

A Coding Efficiency Analysis of V-PCC Substreams on GPU and Mobile Hardware Platforms for Point Cloud Compression

Allan Schuch
anschuch@inf.ufpel.edu.br
ViTech - UFPel
Pelotas, Rio Grande do Sul, Brasil

Gustavo Rehbein
ghrehbein@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas, Rio Grande do Sul, Brasil

Vitor Costa
vscosta@inf.ufpel.edu.br
ViTech - UFPel
Pelotas, Rio Grande do Sul, Brasil

Daniel Palomino
dpalomino@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas, Rio Grande do Sul, Brasil

Luciano Agostini
agostini@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas, Rio Grande do Sul, Brasil

Marcelo Porto
porto@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas, Rio Grande do Sul, Brasil

ABSTRACT

Dynamic point clouds demand efficient compression due to their large data volumes. The MPEG Video-based Point Cloud Compression (V-PCC) standard addresses this by projecting 3D data into 2D substreams encoded with HEVC. While the HEVC Test Model (HM) ensures high efficiency, its complexity hinders practical use. This paper analyzes the coding efficiency of two hardware HEVC encoders when processing V-PCC video substreams: Nvidia NVENC on an RTX 4070 Ti GPU and the Snapdragon 8 Gen 2 on a Samsung Galaxy S23+. Geometry, attribute, and occupancy substreams are encoded directly on each platform and compared with HM under V-PCC test conditions. Results show significant coding efficiency losses for the video substreams: NVENC increases BD-Rate by 167.41% (geometry) and 182.06% (attributes), while Snapdragon achieves 255.61% and 327.72%. For occupancy maps, NVENC required 18.38% more bitrate in lossless mode, whereas Snapdragon achieved 99.9% accuracy with lossy encoding.

KEYWORDS

GPU, V-PCC, HEVC, mobile, hardware, substreams

1 INTRODUCTION

The growing use of dynamic point clouds for representing 3D content introduces substantial challenges in terms of data storage and transmission, necessitating highly efficient compression techniques. To address these challenges, the Moving Picture Experts Group (MPEG) introduced the Video-based Point Cloud Compression (V-PCC) standard, which converts 3D point cloud data into multiple 2D video substreams. These substreams can then be compressed using conventional video codecs [10]. Among available codecs, High Efficiency Video Coding (HEVC) [12] offers superior compression and broader hardware support. The V-PCC reference implementation, Test Model Category 2 (TMC2) [13], uses the software-based

HEVC Test Model (HM) [11] for encoding. While HM provides excellent compression, its high computational cost leads to long encoding times. Hardware-based encoders in Graphics Processing Units (GPUs) and mobile devices present an efficient alternative, accelerating encoding and improving power efficiency [1]. However, these implementations must balance coding efficiency with architectural and power constraints, often simplifying or omitting certain standard coding tools.

While existing literature has investigated the implemented HEVC tools and their coding efficiency impact in 2D video encoding on mobile hardware [3, 4], those analyses rely on emulating hardware constraints within a software reference, an approach that may not accurately reflect the on-device coding efficiency. Other studies compare the efficiency of GPU hardware and software encoders [1] for traditional 2D video. Additionally, recent works focus on reducing the complexity of the V-PCC standard, proposing machine learning techniques to accelerate the encoding process [20, 21]. However, the evaluation of the coding efficiency of current hardware platforms when compressing the video substreams generated within the V-PCC pipeline remains unexplored.

To address these gaps, this paper presents a comparative coding efficiency analysis of two hardware encoders when applied to V-PCC video substreams: Nvidia NVENC on the RTX 4070 Ti GPU [15] and Snapdragon 8 Gen 2 [19] on the Samsung Galaxy S23+ [22]. The geometry, attribute, and occupancy map substreams are encoded on each platform and evaluated against the HM reference. All metrics are derived from on-device encodings, ensuring the results reflect the practical hardware capabilities. This analysis offers insights into how these hardware encoders, optimized for traditional video, perform when compressing the video substreams, a stage that affects the coding efficiency of the final reconstructed point cloud.

2 VIDEO-BASED POINT CLOUD COMPRESSION

Point clouds are a digital representation of 3D objects and scenes, and they are increasingly used in applications such as cultural heritage preservation, autonomous driving, medical

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2026). Lavras/MG, Brazil. Porto Alegre: Brazilian Computer Society, 2026.

