Decentralized perception system with multiple viewpoints

Quentin Picard, Malo Morice, Maryem Fadili, Steve Pechberti

Decentralized perception system with multiple viewpoints

Quentin Picard, Malo Morice, Maryem Fadili, Steve Pechberti
Institut VEDECOM; 23 bis Allée des Marronniers, 78000 Versailles, France

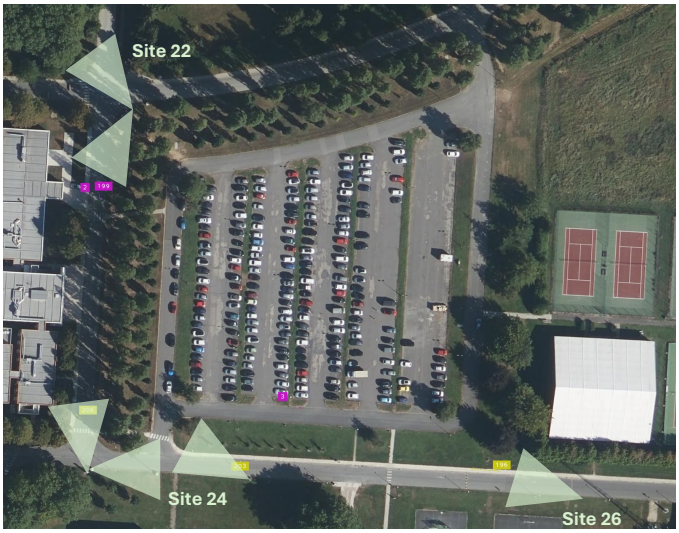
Abstract

Vehicle-to-Infrastructure (V2I) cooperation has emerged as a fundamental technology to overcome the limitations of the individual ego-vehicle perception. Onboard perception is limited by the lack of information for understanding the environment, the lack of anticipation, the drop of performance due to occlusions and the physical limitations of embedded sensors. V2I perception in a cooperative manner improves the ego-vehicle perception range by receiving information from the infrastructure that has another point of view, mounted with sensors, such as camera and LiDAR. This technical paper presents a perception pipeline developed for the infrastructure, based on images with multiple viewpoints. It is designed to be scalable and has five main components: the image acquisition for the modification of camera settings and to get the pixel data, the object detection for fast and accurate detection of four wheels, two wheels and pedestrians, the data fusion module for robust fusion of the 2D bounding boxes from multiple viewpoints, the object tracking to get the history of movement for each object over time and the generation of perception message for V2I communication. The infrastructure-based solution has been implemented and demonstrated in real-world scenarios, including two different intersections with up to six mounted cameras to cover an extended area. The qualitative results show that detected objects have high accuracy with similar performances between two different environments, which proves the scalability of the solution. With a not optimized setup for these first deployments, we observe for the whole pipeline an execution time between 226ms and 256ms depending on the number of objects to be fused in the map based on the processing of six cameras.

Introduction

Significant progress has been made for the onboard perception for the last few years. Learning-based methods have become the state-of-the-art for several perception tasks such as, object detection [1], bounding boxes and segmentation masks for each instance of an object [2], multiple object tracking [3]. The Vehicle-to-Infrastructure (V2I) cooperation improves the ego-vehicle perception range by receiving information from other agents. The extended infrastructure point of view to cover large areas allows to overcome the limitations of the onboard perception that are characterized as follows: the insufficient environment perception information [4], the lack of anticipations, the drop of performances due to occlusions and the physical limitations of onboard sensors.

Many challenges are tackled to build a robust V2I system [5, 6]. It involves different kind of expertise, from the choice, the quantity and the position of sensors, the used hardware, the communication network architecture to retrieve the sensors stream and the wireless network to use for V2X communication. In terms of decentralized perception, the main challenge is to provide accurate and robust data that can be used by the road users under different lights, weather and traffic conditions. This represents a significant cost, when these challenges require a combination of multimodal sensors, such as visible cameras and LiDAR.



