A Machine-Learning-Driven Fast Video-based Point Cloud Compression (V-PCC)

Gustavo Rehbein
ghrehbein@inf.ufpel.edu.br
Universidade Federal de Pelotas
Video Technology Research Group – Vitech
Graduate Program in Computing
Pelotas, Brazil

Eduardo Costa
edfcosta@inf.ufpel.edu.br
Universidade Federal de Pelotas
Video Technology Research Group – Vitech
Pelotas, Brazil

Guilherme Corrêa
gcorrea@inf.ufpel.edu.br
Universidade Federal de Pelotas
Video Technology Research Group – Vitech
Graduate Program in Computing
Pelotas, Brazil

Cristiano Santos
cfdsantos@inf.ufpel.edu.br
Universidade Federal de Pelotas
Video Technology Research Group – Vitech
Graduate Program in Computing
Pelotas, Brazil

Marcelo Porto
porto@inf.ufpel.edu.br
Universidade Federal de Pelotas
Video Technology Research Group – Vitech
Graduate Program in Computing
Pelotas, Brazil

ABSTRACT

In recent years, 3D point cloud content has gained attention due to its application possibilities, such as multimedia systems, virtual, augmented, and mixed reality, through the mapping and visualization of environments and/or 3D objects, real-time immersive communications, and autonomous driving systems. However, raw point clouds demand a large amount of data for their representation, and compression is mandatory to allow efficient transmission and storage. The MPEG group proposed the Video-based Point Cloud Compression (V-PCC) standard, which is a dynamic point cloud encoder based on the use of video encoders through projections into 2D space. However, V-PCC demands a high computational cost, demanding fast implementations for real-time processing and, especially, for mobile device applications. In this paper, a machine-learning-based fast implementation of V-PCC is proposed, where the main approach is the use of trained decision trees to speed up the block partitioning process during the point cloud compression. The results show that the proposed fast V-PCC solution is able to achieve an encoding time reduction of 42.73% for the geometry video sub-stream and 55.3% for the attribute video sub-stream, with a minimal impact on bitrate and objective quality.

KEYWORDS

point clouds, machine learning, V-PCC, complexity reduction

1 INTRODUCTION

Point clouds can be used in many applications, such as 3D environment mapping, representation of historical objects or monuments, and virtual, augmented, and mixed realities, providing detailed 3D representation and multiple points of view of objects and scenes. Regarding the application context, point clouds can be categorized into three types: (Category 1) static, which represents the capture of a scene or object at a single time instant; (Category 2) dynamic, when a sequence of point clouds is captured over time; and (Category 3) dynamically captured through multiple partially overlapping scans, typically during dynamic mapping of a large-scale environment.

Nevertheless, as non-compressed point clouds comprise a large amount of data, storage and transmission become challenging tasks depending on the point cloud resolution and the device on which it is processed. Thus, compression becomes mandatory to allow for the efficient transmission and storage of this type of 3D content. The Motion Picture Experts Group (MPEG) defined two Point Cloud Compression (PCC) standard specifications: the Video-based PCC (V-PCC) [16], related to dynamic point clouds, and the Geometry-based PCC (G-PCC), designed to compress static and dynamically acquired point clouds [8]. While the V-PCC coding approach is based on 2D projections of the point clouds to be further compressed with a regular 2D video encoder, G-PCC encodes the content directly in the 3D space [8].

V-PCC is an interesting alternative to point clouds compression, since the already existing video encoders can be used. Although V-PCC provides an efficient solution for compressing the data generated by point clouds, this encoder inherits the complexity of video encoding. This occurs because V-PCC, after the point cloud 2D projection step, uses an HEVC [19] video encoder. In this process, the HEVC video encoder needs to handle three video sub-streams generated in the 2D projection step by V-PCC [16]. One video sub-stream is generated to indicate the occupancy map, which shows where points actually exist due to the sparse and irregular nature of a point cloud. Additionally, a second sub-stream is necessary to indicate the geometric position of each point, i.e., the position on the Z-axis or depth. A third sub-stream, called Attribute sub-stream, indicates the color or appearance assigned to each point [16].

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2024). Juiz de Fora, Brazil. Porto Alegre: Brazilian Computer Society, 2024.

ISSN 2966-2753

There are many works in literature proposing the use of machine learning to reduce video compression complexity, as [21], [15], [24]. One of the most efficient approach in to predict the blocks partitioning during the encoding process [3], [18], [23], [4]. This type of prediction prevents the encoder to evaluate smaller block sizes, thereby reducing complexity and achieving time savings.

This work proposes a machine-learning-based fast V-PCC implementation that employs an already trained model originally proposed for 2D video compression [3], to reduce the complexity of point cloud compression in the V-PCC standard. To this end, the use and adaptation of a pre-trained model to accelerate video compression is proposed as an alternative to speeding up the point cloud compression process in the Test Model Category 2 (TMC2) [14], the reference software of the V-PCC standard.

The machine learning model used achieved 84% accuracy on the testing dataset [3]. This enabled a 42.73% time reduction in the video encoding step for the geometry sub-stream and a 55.3% reduction for the attributes sub-stream. The loss in coding efficiency, measured with the BD-Rate metric [1], is of 4.2% for geometry (using D2 metric) and 3.2% for color attributes, respectively. The results show that while there is an impact on encoding efficiency, this can be outweighed by the significant gains in complexity reduction.