© 2026 SBC – Brazilian Computing Society.
ISSN 2966-2753

imaging, and immersive environments like virtual and augmented reality. A point cloud is composed of sets of points in three-dimensional space, where each point may include attributes such as color, reflectance, or texture. When sequences of point clouds are organized over time, they form dynamic point clouds, analogous to frames in a video sequence. These dynamic representations can contain millions of points per frame, leading to extremely large data volumes and significant challenges for storage, transmission, and real-time processing.

To mitigate these challenges, compression becomes essential. The MPEG standardization group introduced the Video-based Point Cloud Compression (V-PCC) method for dynamic point clouds [18]. V-PCC exploits established video coding techniques by projecting each 3D point cloud frame onto a 2D domain. In this process, frames are segmented into patches that are packed into 2D images, which can then be efficiently compressed by conventional video encoders. Since projecting the 3D data onto 2D images results in the loss of depth information, auxiliary data must also be transmitted to enable accurate reconstruction. To address this, V-PCC employs multiple video substreams: an attribute substream to encode color information, a geometry substream to represent the depth information, and an occupancy substream with binary data that marks the regions of each frame containing valid points. These substreams are individually compressed using standard video codecs, and the final V-PCC bitstream is formed by combining them together with side information describing the patch structure [10].

Although the V-PCC implementation using the HM encoder provides good coding efficiency, it is not a real-world scenario for point cloud coding, especially when real-time is considered. Therefore, it is important to analyze the coding efficiency delivered by current hardware-based HEVC encoders, such as the Nvidia NVENC and the Snapdragon 8 Gen 2, when encoding the substreams of point cloud data in comparison with the HM implementation.

3 ANALYSIS METHODOLOGY

The methodology presented in this work is based on a comparative analysis of the coding efficiency of two distinct hardware encoders when applied to the V-PCC framework. This approach focuses on evaluating the practical coding efficiency of each encoder by comparing their rate-distortion results against the software reference at the video substream level.

3.1 V-PCC substreams and baseline data generation

To ensure a fair basis for the rate-distortion comparison, a set of reference anchor points was first established. Using the V-PCC reference software, TMC2 version 22.1 (hash 0efc4c5) [13], the initial 32 frames from the dynamic point cloud test sequences (Classes A, B, and C) of the V-PCC's Common Test Conditions (CTC) [14] were processed. This step converted the point cloud data into the necessary 2D video substreams for geometry, attribute, and occupancy map. The 32-frame length was selected because it aligns with the

default Group of Frames (GOF) size defined in the V-PCC reference implementation.

The encoding of all these substreams is handled by the TMC2 software, which internally calls the HM software (version 16.20+scm 8.8) [11]. For the analysis, the geometry and attribute substreams were encoded, employing the standard lossy Random-Access (RA) profile of the HM. In contrast, the occupancy map is encoded in lossless mode in the standard configuration. This ensures the occupancy map remains intact, where any data loss would severely degrade the final reconstructed point cloud quality.

To enable a comprehensive comparison for this work, two distinct baselines were generated for the occupancy map. The standard lossless baseline was created using the default lossless RA profile of the HM. A separate lossy baseline was also created to allow for a fair comparison with the mobile hardware encoder, as the available MediaCodec [7, 8] codec on that platform does not provide an option for lossless encoding. This lossy baseline was generated by employing a lossy RA profile in HM and adding the following parameter to the V-PCC command line: `-offsetLossy0M=255`. This parameter serves to increase the separation between the values representing valid and invalid points in the occupancy map, which helps prevent the binary values (zeros and ones) from being lost during lossy compression. While it is known that lossy occupancy maps compromise final 3D reconstruction, evaluating this configuration is necessary due to mobile platform limitations, enabling an analysis of how much occupancy map information is lost during this process.

All these configurations were run using the five recommended V-PCC rate points (r1 to r5), where r1 offers the lowest bitrate and r5 provides the best quality at a higher bitrate [14]. The resulting (bitrate, PSNR) pairs became the baseline for the subsequent hardware encoder evaluations, while the recorded bitrates also became the target bitrates for the hardware encodings.

3.2 V-PCC substream encoding on the NVENC hardware

The NVENC encoding processes were executed on a platform configured with an Intel Core i9-10900 CPU, 32GB of DDR4 memory, and an Nvidia GeForce RTX 4070 Ti (12GB) graphics card [15], running on Ubuntu 22.04.5 LTS. To manage the encoding of the V-PCC substreams, the `hevc_nvenc` encoder [17], a component of the FFmpeg software (version N-117204-gc65a294f79) [6], was utilized.