(a)

(b)

Figure 1: Implementation in real-world scenarios of the proposed decentralized perception system in two intersections. In (a), six cameras have been installed to cover an L-shape intersection. In (b), four cameras have been installed to cover the second intersection. The bird's eye view (BEV) maps are orthonormal and downloaded from the IGN BD ORTHO® data. The perception system has been tested in different weather conditions at different time of the day which corresponds to a more or less dense traffic to accurately detect four wheels, two wheels and pedestrians. The results are shown with coloured boxes with its corresponding identification for each detected object.

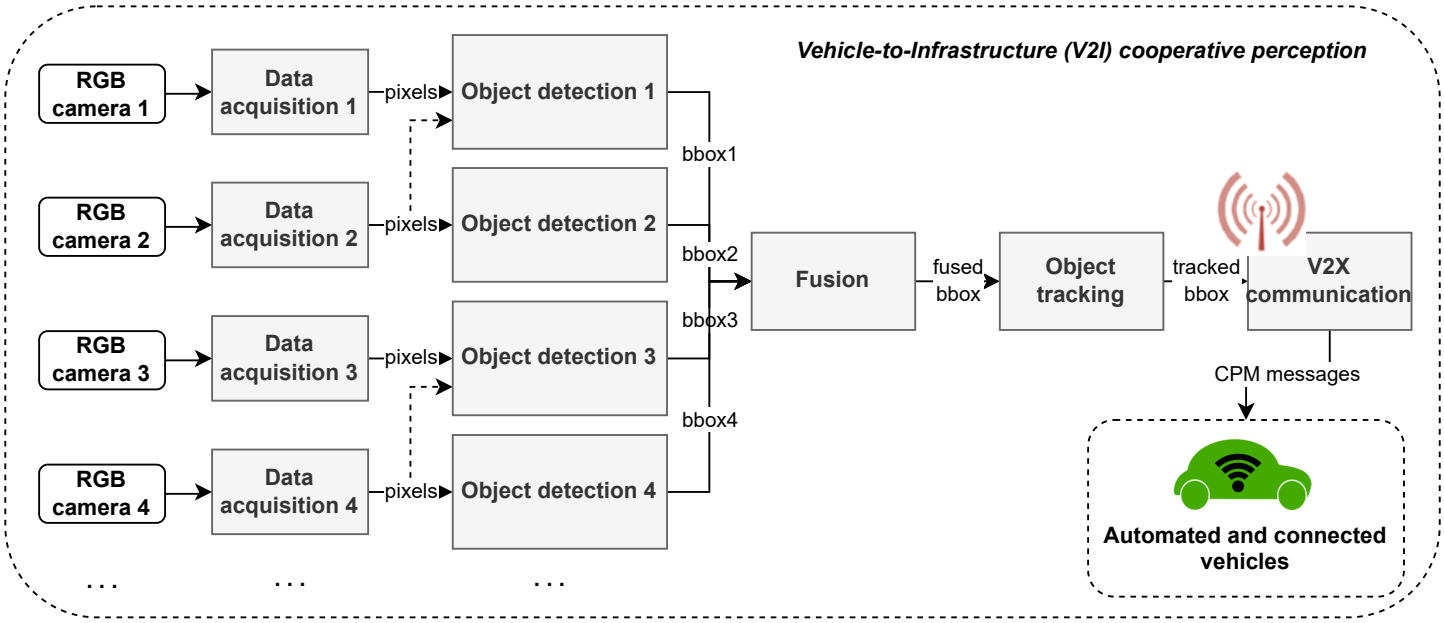


Figure 2: Proposed decentralized perception system with the process of multiple RGB cameras. It is designed to provide perception data to automated and connected vehicles from the infrastructure that has an extended point of view through the use of V2X communication. The system is composed of five key components: data acquisition, object detection, fusion, object tracking and V2X communication.