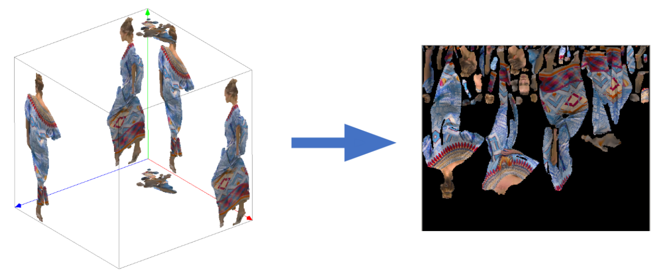
This paper is organized as follows: Section 2 presents related work, Section 3 introduces the V-PCC standard and its operation, Section 4 outlines the proposed method, Section 5 presents the results of the experiments conducted, and Section 6 provides the conclusions.

2 VIDEO-BASED POINT CLOUD COMPRESSION (V-PCC)

The Video-based Point Cloud Compression (V-PCC) is a standard for point cloud compression proposed by MPEG to encode dynamic point clouds [17]. In other words, this encoder aims to compress point cloud content analogous to 2D videos, meaning that it has temporally adjacent frames that provide a sense of motion to the content.

As the name suggests, V-PCC is a video-based encoder, which means it uses a video encoder in its encoding stages, by default the High-Efficiency Video Coding (HEVC) [19][5]. However, to enable encoding via HEVC, a 2D projection process of the point clouds is performed, which is illustrated in Figure 1. As can be seen in Figure 1, the 2D projection process is analogous to having virtual cameras registering parts of the point cloud (Figure 1 (a)), and combining those camera images into a mosaic, i.e. an image that contains the collection of projected 2D patches (Figure 1 (b)) [8].

The encoder flow proposed by the V-PCC standard is illustrated in Figure 2. It is possible to see that the 2D projection stages initially occur through the patch generation, which is responsible for slicing the cloud into several parts. These patches then undergo the padding and packing process to assemble the 2D image. In this process, three video sub-streams are generated: one for geometry information, which maps the changes of the depth of each patch, information which otherwise would be lost in the 2D projection process (generated in Geometry image generation); a second for the occupancy map information, used to inform where points exist in the 2D projection of the point cloud, which is necessary due to the sparse nature of point clouds; and a third for the color information of each point in the point cloud (Attribute image generation). The latter also goes through a process of smoothing the edges present in the patches (Smoothing). After the 2D projection process is completed, these three video sub-streams (Figure 3) are sent to the HEVC video encoder, which needs to be independently activated to handle each of these sub-streams.



(a) (b)

Figure 1: 3D to 2D projection of point cloud in V-PCC. (a) Point cloud projection into planes, (b) collection of patches

The V-PCC standard offers two main configurations: All Intra (AI) and Random Access (RA) [16]. In the AI configuration, each frame is compressed independently, exploring redundancies only within the current frame. This simplifies processing but may result in lower compression efficiency. In contrast, the RA configuration explores redundancies between frames and can reuse data from neighboring frames, allowing for more efficient compression with higher complexity.

2.1 HEVC

High-Efficiency Video Coding (HEVC) [20] is a video coding standard developed by the MPEG group and finalized in 2013. HEVC brought about a 50% increase in encoding efficiency compared to its predecessor, H.264/MPEG-AVC [9]. A portion of these gains is due to the use of a new Coding Tree Units (CTU) partitioning scheme based on recursive quad-trees [10], which can range in size from 8x8 to 64x64 samples. A CTU can be recursively partitioned into multiple Coding Units (CUs) until they reach a size of 8x8 (or 4x4 for chrominance samples). Figure 4 shows an example of the partitioning of a 64x64 CTU along with the Coding Tree structure. In Figure 4, 1's indicate that a subdivision (or split) of CUs occurred, and 0's indicate that there was no subdivision at that level.

2.2 V-PCC complexity analysis

An analysis was conducted to identify the complex behavior of V-PCC stages. For this, the Test Model Category 2 (TMC2) reference software for V-PCC jointly with the GNU profiling (G-Prof) software version 2.9.1 was used [6]. In these experiments, the Random Access temporal configuration was used to encode all 10-bit test sequences of the V-PCC Common Test Conditions (CTCs) [13], five point clouds sequences in total. All five bitrate configurations available on TMC2 (r1, r2, r3, r4, and r5) were used, r1 being the configuration with the lowest bitrate, and r5 being the one with the highest bitrate and quality, respectively.

Figure 5 presents the results (in percentage) of the average time for each coding step, considering the five point cloud sequences.

