Yassin Alkhalili, Thomas Gruczyk, Tobias Meuser, Antonio Fernández Anta, Ahmad Khalil, and Andreas Mauthe. *IEEE Multimedia Big Data (2022)*

Content-Aware Adaptive Point Cloud Delivery

Yassin Alkhalili*‖, Thomas Gruczyk†, Tobias Meuser*‡, Antonio Fernández Anta§, Ahmad Khalil*
and Andreas Mauthe‖
Affliation: *Multimedia Communications Lab, Technical University of Darmstadt, Germany
†Department of Computer Science, Technical University of Darmstadt, Germany
‡University of Oslo, Norway, §IMDEA Networks Institute, Madrid, Spain, and ‖University of Koblenz · Landau, Germany
Email: *[firstname.lastname]@kom.tu-darmstadt.de, †thomas.gruczyk@stud.tu-darmstadt.de,
§antonio.fernandez@imdea.org, ‖mauthe@uni-koblenz.de

***Abstract*—Point clouds are an important enabler for a wide range of applications in various domains, including autonomous vehicles and virtual reality applications. Hence, the practical applicability of point clouds is gaining increasing importance and presenting new challenges for communication systems where large amounts of data need to be shared with low latency. Point cloud content can be very large, especially when multiple objects are involved in the scene. Major challenges of point clouds delivery are related to streaming in bandwidth-constrained networks and to resource-constrained devices. In this work, we are exploiting object-related knowledge, i.e., content-driven metrics, to improve the adaptability and efficiency of point clouds transmission. This study proposes applying a 3D point cloud semantic segmentation deep neural network and using object-related knowledge to assess the importance of each object in the scene. Using this information, we can semantically adapt the bit rate and utilize the available bandwidth more efficiently. The experimental results conducted on a real-world dataset showed that we can significantly reduce the requirement for multiple object point cloud transmission with limited quality degradation compared to the baseline without modifications.**

***Index Terms*—Point Cloud, Semantic Segmentation, Autonomous Vehicles, Virtual Reality**

I. Introduction

Recent advancements in depth sensors technologies and their increasing affordability have led to a renewed interest in the point cloud data for autonomous vehicles or virtual and augmented reality applications. In autonomous vehicles such as cars [1], or inland vessels [2], we need more than just 2D images to enable a greater sense of the surroundings of an autonomous vehicle and a real-time obstacle avoidance function [3]. Therefore, state-of-the-art autonomous vehicles detect the surrounding environment by collecting sensory data from cameras, radars, and LiDARs (Light Detection And Ranging). The LiDAR sensor has shown outstanding performance and great accuracy in converting the physical environment into 3D digital data in real-time [3]. It can represent this 3D data as point clouds. The advantage of this representation method is being adaptive and simple compared to other 3D representations, such as polygon meshes. Point clouds allow us to effectively model 3D data of the surrounding environment and describe the shape and distance of the surrounding objects, such as vehicles and pedestrians. In indoor use cases, point clouds can be captured with an RGB-depth camera(s). The RGB-Depth camera is a sensor that can capture both RGB and depth data and has become widely available in commodity devices, such as Intel RealSense or Apple Truth Depth Camera. Because of their limited range, RGB-Depth sensors are often used to acquire point cloud data only in an indoor setup.

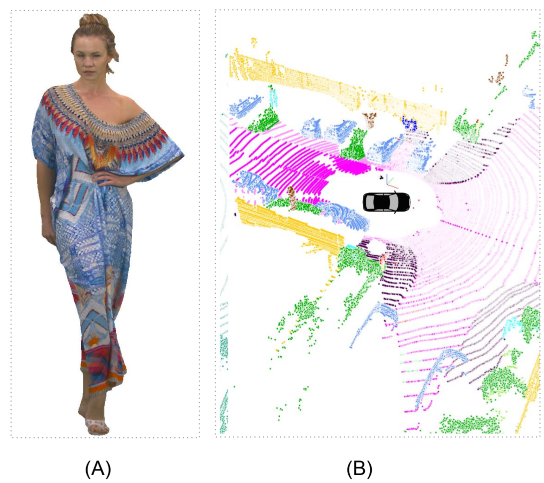

Fig. 1. Point cloud examples: (A) single object point cloud, and (B) multiple object point cloud.

In virtual and augmented realities, the chosen 3D data representation, such as depth images, volumetric grids, polygon meshes, or point clouds, is the dominant influencing factor on the feasibility and limitations of the content streaming experience [4]. The primary difference between point cloud content delivery for virtual or augmented realities applications and delivery of regular 2D video or 360-degree video is the high level of user interaction. Point cloud-based videos capture the 3D space and objects from multiple angles, and thus provide a better immersive experience than regular 2D videos. By that, point cloud-based videos enable new application fields with six Degrees of Freedom (6DoF) [4], allowing the user to freely change their position and orientation in the scenes, which allows high interactivity level with the scenes components, i.e., objects.

