Fast VVC Angular Intra-Prediction for 360° Videos Based on Decision Trees

Ramiro Viana
rgsviana@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas/RS, Brazil

Iago Storch
icstorch@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas/RS, Brazil

Rogério Rosado
rcassanta@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas/RS, Brazil

Marcelo Porto
porto@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas/RS, Brazil

Guilherme Corrêa
gcorrea@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas/RS, Brazil

Luciano Agostini
agostini@inf.ufpel.edu.br
PPGC, ViTech - UFPel
Pelotas/RS, Brazil

ABSTRACT

The rapid growth of immersive media has positioned 360° videos as a key component in applications such as virtual reality, remote education, tourism, and interactive entertainment. But the massive data volume required to represent 360° videos imposes significant computational challenges to deliver this type of content across the high diversity of current and future devices with support for multimedia. Then, highly efficient video encoding algorithms are required in this scenario. Versatile Video Coding (VVC) is the current state-of-the-art standard for video compression, offering specialized tools for efficient 360° video coding. This paper introduces a Machine Learning-based approach to reduce the computational effort of VVC Angular Intra-Prediction (AIP) tool when encoding 360° video content. The proposed method uses a Decision Tree model to adaptively skip vertical prediction modes in the AIP process. Experimental results show an average encoding time reduction of 10.11% with only a 0.66% impact on coding efficiency. To the best of our knowledge, this is the first work to explore the use of Machine Learning to reduce the computational effort of AIP for 360° videos.

KEYWORDS

360° videos, VVC, Angular Intra-Prediction, Machine Learning

1 INTRODUCTION

According to the 2024 Global Internet Phenomena Report, major providers like Google, Facebook, and Netflix account for 65% of fixed and 68% of mobile internet traffic, mainly due to the widespread use of digital video [20]. The need to transmit and store these videos efficiently has driven the development of encoders that compress data while preserving quality.

In recent years, there has been rapid development in Virtual Reality (VR) and immersive technologies, with 360° videos playing an important role in this progress [29]. A 360° video, also known as omnidirectional video, captures all viewpoints of a scene from a single fixed point in space [25]. This type of immersive technology has demonstrated its usefulness across various domains, including entertainment, education, training, and commercial applications such as clothing retail, real estate, and others. Although these and many other applications have gained popularity in recent years, efficient encoding of 360° videos remains crucial to ensure their continued growth and future success [26].

A 360° video is originally spherical in nature, which means it cannot be directly encoded using traditional 2D video encoders. But, there is no specific coding standard designed to encode 360° videos in their spherical format. However, technical committees, academia, and standardization organizations are making efforts to ensure interoperability between 360° video ecosystems to avoid market fragmentation [25].

One common solution is to project 360° videos into a 2D rectangular surface, enabling their encoding with conventional 2D video encoders. Among the available projection schemes, Equirectangular Projection (ERP) is one of the most widely used for 360° video encoding [26]. ERP maps the coordinates of a sphere onto a Cartesian coordinate system, where the horizontal axis represents longitude and the vertical axis represents latitude. However, because a sphere cannot be perfectly flattened onto a 2D plane, the ERP introduces distortions, particularly in regions farther from the equator, where these artifacts become more pronounced [3]. An example of ERP within a 360° sequence is illustrated in Figure 1 using a frame from the *Gaslamp* sequence [8].

Figure 1: Example of a 360° degree video in ERP format.

One of the few video encoders developed to efficiently support 360° video is the Versatile Video Coding (VVC) standard [10], which was released in July 2020 and represents the current state of the art in video coding [11]. Like previous codecs, VVC introduces new tools that improve coding efficiency. Compared to its predecessor, the High Efficiency Video Coding (HEVC) [24], VVC achieves up

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2025). Rio de Janeiro, Brazil. Porto Alegre: Brazilian Computer Society, 2025.

ISSN 2966-2753

to a 50% reduction in bit rate while maintaining the same visual quality [9]. The ability to efficiently encode 360° video is a feature that VVC inherits from HEVC.

This work presents a Machine Learning-based solution to reduce the VVC encoding effort when processing 360° videos, while causing a minor coding efficiency impact. The focus of this solution is the Angular Intra-Prediction (AIP) [6] tool of the VVC. A Decision Tree (DT) model was trained to decide if the vertical modes of the VVC AIP should be skipped or not when processing the current block. The hypothesis is that vertical modes tend to be less relevant in polar regions in function of the horizontal-wise distortion caused by the ERP projection. Then, skipping the evaluation of these modes when they are less relevant seems to be an efficient approach to reduce the computational effort without impact on coding efficiency. The use of DTs was defined because this type of Machine Learning model has a low computational cost and is hardware-friendly, which is important for a real scenario of application. A DT is built with if-then-else statements, which are easy to implement in hardware and add little additional processing effort.