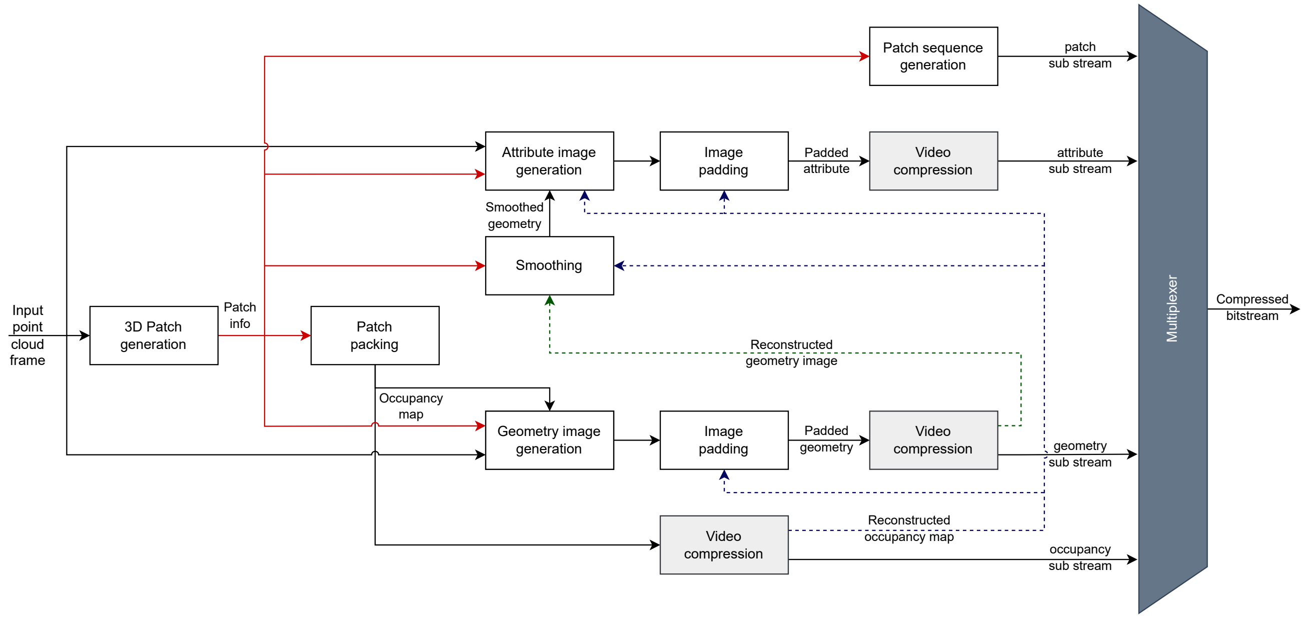
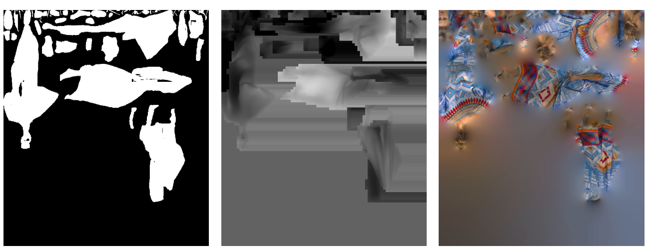


Figure 2: V-PCC encoder diagram [16].



(a) **(b)** **(c)**

Figure 3: Example of an Occupation image (a), Geometry image (b) and Attribute image (c) extracted from V-PCC.

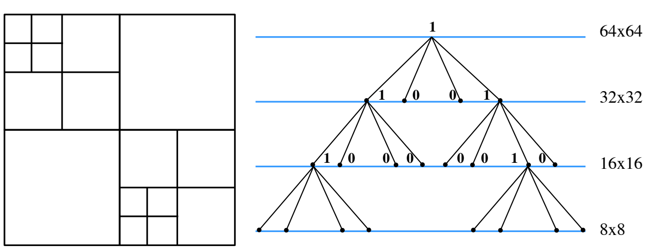


Figure 4: Example of a 64x64 CTU and Coding Tree structure being split into smaller CUs (adapted from [10]).

These results were separated into bitrate settings. In this analysis, it was identified that the video encoding is the most complex stage among the processes of the V-PCC standard. The HEVC demands 90% of the average encoding time of the V-PCC, as it can be seen in Figure 5. This complexity is observed through the sum of the "Video Encoder Steps" and "Video Encoder ME" percentage values.

Table 1 presents the average results in percentage of time used by HEVC to encode each type of video sub-stream (Occupation, Geometry, and Attributes). It is possible to see that the encoding of Geometry and Attribute videos accounts for approximately 99.18% of the total time spent by HEVC, while the Occupation video encoding accounts for only 0.82%.

Table 1: Percentage of time on the HEVC encoding of each sub-stream in V-PCC.

Rate setting	Occupation	Geometry	Attribute
r1	0.59%	54.89%	44.52%
r2	0.57%	55.10%	44.33%
r3	0.53%	54.05%	45.42%
r4	0.49%	53.36%	46.15%
r5	1.95%	50.56%	47.49%
Average	0.82%	53.59%	45.58%

In this regard, improvements to reduce the video encoding complexity are primarily required to make real-time point cloud encoding feasible, especially on devices with limited processing power and/or energy autonomy, such as smartphones, laptops, embedded systems, robotic devices, virtual and mixed-reality glasses, among others.

3 RELATED WORKS

There are several related works in the literature focusing on complexity reduction in HEVC for accelerating video encoding with machine learning, such as [3] and [11], which target at complexity

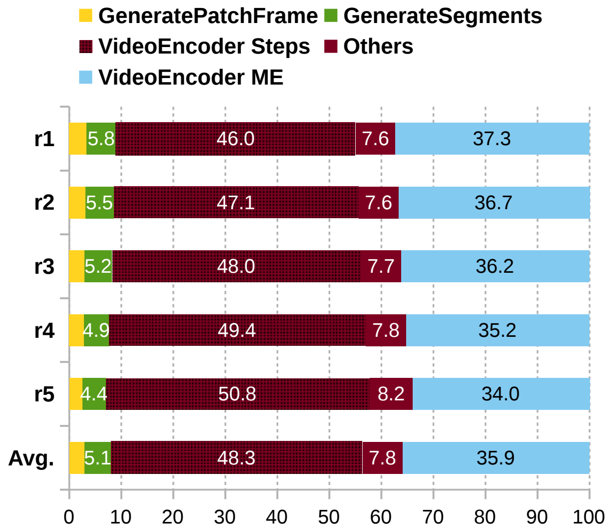


Figure 5: Encoding time percentage of high-complexity tools of V-PCC.

reduction at the CTU level for 2D videos. Also, there are some works targeting at V-PCC complexity reduction, such as [22], [7], and [12]. However, all these studies targeting V-PCC are limited to the All Intra temporal configuration.

