A Joint Perception Scheme For Connected Vehicles

Ahmed N. Ahmed*, Ian Ravijts†, Jens de Hoog*, Ali Anwar*, Siegfried Mercelis*, Peter Hellinckx*
*Department of Electronics and ICT Engineering, IDLab - University of Antwerp - imec, Antwerp, Belgium
†Department of Applied Informatics, AP University of Applied Sciences, Antwerp, Belgium
ahmed.ahmed@uantwerpen.be

***Abstract*—Currently visual sensing systems, used in autonomous vehicle's research, typically perceive the surrounding environment up to 250m ahead of the vehicle. However, the detection reliability drops when the object's position is more than 50m, due to objects being sparse or unclear for the detection model to make a confident detection. Cooperative perception extends the visual horizon of the onboard sensing system, by expanding the sensing range which improves the detection precision. This paper explores early distributed visual data fusion by creating a multi-vehicle dataset using the Carla simulator to create a shared driving scenario, equipping every spawned vehicle with LiDAR, GNSS, and IMU sensors to emulate a real-driving scenario. Furthermore, we investigate the usage of ZeroMQ-based communication system to distribute visual and meta- data across relevant neighboring vehicles. Since the proposed method distributes raw LiDAR data, we utilize point cloud compression to reduce the size of the published data between relevant connected vehicles to satisfy communication bandwidth requirements. Subsequently, we transform and fuse the received data, and apply a deep learning object detection model to detect the objects in the scene. Our experiments prove that our proposed framework improves the detection average precision while satisfying bandwidth requirements.**

***Index Terms*—point clouds, shared situational awareness, connected vehicles, distributed data fusion**

I. Introduction

Situational awareness of complex driving environments is crucial for the safety of autonomous navigation. With recent developments in computer vision using deep learning, the robustness of single-vehicle perception systems has demonstrated significant improvements for tasks such as object detection [1], [2], [3]. Despite these advances, there are still open challenges; for example, limited field of view due to single perceptive view leading to failure in detecting objects that are heavily occluded or further away. This, in turn, can result in catastrophic consequences affecting the reliability and safety of autonomous navigation. Additionally, when a sensor fails due to disconnection or fault in its internal hardware, the agent will not be able to have a sense of its surrounding; therefore, a solution to this would be connecting the agents. Shared situational awareness would provide super-human capabilities to autonomous navigation if vehicles are equipped to transmit/receive information from neighboring actors and fuse/utilize the received information. Recent studies [4]–[6], presents the benefits of visual data sharing to avoid critical situations to enhance traffic safety and reduce fatalities, as it enhances the precision and confidence of the detected objects.

The performance of cooperative driving is dependent on the type of data being shared, the network bandwidth, and the data fusion methodology; all of them pose challenges that need to be tackled. On the one hand, early visual data fusion algorithms fuse raw data is more favourable [6], since the data contains more contextual information when compared with late fusion methodologies that fuses output predictions of other neighbouring vehicles. In late fusion, the prediction relies on single vehicular sensors and detectors, this will only work when both vehicles share a reference object in their detection; additionally it does not solve the issue of previously undetected objects which will remain undetected even after the fusion [6]. On the other hand, the bandwidth and latency requirements of vehicular networks must satisfy data transmission for cooperative perception [7] to dynamically incorporate changes in the driving scene; however, sharing all collected data and the raw visual data leads to high latency, due to their size.

To train detection models and evaluate the detection precision for autonomous vehicles, available datasets such as KITTI [8] and NuScenes [9] are well-known vision benchmark datasets. However, as KITTI's and NuScenes's data were gathered from a single vehicle. As a result, those datasets are only suitable for specific test scenarios such as ego vehicle object detection, making them unrealistic to be utilized in a distributed perception framework. Likewise, it is logistically expensive and time-consuming to deploy multiple cars with visual sensors to record a large-scale dataset suitable for benchmarking for cooperative perception algorithms. Taking into account the aforementioned problems and challenges, in this work, we aim to:

- Investigate the performance using different quantization parameters to compress point clouds.
- Share relevant data using ZeroMQ.
- Transform and fuse the raw point clouds, then perform object detection.
- Create a large-scale multi-vehicle dataset to evaluate the proposed method.