To optimize encoding for diverse scenarios, the NVENC API offers a selection of presets and tuning profiles [16]. For the lossy encodings of the geometry and attribute substreams, the P7 preset with UHQ tuning info was selected, as it provides the most comprehensive set of enabled HEVC tools (the parameters employed by this configuration are detailed in [23]). For the lossless encoding of the occupancy map, the Lossless tuning profile was employed. The RTX 4070 Ti GPU is equipped with dual NVENC instances that can run in parallel via the Split Frame feature [16]. However, this

feature was disabled to ensure an isolated evaluation of a single encoder instance. The Constant Bit Rate (CBR) mode was employed for rate control, with the target bitrate for each test aligned with the corresponding anchor point established by the rates (r1-r5) in the HM-16.20 reference. Additionally, each of the substreams was encoded at a fixed frame rate of 30 fps, a setting that mirrors the default configuration of the HM encoder within the V-PCC framework. The full command line for the NVENC encodings is presented below:

```
$ ffmpeg -hwaccel cuda -hwaccel_output_format cuda -r 30
-s [resolution] -pix_fmt yuv420p -i input.yuv -c:v
hevc_nvenc -b:v [target_bitrate] -preset p7 -tune
[tuning] -split_encode_mode disabled -f hevc output.bin
```

3.3 V-PCC substream encoding on the mobile hardware

A similar methodology was applied for the Snapdragon 8 Gen 2 dedicated encoder, with experiments conducted directly on a Samsung Galaxy S23+ device [22]. The encodings were managed using the Termux application [5] on an Android 15 OS [9]. FFmpeg [6] was used to access the hardware encoder via the native MediaCodec API [8].

For the encoding, the MediaCodec codec was used with its default configuration (the parameters employed by this setup are detailed in [23]). Explicit configuration was limited to the target bitrate, which was set to match the HM reference points, and the rate control mode, which was configured as CBR. A frame rate of 30 fps was maintained across all substreams to align with the default behavior of the HM encoder within the V-PCC framework. It is important to note that the MediaCodec API does not support lossless encoding. Therefore, for the occupancy map, the lossy encoding approach, as the only available option, was performed on this platform, using the same configuration as the geometry and attribute substreams. The command line employed for the Snapdragon 8 Gen 2 encodings is presented below:

```
$ ffmpeg -hwaccel auto -s [resolution] -r 30 -pix_fmt
yuv420p -i input.yuv -c:v hevc_mediacodec -b:v
[target_bitrate] -bitrate_mode cbr -f hevc output.bin
```

3.4 Coding efficiency evaluation

After the encoding stage, the coding efficiency for the video substreams was quantified using metrics tailored to each analysis case. For the geometry and attribute substreams, the BD-Rate and BD-PSNR [2] metrics were calculated for both NVENC and Snapdragon against the HM baseline. For the occupancy map, two distinct evaluation approaches were employed, one for each hardware platform. For the NVENC, which supports lossless encoding, the occupancy map substreams were encoded losslessly, and the resulting average bitrate across the five rate configurations was directly compared to the lossless HM baseline, with the difference expressed as a percentage variation.

In contrast, the Snapdragon 8 Gen 2 encoder lacks lossless encoding support, requiring a different approach for the occupancy map. Since lossy compression can cause pixel values

to deviate from original values (0 or 255), the decoded maps were re-binarized using a threshold of 127. As the occupancy map holds binary data rather than visual content, the PSNR is not the most suitable metric for measuring the accuracy of its values after encoding. As a result, the accuracy of the occupancy map was measured by the percentage of correctly matched points against the original map. This led to an evaluation for the Snapdragon based on a direct comparison of two metrics against the lossy HM baseline: the average bitrate variation and the occupancy map accuracy.