This article presents an implementation of a scalable decentralized perception architecture for V2X perception. It includes the installation of multiple cameras on the infrastructure to cover extended areas that are challenging for the ego-vehicle. The development of the perception modules from the data acquisition, the detection, the fusion, the tracking to the V2X messages generation is detailed. The proposed solution has demonstrated its performances in two different real-world scenarios, including two intersections with multiple viewpoints as shown in Figure 1.

This paper is organized as follows: In section 3, the decentralized solution and its perception are described. The qualitative performances are analyzed in section 4 with the implementation in real-world scenarios as well as the measured real-time performances.

Decentralized perception system

The decentralized perception system is based on image processing from multiple viewpoints using N cameras as shown in Figure 2. It is designed to provide extended perception data to the automated and connected vehicles through the use of Collective Perception Message (CPM) [7, 8]. The ROS2-based pipeline contains five main modules: data acquisition, object detection, fusion, object tracking and V2X communication.

Data acquisition module

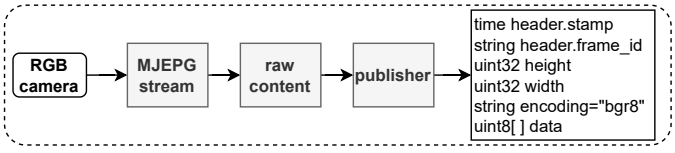


Figure 3: Data acquisition process to acquire the matrix data from the camera using the MJPEG stream.

The data acquisition module has been developed for accessing camera's stream and make modifications of the sensor settings [9]. First, the MJPEG stream is retrieved using the HTTP (Hypertext Transfer Protocol) or RTSP (Real Time Streaming Protocol) protocol with several parameters that allow the modifications of the received image, such as, the ping address, the required stream rate, the resolution and the video codec. Once the raw content of pixels acquired, the matrix data is established and published is a ROS2-topic image format as shown in Figure 3. It provides the timestamp of the created topic, the identification, the height and width of the image, the encoding and the matrix data in 8 bits. The matrix data which contains the pixels to process feeds the object detection module.

Object detection module

The accuracy, robustness and speed of the object detection processing is essential for the rest of the pipeline, which relies on the quality of the estimated bounding boxes. Object detection allows to detect, classify and estimate the position of an object. In the context of this work, only 2D bounding boxes are estimated. There are two types of paradigm, one-stage and two-stage detectors. One-stage methods, such as SSD (Single-Shot Detector) [10] and YOLO (You Only Look Once) [11] use convolutional neural networks (CNN) for bounding boxes predictions and the corresponding classification simultaneously. These methods have been developed for real-time performances but the accuracy of detection is deteriorated for smaller objects. Recent works focus on the trade-off between speed and accuracy but do not focus on the detection performance only [12]. On the other hand, two stage detector is mainly focused on the accuracy of the detected object rather than the computational complexity. For example, Fast-RCNN [13] and Faster-RCNN [14] employ a region proposal network to focus the classification and bounding box detection on specific spatial location. This solution provides high performances in terms of accuracy and high computational complexity that prevent real-time performance compared to a one-stage detector.

Object detection is treated as a regression problem by separating the classification and localization of the bounding box on the image. The YOLOv4 model is used and contains three main parts [15]: the backbone, the neck and the head. In YOLO methods, the input image is divided into grid cells and the backbone extracts the features of the input image from convolution layers. The EfficientNet model is used as feature extractor [16]. The latter uniformly scales network width, depth, and resolution with fixed coefficients from a compound scaling method. It allows to adapt the computational cost of the network based on the resources constraints. The neck improves the representation of

the extracted features with a multi-scale processing in a top-down path. It is used to increase the accuracy of object localization in the image, to detect small objects at different scale using a feature pyramid network [17]. The prediction is performed at each scale of features to obtain the classification and the position of object bounding boxes. The head is the last part of the YOLO model which is responsible for the final detection. It is an anchor-based processing that uses a non maxima suppression to select the best bounding box among many predictions for one object. The classification is separated from the position of the bounding box and introduces independent logistic classifiers to reduce the computation complexity by avoiding the softmax function. The main contributions of the fourth version of YOLO is the introduction of Bag-of-Freebies, which improves the detection by using data augmentation and Bag-of-Specials, which includes the study of various activation functions and different loss algorithms for bounding box regression.