II. Methodology

A. Point clouds compression

To satisfy the latency and network's bandwidth requirements we utilized Draco compression algorithm [10] to reduce the size of the point cloud before publishing it. Draco compression framework is often used as benchmarking algorithm [11], due to its low computational complexity, resulting in a speedy compression time. The work presented in [12] shows that the compressed point cloud using Draco retains reasonable visual

https://doi.org/10.1109/SENSORS52175.2022.9967271

quality when carrying out increasingly lossy compression.

B. Data sharing

The visual and location data retrieved from the vehicle's sensors is to be shared to other relevant vehicles for shared situational awareness; this work utilizes ZeroMQ for data sharing. The data-sharing part is divided into two parts; 1)REQ-ROUTER and 2)publish-subscribe, as shown in Fig. 1. The centralized server works with a REQ-ROUTER pattern, where each vehicle sends its GNSS, IMU data, IP address, publish port, and topics to the server as a request, then server stores this data, and creates a unique ID for each vehicle. Afterward, the service containing the relevancy metrics is initialized to compute the spatial relevancy of the requesting vehicle to the others. To achieve this, we created a service to compute the relevancy of the vehicles to each other. The relevancy metric is based on the distance between vehicles and the heading, as presented in Fig. 2. Using the the GPS coordinates of all vehicles, the relevancy service calculates which other vehicles lie within a certain radius of the requesting vehicle and whether the two vehicles share an intersecting heading as illustrated in Fig. 2. Subsequently, the server replies to the requesting vehicle with the relevant client ID(s). The second part utilizes the publish-subscribe pattern, which enables the vehicles to subscribe to other relevant vehicles (using the received client ID) to subscribe to their data as soon as it is available, as shown in Fig .1. After subscribing and receiving the data from the relevant vehicle(s), the data is then fused with the ego vehicle's data.

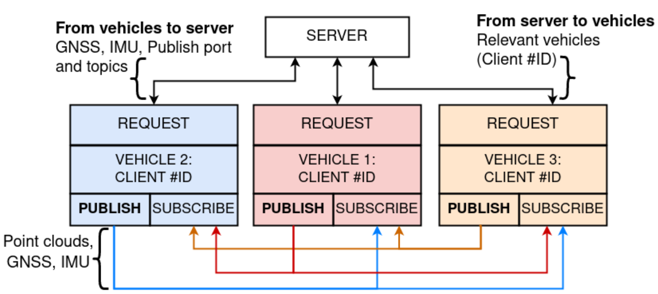

Fig. 1: Data sharing scheme, consisting of the REQ-ROUTER and publish subscribe patterns.

C. Fusion and object detection

The point clouds received from other neighbouring vehicles should be reconstructed and transformed to the ego vehicles pose, as it was captured from pose in the environment. We accomplish this by employing point-to-plane ICP (Iterative Closest Point) registration [13] to find the optimal transformation matrix between the ego and sender's point clouds, which is then used to align the both point clouds. Consequently, the aligned point clouds are concatenated and fed to a pre-trained 3D object detection model to perform 3D object detection. In this work, we use PIXOR [2] detector since it is a single-stage, proposal-free, dense 3D object detector. PIXOR comprises two networks: 1)Backbone and 2) Header. The backbone is composed of convolutional layers to extract an over-complete representation of the input feature and pooling layers, and pooling layers to down-sample the feature map size to save computation and help create a more robust representation.

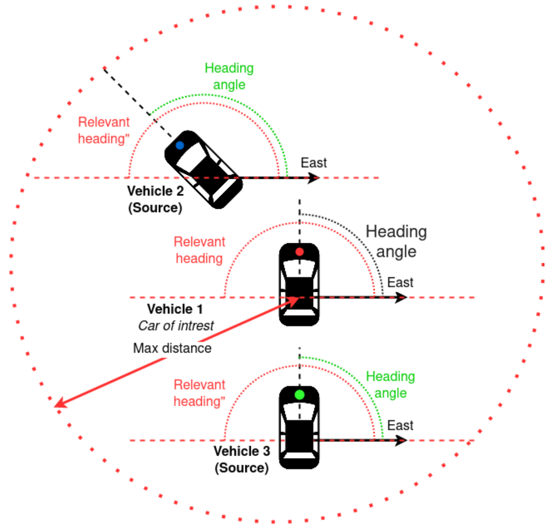

Fig. 2: Relevant metric computation based on the distance and between the vehicle of interest and other neighboring vehicles.