However, despite the advantages of point clouds, there are certain difficulties in adopting point clouds in either autonomous vehicles or augmented/virtual reality scenarios [7]. The captured real-time raw point cloud can be many gigabits

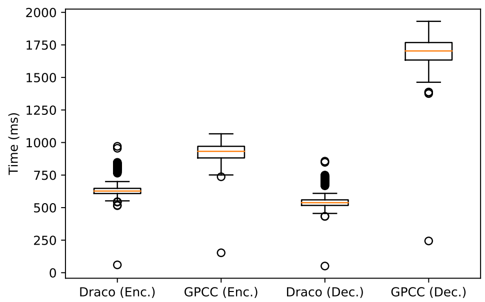

Fig. 2. Comparison of encoding and decoding time, for two point cloud compression methods, namely G-PCC [5] and Draco [6]

per second in data generation, which seriously hinders their application in practical scenarios. For example, the Velodyne VLS-128 can generate 9.6 million points per second [3]. Therefore, major challenges of point cloud-based systems are related to processing and transmitting a large amount of data and adapting the used bit rate to the variable bandwidth conditions [8], [9]. Previous studies have proposed novel compression techniques, including video-based [5] for dense point clouds compression (Figure 1(A)), and geometry-based [5], [10] for sparse point clouds compression, e.g. LiDAR scans (Figure 1(B)) [5], [10]. Even with compression, the bandwidth requirements for compressed point cloud sequences are still relatively high and may be impractical due to added latency. Figure 2 shows overall encoding and decoding time for the same point cloud content using some state-of-the-art compression methods. High encoding and decoding latency make streaming point cloud less applicable for real-time applications.

A point cloud is a collection of points in a 3D space, each having several properties, including point coordinates along (X, Y, and Z) axes, colour values encoded in RGB format, and several others. A point cloud can be a signal object like a human body, see (A) in figure 1, or it can be composed of multiple independent objects. For example, a scan by automotive LiDAR for an urban street can have multiple objects such as vehicles, pedestrians, motorcycles, and trees, look at (B) in figure 1. An additional example is a room scan taken by several RGB-Depth cameras from several vantage points, generated by collecting the point data and then stitching it together by detecting common points. In some situations, however, there is no need to transmit the entire data; instead, only a small portion of the point cloud needs to be delivered. Knowing that each independent point cloud object can be extracted from the mother point cloud, so it can be processed or streamed independently and in controlled quality. This feature seems less applicable to regular 2D images.

With increasing industrial demand, point clouds with deep learning models have lately become a subject of great interest. The point cloud is an essential enabler for many computer vision tasks, including object classification, semantic segmentation, detection, tracking, flow estimation, registration, augmentation, reconstruction, and completion. For example, semantic point cloud segmentation aims to label each point of a point cloud with a corresponding semantic label of what it represents. Semantic segmentation allows identifying and segmentation of different objects within the point cloud. Thus, it enables mapping out point cloud objects that need to be handled differently.

The approach of this research is as follows. To control the resulting network load during multiple object point cloud sequences delivery, we incorporate semantic awareness of the objects which compose the transmitted scenes. We believe that awareness of the objects is a crucial piece of information and can be exploited to enable the content bitrate adaptability in networks with significant bandwidth variations. To evaluate our approach, we used the SemanticKITTI dataset [11], which provides labelled point cloud sequences with distinct classes, including cars, pedestrians, vegetation, etc. We employ a deep neural network for 3D object segmentation to generate semantic information for the underlying point cloud scene, i.e., classes or properties of objects. We leverage the gained semantic awareness to control the volume level of the scene objects. Depending on the application, thus, less essential objects can be transmitted less frequently, in lower quality, or completely filtered out. The resulting possibilities allow better utilization of the available bandwidth and enable content bitrate adaptation in a controlled manner, i.e., increase for the significant objects and decrease for irrelevant objects, and permitting dynamic adapting the bitrate of content to changing network bandwidth or personalized user preferences. We show the proposed approach's effectiveness by applying an experiment-based evaluation. We should point out that our proposed approach is not meant to replace existing point cloud compression, but enhance multiple object point cloud content delivery efficiency by adapting the content to meet the bandwidth or computational requirements with the help of object-related knowledge. This can be combined with any up-/downstream adaptation mechanisms, including compression.

We refer to point cloud semantic segmentation, classification, or labelling, i.e., the task of assigning a semantic label to each point of a point cloud, as point cloud semantic segmentation throughout this paper.

We organize this paper as follows. Section II reviews some related works. Section III describes our proposed method. Section IV details our experimental setup and results. We give the conclusion in section V.

II. Related Works

In the light of the high interest of research and industry in point cloud delivery, various research approaches have been proposed in this regard. However, adaptive delivery is one of the most used methods for reducing bandwidth while transporting high-quality point cloud content. In this section, we review delivery approaches that adapt point cloud content bitrate.

1) Virtual or Augmented Reality: Hosseini and Timmerer are the first authors who proposed an adaptive point cloud streaming approach for virtual or augmented reality applications called DASH-PC [8]. The approach is DASH-compliant for single point cloud streaming and is dynamic adaptive, view-aware, and bandwidth-efficient. The proposed system tackles the changing bandwidth demands for streaming point cloud content by a DASH-compliant MPD (Media Presentation Description) manifest specifically for a point cloud object. Instead of using a dedicated encoder of point cloud objects, they sample the object to provide a range of quality variants. Then, the point cloud object will be fetched on a per-frame basis. As a result, the number of HTTP GET requests is comparable to the frame (object) rate. This may lead to significant issues, particularly in the light of network latency. In a later work, Hosseini extended the work and proposed rate adaptation techniques for streaming multiple point cloud objects [12]. The heuristic's algorithm determines the priority of the point cloud objects based on the camera view, the objects' visibility, and their distance from the camera. Thus, point cloud objects closer to the camera are given higher priority, and they are transmitted in a higher-quality representation. Likewise, point cloud objects farther away are given lower priority and transmitted in a less demanding quality representation.

