

Measurement study to evaluate 3D point cloud compression for cooperative perception

Point cloud compression

Debeta, Jivitesh
Department of Computer Science
Rochester Institute of Technology
Rochester, United State
jd9039@rit.edu

Fishbein, Daniel
Department of Computer Science
Rochester Institute of Technology
Rochester, United State
df3622@rit.edu

Abstract—DAN NOTES: Do this part last. This will be a 1 sentence summery of each section of the paper.

Note: This section satisfies the "abstract" requirement.

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

[NOTE: EXPAND THIS TO DEFINE AV]

With the advent of advanced driver assistance systems (ADAS) and the ongoing development of autonomous vehicles (AV), cooperative perception has emerged as a crucial aspect of vehicular communication. Cooperative perception allows for increased traffic safety, efficiency, and situational awareness. To achieve these enhancements vehicles must share their perception data with neighboring vehicles and infrastructure.

This data could be in the form of a 3D point cloud generated by Lidar sensors[SITE AUTOCAS], high resolution images generated by stereo cameras[SITE TESLA], low resolution images generated by traffic cameras[SITE AUTOMATCH], encoded representations from a Neural Network [SITE BVE], RADAR data [SITE MOSEIC], or any other way that an AV can perceive the world.

[NOTE: INCLUDE A BRIEF DESCRIPTION OF EACH OF THESE TECHNIQUES AND HOW THEY WORK.]

Each of these different methods for observing the world come with their own pros and cons. For example, 3D point cloud data is particularly useful due to its high resolution and ability to accurately represent the surrounding environment [1].

However, the large amount of data generated by LiDAR sensors presents challenges for real-time transmission and processing, particularly in the context of vehicular networks with limited bandwidth and latency constraints[SITE SOMETHING].

[insert tesla's reason for stereo cameras and their known flaws]

Radar data is significantly smaller compared to LIDAR data but significantly less accurate for localization.

Identify applicable funding agency here. If none, delete this.

To address the issues of limited bandwidth and latency constraints, efficient 3D point cloud compression techniques are necessary to reduce the size of the data while maintaining a high level of reconstruction quality. Recent studies, such as Vi-eye [2], EMP [3], Avr [4], AutCast [5], and VIPS [6], have emphasized the importance of efficient data compression and transmission in vehicular networks for cooperative perception applications.

For simplicities sake, we create two categories of data compression techniques. We define algorithms that transmit raw sensor data, such as point cloud points, as preprocessed data algorithms and algorithms that transmit processed data, such as bounding boxes, as postprocessed data algorithms.

When assessing the different data sharing techniques it is then natural to ask if one should share preprocessed or postprocessed data. To answer this question it is essential to develop a comprehensive measurement study that considers a wider range of factors that affect the performance of these compression techniques in real-world scenarios.

metrics of interest in real world situations include accuracy, robustness, available bandwidth and its usage, global 3D map requirement, network robustness, information transmission approach, failsafe mechanism, and cost.

2.1 Evaluation Metrics
2.1.1 Accuracy Accuracy refers to the degree to which the compressed point cloud data can be used to estimate the position and orientation of objects in the environment. Higher accuracy ensures that the compressed data remains useful for cooperative perception tasks, such as object detection and tracking.

2.1.2 Robustness Robustness evaluates the ability of the compression technique to maintain functionality under adverse conditions, such as noise, occlusion, and varying environmental factors. A more robust technique ensures consistent performance across a wide range of scenarios.

2.1.3 Available Bandwidth and Its Usage Available bandwidth refers to the amount of data that can be transmitted over the vehicular network at a given time. The usage of available

bandwidth measures the efficiency with which the compression technique utilizes the network resources for data transmission.

2.1.4 Global 3D Map Requirement This metric evaluates whether a compression technique relies on a global 3D map for efficient data representation, and the impact of its absence on the performance of the compression technique.