III. Results & Discussion

To create distributed perception dataset, we utilize Carla simulator [14]. We employed the multi-agent functionality of Carla to spawn multiple vehicles, where each vehicle is equipped with LiDAR to perceive the environment, GPS, and IMU to retrieve the pose as well as the location of the vehicle. We propose to use LiDAR sensor as point clouds have spatial dimensionality over 2D images, and its versatility in the fusion process as the point cloud data is composed of points rather than pixels. Additionally, image fusion requires a clear zone of overlap, which is unnecessary for point cloud, making it much more robust when fusing data captured from different locations. The built-in LiDAR sensor in Carla is based on ray casting, and we defined the LiDARs attributes to match with the technical specification as Velodyne HDL-64E, generating 250k points/sec, with a horizontal field of view of 360°and frequency of 10Hz. The GPS and IMU used are built-in objects in Carla, where we have utilized the default setting attributes defined by Carla. The ground truth bounding boxes and class labels of the neighboring actors are retrieved and saved at every LiDAR sweep.

We fuse the data broadcasted from vehicles within a radius of 130m based on existing communication protocols [7] and heading intersection of 70° from the ego-vehicle, and evaluate detection. In our distributed perception dataset, we include completely occluded objects, making the task more challenging and realistic. For object detection, we compute Average Precision (AP) at Intersection-over-Union (IoU) threshold of both 0.5 and 0.7. We evaluate the single-vehicle setting without fusion, early fusion at no quantization, and bit quantization parameters of 25, 20, 15. The detection visual results in the five aforementioned setting scenarios are shown in Fig. 3, and