Park et al. proposed a utility-based rate adaptation heuristic of point cloud based content for augmented reality that supports both network and user adaptation [13]. The proposed system adapts the level-of-detail (LoD) of point cloud objects according to their distance to the user location. They proposed a greedy algorithm for rate-utility optimization that allocates bits between different tiles across multiple objects. In more detail, the system reduces bandwidth demands by reducing the object's level of detail depending on its location and distance to the user's viewport and location. They reduce latency by introducing a window-based buffer to respond quickly to user interaction. According to the results of their evaluation, the proposed heuristic provides better utility and user experience over variable throughput-constrained networks compared to existing video streaming approaches.

Another system proposed by Qian et al., called Nebula, considers point cloud video streaming to commodity smartphones as regular 2D video as a strategy to reduce the computational burden [9]. In their work, they offload the heavy lifting operations to a remote render server and employ rate adaptation mechanisms to adjust the video quality to network conditions by streaming different objects with different qualities according to multiple criteria. In order to reduce the motion-to-photon latency, they propose a viewport prediction mechanism and a mega-viewport idea. They present several optimization techniques to minimize the apparent motion-to-photon latency, dynamically adjust to variable network bandwidth, and reduce the system's resource usage while maintaining a high QoE.

2) Autonomous Vehicle: Hoog et al. studied the feasibility of lossy and lossless compression of point cloud data sharing across inland vessels while maintaining usable point cloud quality to set up situational awareness [2]. They found that lossless compression using BZip2 reduces the size of the point cloud to half, sacrificing no information, whereas lossy compression using Draco [6] produces 25% of the original size while still maintaining an acceptable point cloud quality.

Although the previous studies have successfully shown good bit rate adaptation, they suffer from limitations in considering the transmitted objects' semantics. Nonetheless, they can be combined with our proposed approach.

III. Proposed Method

This section describes our method for adaptively compressing point cloud objects based on their importance in the scene. For that purpose, we rely on different components, which are the point cloud semantic segmentation model, point cloud encoder, and point cloud decoder. First, we describe how understanding the scene, i.e., objects semantics, can be used to build a point cloud adaptor that prepares objects present in a point cloud scene for delivery.

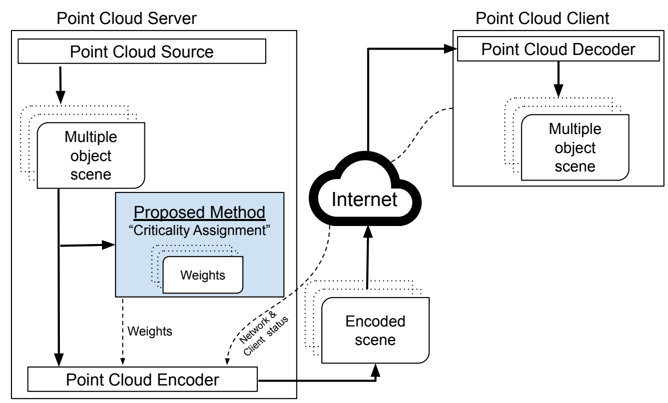

Fig. 3. High-level architecture of point cloud streaming system incorporating our proposed method.

The proposed content-aware bandwidth adaptor is to be integrated with a point cloud streaming system to establish point cloud adaptability and reduce the bandwidth and computational requirements of such streams. The adaptor is implemented as a software component between the point cloud content producer, e.g., LiDAR or recorded files, and the content encoder, as shown in Figure 3. The adaptor is used along with the encoder to prepare objects present in a point cloud scene for streaming. The encoder does not need to be changed with our suggested design. The bitrate is adapted while operating in real-time with the encoding process to satisfy the requirements.

The point cloud scans are subjected to several manipulations by the server before being provided to the client. The point cloud server captures the point cloud scans from the content source. It uses information about the current client and network status to optimize the quality accordingly. These scans are then provided to the encoder, which generates the encoded scenes sent to the client by utilizing the encoding parameters and region of interest (ROI) information given by the adaptor component. ROI information holds the importance of each point.

The assumption is that only a small area of the point cloud scan applies to the user's interest or the application's need. Compressing the entire scan at the same quality and frame rate is unnecessary. Therefore, we reduce the transmitted point cloud sequence bitrate by enabling different update frequencies and allocating different amounts of bits to various objects in each point cloud scan based on the importance of these objects to the user or the intended task. In the following subsections, we present our proposed method for setting and exploiting criticality levels to different objects to establish sharper point cloud bitrate adaptability.

A. Criticality Assignment

Point cloud compression has almost always been considered as a fidelity concern [14] in existing point cloud compression techniques, for example, [15]–[17], where the aim has been to maximize the LoD. However, point cloud content is very application-dependent from a practical standpoint. Therefore, compression may be executed with the purpose of the point cloud content at the receiver side in mind. For example, for decoder agents primarily concerned with localization tasks, removing moving objects' points during the encoding phase will decrease the number of noisy points, increase localization accuracy, and save bitrate [14]. As part of our approach, we rank different objects to their criticality, considering the user's attention or the informative part of the scan. Our strategy is to assign a higher criticality level, i.e., more weight, to the most informative objects to tasks being carried out by the application. Several criticality levels can be offered for the point cloud content, depending on the goals and semantics of the planned point cloud-based application.