Training data are at the core of expected performances for object detection. The developed system is decentralized using cameras mounted in the infrastructure that provide top views of the scene. The detection model is trained using datasets that have been recorded with top view data such as, the A9Dataset [18], Multi-View Traffic Intersection Dataset (MTID) [19], VisDrone [20] and MOT20 challenge [21] for pedestrians detection. The decentralized solution is designed to detect three classes: four wheels, two wheels and pedestrians. Table 1 details the class distribution in the training set by combining data from the cited datasets. It contains 56% of four wheels, 14% of two wheels and 30% of pedestrians.

class prediction	4 wheels	2 wheels	pedestrians
class distribution (#)	79324	20362	42336

Table 1: Class distribution for the training of the object detection algorithm.

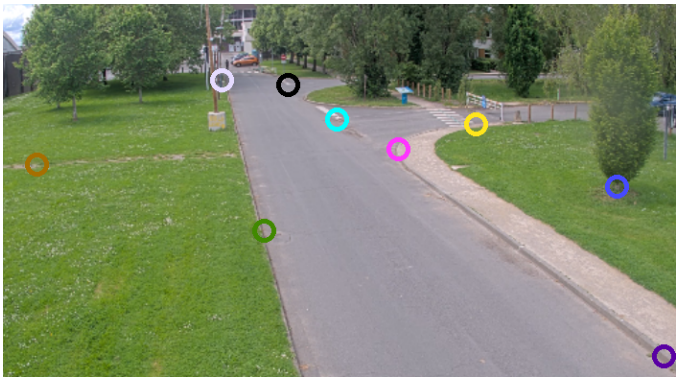
The object detection module provides bounding boxes based on the class distribution in the shape of a ROS2 vision message topic format. It contains the timestamp, the confidence score, the position and size of the bounding box. The module is entirely computed on GPU and the number of process depends on the number of RGB cameras. If there is not enough computational capacity available, one detection module can process two cameras as seen with the dotted arrows in Figure 2.

Fusion module

Each estimated bounding box is in the corresponding camera's image frame. In order to take advantage of each object detection results, the fusion module gathers all predictions from multiple viewpoints in the same frame. It is designed as a two-stage process: First, the projection of each bounding box to the new frame (Image to BEV frame) and Second, the multiple objects fusion using a fusion method based on the distance of the boxes centers.

Image to BEV frame

The projection of each bounding box resulting from the object detection of multiple viewpoints to new main frame is performed using the homography matrix. The latter provides the transformation between two planes, one from the camera and one from a BEV (Bird's Eye View) map. The BEV map is an orthonormal satellite image taken from the IGN BD ORTHO® as shown in Figure 1. The projection matrix is computed using nine points. Figure 4 (a) shows an example of the manually selected points in the camera frame and in (b) the corresponding ones manually selected in the BEV frame. In order to verify the accuracy of each corresponding point, the root mean square error is computed based on the estimated homography matrix. The error for each points is also verified to identify the inaccurate corresponding points. Then, the appropriate corrections are made to decrease the pixel error as low as possible.



(a)

(b)

Figure 4: Example to compute the transformation matrix from one plane as the image frame (a) to another plane as the BEV frame (b).