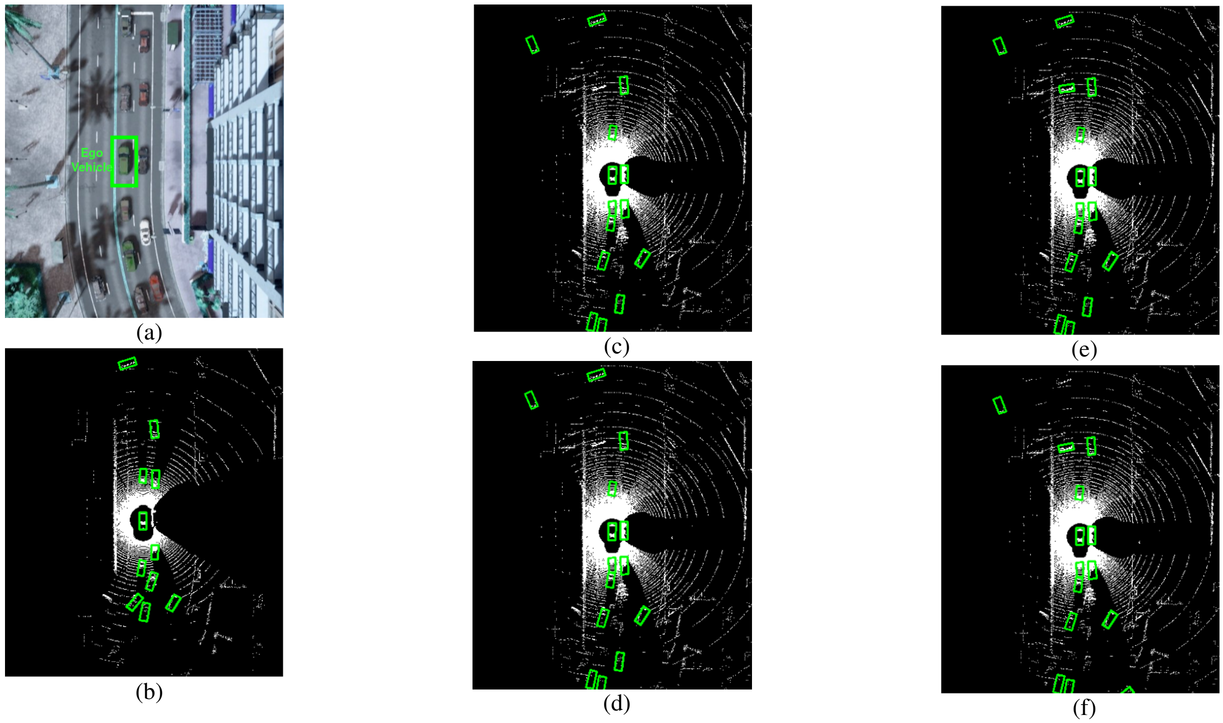

Fig. 3: (a) Represents BEV image of the driving scene. (b) Represents the BEV without fusion, (c) represents the BEV without quantization and (d), (e), (f) are object detection at quantization parameters of 25, 20, 15 respectively.

the numerical assessment is presented in Table. I.

In our experiments, we assumed that the vehicle indicated by green is the ego vehicle receiving data from other relevant vehicles. From the visual comparison shown in Fig .3, it is undeniably clear that with the fusion settings, more objects were detected that were initially occluded using the ego LiDAR only. Regarding the fusion, overall, lower quantization parameter yields in smaller point cloud size to be shared. Setting the quantization parameter to 20 bits seems to be the best middle ground for lossy compression, as it still has a decent output quality while maintaining a compression ratio of 4.2 times. Quantizing to 15 bits, as shown in Fig. 3f, results in increasing the compression ratio; however, this reduces the quality of the point cloud, causing failures in recognizing some objects. Whereas, without quantization, it results in the best AP values; however, it shares the largest point cloud size.

TABLE I: Assessment of point cloud size and object detection's AP after applying different quantization parameters.

Fusion	**Quantization parameter**	**Data shared size per vehicle (Mbps)**	**Average precision**		**Figure**
			IOU 0.5	**IOU 0.7**	
×	No Quantization	4.2	0.75	0.82	3b
✓	No Quantization	4.2	0.84	0.85	3c
✓	25	2.1	0.90	0.93	3d
✓	20	1.4	0.89	0.92	3e
✓	15	0.7	0.77	0.89	3f

IV. Conclusion & Future Work

In this paper, we have created a shared situational awareness framework that is composed of: distributed perception dataset, point cloud compression to reduce the size to satisfy the networking bandwidth, followed by point cloud transformation and fusion at the receiving agent. Our proposed shared situational awareness methodology accomplished prominent results while satisfying the IEEE 802.11p dedicated short-range communication (DSRC) requirements [7].

In future, we plan to extend our work to include an in-depth analysis of the point cloud quality after compression using PC-MSDM [15], to select the best quantization parameter. We will compute the transmission delay based on DSRC V2V communication protocol IEEE 802.11p [7]. Furthermore, we will perform real-life using a 5G stand-alone network.

V. Acknowledgement

This work was supported by the Research Foundation Flanders (FWO) under Grant Number 1S90022N. This work will be validated using 5G network within the Horizon 2020 5G-Blueprint project, funded by the European Commission under agreement No. 952189.

REFERENCES

[1] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[2] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[3] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[4] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song 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*, pages 88–100, 2019.

[5] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision*, pages 605–621. Springer, 2020.

[6] Qi Chen, Sihai Tang, Qing Yang, and Song Fu. Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 514–524. IEEE, 2019.

[7] John B Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[10] Google. Draco: 3D Data Compression, 2018.

[11] Xuebin Sun, Sukai Wang, Miaohui Wang, Zheng Wang, and Ming Liu. A novel coding architecture for lidar point cloud sequence. *IEEE Robotics and Automation Letters*, 5(4):5637–5644, 2020.

[12] Jens de Hoog, Ahmed N Ahmed, Ali Anwar, Steven Latré, and Peter Hellinckx. Quality-aware compression of point clouds with google draco. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 227–236. Springer, 2021.

[13] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.

[14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[15] Gabriel Meynet, Julie Digne, and Guillaume Lavoué. Pc-msdm: A quality metric for 3d point clouds. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3. IEEE, 2019.