We assume that the point cloud content is associated with information about different objects' criticality in point cloud content. A hierarchy of the criticality for the involved objects in the scenes might serve as an example of such information. Using this information, we establish the criticality classes. An example of content criticality information is that an automotive point cloud scan comprises multiple objects, some of which are least relevant to other vehicles. These objects have no contribution to the intended task, for example, vegetation and buildings in urban street scenes [18].

One way to obtain semantic information for point cloud-based content is to use deep learning models. There are many methods to analyse semantic information of point clouds [19]–[22]. However, we believe this is the first work showing how to utilize this information in adaptive point cloud delivery. Figure 4 shows an overview of the criticality assignment process. The pre-trained semantic segmentation model partitions a multiple object scene semantically into different criticality levels, where each point of the given scene is assigned a criticality label.

Looking at Figure 5 (A) and (B), compare the scan before and after applying semantic segmentation for an automotive point cloud scan, respectively. Figure 5 (A) shows a raw scan. Figure 5 (B) shows the same scan after applying semantic

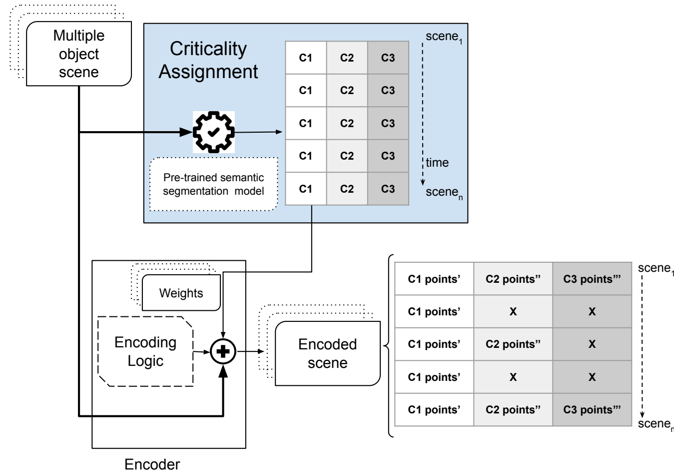

Fig. 4. Detailed view of our proposed method: Criticality assignment. The point-wise semantic segmentation allows assigning a criticality level to each point (in the figure C1, C2 and C3 are criticality levels). The assignment of criticality levels to object classes is application dependent. This enables the encoder to assign different quantization levels and/or frame rates to points depending on their criticality level.

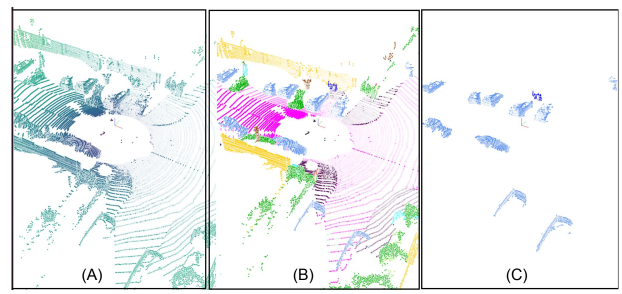

Fig. 5. Visualization of one scan of the SemanticKITTI dataset: (A) The scan with no annotation, (B) The scan with annotation, (C) The scan after adaptation, in this specific case, only the most critical objects, i.e., cars and pedestrians, were considered.

segmentation, where each point in the scan belonging to a car object is coloured blue, and each point belonging to a road is coloured pink. We use this gained point-wise semantic knowledge to estimate the relevance of each object in the scan.

B. Adaptive Quantization and Frame-rate

We aim to adapt the content bitrate to the available bandwidth and avoid a deterioration in overall quality or a compromise of the intended task when delivering point cloud streams to users. Therefore, during the exchange of scans, the sender reduces the bitrate when the network connection deteriorates and increases the bitrate to give a richer experience when the network connection improves. Under the available bit budget, the encoder's rate control component distributes bits to scans. Three levels can be used to allocate bits in order to achieve the average encoding bitrate while adapting to the bandwidth requirements: (i) group of scans level, (ii) scan level, and (iii) object level. The greater the granularity, the higher the level of bitrate control. Increased granularity can enable the encoder to drill down on the points of each constituent object and quantize its data while maintaining the overall quality. Therefore, the awareness of the content, i.e., ROI, is a crucial piece of information and can be exploited to increase the content bitrate control granularity. However, it is important to note that content awareness complicates the rate control procedure since bits must be distributed to represent object importance while still maintaining the target bitrate and avoiding the introduction of large spatial and temporal fluctuations.

To adapt the bit-rate requirements, noninformative objects can be updated infrequently, adopt low quantization levels, or may be omitted altogether if the situation allows. From this perspective, identifying the relevance level of each object within a streamed scene enables better adaptability of point cloud delivery.

In summary, we prepare objects present in a point cloud scene for adaptation by the encoder as follows: We define a priority hierarchy for the involved objects in the scenes. We semantically partition a scene into its constituent semantic objects. We assign quantization and frame rate amounts to each criticality class based on the gained semantics information.

IV. Experimental Evaluation

This section describes our experimental setup, shows the content bitrate adaptation performance, and analyses the efficiency and runtime. We also present our method implementation details.