In [3], machine learning was applied to reduce complexity by early terminating the block size decision process of HEVC Coding Tree Units. The Random Access temporal configuration was employed in the experiments. The model proposed in [3] assigns the best block sizes based on the input frame characteristics, avoiding testing all possible block sizes (64x64, 32x32, and 16x16). The work achieved an average computational complexity reduction of 37% compared to the original encoder, with a marginal increase in BD-Rate of only 0.28%.

In [11], the author employs convolutional neural networks to reduce the complexity of HEVC, predicting the decisions of the Coding Tree Unit (CTU) in the All Intra temporal configuration. The model proposed in [11] was trained and tested using large-scale database with diversiform patterns of CTU partition for each CTU depth. This approach reduces encoding time by 62.25% with BD-Rate increases of 2.12%.

Among the studies focused on machine learning to reduce the V-PCC encoding time, [22] introduces a machine learning approach for early termination targeting complexity reduction of geometry and attribute map coding at CU level. Instead of using the HEVC video encoder, the authors opted for the Versatile Video Coding standard [2], due to its ability to maintain subjective image quality, although with computational complexity up to nineteen times higher than HEVC in the All Intra configuration. The work achieved approximately 55% reduction in encoding time in V-PCC when using the modified VVC as the video encoder, compared to using VVC without the proposed method.

In [7], the author applies a cross-projection algorithm with rate-distortion-oriented decision-making at the geometry level of CU in V-PCC, focusing solely on the All Intra configuration. This approach reduced the average total encoding time by 57.8%, with BD-Rate losses for point-to-point (D1) and point-to-plane (D2) geometries of 0.08% and 0.33%, respectively, and for luma attribute a BD-Rate of 0.16%.

In [12], an unsupervised ML solution using hierarchical clustering is presented as a fast CU size decision method. Its training was based on the geometry stream, similar to other works aiming to reduce complexity in V-PCC. The presented work achieved an average reduction in encoding time of 56.7% to 69.3%, with only a slight increase in BD-Rate (D2), ranging from 0.1% to 0.5%.

To the best of the authors' knowledge, there are no works evaluating the impact of using machine learning models in the Random Access temporal configuration of V-PCC. The fast V-PCC implementation proposed in this paper is focused on Random Access due this configuration provides the best compression efficiency, but also the highest encoding complexity.

4 PROPOSED METHOD

As the compression of 2D projections of point clouds in V-PCC is performed with a 2D video encoder, we explore the impact of using existing complexity reduction solutions, developed for reducing the encoding time of 2D videos, in the context of point clouds. Thus, we employ the machine learning models proposed in [3], which utilizes decision trees to find the best Coding Tree for a CTU, thereby shortening the block partitioning process. Once the best depth of a Coding Unit is found, it is no longer necessary to continue testing at lower levels. This reduces the number of possibilities tested during the Rate-Distortion Optimization (RDO) process and consequently decreases the time required for encoding.

The HEVC standard allows up to four Coding Trees depths: 0 (64x64 CU), 1 (32x32 CU), 2 (16x16 CU), and 3 (8x8 CU). Since it is not possible to subdivide a Coding Unit at the last level, the machine learning models in [3] work over the first three depths. These models were trained using data extracted from test sequences of 2D videos, encoded with HEVC in the Random Access (RA) temporal configuration. The results obtained in [3] show an average complexity reduction of 37% with an increase in BD-Rate of 0.28%.

These same machine learning models were now adopted for use in the context of point cloud compression with V-PCC. To enable this, modifications were made to the version of the HEVC test model (HM), the HEVC reference software [19], used in the V-PCC reference software [16]. Since the version of HEVC used is newer than the one originally referenced in [3], precautions were taken to ensure equivalence between the versions. This step was necessary due to changes in the HEVC code. These models were employed in the encoding of color attributes and geometric sub-streams (Figure 2). Since more than 99% of the total V-PCC video encoding time is related to these two video streaming, we chose not to use the models in the encoding of the occupancy maps, since it does not have significant impact in the total encoding time of V-PCC, as shown in Table 1.

5 EXPERIMENTAL RESULTS

To evaluate the modifications made to the V-PCC reference software, experiments were conducted using the five 10-bit dynamic point clouds indicated in the V-PCC CTC [13]: *longdress*, *loot*, *queen*, *redandblack*, and *soldier*. Figure 6 presents the first frame of each

Table 2: Experimental results obtained from the proposed method.

Test Sequence	Geometry			Attributes			TTR (%)
	ETR (%)	ΔBitrate (%)	ΔPSNR (dB)	ETR (%)	ΔBitrate (%)	ΔPSNR (dB)	
longdress	34.74	3.31	-0.128	43.02	-0.02	-0.089	33.90
loot	40.06	3.28	-0.106	65.37	-0.30	-0.050	43.25
queen	55.40	2.10	-0.160	59.26	-1.45	-0.325	46.60
redandblack	34.45	4.03	-0.119	46.31	0.15	-0.081	34.10
soldier	49.03	1.58	-0.150	62.56	-0.57	-0.087	44.33
Average	42.73	2.86	-0.133	55.30	-0.44	-0.126	40.44

test point cloud sequence used in the experiments. Each test sequence was encoded with the modified and original V-PCC reference sofware, using the five bitrate configurations (r1, r2, r3, r4, and r5), as specified in the CTC. The first 64 frames of each sequence were encoded using the Random Access temporal configuration of V-PCC.