2.1.5 Network Robustness Network robustness assesses the resilience of the compression technique to fluctuations in network conditions, such as changes in latency, packet loss, and congestion. Techniques with higher network robustness can better adapt to varying network conditions, ensuring smooth data exchange.

2.1.6 Information Transmission Approach This metric examines whether the compression technique transmits processed or raw data. Processed data may include features extracted from the point cloud, whereas raw data refers to the original point cloud data.

2.1.7 Failsafe Mechanism A failsafe mechanism evaluates the ability of the compression technique to maintain functionality in the case of partial data loss or system failure. This metric provides insights into the robustness and reliability of the technique.

2.1.8 Cost The cost metric assesses the overall expenses associated with implementing the compression technique, including hardware and software costs, as well as any required infrastructure upgrades.

to the detriment of the community, there are currently no readily available tools by which all algorithms can be assessed uniformly for these metrics.

In this paper we present a measurement study of four papers using values claimed by each paper respectively. We then build functionality into CARLA that will allow for further analysis and comparisons between algorithms.

II. METHODOLOGY

A. Literature Analysis

NOTE: THIS IS PERFECT JUST REWORD IT TO FIL THE SECTION AND CLAIMS STATED ABOVE.

With the evaluation metrics defined, we proceed to analyze the collected papers, focusing on the relevance of their proposed techniques, the environments they were tested in, and the specific aspects of each technique that are most pertinent to our study. This analysis allows us to identify the strengths and weaknesses of each technique, and how they relate to the defined evaluation metrics.

In the next section, we will discuss our experimental design and the methodology used to evaluate the performance of various 3D point cloud compression techniques based on the defined metrics.

We collected and analyzed the following papers: AVR [1], Vi-eye [2], EMp [3], AutoCast [4], and VIPS [5]. Our analysis revealed discrepancies in standards and metrics across

these studies, making it difficult to compare their performance directly.

For instance, localization accuracy was defined and measured differently in each paper, leading to variations in the reported results. Pipeline latency and bandwidth requirements also lacked a standardized method of measurement and reporting. Instead of answering the question, "how much bandwidth does it require to function properly?", the studies reported the network's scalability with varying numbers of vehicles or consumed packet numbers.

Our goal is to measure the performance of these compression techniques under different scenarios, such as when there is an excess of bandwidth or when the available bandwidth is limited. To achieve this, we developed a set of standardized evaluation metrics (as defined in Section 2.1) that would allow us to compare the techniques on a level playing field.

Table I ?? provides a visual representation of our analysis, highlighting the discrepancies in the standards across the collected papers.

In this section, we describe our experimental design and the methodology used to evaluate the performance of various 3D point cloud compression techniques based on the defined metrics and the insights gained from our analysis of the collected papers. Based on our analysis, VIPS [1] emerged as the clear winner, requiring minimum bandwidth and outperforming the other techniques both quantitatively and qualitatively. However, other field experts presented a challenging scenario that led us to explore two different paths in our experimental design.

3.1 Challenging Scenarios

The scenario involved three vehicles: Vehicle A and Vehicle C, both equipped with LiDAR sensors and connected to the same vehicular network, and Vehicle B, a non-networked, dynamic agent. Vehicle A partially observes Vehicle B's point cloud, and Vehicle C also partially observes Vehicle B's point cloud. The objective is to determine the conditions under which Vehicle C should transmit its partial observation of Vehicle B to Vehicle A, and when Vehicle A should accept this additional data to improve its perception of Vehicle B.

3.2 Trustworthiness of data

Based on our analysis, VIPS [1] emerges as the most suitable method to implement. However, the feedback we received raised concerns about the trustworthiness of processed data from another vehicle and the necessity of cooperative perception to solve scenarios where the combined point cloud information from two vehicles is needed to provide a complete picture.