A. Experiment Setups

*1) **Dataset**:* this study uses the SemanticKITTI dataset [11], a large-scale real-world point clouds dataset designed originally for the semantic scene understanding task. It provides 22 consistent point-wise semantic annotated point cloud sequences comprising around 43k scans, in other words, multiple object point clouds. The data originated from a rotating automotive 3D LiDAR sensor covering the 360-degree field of view. However, the object's back-facing part is always occluded. Figure 5 (A) depicts a birds-eye perspective of a LiDAR scan. A variable number of urban objects surrounded the invisible LiDAR sensor in the centre of the image. Each urban object can be identified as one of 28 classes, such as different types of ground, structure, vehicle, nature, and human.

*2) **Model**:* recently, with the development of deep neural networks, performing point cloud semantic segmentation has significantly improved [23]. This study builds upon a state-of-the-art semantic segmentation deep learning model named SalsaNext [24]. The SalsaNext model can semantically label each point of a full 3D LiDAR scan with a corresponding class of what is being represented. Our proposed method will use point-wise semantic prediction in real-time to enable the adaptability of the point cloud content. We modify SalsaNext open source code to input criticality labels instead of object labels. Then we train the model from scratch with our criticality labels set.

Method	Version	Parameters
Draco	DracoPy 1.2.0	Compression level=1 QP = 15 / 12 / 10
G-PCC	release-v14.0	geomTreeType=Octree positionQuantisationEnabled=1 (True) positionQuantisationEnabled=1 (True) sequenceScale=0.100 codingScale=0.100 inputScale=1000 transformType=Prediction srcUnit=Metre srcUnitLength=1 outputUnitLength=0.001 neighbourAvailBoundaryLog2=8 outputBinaryPly=1 (True)

TABLE I
EXPERIMENT SETUP PARAMETERS

(A)

Scan Size in MB		Crit. 3	Crit. 2	Crit. 1
SemanticKITTI Scans	**Mean**	**1.340**	**0.539 (-59%)**	**0.098 (-92%)**
	STD	0.076	0.151	0.069

(B)

Scan encoding latency in ms (Draco)	**Mean**	**42.83**	**17.77 (-58%)**	**3.17 (-92%)**
	STD	3.71	5.02	2.80
Scan decoding latency in ms (Draco)	**Mean**	**12.83**	**5.36 (-58%)**	**0.70 (-94%)**
	STD	2.39	2.09	0.78

(A) shows the resulting scan size in MB used in the experiments under different levels of criticality.
(B) shows the encoding and decoding latency time in milliseconds per scan with Draco [6] compression.
TABLE II

*3) **Metrics**:* the feasibility and effectiveness investigation takes the form of a case study of exchanging automotive point cloud data. The evaluation of the proposed method considers two aspects: (i) the communication cost metric is defined as the average data volume exchanged between two nodes in megabytes per scan, (ii) data processing latency in milliseconds per scan.

*4) **Baseline Approaches**:* we assess the performance using the metrics mentioned above and compare our method against the baseline that encodes the scans without considering content semantics. Two common point cloud encoding mechanisms, Draco[1] and G-PCC[2], are used in our experiments. Draco and G-PCC experiment setup is summarized in Table I.

*5) **Criticality Level Settings**:* since the dataset has a known number of object classes, we create a three levels hierarchy, i.e., criticality levels, based on object relevance. This is exemplified by objects with the highest relevance to be shared with other vehicles, such as cars and pedestrians, which fall under criticality 1. Stationary objects like sidewalks are less critical and fall under criticality 2. The last critical level, i.e., criticality 3, includes all the other object classes. Figure 6 illustrates these criticality levels in a priority hierarchy. Each criticality level includes the previous one; for example,

[1]https://github.com/google/draco
[2]https://github.com/MPEGGroup/mpeg-pcc-tmc13

Yassin Alkhalili, Thomas Gruczyk, Tobias Meuser, Antonio Fernández Anta, Ahmad Khalil, and Andreas Mauthe. *IEEE Multimedia Big Data (2022)*

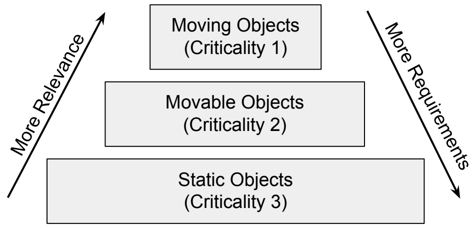

Fig. 6. By using the object's level of dynamicity in an automotive scan, the related objects' priority can be ranked in a hierarchy.

criticality 2 contains the classes of criticality 1 besides its object classes. We attach one of the three criticality levels to each object class and use this as the relevance indicator. As a visual validation, Figure 5 (C) shows a scan after applying adaptation with only objects belonging to critically 1. This can lead to significant bandwidth saving under the assumption that removing less relevant objects does not compromise the intended task. Therefore, as can be seen in Figure 7, it is apparent that each semantic group of points within a scan requires different bandwidth.

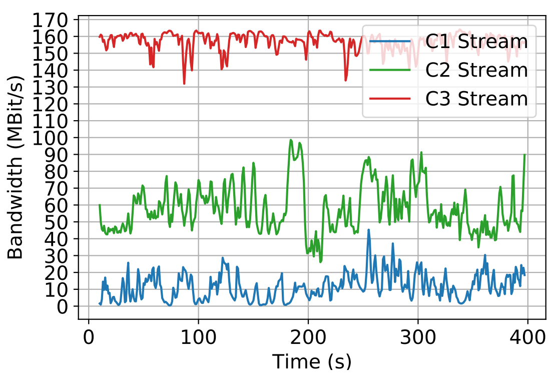

Fig. 7. Enable point cloud sequence adaptability by applying semantic segmentation. Allowing each cumulative semantic group of points to require different bandwidth.