For each experiment, the Peak Signal-to-Noise Ratio (PSNR) of point-to-point (D1) and point-to-plane (D2) metrics [13] were calculated over the geometry information of the reconstructed point clouds. To evaluate the color information, the PSNR was calculated on the luminance channel of the attributes. These metrics were calculated using the Distortion Metric tool, as indicated in the CTC [13]. The experiments were conducted using version 22.1 of TMC2 [14] on a computer with an AMD Opteron 8276s processor and 120GB of RAM.

Figures 7-11 present the Rate-Distortion (RD) curves all 5 test sequences used. In these figures, each point represents a V-PCC encoding using one bitrate setting (r1 to r5) as well as the bitrate results (kbps) and objective quality (PSNR). By analyzing the five sequences, it is possible to observe that the greatest impacts on bitrate and objective quality were obtained in the *queen* sequence (Figure 9), as can be seen by the larger gap between the RD curves. It is also noticeable that the best RD results were obtained in the *soldier* sequence, where both geometry-related curves (D1 and D2, Figure 11 (a) and (b), respectively) and attribute curves (Figure 11 (c)) achieved results close to those obtained with the original V-PCC. Figure 12 shows the average RD curves for the geometry and attribute metrics. It is possible to see that the results for attributes present a lower loss in encoding efficiency compared to the results of V-PCC without the complexity reduction solution implemented.

longdress loot queen redandblack soldier

Figure 6: First frame of the test sequences used.

The Encoding Time Reduction (ETR) results for each sub-stream encoded and the variations in bitrate and PSNR for geometry and attribute sub-streams encoding are presented in Table 2 for each test sequence. For the geometry video sub-streams, the proposed method achieved an ETR ranging from 34.45% to 55.4%, with an average of 42.73%. For the attribute video sub-stream, an even higher ETR was obtained, ranging from 42.02% to 65.37%, with an average of 55.3%. In both cases, the average reduction was above the value obtained in [3], which was 37%. When considering the impact on the total encoding time of the V-PCC, that is, the sum of the total time spent by V-PCC, including the data reading, projection to the 2D space, and the time spent on the sub-streams HEVC encoding, the average Total Time Reduction (TTR) was 40.44%.

When evaluating the average bitrate impact on the encoded videos sub-streams, we observed an increase of 2.86% for geometry and a bitrate decrease of 0.44% for attributes. This indicates that the decisions made by the machine learning models performed better on the attribute sub-streams. This was expected, as the characteristics of the attribute videos generated in V-PCC (Figure 3) are closer to the content used (2D videos) when training the models. The objective quality (PSNR) results also support this hypothesis, with slightly better outcomes for video attribute streams.

Table 3 presents the BD-Rate results for reconstructed point clouds using the point-to-point (D1) and point-to-plane (D2) geometry metrics, as well as the BD-Rate results obtained from the Luminance channel of the color attributes. The average BD-Rate results for the D1 and D2 metrics were 4.2% and 4.75%, respectively, indicating a slightly negative impact on the encoding efficiency. The average impact on the attribute encoding efficiency was slightly better.

Table 3: BD-Rate results.

Test Sequence	D1 BD-Rate (%)	D2 BD-Rate (%)	Luma BD-Rate (%)
longdress	4.631	5.615	2.348
loot	4.536	4.960	1.259
queen	3.907	4.189	8.580
redandblack	5.526	6.174	2.255
soldier	2.400	2.816	1.606
Average	4.200	4.751	3.209

However, when analyzing the results obtained for each test sequence, it is noticeable that the Luma BD-Rate results for the *queen* sequence show a discrepant value, with an 8.58%, compared to

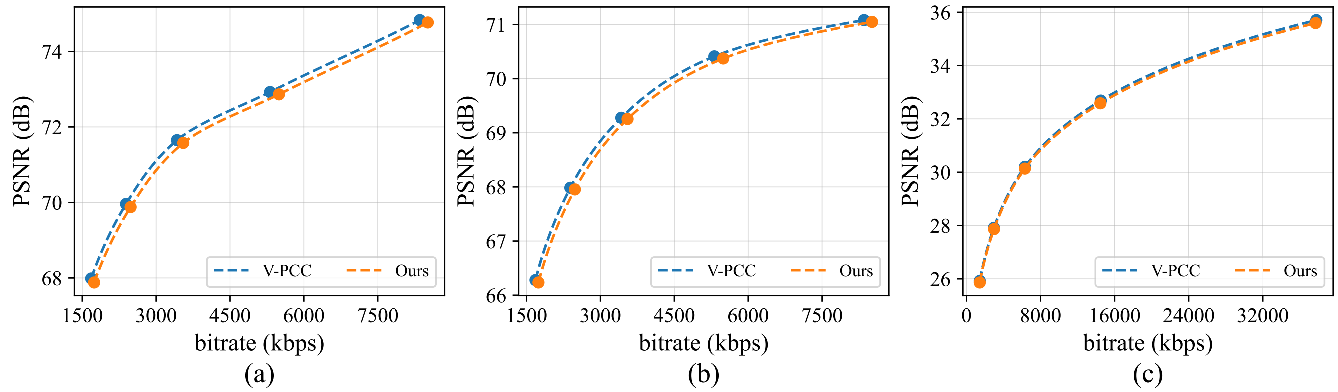


Figure 7: RD curves of geometry D1 (a), D2 (b) and Luma (attribute) (c) obtained from sequence longdress.