4 CODING EFFICIENCY IMPACT EVALUATION

The coding efficiency losses for the video substreams of the NVENC and Snapdragon 8 Gen 2 hardware encoders, when compared to the HM-16.20 reference, are quantified by the BD-Rate, BD-PSNR, and occupancy map variation metrics presented in Table 1. The results reveal a significant coding efficiency gap for the video substreams across both hardware implementations and all test cases, though the magnitude of this gap differs substantially between the consumer GPU and the mobile platform dedicated encoders. This disparity reflects the distinct design trade-offs made in each architecture, which prioritize performance and power efficiency over pure compression effectiveness. The average encoding time results starkly evidence this prioritization: while the HM reference software, running on a system with an Intel Xeon E5-4650 v3 processor, required, on average, about 105 minutes for geometry, 92 minutes for attribute, and about 2 minutes for the occupancy map substreams, the hardware encoders demonstrated a substantial increase in speed, with NVENC taking just over one second and the Snapdragon less than three seconds for the same tasks. These encoding times for the geometry and attribute substreams correspond to 64 frames, due to the use of near and far layers in V-PCC [18], while the occupancy map substream spanned 32 frames.

The NVENC encoder exhibits an overall average BD-Rate increase of 167.41% for geometry and 182.06% for attribute substreams. This considerable coding efficiency loss is an expected result of hardware-accelerated encoding, where exhaustive search algorithms and complex coding decisions available in the HEVC reference software are streamlined to meet real-time processing demands. Although the GPU provides a large power budget, its dedicated encoder is still optimized for speed, resulting in a noticeable but relatively contained efficiency gap compared to the mobile platform.

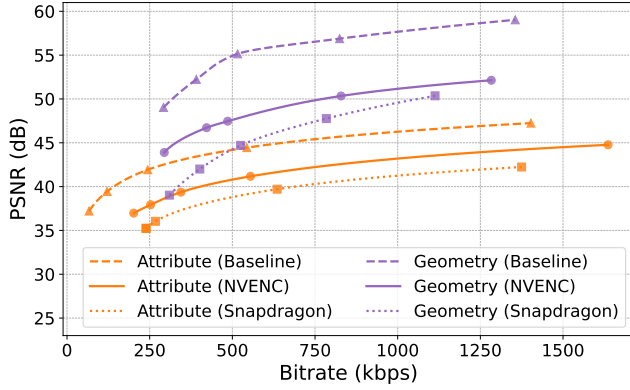
On the other hand, the Snapdragon 8 Gen 2 encoder shows a much more severe coding efficiency loss, with an overall average BD-Rate increase of 255.61% for geometry and 327.72% for attribute substreams. This more pronounced efficiency gap is likely a consequence of the power, thermal, and area constraints inherent to mobile System on a Chip designs. These limitations likely require more aggressive hardware simplifications, which significantly impact the encoder's ability to achieve the compression levels of the HM reference or even the more powerful consumer GPU encoder.

Table 1: BD-Rate, BD-PSNR, and Occupancy Map variation results, compared to HM-16.20.

Class	Sequence	NVENC (RTX 4070 Ti)					Snapdragon 8 Gen 2					
		Geometry		Attribute		Occupancy*	Geometry		Attribute		Occupancy**	
		BD-Rate (%)	BD-PSNR (dB)	BD-Rate (%)	BD-PSNR (dB)		BD-Rate (%)	BD-PSNR (dB)	BD-Rate (%)	BD-PSNR (dB)	Bitrate Var. (%)	Avg. Map Accuracy (%)
A	Queen	112.31	-5.03	138.48	-2.86	12.72	232.63	-9.50	317.49	-4.53	11.79	99.95
	Loot	163.84	-6.54	187.01	-3.46	12.45	228.43	-9.72	395.17	-5.39	14.48	99.93
	Red_and_Black	181.11	-6.56	124.79	-3.17	24.41	251.72	-8.91	167.10	-3.89	11.51	99.94
	Soldier	117.66	-4.78	206.69	-3.72	12.41	278.44	-9.59	548.10	-6.41	14.40	99.80
B	Long_dress	189.86	-6.82	155.59	-3.18	24.08	225.15	-8.88	183.88	-3.69	10.26	99.89
C	Basketball_player	201.70	-7.38	245.57	-3.93	20.30	305.19	-10.56	367.84	-4.99	7.97	99.95
	Dancer	205.37	-7.49	216.30	-4.11	22.26	267.68	-10.76	314.53	-5.41	8.01	99.84
Overall Average		167.41	-6.37	182.06	-3.49	18.38	255.61	-9.70	327.72	-4.90	11.20	99.90

* Lossless encoding of the occupancy map by HM and NVENC resulted in no PSNR variation, with the only change being in the average bitrate.