B. Evaluation Results and Comparisons

In the experimental study, we apply our approach to sequence number 8 from the SemanticKITTI dataset. This sequence has 4071 scans. On average, the scan size is 1.34 megabytes, with a standard deviation of 76 kilobytes. A summary of the scan size and the encoding and decoding latency time (LT) for each criticality group are shown in Table II.

*1) **Bandwidth Savings**:* Our approach reduces the transmitted point cloud sequence bitrate by enabling different frame rates or allocating different amounts of bits to scan points according to the assigned criticality level of the scan points. Compared to a baseline scenario where the scans are compressed with the same frame rate and a quantization parameter. We start by showing the bandwidth saving that can be achieved by applying different frame rates in comparison to a baseline with a fixed frame rate. In Figure 8, we plot the bitrate achieved when the frame rate is equal to 10 for all points and when the frame rate is equal to 10, 5, and 2 for criticality 1, 2, and 3 points, respectively. The bitrates of the transmitted scans throughout the three groups and the entire stream have significantly decreased, as seen in the figure.

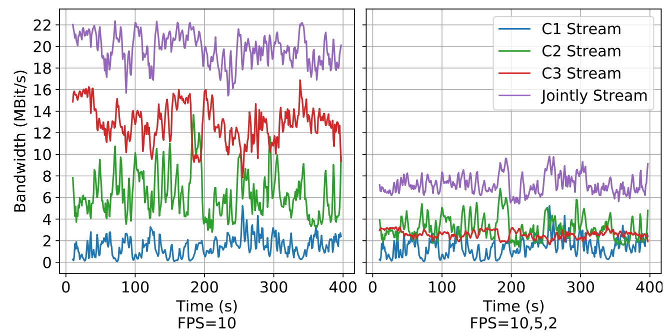

Fig. 8. Bitrates savings by adopting frame rates 10, 5, and 2 for criticality 1, 2, and 3 points, respectively, against fixed frame rate for all points.

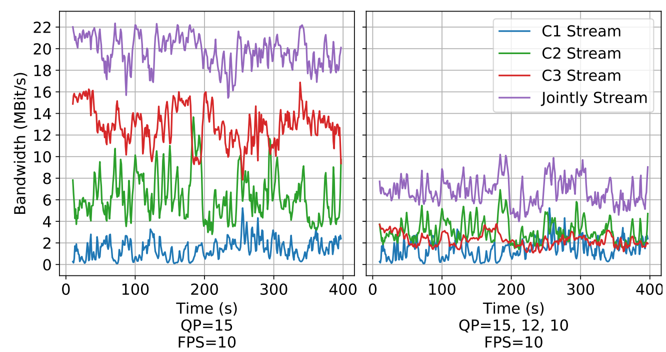

Fig. 9. Bitrates savings by adopting quantization parameters 15, 12, and 10 for criticality 1, 2, and 3 points, respectively, against fixed quantization parameter 15 for all points.

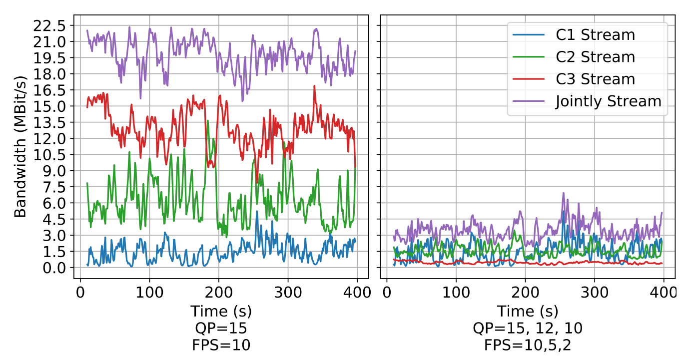

Fig. 10. Bitrates savings by adopting frame rates 10, 5, and 2, and quantization parameters 15, 12, and 10 for criticality 1, 2, and 3 points, respectively, against fixed frame rate and quantization parameters for all points.

Next, we show the bitrate savings achieved by applying different quantization levels per criticality points compared to

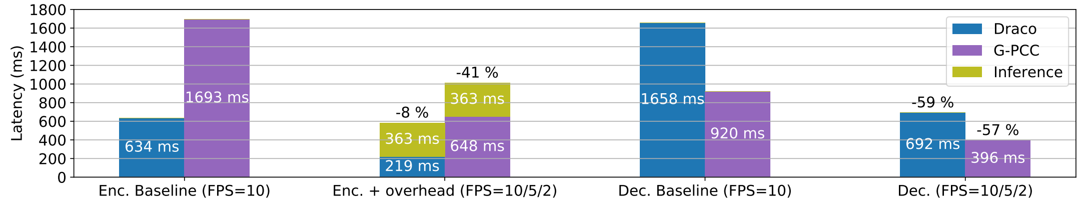

Fig. 11. A comparison of encoding and decoding latency under different frame rates.