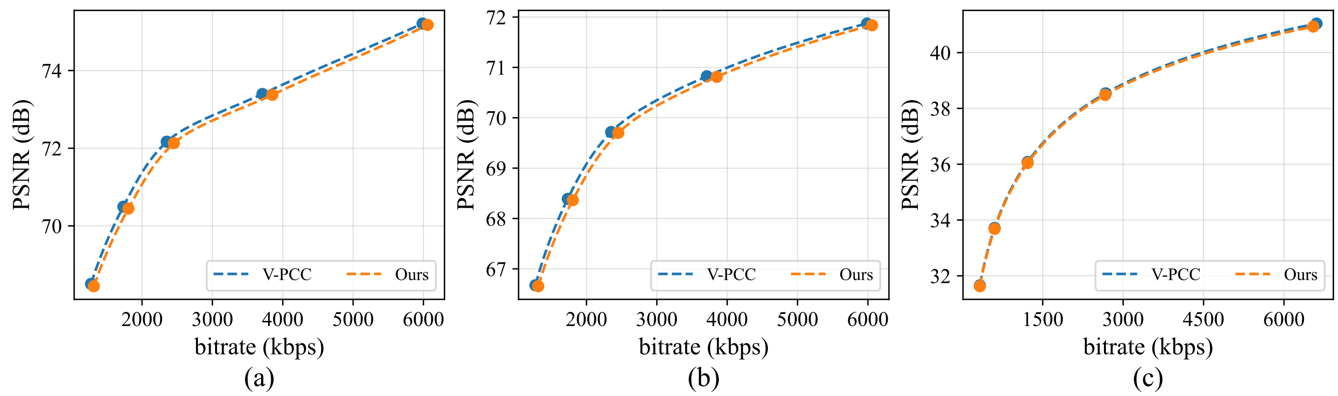


Figure 8: RD curves of geometry D1 (a), D2 (b) and Luma (attribute) (c) obtained from sequence loot.

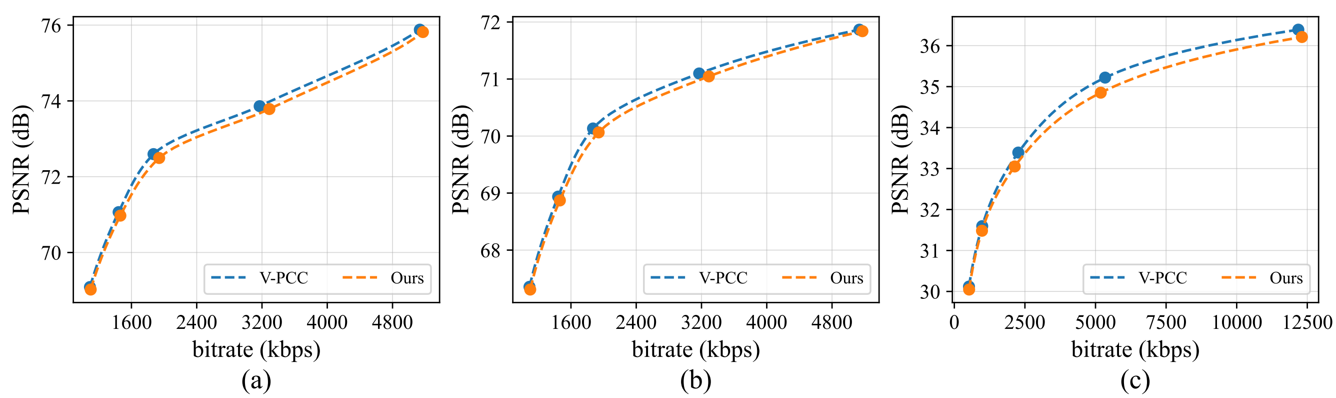


Figure 9: RD curves of geometry D1 (a), D2 (b) and Luma (attribute) (c) obtained from sequence queen.

1.26%-2.35% for the other point clouds. When analyzing the texture characteristics of the sequences used (Figure 6), it is evident that the other sequences (*longdress*, *loot*, *redandblack*, and *soldier*) feature content captured from real life, while the *queen* sequence contains synthetic content. This may justify the discrepant values, since the machine learning models were trained with data from test sequences containing real-life content [3].

When comparing the BD-Rate results obtained in [3], which showed an increase of 0.28%, we notice that the change of context presents significant impact. This indicates that although the models trained for 2D videos show interesting results in terms of time

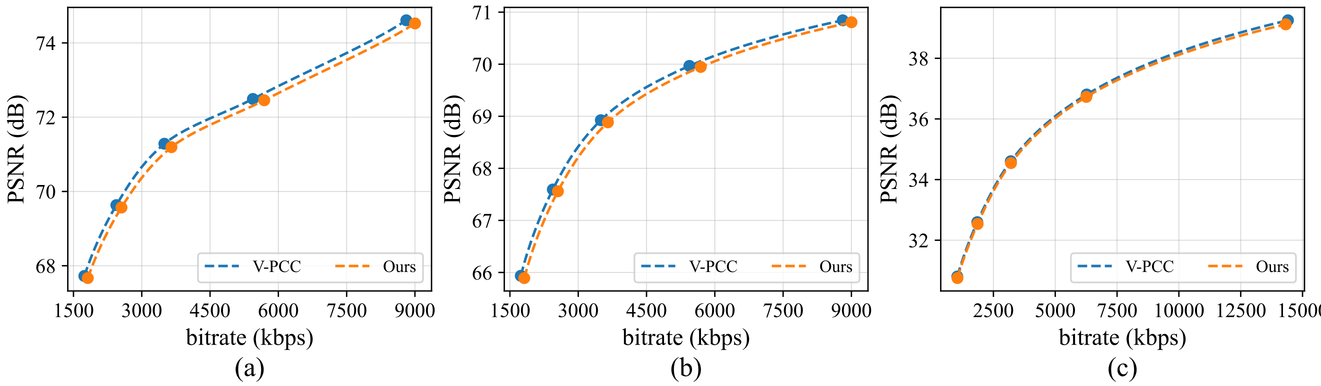


Figure 10: RD curves of geometry D1 (a), D2 (b) and Luma (attribute) (c) obtained from sequence redandblack.