There are some works in the literature focusing on computational effort reduction of the VVC Intra-Frame Prediction when processing 360° videos. The main works are [16], [30], [12], [22], [31], [17], and [21]. Most of these works, like [16], [30], [12], [22], [31], and [17], are focusing on the partitioning process and not on the AIP tool. Only a previous work of our group focused on the AIP tool [21]. Besides, some of these works used heuristics, like [30], [22], and [21], while others used Machine Learning approaches, like [16], [12], [31], and [17]. Then, the current work is the first in the literature using a Machine Learning-based solution to reduce the computational effort of the VVC Intra-Frame Prediction when processing 360° videos.

2 VVC INTRA-FRAME PREDICTION

The Intra-Frame Prediction is responsible for exploiting spatial redundancy within a video frame to improve encoding efficiency. In the VVC standard, the computational effort of Intra-Frame Prediction has increased significantly compared to previous standards [15]. This is a function of the new Intra-Frame Prediction tools introduced by VVC and also of the improvements in tools inherited from the HEVC standard. This way, VVC better leverages spatial redundancy and optimizes the video coding process [1].

Some of these novel or improved Intra-Frame Prediction tools in VVC are [9]: extended Angular Intra-Prediction (AIP), Wide-Angle Intra Prediction (WAIP), 4-Tap Fractional Sample Interpolation Filters, Position-Dependent Prediction Combination (PDPC), Multiple Reference Lines (MRL), Matrix-Based Intra-Picture Prediction (MIP), Intra Sub-Partition (ISP) Mode, Cross-Component Linear Model (CCLM), and Extended Most Probable Mode (MPM) Signaling.

The Intra-mode decision process selects the most suitable Intra prediction mode for each block by evaluating multiple candidate modes using the Rate-Distortion Optimization (RDO) process [27].

The focus of this work is on the extended Angular Intra-Prediction (AIP). AIP was also presented in HEVC, but VVC increased the number of supported angular modes. While HEVC supports 33 angular modes, VVC extends this number to 67, allowing for greater precision in the prediction process [7].

Angular Intra-Prediction (AIP) uses the previously encoded neighboring samples to generate a prediction for the current block. In many angular modes, an interpolation process is employed to determine the predicted sample values. Figure 2 illustrates the VVC AIP modes, where the solid black lines represent the modes already present in HEVC Intra-Frame prediction, while the dotted blue lines indicate the new modes introduced in VVC. Although Planar and DC modes are non-angular prediction modes, they are also included in the VVC AIP tool.

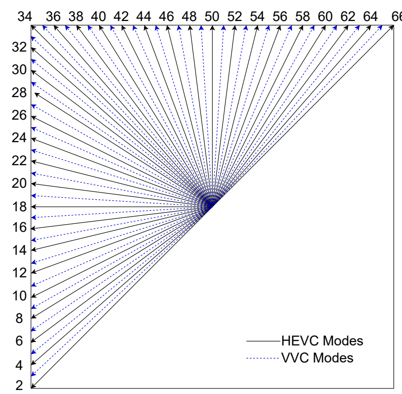

Figure 2: VVC angular Intra-Frame prediction modes [10].

The encoding process in the VVC reference software, known as the VVC Test Model (VTM) [18], employs several heuristics to reduce computational effort, and only a few modes are fully encoded to select the one with the best coding efficiency. These modes are included in a list called the Rate-Distortion list (RD-list) [23]. The first heuristic is called Rough Mode Decision (RMD) [28]. RMD performs a local evaluation to estimate the encoding cost of each candidate mode and inserts a few modes with the lowest costs into the RD-list. The second heuristic is the Most Probable Mode (MPM) [13]. MPM evaluates the most frequently used modes and the modes used to encode the left and above neighbor blocks, to insert at most two additional modes into the RD-list.

The VTM also divides AIP into two steps to avoid an exhaustive evaluation of the 67 Intra-Frame prediction modes. Firstly, RMD evaluates Planar, DC, and 33 angular modes inherited from HEVC (solid black lines in Figure 2) and inserts the best modes into the RD-list. Then, the angular modes adjacent to the angular modes already included in the RD-list are also included in the RD-list. Then, VTM implementation fully evaluates only a few of the available AIP modes.

