TECHNISCHE
UNIVERSITÄT
DARMSTADT

CONTENT-AWARE ADAPTATION OF POINT CLOUD STREAMS

A Model-based Perspective on Processing of Point Cloud Streams

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

MHD YASSIN AL KHALILI, M.SC.

Vorsitz: Prof. Dr.-Ing. Tran Quoc Khanh
Referent: Prof. Dr.-Ing. Dr. h.c. Ralf Steinmetz
Korreferent: Prof. Dr. Andreas Mauthe

Tag der Einreichung: 25. Juni 2024
Tag der Disputation: 17. September 2024

Darmstadt 2024

MHD Yassin Al Khalili, M.Sc.: *Content-Aware Adaptation of Point Cloud Streams,* A Model-based Perspective on Processing of Point Cloud Streams
Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung der Dissertation auf TUprints: 2024
Tag der mündlichen Prüfung: 17. September 2024

ABSTRACT

The utilization of point clouds, a three-dimensional (3D) data representation, has recently experienced a significant rise in popularity. Recent advancements and affordability in 3D sensor hardware have played an essential role in driving the widespread adoption of 3D point clouds across diverse domains. These domains include virtual reality, augmented reality, volumetric video, 3D sensing for robotics, smart cities, telepresence, and automated driving applications. Consequently, the availability of point clouds, characterized by millions of data points per frame, has been increasing steadily since. However, the substantial size of point cloud data presents significant challenges regarding efficient transmission, compression, and processing. Existing methods for point cloud compression tend to prioritize data quality preservation, often overlooking the practical utilization of the data. For instance, the primary concern in autonomous vehicles is machine perception tasks, such as vehicle positioning and object detection. Thus, the focus should be more on relevant objects and less, or not on other irrelevant surrounding objects. Dedicating resources to these task-specific objects conserves valuable transmission bandwidth and enhances the overall utility of the point cloud data at the recipient's end.

In this thesis, we consider the unique characteristics of point clouds, particularly the capability to extract individual objects from the original point cloud, which are then processed and streamed independently. Our first contribution improves the understanding of how various configurations, including compression-related parameters, distance, and reduced frame rate, influence the quality of point cloud objects. For our second contribution, we investigate how these configurations affect both output quality and resource demands. Examining these relationships aims to identify configurations that maximize quality while minimizing resource consumption. Based on the collected data, we generalize our contributions by building a machine learning-based model to predict the quality of a given point cloud.

To enable the adaptability of the point cloud content to changing conditions, our third contribution explores how incorporating object-related information, such as object semantics, into point cloud content streaming impacts adaptability and delivery efficiency compared to conventional methods. Through experiments, we extensively evaluate our contributions and show the significant benefits of content-aware streaming. This approach has the potential to enhance point cloud streaming by enabling dynamic content delivery that adapts to changing scenarios.

In conclusion, this thesis introduces the concept of content-aware point cloud adaptation and compares it with alternative state-of-the-art approaches. The contributions of this thesis represent an initial step towards demonstrating the feasibility of content-aware adaptation for enhancing point cloud delivery efficiency.

KURZFASSUNG

In den letzten Jahren haben 3D-Point Clouds, die Darstellungen von dreidimensionalen (3D) Daten sind, aufgrund von technologischen Fortschritten und günstigeren 3D-Sensoren in zahlreichen Bereichen, wie z. B. virtueller Realität, erweiterter Realität, 3D-Videos, Robotik, intelligenten Städten, Telepräsenz und autonomem Fahren, stark an Popularität gewonnen. Die zunehmende Größe und Komplexität von Point Clouds, die Millionen von Datenpunkten pro Szene enthalten können, stellen jedoch Herausforderungen bei der effizienten Übertragung, Komprimierung und Verarbeitung dar. Bisherige Methoden zur Point Cloud Kompression konzentrieren sich in der Regel eine möglichst konstante Datenqualität und vernachlässigen dabei oft die praktische Nutzbarkeit der Daten. So steht bei autonomen Fahrzeugen beispielsweise die Wahrnehmung durch die Maschine im Vordergrund, insbesondere Aufgaben wie Fahrzeugortung und Objekterkennung. Daraus folgt, dass der Fokus stärker auf relevanten Objekten liegen sollte und weniger oder gar nicht auf weniger relevanten Umgebungsobjekten. Priorisierung dieser aufgabenspezifischen Objekte spart wertvolle Übertragungsbandbreite und erhöht den allgemeinen Nutzen der Point Cloud Daten beim Empfänger.

In dieser Arbeit betrachten wir die besonderen Eigenschaften von Point Clouds, insbesondere die Möglichkeit, einzelne Objekte aus der ursprünglichen Point Cloud zu extrahieren, die anschließend unabhängig voneinander verarbeitet und gestreamt werden. Unser erster Beitrag verbessert das Verständnis, wie verschiedene Konfigurationen, einschließlich komprimierungsrelevanter Parameter, Entfernung des Objektes und reduzierte Bildwiederholrate, die Qualität von Point Cloud-Objekten beeinflussen. In unserem zweiten Beitrag untersuchen wir, wie sich diese Konfigurationen auf die Qualität und den Ressourcenbedarf auswirken. Durch die Untersuchung dieser Zusammenhänge sollen Konfigurationen identifiziert werden, die die Qualität maximieren und gleichzeitig den Ressourcenverbrauch minimieren. Auf Basis der gesammelten Daten generalisieren wir unsere Beiträge durch die Entwicklung eines Maschine-Learning-basierten Modells zur Vorhersage der Qualität einer vorgegebenen Point Cloud. Um die Anpassungsfähigkeit von Point Cloud-Inhalten an wechselnde Bedingungen zu ermöglichen, untersucht unser dritter Beitrag, wie sich die Integration objektbezogener Informationen, wie beispielsweise Objektsemantik, in das Point Cloud-Streaming auf die Anpassungsfähigkeit und die Übertragungsleistung im Vergleich zu herkömmlichen Methoden auswirkt. Mittels umfangreicher Experimente evaluieren wir unsere Beiträge detailliert und belegen die erheblichen Vorteile von inhaltsbasiertem Streaming.

Zusammenfassend führt diese Arbeit das Konzept der inhaltsbasierten Point Cloud-Anpassung ein und vergleicht sie mit alternativen, modernen Ansätzen. Die Beiträge dieser Arbeit stellen einen ersten Schritt dar, um die Machbarkeit der inhalts-

basierten Point Cloud-Anpassung zur Verbesserung der Effizienz der Point Cloud-Übertragung zu demonstrieren.

PREVIOUSLY PUBLISHED MATERIAL

This thesis incorporates research previously presented at scientific conferences and workshops. However, no content from these publications is directly reproduced in this thesis. Table 1 outlines these previous publications, none of which are directly replicated within this thesis, except for tables and figures, primarily for evaluation purposes. A full list of the authors publications can be found in Chapter B.

Chapter	Publications
2	[11] , [13] , [177] , [10]
3	[13]
4	[177]
5	[10]

Table 1: Previously published material.

The contributions presented in this thesis are outcomes of collaborative endeavours and collective teamwork conducted at the Multimedia Communications Lab (KOM) within the Technical University of Darmstadt (TU Darmstadt). Unless explicitly mentioned otherwise, the individuals referenced herein were affiliated with KOM during their involvement. This thesis employs the term "we" to denote the collaborative efforts underlying each contribution. In this section, I offer an overview of the contributions made by co-authors and contributors to the respective works.

In addition to the supervision of Prof. Dr.-Ing. Ralf Steinmetz and Prof. Dr. Andreas Mauthe, Prof. Dr. Ing. Amr Rizk, Prof. Dr. Boris Koldehofe, and Dr.-Ing. Tobias Meuser, listed in alphabetical order, oversaw the development of this thesis and its contributions. During their period of office at KOM, they offered invaluable insights and feedback on the proposed approach, development process, implementation, and methodologies employed. While their input greatly influenced this work, I will acknowledge their specific contributions to each section individually.

Chapter 2 offers a comprehensive overview of the existing research in point cloud processing, compression, and streaming. It examines the state-of-the-art to identify relevant research gaps. While preparing previous publications, I conducted various literature reviews for the related work sections. This work incorporates a more recent review, superseding our outdated survey paper on point cloud content streaming published in [12], which was based on work done by Stefano Acquaviti and Yumeng Chen, students at TU Darmstadt. As a result, much of the analysis of related work has already been published, with the primary analysis conducted in [10, 11, 13, 177]. The related work chapter in this thesis is a refined, reorganized, and restructured compilation of these sections. In conducting the literature reviews, I received support from Anam Tahir (Self-Organizing Systems Lab - TU Darmstadt) and Jannis

Weil (KOM), with whom I shared and discussed relevant papers and identified research gaps. The idea of point cloud adaptation using machine learning and neural networks emerged from frequent discussions with Amr Rizk, Boris Koldehofe, and Tobias Meuser regarding recent advancements in deep and machine learning methods for point cloud processing.

The implemented system presented in Chapter 3, an evaluation tool for point cloud streaming, was developed based on the initial system design collaboratively created with Boris Koldehofe, Andreas Mauthe (University of Koblenz, Germany), Tobias Meuser, Anam Tahir, Jannis Weil, and myself. Tobias Meuser initially developed the implementation. With Thomas Gruczyk, a student at TU Darmstadt, I devised the concept of point cloud adaptations using HTTP-DASH. Thomas Gruczyk and I then developed a prototype based on my multidimensional content streaming system, which was published in [13]. The system's implementation underwent continuous enhancement with valuable feedback and assistance from Tobias Meuser. Although not within the main contributions of this thesis, parts of the implemented prototype are used to generate the video content for the user study presented in Chapter 4.

The user study in Chapter 4, which explores the effect of compression and adaptation of point clouds on their quality, was conducted with equal contributions from Anam Tahir, Jannis Weila, and myself [177]. We collaborated on the user studys design and the data analysis. Jannis Weil helped develop a web-based tool for collecting survey participants' feedback. Tobias Meuser provided feedback at the conceptual level, while Prof. Mu Mu (University of Northampton, UK) offered insights into the soundness of the conducted user study. I assembled the users input and validated it. Additionally, Mu Mu supported us (Jannis, Anam, and myself) in authoring the material by proposing useful manuscript revisions. Furthermore, discussions with Prof Dr. Antonio Fernández Anta (IMDEA institute, Spain), Andreas Mauthe, and Mu Mu contributed to our collective understanding of the results' scope and potential interpretation. Jannis Weil also assisted in visualizing the results. Anam Tahir and Jannis Weil supported the research efforts in our weekly research meetings with iterative feedback.

We present a content-aware point cloud adapter in Chapter 5 and evaluate its performance compared to conventional methods. The concept of point cloud adaptation based on semantics was presented at the ACM MMSys 2022 Doctoral Symposium. The final system design evolved through research meetings at the Multimedia Communications Lab. Tobias Meuser provided ongoing support and feedback during our weekly research meetings. I proposed and refined the initial design based on feedback from Tobias Meuser and Antonio Fernández Anta. Once I implemented the design into a technical prototype with Thomas Gruczyk's assistance, I wrote the manuscript [10]. Furthermore, Tobias Meuser and Antonio Fernández Anta helped review the written manuscript. The evaluation presented in this thesis was conducted with valuable feedback from Antonio Fernández Anta and Tobias Meuser.

To ensure proper English grammar and spelling, AI grammar checking tools were utilized throughout the writing process. However, all content within this thesis is original and reflects my own research and analysis.

CONTENTS

1 INTRODUCTION

"The ability to simplify means to eliminate the unnecessary so that the necessary may speak."

Hans Hofmann

THE emergence of ultra-high-definition video content and display devices has transformed the viewing experience of 2D videos. The progress in resolution has also paved the way for the next wave of innovation that is anticipated to go beyond flat screens and provide a fully immersive experience with increased viewing freedom. The proposed innovation will generate a more realistic and engaging viewer experience by incorporating 3D information. Concurrently, the recent advancements in depth-sensing technologies and their increasing affordability have re-energized the utilization of the 3D information in various applications, such as autonomous driving [197] and Extended Reality (XR)[1] [28]. Additionally, the deployment of 5G networks utilizing millimeter-wave technology for high-speed data transfer allows for the transmission of large amounts of data within milliseconds. Thus, it offers a solution to the high demand for bandwidth and low-latency transmission requirements in various applications. Furthermore, the recent advancements in GPU capabilities enable efficient, real-time processing and analysis of complex data, revolutionizing various fields such as artificial intelligence and computer graphics. These developments present an opportunity for researchers to explore new and innovative ways to utilize this 3D information, from improving the safety and reliability of autonomous vehicles to creating more immersive and interactive experiences in XR applications.

Advancements in capturing, transmitting, and displaying 3D information

3D information can be represented using point clouds. Point clouds are a set of points in a 3D coordinate system with optional attributes such as colour and normals. Point cloud delivers accurate distance measurements for autonomous vehicles and robotic applications and enables a realistic representation of 3D objects and scenes in the metaverse. This inherently allows for content consumption with six degrees of freedom. Besides, the point cloud provides a unified representation of 3D content that may combine natural and synthetic objects. This convergence of natural and synthetic 3D data creates highly immersive and interactive applications in the domain of XR. For instance, point cloud data generated by depth RGB cameras can enhance remote collaboration among workers by delivering a more engaging and interactive experience. As a result, workers can experience the scene from different perspectives

The promising potential of point cloud

1 We use XR to refer to related concepts such as Virtual Reality (VR), Mixed Reality (MR), and Augmented Reality (AR).

and go beyond the limitations of a flat screen by using advanced hardware such as high-refresh-rate head-mounted devices and handheld controllers. Thus, workers can exchange information and collaborate effectively in real time and more realistically.

Challenges for communication systems

Point clouds hold significant potential for representing 3D information. However, their adoption in areas such as XR faces several challenges. Firstly, point cloud streams are expected to primarily rely on mobile networks, which may exhibit unpredictable and fluctuating performance. For instance, to fully exploit the high-speed data transfer capabilities of 5G [107], point cloud delivery needs to be network-adaptive and able to handle the inherent volatility of mobile network connections. Secondly, the continuous high data generation rate of raw point clouds, reaching many gigabits per second in real-time, makes direct usage unfeasible for practical usage; for instance, devices like the Velodyne VLS-128 Light Detection and Ranging (LiDAR) can produce up to 9.6 million points per second [140]. Also, most point cloud-based applications have strict low-delay network requirements, as demonstrated by the current virtual reality video applications, which demand a network latency of less than nine milliseconds for an acceptable quality of experience[135, 158], Lastly, the intensive computation required during content decoding at the receiver side further complicates the issue. For example, top-of-the-line mobile phones like the Samsung Galaxy S8 can decode 1.5 frames per second with 100k points per frame [133].

Therefore, handling and transmission of large amounts of point cloud data with strict low-delay network requirements exceeding the transfer and computing capacity of 5G networks presents significant challenges for modern communication systems [65, 100, 169, 183]. These limitations include managing the volume of data and adjusting the bit rate to accommodate variable network conditions, receiver computation abilities, and different environments and applications.

This thesis proposes a methodology for adapting point cloud content to address some challenges and evaluate its effectiveness. The motivation behind our approach to adaptive point cloud delivery is discussed in the following section.

1.1 MOTIVATION FOR ADAPTIVE POINT CLOUD DELIVERY

What is a point cloud?

A point cloud is a collection of disordered points in a 3D space, each having several attributes, including point coordinates along (X, Y, and Z) axes, colour values encoded in RGB format, among other properties. Point clouds offer a highly accurate and realistic 3D representation and can represent various objects, such as natural objects. The added advantage is that they provide six Degrees of Freedom (6DoF) for an immersive and interactive viewing experience. In comparison, traditional 2D videos offer limited interaction with zero degrees of freedom (0DoF), while 360-degree videos offer three Degrees of Freedom (3DoF) through head rotation. However, the full 6DoF experience can be achieved using a VR headset and hand-held controllers, allowing for body movement and rotation in three axes. The creation of high-quality 3D representations of complex objects, such as the human body depicted in Fig-

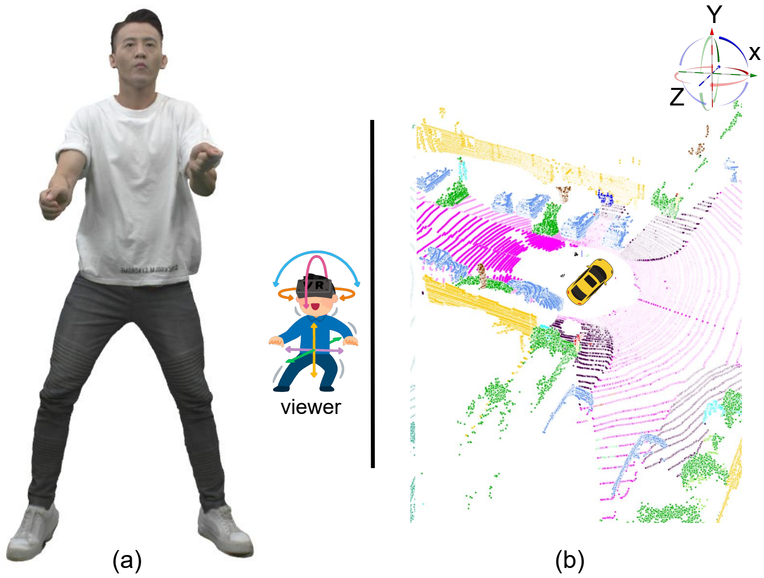


Figure 1.1: Example visualizations of point cloud data. Figure (a) depicts a dense point cloud object that can be viewed from different perspectives. Figure (b) represents a sparse point cloud obtained from an automotive LiDAR, providing accurate measurements of the surrounding objects of the vehicle. The model in (a) and the LiDAR scan in (b) are taken from [184], [18], respectively.

ure 1.1 (a), using point cloud representation requires a substantial number of points, typically in the hundreds of thousands or millions [27]. Similarly, the representation of data generated from an automotive LiDAR sensor using point cloud, as shown in Figure 1.1 (b), requires handling a significant volume of data. For example, devices like the Velodyne VLS-128 can generate up to 9.6M points per second [140], which can result in a large bandwidth requirement for streaming these uncompressed scans: $6\,\frac{\text{bytes}}{\text{point}} \times 9.6\text{M}\,\frac{\text{points}}{\text{second}} \times 8\,\text{bits} = 460.8\text{Mbps}$. There still needs to be optimized point cloud content compression for real-time systems, and there currently needs to be dedicated hardware specifically designed for this purpose.

The motivation behind adaptive point cloud mechanisms is to optimize exchanging point clouds in dynamic applications and fully utilize their potential. These mechanisms adapt the point cloud data based on changing network conditions, user interactivity levels, receiver device capabilities, and application requirements. This ensures the effective fulfillment of point cloud content's intended purpose while reducing data transmission rates. It is crucial in virtual and augmented reality applications, where real-time updates and high-quality content are necessary to create an immersive experience. The user's experience can be optimized by employing adaptive point cloud mechanisms, making the content more accessible and enjoyable [73, 133, 169].

To address the challenges mentioned so far, academia and industry are exploring new and innovative methods to enhance the efficiency of point cloud delivery. The importance of compression in optimizing the performance of point cloud delivery has been established in previous studies [114, 119]. Considerably, previous research has proposed novel point cloud data compression methods [7, 27, 84, 142, 143, 148, 191, 203]. However, despite these advancements, the bandwidth and computing requirements for compressed point cloud sequences remain substantial [26, 98].

It is widely understood that compressing data alone will not effectively manage the challenges of delivering point cloud data. A combination of compression and other methods is necessary to overcome these challenges [55, 100]. Several mechanisms focus on reducing the amount of transferred data by employing techniques such as subsampling or encoding point clouds into multiple representations of varying quality [58, 99, 133, 160, 169, 203]. This approach aims to make the system more adaptive to bandwidth constraints and select a representation suitable for the available bandwidth. To further enhance this method, utility-adaptive approaches are investigated [57, 128]. These approaches allocate bandwidth resources based on the utility of different regions, considering factors like the user's view frustum, the distance between the user and objects, device size, and display resolution. Exploiting redundancies in the original data can help decrease the rate requirements for point cloud data delivery. In some instances, background regions may be deemed less critical, and allocating weighted bitrates or bandwidth resources can be based on the user's field of view [142, 151, 175]. Another mechanism involves frame-skipping [119, 120], such as when the current frame closely resembles the previous frame. Moreover, some data parts might not be visible at any given moment, for example, when an object is occluded by other 3D objects or outside the field of view. Predicting the user's position and field of view while viewing volumetric content can significantly reduce network requirements without impacting visual quality [49, 50, 92, 130, 151]. The adaptation of well-established methods from the 2D video streaming field to point cloud data is evident in the DASH-PC approach [21, 166]. This method is both bandwidth- and user-adaptive, offering multiple representations with varying levels of detail for a point cloud [58]. Consequently, users can select the optimal representation based on their preferences or network conditions. Furthermore, offloading processing to the edge server can significantly reduce the client device's required bandwidth and computing power [41, 101, 133, 191].

Current methods for point cloud compression focus on preserving data fidelity without considering real-world usage at the receiving end. Meanwhile, streaming optimization techniques are adapted from 2D video methods, overlooking the 3D characteristics of point clouds, such as the ability to extract individual objects from the original data. This is then processed and streamed independently. Furthermore, these approaches often lack integration with machine learning for adaptive point cloud delivery.

Understanding content for improved point cloud delivery

Our proposed approach addresses this challenge by selectively encoding and transmitting only the crucial portion of the point cloud data based on the content's object-related information, thus reducing the transmission bandwidth and improving the performance of the received data. For instance, in virtual reality applications, particular objects may be more critical to the user than others [144, 181]. Our proposed methodology can adapt by selectively adapting and transmitting those significant parts of the point cloud data, leading to improvement in the user experience or meeting the resource requirements.

1.2 RESEARCH CHALLENGES

The proposed methodology for adaptive point cloud delivery faces the following research challenges.

Challenge 1: The lack of well-established assessment methods and metrics for point cloud data quality

Compression and adaptation of point cloud data can lead to various types of distortions, including reductions in the number of points and changes in their positions and colours. These distortions can negatively affect the user's experience of the point cloud content. To mitigate this, it is crucial to have a robust and precise mechanism to evaluate the compressed point cloud data quality. This model can help identify the best compression algorithms and parameters while adapting the content to changing conditions. However, developing such reliable mechanisms is challenging due to the diversity of distortions that compression and adaptation can introduce. The 6 degrees of freedom viewing process also makes traditional objective quality evaluation metrics inapplicable to point cloud video quality assessment. Given that point clouds are frequently used in real-time applications, an efficient and fast evaluation method is essential.

Challenge 2: While compression methods are essential for point cloud streaming, achieving efficient delivery requires a multifaceted approach. The challenge lies in integrating complementary methods such as frame skipping, filtration, and dynamic adaptation alongside compression methods.

To provide an accurate perception of surrounding objects for autonomous vehicles or create interactive and personalized immersive experiences for XR viewers, it is essential to stream point cloud data consistently with usable quality in real time. However, this task is challenging due to the bulky and bandwidth-intensive nature of point cloud sequences and the high data rate from hardware sensors. Processing these point cloud sequences requires significant computation and can still be excessively large even after being compressed with state-of-the-art methods. Thus, additional adaptation mechanisms are necessary to control the network load while streaming point cloud sequences and enable content bitrate adaptation in networks with varying bandwidths. This involves adjusting the compression parameters and frame rate or filtering unrelated points to meet the application's requirements and provide the best possible quality to the end user. By leveraging the partitionability of the point cloud content, each point cloud object can be streamed independently and with variable quality.

1.3 RESEARCH GOALS AND CONTRIBUTIONS

This work aims to establish and evaluate a methodology for efficiently transmitting critical point cloud data, incorporating the semantic information of the streamed content. In Figure 4.1, we present an architecture that encompasses the encoding, transmission, decoding, and quality assessment processes, demonstrating the contributions of this thesis. In the following, we list the research goals and their contributions to them.

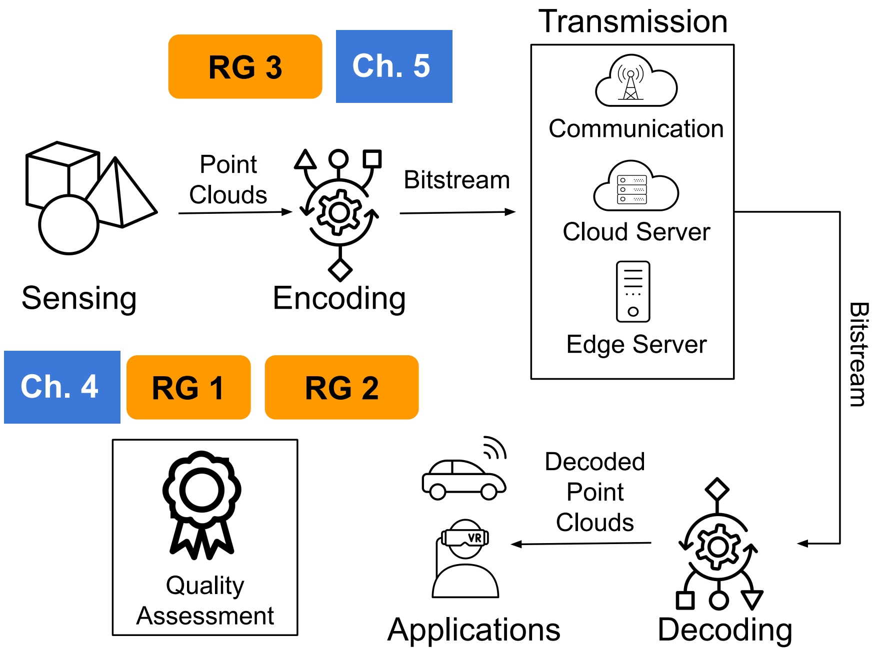


Figure 1.2: This thesis's contributions and research goals (RGs) are presented in an architecture. The quality assessment part encompasses the first and second research goals, which involve conducting a user study to evaluate compressed point cloud sequences and developing a QoE model based on the results. The encoding part corresponds to the third research goal, which involves investigating the impact of incorporating object-related information in point cloud content streaming on the quality and efficiency of the delivery process.

Research Goal 1: *Investigating the impact of compression-related distortions and reduced frame rate on the quality of point cloud objects*

We aim to conduct a user study using state-of-the-art compression methods to assess the impact of various compression quantization levels, frame rates, and camera distances on the perceived quality of point cloud content. The study aims to gain valuable insights into how these factors impact the perception and experience of point cloud content. It will also help to determine how these factors affect the quality of the content. While other extrinsic factors, such as the display resolution, screen size, lighting conditions, and user demographic information, may also impact the perceived quality of point cloud content, they are not the main focus of the study and will not be explored in detail. The results can be used to compare the performance of different compression methods and evaluate their ability to preserve the quality of the point cloud data. The study's results can be used to predict the quality of point cloud sequences and inform the next research goal.

Research Goal 2: *Investigating the correlation between quality and resource demands, with the objective of developing a predictive model for evaluating the quality of point cloud sequences*

The user study results in RG1 are then utilized to determine the trade-offs between resource requirements and quality levels. Moreover, it can be used to develop a Quality of Experience (QoE) model that can predict the perceived quality of point cloud sequences, enabling optimization and adaptation of point cloud streaming mechanisms. Notably, while the objective of Research Goal 2 is to develop a QoE model, it does not aim to find the ultimate solution for all point cloud scenarios or determine the absolute best compression method. Additionally, the QoE model is not intended to replace other quality assessment methods, such as objective metrics, but instead aims to complement them and provide additional insights into the perceived quality of point cloud sequences.

Research Goal 3: *Investigating the impact of incorporating object-related information in point cloud content streaming on the quality and efficiency of the delivery process compared to conventional methods*

This research goal is focused on validating the proposed concept of using high-level clues, such as object semantics, to improve the efficiency of point cloud content delivery through an exhaustive evaluation. This will demonstrate the improved efficiency of the proposed method compared to existing methods by leveraging the high partitionability of the point cloud content for independent processing and variable quality of point cloud objects. The proposed approach aims to enhance the efficiency of point cloud content delivery by adapting the content to meet bandwidth or computational requirements using object-related knowledge. It can be combined with any underlying adaptation mechanisms, including compression. The experimental results will illustrate the effectiveness of the proposed approach in enabling better utilization of the available bandwidth and dynamic adaptation of content bitrate based on changing network bandwidth, the application need, or personalized user preferences. This research goal does not focus on developing a new compression method but rather on incorporating object-related information to enhance the efficiency of point cloud content delivery in combination with existing compression methods.

1.4 STRUCTURE OF THE THESIS

The structure of the thesis is as follows: Chapter 1 outlines the research goals and provides insight into the proposed methodology for delivering adaptive point clouds. Chapter 2 provides the necessary background information for understanding the contributions. This includes an overview of 3D content, compression and visualization, point cloud content adaptation, edge computing, viewport prediction, and point cloud quality assessment. The chapter also identifies research gaps, and reviews related approaches. Chapter 3 presents an experimental tool for point cloud streaming, which generates content for the user study. Chapter 4 focuses on the first and second research goal, which involves conducting a user study to evaluate compressed point cloud sequences, understanding the trade-offs between resource requirements and quality levels, and developing a QoE model based on the user study results. Chapter 5 focuses on the third research goal, which involves investigating the impact of incorporating object-related information in point cloud content streaming on the quality and efficiency of the delivery process. The effectiveness of this approach in facilitating adaptations will be evaluated and compared to conventional methods. Chapter 6 provides the summary of the thesis. It also entails the contributions and future opportunities for further work.

2

BACKGROUND AND RELATED WORK

This chapter provides an overview of the key concepts of various types of 3D data representation and the compression, adaptation, and quality assessment of point cloud content. Figure 2.1 provides an overview of the topics covered in this chapter. This chapter has three main sections, focusing on (i) 3D representation and compression, (ii) adaptation, and (iii) quality assessment of point clouds. Section 2.1 investigates the possible representations of 3D data and examines various point cloud compression mechanisms, including conventional and cutting-edge methods. Section 2.2 explores the core concepts of point cloud adaptation, including adaptive bitrate streaming, viewport predication, and edge computing assistant mechanisms. This section also reviews the most relevant machine learning approaches in this area. Section 2.3 assesses point cloud quality and provides an in-depth review of related research, covering essential topics such as Quality of Service (QoS), Quality of Experience (QoE), and motion-to-photon latency. Additionally, we examine the use of machine learning for QoE modelling. In this thesis, the terms 'point cloud', 'point clouds', '3D', and 'volumetric' were used interchangeably throughout the text when used in conjunction with 'sequence', 'video' or 'content'.

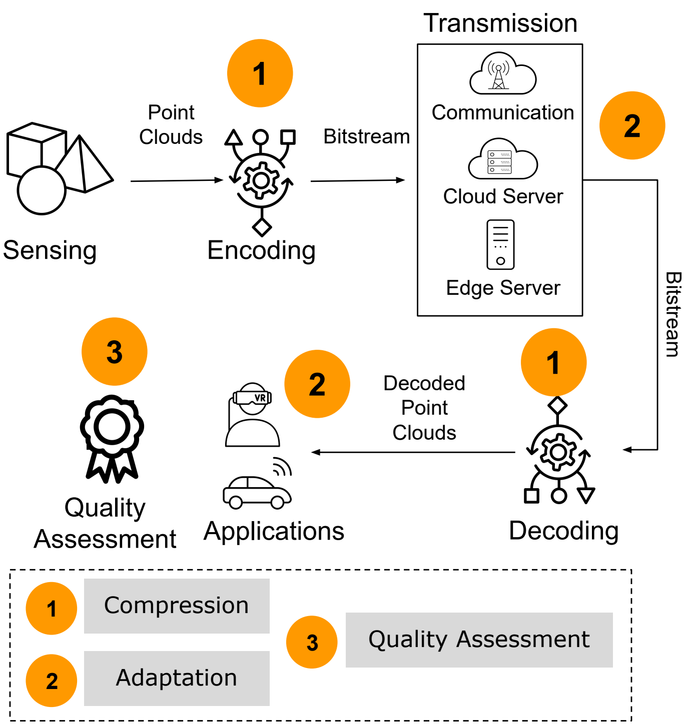


Figure 2.1: Chapter structure: background and related work based on the topics covered in this thesis.

2.1 3D DATA REPRESENTATION AND ASSOCIATED COMPRESSION METHODS

The amount and type of 3D data required for 3D data content applications burden data processing resources. This section considers the representation of 3D data content and the compression methods used to enable adaptive and faster processing.

2.1.1 *Background*

3D data refers to three-dimensional information of objects or characters [113]. This type of data, often featuring realistic details and textures, is utilized to enhance the immersion and realism in multimedia applications, particularly in the context of Extended Reality (XR) [70, 87, 124, 125], and immersive communications [32]. In robotics and autonomous vehicle systems, 3D data creates a map of the surrounding objects and their relative positions, and is used to perceive and navigate the environment [1, 140].

Representations for 3D Data

There are various ways to represent 3D data [71], each with its strengths and weaknesses. The selected 3D data representation, such as depth images [108], volumetric grids [112], polygon meshes [16], or point clouds [97], plays a significant role in determining the feasibility and constraints of the content streaming experience in virtual and augmented realities [200]. According to Hughes [71], the two most widely used methods are point clouds and polygonal mesh representations. Point clouds store 3D data as individual points on an underlying sampled surface with their associated attributes, such as colour, while polygonal meshes represent 3D objects using vertices, edges, and faces [26]. Faces are typically triangles but can also be squares or other shapes [105]. Compared to 3D meshes, point clouds provide a simpler, denser, more naturalistic representation [25, 26, 89]. Emin Zerman et al. [193] concluded that meshes offer superior visualization and are particularly beneficial for applications requiring high-bitrate bandwidth, such as those exceeding 50 Mbps. Point clouds are better suited for scenarios characterized by constrained bandwidth, typically under 20 Mbps.