** Lossy encoding for both HM and Snapdragon 8 Gen 2. The second metric is the occupancy map accuracy.

**Figure 1: RD curves for the geometry and attribute substreams of the *Loot* test sequence.**

It should be noted that the highest BD-Rate losses for both hardware encoders are consistently observed in the Class C sequences. While likely attributed to their higher geometry precision (11-bit) and resulting higher atlas resolutions, other sequence characteristics like content complexity and motion may also contribute. This amplified complexity exacerbates the negative impact of the hardware-driven simplifications, leading to larger efficiency losses.

The analysis of the occupancy map was evaluated across two distinct scenarios. In the lossless comparison between NVENC and HM, NVENC showed an average bitrate variation of 18.38%, suggesting that its entropy coding stage is less optimized for binary data than the HM implementation. Conversely, the Snapdragon encoder does not support lossless encoding. When compared against a lossy HM baseline, it shows a smaller average bitrate increase of 11.20% and achieves an average of 99.90% occupancy map accuracy. While this accuracy appears very high, any loss in the occupancy map can have a significant impact on the final reconstructed point cloud quality. These errors represent incorrect information about the validity of points, which can lead to a significant degradation in visual quality. Therefore, this result suggests that while lossy occupancy map encoding

is functional on constrained platforms, it carries a notable risk of introducing critical reconstruction errors.

Figure 1 provides a visual representation of the coding efficiency gaps for the video substreams of the *Loot* sequence, chosen as a representative case because its BD-Rate results for both platforms approximate the overall averages presented in Table 1. However, it is important to acknowledge a methodological limitation regarding the BD-Rate calculation revealed by the curves. The accuracy of this metric relies on sufficient overlap between the PSNR ranges of the two curves being compared. In several instances, particularly for the geometry substreams, the hardware encoders could not achieve quality levels comparable to the HM baseline within the tested bitrate range, resulting in poor curve overlap. Thus, the BD-Rate values in these cases are derived from extrapolation rather than interpolation. While they still correctly indicate a significant coding efficiency loss, they should be interpreted as approximations of the efficiency gap rather than precise measurements.

5 CONCLUSIONS

This paper presented a comparative coding efficiency analysis of the Nvidia NVENC and Snapdragon 8 Gen 2 hardware encoders for the compression of the video substreams of dynamic point clouds within the V-PCC framework. The on-device evaluation revealed that both architectures incur significant BD-Rate losses for the video substreams against the HM reference software, with the power-constrained Snapdragon 8 Gen 2 showing a much more severe efficiency gap than the high-performance NVENC. The occupancy map analysis also showed a higher bitrate cost for the hardware encoders under their respective lossy and lossless configurations. By evaluating these hardware encoders outside their native domain of natural video, this work provides a valuable benchmark on their video encoding limitations when tasked with compressing non-traditional V-PCC content. Future work will extend this analysis to an end-to-end evaluation, using 3D quality metrics to quantify the impact of these hardware constraints on the reconstructed point cloud quality.

ETHICAL CONSIDERATIONS

Not applicable.

ACKNOWLEDGMENTS

The authors acknowledge the use of large language models throughout the manuscript to assist with text editing, style and grammar refinement.