Multiple objects fusion

Once the projection done, all predicted bounding boxes from multiple object detectors are in the same frame. Multiple techniques are used to ensemble boxes [22, 23]. The Non-maximum Suppression (NMS) is the most widespread method to fuse redundant boxes. The NMS approach takes into account the Intersection over Union (IoU) between the highest confident predicted bounding boxes. If the IoU is beyond a given threshold, the predictions are identified as one object and the most confident one become the final detection. The NMS is widely used for object detection because of the provided performances and the simple use and implementation of the algorithm for real-time performances. However, this fusion method leads to an aggressive suppression of predictions that can be considered as true positive detections. The Soft-NMS [24] method proposes to replace the suppression of low confident detections in NMS by reducing the detection scores according to a decay function. This approach allows to reduce the suppression of true positive detections but can be computational expensive with a large number of detections ($\mathcal{O}(n^2)$). Other fusion methods use weighted techniques, such as the Non-maximum weighted (NMW) [25] and the Weighted boxes fusion (WBF) [26]. While the NMW uses the IoU value to weight the fusion boxes, the WBF uses the confidence score of all boxes to compute the fusion as a weighted average. The latter provides better results than the NMS, Soft-NMS and NMW with the mean average precision metric but is worse than the NMS when a scenario provide a large amount of overlapping boxes with different confident scores.

The proposed system is designed to incorporate multiple sensors with overlapping point of view. As the projection by homography is not as accurate as the transformation matrix $[\mathbf{R}|t]$, the same detected object from multiple sensors may not be accurately placed in the same spot. This prevents the use of the IoU that is used in many fusion techniques that consider an overlap of bounding boxes for the same object. Therefore, an NMS-like is used by taking the distance between the center of

bounding boxes instead of the IoU. The fusion between two bounding boxes is performed using the mean position in the map. The following steps are followed to compute the fusion between multiple bounding boxes:

Algorithm 1 Fusion algorithm based on the distance between centers

input: list of boxes with element in [xmin, ymin, xmax, ymax]
list of class IDs corresponding to each boxes
list of confidence scores corresponding to each boxes
list of camera IDs corresponding to each boxes
param maximum distance between centers of boxes (in meters)
param conversion factor from meters to pixels
output: list of fused bounding boxes in [xmin, ymin, xmax, ymax]
list of fused class IDs
list of fused confidence scores
list of fused camera IDs
1/ convert the maximum distance in meters to pixels
2/ iterate through each box
3/ estimate the center of two bounding boxes
4/ compute the distance between centers
5/ fuse boxes if the distance is less than the maximum distance and they belong to the same class
6/ the fusion result is the mean position of the selected boxes

The following modules are based on the fused bounding boxes that are positioned in the BEV map.

Object tracking module

In order to track the fused objects over time, object tracking methods follow two main paradigms [27]: tracking-by-detection (TBD) and joint detection and tracking (JDT). Both paradigms take into account four process, the backbone for feature extraction, the object detection, the object tracking and the data association. The TBD aims to detect and track multiple objects as two independent tasks. Indeed, the tracking process takes into account the results of the detection [28, 3]. In [3], a feature extractor is employed to increase the robustness of the tracking during data association. Despite real-time processing, the TBD pipeline is not optimized due to the two-stage processing [27] and those methods are limited by the occurrence of many false positives [29]. On the other hand, the JDT paradigm simultaneously performs the detection and tracking in a single pass with a shared backbone [30]. However, the detection process does not take into account tracking cues that are the crucial for a robust and accurate track over multiple frames [27]. Furthermore, sharing one backbone implies that the training loss of the tracking and the detection must be compatible. The re-identification process of one tracked object when it is lost focuses on intra-class variance whereas the detection aims to increase inter-class difference and minimize intra-class variance [27, 31]. Despite being more accurate than TBD methods, the computational complexity of the JDT is more important which prevent real-time processing.

The proposed architecture implements the Simple online and real-time tracking (SORT) algorithm [28]. This method belongs the the TBD paradigm and has been selected due to its low complexity to implement and for real-time processing. SORT uses bounding box to identify and track multiple objects. The algorithm uses a Kalman filter [32] to predict the future state of an object and update it in terms of positions and speed from observations. The Hungarian algorithm [33] used in SORT is responsible of data association between consecutive images by taking into account the Mahalanobis distance. The latter measures the standard deviation of the detection with respect to the average localization of the tracked bounding box. The association algorithm solves the assignment problem by minimizing the overall cost of the incorrect association. This part has been improved by DeepSORT [3] which uses, in addition of the Mahalanobis distance, a feature extractor for a deep appearance descriptor of each detected object. This technique makes the tracking process more robust but increases the computational complexity.