Contrasting point clouds and mesh representations

Polygonal mesh representations are commonly used in computer graphics, animation, and gaming applications. Point clouds are simpler to generate since they do not require creating a connected surface like meshes. Therefore, point clouds are often better suited than meshes for applications that involve capturing and analyzing raw 3D data, particularly in cases where the data is acquired from sensors such as Light Detection and Ranging (LiDAR) scanners or RGB-Depth cameras [140]. However, meshes are frequently utilized to represent the output of 3D modelling software. Figure 2.2 compares mesh and point cloud representations for a 3D object. The mesh representation on the right accurately captures fine details of the object, while the point cloud representation on the left offers a lightweight alternative [148]. However, many natural objects, such as hair or forests, have a special shape that cannot be easily shown using polygonal meshes. Furthermore, meshes present challenges due to their significant data volume and high rendering complexity. While various compression methods have been developed for meshes in previous research [60, 67], these methods have primarily concentrated on computer-generated 3D models. The choice between point clouds and meshes will depend on the specific requirements of a given application.

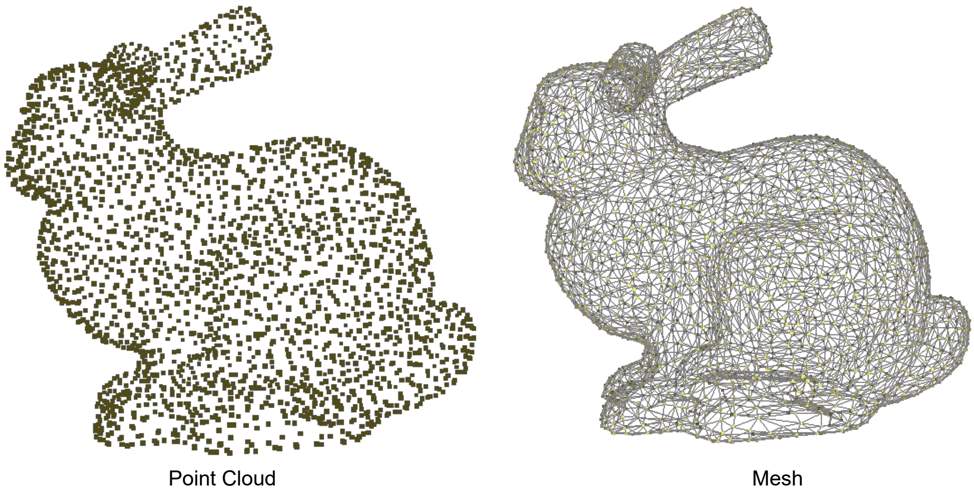


Figure 2.2: Comparison of point cloud and mesh representations. A 3D object, "Stanford Bunny" [86], rendered in two different formats.

The critical distinction between point clouds and polygonal meshes is their structure. Point clouds are unstructured representations of the surface of an object, while polygonal meshes are structured representations. Point clouds lack the concept of edges, faces, and polygons that are present in meshes, and they do not have ordering relations. Point clouds lack connectivity information that can help predict neighboring points and establish geometric correlations among them [147]. However, point clouds are faster to acquire and more straightforward than mesh representations [115]. Because they lack connectivity restraints, point clouds are more convenient for computing, transmitting, and storing real-time acquired 3D scenes. Point clouds are not new, but in recent years, there has been growing interest in their use as a representation of 3D data from both academia and industry. This is due to the adaptability of point clouds to different data sources, such as LiDAR and depth RGB cameras. In indoor uses, point clouds can be captured with an RGB-Depth camera [167]. The RGB-D sensor can capture both RGB and depth data and has become widely available in commercial devices, such as Intel RealSense or Apple Truth Depth Camera. Due to their limited range, RGB-D sensors are typically used to acquire point cloud data only in indoor settings.

A 3D point cloud is a set of points $\{x_1, x_2, x_3, \ldots, x_n\}$ that exist in the three-dimensional space. An illustrative example is depicted in Figure 2.3 with each point representing a data sample with its spatial coordinates encoded along the X, Y, and Z axes. Each cloud point contains geometric and attribute information, which helps describe its visual characteristics. These attributes can vary depending on the use case, but they often include colour values (R, G, B), normal vectors (nx, ny, nz), and sometimes reflectance information. The geometric information of a point cloud re-

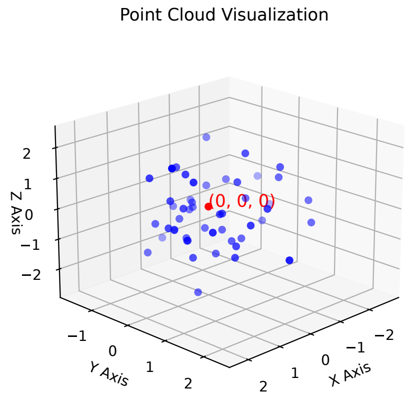


Figure 2.3: Visualization of a random point cloud: A collection of three-dimensional points scattered in space.

lates to the position of each point within a Cartesian coordinate system. Point clouds are typically represented by the (X, Y, Z) coordinates, indicating the spatial location of the points. Table 2.1 illustrates a selection of attributes associated with points in a point cloud. The attributes listed in this table are incomplete, and a point cloud may contain components that are not included in this list. The choice of attributes in a point cloud is application-dependent, meaning that each application may require specific attributes. The term "cloud" is used to convey the unstructured and unordered nature of the point set [123], comparable to a scattered collection of points rather than a well-defined geometric object. In most scenarios, the geometric coordinates in a 3D point cloud are typically represented as floating point values. However, in specific specialized applications, such as real-time systems, utilizing an integer representation for the coordinates can be advantageous. This alternative approach offers benefits such as saving CPU cycles and optimizing memory usage [26].

Attribute	**Description**
X-coordinate	Position of the point along the X-axis
Y-coordinate	Position of the point along the Y-axis
Z-coordinate	Position of the point along the Z-axis
Colour (R, G, B)	Colour information of the point
Intensity	Level of the intensity of the returned pulse
Normal Vector (nx, ny, nz)	Surface normal vector of the point
Confidence	Measurement of the point's confidence level
Classification	Categorizes a point into classes, such as ground, building, vegetation, and others
Scan angle	The angle of the pulse that the point was scanned at
Timestamp	The time when the point measurement was taken

Table 2.1: Attributes associated with points in a point cloud.

This thesis focuses specifically on point cloud data rather than other forms of 3D data representation, such as polygonal mesh or voxel-based models. Point clouds offer a representation of 3D data that can be easily captured and processed from real-world scenes using LiDAR or other sensing hardware [169]. Additionally, point clouds are becoming increasingly important for applications such as autonomous driving, robotics, and virtual and augmented reality, where capturing and processing 3D spatial data accurately is critical. Therefore, this thesis seeks to address the challenges and opportunities associated with point cloud data to contribute to this field's ongoing advancement.

Degrees of Freedom

In the XR context, the term Degrees of Freedom (DoF) refers to the extent of interaction and movement that a user can experience within a virtual environment [121, 169]. In traditional videos, users have limited interactivity. Using the interface buttons, they can only control playback through essential functions such as play, pause, stop, rewind, and fast-forward, resulting in zero Degrees of Freedom (0DoF) [13]. In 360-degree videos, users who are wearing a Virtual Reality (VR) headset can move and experience three Degrees of Freedom (3DoF). This allows them to look around in all directions by rotating their head. With six Degrees of Freedom (6DoF) VR environments, users are not limited to simply changing their head orientations but are also able to move and position themselves within the virtual reality freely. This is achieved through a VR headset and hand-held controllers, allowing for movement and rotation in the X, Y, Z, pitch, yaw, and roll axes. This additional freedom offers a more immersive and interactive experience for the user [169]. Figure 2.4 showcases the interaction with the content at various DoF levels.

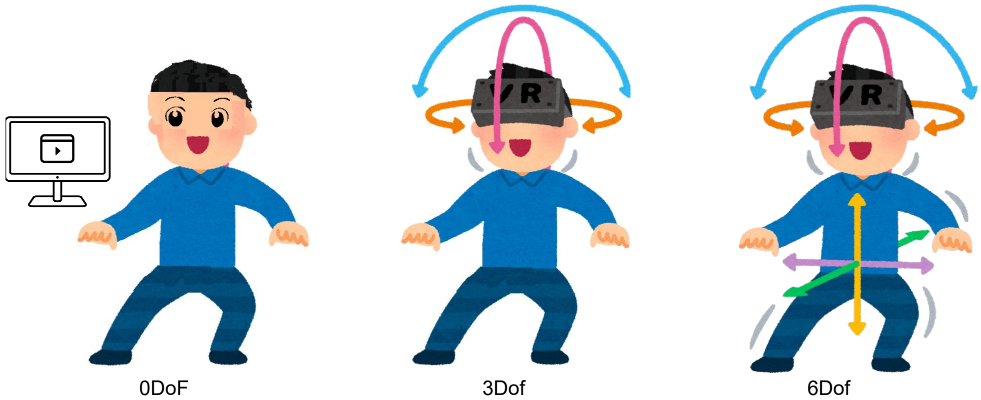


Figure 2.4: Comparing Degrees of Freedom. User interaction with content at 0DoF, 3DoF, and 6DoF. Image source[1]

6DoF data can represent the movement and rotation of various types of objects within a virtual environment. For example, the movement and rotation of human body parts, such as arms or legs, can be represented as 6DoF data, which can be used in virtual reality applications. Additionally, LiDAR can be used to measure the movement and rotation of any physical object, such as cars and pedestrians, that can be represented as 6DoF data [73]. 6DoF data can be represented using meshes and point clouds, as discussed in the previous section.

1 `https://venturebeat.com/games/how-virtual-reality-positional-tracking-works/` last accessed: December 2, 2024.

Static and Dynamic Point Clouds

Static 3D point clouds refer to data representing stationary or unchanging scenes or objects. These point clouds are generated by scanning or capturing the geometry and spatial information of a fixed object or scene at a specific moment. Static point clouds find applications in architectural modelling, 3D mapping, and digital preservation. An example of a static point cloud is the Stanford Bunny, depicted in Figure 2.2.

Dynamic 3D point clouds capture objects or scenes in motion or that experience changes over time. These point clouds depict the spatio-temporal evolution of objects or environments. To create dynamic point clouds, multiple scans or frames of a moving object or scene are captured and then aligned and integrated to form a sequence of point cloud data. This enables capturing time-varying information, including object motion, deformations, and interactions. Dynamic point clouds are utilized in autonomous driving, virtual reality, augmented reality, volumetric video, and telepresence applications.

	Dancer	Thaidancer
Approximate point count in a frame	2,592,758	3,197,804
Approximate 30 FPS bitrate (uncompressed) in MB/s	1,245.6	1,504.8

Figure 2.5: Two examples of dynamic point clouds are presented, with the number of points per frame and the bandwidth required for transmitting their uncompressed data. It should be noted that the count of points can fluctuate from one frame to another.

To achieve visually appealing and naturalistic 3D representations, it is important to work with high-density point clouds. Figure 2.5 illustrates two dynamic point clouds: the "Dancer" from the Owlii dynamic human mesh sequence dataset [184] and "Thidancer" from the 8i Voxelized Surface Light Field dataset [83], along with a representative frame. These examples provide information on the uncompressed content's point count and corresponding bitrates (in MBytes/s). The high corresponding bitrates, around 1500 MB/s, highlight the necessity for efficient compression and adaptation mechanisms to handle such high-bandwidth data. The primary emphasis of this thesis will be on dynamic point clouds. Consequently, the following sections will explore and discuss only compression methods specifically designed for dynamic point clouds.

2.1.2 *Point Cloud Compression*

Point cloud compression reduces the size of point cloud data while maintaining a sufficient level of fidelity for the intended application [185]. Compressing point clouds can significantly reduce the storage and transmission requirements while maintaining essential information [26]. The two primary compression approaches are lossy and lossless compression [162]. Lossless compression minimizes data size by detecting and eliminating statistical redundancy, thereby keeping all the original data intact. Lossy compression decreases data size by discarding unnecessary information (e.g., visually insignificant) through quantization. Quantization reduces the number of bits used to represent the point data. For example, the X, Y, and Z coordinates may be quantized to a smaller number of bits, reducing the total size of the point cloud data.

Dynamic point cloud data is typically captured at a high resolution and contains much redundant information, leading to large file sizes and significant storage and transmission costs [100]. Thus, dynamic point cloud data, with high spatial and temporal redundancy, is often massive. According to Cao [26], point cloud data has different compression mechanisms, including 1D traversal compression, 2D projection utilizing existing 2D video compression methods, and 3D correlations through direct 3D data analysis.

The methodology of 1D traversal compression is popular in point cloud compression. The fundamental principle of 1D prediction methods is constructing tree-based connectivity that uses geometric distances between points to establish neighborhood relations [147]. This converts geometry data into a 1D structure. Several studies [66, 88, 134] have explored this concept, using mechanisms such as minimization of prediction residuals and multiple geometry predictors to address correlations between neighbouring points. These compression methods are simple to implement; however, tree-based traversal methods have limited compression performance because they do not fully consider the 3D spatial correlations, indicating the need for exploiting higher dimensional correlations [26].

Additionally, differential coding serves as a compression technique employed for point cloud data, leveraging the spatial correlation among adjacent points. This method represents each point within a point cloud as the disparity between its coordinates and those of its nearest neighbour. This approach remarkably reduces the required number of bits for storing the point cloud. One advantage of differential coding is that it can be combined with other compression methods, such as octree and kd-tree compression. The algorithm can achieve even higher compression ratios by applying differential coding to the point cloud and other compression methods, as demonstrated by Huang et al. in [68].

3-D-to-2-D projection-based approaches convert the 3D point cloud into 2D images or videos through projection or mapping methods and then use existing image or video coding algorithms to compress the data. Various studies [5, 62] have employed methods for this purpose. Some involve dividing the point cloud into patches and representing each patch as a height field, while others project a grid pattern onto the object and compress the resulting 3D curves. Additional approaches include using view representation combining octree and projection-based methods and utilizing video codecs to compress the geometry and texture information of dynamic point clouds. The advantage of projection-based methods lies in their efficiency and ability to leverage existing image or video compression methods. These methods are expected to benefit from future advancements in image or video codecs.

Finally, the 3D correlation methodology has three main approaches to point cloud compression: voxel-based, clustering-based, and geometric-based. Voxel-based methods divide the point cloud into a regular grid of voxels and then encode each voxel as either occupied or not occupied. This is a simple and efficient way to compress point clouds, but it can only be helpful if the voxel size is manageable. Many deep learning methods for point clouds utilize a voxel representation, similar to the approach taken by Qi et al. in their study [132]. This representation offers a structured and regular format that seamlessly integrates the grid-based architecture of neural networks.

The clustering-based approach can reduce the data needed to represent a point cloud by grouping points with similar attributes. This approach's efficiency relies on successfully identifying meaningful clusters within the point cloud's overall structure. The hierarchical segmentation method introduced by Zhang et al. [196] is an excellent example of how colour information can effectively compress point clouds. By initially performing a global colour segmentation, points are grouped based on their colours. Following this, local segmentation within the geometric space is performed to ensure colour consistency within the resulting clusters. This combined approach of color and geometric segmentation allows for efficient intra-prediction. Similarly, the method proposed by Lien et al. [96] uses a skeleton-based motion estimation approach for kinematic parameter extraction. This approach clusters the point cloud into point sets associated with skeletons, which allows for more accurate motion estimation. However, the noise present in the point clouds can significantly affect the accuracy of the clustering, which could potentially hinder the compression process. Geometric-driven methods, the technique introduced by Navarrete et al. [122], showcase the effective use of geometric and color features for point cloud compression. The points are clustered into planar patches, approximated using Gaussian

Mixture Models (GMMs). By adjusting the configuration of the model, it is possible to create multiple Level Of Detail (LoD), which adds flexibility to the compression process. Overall, while each method presents a unique approach to the point cloud compression problem, they all use some form of clustering to reduce the complexity of the point clouds. These methods allow for efficient representation and transmission of 3D point cloud data, whether based on color, geometry, or a combination of both. However, they also highlight the challenges involved in this process, such as the effect of noise on the accuracy of clustering or the need to balance compression efficiency with quality.

2.1.3 *Standardized Point Cloud Compression*

The recently standardized MPEG Video codec based Point Cloud Compression (V-PCC) and Geometry codec based Point Cloud Compression (G-PCC) codecs are the most current and efficient solutions available for point cloud compression [46]. These codecs are part of the MPEG-I set of standards, which aim to develop critical mechanisms for immersive media [148]. As the MPEG G-PCC and V-PCC codecs are often utilized in this thesis, they will be examined in this section. These codecs embody the two primary mechanisms for organizing point cloud data for coding purposes: surface-based and patch-based approaches. A surface is a data structure in which points are represented by a parametrized surface model, e.g., a collection of triangles. A patch organizes points into clusters of a certain size appropriate for 3D to 2D projections. This thesis focuses exclusively on coding geometry and colour for dynamic point clouds.

V-PCC

V-PCC breaks the input point cloud into patches based on vector similarity. These patches are then orthogonally projected onto a 2D plane, generating geometry and attribute images by assembling the 2D patches into an image [46, 72]. V-PCC can provide real-time decoding capabilities for virtual and augmented reality applications and immersive communications. It leverages existing and future video compression mechanisms and the broader video ecosystem, including hardware acceleration, transmission services, and infrastructure.

G-PCC

G-PCC [148] focuses on efficient lossless and lossy compression for various applications, including autonomous driving and 3D mapping that utilize LiDAR-generated point clouds. It includes several geometric-driven approaches within its framework. The main objective is to reduce the size of the point cloud data while keeping it as accurate as possible in terms of shapes and important details.

In summary, point cloud compression is an essential technique for managing the storage and transmission of large 3D data sets. A popular approach to point cloud compression involves using hierarchical data structures such as kd-trees and octrees.

However, the compression ratio must be balanced against the fidelity of the reconstructed data to ensure high-quality representation.

2.1.4 *Recent Point Cloud Compression Methods*

In recent years, machine learning methods, particularly deep learning, have been employed for point cloud compression [3, 141]. Autoencoders, a type of neural network, can be trained to encode and decode point cloud data, resulting in compact representations that minimize the reconstruction error [127]. Graph neural networks and Convolutional Neural Networks (CNNs) adapted for 3D data have been used to exploit spatial and attribute correlations for more efficient compression [149].

2.1.5 *Deep Learning for Point Cloud Compression*

While the application of deep learning for point cloud compression is still in its early stages, we can expect that the current pace of research will soon result in robust and effective methods [51]. Deep learning methods have shown an unparalleled ability to learn patterns and correlations in complex and high-dimensional data, making them potentially well-suited for 3D point cloud compression [8, 141]. One example of this is the application of CNNs to point cloud compression. CNNs can learn spatial hierarchies and local and global patterns in data, which could be highly useful in identifying and representing redundancies in point cloud data. Some recent work has begun to explore the use of CNNs for point cloud compression, such as PointCNN [94], and the results are promising. Generative models like Generative Adversarial Networks (GANs) [34] also have the potential for point cloud compression. A GANs could be trained to generate point clouds that resemble those in the training set. The latent vector used to create these point clouds could then serve as a compact representation of the original point cloud. This is an area of active research, and there is much to explore. Furthermore, the application of transformer-based architectures, such as the ones used in large language models [176], to point cloud compression could be another promising direction. Transformers have shown an unparalleled ability to model complex patterns and correlations in data, which could make them well-suited for this task. Finally, the idea of using machine learning not just for compression itself but also to optimize the various stages involved in compression schemes is interesting. This could involve using machine learning to predict the optimal parameters for a given compression scheme or choose the most effective one for a given point cloud.

2.1.6 *Neural Radiance Fields (NeRFs) in 3D Vision*

NeRFs are a new way to create 3D models from 2D images. They work by training a machine learning model to predict the light intensity at any point in a 2D image. This allows the model to generate new views of the object or scene from different angles. NeRFs are more efficient than other 3D modelling methods because they only store the information needed to predict the light intensity. This means they can represent complex objects and scenes in a much smaller file size. NeRFs have the potential to revolutionize 3D graphics. They could be used to create realistic 3D models of objects and scenes that would be impossible to create with traditional methods. They could also compress 3D models into a much smaller file size, making them easier to share and use. NeRFs are still a new technology, but they can potentially be very important in the future of 3D graphics [43].

2.1.7 *Discussion*

Point clouds can consist of billions of points and their associated attributes, making coding crucial for storage and transmission. Various coding approaches for point cloud coding exist in the literature. Point cloud representation and compression are critical for enabling efficient storage, transmission, and visualization of 3D data. Different methods offer trade-offs regarding compression efficiency, visual quality, and computational complexity. The choice of representation and compression methods depends on the application's specific requirements and constraints, such as the desired level of detail, the available processing power, and the importance of preserving the original data's fidelity. While there are different approaches to point cloud compression, they are still in their early stages. The need for efficient and real-time methods remains high.Therefore, several studies [55, 100] highlights the need for further research into methods that complement compression for point cloud delivery.

The need for content-aware approaches

Future research directions could explore the integration of content-aware approaches with compression methods to achieve better resource allocation and user experience. For instance, a content-aware compression algorithm could dynamically adjust the LoD based on the importance of specific point cloud features for the receiver's task. This could lead to more efficient transmission of point cloud data, while still preserving the receiver's ability to perceive critical information. However, it is important to acknowledge the limitations of content-aware approaches. One challenge lies in the computational overhead of content analysis can itself be a resource burden. Therefore, future research should also explore methods for lightweight content analysis to ensure the efficiency of content-based adaptation methods. Consequently, content-aware approaches are often very much application-tailored. A method that optimizes point cloud quality for virtual reality application might not be suitable for real-time autonomous driving application. Developing generalizable content-aware methods that can be adapted to different applications remains an open research gap.

2.2 POINT CLOUD ADAPTATION

Point cloud adaptation involves adjusting the properties of a point cloud, such as its resolution, level of detail, or attributes, to accommodate varying requirements or constraints. This can include bandwidth limitations, device capabilities, user preferences, or the specific application context. The following subsections will provide an overview of various approaches and methods. Regular video streaming adaptation will be discussed first, as it greatly influenced the adaptation of point cloud streaming. Then, we will explain edge computing and viewport considerations before diving into the state-of-the-art point cloud adaptation mechanisms.

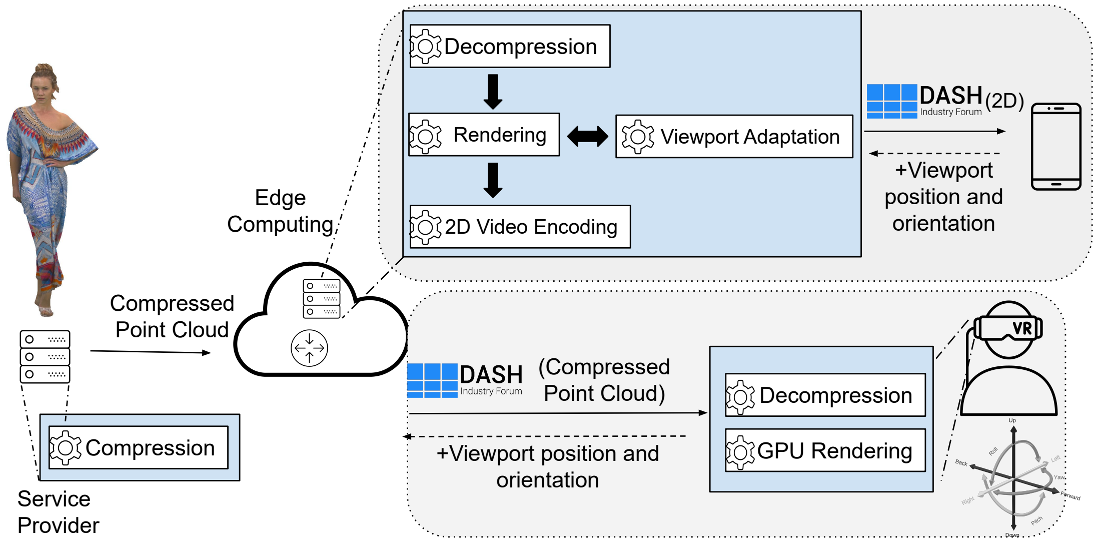


Figure 2.6: This figure overviews a point cloud streaming system and highlights the main mechanisms involved. The figure shows two point cloud streams. The upper stream is supported by edge computing to reduce the computation burden on mobile clients.

2.2.1 *Background*

Edge Computing

Edge computing refers to processing data geographically close to the source or end-users rather than relying solely on centralized data centers. In point cloud adaptation, edge computing can offload computationally intensive tasks, such as compression or rendering, to edge devices. This can reduce the latency and bandwidth requirements, allowing for more efficient and responsive streaming and processing. Edge computing can also facilitate collaborative and distributed processing, enabling low-latency adaptation. Several studies demonstrate the effectiveness and limitations of incorporating edge computing in point cloud streaming [29, 41, 109, 150, 183, 191, 198].

Viewport in 2D and 3D Perspectives

Leveraging the viewport is commonly used in 360° video streaming, involving the selective streaming of only the portion of the frame within the user's field of view [80, 151]. Similarly, in point cloud streaming, the technique can be extended to send only the points that fall within the user's view frustum [174]. This frustum represents the intersection of six planes determined by the user's view and projection matrices, as depicted in Figure 2.7. This approach involves calculating the six planes and evaluating each point to determine if it falls within the frustum, using view frustum culling [52].

Viewport prediction estimates the user's Field of View (FOV) or Region Of Interest (ROI) in a 3D scene based on their previous viewing behavior, head movements, or other contextual information. Accurate viewport prediction can enable more efficient point cloud adaptation by prioritizing the streaming and rendering of points likely to be visible to the user. Various methods have been proposed for viewport prediction, including machine learning approaches, such as Recurrent Neural Networks (RNNs) and reinforcement learning, as well as traditional heuristics and algorithms [53].

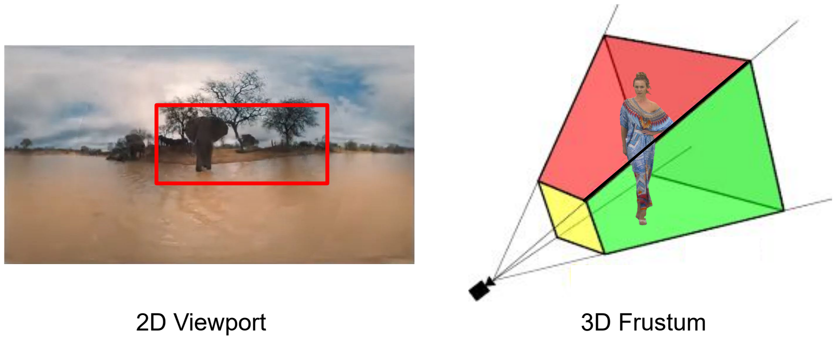


Figure 2.7: Field of view (FOV) in both 2D and 3D perspectives. Image source [37]

2.2.2 *Adaptive 2D Content Streaming*

Numerous previous studies [79, 85, 145, 178] have focused on enhancing the QoE in 2D media streaming systems by employing rule-based or heuristic-based approaches on the client side. These traditional adaptive bitrate approaches can be grouped into three classes based on the measured inputs: throughput-based, buffer-based, and hybrid.

Throughput-Based Approaches: In this category of heuristic-based Adaptive Bitrate Streaming (ABR) approaches, the media bitrate prediction relies solely on the estimated network throughput of the content client. The bitrate for the upcoming segment is determined based on the calculated estimated bandwidth, which is derived from the rate of the previously downloaded segment. This approach addresses the challenge of video rate variations [189]. Notable algorithms in this category include PANDA, which stands for "Probe AND Adapt", [95] and FESTIVE [74]. FESTIVE focuses on adapting the bitrate while prioritizing efficiency, fairness, and stability. It is also the first algorithm to consider balancing multiple bandwidth-sharing clients. The algorithm calculates the harmonic mean of the throughput measured during the download of the previous 20 segments and uses the result to determine a reference rate proportional to the bitrate. In cases where the buffer size is below the maximum, a download is scheduled immediately; otherwise, a randomized delay is introduced to mitigate start time biases. The evaluation of FESTIVE is based on simulated videos and throughput traces, demonstrating improvements in various aspects compared to commercial players at the time.

Similarly, PANDA is a client-side rate adaptation algorithm proposed by Xiaoqing et al. [95]. Prior to each segment download, the bandwidth is estimated and smoothed. The video bitrate is then selected, and the segment download is scheduled. PANDA's optimization goals include avoiding buffer underruns, ensuring high-quality smoothness, achieving high average quality, and promoting fairness. The algorithm's key focus is its probing technique, where the targeted average bitrate is gradually increased by $K * w$ per time unit. Here, K represents the probing convergence rate multiplied by the additive increase rate w. This probing mechanism explores the available channel capacity and directly addresses the problem of bitrate fluctuation. PANDA outperformed its competitors by approximately 75% in cases of instability and exhibited higher fairness. However, it was less responsive to bandwidth drops.

Buffer-Based Approaches: The QoE objectives of the following approaches are achieved by considering only the buffer occupancy of the video client. The aim is to ensure a smooth streaming experience and minimize stalling time by selecting an appropriate bitrate based on the buffer level [69]. In the buffer-based approaches, an algorithm proposed by the authors of this paper [69], the algorithm relies solely on the buffer level. This can lead to the user requesting a high-quality streaming experience as the buffer size approaches minimum levels. The algorithm includes different versions, namely BBA-0, BBA-1, BBA-2, and BBA-O, which map the buffer size to a corresponding video bitrate to enhance the quality. Each version operates within different zones of the mapping function. For instance, BBA-2 utilizes the risky zone, resulting in more frequent rebuffering than BBA-1, which operates only within the safe zone. BBA-2 also incorporates a capacity estimation on startups to achieve higher video rates. Evaluation results indicate a reduction in rebuffering by 10-20% compared to the default algorithm used by Netflix at the time [69]. Another notable buffer-based approach is the BOLA, which stands for Buffer Occupancy-based Lyapunov Algorithm, [154], which formulates the problem as a utility maximization problem considering the average bitrate and rebuffer duration as critical components. The simplest implementation, BOLA-BASIC, triggers the decision-making process before each download. However, this may result in stalling if the bandwidth suddenly drops. To address this, BOLA-FINITE allows for restarting a download with a different bitrate. The authors identified the potential for oscillation in BOLA-FINITE when the difference in download time is smaller than the segment duration. They proposed two solutions: capping the bitrate (BOLA-O) or allowing a higher bitrate level than sustainable (BOLA-U). These improvements aim to reduce freezing while maximizing viewing quality without the need to predict the current network conditions.

Hybrid ABR Approaches: The final traditional approach combines the two heuristic-based adaptive bitrate methods discussed earlier, incorporating both throughput and buffer size as state variables to further enhance the overall QoE. Notable examples of this approach include ELASTIC [35] and FastMPC [190]. ELASTIC addresses the fairness issues that arise when multiple video streaming clients coexist in the same network domain or stream the same media. To overcome this challenge, Luca et al. developed ELASTIC (fEedback Linearization Adaptive STreamIng Controller) as a hybrid client-side ABR algorithm that leverages control theory to provide solutions. The algorithm dynamically prefetches video resolution based on current network conditions, using a single controller on the user side to ensure smooth video streaming even with a small buffer. Unlike throughput-based approaches like PANDA and FESTIVE, ELASTIC improves fairness and channel utilization. In [190], FastMPC is introduced as a control theoretic mechanism on the client side to adapt the video quality to the available network conditions. The problem is formulated as a maximization problem of the QoE. This mathematical model enables flexible QoE optimization based on throughput and buffer size information. The deployment of the algorithm involves three steps: (1) predict the throughput using existing methods, (2) solve the optimization problem, and (3) apply the results and download the segment. However, throughput estimations may need to be more reliable in situations with limited network capabilities. The authors upgraded the model to RobustMPC to address this limitation by introducing throughput estimate boundaries and choosing the lower bound for the optimization step.

Another client-side ABR algorithm, QUETRA [186], leverages queuing theory by designing the Dynamic Adaptive Streaming over HTTP (DASH) client as an M/D/1/K queuing system with the assumption that the arriving segments follow a Poisson distribution. Throughput measurements are obtained using methods such as the last measurement, moving average, Exponential Moving Average (EMA), Gradient Adaptive EMA, Low Pass EMA, and Kaufman's Adaptive Moving Average (KAMA). It was found that EMA and moving average methods could increase the streamed video bitrate by up to 4.5%. However, other estimation methods with larger time frames were not successful. The measurement then determines the next segment's quality level to ensure the buffer occupancy level matches the buffer slack, representing the absolute difference between the actual buffer occupancy and the mean buffer occupancy. This approach improved measured QoE compared to the dash.js [136] default algorithm and BBA.

Lastly, we examined DYNAMIC [153], the current default algorithm of the dash.js project. DYNAMIC works in conjunction with the previously introduced BOLA algorithm. It was observed that BOLA does not download segments in the highest sustainable quality when the buffer level is low or empty, particularly at the start of a video stream or when seeking another position on the timeline. To address this, the authors introduced THROUGHPUT, which only downloads segments based on the measured throughput. The role of DYNAMIC in this system is to switch between both algorithms based on the client's buffer level and the bitrate chosen by the algorithms using switching rules. DYNAMIC achieves higher bitrate streaming and lower rebuffering time than a pure BOLA approach.

2.2.3 *Point Cloud Adaptation with Traditional Approaches*

Adaptive 2D video streaming methods have inspired similar approaches for point cloud streaming [58, 78, 169]. Adaptive 2D video streaming concepts, such as bitrate adaptation, adaptive chunking, and buffering, can be adapted and extended for point cloud streaming to enhance the user experience. For instance, the adaptive streaming methods used for 2D content, like DASH [42, 156] and Apple's HTTP Live Streaming (HLS) [40], can be applied to 3D content. These methods dynamically adjust video quality based on network conditions, user preferences, or device capabilities.

A variety of Point Cloud Compression (PCC) methods, such as the V-PCC encoder [46], that compress original point cloud data. By altering the quantization parameters, multiple versions of the data can be generated, each with a distinct bit rate and quality. A user can then access these compressed versions for rendering a 3D scene with 6DoF. This DASH-compliant approach offers dynamic adaptability, view awareness, and bandwidth efficiency. To address the fluctuating bandwidth requirements of streaming point cloud content, the proposed system utilizes a DASH-compliant Media Presentation Description (MPD) manifest explicitly designed for point cloud objects. Rather than employing a dedicated encoder for point cloud objects, they employ sampling methods to generate a range of quality variations. Subsequently, the point cloud object is fetched on a per-frame basis, resulting in a number of HTTP GET requests comparable to the frame rate. One limitation of this work is that it only supports a single point cloud object. This approach may give rise to significant challenges, particularly regarding network latency. In a subsequent work, Hosseini expanded upon their research by proposing rate adaptation methods for streaming multiple point cloud objects [57]. The algorithmic heuristics prioritize point cloud objects based on their visibility, distance from the camera, and camera view. Consequently, point cloud objects closer to the camera are assigned higher priority and transmitted in higher-quality representations. Conversely, objects farther away are assigned lower priority and transmitted in less demanding quality representations.