3 DT BASED VVC INTRA-FRAME ACCELERATION FOR 360° VIDEOS

This work proposes a solution to accelerate the VVC encoder when processing 360° videos by using a Decision Tree (DT) model to selectively skip the AIP Vertical modes.

The entire process of dataset organization, as well as the training and testing of the Decision Trees, was carried out in Python using the *Scikit-Learn* library on the Google Colab platform.

Figure 3 illustrates the implementation of the model, highlighting in blue the decision process of whether to execute or skip the Vertical AIP Modes.

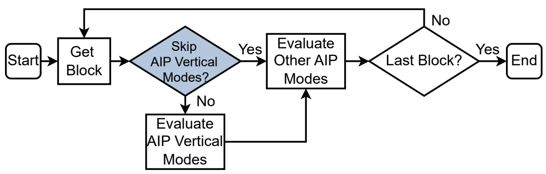

Figure 3: Proposed method flowchart.

The first step in developing the DT was to define which features were relevant for training the models, followed by the feature extraction process itself. To extract meaningful features from the encoding stages targeted for optimization, the VTM encoder was executed with minor modifications, incorporating only routines for feature extraction. This work used the VTM version 19 [18].

The training process used 12 360° video sequences to extract the features: *AcademicBuilding*, *AdministrationBuilding*, *EastGate*, *Library*, *Runners*, *SouthGate*, *SiyuanGate*, *StudyRoom*, *Sward*, *kaixuanmenye_pano*, *lanqiuchang_pano*, and *shuangziqiao_pano*. These sequences are available in the SJTU dataset [19].

VTM supports different configurations, and the configuration used in this experiment was the All-Intra, which is the configuration recommended for experiments targeting the Intra-Frame Prediction. In this case, all frames of a video are encoded using Intra-Frame Prediction.

The first 16 frames of each sequence were encoded four times, once for each Quantization Parameter (QP): 22, 27, 32, and 37. QP is a parameter used to define the level of compression rate, and the higher the QP, the higher the compression rate, but also the higher the degradation in image quality. These QP values are recommended by the Common Test Conditions (CTCs) document for 360° videos [8]. This document defines how the experiments must be conducted when the VVC is evaluated.

A brief description of the extracted features is provided in Table 1. The first 10 features in Table 1 are already available as internal variables within the VTM and bring information about the encoding process that is relevant to the AIP modes definition. The last seven features required additional calculations to be generated, but these calculations are simple, inserting a small overhead in computational effort. Considering the limited space, these features will not be detailed in this paper.

After feature extraction, two datasets were created for each block size: one for training the DTs and another for testing them. To ensure an unbiased training process, both datasets were balanced by limiting their sizes to 100,000 rows for each dataset.

These two datasets were initially used in a Python script to search for the best hyperparameters for each Decision Tree. The script employs Random Search [4] to explore different hyperparameter configurations using the training and testing datasets, aiming to identify the combination that yields the best results according to the F1-Score metric [14] over multiple iterations. The F1-Score is a key metric for evaluating Machine Learning models, as it combines both Precision and Recall into a single value. A high Precision indicates a low rate of false positives — in this context, cases where AIP is incorrectly skipped but should have been executed. Such false positives may lead to losses in coding efficiency, even if they contribute to encoding time reduction. On the other hand, a high Recall indicates a low rate of false negatives — cases where AIP is executed but should have been skipped. These false negatives reduce potential encoding time savings without any improvement in coding efficiency.

Table 1: Description of the features extracted for 360° Intra-Frame.

Feature	Quantity
Quantization Parameter	1
Partitioning depth	1
QT partitioning depth	1
MT partitioning depth	1
Weighted-to-Spherically Peak Signal-to-Noise Ratio (WSPSNR)	1
Ratio Distortion (RD) cost	1
Angular mode SAD	1
Angular mode SATD	1
MIP mode SAD	1
MIP mode SATD	1
Sum of the pixels in the block	1
Average of the pixel values in the block	1
Variance of the pixel values in the block	1
Horizontal and vertical gradients of the block	2
Ratio between the gradients of the block	1
Ratio between the pixel gradients	1
Standard deviation of the pixels in the block	1

Table 2: Search space for each hyperparameter in the Random Search for Decision Trees.