The object tracking module provides the identification (ID), the position and the speed of each tracked object. In addition, the orientation θ is computed based on the v_x and v_y speed vectors respecting the telecommunication standards of the ETSI report for V2X communication [7, 8].

V2X communication module

<table>
<tr><td colspan="5">CPM</td></tr>
<tr><td rowspan="4">Header</td><td colspan="4">CPM content</td></tr>
<tr><td rowspan="3">generation delta time</td><td colspan="3">CPM param</td></tr>
<tr><td rowspan="2">management container</td><td colspan="2">perception data</td></tr>
<tr><td>container ID</td><td>container data</td></tr>
</table>

Figure 5: Collective perception message (CPM) structure used and defined by the ETSI telecommunication standards.

As shown in Figure 2, the decentralized V2X communication module is responsible for providing CPM messages to automated and connected vehicles. To do so, a json file is generated and encoded to be provided to the connected vehicles. This file contains the fields that need to be filled following the telecommunication standards.

Figure 5 shows the implemented CPM structure. It is composed of 1/ the header that describes the protocol version, the message and the station identifications. 2/ the generation delta time allows to represent the age of the CPM which corresponds to the difference between the encoded one in the received CPM and the local one. 3/ the CPM parameters contain the station type and the reference position of the reference point in the map in the management container. Parameters also define the perception data with the identification of the container and the perceived object in the data. The latter includes the number of perceived object in the encoded CPM and multiple information of each object that are taken from the tracking module, such as, the object ID, the distances, the speeds, the angle, the classification and the confidence score.

Those information are written in a json file, which is encoded and transmitted to the connected vehicles through a wireless network. In our experiments, the connectivity was made through the 5G network. Then, the decentralized perception data feeds the onboard perception that allows to overcome limitations, such as occlusions with an extended covered area by the infrastructure.

Performance analysis

This section details the qualitative results of the proposed solution acquired in real-world scenarios in addition of the measured real-time performances.

Implementation in real-world scenarios

The decentralized system has been implemented in two intersections as shown in Figure 1 (a) and (b) respectively. Figure 1 (a) illustrates the use of six cameras to cover the intersection of an L-shape around the parking lot. Figure 1 (b) illustrates the use of four cameras to cover an intersection with multiple roads. In both figures, the results of the object tracking is shown that relies on the fusion of the object detection in each frame. The orientation and the speed of each object is highlighted by the drawn arrow.

Figure 6 and 7 show the qualitative results of the object detection for each cameras installed in the first L-shape intersection and in the sec-

ond one respectively, synchronized with the BEV map results. In Figure 6, the first two images corresponds to the site 22, the next three images to the site 24 and the last one in the bottom right to the site 26. Qualitatively, every relevant objects in the scene is detected with a high confidence score in rainy conditions. In Figure 7, the site 43_0, 43_1, 43_2 and 43_3 corresponds to the top left, top right, bottom left and bottom right images respectively. This intersection is denser and more complex with lots of traffic and a large overlap between all installed cameras. The qualitative results show high performances in terms of detection in sunny conditions, even for small object in a dark area, as it can be seen with the detected pedestrian in the top right image.

The fusion of the detected bounding boxes highlighted in Figure 1 (a) and (b) mostly relies of the accuracy of the projection matrix and the maximum distance parameter used to fuse multiple boxes based on their estimated center. The latter has been set to 4 meters and 6 meters for the L-shape and the second intersection respectively. Indeed, a lower distance is favorable to avoid the duplication of pedestrians detection, a higher distance is favorable for the fusion of four wheels detection. Duplication of boxes in the fusion results have been noticed in the second intersection with a low distance parameter for fusion because of the large amount of overlap between the cameras and the complex shared traffic between road users. Finding the right trade-off is crucial at this stage to obtain the highest accuracy.

Real-time performances

	acquisition	detection	fusion	tracking	CPM
latency (ms) ↓	170	30	7	6	13

Table 2: Average latency in milliseconds (ms) measured for each part of the proposed solution, from data acquisition to the generation of CPMs.