Van der Hooft et al. introduced PC-DASH, a framework for streaming scenes that comprise multiple point cloud objects [169]. They used PCC to create various quality versions of the objects and suggested several rate adaptation heuristics considering the users position and viewing angle.

2.2.4 *Point Cloud Adaptation with Machine Learning Methods*

In addition to traditional approaches, new machine learning models have become evident in recent years [75]. Utilizing large datasets during the learning phase holds the promise of self-tuning numerous parameters to achieve optimal results in various scenarios [19, 63, 145]. This section presents the current alternatives to traditional point cloud adaptive streaming approaches.

Machine learning methods, particularly reinforcement [164] and deep learning [44, 126], have shown promise in the area of point cloud adaptation [51, 181]. Neural networks can be trained to learn efficient representations of point cloud data, enabling adaptive compression and streaming [47, 51, 64]. Moreover, machine learning algorithms can be applied to predict user behaviour [49, 50, 92] and viewport prediction [90], or to optimize point cloud rendering based on device capabilities and user preferences [133, 170]. These methods can lead to more personalized and responsive point cloud experiences while minimizing the computational and bandwidth requirements [53].

Machine learning adaptive bitrate algorithms can be divided into two main categories. The first category includes predictive algorithms, which operate similarly to heuristic approaches by directly predicting streaming parameters to optimize the quality of experience. The second category includes augmenting approaches, which do not provide standalone results but are used with traditional approaches to enhance their performance. These approaches cannot be used independently and require integration with existing methods.

Predictive Approaches: The authors in this paper [110] have developed a Reinforcement Learning model called "Pensieve" to minimize suboptimal bitrate changes caused by inflexible algorithms. Starting from zero experience, the Pensieve model utilizes its own prediction results to optimize future decision-making during training by generating ABR models on the fly. The model focuses solely on video delivery quality without considering external network factors. For instance, if the video provides a high QoE without buffering, the reinforcement mechanism rewards the model; otherwise, it penalizes it. The fundamental architecture of Pensieve combines Long Short-Term Memory (LSTM) and CNNs architectures within a reinforcement learning model. To optimize chunk selection for the QoE formula, the authors have built an Actor-Critic Network that chooses the optimal bitrate based on the current client state. The state consists of sequential data, including the last 16 throughput measurements, download times, chunk sizes, instantaneous input data buffer size, remaining chunks, and the bitrate of the last chunk. Training is performed simultaneously on both the actor and critic networks, but only the actor network is deployed for testing.

Supervised learning approaches have been introduced to enhance algorithm performance in the adaptive video streaming domain. An example of this approach is in the paper SMASH [146], where the supervised technique utilizes classification methods with labelled data, representing the encoding of streaming chunks at different qualities and resolutions. SMASH is implemented as a model prediction of adaptive video streaming on the user side. Additionally, the bitrate prediction is achieved through learning from the output of nine different ABR algorithms, which vary between throughput-based, buffer-based, and hybrid algorithms. To collect input data, the streaming video is conducted over three different network traces, including 3G, 4G, and Wi-Fi. The dataset mainly consists of logged features, such as stalls, codec, and chunk index, that help predict the quality of the next selected segment. Various classifier models are selected for evaluation, including Logistic Regression (LR), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors (KNN), Decision Tree Classifier (DTC), Gaussian Naive Bayes (GNB), Ada Boost Classifier (ABC), Random Forest Classifier (RFC), and Multilayer Perceptron (MLP). After evaluating each model using a 10-fold cross-validation, the authors find that the RFC achieves the highest accuracy. The evaluation metrics used include bitrate, number of switches, number of stalls, and stall duration.

Augmenting Approaches: In Oboe [6], the authors take an interesting approach by augmenting existing algorithms instead of creating new ones. They optimize the runtime parameters of the algorithms based on the current network conditions. The approach consists of two processes. First, an offline process creates a config map by exploring different ABR configurations for a given throughput trace. The best configuration is saved in a file that maps network states to the calculated optimum configuration. This map is used in the online stage to choose the best parameter combination relative to the observed network state. The model continuously monitors for significant changes that warrant a parameter change and reconfigures accordingly. This approach was applied to BOLA [154], MPC [190], and Pensieve [110], demonstrating performance improvements of up to 24% for each model.

The ERUDITE paper [36] follows a similar direction by optimizing the parameters of models. Building on Oboe's principles, the authors extend the approach by leveraging data on bandwidth fluctuations and video features. Various MLP and CNNs architectures with different segment durations were implemented and evaluated. The CNNs model outperformed the MLP in terms of QoE. A "trigger module" was also introduced before the neural network to update the model parameters when necessary.

In another paper by Yan et al. [187], Fugu is introduced as a hybrid ABR model that combines a control technique similar to MPC with a Transmission Time Predictor (TTP). The MPC controller selects the bitrate linearly based on the predicted throughput. At the same time, Fugu replaces the throughput predictor with TTP, which predicts the time for file download as a statistical variable. Based on the last eight segment sizes, the predicted transmission time is passed to the MPC controller for bitrate selection. Fugu's TTP model is trained using classical supervised learning. The training takes place in situ, in a real environment, with daily retraining. It utilizes a large-scale streaming environment accumulated over 14.2 years of video streaming time. The authors conducted experiments on Puffer, their own TV streaming platform, revealing significant variations in results based on streaming time.

Park et al. presented a utility-based rate adaptation heuristic for point cloud content in augmented reality, incorporating network and user adaptation [128]. Their system dynamically adjusts the LoD of point cloud objects based on their proximity to the user's location. They proposed a greedy algorithm to optimize rate and utility that allocates bits among different tiles across multiple objects. Specifically, the system reduces bandwidth requirements by decreasing the level of detail of objects based on their position and distance from the user's viewport. In order to minimize latency, they introduced a window-based buffer that enables swift response to user interactions. The evaluation results demonstrate that their proposed heuristic offers enhanced utility and user experience when dealing with varying throughput-constrained networks, surpassing existing video streaming approaches.

Qian et al. introduced a system known as Nebula, which offers a novel approach to point cloud video streaming on regular 2D video-capable smartphones to alleviate the computational load [133]. Their approach involves offloading resource-intensive operations to a remote render server and implementing rate adaptation mechanisms to optimize video quality based on network conditions. To minimize the photon-to-motion latency, they propose a viewport prediction mechanism and the concept of a mega-viewport. They also present various optimization methods to reduce the perceived latency, dynamically adjust to varying network bandwidth, and optimize resource utilization while ensuring a high QoE.

In their research, Hoog et al. investigated the viability of compressing point cloud data for sharing across inland vessels while preserving usable point cloud quality for situational awareness [56]. Their findings revealed that lossless compression with BZip2 reduced the point cloud size by 50% without sacrificing any information. Additionally, they observed that lossy compression using Draco [45] achieved a point cloud size of 25%

2.2.5 *Discussion*

Point clouds can consist of billions of points and their associated attributes, making coding and compression crucial for transmission. Various approaches for point cloud coding exist in the literature. Point cloud adaptation is essential for ensuring efficient and high-quality user experiences across various devices, network conditions, and applications, including XR and autonomous driving. Methods inspired by adaptive 2D video streaming, edge computing, and viewport prediction can be combined and extended for point cloud adaptation. The literature shows that incorporation of machine learning approaches offers the potential for further improvements in adaptability and responsiveness, enabling more effective resource utilization and improved user experiences.

Research gap in advancing machine learning for point cloud adaptation

However, achieving a balance between quality and resource allocation during streaming for the intended application of the point cloud remains a challenge. Existing methods often rely on pre-defined rules or user preferences to adjust the streaming parameters. The integration of machine learning offers a potential path toward more intelligent and dynamic point cloud adaptation. However, a significant research gap exists in developing efficient learning models for real-time point cloud adaptation. Challenges include the need for large, labelled datasets specifically designed for point cloud adaptation tasks. Additionally, designing lightweight and computationally efficient learning models is crucial to avoid introducing latency in the streaming process. Future research efforts should focus on addressing these challenges. This could involve exploring techniques like deep learning to leverage existing knowledge from related domains like 3D object recognition. By closing this research gap, machine learning-driven point cloud adaptation has the potential to significantly improve streaming experiences. This approach could ensure high quality content delivery while optimizing resource utilization across diverse applications.

2.3 POINT CLOUD QUALITY ASSESSMENT

Point cloud content can provide the required 6DoF for truly immersive media. However, achieving 6DoF is limited by the current bandwidth limitations of best-effort networks. Therefore, recent efforts have focused on the efficient delivery of point clouds using a combination of compression and adaptive streaming mechanisms. The impact of these mechanisms on the user-perceived quality needs to be accurately evaluated. Assessing the quality of point cloud content is crucial for evaluating the effectiveness of compression, adaptation, and rendering mechanisms and optimizing user experiences [54]. Point cloud quality assessment involves evaluating the visual quality or other performance metrics of point cloud content, considering the specific application context and user requirements [82, 173]. This section provides an overview of approaches and concepts related to point cloud quality assessment.

2.3.1 *Background*

QoS and QoE

In evaluating the quality of a content streaming service, it is important to understand the difference between QoS and QoE.

QoS focuses on ensuring uninterrupted multimedia transmission, which is critically important when network capacity is insufficient [106]. This is particularly relevant for real-time multimedia transmissions like video conferencing, Internet telephony, IPTV, and online gaming [155]. While specific applications may prioritize minimal latency and reliable response time, others may demand high-quality visual content. For instance, real-time conversational services such as voice calls and live-streaming are highly sensitive to delay but somewhat tolerant of errors. Services like real-time gaming are intolerant to errors, whereas audio and video streaming services are error-tolerant and have less strict delay requirements. Interactive services function based on request-response patterns and allow prioritization based on the end-user or service type. Services like email notifications are the least sensitive to delay; for these, best-effort data delivery is acceptable. QoS is a technical measure of the performance of a network or system and includes metrics such as latency, packet loss, and bit rate [131]. Service providers often use it to monitor and optimize network performance.

QoE refers to the user's subjective experience and considers factors such as the content itself, visual and audio quality, and interactivity [59, 157]. The EU Qualinet community defines QoE as "the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the users personality and current state" [24]. QoE is a more holistic measure of the overall user experience. QoS contributes to QoE, see Figure 2.8. While QoS can provide helpful information about the technical performance of a network, it does not necessarily reflect the user's experience. Therefore, it is important to consider both QoS and QoE when evaluating the quality of a media streaming service [20]. In the context of point cloud quality assessment, the end-users satisfaction is directly tied to the perceived quality of the content received on the client side. Numerous research studies have utilized QoE as a primary metric for evaluating the performance of their proposed methods [91, 93, 129, 161, 173, 174, 194, 198]. Hence, QoE plays a pivotal role in evaluating the quality of the delivered service. Therefore, both QoS and QoE metrics can be employed to evaluate the performance of compression, adaptation, and rendering methods, as well as to guide the optimization of streaming mechanisms [17].

QoE evaluation methods can be categorized into subjective, objective, and hybrid methods, as discussed by Maia et al. [106]. Subjective methods are based on ITU standards [116] and involve experts rating the quality of delivered video content. These ratings are typically provided using metrics like Mean Opinion Score (MOS) or Degradation MOS (DMOS). MOS, a subjective measure serving as a simpler alternative to Peak Signal-to-Noise Ratio (PSNR), calculates multimedia visual quality by considering network conditions and specific traffic traits [172]. MOS and DMOS are usually computed as averages of the collected ratings for each content piece. However, this subjective approach is often considered inefficient due to its reliance on limited observers, distortions, and high costs.

In contrast, objective quality assessment methods evaluate factors through QoS metrics such as packet loss rate, latency, jitter, bitrate, and frame rate, along with external variables like content type, viewer demographics, and device type [106]. Objective models commonly use metrics like Moving Picture Quality Metric (MPQM), Perceptual Video Quality Measure (PVQM), and Visual Signal-to-Noise Ratio (VSNR) [118].

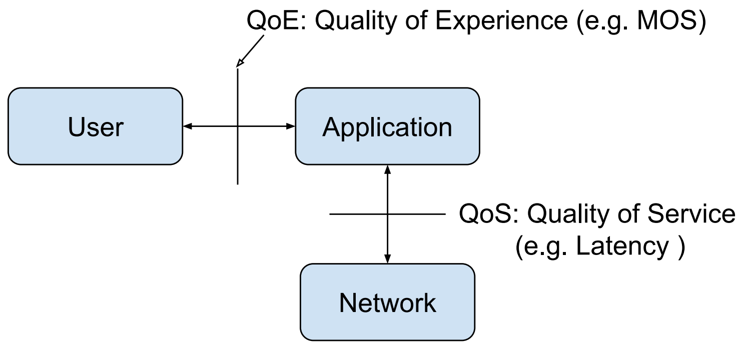


Figure 2.8: Comparing QoS and QoE. QoS is the network's contribution to QoE.

2.3.2 *Objective Evaluation for Point Cloud Content*

Various objective metrics have been employed in the past to measure the quality of point clouds. It is important to differentiate between assessing the quality of a point cloud object relative to a modified version of the same object and the quality of the displayed field of view, i.e., what a user sees when using a head-mounted display.

In order to compare the quality of modified point clouds, MPEG has suggested two PSNR-based metrics, known as point-to-point and point-to-plane geometry distortion metrics [165]. The former computes the mean square error (MSE) between the original and reconstructed points for both geometry and colour. The latter determines the MSE between the surface plane and reconstructed points. PSNR values are derived from the volume resolution for geometry and colour depths for each colour channel. While these metrics are pertinent for evaluating the effectiveness of compression methods for volumetric media, they do not provide insight into how a user visually perceives the related point cloud object(s) from a particular viewpoint or angle. Established metrics for traditional video streaming have recently been used to gauge the visual quality of displayed point cloud content relative to a certain standard, such as uncompressed point cloud objects. These metrics include the PSNR, the structured similarity index (SSIM) [159], and the multiscale SSIM [7], among others.

Despite providing insight into the visual quality of displayed content, these metrics factor in the background of the viewed scenes. The background contributes less to perceived quality, as users are expected to concentrate mainly on foreground objects. One study explored background removal for images generated from point cloud data, utilizing a MATLAB-based tool for assisted removal [163]. Although this feature is valid, it incurs a substantial computational cost when video is involved.

2.3.3 *Subjective Evaluation for Point Cloud Content*

The subjective quality of 3D content has been studied less extensively than that of 2D video. A recent summary of previous research on subjective evaluation and objective metrics for point clouds is provided by Dumic et al. [39]. Furthermore, extensive surveys exist on using machine learning models for QoE assessment and prediction [4, 81]. In the following, we describe selected works on QoE assessment in more detail.

Various subjective evaluation methods can be broadly classified into single-stimulus and double-stimulus methods based on how the evaluated content is presented to the observer [159]. Single stimulus methods show the observer a single version of the content and ask them to rate its quality on a scale or make a binary decision such as "Yes" or "No". For example, in a single stimulus method, an observer might be shown an image and asked to rate its quality on a scale of 1 to 5. Double stimulus methods show the observer two versions of the content and ask them to compare the quality of the two versions. For example, an observer might be shown two versions of an image and asked to indicate which one is of higher quality. Double stimulus methods are also known as side-by-side or paired comparison methods.

2.3.4 *Motion-to-Photon Latency*

Motion-to-photon latency is a critical metric in virtual and augmented reality applications. It measures the time it takes for a user’s motion or input to be reflected in the rendered content. High motion-to-photon latency can lead to degraded user

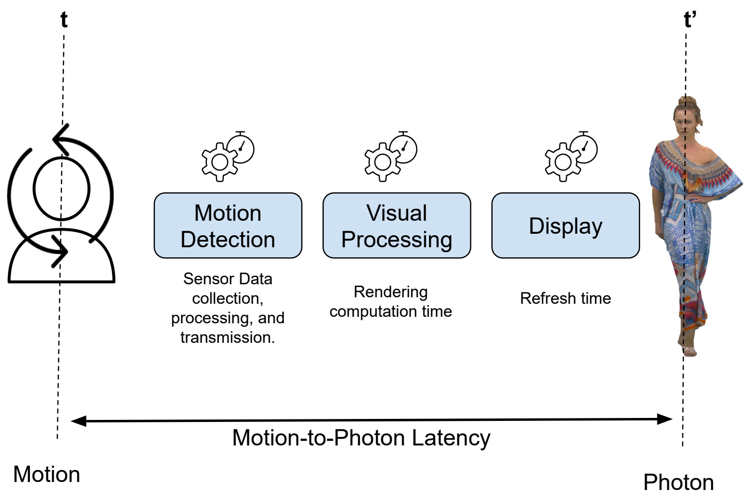


Figure 2.9: The primary contributors to motion-to-photon latency in the head-mounted display setup.

experiences, such as motion sickness or disorientation. In the context of point cloud quality assessment, motion-to-photon latency can be used to evaluate the responsiveness and interactivity of point cloud rendering and adaptation methods, particularly in immersive VR/Augmented Reality (AR) applications [199].

2.3.5 *Point Cloud Quality Assessment with Machine Learning*

Machine learning methods have been increasingly employed for point cloud quality assessment because they can learn complex relationships between point cloud data and perceived quality. Supervised learning approaches, such as deep learning models, can be trained on labeled datasets to predict quality scores or classify point cloud data based on visual quality, accuracy, or other performance metrics. These machine learning models can then be used to guide the optimization of compression, adaptation, and rendering methods and to evaluate the quality of point cloud data in various application contexts.

2.3.6 *Discussion*

Point cloud quality assessment is essential for evaluating and optimizing the performance of point cloud compression, adaptation, and rendering methods and for ensuring high-quality user experiences. Both objective QoS and subjective QoE metrics can be employed to assess the quality of point cloud data, depending on the specific application context and user requirements.

Machine learning approaches offer promising solutions for point cloud quality assessment. They can learn complex relationships between point cloud data and perceived quality, enabling more accurate and efficient quality evaluation and optimization.

In the literature, point cloud quality assessment methods are still limited and need to be integrated into the point cloud streaming process. Therefore, further research is required to understand the impact of compression on the overall quality of point cloud perception, in addition to the necessary knowledge of resource demands for each quality setting. This is a prerequisite for optimizing point cloud streaming mechanisms. Machine learning methods offer promising advances for point cloud quality assessment. They can learn complex relationships between point cloud data and perceived quality, enabling more accurate and efficient quality evaluation. While existing machine learning models excel at assessing pre-recorded point cloud quality, assessing the generalisability of machine learning quality models in the dynamic nature of streaming point clouds is still required. The point cloud streaming process requires real-time quality assessment that can process point clouds incrementally, adapting their evaluation as new data arrives. Additionally, computational efficiency becomes paramount to avoid introducing latency into the streaming process. Therefore, future research should focus on developing lightweight and real-time capable machine learning models for point cloud streaming. This could involve exploring learning approaches to leverage pre-trained models and investigating efficient processing of streaming data. Furthermore, developing new metrics that accurately capture user experience in the context of point cloud streaming is crucial for effective model training and evaluation. Addressing this gap can unlock the full potential of machine learning for ensuring an adaptive and real-time point cloud streaming process.

Research gap in point cloud quality assessment

2.4 SUMMARY

This chapter has provided a background, concepts, and methods related to point cloud representation, compression, adaptation, and quality assessment. Future research should focus on integrating machine learning methods for adaptive point cloud processing, addressing challenges such as quality assessment, real-time adaptation and efficient resource utilization.

3 STREAMING.KOM: EXPERIMENTAL TOOL FOR POINT CLOUD STREAMING

Experimental platforms have become indispensable research tools because they provide a controlled and customizable way to test newly proposed mechanisms. The efficiency of the researchers' proposed mechanisms can be investigated and validated under various settings and scenarios. These platforms also provide a foundation for reproducibility, enabling other researchers to build upon existing work and expand the knowledge base. Experimental platforms are helpful in the context of point cloud streaming for investigating innovative adaptive streaming mechanisms and evaluating the quality of compressed point cloud data. Researchers can replicate various network situations and Quality of Service (QoS) setups. The availability of such an environment is helpful for researchers as it can be controlled and allows them to evaluate the effects of various adaption mechanisms on streaming performance. However, there is currently no open-source codebase that can be used for this purpose.

This chapter introduces a tool, Streaming. KOM, to close the gap in open-source platforms for streaming point cloud content. This tool expands the capabilities of the Dynamic Adaptive Streaming over HTTP (DASH) protocol to enable point cloud streaming. Additionally, it utilizes the Unity game engine as a powerful rendering platform for visualizing the streamed point cloud data. The Streaming.KOM tool is intended to stream point cloud content in a controlled manner, allowing empirical bandwidth measurements at runtime. These measurements help researchers understand specific bandwidth overhead and other critical factors affecting the Quality of Experience (QoE) of point cloud content streaming, thereby pinpointing areas for improvement. This tool provides an interface for the community to implement and assess their suggested mechanisms. Streaming.KOM is made accessible as an open source[1] for the betterment of the community and to promote further investigation into point cloud quality and adaptive point cloud streaming mechanisms.

This chapter presents Streaming.KOM is a tool that expands the DASH framework for point cloud streaming. The contribution of this work can be summarized as follows. Firstly, designing a platform for point cloud streaming based on the Unity and DASH methods, providing a platform to evaluate the performance of streaming mechanisms. To our knowledge, this is the first experimental platform that combines both methods for point cloud streaming. Secondly, open-sourcing the platform code makes it readily available for other researchers to use and adapt for their experiments. This is a relevant contribution to the field. According to our information, it is the only open-source point cloud experimental streaming platform currently available.

1 `https://github.com/yaseenit/Streaming.KOM`

This contribution allows for a better understanding of the performance characteristics of point cloud streaming and can help to establish more effective streaming mechanisms in the future.

3.1 CONCEPTUAL OVERVIEW OF STREAMING.KOM

The Streaming.KOM tool is a client-server-based platform that enables dynamic adaptive streaming of point cloud data over a network by combining the DASH standard with the Unity game engine to visualize the point cloud data on the client side. Streaming.KOM is designed to enable empirical bandwidth measurements at runtime, facilitating a better understanding of specific bandwidth overhead and other critical factors that affect the QoE of point cloud content streaming. Additionally, it provides a controllable and configurable testing environment for point cloud streaming experiments. It allows researchers to introduce various network impairments and QoS configurations during experiments and evaluate the streaming's performance under different circumstances. Figure 3.1 provides a conceptual overview of the architecture of the tool, consisting of three main components: the DASH manifest generator, the server, and the client.

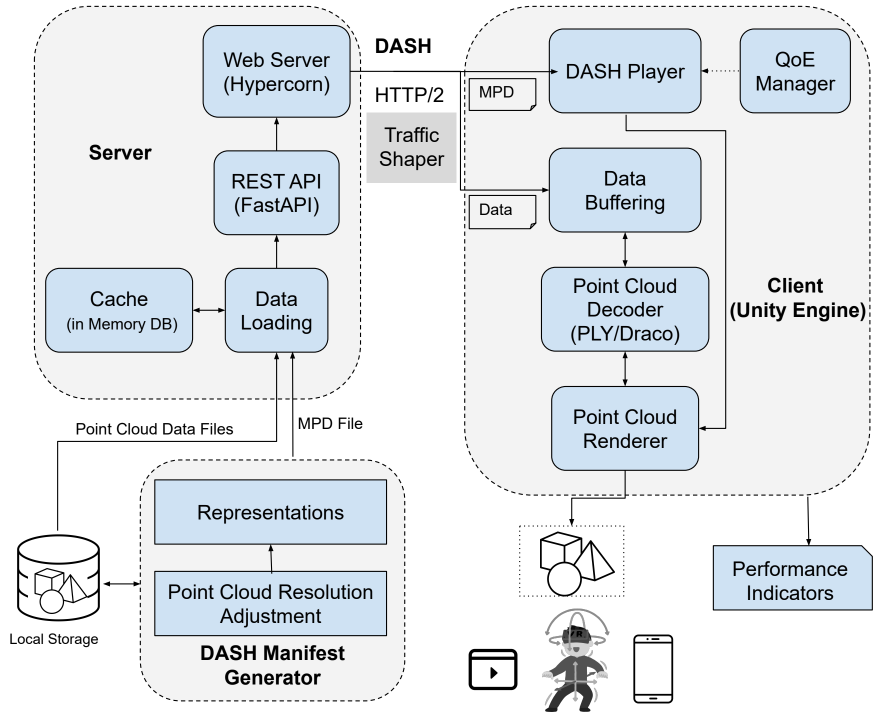


Figure 3.1: Conceptual overview of the Streaming.KOM architecture.

The DASH manifest generator component is responsible for reading raw point cloud data from local storage, compressing it using a compression algorithm, and converting it into DASH segments. These segments are then used to generate a Media Media Presentation Description (MPD) file, which describes the different bitrates and qualities of the point cloud data available for streaming.

The server component of the Streaming.KOM stores the compressed point cloud data and serves it over HTTP using the DASH protocol. A traffic shaper is implemented in the tool to enable researchers to introduce various network impairments and QoS configurations during streaming experiments. The traffic shaper sits between the client and the server, allowing researchers to simulate different network conditions and evaluate the performance of the point cloud streaming under various circumstances.

The client component of the Streaming.KOM is a Unity-based client that retrieves DASH segments from the server, decodes and renders the point cloud data, and adapts to changing network conditions using a QoE manager. The client incorporates several subcomponents, including a DASH player for handling the streaming of DASH segment, a data buffering module for managing data buffering to prevent playback interruptions, a point cloud decoder for decoding the compressed point cloud data, and a point cloud renderer for rendering the decoded point cloud data in the Unity game engine.

Overall, the Streaming.KOM is intended to offer a flexible and adaptable platform for evaluating the efficacy of newly developed mechanisms for streaming point cloud data. In the following section, we comprehensively describe the tool's implementation and the associated technologies.

3.2 TOOL DESIGN AND IMPLEMENTATION

In this section, we provide details of the implementation of the Streaming.KOM tool components, including the server, DASH Manifest Generator, and client.

3.2.1 *Server Implementation*

The server component of the Streaming.KOM is implemented using Python and is responsible for managing the storage and delivery of compressed point cloud data. It also serves as a bridge between the client and the compressed point cloud data, receiving requests from the client for specific DASH segments and sending those segments to the client in response. It is implemented using the open-source Hypercorn web server software, commonly used for HTTP-based content delivery. The server leverages FastAPI, a Python web framework, to build APIs that enable communication between the server and the Unity-based client. To optimize I/O operations, the server employs an in-memory database as a cache to speed up data retrieval from disks.

3.2.2 *Point Cloud DASH Manifest Generator*

The DASH Manifest Generator component converts the raw point cloud data into different quality DASH segments. The DASH segments are used for adaptive streaming, where only the necessary segments are streamed based on the client's requests and network conditions.