Hyperparameter	Search Space
Criterion	[gini, entropy]
Min Samples Split	[2] ⋃ [25, 501, step 25]
Min Samples Leaf	[1] ⋃ [20, 101, step 20]
Max Features	[1] ⋃ [5, 26, step 10]
Max Depth	[1] ⋃ [10, 101, step 5]
Max Leaf Nodes	[1] ⋃ [20, 701, step 20]

Table 2 presents the evaluated hyperparameters along with the corresponding search space for each one. Each search space was defined using relatively large step sizes and bounded ranges in order to reduce computational load and memory usage, thereby preventing crashes caused by limited RAM on the server where the search was executed.

Using the search spaces defined in Table 2, the script executed the hyperparameter search over 300 iterations to identify the best values for each parameter. The resulting hyperparameter configurations for each of the Decision Trees are presented in Table 3.

The selected hyperparameters were then used to generate the corresponding Decision Trees using the training datasets.

Table 3: Calculated hyperparameters for each Decision Tree.

Hyperparameter	DT Skip Modes
Criterion	gini
Min Samples Split	425
Min Samples Leaf	20
Max Features	25
Max Depth	75
Max Leaf Nodes	700
F1-Score	0.64

Table 4: Training sets and test results for each Decision Tree.

Evaluation Metric	Trained DT Results
Accuracy	0.63
Precision	0.63
Recall	0.64
F1-Score	0.64

Table 4 presents the results for general Machine Learning metrics: Accuracy, Precision, Recall, and F1-Score. These results correspond to the performance of each trained Decision Tree model when evaluated on the testing datasets. The Precision, Recall, and F1-Score metrics were previously discussed. Accuracy, the simplest of these metrics, indicates the proportion of correct predictions among all predictions made.

Analyzing Table 4, one can observe that the trained DT achieved a good performance across the evaluated Machine Learning metrics. The training results were considered satisfactory, and the model was subsequently implemented within the VTM to assess its behavior in a real encoding scenario. To enable this integration, the trained DT was converted to C++ using the Python *m2cgen* library [2], as the VTM software is developed in C++.

4 RESULTS AND DISCUSSION

The trained DT was inserted inside the VTM 19.0 to be evaluated. This version is the last VTM version that supports efficient encoding of 360° videos. The experiments to evaluate the DT behavior followed the CTCs [8] specifications, now considering the sequences defined in the CTCs document. The All-Intra configuration was used, and four QP values (22, 27, 32, and 37) were considered to encode the first 80 frames from eight different video sequences. It is important to highlight that the sequences used in this experiment are different from those used during the training and test processes described in the previous section.

Coding efficiency losses were evaluated using the Bjørntegaard Delta-Bitrate (BD-BR) metric [5]. BD-BR is a widely used metric in the video coding community, and it indicates, for the same objective visual quality, the percentage of increase or decrease in bit-rate caused by a modification over an anchor encoder. The gain in encoding time was quantified using the Time Reduction (TR) metric, which reflects the overall time savings achieved by comparing the modified VTM 19.0 with Decision Trees to the original VTM 19.0.

All experiments were conducted on a server with two Intel Xeon CPU E5-2640 v3 @ 2.60 GHz, 16 cores each and 96 GB of DDR4 RAM. All videos were encoded using a single processor core to ensure that no sequence interfered with the encoding of another.

Table 5: Time reduction and coding efficiency results with the VTM modified to selective skip vertical modes.

Sequence	TR	BD-BR
ChairliftRide	09.20%	0.62%
Gaslamp	09.34%	1.25%
Harbor	09.94%	1.08%
KiteFlite	11.66%	0.60%
SkateboardInLot	09.80%	0.75%
SkateboardTrick	08.45%	0.34%
Train	12.24%	0.32%
Trolley	10.27%	0.34%
Overall Average	**10.11%**	**0.66%**

The results, covering all the video sequences used and an overall average, are presented in Table 5.

Analyzing Table 5, one can observe that the VTM accelerated with Decision Trees for Vertical Modes achieved an average reduction of 10.11% in the total VVC encoding time. Furthermore, a slight decrease in compression efficiency was observed, with an average increase of 0.34% in BD-BR. Additionally, the impact of the trained DT executions on the fast VTM implementation was minimal, resulting in a total encoding time overhead of only 0.26%.