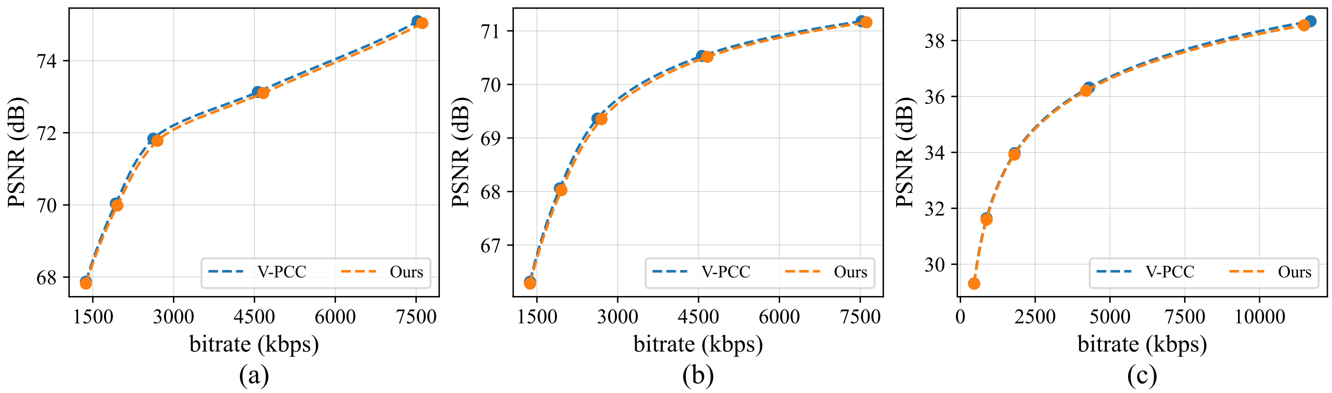


Figure 11: RD curves of geometry D1 (a), D2 (b) and Luma (attribute) (c) obtained from sequence soldier.

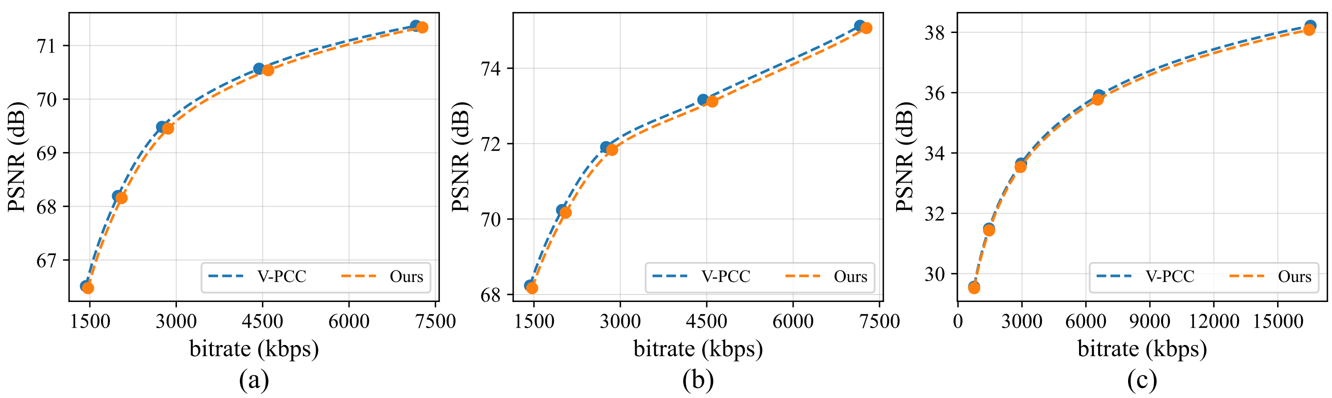


Figure 12: RD curve of average results of geometry D1 (a), D2 (b) and Luma (attribute) (c) obtained from experiments.

savings, they do not exhibit the same accuracy when used outside their original context. Furthermore, the results suggest that due to the unique characteristics of the geometry and attribute sub-streams, specialized models should achieve better results in both encoding efficiency and ETR.

Figure 13 provides a visual comparison between the point clouds reconstructed using the V-PCC reference software and our proposed fast implementation. Figure 13 (a) and (b) show the first frame of the *queen* sequence encoded with the r3 bitrate configuration. Figure 13 (c) and (d) show the first frame of the *soldier* sequence encoded with the r5 bitrate configuration. Note that in both cases

(a) (b) (c) (d)

Figure 13: Reconstructed point clouds: queen using rate setting r3 ((a) V-PCC, (b) Our Fast V-PCC), and soldier using rate setting r5 ((c) V-PCC, (d) Our Fast V-PCC)).

(the test sequences with the highest and lowest impact on encoding efficiency, respectively), there is no noticeable drop in visual quality.