The generator utilizes the Open3D Python library to convert raw point cloud data into the Polygon File Format (PLY) if needed. Once the data is in PLY format, it can be compressed into different levels. The compression level can be adjusted based on the desired trade-off between visual quality and file size. Higher compression results in smaller file sizes but potentially lower visual quality. In Streaming.KOM, the Draco compression library developed by Google is utilized to compress the data with varying qualities and bit rates. Draco provides high compression ratios while maintaining the visual quality of the point cloud data with reasonable compression time. Once the PLY files have been compressed, the DASH Manifest Generator groups the compressed files into DASH segments with varying qualities and bit rates. The ZIP format packages the segments for a specific duration, typically 2 seconds. The DASH Manifest Generator generates the MPEG-DASH manifest file (MPD), which contains essential metadata for the client to request and receive the appropriate DASH segments. The manifest file provides the client with information about the available qualities and bit rates of the point cloud data, enabling the client to choose the appropriate segments based on current network conditions, device capabilities, and desired quality of experience. An example of a DASH manifest file for point cloud media content is shown in Listing 1, showing that a given point cloud has multiple representations with varying bitrates.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD type="static" minBufferTime="1.0" mediaPresentationDuration="20.0">
        <BaseURL>D:/Datasets/dash/ow1ii\_dancer\_360</BaseURL>
        <Period id="0" start="PT0.0S" end="PT1.0S" duration="PT1.0S">
                <AdaptationSet mimeType="pointcloud/plY+zip" id="0">
                        <Representation id="1" bandwidth="2476617K">
                                <BaseURL>qp7/0.ply.zip</BaseURL>
                        </Representation>
                        <Representation id="2" bandwidth="9614801K">
                                <BaseURL>qp8/0.ply.zip</BaseURL>
                        </Representation>
                        <Representation id="3" bandwidth="36467553K">
                                <BaseURL>qp9/0.ply.zip</BaseURL>
                        </Representation>
                        <Representation id="4" bandwidth="135473006K">
                                <BaseURL>qp10/0.ply.zip</BaseURL>
                        </Representation>
                </AdaptationSet>
        </Period>
```

Listing 1: Example of a DASH manifest file for point cloud media content illustrating that a given point cloud has multiple representations with varying bitrates.

3.2.3 *Client Implementation*

The DASH segments are requested and received from the server by the Streaming.KOM client is also in charge of decoding them and presenting the point cloud data. It is created utilizing the C# programming language and the Unity game engine. The client includes the following subcomponents:

DASH Player. Based on the current network conditions, the DASH player requests DASH segments from the server. It uses the UnityWebRequest API to request the DASH segments over HTTP. The DASH player also adapts to changing network conditions using the QoE manager, which determines the appropriate quality level of DASH segments to request based on the current network bandwidth.

Point Cloud Decoder. The point cloud decoder decodes the compressed point cloud data received from the server into a format that the Unity game engine can render. It uses the Draco library to decode the point cloud data. First, the compressed frames of a given segment are extracted by decompressing them from the ZIP format.

Point Cloud Renderer. The point cloud renderer renders the point cloud data in the Unity game engine. It uses the Point Cloud Visualization plugin for Unity.

QoE Manager. The QoE manager is responsible for adapting to changing network conditions by requesting appropriate quality levels of DASH segments based on the current network bandwidth. It uses the buffer occupancy and network bandwidth measurements to estimate the current network conditions and determine the appropriate quality level of DASH segments to request. Notably, Streaming.KOM provides flexibility in easily implementing different adaptation policies. Hence, researchers can customize and replace the default policy according to their research needs.

Overall, the Streaming.KOM client is built to request and receive point cloud data over a network, utilizing a combination of Unity and the Draco compression library in its implementation.

3.3 CONFIGURATION AND CUSTOMIZATION

One of the advantages of the Streaming.KOM tool is its flexibility and configurability, thus, researchers can customize the tool according to their specific requirements. Streaming.KOM provides various configuration options that can be adjusted to fit the needs of specific experiments. In this section, we describe some key configuration options that can be customized in the Streaming.KOM tool environment.

Network and QoS Configuration.

The Streaming.KOM tool allows researchers to introduce various network impairments during experiments to evaluate the streaming's performance under different circumstances. This feature enables researchers to gain insights into the streaming's performance and identify areas for improvement under various network conditions. For example, the tool can be configured to simulate bandwidth limitations, latency, and packet loss, enabling researchers to evaluate the streaming's performance under realistic network conditions and identify areas for improvement. To test the impact of bandwidth impairments in a real-world situation, the server can be run in a docker container on a Linux system, and the docker-tc tool[2] can be used to manipulate the outgoing traffic. The tool limits bandwidth, adds delays, randomly drops packets, and manipulates packets. A command-line interface can control the traffic between the client and the server. Figure 3.2 shows the integration of the Streaming.KOM tool with a traffic shaper tool using Docker containers.

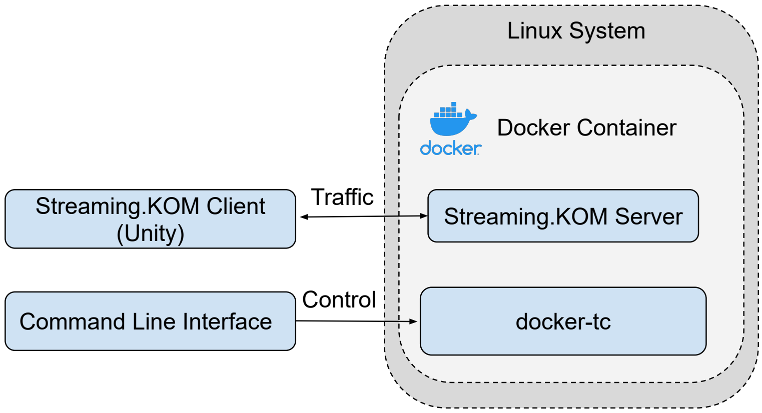


Figure 3.2: Integration of Streaming.KOM client and server with Docker container-based traffic shaper for evaluating system performance under different network conditions.

DASH Configuration The tool provides various DASH configuration options that can be customized according to the requirements of the experiment. These options include the segment duration, the number of segments per representation, and the representation bitrate, which can be adjusted using different compression quantization levels and object rates per segment.

Unity Configuration The Streaming.KOM test is built using the Unity game engine; hence, it provides various configuration options for Unity that can be customized to suit the experiment's requirements. These options include the rendering quality, the graphics settings, and the point cloud shaders.

Customizing Adaptation Mechanisms The tool's customizability allows researchers to implement their adaptation mechanisms to replace the de-

2 https://github.com/lukaszlach/docker-tc

fault one according to their specific research needs. Researchers can implement different policies for buffer management, bitrate adaptation, and segment selection. This feature allows researchers to experiment with new mechanisms and evaluate their impact on streaming's overall performance. For instance, the tool offers multiple options for adaptive bitrate policies, including the lowest bandwidth policy, which continuously streams at the lowest possible quality based on the available bandwidth, the highest bandwidth policy, which continuously streams at the highest possible quality based on the available bandwidth, and the available bandwidth policy, which continuously streams at the highest possible quality that can be downloaded promptly.

In summary, the Streaming.KOM system provides a range of configuration options that can be customized to fit the experiment's needs. This flexibility and configurability enable researchers to evaluate the system's performance under different conditions. Therefore, researchers can conduct experiments and measure results more effectively.

3.4 PERFORMANCE INDICATORS

In order to evaluate the performance of the point cloud streaming, several performance indicators can be measured during experiments. These indicators provide insights into the streaming's performance and can be used to identify areas for improvement. The following sections describe the different performance indicators measured during the experiments.

- **Throughput:** refers to the amount of data transmitted per unit of time, measured in bits per second (bps). Throughput is a critical performance indicator as it can affect the quality of the point cloud representation.
- **Buffer Occupancy:** refers to the amount of data stored in the client's buffer. Buffer occupancy can determine the time required to accurately render the point cloud representation.

Using these performance indicators, a series of experiments can be conducted to evaluate a proposed streaming mechanism under various network conditions and configurations.

3.5 DISCUSSION AND CONCLUSION

In this section, the limitations of the proposed tool, potential future research directions, and summary of the contributions of Streaming are discussed.KOM tool are discussed.

First, it is unavoidable to highlight the limitations of the proposed tool. A limitation of Streaming.KOM tool currently supports only the Draco compression library for point cloud data, which may be insufficient for specific experiments. Therefore, future work could incorporate other compression algorithms such as Geometry codec based Point Cloud Compression (G-PCC) and Video codec based Point Cloud Compression (V-PCC) to address this limitation. Another limitation of the tool is that it only supports streaming over a local network. As more applications move to the cloud, there is a growing need for a cloud-based point cloud streaming tool. Thus, future work can focus on developing a cloud-based version of Streaming.KOM tool that supports streaming over the internet and enabling edge computing.

Moreover, the Streaming.KOM tool does not include QoE as a performance indicator, considering factors such as visual quality, smoothness, and interactivity. These are crucial performance indicators for point cloud streaming. Future work can focus on developing a QoE model to measure the overall quality of the point cloud streaming while considering factors such as visual quality and interactivity.

In summary, Streaming.KOM tool provides a controllable and configurable testing environment for point cloud streaming experiments. It also leverages the DASH protocol and Unity game engine methods to enable dynamic adaptive streaming of point cloud data, enabling empirical bandwidth and latency measurements at runtime. While the tool has some limitations, it provides a foundation for future research in point cloud streaming. We hope to encourage further research and development in this field by offering the tool as open-source.

4

EVALUATING AND MODELLING THE QUALITY OF POINT CLOUD DATA

4.1 INTRODUCTION

This chapter aims to understand how adaptation variables, such as quantization levels, distances to the camera, and frame rates, impact the perceived quality of point cloud content. Compressing and adapting point cloud data can result in various forms of distortion, including changes in the number of points, their positions, and colours [165]. These distortions can have a negative impact on the user's perception and experience of the point cloud content. Therefore, it is crucial to have an accurate Quality of Experience (QoE) model in order to evaluate the quality of the compressed point cloud data [201]. This model can help to identify the best compression algorithms and parameters while adapting the content to changing conditions. However, developing a reliable QoE model is still challenging due to the variety of distortions introduced by the compression and adaptation processes [100]. The six Degrees of Freedom (6DoF) viewing process also makes traditional objective quality evaluation metrics inapplicable to point cloud video quality assessment [182]. Despite the increasing popularity of point cloud content streaming, there is limited research on QoE models for this type of streaming, making it challenging to optimize the performance of the streaming systems. Given that point clouds are often used in real-time applications, having an efficient and fast QoE model is crucial.

Challenges in the evaluation of point cloud content

As illustrated in Figure 4.1, this chapter explores two research goals. We investigate the first research goal (RG1): *Investigating the impact of compression-related distortions and reduced frame rate on the quality of point cloud objects*. This research will analyse the specific types of distortions introduced by compression and how they influence the user visual perception. Additionally, we will examine the impact of lower frame rates, a common compression technique, on the user experience. The second research goal: *Investigating the correlation between quality and resource demands, with the objective of developing a predictive model for evaluating the quality of point cloud sequences*. By addressing both perceptual impact (RG1) and resource allocation (RG2), this chapter aims to pave the way for creating a method for content-aware point cloud streams that achieves the intended application of the point cloud content while minimizing resource demands, as we will see in the next chapter.

Meeting users' expectations requires understanding the factors influencing QoE and efficiently managing resources to optimize video quality. Understanding how adaptations affect the perceived quality of point cloud content is crucial for effectively adapting such content to dynamically changing conditions during streaming. We conduct a user study to gather valuable insights and data to achieve this. Our study results will provide important information regarding the effectiveness of different state-of-the-art compression mechanisms in maintaining the quality of point cloud data. Moreover, the study will help us understand the trade-offs between resource requirements and quality levels, which is essential for developing mechanisms that can adapt the point cloud content to dynamic conditions and device capabilities through mechanism transitions, contributing to shaping the development of an effective adaptation mechanism and streamlining transitions between various mechanisms. Mechanism transitions have been widely applied in communication systems to meet quality requirements [14, 138]. This concept has been utilized to address a range of problems, such as ensuring quality requirements and handling unpredictable network conditions [76, 80, 103, 104, 137, 179]. The collected data of the user study will be leveraged to construct a QoE model. This model will serve as a predictive tool for assessing the perceived quality of point cloud sequences, enabling the adaptation of point cloud streaming mechanisms accordingly.

Challenges in the evaluation of point cloud content

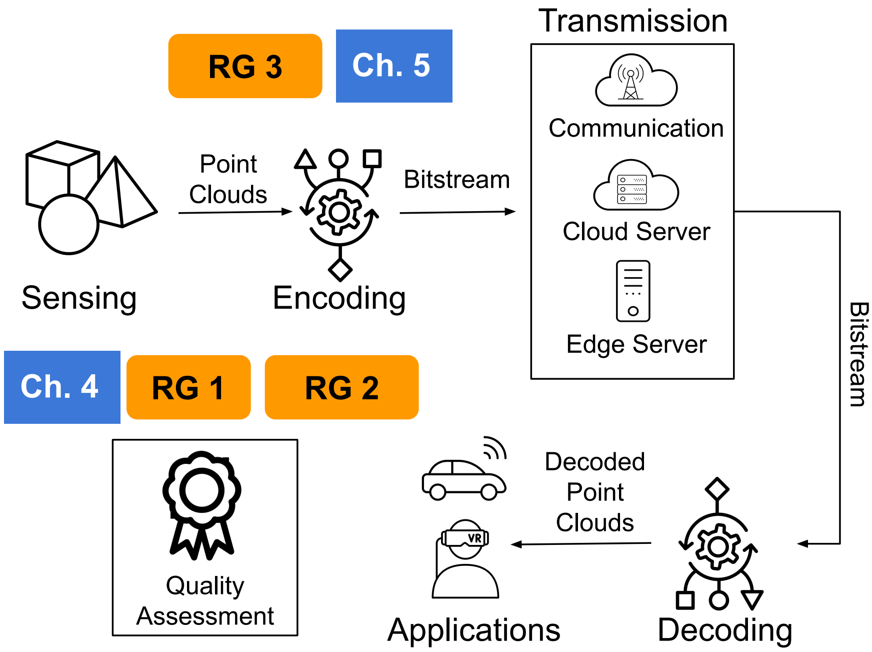


Figure 4.1: Point cloud streaming pipeline from the creation of content via sensing to the consumption by users in an application. The quality assessment part encompasses the first and second research goals, which will be addressed in this chapter.

This chapter explores the subjective perception of quality in compressed point cloud sequences, examining the influence of adaptation variables like quantization level, frame rate, and camera distance on perceived quality. We utilized two dynamic human figure point cloud sequences and evaluated their perceived quality through subjective user study to gather users' opinions on the compressed point cloud quality. We considered five levels of compression degradation, three different camera distances, and three different frame rates. To attain these different compression levels, we employed two state-of-the-art point cloud compression mechanisms, Video codec based Point Cloud Compression (V-PCC) by MPEG [48] and Draco by Google [45]. The Mean Opinion Score (MOS), usually referred to as MOS score, was used to measure the overall video quality. In accordance with the International Telecommunications Union (ITU-T) standards [168], a five-point Absolute Category Rating (ACR) scale was used, with a range from 1 for bad quality to 5 for excellent quality [15]. The resulting MOS scores refer to a numerical measure of the human-judged overall quality of video. Our investigation explored how different quality levels in the point cloud affect resource requirements during streaming, e.g., bandwidth usage, and latency. Furthermore, we develop models for predicting the QoE of point cloud sequences using various machine learning algorithms. We assess their accuracy by comparing the prediction results to the MOS obtained from our subjective study. The results suggest that the proposed models can accurately predict and reflect the users' perceived quality.

The remainder of this chapter is organized as follows. Section 4.2 provides an overview of the user study setup, outlining the methodology employed, including point cloud sequences selection, experimental design, and data collection procedures. In Section 4.3, we detail the conducting of the user study. Section 4.4 presents and analyses the findings, discussing the implications of the results. Section 4.5 is dedicated to developing and evaluating the machine learning models. Finally, Section 4.6 concludes the chapter with a summary of key findings and directions for future research.

4.2 EXPERIMENT DESIGN AND USER STUDY SETUP

This section provides an overview of our user study setup, covering the procedures undertaken, including selecting point cloud sequences, compression methods, and content generation.

4.2.1 *Selection of Point Cloud Sequences*

In our study, we selected two naturalistic full-body human figures, the *Dancer* from the Owlii dynamic human mesh sequence dataset [184] and *Thaidancer* from the 8i voxelized surface light field dataset [83], which are publicly available datasets. These figures were chosen as they demonstrate a range of variations. The *Dancer* has minimal texture details, as they wear a simple white shirt and jeans, while the *Thaidancer* has more intricate features, especially in their dress, neck pieces, and crown. The movements of the *Dancer* are fast and energetic, while the *Thaidancer's* movements are slow and fluid. Our goal was to include two distinct objects to examine the extent to which the perceived quality depends on the characteristics of the objects. Figure 4.2(a) shows the *Dancer* at a near distance. This point cloud sequence was recorded at 30 frames per second and comprises 600 frames, each containing approximately 2.6 million points. Figure 4.2(d) shows the *Thaidancer* at a near distance. This point cloud sequence was captured at 30 frames per second and consists of 300 frames, each containing more than 3 million points. Each point in the point cloud sequences has 3D coordinates and red, green, and blue colour channels. As commodity hardware like the Realsense LiDAR and D-RGB camera do not directly provide normals, we only retain point position and colour information.

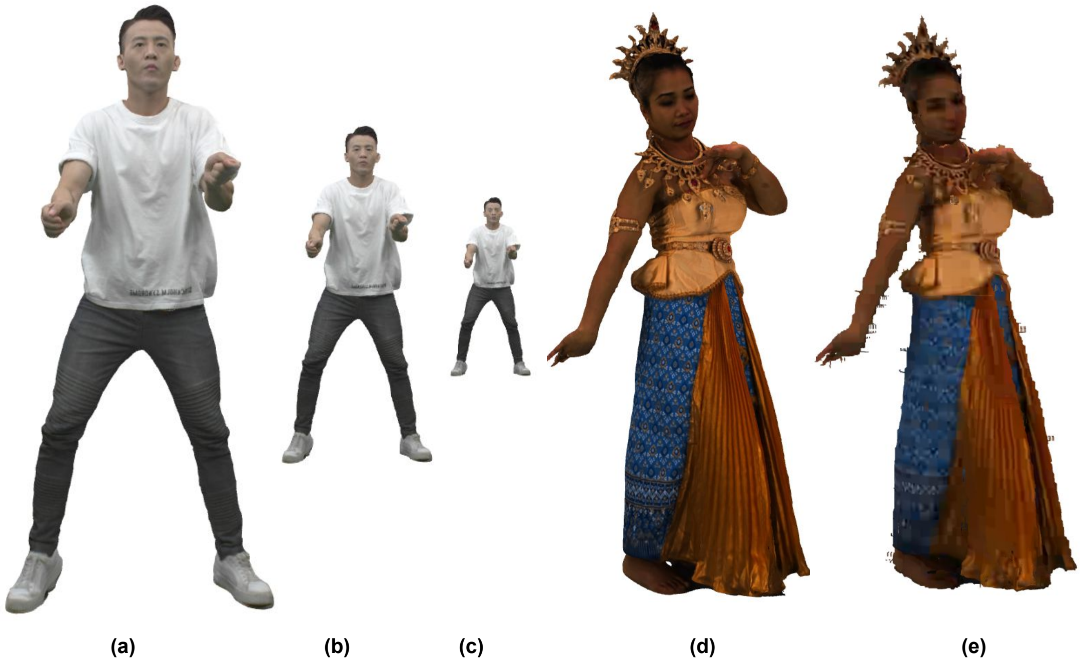


Figure 4.2: Example visual views of the point cloud objects used in this study. Figures (a), (b), and (c) show the *Dancer* point cloud at near, medium, and far distances, respectively. Figures (d) and (e) show the *Thaidancer* point cloud encoded using V-PCC with m5 and mb quantization parameters, respectively.

4.2.2 *Point Cloud Compression and Videos Generation*

In our study, we evaluated two state-of-the-art point cloud compression methods, V-PCC [48] and Draco [45]. V-PCC, which is by far the most used codec for point clouds, utilizes an advanced 2D video codec by projecting point clouds onto frames. At the same time, Draco achieves compression by quantizing each point's information and conducting mesh compression. Both methods can be customized, leading to file size and quality variations. We defined five quantization levels for each compression method applied to the raw point cloud sequences, labelled as *(mb, m0, m1, m3, m5)* for MPEG V-PCC and *(d8, d9, d10, d11, d16)* for Draco. Each quantization level behaves differently. For instance, quantization levels like *m5* are designed to deliver higher quality but require more bandwidth than other quantization levels, such as *m3*. To vary the compression levels of V-PCC, we adjusted the geometry quantization parameter *geometryQP*, attribute quantization parameter *attributeQP*, and occupancy precision *occupancyPrecision*. The lowest quality level, denoted as *mb*, was generated with *geometryQP* = 51, *attributeQP* = 51, and *occupancyPrecision* = 4. For Draco, only the position quantization parameter *positionQP* was set to the desired quantization

Compression Method	Quantization Levels		Parameters		
			GeometryQP	AttributeQP	OccupancyPrecision
V-PCC	0	mb	51	51	4
	1	m0	36	44	4
	2	m1	24	32	4
	3	m3	20	27	4
	4	m5	16	22	2
			PositionQP (number of bits)		
Draco	0	d8	8		
	1	d9	9		
	2	d10	10		
	3	d11	11		
	4	d16	16		

Table 4.1: Quantization levels and parameters for V-PCC and Draco compression methods.

level. For instance, to achieve the *d8* quantization level, *positionQP* was set to 8. Table 4.1 summarizes the quantization levels and parameters for both MPEG V-PCC and Draco compression methods. Figure 4.2(d) and Figure 4.2(e) show the *Thaidancer* object compressed using V-PCC at the *m5* and *mb* quantization levels, respectively.

The point cloud sequences were encoded at each of the five quantization levels. Because of the limitations in participants' commodity PC hardware, real-time playback was not feasible for the user to experiment with V-PCC due to the heavy decoding computations. Hence, we rendered the point cloud sequences from fixed camera perspectives using lossless encoding and stored the results as H.264 video files, which did not add any compression loss or artefacts to the content. The point cloud sequences were also rendered at three different distances (near, medium, and far), and the object's size appeared smaller as the distance increased. Figure 4.2(a), (b), and (c) respectively depict the *dancer* at near, medium, and far distances. To evaluate the performance of V-PCC, we also explored different frame rates, rendering the point cloud sequence at frame rates of 30, 15, and 10 frames per second. The higher the frame rate is, the more fluid the animation is. For Draco, the frame rate was fixed at 30 frames per second to reduce the workload on study participants and minimize the number of videos that needed to be rated. This resulted in a total of 120 different configurations to be evaluated.

4.3 CONDUCTING THE USER STUDY

This part of the work aims to conduct a subjective user study to investigate the human responses to the combined effect of compression levels and methods, frame rate, distance, and object type. At the end, we should obtain data showing each parameter's effect on MOS.

A remote testing paradigm can be employed to evaluate point clouds, as concluded in this recent research study [152]. We implemented a webpage-based rating application to evaluate the quality of the generated videos with a large number of participants. The user study was conducted offline on each participant's computer to minimize the effect of video buffering delay. This allows the quality to be estimated based only on content-related factors when the transmission is free from degradation. Participants were instructed to download and extract an archive containing all the study files. The experiment was initiated by opening a webpage using a local web browser, which provided a step-by-step guide for completing the study.

The video quality evaluation was divided into two parts, one for each point cloud object sequence, i.e., *Dancer* and *Thaidancer*. The participants were presented with 45 videos for each part, based on the different experiment configurations such as encoding method, quantization level, distance, and frame rate. Each participant was randomly assigned the order of the objects and the videos for each object. Before starting each part, participants were shown a reference video of ideal quality generated from uncompressed point cloud data at a frame rate of 30 fps and a near distance. After viewing a point cloud video for at least 3 seconds, participants could select one of five quality levels: 'bad', 'poor', 'fair', 'good', or 'excellent', with corresponding scores of 1, 2, 3, 4, and 5, respectively.

Once a quality level was chosen, the survey would proceed to the following video. The data was collected locally and saved as a JSON file in the participant's download directory at the end of the survey. Participants were then allowed to upload their results to our servers anonymously. The entire assessment process takes approximately 15 to 20 minutes to complete. We utilized the Prolific crowdsourcing platform [1], involving a total of 102 participants. Each participant was incentivized with a monetary reward of approximately 5 euros.

4.4 EVALUATION RESULTS FROM USER STUDY

This section presents the findings from our examination of user study results. The following subsections dive into a focused analysis of specific configurations.

Figure 4.3 visually presents the MOS for each experiment configuration, arranged in ascending order along the Y-axis, and each subplot showcases the corresponding independent variable. The data reveals that the range of MOS spans from a minimum of 1.12 to a maximum of 4.37. The first 20 configurations represent the lowest quality level and show that MOS significantly increases with higher distance This makes sense as higher distances tend to obscure the nuances of quality. At low-quality levels, the compressor becomes more tolerant of errors; consequently, more artifacts appear in the content. The frame rate does not significantly impact the outcomes, regardless of whether it is high or low when the quantization level is low. Beyond configuration 20, the increase in MOS is relatively linear. Subplot (a) highlights the notable influence of quantization level on MOS, demonstrating higher scores in experiments with increased quantization levels. Subplot (c) highlights the impact of

1 `https://www.prolific.com/` last accessed: December 2, 2024.

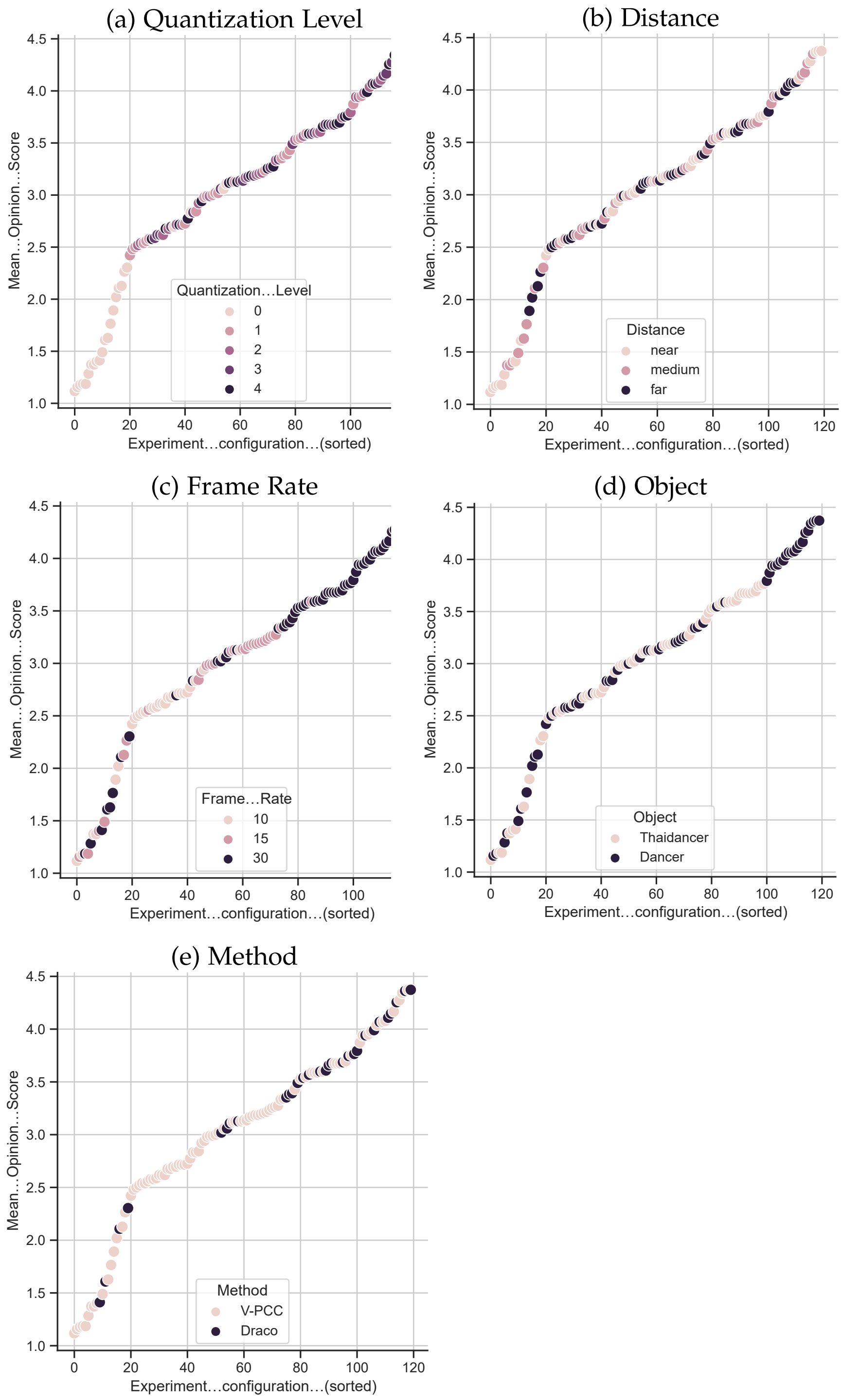


Figure 4.3: This visual breakdown presents a view of MOS across various experiment configurations, sorted in ascending order. Each subplot highlights a distinct factor's impact on MOS: (a) Quantization Level, (b) Distance, (c) Frame Rate, (d) Object, and (e) Compression Method. Refer to Table 4.1 for details on the quantization levels.

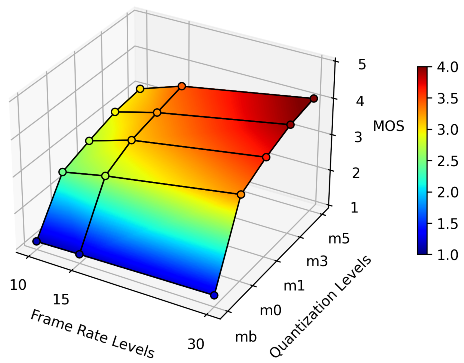
frame rate on MOS, indicating higher scores for experiments using 15 or 30 frames per second, with the latter resulting in higher scores. In subplot (b), we observe that the highest MOS values are achieved at near distances. This could be due to two factors. First, participants may be biased towards giving higher scores to near objects, as the reference object was also at a near distance. Second, users may be more reluctant to give the highest score to far objects. In subplot (d), configurations 100 to 119 illustrate that participants consistently assigned higher scores to the *Dancer* object compared to the *Thaidancer* object. We will delve into a more detailed discussion of the impact of objects on the MOS in an upcoming subsection. It is worth noting that these results cover both Draco and V-PCC compression methods. As illustrated in subplot (d), most configurations belong to V-PCC, as Draco configurations are limited to a frame rate of 30. Notably, we observe that Draco configurations are predominantly in the upper half of the experiment configurations, as only 30 fps was used, and therefore have a comparatively high MOS. In the following subsections, we will detail the findings concerning each compression method.

4.4.1 *Video Codec Based Compressed Point Clouds*

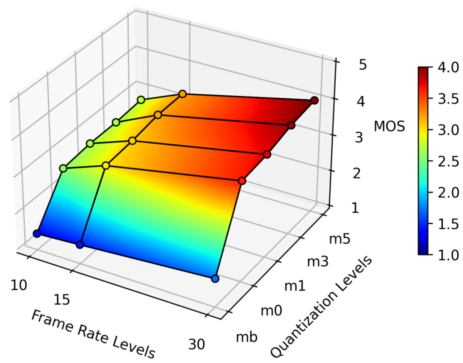
The results of the MOS ratings for the point cloud sequences compressed using the V-PCC method are shown in Figure 4.7. This figure illustrates the aggregated MOS values for the V-PCC compression method at various distance levels (a) near, (b) medium, and (c) far, as well as different frame rates and quantization levels for the aggregated data of both objects.Despite the individual ratings falling within the range [1, 5], the highest MOS across all configurations was 4.37 (near, 30 fps, m5). This can be attributed to the aggregated nature of the MOS, wherein participants might have been cautious about assigning the highest rating of 5 due to subtle differences in quality. Additionally, it is possible that the visual quality of the content could have appeared more impressive, as users often seek better quality even if it is already satisfactory. As shown in Figure 4.7(a), when the participants were close to the object, the MOS steadily increased with the quantization level. The results indicate that higher frame rates had a positive impact on the MOS for quantization levels *m0* to *m5*, but had little effect on the lowest quantization level *(mb)*.

At a medium distance, as shown in Figure 4.7(b), the MOS levels become more consistent. The highest MOS reached is 4.02 for (medium, 30 fps, m5). Most quantization levels at 30 fps have similar MOS values as the near setting, with slightly higher values for the lower quantization levels *mb* and *m0*. Notably, the impact of frame rate is more significant in this setting compared to the near setting, where lower frame rates of 10 and 15 at medium distance result in a noticeable decrease in MOS compared to the near setting.

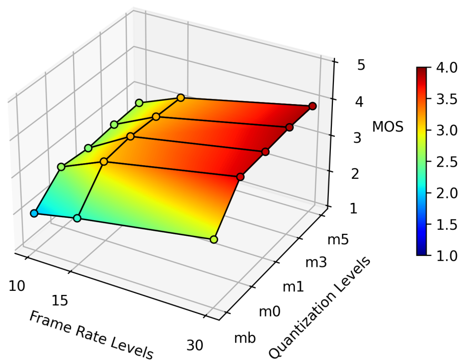
The results for far objects, as shown in Figure 4.7(c), exhibit a more flattened surface. The highest MOS of 3.87 is achieved at (far, 30 fps, m5) with lower MOS values compared to the other two distance settings for high quantization levels. On the other hand, the *mb* and *m0* quantization levels result in higher MOS values compared to



(a) Distance: near



(b) Distance: medium



(c) Distance: far

Figure 4.7: Aggregated MOS for the V-PCC compression method with distance levels: (a) near, (b) medium, and (c) far. Each subplot shows the MOS over **both objects** for individual frame rate and quantization levels.

the near or medium distance levels. When considering a fixed frame rate, the results for quantization levels *m1* to *m5* are similar to the medium distance setting.

Our findings suggest that changes in an object's compression quality become less noticeable as it moves further away. In comparison, the impact of different frame rates is still recognizable and significantly impacts the MOS. For instance, increasing the frame rate of a far object with a quantization level of *m0* from 10 to 30 has a more significant positive effect on the MOS than upgrading its quantization level to *m5*. These results can aid in optimizing future point cloud streaming mechanisms, particularly when the current distance to the user is known.

4.4.2 *Draco Compressed Point Clouds*

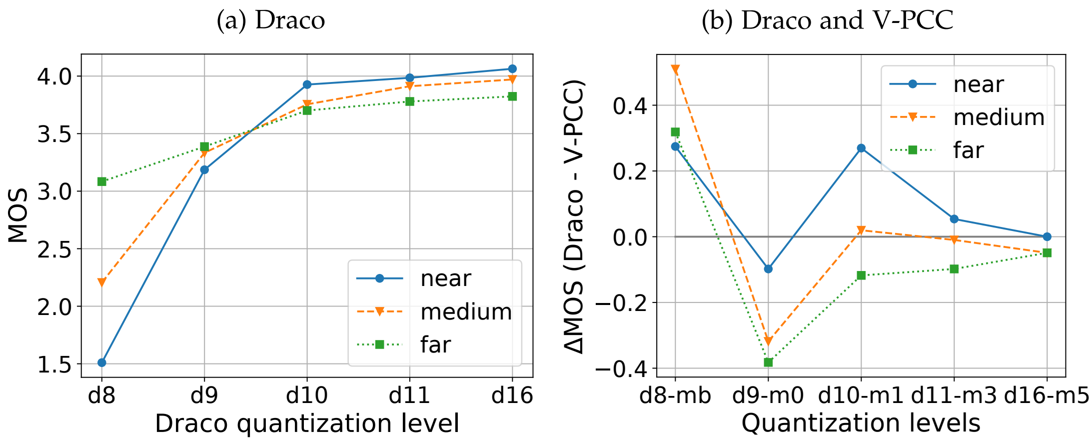


Figure 4.8: Summary of Draco MOS analysis, including: (a) aggregated MOS for 30 objects per second using Draco compression in distance levels near, medium, and far for both objects. (b) comparison of MOS difference between Draco and V-PCC quality settings with the same configurations. A positive value suggests that Draco achieved a higher MOS.

The results for Draco compression are shown in Figure 4.8(a). With a fixed frame rate of 30 frames per second, the trend observed is similar to that of V-PCC. As the viewing distance increases, the influence of the quantization level on the MOS score becomes less noticeable. Users appear less inclined to assign either high or low scores, suggesting a convergence of perceived quality as the object moves further away.

To facilitate a comparative analysis between Draco and V-PCC, both operating at 30 frames per second, Figure 4.8(b) compares their respective results. The findings reveal that at the first quantization level *d8-mb*, Draco received a significantly higher MOS score compared to V-PCC, while V-PCC performed better at the subsequent quantization levels. As the quality levels increased, the difference in MOS between the two methods became negligible. Except for the pair *d8-mb*, the results suggest that users preferred V-PCC as the viewing distance increased. Conversely, Draco was favoured over V-PCC in the near setting.

4.4.3 *Differences in User Study Objects*

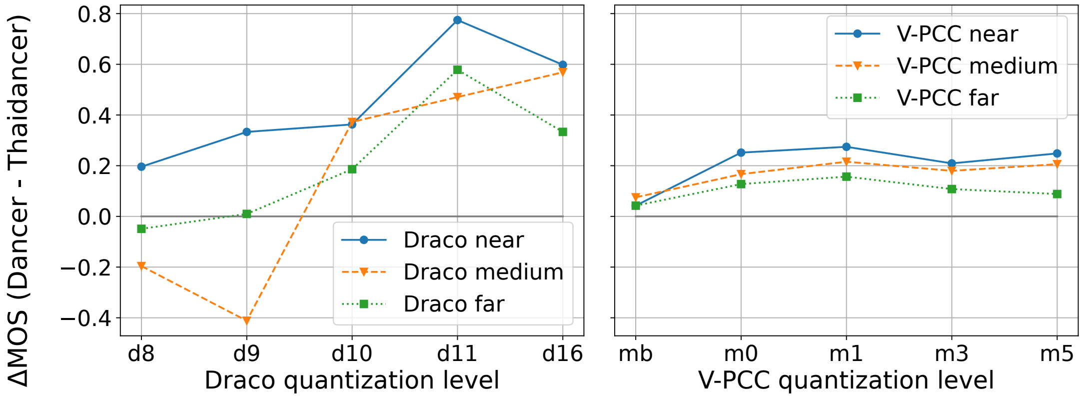


Figure 4.9: Variation in MOS between the *Dancer* and *Thaidancer* objects across all distance and quantization levels for both Draco and V-PCC. A positive value implies that the *Dancer* object received a higher MOS