Table 2 details the execution time of each part of the proposed pipeline, from data acquisition to the generation of the CPM message. The latency of 170ms for data acquisition corresponds to the time between the moment when the sensors captures the light and the moment when the image is received through an RJ45 connection with a resolution of 640×360. The used cameras are AXIS Q1798-LE and P1455-LE and have been installed based on the configuration simulated in the AXIS software with specific features, such as, height installation of 6 meters, horizontal field of view or tilt angle in degrees. This camera operates as a black box with internal processes through the image signal processor. It prevents data acquisition with very low latency. The object detection module is the only part to run on GPU using an NVIDIA RTX A5500. The measured latency from receiving the image to the bounding box output is 30ms. One object detection process runs on one available GPU. Multiple viewpoints require to process each images in real-time and the amount of compute is ideally proportional to the number of installed cameras. However, only three GPUs were available. Therefore, a buffer has been added that enables the detection module to process two consecutive images from two different cameras. This adds an average latency of 30ms per graphics card to process each captured frame relating to the same scene. The rest of the pipeline is processed on AMD EPYC 7763 CPUs. The average latency for object fusion has been measured with 7ms per fused object. The complexity is estimated to be $\mathcal{O}(n)$. The process of tracking with the SORT algorithm is very low with 6ms and the generation of CPM messages in a json format takes around 13ms. The mean execution time of the entire pipeline is around 226ms to 256ms, depending on the number of object to be fused, with a process comprising up to six cameras.

Figure 6: Object detection results for each installed camera in the L-shape intersection. The latter is composed of six sensors to cover the roads of the intersection.

Figure 7: Object detection results for each installed camera in the second intersection. The latter is composed of four sensors to cover the entire area.

Conclusions and perspectives

This technical paper presents the decentralized perception system designed to support the process of multiple installed cameras with different viewpoints for V2I cooperative perception. Each module of the perception pipeline is described and developed to be scalable. Methods from the literature has been identified and selected for each modules, apart from the fusion one that proposes a new object fusion approach that takes into account the distance metric instead of the IoU. The system has been installed with different number of sensors in two intersections and has been tested in real-world conditions with automated and connected vehicles to receive the decentralized perception data. The performance analysis shows high accuracy in terms of detection with a latency of around 226ms for the entire pipeline that is not optimized, mainly due to the sensors.

One major perspective to reduce the latency is to change the used sensor which alone causes 75% of execution time. Other perspectives are planned, such as: 1/ the use of a transformation matrix between the camera frame and the BEV one using a camera pinhole model instead of the homography matrix. This would improve the accuracy of the bounding box projection. 2/ With an accurate transformation from the camera frame to BEV, the fusion method could take into account the IoU value between the estimated objects, which is mostly used to provide the best accuracy as possible. 3/ The last perceptive aims to evaluate quantitatively the decentralized perception system which requires

the acquisition of a ground truth in the experimental intersections.

References

1. H. Zhang and K. Wang, "Advances and perspectives on applications of deep learning in visual object detection," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 43, pp. 1289–1305, 08 2017.

2. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

3. 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)*, pp. 3645–3649, 2017.

4. G. Cui, W. Zhang, Y. Xiao, L. Yao, and Z. Fang, "Cooperative perception technology of autonomous driving in the internet of vehicles environment: A review," *Sensors*, vol. 22, no. 15, 2022.

5. G. M. Hinz, M. Buechel, F. Diehl, G. Chen, A. Krämmer, J. Kuhn, V. Lakshminarasimhan, M. Schellmann, U. Baumgarten, and A. Knoll, "Designing a far-reaching view for highway traffic scenarios with 5G-based intelligent infrastructure," in *8. Tagung Fahrerassistenzsysteme*, (Munich, Germany), TÜV Süd, Nov. 2017.