The comparisons with other related works are not easy. As previously discussed, most of the related works focused on the partitioning process [16], [30], [12], [22], [31], and [17], without relation with the AIP tool, focused in this work. Then, comparisons with these works are not relevant, since they are focusing on other encoding tools.

The only work comparable to our work is a previous work of our group [21], which used a heuristic to reduce the AIP computational effort when encoding 360° videos. That previous work reached a time reduction of 8.27% with a BD-BR loss of 0.34%. The solution presented in this paper reached a higher time reduction than that reached in [21], but also with a higher coding efficiency loss.

5 CONCLUSIONS

This work presented a hardware-friendly Machine Learning solution to reduce the computational effort of the Intra-Frame Prediction step in VVC when encoding 360° videos.

The solution employs a Decision Tree to selectively skip the vertical modes inside the Intra-Frame Prediction, achieving an average reduction of 10.11% in the total VVC encoding time, with an average coding efficiency loss of only 0.66% in terms of BD-BR.

This proposed solution achieved competitive results when compared to other works in the literature, offering a good trade-off between processing time reduction and coding efficiency loss. To the best of the authors' knowledge, this is the first paper in the literature proposing a ML based solution to reduce the vertical modes computational effort on the VVC encoder for 360° videos.

ACKNOWLEDGMENTS

The authors of this work would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) Finance Code 001, CNPq, and FAPERGS for funding this research.

REFERENCES

[1] Larissa Araújo, Adson Duarte, Bruno Zatt, Guilherme Correa, and Daniel Palomino. 2024. Fast ISP Mode Decision for the Versatile Video Coding Intra Prediction Using Machine Learning. In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web* (Juiz de Fora/MG). SBC, Porto Alegre, RS, Brasil, 162–170. https://doi.org/10.5753/webmedia.2024.241692

[2] BayesWitnesses. 2022. m2cgen: Model 2 Code Generator. https://pypi.org/project/m2cgen/. Acessado em: 11 ago. 2025.

[3] Bernardo Beling, Iago Storch, Luciano Agostini, Bruno Zatt, Sergio Bampi, and Daniel Palomino. 2020. ERP-Based CTU Splitting Early Termination for Intra Prediction of 360 videos. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. 359–362. https://doi.org/10.1109/VCIP49819.2020.9301879

[4] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research* 13 (2012), 281–305.

[5] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. Technical Report VCEG-M33, ITU-TSG16/Q6, Austin, Texas, USA.

[6] Vinicius Borges, Murilo Perleberg, Marcelo Porto, and Luciano Agostini. 2024. High-Throughput Hardware Design for the Complete VVC Angular Intra Prediction. In *2024 31st IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. 1–5. https://doi.org/10.1109/ICECS61496.2024.10848892

[7] Vinicius Borges, Murilo Perleberg, Marcelo Porto, and Luciano Agostini. 2025. Hardware Design for VVC Angular Intra Prediction Modes with Coding Efficiency Awareness. In *2025 IEEE 16th Latin America Symposium on Circuits and Systems (LASCAS)*, Vol. 1. 1–5. https://doi.org/10.1109/LASCAS64004.2025.10966271

[8] Jill Boyce, Elena Alshina, Adeel Abbas, and Yan Ye. 2018. *JVET common test conditions and evaluation procedures for 360° video.* JVET output document JVET-J1012. Joint Video Exploration Team (JVET), ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Chengdu, China. https://www.researchgate.net/publication/326506583_JVET-J1012_JVET_common_test_conditions_and_evaluation_procedures_for_360_video Presented at the 4th JVET meeting, Chengdu, China, October 15–21, 2016.

[9] 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. https://doi.org/10.1109/TCSVT.2021.3101953

[10] Adrian Browne, Yan Ye, and Seung Hwan Kim. 2022. Algorithm description for Versatile Video Coding and Test Model 19 (VTM 19). *Jt. Video Expert. Team ITU-T SG 16 WP 3 ISO/IEC JTC 1/SC 29, 26th Meet. by teleconference* (April 2022).