As depicted in Figure 4.3(d), the results indicate that the *Dancer* object consistently receives higher scores compared to the *Thaidancer* object. This observation is further emphasized in Figure 4.9, where the plot illustrates that the *Dancer* object generally receives superior scores in comparison to the *Thaidancer* object. The only exception is when the *Thaidancer* is viewed at medium and far distances using the Draco compression method and lower quality settings *d8* and *d9* (only for far), where it receives higher scores. However, the difference between the two objects is more significant for higher-quality Draco settings. The results from our user study suggest a possible correlation between the level of detail in the objects and the perceived quality of the compressed point clouds. With its higher level of detail, the *Thaidancer* object appears to result in more recognizable artefacts for Draco and V-PCC compression methods. Therefore, the participants become more demanding in terms of perceived quality. Fine details are difficult to distinguish at lower quality settings, and participants become less demanding and may prefer the *Thaidancer* object due to smoother movements. Further investigation into the impact of different objects on perceived quality is needed.

4.4.4 *Analyzing the Tradeoff Between Quality and Resource Utilization*

Video streaming content may be viewed on various devices with distinct capabilities and transmitted via networks characterized by limited bandwidth and varying video display resolutions. The compression of point clouds involves balancing the desired visual quality against the resources required to attain that quality [9]. This balance is influenced by several factors, such as the bit rate of the point cloud stream, the encoding and decoding latency for each frame, and the processing capability of the end devices.

Decoding / Encoding time (in seconds)		Object / Quantization Level				
		Dancer				
Method	FR	mb	m0	m1	m3	m5
V-PCC	10	33.59 / 1023.31	37.10 / 1051.17	40.58 / 1092.50	43.516 / 1150.33	44.37 / 1243.30
	15	50.41 / 1535.56	55.64 / 1576.68	60.88 / 1638.72	65.26 / 1725.75	66.51 / 1864.69
	30	100.7 / 3068.65	111.33 / 3151.56	121.72 / 3275.63	130.49 / 3449.40	133.09 / 3727.24
Draco	30	d8	d9	d10	d11	d16
		6.57 / 27.85	7.54 / 31.50	8.27 / 34.55	8.40 / 36.70	8.14 / 46.96

Decoding / Encoding time (in seconds)		Object / Quantization Level				
		Thaidancer				
Method	FR	mb	m0	m1	m3	m5
V-PCC	10	50.82 / 1360.86	52.67 / 1365.07	52.88 / 1373.88	55.99 / 1419.66	56.91 / 1556.75
	15	76.03 / 2029.22	78.76 / 2035.84	79.20 / 2047.24	83.76 / 2117.65	85.11 / 2321.84
	30	152.20 / 4064.36	157.69 / 4076.65	158.42 / 4100.52	167.62 / 4239.62	170.40 / 4650.21
Draco	30	d8	d9	d10	d11	d16
		8.23 / 34.25	8.94 / 38.08	9.61 / 41.66	9.86 / 44.56	9.73 / 57.06

Table 4.2: Decoding and encoding time (in seconds) for V-PCC and Draco compression methods at different frame rates and quantization levels for *Dancer* and *Thaidancer* sequences. Lower values are better as they indicate faster processing.

Bitrate (in Mbits/s)		Object / Quantization Level									
		Dancer					Thaidancer				
Method	FR	mb	m0	m1	m3	m5	mb	m0	m1	m3	m5
V-PCC	10	0.96	1.67	2.02	4.40	13.46	1.55	3.50	4.38	10.59	28.56
	15	1.45	2.51	3.03	6.60	20.19	2.33	5.26	6.57	15.89	42.93
	30	2.91	5.03	6.07	13.21	40.43	4.66	10.52	13.14	31.77	85.76
Draco	30	d8	d9	d10	d11	d16	d8	d9	d10	d11	d16
		547.28	735.53	963.34	1215.62	2420.79	860.27	1076.05	1325.23	1602.85	3006.42

Table 4.3: Bitrates (in Mbits/s) for Draco and V-PCC compression methods at various frame rates and quantization levels for *Dancer* and *Thaidancer* Sequences. Lower bitrates are preferable as they represent more efficient compression.

Our study uncovered that although the MOS for Draco and V-PCC compression methods are relatively similar to our selected configurations, there is a substantial difference in the bit rates for the compressed streams. The raw point cloud sequences have an average data rate of approximately 3.6 Gbit/s, 5.5 Gbit/s, and 10.9 Gbit/s for 10 fps, 15 fps, and 30 fps, respectively. The results showed that while the bit rates of Draco streams are much higher than those of V-PCC streams, they are still lower than the raw point cloud sequences. However, Draco's encoding and decoding times were much faster than those of V-PCC on our system with an AMD Ryzen 7 5800X processor. The average encoding time for V-PCC ranges from 119.2 to 140 seconds per frame, and the average decoding time is from 4.2 to 5 seconds per frame. In contrast, Draco's encoding time ranges from 1 to 1.7 seconds per frame, and its decoding time is from 0.24 to 0.29 seconds per frame. Table 4.2 provides a comparison of the time it takes for decoding and encoding with V-PCC and Draco methods at various frame rates and levels of quantization for the *Dancer* and *Thaidancer* sequences. It is important to note that to support live streaming, a delay-intolerant service, compression methods that combine low data rates with fast encoding and decoding times are required.

Thaidancer MOS		Distance / Quantization Level														
		Near					Medium					Far				
Method	FR	mb	m0	m1	m3	m5	mb	m0	m1	m3	m5	mb	m0	m1	m3	m5
V-PCC	10	1.11	2.48	2.71	3.05	3.12	1.37	2.53	2.67	2.69	2.77	1.89	2.72	2.51	2.61	2.71
	15	1.18	2.55	2.99	3.17	3.27	1.14	2.98	2.92	3.13	3.18	2.26	2.99	3.19	3.18	3.11
	30	1.18	3.01	3.33	3.58	3.75	1.62	3.43	3.52	3.67	3.69	2.69	3.58	3.59	3.67	3.67
Draco	30	d8	d9	d10	d11	d16	d8	d9	d10	d11	d16	d8	d9	d10	d11	d16
		1.41	3.01	3.74	3.59	3.76	2.30	3.53	3.56	3.67	3.68	3.10	3.38	3.60	3.49	3.65

Dancer MOS		Distance / Quantization Level														
		Near					Medium					Far				
Method	FR	mb	m0	m1	m3	m5	mb	m0	m1	m3	m5	mb	m0	m1	m3	m5
V-PCC	10	1.17	2.42	2.71	2.83	2.94	1.37	2.57	2.61	2.61	2.67	2.01	2.50	2.53	2.58	2.57
	15	1.15	2.84	3.16	3.34	3.58	1.49	3.0	3.21	3.26	3.25	2.12	3.23	3.20	3.13	3.12
	30	1.28	3.54	3.98	4.27	4.37	1.76	3.87	3.94	4.16	4.34	2.83	3.95	4.03	4.07	4.06
Draco	30	d8	d9	d10	d11	d16	d8	d9	d10	d11	d16	d8	d9	d10	d11	d16
		1.60	3.35	4.10	4.37	4.36	2.10	3.12	3.94	4.14	4.25	3.05	3.39	3.79	4.06	3.99

Table 4.4: MOS for *Thaidancer* and *Dancer* sequences are provided for both Draco and V-PCC compression methods at different distances and quantization levels. Higher MOS values indicate better perceived quality.

The results from previous sections indicated that higher quantization levels and frame rates lead to a higher MOS. However, this also means an increase in data rates. Table 4.3 shows a comparison of bitrates for V-PCC and Draco compression methods at various frame rates and quantization levels, providing insights into the efficiency of compression for both *Dancer* and *Thaidancer* sequences. In some instances, the participants may prefer a lower bit rate stream. For this discussion, let us disregard the encoding and decoding timing requirements. Moreover, let us concentrate on a scenario in which a client is equipped with a 50 Mbit/s connection in a near distance setting viewing *Thaidancer* sequence. With V-PCC, the client has the option to stream content, such as quantization level *m5* at 15 fps or quantization level *m3* at 30 fps (as shown in Table 4.3). Notably, based on Table 4.4, an inspection of the data in the first table clearly shows that *m3* at 30 fps yields a higher MOS rating, even though it has a lower bit rate, compared to *m5* at 15 fps. Figure 4.10 illustrates the relationship between different configurations of V-PCC and Draco quantization levels and frame rates and their impact on bitrates, encoding, and decoding time, affecting the quality of point cloud sequences for the *Dancer* object at a near distance. The figure reveals that various configurations can yield equivalent MOS scores while imposing differing bit rates and computation requirements. Figure 4.10 shows that the configurations within the same colour zone can be interchanged to achieve comparable MOS scores while exhibiting varying resource demands. For instance, consider the configurations (m3, 30fps) and (m5, 30fps) in Figure 4.10. Both configurations yield a high MOS score; however, the latter configurations demand a significantly higher bit rate, approximately three times more, than the former. Furthermore, these configurations, i.e. (m3, 30fps), require less encoding and decoding time.

Combining the expected resource requirements with estimating the MOS is crucial for effective adaptation schemes [145]. Therefore, leveraging the QoE knowledge obtained from the user study, we can develop an adaptive bitrate scheme to ensure a consistent and uninterrupted streaming experience for users, especially in the face of varying network conditions and diverse computing capabilities at the receiver side. We will provide more details on streaming adaptivity in the next chapter.

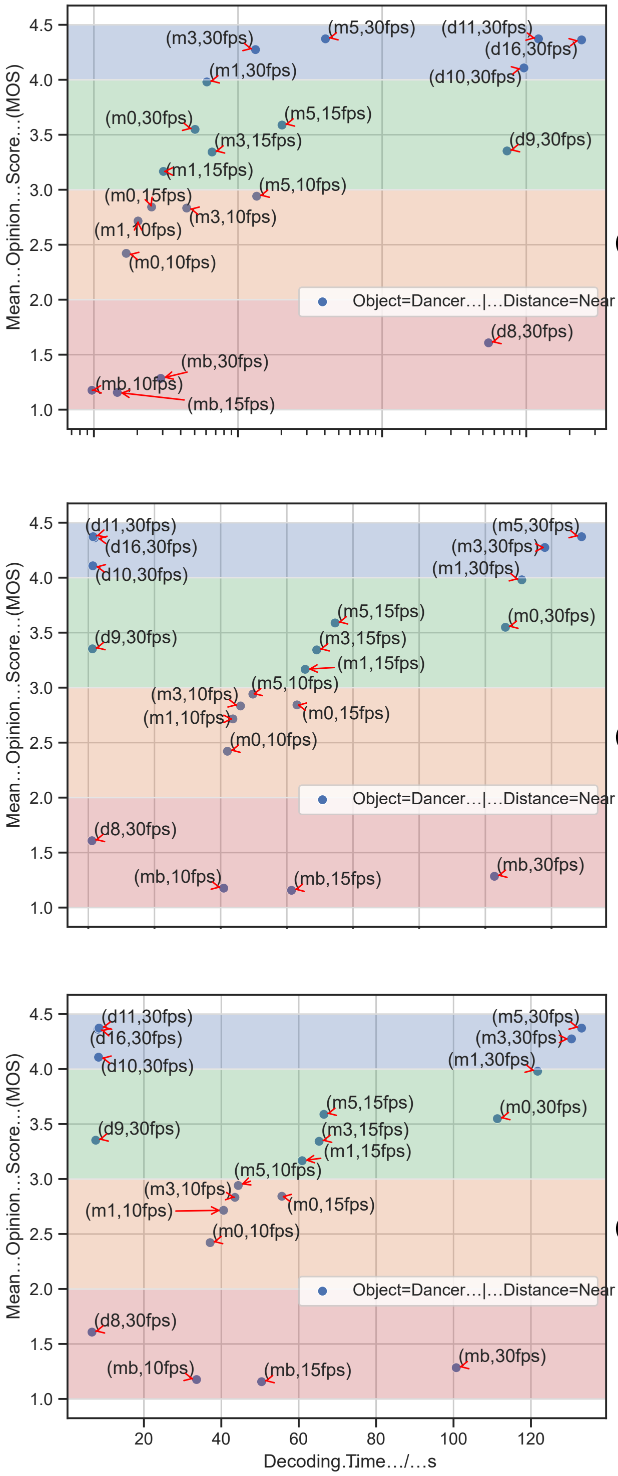
4.5 MODELLING QOE WITH MACHINE LEARNING

This part of the work aims to calculate MOS using a machine learning algorithm based on a data set of rated videos, i.e., subjective MOS, with various parameters, including distance, quantization parameters, and frame rate. We aim to extend the reach of our findings beyond our experimental configurations. We want to improve our understanding by making our results applicable to scenarios that were not part of our study or training phase. To do this, we will use supervised machine learning, which enables us to apply our previous findings more broadly. These QoE models could then be utilized in an adaptive streaming system considering a more comprehensive range of configurations. The adaptive streaming system can utilize the QoE model to determine the configurations for adjusting the point cloud video to match specific network conditions, such as decreasing throughput between the server and the client. If the MOS falls below a predefined threshold, the system responds by modifying parameters such as compression methods, quantization parameters, and frame rate, followed by recalculating the MOS. Figure 4.11 illustrates how the QoE model can be integrated with the adaptive streaming server.

In the context of this study, we treat users' MOS assessments as synonymous with QoE. These MOS scores, reflecting users' personal judgments on point cloud content, play a crucial role in shaping the overall perception of quality and user satisfaction. Our study centres around MOS scores, deliberately omitting factors such as buffering, network parameters, and user interface considerations. This intentional focus allows us to concentrate on users' subjective evaluations of the quality of point cloud content.

The modelling of QoE has been done in three steps: feature extraction, training of the machine learning model, and evaluation of the predictions. The data used in this process included approximately 9,000 samples, each comprising five features: frame rate, distance, V-PCC parameters including *geometryQP*, *attributeQP*, and *occupancyPrecision* which are listed in detail in Table 4.1, along with the corresponding user-rated scores ranging from 1 to 5. These user ratings acted as the targets for our models. To improve the reliability of the model, outliers were removed through boxplot-based outlier detection [38].

SHapley Additive exPlanations (SHAP) [102] was employed to determine feature importance. The results indicate that the *occupancyPrecision* is not a significant factor in the model's accuracy. This is attributed to the limited variability of values, primarily 2 and 4, across all configurations. Such a narrow range may not provide sufficient information for *occupancyPrecision* to predict the target variable effectively. Figure 4.12 illustrates the absolute mean SHAP values. These values generally indicate the importance of features and highlight their significant contributions to the model's predictions. The insights from SHAP analysis reveal that the *FrameRate* is the most crucial feature influencing the target, i.e., MOS. Both *attributeQP* and *geom-*



(a) Bitrate

(b) Encoding Time

(c) Decoding Time

Figure 4.10: Impact of (a) bit rate, (b) encoding time, and (c) decoding time on MOS for *Dancer* object at near distance under different V-PCC and Draco quantizations and frame rate configurations.

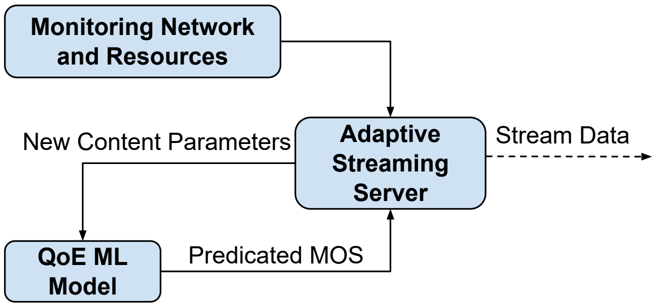


Figure 4.11: Conceptual diagram illustrating the integration of a pre-trained QoE model with an adaptive streaming system.

etryQP have nearly equal impacts on predicted values, while *Distance* has the most minor influence. Additionally, these insights aid in refining models by identifying *occupancyPrecision* as an unimportant feature.

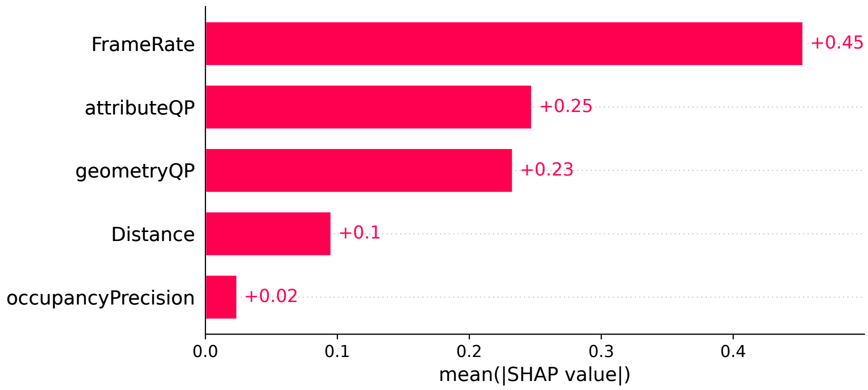


Figure 4.12: A visual representation illustrating the importance of features based on the absolute mean SHAP values. Each bar represents the significance of a specific feature in influencing the model's predictions, and higher values indicate greater importance.

The evaluation of the machine learning model involved a leave-one-out cross-validation technique [180]. This process involved dividing the data into 45 folds, with 44 folds used for training and one for testing. This cycle was repeated 45 times, with each fold as the test set once. We grouped the user study responses based on these folds to ensure fairness and accuracy. This cross-validation approach provides an unbiased evaluation of the model's performance and helps detect overfitting, thereby ensuring the robustness and reliability of the model.

Table 4.5 displays the results of evaluating the performance of regression and classification models using the Scikit-learn[2] Python library. The table presents the R-squared (R2) score and the Mean Squared Error (MSE) [2] statistical metrics of the MOS providing insights into prediction accuracy, averaged across all cross-validation

2 `https://scikit-learn.org/stable/`, last accessed on December 2, 2024

QoE Model Type	R2 score ↓	MSE
Gradient Boosting Regressor	0.9754	0.0174
Polynomial Regression (Degree 2)	0.9677	0.0229
Random Forest Regressor	0,9560	0,0312
Decision Tree Regressor	0.9456	0.0386
Decision Tree Classifier	0.9439	0.0398
MLP Classifier	0.9424	0.0409
Gradient Boosting Classifier	0,9323	0,0480
Random Forest Classifier	0,9063	0,0665
MLP Regressor	0,9021	0,0695
Logistic Regression	0,8503	0,1063
Ridge Regression	0,8183	0,1291
Linear Regression	0,8183	0,1291
K Neighbors Regressor	0,7343	0,1887
Lasso Regression	0,6998	0,2133

Table 4.5: Performance of various machine learning models evaluated using the R2 score (higher is better) and the Mean Squared Error (lower is better), sorted by decreasing performance.

rounds. Our analysis showed that the Gradient Boosting Regression model performed best, as indicated by its high R2 score and low MSE.

Additionally, we investigated the advantages of using classification models, where the probability per class prediction serves as a QoE distribution instead of predicting the MOS, as in the regression models. Figure 4.13 compares the actual QoE distribution to the predicted distribution for a chosen configuration. As an illustration, the Decision Tree Classifier can accurately predict the perceived quality distribution, even though none of the related samples were seen during training. For instance, Figure 4.13 suggests that less than 11% of users are likely to view this specific configuration as poor or bad, providing additional insights by predicting the score distribution rather than the MOS.

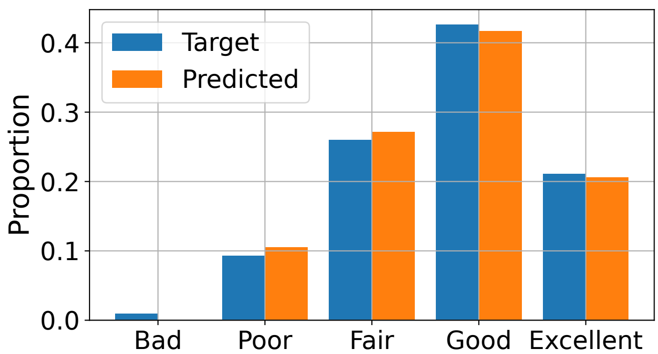


Figure 4.13: Actual and predicted QoE distributions for the test fold with medium distance, 30 fps, and m1 quantization level using the Decision Tree Classifier.

4.6 DISCUSSION AND CONCLUSIONS

This chapter aimed to comprehend the impact of different adaptations and create a QoE model for lossy compressed point cloud sequences. This information is essential for developing adaptive mechanisms for the streaming of point cloud sequences. Analysing how a single parameter can influence the viewer's satisfaction can assist in determining the optimal balance between various parameters, ultimately improving video quality for users. Therefore, we investigated how various adaptation variables, such as quantization levels, distances from the camera, and frame rates, affect point cloud quality perception. To understand the user's experience, we conducted a subjective user study, gathering data on the performance of two popular compression methods, Draco and V-PCC, and their trade-offs between resource requirements and quality levels. Our user study aimed to investigate the impact of varying frame rates, quantization levels, and distances on point cloud objects using both Draco and V-PCC compression methods. The user's perception is evaluated using ACR, a scale from 1 to 5. Building upon this foundation, we employed machine learning to model the QoE of point cloud sequences. We assessed their accuracy by comparing the prediction results to the MOS scores obtained from the user study. By feeding the machine learning models with different factors like precision and geometry quality, we aimed to predicate the QoE for configurations the model had not previously encountered.

We concluded that the quantization level is important in assessing the video quality and frame rate, directly affecting customer satisfaction. Our user study findings showed that the frame rate's impact on quality is negligible when the quantization level is low, but it becomes positive at higher quantization levels. Additionally, there is an inverse correlation between distance and MOS, meaning greater distance leads to lower MOS scores.

When the V-PCC compression method was used on point cloud sequences with participants in close proximity to the object, MOS steadily increased with the quantization level. Moreover, our findings demonstrated that higher frame rates had a positive impact on MOS scores for quantization levels *m0* to *m5* but had minimal effect on the lowest quantization level *mb*. Interestingly, the influence of frame rate was more pronounced in specific scenarios, mainly when the frame rate was lower (10 and 15 FPS) at medium distances, leading to a noticeable decrease in MOS compared to the near setting. Conversely, the quantization levels *mb* and *m0* at far distance level resulted in higher MOS values compared to the near or medium distance levels. For example, increasing the frame rate of a distant object with a quantization level of *m0* from 10 to 30 had a more significant positive effect on MOS than upgrading its quantization level to *m5*.

We also noticed that the type of point cloud object affects the user's acceptance. Our results indicate that the level of detail in point cloud objects may correlate with the perceived quality of compressed point clouds. With its greater detail, the *Thaidancer* object produced more noticeable artifacts in Draco and V-PCC compression, especially at lower-quality settings. Consequently, participants tended to give higher scores to the *Dancer* object, which had minimal texture details and a simpler appearance.

While Draco and V-PCC MOS scores for our selected configurations are relatively similar, there is a substantial difference in the bit rates for the compressed streams. Notably, encoding and decoding times for Draco were significantly faster than those for V-PCC. The study results reveal that different configurations can achieve similar MOS scores while imposing varying bit rate, encoding, and decoding latency requirements. Our findings indicated that increasing frame rates and quantization levels individually improve perceived quality. However, we also found that a higher bitrate does not necessarily equate to a higher MOS. Additionally, the degradation in quality due to quantization becomes less noticeable with increasing distance from the camera. Understanding the correlation between different configurations, MOS scores, and resource requirements is essential for developing multi-mechanism transition-based adaptive bitrate schemes that ensure a consistent streaming experience under changing network conditions.

Modelling the QoE based on data collected from the subjective user study revealed that *occupancyPrecision* does not significantly impact the model's predictability. The *FrameRate* is the most critical feature influencing the target variable, namely the MOS. Additionally, our analysis offers additional insights through predictions of score distributions using a classification model rather than relying solely on the actual MOS.

Our exploration of point cloud quality based on real user experiences and machine predictions deepened our understanding. We now possess insights that can steer improvements in immersive experiences. The developed model can predict the perceived quality of point cloud sequences, enabling optimization and adaptation in real-world scenarios. While our study is based on reliable QoE derived from user feedback, it is essential to acknowledge the limitations. The user study was conducted on a small scale with a hundred participants, and the perceptual scores provided by users in a controlled environment may not reflect quality in real-world scenarios. Future research can explore larger datasets, explore the results obtained using standard desktop PCs versus more immersive end devices, such as VR/AR headsets, and use enhanced machine learning methods to further refine our understanding.

4.7 SUMMARY

In this chapter, our objective was to explore the influence of various adaptations on the quality of compressed point cloud sequences and to construct a MOS model. To achieve this, we conducted a user study, gathering data on the performance of different compression methods and pinpointing the critical factors that affect QoE. Subsequently, we leveraged machine learning techniques to predict MOS. The proposed machine learning model estimates MOS based on content and quantization parameters. This approach allowed us to gain deeper insights into the impact of these adaptations on the overall user experience and refine our understanding of the QoE landscape.

5 CONTENT-AWARE POINT CLOUD STREAMING

In the previous chapter, we examined the different factors that affect the Quality of Experience (QoE) with point cloud content. Our main goal was to understand better how various adaptations can impact the overall quality of point cloud content. Therefore, we conducted a user study that equipped us with a clear understanding of QoE of point cloud content and its resource demands, which enabled us to develop a machine learning model to predict QoE scores effectively. This chapter represents a logical progression from our findings in Chapter 4. Here, we aim to explore how to integrate semantic and object-level information, focusing on how these elements can be leveraged to enhance the efficacy and adaptability of point cloud content streaming.

In this chapter, we will explore the role of semantics and object knowledge in adapting point cloud content while streaming. We will examine how semantic information can influence adaptation decisions, from prioritizing particular objects to optimizing resource allocation. This chapter explores a mechanism for enhancing content adaptation by considering the contextual information in the point cloud content. Additionally, we will investigate how this proposed approach impacts the overall QoE, aiming to meet the requirements of the intended point cloud applications. These requirements could include accuracy, latency, compatibility with changing bandwidth, or any other specifications necessary for the intended applications to function optimally. Through a series of experiments, we will demonstrate the practical applications of this approach and the benefits it can bring to point cloud content delivery. We discuss the challenges we may encounter and the exciting possibilities that arise as we integrate semantics and object knowledge into an adaptive streaming point cloud system.

As illustrated in Figure 5.1, in this chapter, we investigate the third research goal: *Investigating the impact of incorporating object-related information in point cloud content streaming on the quality and efficiency of the delivery process compared to conventional methods*. We assume that point cloud content is heterogeneous and can be prioritized based on specific application requirements. This assumption paves the way for a thorough investigation of how this prioritization can be integrated into the content streaming process, which could significantly improve both the quality of the final output and the delivery system's efficiency.

The structure of this chapter is as follows. In Section 5.1, we discuss the proposed adaptive point cloud compression method based on incorporating semantic content information. We introduce our mechanism for adapting point cloud content based on the importance of each object. We then elaborate on our approach to integrating semantic information into the data encoding phase, presenting how we encode point cloud scans to adapt to changing bandwidth for efficient delivery. We rely on

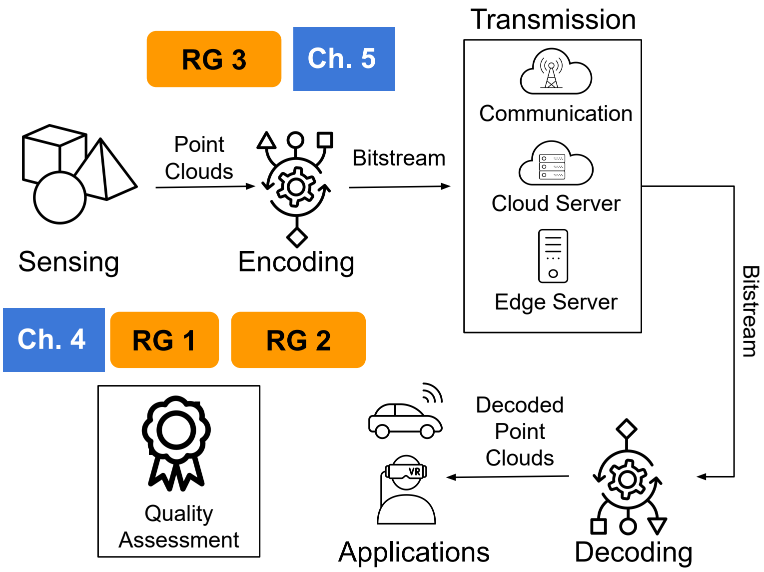


Figure 5.1: Point cloud streaming pipeline from the creation of content via sensing to the consumption by users in an application. In this chapter, we explore the third research goal: a thorough investigation into the influence of object-related information inclusion in point cloud content streaming on the delivery process's quality and efficiency.

semantic information prediction to assign different criticality levels point-wise. Additionally, we discuss how quantization and frame rate can be adjusted to enable bit rate adaptability. In Section 5.2, we present the evaluation of our approach, consolidating the implementation of the proposed method, along with details about the dataset and metrics used to measure performance compared to the traditional methods. Finally, we discuss in Section 5.3 the contributions and limitations of the proposed method.

5.1 METHODOLOGY: CONTENT-AWARE ADAPTIVE POINT CLOUD STREAMING

In this section, we propose a method that enables the adaptive compression of point cloud objects based on their significance in the scene. This approach's philosophy is based on the recognition that different areas of point clouds can exhibit varying levels of importance. Prioritizing points based on significance allows for efficient resource allocation, investing more resources on important points while saving them on less relevant points [23]. This approach provides dual benefits: first, it reduces the data volume by minimizing the number of irrelevant points by removing them, thus optimizing resource usage; second, it maintains or enhances the quality of relevant points to serve the intended goals of the data for a given application. We leverage semantic analysis, which detects foreground objects and background regions in point cloud scenes. We categorize foreground object points into stationary and moving points. For stationary points, we obtain benefits by separating them from the other points; this allows us to send them only once and efficiently reconstruct them by referencing their location in the previous frames, thereby minimizing resource consumption, including bandwidth and computation. Furthermore, moving points are compressed based on the intended goal of the point cloud data application. For applications that utilize moving points, fine compression configuration is applied, achieving compression of relevant objects with minimal quality degradation. Alternatively, for applications that do not require information about moving points, which may be considered noise in some cases, a harsher compression configuration is applied, or these points are eliminated altogether to save resources on these objects.

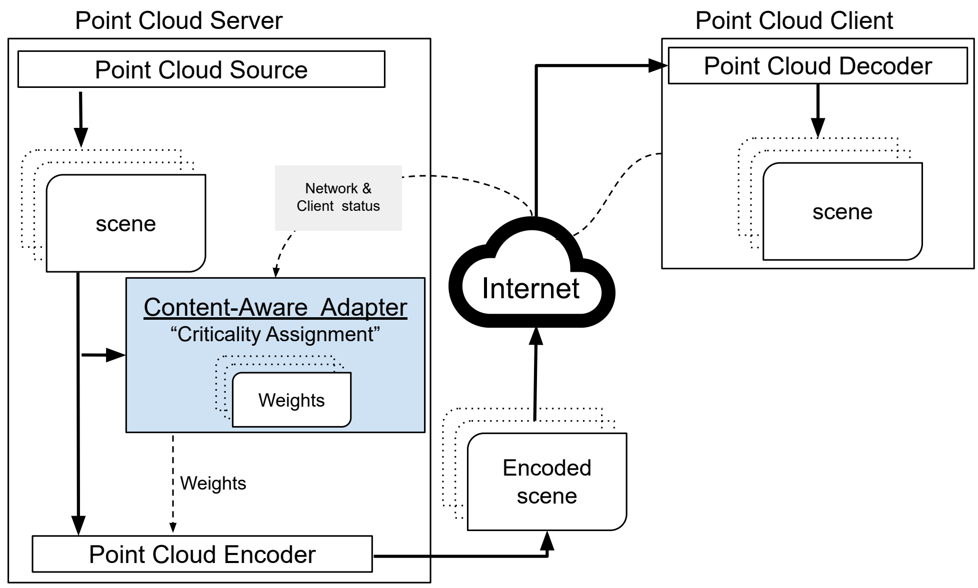


Figure 5.2: Our point cloud streaming system consists of a point cloud server (left) and a client (right). We propose a content-aware adapter that is running on the server.

This approach utilizes different components: a point cloud semantic segmentation model, a point cloud encoder, and a decoder. First, we use a semantic segmentation model to analyse the scene by identifying the objects and their semantics in the point cloud. This information is then used to construct a point cloud adaptor, which selectively assigns weights to the points in the scene according to their importance, preparing them for efficient delivery. Our proposed content-aware adaptor aims to enhance the adaptability of point cloud streaming systems and reduce the bandwidth and computational demands of these streams. The adaptor is designed as a software component between the point cloud content producer, such as a LiDAR sensor or recorded files, and the content encoder, as illustrated in Figure 5.2. It is used with the encoder to selectively compress and prepare the objects in the point cloud scene for streaming. Importantly, our design does not require any modifications to the existing encoder, allowing seamless integration into existing systems. Our proposed adaptor can adjust the bitrate during the encoding process to meet the desired requirements.

The server's point cloud scans undergo various modifications before being delivered to the client. The point cloud server acquires the scans from the content source and uses monitoring information about the current client and network conditions [139] to optimize the quality accordingly. These scans are then passed to the encoder, which generates the encoded scenes to be sent to the client. The encoder utilizes weights provided by the adaptor component to compress the point cloud selectively. Weights are used to assign importance to each point in the point cloud, allowing adaptive compression.

Our approach is based on the assumption that different regions of the point cloud scan have distinct importance, influencing their relevance to the user's interest or the application's needs. Thus, it is necessary to compress the entire scan at a different quality and frame rate. To address this, we aim to reduce the transmitted point cloud sequence bitrate by allowing for varying update frequencies and allocating different amounts of bits to various objects in each point cloud scan based on their importance. This is achieved by assigning different criticality levels to objects (set of points) in the point cloud. A primary component of the proposed method involves determining the criticality levels of different objects in the point cloud. This information is then utilized to optimize the compression and delivery of the point cloud.

The proposed adaptive point cloud compression method is not limited to content with multiple objects. It can also be applied to single-object point clouds, as different areas of an object can exhibit varying levels of importance. For instance, in a single-object point cloud featuring a singer, the singer's face will likely be the most visually salient area for viewers, justifying higher prioritization during compression.

5.1.1 *Criticality Assignment*

Traditionally, point cloud compression has primarily focused on preserving the integrity and quality of the point cloud data [200], often to optimize the quality [171, 192, 204]. However, from a practical standpoint, the content of a point cloud can vary greatly depending on the specific application. As such, it is important to consider the intended purpose of the point cloud data at the receiving end when designing adaptive streaming mechanisms. For example, consider exchanging point cloud data between multiple vehicles equipped with 3D sensors from various locations [197]. These sensors are inherently limited by susceptibility to occlusion and loss of detail for distant objects. Combining the vehicles' individual views allows for a more comprehensive and higher-resolution view, significantly improving perception quality for the participating vehicles [197]. To further clarify this matter, in the case of point cloud data intended for localization tasks, it may be more beneficial to remove points from moving objects during the encoding phase in order to decrease the number of noisy points, improve localization accuracy, and save on bitrate, as demonstrated by Zhao et al. [200]. On the contrary, the optimal approach for applications involving point clouds in traffic surveillance may involve removing points related to stationary objects during the encoding phase. This step can effectively decrease the number of irrelevant points, ultimately improving the accuracy of the intended surveillance

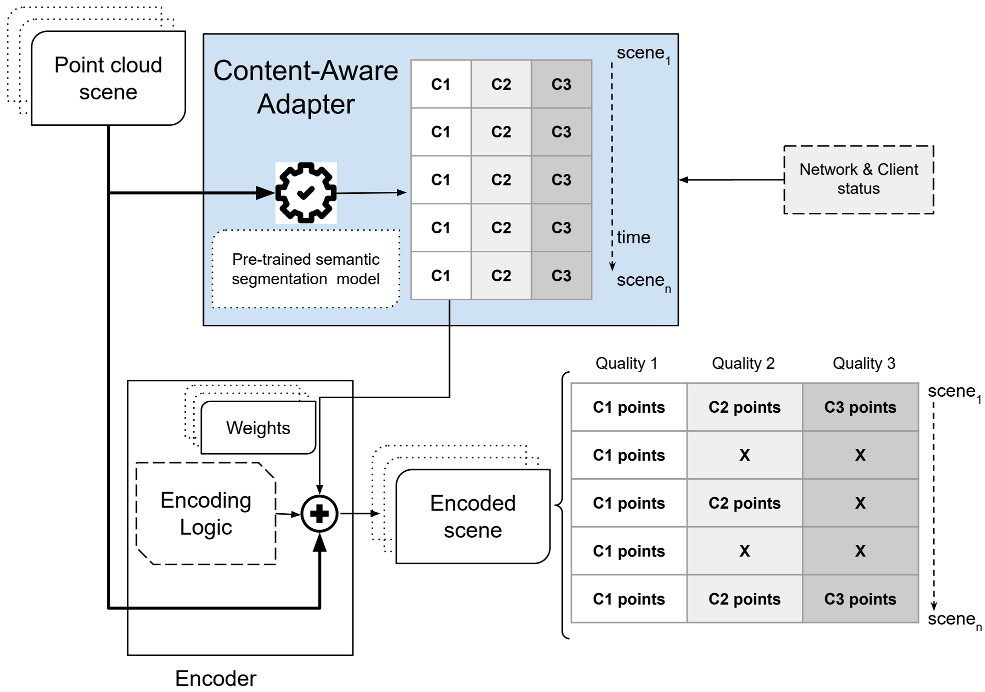


Figure 5.3: Detailed view of our proposed content-aware adapter. The point-wise semantic segmentation allows assigning a criticality level to each point (C1, C2, and C3 are criticality levels in the figure). The specific assignment of criticality levels to object classes depends on the specific application in question. This enables the encoder to assign different quantization levels and/or *frame rates* to points depending on their criticality level. X means no data.

task. Therefore, our proposed approach shifts the focus from maximizing Level Of Detail (LoD) to optimizing point cloud content for specific use cases.