6. A. Krämmer, C. Schöller, D. Gulati, V. Lakshminarasimhan, F. Kurz, D. Rosenbaum, C. Lenz, and A. Knoll, "Providentia - a large-scale sensor system for the assistance of autonomous vehicles and its evaluation," *Field Robotics*, vol. 2, pp. 1156–1176, 2019.

7. *Intelligent Transport Systems (ITS); Vehicular Communications; Analysis of the Collective Perception Service (CPS)*, 2019. Standard ETSI TR 103 562 V2.1.1, ETSI TC ITS.

8. *Intelligent Transport Systems (ITS); Vehicular Communications; Collective Perception Service*, 2023. Standard ETSI TR 103 324 V2.1.1, ETSI TC ITS.

9. Q. Picard, C. Iverach-Brereton, and M. Hosmar, "Ros driver - axis camera." `https://github.com/ros-drivers/axis_camera/tree/humble-devel` (visited: 2024-09-25).

10. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 21–37, Springer International Publishing, 2016.

11. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

12. J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

13. R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

14. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

15. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

16. M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, PMLR, 09–15 Jun 2019.

17. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

18. C. Creß, W. Zimmer, L. Strand, M. Fortkord, S. Dai, V. Lakshminarasimhan, and A. Knoll, "A9-dataset: Multi-sensor infrastructure-based dataset for mobility research," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 965–970, 2022.

19. M. Jensen, A. Møgelmose, and T. Moeslund, "Presenting the multi-view traffic intersection dataset (mtid): A detailed traffic-surveillance dataset," in *IEEE 23rd International Conference on Intelligent Transportation Systems 2020*, (United States), IEEE, 2020. 23rd IEEE International Conference on Intelligent Transportation Systems 2020 ; Conference date: 20-09-2020 Through 23-09-2020.

20. P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7380–7399, 2021.

21. P. Dendorfer, H. Rezatofighi, A. Milan, J. Q. Shi, D. Cremers, I. D. Reid, S. Roth, K. Schindler, and L. Leal-Taix'e, "Mot20: A benchmark for multi object tracking in crowded scenes," *ArXiv*, vol. abs/2003.09003, 2020.

22. E. Razinkov, I. Saveleva, and J. Matas, "Alfa: Agglomerative late fusion algorithm for object detection," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2594–2599, 2018.

23. H. Lee and H. Kwon, "Dbf: Dynamic belief fusion for combining multiple object detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1499–1514, 2021.

24. N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms — improving object detection with one line of code," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5562–5570, 2017.

25. H. Zhou, Z. Li, C. Ning, and J. Tang, "Cad: Scale invariant framework for real-time object detection," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 760–768, 2017.

26. R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104117, 2021.

27. J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12347–12356, 2021.

28. 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, 2016.

29. A. Belmouhcine, J. Simon, L. Courtrai, and S. Lefèvre, "Robust deep simple online real-time tracking," in *2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 138–144, 2021.

30. X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, (Berlin, Heidelberg), p. 474–490, Springer-Verlag, 2020.

31. D. Chen, S. Zhang, W. Ouyang, J. Yang, and Y. Tai, "Person search via a mask-guided two-stream cnn model," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 764–781, Springer International Publishing, 2018.

32. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

33. H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

Author Information

Quentin Picard (quentin.picard@vedecom.fr)
Malo Morice (malo.morice@vedecom.fr)
Maryem Fadili (maryem.fadili@vedecom.fr)
Steve Pechberti (steve.pechberti@vedecom.fr)

Acknowledgments

The authors acknowledge the infrastructure and the support of the SCARLET team working at the VEDECOM Institute. This work has been part of the European 5G Open Road project. Special thanks to colleagues at VEDECOM, Mohamed-Cherif Rahal, Laurent Fevrier, Benoît Lusetti, Jérémy Lefebvre, Angel Denis, Ghada Sayari for their support and their work in this project. The authors would also like to thank the Nokia Bell Labs Paris-Saclay for setting up the network architecture allowing real-world demonstrations, and in particular Jean-Luc Lafragette and Jean-Olivier Mescam.