. A further challenge is to integrate multiple cameras to provide the view from different perspectives. This can enhance the accuracy of the Collective Perception model, but information integration must be done accurately to avoid duplicates. Other objectives are to include in the 3P platform new modules for retrieving additional data such as free space areas, geographic coordinates and spatial occupancy of the detected objects.

Acknowledgments

The work described in this paper was performed with the support of the ICT4CART project that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 768953.

References

[1] A. M. Lekkas, "Guidance and path-planning systems for autonomous vehicles", Ph.D. Thesis, Norwegian University of Science and Technology, 2014.

[2] CAR 2 CAR Communication Consortium, "C-ITS: Cooperative Intelligent Transport Systems and Services".

[3] G. A. G. Castillo, E. Bonetto, D. Brevi, F. Scappatura, A. Sheikh and R. Scopigno, "Latency assessment of an ITS safety application prototype for protecting crossing pedestrians," IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020.

[4] 5G Automotive Association (5GAA), "5GAA Releases White Paper on C-V2X Use Cases: Methodology, Examples and Service Level Requirements".

[5] J. H. Günther, H. R. Riebl, L. Wolf, and C. Facchi, "Collective perception and decentralized congestion control in vehicular ad-hoc networks", IEEE Vehicular Networking Conference (VNC), 2016.

[6] T. Schmickl, C. Möslinger, and K. Crailsheim, "Collective perception in a robot swarm", International Workshop on Swarm Robotics (pp. 144-157), Springer, September 2006.

[7] M. Herrmann, J. Müller, J. Strohbeck, and M. Buchholz, "Environment Modeling Based on Generic Infrastructure Sensor Interfaces Using a Centralized Labeled-Multi-Bernoulli Filter", IEEE Intelligent Transportation Systems Conference (ITSC), October 2019.

[8] M. Bieshaar, G. Reitberger, S. Zernetsch, B. Sick, E. Fuchs, and K. Doll, "Detecting intentions of vulnerable road users based on collective intelligence", arXiv preprint arXiv:1809.03916, 2018.

[9] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods", in International Journal of Innovative Research in Computer and Communication Engineering, U.K., 2014.

[10] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey", in International Journal of Computer Vision, 128(2), 261-318, 2020.

[11] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T. K. Kim, "Multiple object tracking: A literature review", arXiv preprint arXiv:1409.7618, 2014.

[12] L. Fan, Z. Wang, B. Cail, C. Tao, Z. Zhang, Y. Wang, F. Zhang, "A survey on multiple object tracking algorithm", in 2016 IEEE International Conference on Information and Automation (ICIA), August 2016.

[13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric", in 2017 IEEE International Conference on Image Processing (ICIP), IEEE, September 2017.

[14] ETSI TR 103 562 V2.1.1 (2019-12) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2, December 2019.

[15] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement", arXiv preprint arXiv:1804.02767, 2018.

[16] T. Said, S. Ghoniemy, and O. Karam, "Real-time multi-object detection and tracking for autonomous robots in uncontrolled environments", Seventh International Conference on Computer Engineering & Systems (ICCES), pp. 67-72, IEEE, November 2012.

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters", IEEE transactions on pattern analysis and machine intelligence, 37(3), 583-596, 2014.

[18] Z. Chen, Z. Hong, and D. Tao, "An experimental survey on correlation filter-based tracking", arXiv preprint arXiv:1509.05520, 2015.

[19] A. Bewley, Z. Ge, L. Ott F. Ramos, and B. Upcroft, "Simple online and realtime tracking", in 2016 IEEE International Conference on Image Processing (ICIP) (pp. 3464-3468), IEEE, September 2016.

[20] F. Luetteke, X. Zhang, and J. Franke, "Implementation of the hungarian method for object tracking on a camera monitored transportation system", in ROBOTIK 2012, 7th German Conference on Robotics (pp. 1-6), VDE, May 2012