As part of our approach, we assign different criticality levels to objects' points in the point cloud based on their importance to the user's attention or the informative part of the scan. Our strategy is to assign more weight to the most informative objects essential for the tasks being carried out by the point cloud application. By considering the objectives and semantics of the intended point cloud-based application, we can ensure that the reduction in quality primarily impacts less relevant points while preserving the most critical information during point cloud data compression. The proposed approach allows for flexibility in assigning criticality levels to the point cloud content, as different levels can be offered depending on the goals and semantics of the planned point cloud-based application. This approach enables content-aware compression, resulting in higher efficiency compared to traditional blind compression methods. This ensures that crucial information is preserved while minimizing transmitted data. Therefore, we propose incorporating metadata associated with the point cloud to indicate the varying levels of importance of different objects within the point cloud. One example of such information could be a hierarchy of the objects' criticality in the scene. By leveraging this information, we establish criticality classes for the objects in the point cloud. For instance, an automotive point cloud scan may comprise multiple objects, such as vegetation, buildings, and other vehicles. While some of these objects may be essential for the intended task, such as identifying other vehicles on the road, other stationary objects, such as vegetation and buildings in an urban street scene, may not contribute to the task. By assigning criticality classes based on this information, we can selectively compress the most informative objects while discarding the less important ones, thus reducing the transmitted data and improving the overall performance [22, 197].

One approach to obtain semantic information for point cloud-based content is to utilize deep learning models. There are various methods for analysing semantic information of point clouds [30, 61, 188, 195]. However, to our knowledge, this is the first work demonstrating how to utilize this information in adaptive point cloud delivery. Figure 5.3 provides an overview of the criticality assignment process. Here, a pre-trained semantic segmentation model is used to semantically partition a scene with multiple objects into different criticality levels. Each point in the scene is then assigned a criticality label based on the semantic information obtained from the model. This enables selective compression of the point cloud data, with the most informative objects encoded in a high quality level and the less important ones assigned a lower quality. This ultimately may lead to more efficient point cloud delivery.

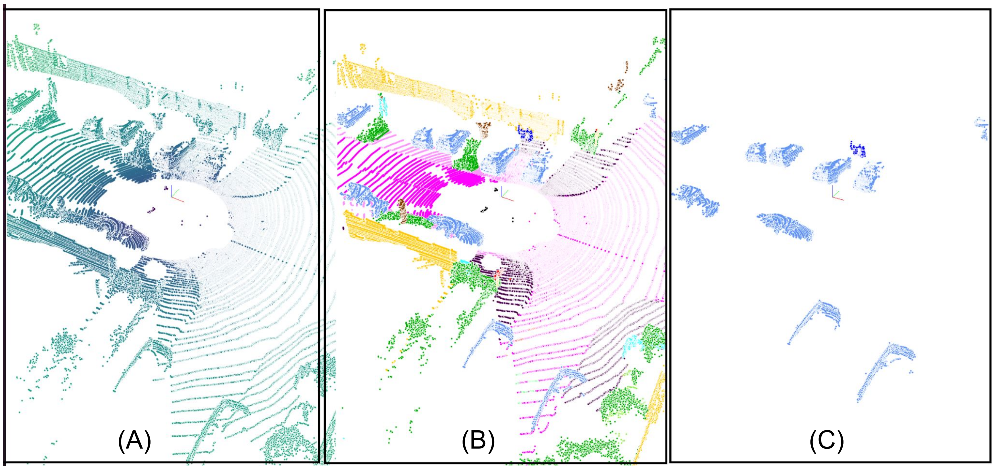


Figure 5.4: Visualization of one scan of the SemanticKITTI dataset: (A) The scan with no annotation, (B) The scan with annotation, (C) The scan after adaptation, in this specific case, only the most critical objects, i.e., cars (or pedestrians), were considered.

As seen in Figure 5.4 (A) and (B), an example of the difference between an automotive point cloud scan before and after applying semantic segmentation can be observed. Figure 5.4 (A) displays a raw scan, while Figure 5.4 (B) shows the same scan after applying semantic segmentation. In Figure 5.4 (B), each point in the scan belonging to a car object is coloured blue, and each point belonging to a road is coloured pink. In Figure 5.4 (C) displays only the critical objects in the scan, which are cars in this case. By gaining this point-wise semantic knowledge, we can estimate the relevance of each object in the scan, which allows for more efficient and effective point cloud compression and delivery. This semantic segmentation provides a way to identify the objects present in the point cloud scan and their relative importance, which can then be used to selectively compress the data. By assigning higher criticality levels to the more informative objects and lower levels to the less important ones, we can effectively transmit only the most relevant data, allowing optimized bitrate utilization.

5.1.2 *Adaptive Quantization and Frame-rate*

Our goal is to adapt the content bitrate to the available bandwidth while ensuring that the overall quality and the intended task are not compromised when delivering point cloud streams to users. To achieve this, during the exchange of point cloud scans, the sender adjusts the bitrate to match the network conditions. By reducing the bitrate when the network connection deteriorates and increasing the bitrate to provide a richer experience when the network connection improves. For instance, cars with limited network connectivity, streaming point clouds effectively requires reducing the bitrate. With our proposed approach, this reduction prioritizes details of pedestrians, vehicles, and buildings, while less critical background elements can undergo heavier compression. The rate control component of the encoder is responsible for allocating bits to scans under the available bit budget. To achieve a balance between the average encoding bitrate and the bandwidth requirements, we propose to use three levels of granularity for allocating bits: (i) group of scans level, (ii) scan level, and (iii) object level. Group of scans level refers to adjusting the bitrate for a group of point cloud scans together. Scan level involves adjusting the bitrate for individual point cloud scans, allowing for finer-grained control based on the content of each scan. The object level is the most granular level, where the bitrate is adjusted for specific objects within a point cloud scan, enabling selective compression based on object importance. The greater the granularity, the higher the level of bitrate control, which allows for a more fine-tuned adaptation of the point cloud streams to the changing network conditions. By increasing the granularity, the encoder can better understand each object within the point cloud and quantize its data while maintaining the overall quality. Therefore, knowledge of the weights or content awareness is a piece of crucial information that can be exploited to increase the granularity of content bitrate control. However, it is important to note that this increased content awareness also adds complexity to the rate control procedure, as bits must be distributed in a way that accurately represents the importance of each object while still maintaining the target bitrate and avoiding large spatial and temporal fluctuations.

To effectively adapt the bitrate requirements, our approach allows non-informative objects in the point cloud to be updated infrequently, use low quantization levels, or even be omitted entirely, depending on the application's needs. By assessing the relevance level of each object within a streamed scene, we can make more informed decisions about compressing and transmitting the data, resulting in a more efficient and adaptable point cloud delivery. This approach of selectively compressing the point cloud data based on the relevance of each object allows for more efficient use of resources and ensures that the most important information is transmitted without compromising on quality or the intended task. In this work, our primary focus will not be on the rate control algorithm but rather on how understanding the content of the point cloud can help achieve the adaptability of the point cloud streams. By identifying the relevance level of each object within a point cloud, we can make more informed decisions about compressing and transmitting the data, resulting in a more adaptable point cloud delivery. While the rate control algorithm plays a crucial role in achieving this goal, our focus will be on demonstrating how understanding the point cloud's content can help achieve more efficient and adaptable streaming.

To prepare objects present in a point cloud scene for adaptation by the encoder, we propose the following approach. First, we define a priority hierarchy for the objects present in the scene. Next, we use semantic segmentation techniques to partition the scene into its semantic objects. Finally, we assign appropriate quantization and frame rate amounts to each criticality class based on the semantic information obtained from the segmentation process.

In summary, our approach involves identifying the relative importance of each object in the point cloud and using this information to selectively compress the data, ensuring that the most important information is transmitted while adapting the data bitrate to the available bandwidth. In this work, we aim to investigate how understanding the content of the point cloud and utilizing this information can lead to more efficient and adaptive delivery. By adapting the compression and transmission of the data based on the relative importance of each object, we can ensure that the most crucial information is transmitted. This ultimately may result in more efficient and effective point cloud delivery, adapting to changing network conditions while maintaining the overall quality and intended task of the point cloud data.

5.2 EVALUATION

Our proposed approach, integrating semantic information with a content bitrate adaptation mechanism to enhance point cloud delivery, requires evaluation. To achieve this, we utilize the SemanticKITTI point cloud datasets [18], providing a dataset of semantically annotated point cloud scans as the foundation for our evaluation. This section will introduce our scenario and experimental setup, shaping the evaluation process. We will demonstrate the performance of our proposed content bitrate adaptation method and analyse its efficiency and runtime compared to a baseline approach. A detailed overview of the implementation details of our proposed method will also be provided. The experimental setup includes a description of the dataset and hardware, the parameters, the settings used for the implementation, and any relevant details for result reproduction.

5.2.1 *Evaluation Dataset*

This study utilizes the SemanticKITTI dataset [18], a large-scale, real-world point cloud dataset originally designed for semantic scene understanding. It provides 22 consistent point-wise semantic annotated point cloud sequences comprising around

43k scans. The data originates from a rotating automotive 3D LiDAR sensor, which covers a 360-degree field of view with a frequency of 10 scans per second. However, it should be noted that the back-facing part of the objects is always occluded. Generated point cloud scans include multiple street objects. Figure 5.4 (A) illustrates a birds-eye perspective of a LiDAR scan, where a variable number of urban objects surround the invisible LiDAR sensor in the center of the image. The dataset includes 28 different urban object classes: car, person, motorcycle, building, sidewalk, and vegetation. Each urban object can be identified as one of the 28 classes, such as different types of ground, structure, vehicle, nature, and human. Each point in the point cloud has attributes that include its (x, y, z) coordinates in a 3D space. The dataset provides annotated, real-world point cloud data that can be used to evaluate and test the performance of our proposed method.

We did not consider a video point cloud dataset similar to what we used in Chapter 4 because we need a semantically labelled dataset for point cloud video sequences. Therefore, we opted for a LiDAR-based dataset, which is more common in automotive use cases [111]. For example, consider exchanging point cloud data between vehicles equipped with 3D sensors from various locations [77, 117, 197]. These sensors are inherently limited by susceptibility to occlusion and loss of detail for distant objects. To address these limitations, raw sensor data from multiple vehicles from various locations can be shared with a nearby edge server, so objects occluded in the views of some vehicles can be easily perceived by others, like, for example, blind spots. Other researchers, such as in [31], tried to share processed data, but these approaches are limited to data granularity, meaning that missed detections will still be missed after sharing. Additionally, shared, processed data need more generality. Therefore, sharing the raw data in a universal format that is compatible with various applications and combining sensor data can lead to a higher resolution and overcome loss of detail for distance objects. A shared server then fuses the individual views into a more comprehensive and higher-resolution representation, significantly enhancing perception quality for all participating vehicles [197]. Another use case involves exchanging raw point cloud data between river vessels to enhance the perception of their surroundings [56]. The case study of automotive point cloud data [111] is chosen because it represents a real-world scenario with multiple objects and complex scenes, thus providing a meaningful and challenging tool for evaluating the proposed method. The case study results will provide insights into the performance of the proposed method in a real-world scenario and demonstrate its effectiveness in adapting the point cloud data transmission to the changing network conditions while maintaining the overall quality and intended task.

5.2.2 *Semantic Segmentation Deep Learning Model*

To achieve adaptability in point cloud delivery, the proposed method utilizes real-time point-wise semantic prediction. This involves employing a deep learning model, namely SalsaNext [33], to analyse the semantic content of the point cloud and assign criticality labels to individual points within the scene. These criticality labels indicate the relative importance of each point, enabling the system to prioritize the transmission of highly relevant information while reducing the bitrate for less critical elements.

Several modifications have been made to the original architecture to adapt the SalsaNext model for point-wise semantic prediction of criticality levels: first, the input to the model is modified to accept criticality labels instead of object labels. This allows the model to focus on identifying the relative importance of each point rather than classifying them into specific object categories. Secondly, a new training dataset is created, where each point in the point cloud is assigned a corresponding criticality label. This training data is used to train the model from scratch, enabling it to learn the relationships between point features and criticality levels. Lastly, the output layer of the deep neural network is modified to predict criticality levels instead of object classes. This ensures that the model's output directly corresponds to the desired information for adaptable point cloud delivery. By implementing these modifications, the SalsaNext model is effectively transformed into a point-wise semantic prediction tool that can analyse the content of the point cloud scan and assign criticality labels to individual points.

5.2.3 *Evaluation Metrics*

To assess the feasibility and effectiveness of our proposed method in the context of exchanging automotive point cloud data, the evaluation of our approach considers two key aspects: (1) The communication cost metric is the average data volume exchanged between two nodes in megabytes per scan. This metric allows us to evaluate the efficiency of our method in terms of the amount of data transmitted between the sender and the receiver. (2) The data processing latency in milliseconds per scan. This metric allows us to evaluate the time it takes to process the point cloud data and prepare it for transmission. By considering these two aspects, we can assess the overall performance of our proposed method in terms of communication cost and data processing latency, which are important factors in real-world point cloud streaming applications.

5.2.4 *Baseline Approaches And Criticality Level Settings*

To evaluate the performance of our proposed method, we use the communication cost and data processing latency metrics mentioned above. Additionally, we compare our method against a baseline that encodes the scans without considering the content semantics. In order to do this, we use two common point cloud encoding mechanisms, Draco[1] and G-PCC[2], in our experiments.

The experiment setup for Draco and G-PCC is summarized in Table 5.1. This table summarizes the parameters and settings used for the experiments, including the encoder configurations and the version used.The parameters section includes various

1 `https://github.com/google/draco` last accessed: December 2, 2024.
2 `https://github.com/MPEGGroup/mpeg-pcc-tmc13` last accessed: December 2, 2024.

Method	Version	Parameters
Draco	DracoPy 1.2.0	Compression level=1
		Quantization Parameter (QP) = 15 / 12 / 10
G-PCC	release-v 14.0	geomTreeType = Octree
		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 5.1: Experiment setup parameters

settings used during the compression process. For the Draco method, the compression level is set to 1, indicating a low compression ratio but with the benefit of faster decompression. Different parts of the point cloud use different Quantization Parameter (QP) values. For example, we use QP values of 15, 12, and 10 to represent three quality levels: low, medium, and high. For the G-PCC method, the geometry tree type is set to *Octree*, a common choice for 3D point cloud compression. Position quantization is enabled, which helps reduce the compressed data size. The sequence and coding scales are set to 0.1, and the input scale is set to 1000. The transform type is set to *Prediction*, and the source and output units are set to *Metre* with a source unit length of 1 and an output unit length of 0.001. The neighbour availability boundary, log2, is set to 8, defining the volume within which octree nodes are available for occupancy contextualization and intra-occupancy prediction. Finally, the output binary Polygon File Format (PLY) is enabled, indicating that the output will be in binary PLY format. Default values are applied for parameters not listed in Table 5.1. For more details, please refer to the official documentation available online[3]. By comparing our method against the baseline, the two commonly used encoding mechanisms, we can evaluate its performance and demonstrate its effectiveness in improving the adaptability and efficiency of point cloud data transmission.

Since the dataset has a known number of object classes, we create a three-level hierarchy, i.e., criticality levels, based on object relevance. This is exemplified by the most relevant objects, such as cars and pedestrians, which fall under criticality 1. Road objects, like road signs and side-walks, are less critical and fall under criticality 2. The last critical level, i.e., criticality 3, includes all other object classes, including buildings and vegetation. Figure 5.5 illustrates these criticality levels in a priority hierarchy. The higher the criticality, the more relevant this example hierarchy is, and

3 `https://github.com/MPEGGroup/mpeg-pcc-tmc13/blob/master/doc/README.options.md` last accessed: January 30, 2024.

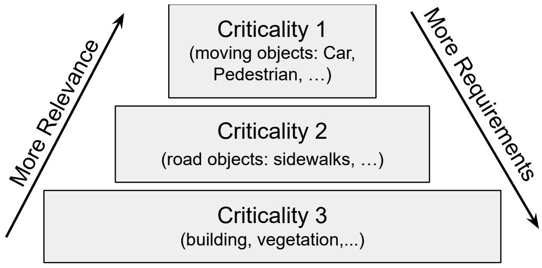


Figure 5.5: Utilizing the object's level of dynamicity in an automotive scan exemplifies that the priority of objects can be ranked in a hierarchy.

fewer requirements like bandwidth and computation are needed. Each criticality level includes the previous one; for example, 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. The hierarchy, which should be defined in advance, is application-dependent and ought to reflect the relevance of points within the transmitted data.

As a visual validation, referring back to Figure 5.4 (C), which shows a scan after applying adaptation with only objects belonging to criticality 1. This leads to significant bandwidth savings assuming that removing less relevant objects does not compromise the intended task. To illustrate this further, Figure 5.6 shows a point cloud sequence throughout around 4017 scans. The figure illustrates the bandwidth requirements for each class. By decomposing the original stream into multiple streams, our approach enables adaptation, allowing data to be adapted based on resource constraints, such as bandwidth in this case. This enables each cumulative semantic group of points to have different bandwidth requirements. This highlights the importance of assigning different criticality levels to different objects points, as it enables adaptivity in point cloud delivery and ensures that the most important objects are transmitted with high quality. In contrast, less important objects are transmitted with lower quality or omitted altogether, depending on the network conditions.

5.2.5 *Evaluation Results and Comparisons*

This section is dedicated to the experimentation of the proposed method regarding our implementation. In the experimental study, we apply our approach to sequence number 8 from the SemanticKITTI dataset [18]. This sequence comprises 4071 scans, each with an average size of 1.34 megabytes and a standard deviation of 76 kilobytes. To evaluate the performance of our approach, we measure the scan size and the encoding and decoding latency time (LT) for each criticality group.

Bandwidth Savings

Our approach reduces the transmitted point cloud sequence bitrate by enabling differentframe rates or allocating different amounts of bits to scan points according to the assigned criticality level of the scan points. This is compared to a baseline sce-

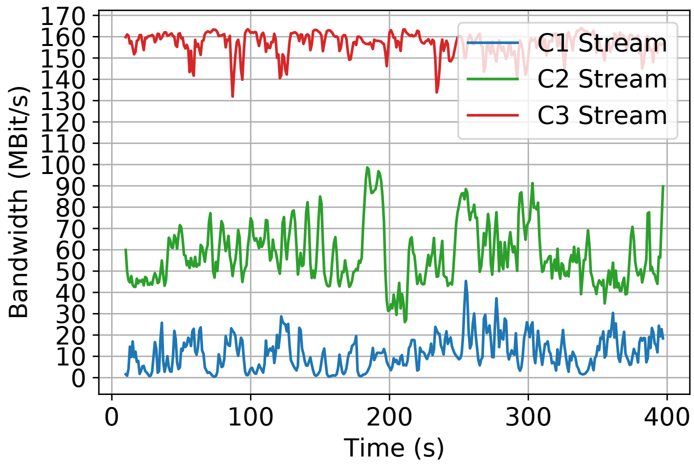


Figure 5.6: Bandwidth required by the points of each criticality level. Enable point cloud sequence adaptability by applying semantic segmentation. Allowing each cumulative semantic group of points to require a different bandwidth. This data was collected throughout around 4017 scans. The bandwidth fluctuates due to changing point density generated at a given point in time.

nario, where the scans are compressed with the same frame rate and a *quantization parameter*.

To demonstrate the effectiveness of our approach, we start by showing the bandwidth saving that can be achieved by applying different *frame rates* based on the significance of the transmitted points. The Figure 5.7 compares bandwidth usage across various data streams before and after implementing adaptive *frame rates* based on criticality levels. It shows 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 X-axis represents time in seconds (ranging from 0 to 400 seconds), while the Y-axis denotes bandwidth in MBit/s. The c1 (Criticality Level 1) stream, representing data with the highest criticality, maintains a bandwidth of around 2 MBit/s in both figure segments, indicating no applied adaptation. In the c2 stream, in the left part (without adaptation), this stream's bandwidth fluctuates around 5 MBit/s. After applying adaptation (right part), the bandwidth reduces to approximately 3 MBit/s due to a frame rate decrease from 10 to 5 frames per second. In the c3 steam, looking at the left part, the bandwidth of the stream fluctuates around 12 MBit/s. Following adaptation, the bandwidth significantly drops to around 3 MBit/s due to a harsher reduction in frame rate, from 10 to 2 frames per second. The joint stream represents the combined bandwidth usage of all three streams (C1 + C2 + C3). Without adaptation, the total bandwidth fluctuates around 20 MBit/s, while after adaptation, the joint bandwidth drops significantly to approximately 7 MBit/s, showcasing substantial savings. Adapting *frame rates* based on criticality levels leads to significant bandwidth savings, particularly for less critical data streams. Among them, the c3 stream experiences the most substantial reduction in bandwidth due to the applied harsh adaptation. The jointly represented stream demonstrates the overall effectiveness of the approach by exhibiting a substantial drop in bandwidth after adaptation.

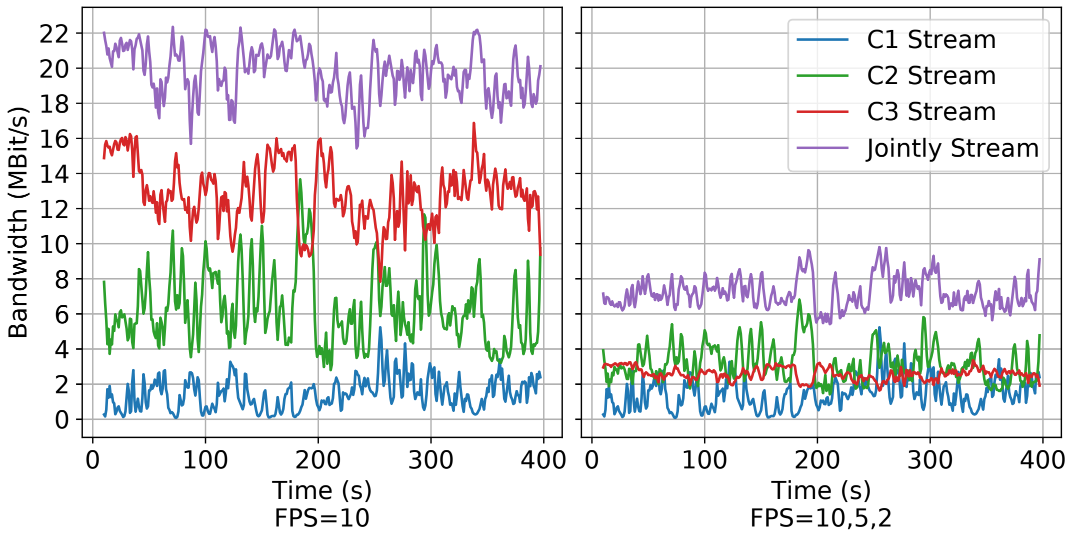


Figure 5.7: Bitrates savings by adopting *frame rates* 10, 5, and 2 for criticality 1, 2, and 3 points, respectively, against fixed frame rate for all points.

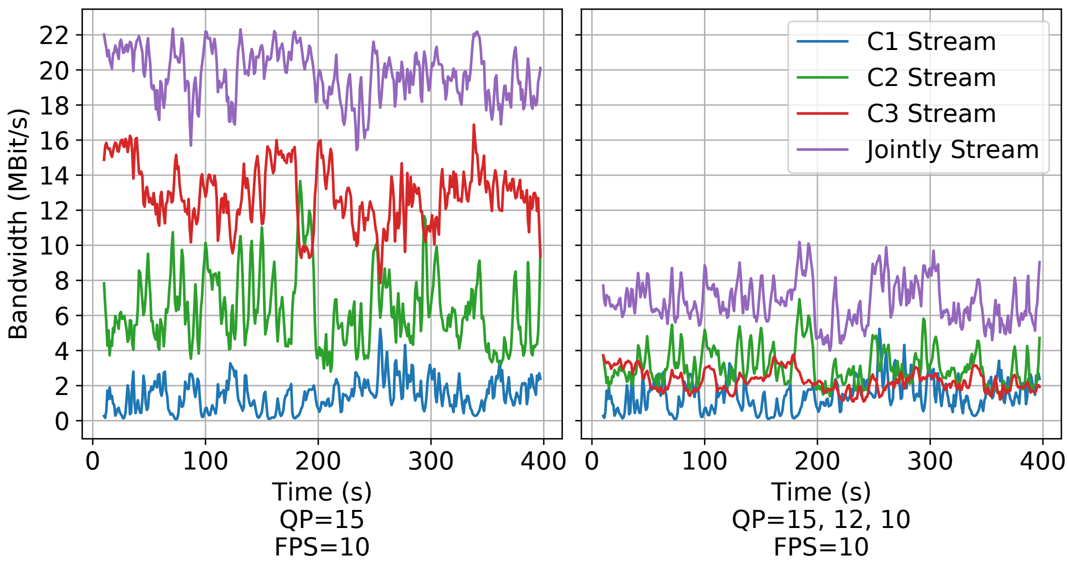


Figure 5.8: Bitrates savings by adopting *quantization parameters* (QP) 15, 12, and 10 for criticality 1, 2, and 3 points, respectively, against fixed *quantization parameter* 15 for all points.

Next, we show the bitrate savings achieved by applying different quantization levels per criticality points compared to a baseline with a fixed *quantization parameters* for all the scan points. The compression algorithm employed here is Draco. In Figure 5.8, we plot the bitrate achieved when the *quantization parameters* 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. Additionally, As in the previous adaption, C1 stream bandwidth usage remains constant at around 2 MBit/s in both parts of the figure, indicating no adaptation applied. C2 stream without adaptation (in the left part), its bandwidth fluctuates around 5 MBit/s. After applying adaptation (right part), the bandwidth is reduced to approximately 2 MBit/s due to a decrease in quantization level from 15 to 12. C3 stream, in the left part, its bandwidth fluctuates around 12 MBit/s. After adaptation, the bandwidth drops significantly to around 3 MBit/s due to a harsher quantization level (from 15 to 10). Jointly stream represents the combined bandwidth usage of all three streams (c1 + c2 + c3). In the left part (without adaptation), the bandwidth fluctuates around 20 MBit/s. After adaptation, the joint bandwidth is significantly reduced to approximately 6 MBit/s, demonstrating the overall effectiveness of the approach, this method provides fine-grained control over the quality of the point cloud based on criticality levels. It allows for dynamic adaptation of specific parts of the point cloud, enabling *transitions* between different frame rates (e.g., from 10,10,10 FPS to 10,5,2 FPS). This ensures that all data is transmitted (if necessary) while prioritizing the more important sections.

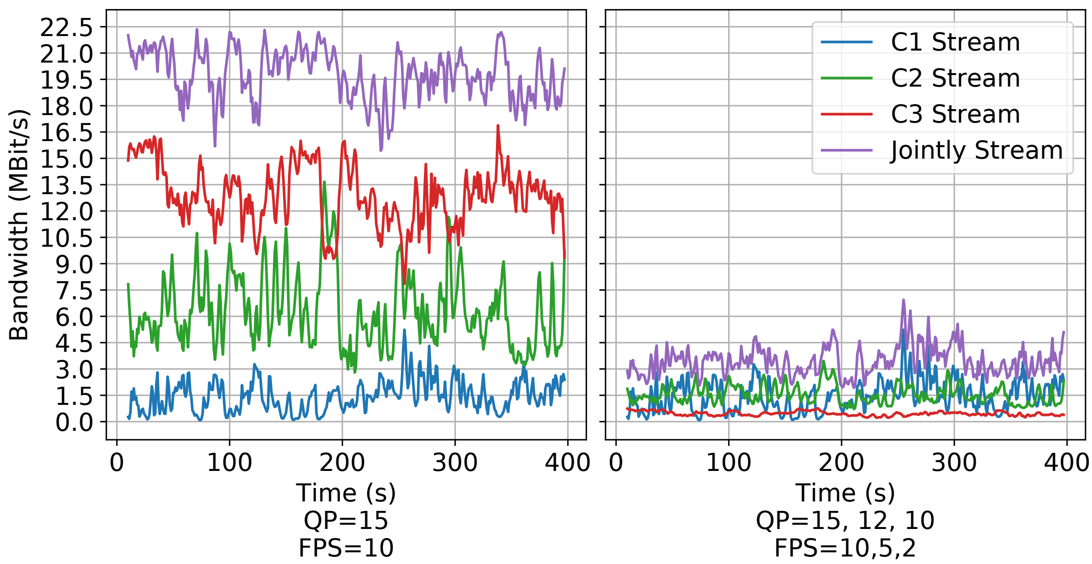


Figure 5.9: Bitrates savings by adopting *frame rates* (FPS) 10, 5, and 2, and *quantization parameters* (QP) 15, 12, and 10 for criticality 1, 2, and 3 points, respectively, against fixed frame rate and *quantization parameters* for all points.

We also show in Figure 5.9 the bitrate savings achieved by applying different quantization levels and *frame rates* per criticality points in comparison to a baseline with fixed *quantization parameters* and frame rate for all the scan points. These figures demonstrate that our content-aware adaptation approach can achieve signif-

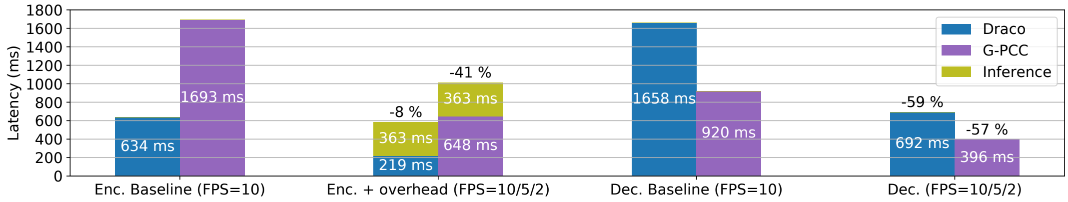


Figure 5.10: A comparison of encoding and decoding latency under different *frame rates*. The bar in the graph represents the average encoding/decoding latency across all *frame rates* for the evaluated sequence of the dataset. The inference latency attributed to criticality assignment.

icant bitrate savings compared to the classic Draco baseline. This figure compares the bandwidth usage of different data streams before and after applying adaptive quantization levels and *frame rates* based on criticality levels. C1 bandwidth usage remains constant at around 2 MBit/s throughout the figure, as no adaptation is applied (fixed quantization level of 15 and frame rate of 10). For the C2 stream, the bandwidth in the left part (without adaptation) fluctuates around 5 MBit/s, with original values of a 15 quantization level and a 10 frame rate. In the right part (with adaptation), the bandwidth reduces to approximately 3 MBit/s due to a combined adaptation of decreased quantization level (15 to 10) and lower frame rate (10 to 5). For the C3 stream, the bandwidth in the left part (without adaptation) fluctuates around 12 MBit/s, with original values of a 15 quantization level and a 10 frame rate. In the right part (with adaptation), the bandwidth significantly drops to around 1 MBit/s due to a combined adaptation of a harsher quantization level reduction (15 to 10) and a lower frame rate (10 to 2). Overall, our approach outperforms the classic Draco baseline by achieving 65% to 82% lower bitrates.

Influence on Computational Resources

Our approach reduces the transmitted point cloud sequence bitrate and improves the encoding and decoding latency time by applying different *frame rates* for the points based on their assigned criticality level. This is demonstrated in Figure 5.10, where we compare the encoding and decoding latency of two common point cloud encoding mechanisms, Draco and G-PCC, under different *frame rates*. Encoding latency is the time to compress a point cloud frame, and decoding latency is the time to decompress it. The results show that using adaptive *frame rates* results in lower latency than a baseline scenario with a fixed frame rate for all points. Additionally, it is worth noting that the average SalsaNext inference time per scan is 4560 milliseconds on the CPU and 36.19 milliseconds on the GPU. This further highlights the potential of our approach in reducing computational requirements when adaptation is necessary to meet computational constraints. It is worth mentioning that when full quality is used, our method might introduce unnecessary inference latency. In that case, the inference component can be deactivated.

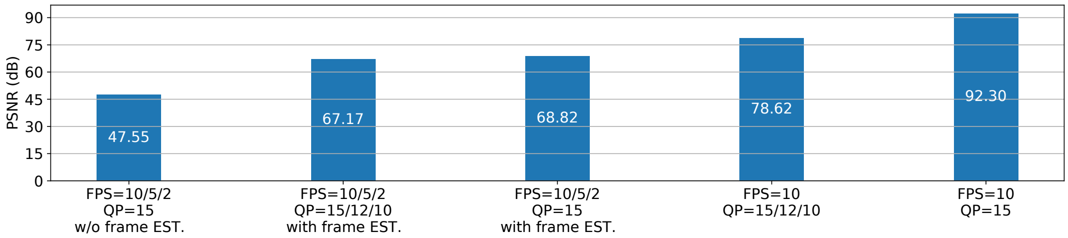


Figure 5.11: A comparison of average PSNR under different adaptations. Higher values are better. The first bar from the right represents the PSNR when using fixed frame rate and fixed quantization, serving as a reference point.

The figure analyses how compression techniques impact processing speed, considering fixed and adapted *frame rates*. Draco and G-PCC are compared at a fixed frame rate (10 FPS) for encoding and decoding times. Draco encodes faster (634 ms) than G-PCC (1693 ms), but G-PCC decodes quicker (920 ms) compared to Draco (1658 ms). Interestingly, adapting the frame rate (10/5/2 FPS) with our exemplary criticality levels (C1, C2, C3) significantly improves the average encoding times (Draco: from 634 seconds to 582 seconds, G-PCC: from 1693 seconds to 1011 seconds) and decoding times (Draco: from 1658 seconds to 692 seconds, G-PCC: from 920 seconds 396 Y seconds) for both methods. This highlights the potential benefits of frame rate adaptation, even when factoring in the overhead generated by semantic segmentation inference calculations.

Quality Evaluation

To evaluate the impact of our proposed method on the quality of the scanned point cloud data, we measure the quality using Peak Signal-to-Noise Ratio (PSNR). PSNR is a widely-used objective quality measurement tool[4] that compares the original and modified point cloud data. By using the PSNR metric, we can quantify the difference in quality between the original and the adapted point clouds. This allows us to understand how the criticality-based adaptations affect the quality of the point cloud data and how it compares to the baseline scenario where no adaptations are applied.

Maintaining high quality for critical points is crucial, but it is also good to maintain decent quality for less important points. We propose a technique that can be applied at the receiver side to mitigate the loss of quality due to a reduced frame rate. The technique involves introducing in-between frames, which are created by estimating the positions of 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, which can be used to generate these in-between frames. Additionally, using the LiDAR's position and orientation, piggybacked on higher criticality frames, has minimal impact on bandwidth and computing cost.

4 `https://github.com/MPEGGroup/mpeg-pcc-tmc2` last accessed: February 1, 2024.

In Figure 5.11, we present the average PSNR measurements achieved by our approach across the entire point cloud sequence under different adaptations. The results demonstrate that the overall quality of the point cloud data, as estimated by the PSNR metric, remained relatively high despite the adaptations made to the data. This results from our content-aware approach, which prioritizes the preservation of high-criticality points while still maintaining good quality for less relevant points.

Implementation Details

Our point cloud semantic segmentation model implementation was built using Python 3.7 and PyTorch 1.1. The training was conducted on a server running Ubuntu 18.04, equipped with two NVIDIA RTX 2080 GPUs and an Intel Xeon Silver 4112 CPU. The model inference was executed on a Windows 10 based system, using a single RTX 3080 Ti in combination with an AMD Ryzen 7 5800X CPU. The model's architecture is mainly based on SalsaNext, with some modifications to adapt it to our specific use case. We reduced the batch size to six to optimize the model's performance and turned off the K-Nearest Neighbors (KNN) post-processing. The original label configuration was replaced with criticality labels instead of object labels. The training process was carried out over 150 epochs, with an initial learning rate of 0.01 using Stochastic Gradient Descend (SGD). For visualizing the LiDAR point clouds, we leveraged the API for SemanticKITTI [5]. All scripts and resources used to reproduce the results of this paper will be made publicly available on GitHub [6] for further study and replication.

5.3 DISCUSSION AND CONCLUSIONS