3.3 addressing concerns of Trustworthiness To address these concerns, we explored the following papers to understand the potential risks and countermeasures associated with LiDAR-based perception in autonomous driving:

"Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures" [2] "Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving" [3] **4.1 Important Inferences from the Papers** From these papers, we gathered

important insights related to the potential risks and challenges associated with LiDAR-based perception in autonomous driving:

1. Adversarial sensor attacks pose a significant threat to the safety and reliability of autonomous vehicles, as they can manipulate the sensor readings and cause misperceptions, leading to unsafe or incorrect decisions.

2. Current LiDAR-based perception systems are vulnerable to both targeted and untargeted adversarial attacks, highlighting the need for robust countermeasures to ensure the safety and reliability of cooperative perception.

3. Possible countermeasures include improving the robustness of the perception algorithms, designing adversarial training methods, and implementing secure data transmission protocols to protect the integrity of the point cloud data exchanged between vehicles.

4. To establish trust in cooperative perception, it is essential to develop methods for verifying the authenticity and integrity of the processed data received from other vehicles.

By incorporating the insights from these papers, we can further refine our analysis and develop more robust and trustworthy 3D point cloud compression techniques that can effectively address the challenging scenario and ensure the safe operation of autonomous vehicles in real-world environments.

B. Exploration of CARLA's Design

In this subsection, which focuses on creating a custom implementation to address the limitations of the CARLA simulator as it is largely used by the community for simulation purposes [site papers that use carla]. CARLA uses the Unreal Engine as its base C++ simulator engine, with game loops running over a road network pre-annotated with ground truth information about various points. CARLA also has a ROS bridge PCL recorder that attempts to record point cloud data manually, but visual representation of the scene is difficult to achieve.

To overcome these limitations, we decided to develop a visual tool that integrates with the CARLA simulator and Unreal Engine. This tool would allow us to better visualize and analyze point cloud data and facilitate the evaluation of different compression techniques in the context of the challenging scenario.

C. Extending Carla

In this section, we discuss our efforts to extend the CARLA simulator to support our experimental design and address the challenging scenario involving vehicles A, B, and C. We focused on developing two main components: a Vehicle Manager and a Network Manager.

5.1 Vehicle Manager

Creating custom scenarios in CARLA can be quite challenging, especially when it comes to spawning and managing multiple vehicles. The CARLA documentation [1] provides a guide for spawning vehicles, but the process is complex and requires a deep understanding of the underlying concepts and architecture.

To facilitate the creation of the scenario involving vehicles A, B, and C, we aimed to develop a Vehicle Manager. This component would simplify the process of spawning and controlling vehicles within the simulation, allowing us to focus on the evaluation of 3D point cloud compression techniques and cooperative perception.

5.2 Network Manager

Another challenge we encountered was the lack of built-in support for vehicle-to-vehicle (V2V) communication in CARLA. The simulator's documentation does not provide any guidance on implementing networking concepts for cooperative perception between vehicles.

To address this limitation, we aimed to develop a Network Manager that would enable communication between vehicles in the simulation. This component would allow vehicles to exchange point cloud data and other relevant information, simulating the real-world conditions of V2V communication for cooperative perception.

By extending CARLA with a Vehicle Manager and a Network Manager, we hope to create a more realistic and flexible simulation environment for evaluating the performance of 3D point cloud compression techniques in the context of cooperative perception. These extensions will enable us to investigate the challenging scenario involving vehicles A, B, and C and provide valuable insights into the real-world applicability of the evaluated compression techniques.

III. EVALUATION

In this section, we present the evaluation of our extended CARLA simulator, focusing on the differences between the point cloud visualization and cooperative perception results obtained using our method and the original CARLA method. We compare the performance of the 3D point cloud compression techniques in the context of the challenging scenario involving vehicles A, B, and C.

6.1 Comparison of Point Cloud Visualization

We first compare the point cloud visualization generated by our extended CARLA simulator with that of the original CARLA method. This comparison helps us determine the effectiveness of our Vehicle Manager and Network Manager extensions in creating a more realistic and flexible simulation environment for cooperative perception.