6 CONCLUSION

This work presented a solution for reducing the encoding time of the video encoding stage of the V-PCC reference software, utilizing an existing machine learning model from the literature, trained for the context of 2D videos. This model was incorporated into the functionality of the V-PCC reference software. The experiments were conducted using the Random Access temporal configuration of V-PCC, delivering a reduction in encoding time of 42.73% for geometry streams and 55.3% for attribute sub-streams. The method achieved significant encoding time reduction with a minimal impact on bitrate and objective quality. The results also demonstrated that models trained for common 2D videos may not perform as well in the context of geometry sub-stream encoding as they do in attribute sub-stream encoding, indicating that specialized models could yield even better results. As future work, we plan to explore the use of specialized machine learning models for point cloud encoding, utilizing data extracted from V-PCC, with the expectation of achieving better results in encoding time reduction and BD-Rate.

REFERENCES

[1] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. *ITU SG16 Doc. VCEG-M33* (2001).

[2] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. 2021. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (2021), 3736–3764.

[3] Guilherme Correa, Pedro Assuncao, Luis A. da Silva Cruz, and Luciano Agostini. 2014. Classification-based early termination for coding tree structure decision in HEVC. In *2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 239–242.

[4] Guilherme Correa, Pedro A. Assuncao, Luciano Volcan Agostini, and Luis A. da Silva Cruz. 2015. Fast HEVC Encoding Decisions Using Data Mining. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 4 (2015), 660–673.

[5] Tianyu Dong, Kyutae Kim, and Euee S. Jang. 2021. Performance Evaluation of the Codec Agnostic Approach in MPEG-I Video-Based Point Cloud Compression. *IEEE Access* 9 (2021), 167990–168003.

[6] Jay Fenlason. 2024. Gprof. https://ftp.gnu.org/old-gnu/Manuals/gprof-2.9.1/html_mono/gprof.html. Accessed: 2024-04-15.

[7] Wei Gao, Hang Yuan, Ge Li, Zhu Li, and Hui Yuan. 2023. Low Complexity Coding Unit Decision for Video-Based Point Cloud Compression. *IEEE Transactions on Image Processing* 33 (2023), 149–162.

[8] Danillo Graziosi, Ohji Nakagami, Satoru Kuma, Alexandre Zaghetto, Teruhiko Suzuki, and Ali Tabatabai. 2020. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing* 9 (2020), e13.

[9] Dan Grois, Detlev Marpe, Amit Mulayoff, Benaya Itzhaky, and Ofer Hadar. 2013. Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders. In *2013 Picture Coding Symposium (PCS)*. 394–397.

[10] Il-Koo Kim, Junghye Min, Tammy Lee, Woo-Jin Han, and JeongHoon Park. 2012. Block partitioning structure in the HEVC standard. *IEEE transactions on circuits and systems for video technology* 22, 12 (2012), 1697–1706.

[11] Tianyi Li, Mai Xu, and Xin Deng. 2017. A deep convolutional neural network approach for complexity reduction on intra-mode HEVC. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1255–1260.

[12] Yue Li, Jun Huang, Chaofeng Wang, and Hongyue Huang. 2024. Unsupervised learning-based fast CU size decision for geometry videos in V-PCC. *Journal of Real-Time Image Processing* 21, 1 (2024), 11.

[13] MPEG. 2020. Common Test Conditions for V3C and V-PCC. *ISO/IEC JTC 1/SC 29/WG 11* (2020).

[14] MPEG. 2024. Video Point Cloud Compression - VPCC - mpeg-pcc-tmc2 test model candidate software. https://github.com/MPEGGroup/mpeg-pcc-tmc2.

[15] Sang-hyo Park and Je-Won Kang. 2020. Fast multi-type tree partitioning for versatile video coding using a lightweight neural network. *IEEE Transactions on Multimedia* 23 (2020), 4388–4399.

[16] Marius Preda. 2020. V-PCC codec description. *ISO/IEC JTC 1/SC 29/WG 7, Virtual* (2020).

[17] R Schaefer. 2017. Call for proposals for point cloud compression V2. In *ISO/IEC JTC1 SC29/WG11 MPEG, 117th Meeting. Hobart, TAS.*

[18] Yun Song, Biao Zeng, Miaohui Wang, and Zelin Deng. 2022. An efficient low-complexity block partition scheme for VVC intra coding. *Journal of Real-Time Image Processing* (2022), 1–12.

[19] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology* 22, 12 (2012), 1649–1668.

[20] Vivienne Sze, Madhukar Budagavi, and Gary J Sullivan. 2014. High efficiency video coding (HEVC). In *Integrated circuit and systems, algorithms and architectures*. Vol. 39. Springer, 40.

[21] Alexandre Tissier, Wassim Hamidouche, Jarno Vanne, F Galpin, and Daniel Menard. 2020. CNN oriented complexity reduction of VVC intra encoder. In *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3139–3143.

[22] Yihan Wang, Yongfang Wang, Tengyao Cui, and Zhijun Fang. 2024. Fast Video-Based Point Cloud Compression Based on Early Termination and Transformer Model. *IEEE Transactions on Emerging Topics in Computational Intelligence* (2024).

[23] Natasha Westland, André Seixas Dias, and Marta Mrak. 2019. Decision trees for complexity reduction in video compression. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2666–2670.

[24] Yun Zhang, Sam Kwong, and Shiqi Wang. 2020. Machine learning based video coding optimizations: A survey. *Information Sciences* 506 (2020), 395–423.