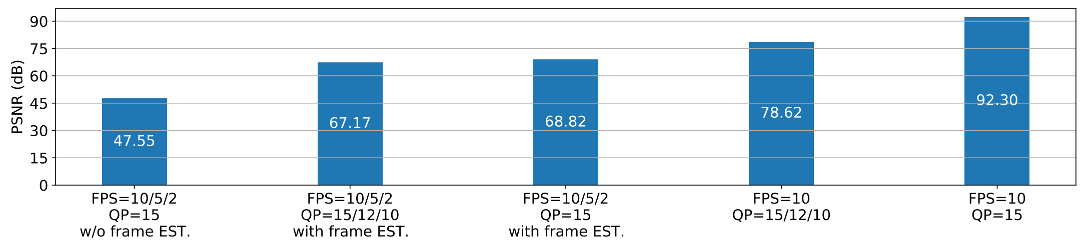

Fig. 12. A Comparison of average PSNR under different adaptations.

a baseline with a fixed quantization parameter for all the scan points. In Figure 9, we plot the bitrate achieved when the quantization parameter is equal to 15 for all points and when it is equal to 15, 12, and 10 for criticality 1, 2, and 3 points, respectively. The figure shows a significant data reduction in the bitrates of the transferred scan's overall stream. We next show in Figure 10 the bitrate savings achieved by applying different quantization levels and frame rates per criticality points in comparison to a baseline with fixed quantization parameter and frame rate for all the scan points. The figures show that content-aware adaptation can achieve bitrates saving from 65% up to 82% compared to classic Draco baseline.

*2) **Efficiency Gains**:* It is, of course, possible that the reduction of data can be translated to improvements in encoding and decoding latency time. We show the computational requirement reduction that can be achieved by applying different frame rates for criticality 1, 2, and 3 points compared to a baseline with a fixed frame rate for all points. In Figure 11, we plot the encoding and decoding latency required by two common point cloud encoding mechanisms, Draco and G-PCC, under different frame rates. The figure shows that the adaptive frame rate performs better than the baseline with a fixed frame rate.

We should point out that the average SalsaNext inference time per scan is 4560 milliseconds and 36.19 milliseconds on CPU and GPU, respectively.

*3) **What About Quality?**:* To illustrate the consequence of our adaptations on the scan's quality, we estimate the quality using PSNR throughout our adaptations. We used a quality measure tool[3] by MPEG that determines objective PSNR for the provided original and modified point clouds.

[3]https://github.com/MPEGGroup/mpeg-pcc-tmc2

Although the quality for high criticality points counts the most, it remains important to maintain good quality for less relevant points. We suggest a simple technique that may be applied at the receiver side to mitigate the quality loss resulting from a reduced frame rate. The key idea is to compensate for the quality loss in the reduced frame rate scans by introducing in-between frames, which can be created by finding a transformation that estimates the positions of the points between two actual frames using the relative position and orientation of the LiDAR. The SemanticKITTI dataset contains information on the LIDAR's relative location and orientation. The position and orientation of the LiDAR, which may be piggybacked to the higher criticality frames, have minimal impact on the bandwidth and computing cost.

In Figure 12, we plot the average PSNR measurements achieved by the entire point cloud sequence under different adaptations. The figure shows that the overall quality (estimated by PSNR) stayed relatively high because of our adaptations.

C. Implementation Details

For the model implementation, Python 3.7 and PyTorch 1.1 were used. The training was performed on a machine running Ubuntu 18.04 with two NVIDIA RTX 2080 GPUs and an Intel Xeon Silver 4112 CPU. The model inference was executed on a Windows 10 based system on a single RTX 3080 Ti in combination with an AMD Ryzen 7 5800X CPU. The architecture is mostly based on SalsaNext. For our use case, we reduced the batch size to a value of six and disabled the KNN post-processing. Also, the original label configuration was replaced with criticality labels instead of object labels. Then, training was carried out over 150 epochs with an initial learning rate of 0.01 using Stochastic Gradient Descend. Finally, for the LiDAR point cloud visualizations, we leveraged the API for SemanticKITTI[4]. All scripts to reproduce the results of this paper will be made publicly available on GitHub[5].

V. Conclusion

In most point cloud content exchange scenarios, a small part of the point cloud can be prioritized instead of transmitting the entire data. Based on that idea, we developed an approach to transmit the prioritized objects with enhanced quality relative to the rest of the objects. Our work aims to explore and evaluate object-related information's effectiveness in facilitating adaptation in multiple object point cloud content streaming, which is crucial for the seamless delivery of these objects under dynamic environmental conditions. In this work, we evaluate our approach with SemanticKITTI scans, showing how content awareness enables the adaptability of point cloud content delivery. This includes predicting the objects' semantics, which is in turn used for estimating each object's importance in the transmitted scan. The results show that our approach leads to improved bandwidth and coding performance compared to a situation without awareness of the content.

Acknowledgment

This work has been funded by the German Research Foundation (DFG) as part of project C3 and B1 within the Collaborative Research Center (CRC) 1053 MAKI. It has been co-funded by the LOEWE initiative (Hessen, Germany) within the emergenCITY centre.