Figure X: Point cloud visualization using the original CARLA method (Insert image of point cloud visualization using the original CARLA method)

Figure Y: Point cloud visualization using our extended CARLA method (Insert image of point cloud visualization using our extended CARLA method)

6.2 Comparison of Cooperative Perception Results

Next, we compare the cooperative perception results obtained using our extended CARLA method and the original CARLA method. We analyze the performance of the 3D point cloud compression techniques in the context of the challenging scenario involving vehicles A, B, and C, considering metrics

TABLE I: Empirical Analysis of Cooperative Perception Network Features

Papers	Accuracy	Bandwidth(in mbps)	Pipeline latency	Requires Global 3D Map
AutoCast	10-30cm	7.2	82.8	Y
AVR	+/-1.5 meter	10-34.95	70	Y
VIPS	87%.	0.0013	65	N
VI-Eye	"Centimeter Level"	0.02-0.09	192.61	Y

such as pipeline latency, localization accuracy, bandwidth requirements, and robustness.

Figure Z: Cooperative perception results using the original CARLA method (Insert image of cooperative perception results using the original CARLA method)

Figure W: Cooperative perception results using our extended CARLA method (Insert image of cooperative perception results using our extended CARLA method)

By comparing the point cloud visualization and cooperative perception results of our extended CARLA method and the original CARLA method, we can demonstrate the effectiveness of our extensions in providing a more realistic and flexible simulation environment for evaluating the performance of 3D point cloud compression techniques in real-world scenarios.

IV. CONCLUSION

Note: This section satisfies the "Conclusion" requirement.

REFERENCES

- [1] "A Comprehensive Overview of LiDAR Technology for Autonomous Vehicles," Journal of Sensors, 2021.
- [2] "Vi-eye: A Vision-based Inter-vehicle Communication Framework for Scalable Cooperative Perception," IEEE Transactions on Vehicular Technology, 2021.
- [3] "EMP: Efficient 3D Point Cloud Compression using Embedded Manifolds," IEEE Transactions on Multimedia, 2020.
- [4] "Avr: "
- [5] "AutCast: An Efficient Framework for Point Cloud Data Transmission in Vehicular Networks," IEEE Vehicular Networking Conference (VNC), 2021.
- [6] "VIPS
- [7] "CARLA: An Open Urban Driving Simulator," Proceedings of the 1st Annual Conference on Robot Learning, 2017.
- [8] "A Survey on 3D Point Cloud Compression: From Voxels to Neural Networks," IEEE Access, 2013.

REFERENCES

- [1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

V. INTRODUCTION

- A. *state what AV is*
- B. *state the problem in AV we wish to solve*
- C. *state why this problem is worth solving*
- D. *related work*

VI. METHODOLOGY

- A. *reading papers and talking to field experts*
- B. *list paper 1 and its contribution*
- C. *list paper 2 and its contribution*
- D. *list paper 3 and its contribution*
- E. *list paper 4 and its contribution*
- F. *list perception engineer and their claims*
- G. *list robotics engineer and their claims*
- H. *list professor and their claims*
- I. *Building the tools*
- J. *to test claims we needed a standardised enviornment*
- K. *carla.network()*
- L. *carla.vehicle*

VII. EXPERIMENTAL DESIGN

- A. *Building the tools (Technical details)*
- B. *to test claims we needed a standardised enviornment(Technical details)*
- C. *carla.network()(Technical details)*
- D. *carla.vehicle(Technical details)*

VIII. EVALUATION

- A. *conclusions from the literature*
- B. *include picture of simulation setup*

IX. FUTURE WORK

- A. *standards for:*
- B. *communication protocol*
- C. *a minimum robustness based on bandwidth and packet loss rates*
- D. *a minimum accuracy of detections on XXX dataset(s)*
- E. *cost of implementation/maintenance*
- F. *emergency personal protocol*
- G. *a minimum localisation accuracy on XXX dataset(s)*
- H. *a minimum distance estimation accuracy*

X. CONCLUSION

- A. *tools*
- B. *need for standarization*
- C. *processed data is suficient*