In this section, we discuss the results and interpret findings obtained from enabling point cloud adaptation based on content awareness. In this chapter, we addressed the research goal: *Investigating the impact of incorporating object-related information in point cloud content streaming on the quality and efficiency of the delivery process compared to conventional methods*. We proposed a content-aware adaptor that integrates semantic and object-level information. Our method enables fine-grained control over the quality of the point cloud, depending on the criticality levels originating from user preferences or specific application requirements. It allows for dynamic adaptation of certain parts of the point cloud (e.g., transitioning from 10,10,10 FPS to 10,5,2 FPS) so that the overall data is still transmitted if necessary, while prioritizing the important parts. This fine-grained control allows for bandwidth reduction by allocating higher bitrates to critical objects for enhanced quality, while less critical objects are assigned lower bitrates. We evaluated our proposed adaptor using real-world data and exemplary criticality assignment, resulting in significant bandwidth savings and latency improvements compared to using fixed frame rate and fixed quantization

5 `https://github.com/PRBonn/semantic-kitti-api` last accessed: December 2, 2024.
6 `https://github.com/yaseenit/Content-Aware-Adaptive-Point-Cloud-Delivery.git` last accessed: December 2, 2024.

level for the whole point cloud. Our approach effectively lowered the required bitrate of transmitted point cloud sequences by adjusting frame rates and quantization levels based on assigned criticality levels. This results in bandwidth savings, particularly for less critical data streams. With our exemplary criticality assessment, our results showed that our approach can achieve significant bitrate savings from 65% up to 82% compared to classic Draco baseline. We further showcased the benefits of our content-aware adaptation approach by illustrating its positive effects on encoding and decoding latency. By adjusting frame rates based on criticality, we achieved lower latency, enhancing the processing speed for point cloud streaming. For encoding, our approach achieved a latency reduction of 8% for Draco and 41% for G-PCC, with our exemplary criticality assessment. Similarly, for decoding, our approach resulted in a latency reduction of 59% for Draco and 57% for G-PCC. Our findings emphasize two key points: (1) adaptive frame rates lead to decreased encoding and decoding latency compared to a fixed frame rate approach, resulting in faster processing times. (2) despite the additional overhead of criticality assignment, frame rate adaptation enhances overall processing speed, potentially reducing the computational resources required for streaming. Additionally, we addressed the potential impact of our content-aware adaptation on point cloud data quality. Our approach prioritizes maintaining high PSNR for critical points while permitting some quality reduction in less critical areas to balance bandwidth savings with acceptable quality levels. To mitigate potential quality loss, we introduced in-between frames by leveraging relative position and orientation information from Light Detection and Ranging (LiDAR) to estimate missing points between actual frames on the receiver side. This technique offers improved visual quality in areas with less critical data while incurring minimal overhead. The overall PSNR of the adapted scans remains relatively close to the PSNR of the original scans despite the adaptations, indicating successful quality preservation through prioritization and in-between frame interpolation. Overall, our content-aware adaptation proves to be a valuable mechanism for optimizing point cloud streaming efficiency, including bandwidth and computational resources, without significantly compromising quality.

While the presented findings presented offer valuable insights, it is important to acknowledge the limitations of this work. For example, the effectiveness of this approach relies heavily on accurately assigning criticality levels to different objects. Consequently, if the semantic segmentation model performs poorly, this could result in incorrectly assigned criticality levels and yield undesirable outcomes. Moreover, a limitation relates to the need for a manual definition of the hierarchy of importance within the streaming system. The other limitation concerns computational cost. Assigning criticality levels and potentially performing real-time adaptation can add computational overhead, impact processing speed, and potentially negate some of the efficiency gains. Also, the overall system becomes more complex with additional components for criticality assignment and adaptation. This can make it more challenging to implement and maintain. While the approach aims to minimize quality loss, reducing the bitrate and frame rate for less critical areas will still decrease visual quality. This might be unacceptable for specific applications requiring high fidelity across the entire scene. The technique of introducing in-between frames to improve visual quality might introduce artefacts (distortions or inconsistencies) if the estimation of missing points is not accurate. The suggested content-aware model focuses solely on encoding and decoding each raw input frame, implying that two consecutive frames are transmitted similarly without considering the motion and spatial relationships between them. A limitation could also appear when streaming in full quality. In this case, point cloud analysis, i.e. criticality assignment, is not required, and if it is kept active, it introduces inference latency without any significant gains. In addition, the current evaluation relies on a specific real-world dataset. This dataset might need to be more generic to represent the full range of potential use cases for point cloud streaming. For example, the dataset might not include the requirements of virtual reality applications, which often demand high fidelity across the entire scene.

Future work could address some of these limitations. It could use machine learning techniques to dynamically assign criticality levels based on real-time context. This could involve analysing the scene content, user preferences, and application requirements to adjust criticality on the fly. It can also develop mechanisms to incorporate user feedback into criticality assignments. This could allow users to specify which objects they consider important, improving personalization. Future work could involve evaluating the overall perceived quality when the adapted scene has different qualities within it. Evaluating our proposed approach using datasets that reflect various use cases, including virtual reality, could also a future work direction. This will provide a more comprehensive understanding of the approach's effectiveness in different scenarios.

5.4 SUMMARY

The chapter investigates the impact of incorporating object information on the quality and efficiency of point cloud delivery compared to conventional methods, specificity, Draco and Geometry codec based Point Cloud Compression (G-PCC). A content-aware adaptor is proposed, which integrates semantic and object-level details. The method prioritizes critical objects based on user preferences or application requirements, reducing bandwidth and improving latency. Real-world data evaluation demonstrates significant bandwidth savings and latency improvements compared to traditional approaches. The approach effectively reduces required bitrate transmission by adjusting frame rates and quantization levels , particularly for less critical data streams, based on a criticality assessment using a deep learning model. Furthermore, the method positively affects encoding and decoding latency, enhancing processing speed. Overall, this work represents the initial step towards demonstrating the feasibility of content-aware adaptation for enhancing point cloud delivery efficiency.

6 CONCLUSIONS AND OUTLOOK

To conclude our work, we provide a summary of the content covered in the previous chapters and highlight the main contributions of this research. Additionally, we identify potential areas for future work.

6.1 SUMMARY OF THE THESIS

In Chapter 1, we described the challenges for point cloud delivery and provided insight into the proposed methodology for adaptive point cloud delivery. We motivated the potential benefits of using the semantic information of the streamed content. Following the motivation and challenges, we described the research goals related to the aforementioned task in detail. In Chapter 2, the background and context of the research are established. A comprehensive analysis of relevant literature to our contributions is presented. In Chapter 3, we proposed a tool to evaluate adaptive point cloud streaming. In Chapter 4, we addressed the first and second research goals. A user study was conducted to evaluate compressed point cloud sequences. This study helped to understand the trade-offs between resource requirements and perceived quality. Additionally, a QoE model was developed based on the user study results. In Chapter 5, the third research goal was investigated. The focus was on the impact of incorporating object-related information within point cloud content streaming. The research explored how this approach affects the quality and efficiency of the delivery process. In the following, we present the contributions of this thesis.

6.1.1 *Contributions*

For the first research goal: *"Investigating the impact of compression-related distortions and reduced frame rate on the quality of point cloud objects"*. This work established an understanding of how different adaptations affect the quality of compressed point cloud sequences. This knowledge is a prerequisite for creating adaptive methods for streaming such sequences. Therefore, we studied how factors like quantization levels, distances from the camera, and frame rates influence point cloud quality perception. To grasp the user's experience, we conducted a subjective study, comparing two popular compression methods, Draco and Video codec based Point Cloud Compression (V-PCC), to see how they balance resource usage and quality levels. We found that both the quantization level and frame rate play important roles in determining video quality and customer satisfaction. Our study revealed that when the quantization level is low, the impact of frame rate on quality is minimal. However, as the quantization level increases, the frame rate begins to positively influence quality. Additionally, we observed an inverse relationship between distance and Mean

Opinion Score (MOS), indicating that greater distance leads to lower MOS scores. When we applied the V-PCC compression method to point cloud sequences with participants close to the object, MOS consistently increased as the quantization level increased. Interestingly, the influence of frame rate was more noticeable in specific scenarios, especially when the frame rate was lower (10 and 15 FPS) at medium distances, resulting in a noticeable decrease in MOS compared to the near setting.

Building upon the user study results, we investigate the second research goal: *"Investigating the correlation between quality and resource demands, with the objective of developing a predictive model for evaluating the quality of point cloud sequences"*. Although Draco and V-PCC MOS for our selected configurations were similar, there was a significant difference in the bit rates for the compressed streams. Notably, Draco's encoding and decoding times were considerably faster than V-PCC's. Our study revealed that different configurations can achieve similar MOS while requiring varying bit rates, encoding, and decoding latency. We also found that increasing frame rates and quantization levels individually improved perceived quality. However, we observed that a higher bitrate did not necessarily lead to a higher MOS. Additionally, the degradation in quality due to quantization became less noticeable with increasing distance from the camera. Expanding on that, we utilized machine learning to create a model for the MOS of point cloud sequences and assessed its accuracy by comparing its predictions to MOS obtained from the user study. By providing various factors like precision and geometry quality to the machine learning models, we aimed to predict MOS for configurations not encountered before. The results revealed that the frame rate emerged as the most crucial feature influencing the MOS. Furthermore, our analysis provided additional insights by predicting score distributions using a classification model rather than relying solely on the actual MOS scores.

For the third research goal: *"Investigating the impact of incorporating object-related information in point cloud content streaming on the quality and efficiency of the delivery process compared to conventional methods"*. We introduced a content-aware adapter that incorporates both semantic and object-level details. Our approach focuses on prioritizing important objects based on either user preferences or specific application needs. This prioritization enables us to reduce bandwidth by allocating higher bitrates to critical objects, ensuring better quality while assigning lower bitrates to less critical objects. As a result, overall bandwidth usage decreases, and latency improves because crucial data is transmitted faster, leading to lower latency. We tested our proposed adapter using real-world data, which resulted in significant savings in bandwidth and improvements in latency compared to traditional streaming methods. By adapting frame rates and quantization levels based on assigned weights, our approach effectively reduced the bitrate of transmitted point cloud sequences. This led to bandwidth savings, especially for less critical data streams. With exemplary criticality assessment, our results demonstrated that our approach can achieve substantial bitrate savings, ranging from 65% up to 82% compared to the classic Draco baseline. Furthermore, we highlighted the benefits of our content-aware adaptation approach by showing its positive effects on encoding and decoding latency.

By adapting frame rates based on weights, we achieved lower latency times, enhancing the processing speed for point cloud streaming. Specifically, our approach reduced encoding latency by 8% for Draco and 41% for G-PCC with our exemplary criticality assessment. Similarly, for decoding, our approach resulted in a latency reduction of 59% for Draco and 57% for G-PCC. Our findings underscore two key points: Firstly, adaptive frame rates lead to decreased encoding and decoding latency compared to a fixed frame rate approach, resulting in faster processing times. Secondly, despite the additional overhead of criticality assignment, i.e. point cloud analysis, frame rate adaptation enhances overall processing speed, potentially reducing the computational resources required for streaming. Furthermore, we explored how our content-aware adaptation could affect the quality of point cloud data. Our method focuses on maintaining a high PSNR for crucial points, while allowing for some reduction in quality in less critical areas.To address potential quality loss, we introduced in-between frames using information from LiDAR to estimate missing points between actual frames on the receiver side. This enhances visual quality in areas with less critical data while incurring minimal overhead.

6.2 FUTURE OUTLOOK

The user study was relatively small, with only a hundred participants, and the scores provided by users in a controlled environment may not fully represent quality in real-world settings and may not capture the full range of factors present in real-world scenarios. Larger study with a more participant pool could provide more statistically significant and generalizable results. Future research could expand on this by using larger point cloud datasets and comparing results between standard desktop PCs and more immersive devices like VR/AR headsets. Future work could investigate more compression protocols and parameters, considering not only the bitrate-quality trade-off but also factors like computational complexity and decoding latency. We could further refine our understanding of quality-influencing factors and create a more general quality model by leveraging machine learning methods with richer datasets.

Further exploration could involve using machine learning methods to dynamically assign criticality levels based on real-time context, incorporating user feedback into criticality assignment, and investigating alternative methods for data adaptation beyond frame rate and quantization adjustments. This might include techniques like downsampling point clouds for less critical areas. Future work could also involve evaluating the approach using datasets that reflect a broader variety of use cases, including virtual reality, to gain a more comprehensive understanding of its effectiveness in different scenarios. This will allow for a more comprehensive evaluation of our content-aware approach efficiency gains across diverse use cases.

Building on the findings of this work on content-aware adaptation for improved point cloud delivery efficiency, future research could investigate mechanisms transitions for handling varying network and receiver conditions in the context of the

Multi-Mechanisms Adaptation for the Futrue Internet (MAKI) to further enhance the adaptability and efficiency of the streaming system.

Our proposed content-aware approach is strongly application-dependent. A content-aware method that optimizes point cloud quality for autonomous driving application might not be suitable for for virtual reality application. Developing generalizable content-aware methods that can be adapted to different applications remains an open research question. Future research could address this by investigating methods for transferring knowledge between applications or by exploring domain-agnostic features for point cloud analysis.

Future research should focus on the integration of advanced AI methods, such as Generative AI and Large Language Models (LLMs), which are significantly transforming key sectors of video-related research fields [202]. However, this advancement also introduces unique challenges, including the need for large-scale datasets, and high computational demands. Consequently, academia must approach the rapid development of these advancements. Future work might aiming to develop advanced adaptive approaches that maximize the benefits of AI, which lies in their ability to automate processes.

ACKNOWLEDGMENTS

This work has been funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 "MAKI Multi-Mechanisms Adaptation for the Future Internet".

BIBLIOGRAPHY

[1] Rashid Abbasi, Ali Kashif Bashir, Hasan J Alyamani, Farhan Amin, Jaehyeok Doh, and Jianwen Chen. "LiDAR Point Cloud Compression, Processing And Learning For Autonomous Driving." In: *IEEE Transactions on Intelligent Transportation Systems* 24.1 (2022), pp. 962–979.

[2] Mohan S Acharya, Asfia Armaan, and Aneeta S Antony. "A Comparison Of Regression Models For Prediction Of Graduate Admissions." In: *2019 International Conference On Computational Intelligence In Data Science (ICCIDS)*. IEEE. 2019, pp. 1–5.

[3] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. "Learning Representations And Generative Models For 3D Point Clouds." In: *International Conference On Machine Learning*. 2018, pp. 40–49.

[4] Arslan Ahmad, Atif Bin Mansoor, Alcardo Alex Barakabitze, Andrew Hines, Luigi Atzori, and Ray Walshe. "Supervised-learning-based QoE Prediction Of Video Streaming In Future Networks: A Tutorial With Comparative Study." In: *IEEE Communications Magazine* 59.11 (2021), pp. 88–94.

[5] Anique Akhtar, Wen Gao, Li Li, Zhu Li, Wei Jia, and Shan Liu. "Video-based Point Cloud Compression Artifact Removal." In: *IEEE Transactions on Multimedia* 24 (2021), pp. 2866–2876.

[6] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. "Oboe: Auto-tuning Video Abr Algorithms To Network Conditions." In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 2018, pp. 44–58.

[7] Evangelos Alexiou and Touradj Ebrahimi. "Towards A Point Cloud Structural Similarity Metric." In: *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2020, pp. 1–6.

[8] Evangelos Alexiou, Kuan Tung, and Touradj Ebrahimi. "Towards Neural Network Approaches For Point Cloud Compression." In: *Applications Of Digital Image Processing XLIII*. Vol. 11510. SPIE. 2020, pp. 18–37.

[9] Evangelos Alexiou, Irene Viola, Tomás M Borges, Tiago A Fonseca, Ricardo L De Queiroz, and Touradj Ebrahimi. "A Comprehensive Study Of The Rate-distortion Performance In MPEG Point Cloud Compression." In: *APSIPA Transactions on Signal and Information Processing* 8 (2019), e27.

[10] Yassin Alkhalili, Thomas Gruczyk, Tobias Meuser, Antonio Fernández Anta, Ahmad Khalil, and Andreas Mauthe. "Content-Aware Adaptive Point Cloud Delivery." In: *2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM)*. IEEE. 2022, pp. 13–20.

[11] Yassin Alkhalili, Manisha Luthra, Amr Rizk, and Boris Koldehofe. "3-D Urban Objects Detection and Classification From Point Clouds." In: *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems (DEBS'19)*. 2019, pp. 209–213.

[12] Yassin Alkhalili, Tobias Meuser, and Ralf Steinmetz. "A Survey Of Volumetric Content Streaming Approaches." In: *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*. IEEE. 2020, pp. 191–199.

[13] Yassin Alkhalili, Jannis Weil, Anam Tahir, Tobias Meuser, Boris Koldehofe, Andreas Mauthe, Heinz Koeppl, and Ralf Steinmetz. "Towards QoE-Driven Optimization of Multi-Dimensional Content Streaming." In: *Proceedings of the Conference on Networked Systems (NetSys 2021)*. European Association of Software Science and Technology. 2021.

[14] Bastian Alt, Markus Weckesser, Christian Becker, Matthias Hollick, Sounak Kar, Anja Klein, Robin Klose, Roland Kluge, Heinz Koeppl, Boris Koldehofe, et al. "Transitions: A Protocol-independent View Of The Future Internet." In: *Proceedings of the IEEE* 107.4 (2019), pp. 835–846.

[15] Shahrukh Athar and Zhou Wang. "A Comprehensive Performance Evaluation Of Image Quality Assessment Algorithms." In: *IEEE Access* 7 (2019), pp. 140030–140070.

[16] Marco Attene, Marcel Campen, and Leif Kobbelt. "Polygon Mesh Repairing: An Application Perspective." In: *ACM Computing Surveys (CSUR)* 45.2 (2013), pp. 1–33.

[17] Nabajeet Barman and Maria G Martini. "QoE Modeling For Http Adaptive Video Streaming–a Survey And Open Challenges." In: *IEEE Access* 7 (2019), pp. 30831–30859.

[18] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. "Semantickitti: A Dataset For Semantic Scene Understanding Of LiDAR Sequences." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9297–9307.

[19] Yizhak Ben-Shabat, Jonathan Paul, Eviatar Segev, Oren Shrout, and Stephen Gould. "IKEA Ego 3D Dataset: Understanding Furniture Assembly Actions From Ego-View 3D Point Clouds." In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 4355–4364.

[20] Abdelkader Benmir, Ahmed Korichi, Abdelhabib Bourouis, and Mohammed Alreshoodi. "Survey On QoE/QoS Correlation Models For Video Streaming Over Vehicular Ad-hoc Networks." In: *Journal of Computing and Information Technology* 26.4 (2018), pp. 267–287.

[21] Abdelhak Bentaleb, Bayan Taani, Ali C Begen, Christian Timmerer, and Roger Zimmermann. "A Survey On Bitrate Adaptation Schemes For Streaming Media Over Http." In: *IEEE Communications Surveys & Tutorials* 21.1 (2018), pp. 562–585.

[22] Daniel Bischoff, Florian A Schiegg, Dieter Schuller, Jens Lemke, Benjamin Becker, and Tobias Meuser. "Prioritizing Relevant Information: Decentralized V2X Resource Allocation For Cooperative Driving." In: *IEEE Access* 9 (2021), pp. 135630–135656.

[23] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. "Salient Object Detection: A Survey." In: *Computational Visual Media* 5 (2019), pp. 117–150.

[24] Kjell Brunnström, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastian Egger, Marie-Neige Garcia, Tobias HoSSfeld, Satu Jumisko-Pyykkö, Christian Keimel, Mohamed-Chaker Larabi, et al. "Qualinet White Paper On Definitions Of Quality Of Experience." In: *The Fifth Qualinet Meeting - HAL Id: hal-00977812* (2013).

[25] Mai Bui, Lin-Ching Chang, Hang Liu, Qi Zhao, and Genshe Chen. "Comparative Study Of 3D Point Cloud Compression Methods." In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, pp. 5859–5861.

[26] Chao Cao, Marius Preda, and Titus Zaharia. "3D Point Cloud Compression: A Survey." In: *The 24th International Conference on 3D Web Technology*. 2019, pp. 1–9.

[27] Chao Cao, Marius Preda, Vladyslav Zakharchenko, Euee S Jang, and Titus Zaharia. "Compression Of Sparse And Dense Dynamic Point Cloudsmethods And Standards." In: *Proceedings of the IEEE* 109.9 (2021), pp. 1537–1558.

[28] Leonor Adriana Cárdenas-Robledo, Óscar Hernández-Uribe, Carolina Reta, and Jose Antonio Cantoral-Ceballos. "Extended Reality Applications In Industry 4.0.–A Systematic Literature Review." In: *Telematics and Informatics* 73 (2022), p. 101863.

[29] Jacob Chakareski. "VR/AR Immersive Communication: Caching, Edge Computing, And Transmission Trade-offs." In: *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. 2017, pp. 36–41.

[30] Jingdao Chen, Zsolt Kira, and Yong K Cho. "Deep Learning Approach To Point Cloud Scene Understanding For Automated Scan To 3D Reconstruction." In: *Journal of Computing in Civil Engineering* 33.4 (2019), p. 04019027.

[31] 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*. 2019, pp. 88–100.

[32] Alexander Clemm, Maria Torres Vega, Hemanth Kumar Ravuri, Tim Wauters, and Filip De Turck. "Toward Truly Immersive Holographic-type Communication: Challenges And Solutions." In: *IEEE Communications Magazine* 58.1 (2020), pp. 93–99.

[33] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. "Salsanext: Fast, Uncertainty-aware Semantic Segmentation Of LiDAR Point Clouds." In: *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*. Springer. 2020, pp. 207–222.

[34] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil Bharath. "Generative Adversarial Networks: An Overview." In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 53–65.

[35] Luca De Cicco, Vito Caldaralo, Vittorio Palmisano, and Saverio Mascolo. "Elastic: A Client-side Controller For Dynamic Adaptive Streaming Over Http (DASH)." In: *2013 20th International Packet Video Workshop*. IEEE. 2013, pp. 1–8.

[36] Luca De Cicco, Giuseppe Cilli, and Saverio Mascolo. "Erudite: A Deep Neural Network For Optimal Tuning Of Adaptive Video Streaming Controllers." In: *Proceedings of the 10th ACM Multimedia Systems Conference*. 2019, pp. 13–24.

[37] Discovery. "Elephants on the Brink (360 Video)." In: *YouTube video* (2015). Available at: `https://www.youtube.com/watch?v=2bpICIClAIg`.

[38] YH Dovoedo and Subha Chakraborti. "Boxplot-based Outlier Detection For The Location-scale Family." In: *Communications in Statistics-Simulation and Computation* 44.6 (2015), pp. 1492–1513.

[39] Emil Dumic, Carlos Rafael Duarte, and Luis A da Silva Cruz. "Subjective Evaluation And Objective Measures For Point Cloudsstate Of The Art." In: *2018 First International Colloquium on Smart Grid Metrology (SmaGriMet)*. IEEE. 2018, pp. 1–5.

[40] Kerem Durak, Mehmet N Akcay, Yigit K Erinc, Boran Pekel, and Ali C Begen. "Evaluating The Performance Of Apples Low-latency Hls." In: *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2020, pp. 1–6.

[41] Chuan Feng, Pengchao Han, Xu Zhang, Bowen Yang, Yejun Liu, and Lei Guo. "Computation Offloading In Mobile Edge Computing Networks: A Survey." In: *Journal of Network and Computer Applications* (2022), p. 103366.

[42] Alexander Frömmgen. "Programming Models and Extensive Evaluation Support for MPTCP Scheduling, Adaptation Decisions, and DASH Video Streaming." PhD thesis. Technische Universität Darmstadt, 2018.

[43] Kyle Gao, Yina Gao, Hongjie He, Denning Lu, Linlin Xu, and Jonathan Li. "Nerf: Neural Radiance Field In 3D Vision, A Comprehensive Review." In: *arXiv preprint arXiv:2210.00379* (2022).

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

[45] Google, Inc. *Draco: Compression library*. `https://github.com/google/draco`. 2017.

[46] Danillo Graziosi, Ohji Nakagami, Shinroku Kuma, Alexandre Zaghetto, Teruhiko Suzuki, and Ali Tabatabai. "An Overview Of Ongoing Point Cloud Compression Standardization Activities: Video-based (V-PCC) And Geometry-based (G-PCC)." In: *APSIPA Transactions on Signal and Information Processing* 9 (2020), e13.

[47] André FR Guarda, Nuno MM Rodrigues, and Fernando Pereira. "Point Cloud Coding: Adopting A Deep Learning-based Approach." In: *2019 Picture Coding Symposium (PCS)*. IEEE. 2019, pp. 1–5.

[48] Céline Guede, Pierre Andrivon, Jean-Eudes Marvie, Julien Ricard, Bill Redmann, and Jean-Claude Chevet. "V-PCC Performance Evaluation Of The First MPEG Point Codec." In: *SMPTE Motion Imaging Journal* 130.4 (2021), pp. 36–52.

[49] Serhan Gül, Sebastian Bosse, Dimitri Podborski, Thomas Schierl, and Cornelius Hellge. "Kalman Filter-based Head Motion Prediction For Cloud-based Mixed Reality." In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 3632–3641.

[50] Serhan Gül, Dimitri Podborski, Thomas Buchholz, Thomas Schierl, and Cornelius Hellge. "Low-latency Cloud-based Volumetric Video Streaming Using Head Motion Prediction." In: *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. 2020, pp. 27–33.

[51] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. "Deep Learning For 3D Point Clouds: A Survey." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.12 (2020), pp. 4338–4364.

[52] Bo Han, Yu Liu, and Feng Qian. "Vivo: Visibility-aware Mobile Volumetric Video Streaming." In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–13.

[53] Teresa Hirzle, Florian Müller, Fiona Draxler, Martin Schmitz, Pascal Knierim, and Kasper Hornbæk. "When XR And Ai Meet-a Scoping Review On Extended Reality And Artificial Intelligence." In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–45.

[54] Jeroen van der Hooft, Maria Torres Vega, Christian Timmerer, Ali C Begen, Filip De Turck, and Raimund Schatz. "Objective And Subjective QoE Evaluation For Adaptive Point Cloud Streaming." In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2020, pp. 1–6.

[55] Jeroen van der Hooft, Maria Torres Vega, Tim Wauters, Christian Timmerer, Ali C Begen, Filip De Turck, and Raimund Schatz. "From Capturing To Rendering: Volumetric Media Delivery With Six Degrees Of Freedom." In: *IEEE Communications Magazine* 58.10 (2020), pp. 49–55.

[56] Jens de Hoog, Ahmed N Ahmed, Ali Anwar, Steven Latré, and Peter Hellinckx. "Quality-aware Compression Of Point Clouds With Google Draco." In: *16th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2021)*. Springer. 2022, pp. 227–236.

[57] Mohammad Hosseini. "Adaptive Rate Allocation For View-aware Point-cloud Streaming." In: *arXiv preprint arXiv:1911.00812* (2019).

[58] Mohammad Hosseini and Christian Timmerer. "Dynamic Adaptive Point Cloud Streaming." In: *Proceedings of the 23rd Packet Video Workshop*. 2018, pp. 25–30.

[59] Tobias HoSSfeld, Lea Skorin-Kapov, Poul E Heegaard, and Martin Varela. "Definition Of QoE Fairness In Shared Systems." In: *IEEE Communications Letters* 21.1 (2016), pp. 184–187.

[60] Junhui Hou, Lap-Pui Chau, Nadia Magnenat-Thalmann, and Ying He. "Compressing 3-d Human Motions Via Keyframe-based Geometry Videos." In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.1 (2014), pp. 51–62.

[61] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. "Point-to-voxel Knowledge Distillation For LiDAR Semantic Segmentation." In: *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. 2022, pp. 8479–8488.

[62] Hamidreza Houshiar and Andreas Nüchter. "3D Point Cloud Compression Using Conventional Image Compression For Efficient Data Transmission." In: *2015 XXV International Conference on Information, Communication and Automation Technologies (ICAT)*. IEEE. 2015, pp. 1–8.

[63] Chenn-Jung Huang, Hao-Wen Cheng, Yi-Hung Lien, and Mei-En Jian. "A Survey on Video Streaming for Next-Generation Vehicular Networks." In: *Electronics* 13.3 (2024), p. 649.

[64] Tianxin Huang and Yong Liu. "3D Point Cloud Geometry Compression On Deep Learning." In: *Proceedings of the 27th ACM International Conference on Multimedia*. 2019, pp. 890–898.

[65] Yakun Huang, Xiuquan Qiao, Haowen Wang, Xiang Su, Schahram Dustdar, and Ping Zhang. "Multi-player Immersive Communications And Interactions In Metaverse: Challenges, Architecture, And Future Directions." In: *arXiv preprint arXiv:2210.06802* (2022).

[66] Yan Huang, Jingliang Peng, C-C Jay Kuo, and M Gopi. "Octree-based Progressive Geometry Coding Of Point Clouds." In: *PBG@ SIGGRAPH*. 2006, pp. 103–110.

[67] Yan Huang, Jingliang Peng, C-C Jay Kuo, and M Gopi. "A Generic Scheme For Progressive Point Cloud Coding." In: *IEEE Transactions on Visualization and Computer Graphics* 14.2 (2008), pp. 440–453.

[68] Yan Huang, Bin Wang, C-C Jay Kuo, Hui Yuan, and Jingliang Peng. "Hierarchical Bit-wise Differential Coding (HBDC) Of Point Cloud Attributes." In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 4215–4219.

[69] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. "A Buffer-based Approach To Rate Adaptation: Evidence From A Large Video Streaming Service." In: *Proceedings of the 2014 ACM conference on SIGCOMM*. 2014, pp. 187–198.