REFERENCES

- [1] Kasidis Arunruangsirilert and Jiro Katto. 2024. Evaluation of Hardware-based Video Encoders on Modern GPUs for UHD Live-Streaming. In *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*. 1–9. doi:10.1109/ICCCN61486.2024.10637525
- [2] G. Bjontegaard. 2008. *Improvements of the BD-PSNR model*. Technical Report VCEGA11. VCEG.
- [3] Vitor Costa, Murilo Perleberg, Luciano Agostini, and Marcelo Porto. 2024. Coding Efficiency and Time Evaluation of Apple A15 Bionic Chipset HEVC Encoder. In *2024 IEEE 15th Latin America Symposium on Circuits and Systems (LASCAS)*. 1–5. doi:10.1109/LASCAS60203.2024.10506151
- [4] Vitor Costa, Murilo Perleberg, Luciano Agostini, and Marcelo Porto. 2025. Análise da Eficiência de Compressão do Codificador HEVC Embarcado no Smartphone Samsung Galaxy S23 Plus. In *Proceedings of the 31st Brazilian Symposium on Multimedia and the Web (Rio de Janeiro/RJ)*. SBC, Porto Alegre, RS, Brasil, 516–520. doi:10.5753/webmedia.2025.16013
- [5] Termux Developers. 2022. *Termux*. Retrieved June 18, 2026 from <https://github.com/termux/termux-app/releases/tag/v0.118.3>
- [6] Fast Forward Moving Pictures Experts Group (FFmpeg). [n. d.]. *FFmpeg Multimedia Framework*. Retrieved June 18, 2026 from <https://ffmpeg.org/>
- [7] Fast Forward Moving Pictures Experts Group (FFmpeg). [n. d.]. *FFmpeg Codecs Documentation: MediaCodec*. Retrieved June 18, 2026 from <https://ffmpeg.org/ffmpeg-codecs.html#MediaCodec>
- [8] Google. [n. d.]. *MediaCodec / Android Developers*. Retrieved June 18, 2026 from <https://developer.android.com/reference/android/media/MediaCodec>
- [9] Google. 2024. *Android Developers - Android 15*. Retrieved June 18, 2026 from <https://developer.android.com/about/versions/15>
- [10] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. 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. doi:10.1017/ATSIP.2020.12
- [11] Fraunhofer HHI. 2018. *HEVC Test Model (HM) HM-16.20+SCM-8.8*. Retrieved June 18, 2026 from https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.20+SCM-8.8/
- [12] ITU-T. 2024. Recommendation H.265: High efficiency video coding. Standard H.265 (07/24).
- [13] MPEG. [n. d.]. *Video point cloud compression - VPCC - mpeg-pcctmc2 test model candidate software*. Retrieved June 18, 2026 from <https://github.com/MPEGGroup/mpeg-pcc-tmc2/tree/release-v22.1>
- [14] MPEG. 2020. *Common Test Conditions for V3C and V-PCC*. Technical Report N19518. ISO/IEC JTC1/SC29/WG11.
- [15] NVIDIA. 2023. *GeForce RTX 4070 Family*. Retrieved June 18, 2026 from <https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4070-family/>
- [16] NVIDIA. 2025. *NVENC Video Encoder API Programming Guide*. Retrieved June 18, 2026 from <https://docs.nvidia.com/video-technologies/video-codec-sdk/13.0/nvenc-video-encoder-api-prog-guide/index.html>
- [17] NVIDIA. 2025. *Using FFmpeg with NVIDIA GPU Hardware Acceleration*. Retrieved June 18, 2026 from <https://docs.nvidia.com/video-technologies/video-codec-sdk/13.0/ffmpeg-with-nvidia-gpu/index.html>
- [18] Marius Preda. 2020. V-PCC codec description. https://www.mpeg.org/wp-content/uploads/mpeg_meetings/134_OnLine/w20352.zip
- [19] Qualcomm. 2022. *Snapdragon 8 Gen 2 Mobile Platform*. Retrieved June 18, 2026 from <https://www.qualcomm.com/products/mobile/dragon/smartphones/snapdragon-8-series-mobile-platforms/snapdragon-8-gen-2-mobile-platform>
- [20] Gustavo Rehbein, Guilherme Corrêa, Cristiano Santos, and Marcelo Porto. 2025. Método Rápido para Codificação de Nuvens de Pontos Dinâmicas Baseado em Predição de Particionamento de Blocos. In *Proceedings of the 31st Brazilian Symposium on Multimedia and the Web (Rio de Janeiro/RJ)*. SBC, Porto Alegre, RS, Brasil, 358–365. doi:10.5753/webmedia.2025.15951
- [21] Gustavo Rehbein, Eduardo Costa, Guilherme Corrêa, Cristiano Santos, and Marcelo Porto. 2024. A Machine-Learning-Driven Fast Video-based Point Cloud Compression (V-PCC). In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web (Juiz de Fora/MG)*. SBC, Porto Alegre, RS, Brasil, 20–27. doi:10.5753/webmedia.2024.242069
- [22] Samsung. 2023. *Galaxy S23+ Tech Specs*. Retrieved June 18, 2026 from <https://www.samsungmobilepress.com/media-assets/galaxy-s23-plus?tab=specs>
- [23] Allan Schuch, Vitor Costa, Daniel Palomino, and Marcelo Porto. 2025. Analysis of Real-Time Hardware-based HEVC Encoders on GPU and Mobile Platforms. In *2025 International Conference on Visual Communications and Image Processing (VCIP)*. 1–5. doi:10.1109/VCIP67698.2025.11396915