References

[1] J.-V. Zacchi and M. Trapp, "Towards collaborative perception in automated driving: Combining vehicle and infrastructure perspectives," in *CARS 2021 6th International Workshop on Critical Automotive Applications: Robustness & Safety*, Sep. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03366395

[2] J. de Hoog, A. N. Ahmed, A. Anwar, S. Latré, and P. Hellinckx, "Quality-aware compression of point clouds with google draco," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*. Springer International Publishing, Oct. 2021, pp. 227–236. [Online]. Available: https://doi.org/10.1007/978-3-030-89899-1_23

[3] R. Roriz, J. Cabral, and T. Gomes, "Automotive LiDAR technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6282–6297, Jul. 2022. [Online]. Available: https://doi.org/10.1109/tits.2021.3086804

[4] J. van der Hooft, T. Wauters, F. D. Turck, C. Timmerer, and H. Hellwagner, "Towards 6dof HTTP adaptive streaming through point cloud compression," in *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, Oct. 2019. [Online]. Available: https://doi.org/10.1145/3343031.3350917

[5] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: video-based (v-PCC) and geometry-based (g-PCC)," *APSIPA Transactions on Signal and Information Processing*, vol. 9, no. 1, Apr. 2020. [Online]. Available: https://doi.org/10.1017/atsip.2020.12

[6] Google, "Draco 3d graphics compression." [Online]. Available: https://github.com/google/draco

[4] https://github.com/PRBonn/semantic-kitti-api

[5] https://github.com/yaseenit/Content-Aware-Adaptive-Point-Cloud-Delivery.git

[7] Z. Liu, Q. Li, X. Chen, C. Wu, S. Ishihara, J. Li, and Y. Ji, "Point cloud video streaming: Challenges and solutions," *IEEE Network*, vol. 35, no. 5, pp. 202–209, Sep. 2021. [Online]. Available: https://doi.org/10.1109/mnet.101.2000364

[8] M. Hosseini and C. Timmerer, "Dynamic adaptive point cloud streaming," in *Proceedings of the 23rd Packet Video Workshop*. ACM, Jun. 2018. [Online]. Available: https://doi.org/10.1145/3210424.3210429

[9] F. Qian, B. Han, J. Pair, and V. Gopalakrishnan, "Toward practical volumetric video streaming on commodity smartphones," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*. ACM, Feb. 2019. [Online]. Available: https://doi.org/10.1145/3301293.3302358

[10] M. Bui, L.-C. Chang, H. Liu, Q. Zhao, and G. Chen, "Comparative study of 3d point cloud compression methods," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, Dec. 2021. [Online]. Available: https://doi.org/10.1109/bigdata52589.2021.9671822

[11] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019. [Online]. Available: https://doi.org/10.1109/iccv.2019.00939

[12] M. Hosseini, "Adaptive rate allocation for view-aware point-cloud streaming," Sep. 2017. [Online]. Available: https://doi.org/10.13140/RG.2.2.23436.26244

[13] J. Park, P. A. Chou, and J.-N. Hwang, "Rate-utility optimized streaming of volumetric media for augmented reality," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 149–162, Mar. 2019. [Online]. Available: https://doi.org/10.1109/jetcas.2019.2898622

[14] L. Zhao, K.-K. Ma, Z. Liu, Q. Yin, and J. Chen, "Real-time scene-aware LiDAR point cloud compression using semantic prior representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5623–5637, Aug. 2022. [Online]. Available: https://doi.org/10.1109/tcsvt.2022.3145513

[15] I. Viola and P. Cesar, "Volumetric video streaming: Current approaches and implementations," Sep. 2022. [Online]. Available: https://arxiv.org/abs/2209.01982

[16] Y. Zhu, Y. Huang, X. Qiao, Z. Tan, B. Bai, H. Ma, and S. Dustdar, "A semantic-aware transmission with adaptive control scheme for volumetric video service," *IEEE Transactions on Multimedia*, pp. 1–13, Oct. 2022. [Online]. Available: https://doi.org/10.1109/tmm.2022.3217928

[17] Y. Yu, W. Zhang, G. Li, and F. Yang, "A regularized projection-based geometry compression scheme for LiDAR point cloud," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, Sep. 2022. [Online]. Available: https://doi.org/10.1109/tcsvt.2022.3211084

[18] D. Bischoff, F. A. Schiegg, D. Schuller, J. Lemke, B. Becker, and T. Meuser, "Prioritizing relevant information: Decentralized v2x resource allocation for cooperative driving," *IEEE Access*, vol. 9, pp. 135 630–135 656, Sep. 2021. [Online]. Available: https://doi.org/10.1109/access.2021.3116317

[19] J. Chen, Z. Kira, and Y. K. Cho, "Deep learning approach to point cloud scene understanding for automated scan to 3d reconstruction," *Journal of Computing in Civil Engineering*, vol. 33, no. 4, Jul. 2019. [Online]. Available: https://doi.org/10.1061/(asce)cp.1943-5487.0000842

[20] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, "A review of deep learning-based semantic segmentation for point cloud," *IEEE Access*, vol. 7, pp. 179 118–179 133, Dec. 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2958671

[21] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2dpass: 2d priors assisted semantic segmentation on lidar point clouds," Jul. 2022. [Online]. Available: https://arxiv.org/abs/2207.04397

[22] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," Jun. 2022. [Online]. Available: https://arxiv.org/abs/2206.02099

[23] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3d: A review of point cloud semantic segmentation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 38–59, Dec. 2020. [Online]. Available: https://doi.org/10.1109/mgrs.2019.2937630

[24] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving," Mar. 2020. [Online]. Available: https://doi.org/10.48550/arxiv.2003.03653