[70] Sarah Hudson, Sheila Matson-Barkat, Nico Pallamin, and Guillaume Jegou. "With Or Without You? Interaction And Immersion In A Virtual Reality Experience." In: *Journal of Business Research* 100 (2019), pp. 459–468.

[71] John F Hughes, Andries Van Dam, Morgan McGuire, James D Foley, David Sklar, Steven Feiner, and Kurt Akeley. *Computer Graphics: Principles And Practice*. Pearson Education, 2014.

[72] Euee S Jang, Marius Preda, Khaled Mammou, Alexis M Tourapis, Jungsun Kim, Danillo B Graziosi, Sungryeul Rhyu, and Madhukar Budagavi. "Video-based Point-cloud-compression Standard In MPEG: From Evidence Collection To Committee Draft [standards In A Nutshell]." In: *IEEE Signal Processing Magazine* 36.3 (2019), pp. 118–123.

[73] Jong-Beom Jeong, Soonbin Lee, Il-Woong Ryu, Tuan Thanh Le, and Eun-Seok Ryu. "Towards Viewport-dependent 6DoF 360 Video Tiled Streaming For Virtual Reality Systems." In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 3687–3695.

[74] Junchen Jiang, Vyas Sekar, and Hui Zhang. "Improving Fairness, Efficiency, And Stability In Http-based Adaptive Video Streaming With Festive." In: *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. 2012, pp. 97–108.

[75] Michael I Jordan and Tom M Mitchell. "Machine Learning: Trends, Perspectives, And Prospects." In: *Science* 349.6245 (2015), pp. 255–260.

[76] Sounak Kar. "Performance Evaluation of Transition-based Systems with Applications to Communication Networks." PhD thesis. Technische Universität Darmstadt, 2022.

[77] Ahmad Khalil, Tobias Meuser, Yassin Alkhalili, Antonio Fernández Anta, Lukas Staecker, Ralf Steinmetz, et al. "Situational Collective Perception: Adaptive And Efficient Collective Perception In Future Vehicular Systems." In: *International Conference on Vehicle Technology and Intelligent Transport Systems*. 2022, pp. 346–352.

[78] Koffka Khan and Wayne Goodridge. "Future DASH Applications: A Survey." In: *International Journal of Advanced Networking and Applications* 10.2 (2018), pp. 3758–3764.

[79] Christian Koch. "Proactive Mechanisms for Video-on-Demand Content Delivery." PhD thesis. Technische Universität Darmstadt, 2018.

[80] Christian Koch, Arne-Tobias Rak, Michael Zink, Ralf Steinmetz, and Amr Rizk. "Transitions Of Viewport Quality Adaptation Mechanisms In 360 Degree Video Streaming." In: *Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. 2019, pp. 14–19.

[81] Georgios Kougioumtzidis, Vladimir Poulkov, Zaharias Zaharis, and Pavlos Lazaridis. "A Survey On Multimedia Services QoE Assessment And Machine Learning-based Prediction." In: *IEEE Access* 10 (2022), pp. 19507–19538.

[82] Georgios Kougioumtzidis, Vladimir Poulkov, Zaharias Zaharis, and Pavlos Lazaridis. "QoE Assessment Aspects For Virtual Reality And Holographic Telepresence Applications." In: *Future Access Enablers for Ubiquitous and Intelligent Infrastructures: 6th EAI International Conference, FABULOUS 2022, Virtual Event, May 4, 2022, Proceedings*. Springer. 2022, pp. 171–180.

[83] Maja Krivokua, Philip A. Chou, and Patrick Savill. *8i Voxelized Surface Light Field (8ivslf) Dataset*. `https://mpeg-pcc.org/index.php/pcc-content-database/8i-voxelized-surface-light-field-8ivslf-dataset/`. 2018.

[84] Maja Krivokua, Philip A Chou, and Maxim Koroteev. "A Volumetric Approach To Point Cloud CompressionPart II: Geometry Compression." In: *IEEE Transactions on Image Processing* 29 (2019), pp. 2217–2229.

[85] Jonathan Kua, Grenville Armitage, and Philip Branch. "A Survey Of Rate Adaptation Techniques For Dynamic Adaptive Streaming Over HTTP." In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1842–1866.

[86] Stanford University Computer Graphics Laboratory. *The Stanford bunny*. Accessed on June 10, 2023. 2005. URL: `http://www.graphics.stanford.edu/data/3Dscanrep/`.

[87] Zeqi Lai, Y Charlie Hu, Yong Cui, Linhui Sun, and Ningwei Dai. "Furion: Engineering High-quality Immersive Virtual Reality On Today's Mobile Devices." In: *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 2017, pp. 409–421.

[88] Kyungjin Lee, Juheon Yi, Youngki Lee, Sunghyun Choi, and Young Min Kim. "Groot: A Real-time Streaming System Of High-fidelity Volumetric Videos." In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–14.

[89] Yongjun Lee, Jaehoon Sim, Dong Ho Kim, and Dongho You. "A Comparison of Serialization Formats for Point Cloud Live Video Streaming over WebRTC." In: *2024 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2024, pp. 1–3.

[90] Jie Li, Ling Han, Chong Zhang, Qiyue Li, and Zhi Liu. "Spherical Convolution Empowered Viewport Prediction In 360 Video Multicast With Limited Fov Feedback." In: *ACM Transactions on Multimedia Computing, Communications and Applications* 19.1 (2023), pp. 1–23.

[91] Jie Li, Xiao Wang, Zhi Liu, and Qiyue Li. "A QoE Model In Point Cloud Video Streaming." In: *arXiv preprint arXiv:2111.02985* (2021).

[92] Li Li, Zhu Li, Vladyslav Zakharchenko, Jianle Chen, and Houqiang Li. "Advanced 3D Motion Prediction For Video-based Dynamic Point Cloud Compression." In: *IEEE Transactions on Image Processing* 29 (2019), pp. 289–302.

[93] Yang Li, Chenglong Dou, Yuan Wu, Weijia Jia, and Rongxing Lu. "NOMA Assisted Two-Tier VR Content Transmission: A Tile-Based Approach for QoE Optimization." In: *IEEE Transactions on Mobile Computing* 23.5 (2024), pp. 3769–3784.

[94] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. "PointCNN: Convolution On X-transformed Points." In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, pp. 828–838.

[95] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C Begen, and David Oran. "Probe And Adapt: Rate Adaptation For Http Video Streaming At Scale." In: *IEEE Journal on Selected Areas in Communications* 32.4 (2014), pp. 719–733.

[96] Jyh-Ming Lien, Gregorij Kurillo, and Ruzena Bajcsy. "Skeleton-based Data Compression For Multi-camera Tele-immersion System." In: *Advances in Visual Computing: Third International Symposium, ISVC 2007, Lake Tahoe, NV, USA, November 26-28, 2007, Proceedings, Part I 3*. Springer. 2007, pp. 714–723.

[97] Lars Linsen. *Point Cloud Representation*. Univ., Fak. für Informatik, Bibliothek Technical Report, Faculty of Computer , 2001.

[98] Jianqiang Liu, Jian Yao, Jingmin Tu, and Junhao Cheng. "Data-adaptive Packing Method For Compression Of Dynamic Point Cloud Sequences." In: *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2019, pp. 904–909.

[99] Yu Liu, Bo Han, Feng Qian, Arvind Narayanan, and Zhi-Li Zhang. "Vues: Practical Mobile Volumetric Video Streaming Through Multiview Transcoding." In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 2022, pp. 514–527.

[100] Zhi Liu, Qiyue Li, Xianfu Chen, Celimuge Wu, Susumu Ishihara, Jie Li, and Yusheng Ji. "Point Cloud Video Streaming: Challenges And Solutions." In: *IEEE Network* 35.5 (2021), pp. 202–209.

[101] Wen-Chih Lo, Chih-Yuan Huang, and Cheng-Hsin Hsu. "Edge-assisted Rendering Of 360 Videos Streamed To Head-mounted Virtual Reality." In: *2018 IEEE International Symposium on Multimedia (ISM)*. IEEE. 2018, pp. 44–51.

[102] Scott M. Lundberg and Su-In Lee. "A Unified Approach To Interpreting Model Predictions." In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2017, pp. 4768–4777.

[103] Manisha Luthra, Boris Koldehofe, Niels Danger, Pascal Weisenberger, Guido Salvaneschi, and Ioannis Stavrakakis. "Tcep: Transitions In Operator Placement To Adapt To Dynamic Network Environments." In: *Journal of Computer and System Sciences* 122 (2021), pp. 94–125.

[104] Manisha Luthra, Boris Koldehofe, and Ralf Steinmetz. "Transitions For Increased Flexibility In Fog Computing: A Case Study On Complex Event Processing." In: *Informatik Spektrum* 42 (2019), pp. 244–255.

[105] Adrien Maglo, Guillaume Lavoué, Florent Dupont, and Céline Hudelot. "3D Mesh Compression: Survey, Comparisons, And Emerging Trends." In: *ACM Computing Surveys (CSUR)* 47.3 (2015), pp. 1–41.

[106] Orlewilson Bentes Maia, Hani Camille Yehia, and Luciano de Errico. "A Concise Review Of The Quality Of Experience Assessment For Video Streaming." In: *Computer Communications* 57 (2015), pp. 1–12.

[107] Estabraq H Makiyah and Nassr N Khamees. "Emulation Of Point Cloud Streaming Over 5G Network." In: *International Journal of Information Technology* (2024), pp. 1–15.

[108] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. "Characterizations Of Noise In Kinect Depth Images: A Review." In: *IEEE Sensors journal* 14.6 (2014), pp. 1731–1740.

[109] Simone Mangiante, Guenter Klas, Amit Navon, Zhuang GuanHua, Ju Ran, and Marco Dias Silva. "VR Is On The Edge: How To Deliver 360 Videos In Mobile Networks." In: *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. 2017, pp. 30–35.

[110] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. "Neural Adaptive Video Streaming With Pensieve." In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 2017, pp. 197–210.

[111] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. "3D Object Detection For Autonomous Driving: A Review and New Outlooks." In: *arXiv preprint arXiv:2206.09474* (2022).

[112] Ajith Mascarenhas, Martin Isenburg, Valerio Pascucci, and Jack Snoeyink. "Encoding Volumetric Grids For Streaming Isosurface Extraction." In: *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.* IEEE. 2004, pp. 665–672.

[113] Kenton McHenry and Peter Bajcsy. "An Overview Of 3D Data Content, File Formats And Viewers." In: *National Center for Supercomputing Applications* 1205 (2008), p. 22.

[114] Rufael Mekuria, Kees Blom, and Pablo Cesar. "Design, Implementation, And Evaluation Of A Point Cloud Codec For Tele-immersive Video." In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.4 (2016), pp. 828–842.

[115] Facundo Mémoli and Guillermo Sapiro. "Comparing Point Clouds." In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 2004, pp. 32–40.

[116] "Methods For The Subjective Assessment Of Video Quality, Audio Quality And Audiovisual Quality Of Internet Video And Distribution Quality Television In Any Environment." In: *Recommendation ITU-T P.913 (06/21)* (2021).

[117] Tobias Meuser, Björn Richerzhagen, Ioannis Stavrakakis, The An Binh Nguyen, and Ralf Steinmetz. "Relevance-Aware Information Dissemination In Vehicular Networks." In: *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2018, pp. 588–599.

[118] Anush K Moorthy, Kalpana Seshadrinathan, and Alan C Bovik. "Image And Video Quality Assessment: Perception, Psychophysical Models, And Algorithms." In: *Perceptual Digital Imaging: Methods and Applications* (2017), pp. 55–81.

[119] Carlos Moreno and Ming Li. "A Comparative Study Of Filtering Methods For Point Clouds In Real-time Video Streaming." In: *Proceedings of the World Congress on Engineering and Computer Science*. Vol. 1. 2016, pp. 388–393.

[120] Carlos Moreno and Ming Li. "Frame Filtering And Skipping For Point Cloud Data Video Transmission." In: *Advances in Science, Technology and Engineering Systems Journal* 2.1 (2017), pp. 76–83.

[121] Mu Mu, Murtada Dohan, Alison Goodyear, Gary Hill, Cleyon Johns, and Andreas Mauthe. "User Attention And Behaviour In Virtual Reality Art Encounter." In: *Multimedia Tools and Applications* (2022), pp. 1–30.

[122] Javier Navarrete, Diego Viejo, and Miguel Cazorla. "Compression And Registration Of 3D Point Clouds Using GMMS." In: *Pattern Recognition Letters* 110 (2018), pp. 8–15.

[123] Johannes Otepka, Sajid Ghuffar, Christoph Waldhauser, Ronald Hochreiter, and Norbert Pfeifer. "Georeferenced Point Clouds: A Survey Of Features And Point Cloud Management." In: *ISPRS International Journal of Geo-Information* 2.4 (2013), pp. 1038–1065.

[124] Selcen Ozturkcan. "Service Innovation: Using Augmented Reality In The Ikea Place App." In: *Journal of Information Technology Teaching Cases* 11.1 (2021), pp. 8–13.

[125] Federica Pallavicini, Alessandro Pepe, and Maria Eleonora Minissi. "Gaming In Virtual Reality: What Changes In Terms Of Usability, Emotional Response And Sense Of Presence Compared To Non-immersive Video Games?" In: *Simulation & Gaming* 50.2 (2019), pp. 136–159.

[126] Xiang Pan, Qing Lin, Siyi Ye, Li Li, Li Guo, and Brendan Harmon. "Deep Learning Based Approaches From Semantic Point Clouds To Semantic Bim Models For Heritage Digital Twin." In: *Heritage Science* 12.1 (2024), p. 65.

[127] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. "Masked Autoencoders For Point Cloud Self-Supervised Learning." In: *European Conference On Computer Vision*. 2022, pp. 604–621.

[128] Jounsup Park, Philip A Chou, and Jenq-Neng Hwang. "Rate-utility Optimized Streaming Of Volumetric Media For Augmented Reality." In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.1 (2019), pp. 149–162.

[129] Pablo Pérez. "Exploring The Realverse: Building, Deploying, And Managing QoE In XR Communications." In: *2022 ITU Kaleidoscope-Extended reality–How to Boost Quality of Experience and Interoperability*. IEEE. 2022, pp. 1–11.

[130] Stefano Petrangeli, Gwendal Simon, and Viswanathan Swaminathan. "Trajectory-based Viewport Prediction For 360-degree Virtual Reality Videos." In: *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE. 2018, pp. 157–160.

[131] Md Jalil Piran, Quoc-Viet Pham, SM Riazul Islam, Sukhee Cho, Byungjun Bae, Doug Young Suh, and Zhu Han. "Multimedia Communication Over Cognitive Radio Networks From QoS/QoE Perspective: A Comprehensive Survey." In: *Journal of Network and Computer Applications* 172 (2020), p. 102759.

[132] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. "PointNet++: Deep Hierarchical Feature Learning On Point Sets In A Metric Space." In: *Advances in neural information processing systems* 30 (2017).

[133] Feng Qian, Bo Han, Jarrell Pair, and Vijay Gopalakrishnan. "Toward Practical Volumetric Video Streaming On Commodity Smartphones." In: *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*. 2019, pp. 135–140.

[134] Zizheng Que, Guo Lu, and Dong Xu. "Voxelcontext-net: An Octree Based Framework For Point Cloud Compression." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6042–6051.

[135] Kjetil Raaen and Ivar Kjellmo. "Measuring Latency In Virtual Reality Systems." In: *Entertainment Computing-ICEC 2015: 14th International Conference, ICEC 2015, Trondheim, Norway, September 29-Ocotober 2, 2015, Proceedings 14*. Springer. 2015, pp. 457–462.

[136] Benjamin Rainer, Stefan Lederer, Christopher Müller, and Christian Timmerer. "A Seamless Web Integration Of Adaptive Http Streaming." In: *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE. 2012, pp. 1519–1523.

[137] Björn Richerzhagen, Nils Richerzhagen, Julian Zobel, Sophie Schönherr, Boris Koldehofe, and Ralf Steinmetz. "Seamless Transitions Between Filter Schemes For Location-based Mobile Applications." In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE. 2016, pp. 348–356.

[138] Nils Richerzhagen. "Transition in Monitoring and Network Offloading - Handling Dynamic Mobile Applications and Environments." PhD thesis. Technische Universität Darmstadt, 2019.

[139] Nils Richerzhagen, Björn Richerzhagen, Rhaban Hark, Dominik Stingl, Andreas Mauthe, Alberto E Schaeffer-Filho, Klara Nahrstedt, and Ralf Steinmetz. "Adaptive Monitoring For Mobile Networks In Challenging Environments." In: *Advances in Computer Communications and Networks From Green, Mobile, Pervasive Networking to Big Data Computing*. River Publishers, 2022, pp. 91–126.

[140] Ricardo Roriz, Jorge Cabral, and Tiago Gomes. "Automotive LiDAR Technology: A Survey." In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2021), pp. 6282–6297.

[141] Michael Rudolph, Aron Riemenschneider, and Amr Rizk. "Progressive Coding for Deep Learning based Point Cloud Attribute Compression." In: *Proceedings of the 16th International Workshop on Immersive Mixed and Virtual Environment Systems*. 2024, pp. 78–84.

[142] Michael Rudolph and Amr Rizk. "View-Adaptive Streaming Of Point Cloud Scenes Through Combined Decomposition And Video-Based Coding." In: *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*. 2022, pp. 41–49.

[143] Michael Rudolph, Stefan Schneegass, and Amr Rizk. "RABBIT: Live Transcoding of V-PCC Point Cloud Streams." In: *Proceedings of the 14th Conference on ACM Multimedia Systems*. 2023, pp. 97–107.

[144] Pietro Ruiu, Lorenzo Mascia, and Enrico Grosso. "Saliency-Guided Point Cloud Compression for 3D Live Reconstruction." In: *Multimodal Technologies and Interaction* 8.5 (2024), p. 36.

[145] Yusuf Sani, Andreas Mauthe, and Christopher Edwards. "Adaptive Bitrate Selection: A Survey." In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2985–3014.

[146] Yusuf Sani, Darijo Raca, Jason J Quinlan, and Cormac J Sreenan. "Smash: A Supervised Machine Learning Approach To Adaptive Video Streaming Over Http." In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2020, pp. 1–6.

[147] Ruwen Schnabel and Reinhard Klein. "Octree-based Point-cloud Compression." In: *PBG@ SIGGRAPH* 3 (2006).

[148] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokua, Sébastien Lasserre, Zhu Li, et al. "Emerging MPEG Standards For Point Cloud Compression." In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.1 (2018), pp. 133–148.

[149] Anuj Sharma, Sukhdeep Singh, and S Ratna. "Graph Neural Network Operators: A Review." In: *Multimedia Tools and Applications* 83.8 (2024), pp. 23413–23436.

[150] Shu Shi, Varun Gupta, Michael Hwang, and Rittwik Jana. "Mobile VR On Edge Cloud: A Latency-driven Design." In: *Proceedings of The 10th ACM Multimedia Systems Conference*. 2019, pp. 222–231.

[151] Yuang Shi, Bennett Clement, and Wei Tsang Ooi. "QV4: QoE-based Viewpoint-Aware V-PCC-encoded Volumetric Video Streaming." In: *Proceedings of the 15th ACM Multimedia Systems Conference*. 2024, pp. 144–154.

[152] Ashutosh Singla, Shuang Wang, Steve Göring, Rakesh Rao Ramachandra Rao, Irene Viola, Pablo Cesar, and Alexander Raake. "Subjective Quality Evaluation of Point Clouds using Remote Testing." In: *Proceedings of the 2nd International Workshop on Interactive eXtended Reality*. 2023, pp. 21–28.

[153] Kevin Spiteri, Ramesh Sitaraman, and Daniel Sparacio. "From Theory To Practice: Improving Bitrate Adaptation In The DASH Reference Player." In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15.2s (2019), pp. 1–29.

[154] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K Sitaraman. "Bola: Near-optimal Bitrate Adaptation For Online Videos." In: *IEEE/ACM Transactions On Networking* 28.4 (2020), pp. 1698–1711.

[155] Ralf Steinmetz and Klara Nahrstedt. *Multimedia: Computing, Communications And Applications*. Pearson Education India, 2012.

[156] Thomas Stockhammer. "Dynamic Adaptive Streaming Over Http– Standards And Design Principles." In: *Proceedings of the second annual ACM conference on Multimedia systems*. 2011, pp. 133–144.

[157] Denny Stohr. "User-centric Video in the Future Internet: QoE in Participatory Video Generation and Distribution." PhD thesis. Technische Universität Darmstadt, 2018.

[158] Jakob Struye, Filip Lemic, and Jeroen Famaey. "Towards Ultra-low-latency Mmwave Wi-fi For Multi-user Interactive Virtual Reality." In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6.

[159] Honglei Su, Zhengfang Duanmu, Wentao Liu, Qi Liu, and Zhou Wang. "Perceptual Quality Assessment Of 3D Point Clouds." In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 3182–3186.

[160] Shishir Subramanyam, Irene Viola, Alan Hanjalic, and Pablo Cesar. "User Centered Adaptive Streaming Of Dynamic Point Clouds With Low Complexity Tiling." In: *Proceedings of the 28th ACM international conference on multimedia*. 2020, pp. 3669–3677.

[161] Shishir Subramanyam, Irene Viola, Jack Jansen, Evangelos Alexiou, Alan Hanjalic, and Pablo Cesar. "Subjective QoE Evaluation Of User-centered Adaptive Streaming Of Dynamic Point Clouds." In: *2022 14th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2022, pp. 1–6.

[162] Xuebin Sun, Han Ma, Yuxiang Sun, and Ming Liu. "A Novel Point Cloud Compression Algorithm Based On Clustering." In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2132–2139.

[163] Yuan-Chun Sun, Sheng-Ming Tang, Ching-Ting Wang, and Cheng-Hsin Hsu. "On Objective And Subjective Quality Of 6DoF Synthesized Live Immersive Videos." In: *Proceedings of the 2nd Workshop on Quality of Experience in Visual Multimedia Applications*. 2022, pp. 49–56.

[164] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[165] Dong Tian, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro. "Geometric Distortion Metrics For Point Cloud Compression." In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 3460–3464.

[166] Christian Timmerer and Herman Hellwagner. "Http Adaptive Streaming: Where Is It Heading?" In: *Proceedings of the Brazilian Symposium on Multimedia and the Web*. 2020, pp. 349–350.

[167] Chi-Yi Tsai and Chih-Hung Huang. "Indoor Scene Point Cloud Registration Algorithm Based On Rgb-d Camera Calibration." In: *Sensors* 17.8 (2017), p. 1874.

[168] IT Union. "Methods For The Subjective Assessment Of Video Quality Audio Quality And Audiovisual Quality Of Internet Video And Distribution Quality Television In Any Environment." In: *Series P: Terminals And Subjective And Objective Assessment Methods* (2016).

[169] Jeroen Van Der Hooft, Tim Wauters, Filip De Turck, Christian Timmerer, and Hermann Hellwagner. "Towards 6DoF Http Adaptive Streaming Through Point Cloud Compression." In: *Proceedings of the 27th ACM International Conference on Multimedia*. 2019, pp. 2405–2413.

[170] Adam Viola, Sahil Sharma, Pankaj Bishnoi, Matheus Gadelha, Stefano Petrangeli, Haoliang Wang, and Viswanathan Swaminathan. "Trace Match & Merge: Long-term Field-of-view Prediction For AR Applications." In: *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE. 2021, pp. 1–9.

[171] Irene Viola and Pablo Cesar. "Volumetric Video Streaming: Current Approaches And Implementations." In: *Immersive Video Technologies* (2023), pp. 425–443.

[172] Fei Wang, Zesong Fei, Jing Wang, Yifan Liu, and Zhikun Wu. "Has QoE Prediction Based On Dynamic Video Features With Data Mining In LTE Network." In: *Science China Information Sciences* 60.4 (2017), pp. 1–14.

[173] Lisha Wang, Chenglin Li, Wenrui Dai, Shaohui Li, Junni Zou, and Hongkai Xiong. "QoE-driven Adaptive Streaming For Point Clouds." In: *IEEE Transactions on Multimedia* 25 (2023), pp. 2543–2558.

[174] Lisha Wang, Chenglin Li, Wenrui Dai, Junni Zou, and Hongkai Xiong. "QoE-driven And Tile-based Adaptive Streaming For Point Clouds." In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 1930–1934.

[175] Na Wang, Haoliang Wang, Stefano Petrangeli, Viswanathan Swaminathan, Fei Li, and Songqing Chen. "Towards Field-of-view Prediction For Augmented Reality Applications On Mobile Devices." In: *Proceedings of the 12th ACM International Workshop on Immersive Mixed and Virtual Environment Systems*. 2020, pp. 13–18.

[176] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. "Emergent Abilities Of Large Language Models." In: *arXiv preprint arXiv:2206.07682* (2022).

[177] Jannis Weil, Yassin Alkhalili, Anam Tahir, Thomas Gruczyk, Tobias Meuser, Mu Mu, Heinz Koeppl, and Andreas Mauthe. "Modeling Quality of Experience for Compressed Point Cloud Sequences based on a Subjective Study." In: *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2023, pp. 135–140.

[178] Stefan Wilk. "Quality-aware Content Adaptation in Digital Video Streaming." PhD thesis. Technische Universität Darmstadt, 2016.

[179] Stefan Wilk, Roger Zimmermann, and Wolfgang Effelsberg. "Leveraging Transitions For The Upload Of User-generated Mobile Video." In: *Proceedings of the 8th International Workshop on Mobile Video*. 2016, pp. 1–6.

[180] Tzu-Tsung Wong. "Performance Evaluation Of Classification Algorithms By K-fold And Leave-one-out Cross Validation." In: *Pattern Recognition* 48.9 (2015), pp. 2839–2846.

[181] Huajie Wu, Yihang Li, Wei Xu, Fanze Kong, and Fu Zhang. "Moving Event Detection from LiDAR Point Streams." In: *Nature Communications* 15.1 (2024), p. 345.

[182] Xinju Wu, Yun Zhang, Chunling Fan, Junhui Hou, and Sam Kwong. "Subjective Quality Database And Objective Study Of Compressed Point Clouds With 6DoF Head-mounted Display." In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.12 (2021), pp. 4630–4644.

[183] Minrui Xu, Wei Chong Ng, Wei Yang Bryan Lim, Jiawen Kang, Zehui Xiong, Dusit Niyato, Qiang Yang, Xuemin Sherman Shen, and Chunyan Miao. "A Full Dive Into Realizing The Edge-enabled Metaverse: Visions, Enabling Technologies, And Challenges." In: *IEEE Communications Surveys & Tutorials* (2022).

[184] Yi Xu, Yao Lu, and Ziyu Wen. *Owlii Dynamic Human Textured Mesh Sequence Dataset*. `https://mpeg-pcc.org/index.php/pcc-content-database/owlii-dynamic-human-textured-mesh-sequence-dataset/`. 2017.

[185] Yiling Xu, Ke Zhang, Lanyi He, Zhiqian Jiang, and Wenjie Zhu. "Introduction To Point Cloud Compression." In: *ZTE Communications* 16.3 (2018), p. 8.

[186] Praveen Kumar Yadav, Arash Shafiei, and Wei Tsang Ooi. "Quetra: A Queuing Theory Approach To DASH Rate Adaptation." In: *Proceedings of the 25th ACM international conference on Multimedia*. 2017, pp. 1130–1138.

[187] Francis Y Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Alexander Levis, and Keith Winstein. "Learning In Situ: A Randomized Experiment In Video Streaming." In: *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. Vol. 20. USENIX Association, 2020, pp. 495–511.

[188] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. "2dpass: 2d Priors Assisted Semantic Segmentation On LiDAR Point Clouds." In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*. Springer. 2022, pp. 677–695.

[189] Abid Yaqoob, Ting Bi, and Gabriel-Miro Muntean. "A Survey On Adaptive 360 Video Streaming: Solutions, Challenges And Opportunities." In: *IEEE Communications Surveys & Tutorials* 22.4 (2020), pp. 2801–2838.

[190] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. "A Control-theoretic Approach For Dynamic Adaptive Video Streaming Over Http." In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 2015, pp. 325–338.

[191] Ziyu Ying, Shulin Zhao, Sandeepa Bhuyan, Cyan Subhra Mishra, Mahmut T Kandemir, and Chita R Das. "Pushing Point Cloud Compression To The Edge." In: *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE. 2022, pp. 282–299.

[192] Youguang Yu, Wei Zhang, Ge Li, and Fuzheng Yang. "A Regularized Projection-based Geometry Compression Scheme For LiDAR Point Cloud." In: *IEEE Transactions on Circuits and Systems for Video Technology* 33.3 (2023), pp. 1427–1437.

[193] Emin Zerman, Cagri Ozcinar, Pan Gao, and Aljosa Smolic. "Textured Mesh Vs Coloured Point Cloud: A Subjective Study For Volumetric Video Compression." In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2020, pp. 1–6.

[194] Cunhui Zhang, Yangjie Cao, Zhi Liu, Rui Yin, Yongdong Zhu, and Xianfu Chen. "Trans-rl: A Prediction-control Approach For QoE-aware Point Cloud Video Streaming." In: *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE. 2022, pp. 1899–1904.

[195] Jiaying Zhang, Xiaoli Zhao, Zheng Chen, and Zhejun Lu. "A Review Of Deep Learning-based Semantic Segmentation For Point Cloud." In: *IEEE Access* 7 (2019), pp. 179118–179133.

[196] Ke Zhang, Wenjie Zhu, and Yiling Xu. "Hierarchical Segmentation Based Point Cloud Attribute Compression." In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 3131–3135.

[197] Xumiao Zhang, Anlan Zhang, Jiachen Sun, Xiao Zhu, Y Ethan Guo, Feng Qian, and Z Morley Mao. "Emp: Edge-assisted Multi-vehicle Perception." In: *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 2021, pp. 545–558.

[198] Yuan Zhang, Lingjun Pu, Tao Lin, and Jinyao Yan. "QoE-oriented Mobile Virtual Reality Game In Distributed Edge Networks." In: *IEEE Transactions on Multimedia* 25 (2023), pp. 9132–9146.

[199] Jingbo Zhao, Robert S Allison, Margarita Vinnikov, and Sion Jennings. "Estimating The Motion-to-photon Latency In Head Mounted Displays." In: *2017 IEEE Virtual Reality (VR)*. IEEE. 2017, pp. 313–314.

[200] Lili Zhao, Kai-Kuang Ma, Zhili Liu, Qian Yin, and Jianwen Chen. "Real-time Scene-aware LiDAR Point Cloud Compression Using Semantic Prior Representation." In: *IEEE Transactions on Circuits and Systems for Video Technology* 32.8 (2022), pp. 5623–5637.

[201] Guoquan Zheng and Liang Yuan. "A Review Of QoE Research Progress In Metaverse." In: *Displays* 77 (2023), p. 102389. ISSN: 0141-9382.

[202] Pengyuan Zhou, Lin Wang, Zhi Liu, Yanbin Hao, Pan Hui, Sasu Tarkoma, and Jussi Kangasharju. "A Survey On Generative AI And LLM For Video Generation, Understanding, And Streaming." In: *arXiv preprint arXiv:2404.16038* (2024).

[203] Wenjie Zhu, Zhan Ma, Yiling Xu, Li Li, and Zhu Li. "View-dependent Dynamic Point Cloud Compression." In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.2 (2020), pp. 765–781.

[204] Yuanwei Zhu, Yakun Huang, Xiuquan Qiao, Zhijie Tan, Boyuan Bai, Huadong Ma, and Schahram Dustdar. "A Semantic-aware Transmission With Adaptive Control Scheme For Volumetric Video Service." In: *IEEE Transactions on Multimedia* 25 (2023), pp. 7160–7172.

All web pages cited in this work have been checked in June 2024. However, due to the dynamic nature of the World Wide Web, their long-term availability cannot be guaranteed.

7

A

LIST OF ACRONYMS

A.1 LIST OF ACRONYMS

0DoF zero Degrees of Freedom

3DoF three Degrees of Freedom

6DoF six Degrees of Freedom

ABR Adaptive Bitrate Streaming

ACR Absolute Category Rating

AR Augmented Reality

CNNs Convolutional Neural Networks

DASH Dynamic Adaptive Streaming over HTTP

DoF Degrees of Freedom

FOV Field of View

G-PCC Geometry codec based Point Cloud Compression

GANs Generative Adversarial Networks

GMMs Gaussian Mixture Models

KNN K-Nearest Neighbors

LiDAR Light Detection and Ranging

LoD Level Of Detail

LSTM Long Short-Term Memory

MAKI Multi-Mechanisms Adaptation for the Futrue Internet

MLP Multilayer Perceptron

MOS Mean Opinion Score

MPD Media Presentation Description

MR Mixed Reality

NeRFs Neural Radiance Fields

PCC Point Cloud Compression

PLY Polygon File Format

PSNR Peak Signal-to-Noise Ratio

QoE Quality of Experience

QoS Quality of Service

RNNs Recurrent Neural Networks

ROI Region Of Interest

SGD Stochastic Gradient Descend

SHAP SHapley Additive exPlanations

V-PCC Video codec based Point Cloud Compression

VR Virtual Reality

XR Extended Reality

B

AUTHOR'S PUBLICATIONS

MAIN PUBLICATIONS

[1] Jannis Weil, Yassin Alkhalili, Anam Tahir, Thomas Gruczyk, Tobias Meuser, Mu Mu, Heinz Koeppl, and Andreas Mauthe. "Modeling Quality of Experience for Compressed Point Cloud Sequences based on a Subjective Study." In: *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2023, pp. 135–140.

[2] Yassin Alkhalili, Thomas Gruczyk, Tobias Meuser, Antonio Fernández Anta, Ahmad Khalil, and Andreas Mauthe. "Content-Aware Adaptive Point Cloud Delivery." In: *2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM)*. IEEE. 2022, pp. 13–20.

[3] Yassin Alkhalili, Jannis Weil, Anam Tahir, Tobias Meuser, Boris Koldehofe, Andreas Mauthe, Heinz Koeppl, and Ralf Steinmetz. "Towards QoE-Driven Optimization of Multi-Dimensional Content Streaming." In: *Proceedings of the Conference on Networked Systems (NetSys 2021)*. European Association of Software Science and Technology. 2021.

[4] Yassin Alkhalili, Tobias Meuser, and Ralf Steinmetz. "A Survey Of Volumetric Content Streaming Approaches." In: *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*. IEEE. 2020, pp. 191–199.

CO-AUTHORED PUBLICATIONS

[5] Ahmad Khalil, Tobias Meuser, Yassin Alkhalili, Antonio Fernández Anta, Lukas Staecker, Ralf Steinmetz, et al. "Situational Collective Perception: Adaptive And Efficient Collective Perception In Future Vehicular Systems." In: *International Conference on Vehicle Technology and Intelligent Transport Systems*. 2022, pp. 346–352.

[6] Wael Alkhatib, Steffen Schnitzer, Wei Ding, Peter Jiang, Yassin Alkhalili, and Christoph Rensing. "Comparison of Feature Selection Techniques for Multi-label Text Classification against a New Semantic-based Method." In: *Computational Linguistics and Intelligent Text Processing: 19th International Conference, CICLing 2018*. 2018. URL: `http://www.cicling.org/2018/intranet/preprint/papers/paper_136.pdf`.

[7] Yassin Alkhalili, Manisha Luthra, Amr Rizk, and Boris Koldehofe. “3-D Urban Objects Detection and Classification From Point Clouds.” In: *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems (DEBS'19)*. 2019, pp. 209–213.

§8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass von mir zu keinem vorherigen Zeitpunkt bereits ein Promotionsversuch unternommen wurde. Andernfalls versichere ich, dass der promotionsführende Fachbereich über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs informiert ist.

§9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation, abgesehen von den in ihr ausdrücklich genannten Hilfsmitteln, selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde. Weiterhin versichere ich, dass die "Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Technischen Universität Darmstadt" sowie die "Leitlinien zum Umgang mit digitalen Forschungsdaten an der TU Darmstadt" in den jeweils aktuellen Versionen bei der Verfassung der Dissertation beachtet wurden.

§9 Abs. 2 PromO

Ich versichere hiermit, dass die vorliegende Dissertation bisher noch nicht zu Prüfungszwecken gedient hat.

Darmstadt, 25. Juni 2024

MHD Yassin Al Khalili

colophon

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`https://bitbucket.org/amiede/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

Final Version as of December 2, 2024 (`classicthesis`).