[11] Bolin Chen, Zhao Wang, Binzhe Li, Shiqi Wang, and Yan Ye. 2023. Compact Temporal Trajectory Representation for Talking Face Video Compression. *IEEE Transactions on Circuits and Systems for Video Technology* (2023), 1–14. https://doi.org/10.1109/TCSVT.2023.3271130

[12] Yan Chen, Tiansong Li, Haokun Liu, Shaoguo Cui, Li Yu, Kejun Wu, and Hongkui Wang. 2024. A fast intra CU partition algorithm in Versatile Video Coding for 360-degree video. *Journal of Visual Communication and Image Representation* 104 (2024), 104305. https://doi.org/10.1016/j.jvcir.2024.104305

[13] Yamei Chen, Li Yu, Hongkui Wang, Tiansong Li, and Shengwei Wang. 2020. A novel fast intra mode decision for versatile video coding. *Journal of Visual Communication and Image Representation* 71 (2020), 102849. https://doi.org/10.1016/j.jvcir.2020.102849

[14] Ramatoulaye Diallo, Codjo Edalo, and O Olawale Awe. 2024. Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score. In *Practical Statistical Learning and Data Science Methods: Case Studies from LISA 2020 Global Network, USA*. Springer, 283–312.

[15] Adson Duarte, Bruno Zatt, Guilherme Correa, and Daniel Palomino. 2023. Fast Intra Mode Decision Using Machine Learning for the Versatile Video Coding Standard. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1–5. https://doi.org/10.1109/ISCAS46773.2023.10181769

[16] Jose N. Filipe, J. Carreira, Luis M. N. Tavora, Sergio M. M. de Faria, Antonio Navarro, and Pedro A. A. Assuncao. 2021. Tree-Based Ensemble Methods for Complexity Reduction of VVC Intra Coding. In *2021 Telecoms Conference (ConfTELE)*. 1–6. https://doi.org/10.1109/ConfTELE50222.2021.9435476

[17] Jose N. Filipe, Luis M. N. Tavora, Sergio M. M. Faria, Antonio Navarro, and Pedro A. A. Assuncao. 2025. Linear Multivariate Decision Trees for Fast QTMT Partitioning in VVC. *IEEE Open Journal of Signal Processing* 6 (2025), 175–183. https://doi.org/10.1109/OJSP.2025.3528897

[18] Fraunhofer Heinrich Hertz Institute (HHI) Joint Video Experts Team (JVET). 2022. VTM reference software for VVC version 19.0. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/releases/VTM-19.0.

[19] Xu Liu, Yongcheng Huang, Li Song, Rong Xie, and Xiaokang Yang. 2017. The SJTU UHD 360-Degree Immersive Video Sequence Dataset. In *2017 International Conference on Virtual Reality and Visualization (ICVRV)*. IEEE, 400–401. https://doi.org/10.1109/ICVRV.2017.00095

[20] Samir Marwaha. 2024. *Sandvine's 2024 Global Internet Phenomena Report: Global Internet Usage Continues to Grow.* https://www.applogicnetworks.com/blog/sandvines-2024-global-internet-phenomena-report-global-internet-usage-continues-to-grow

[21] Rogério Rosado, Otávio Santos, Franklin de Oliveira, Lucas Silva, Vanessa Aldrighi, Iago Storch, Gustavo Sanchez, Daniel Palomino, and Luciano Agostini. 2025. Fast Heuristic for VVC Intra-frame Prediction Targeting 360° Video Formats. In *2025 IEEE 16th Latin America Symposium on Circuits and Systems (LASCAS)*, Vol. 1. 1–5. https://doi.org/10.1109/LASCAS64004.2025.10966322

[22] Rogério Rosado, Otávio Santos, Franklin Oliveira, Lucas Silva, Gustavo Sanchez, Iago Storch, Daniel Palomino, and Luciano Agostini. 2024. A Coding-Efficiency-Aware Fast Heuristic for VVC Intra-Frame Prediction Targeting 360° Videos. In *Proceedings of the 30th Brazilian Symposium on Multimedia and the Web* (Juiz de Fora/MG). SBC, Porto Alegre, RS, Brasil, 355–359. https://doi.org/10.5753/webmedia.2024.243128

[23] Mário Saldanha, Gustavo Sanchez, César Marcon, and Luciano Agostini. 2021. Performance analysis of VVC intra coding. *Journal of Visual Communication and Image Representation* 79 (2021), 103202. https://doi.org/10.1016/j.jvcir.2021.103202

[24] Icaro Siqueira, Guilherme Correa, and Mateus Grellert. 2020. Rate-Distortion and Complexity Comparison of HEVC and VVC Video Encoders. In *2020 IEEE 11th Latin American Symposium on Circuits and Systems (LASCAS)*. 1–4. https://doi.org/10.1109/LASCAS45839.2020.9069036

[25] Iago Storch, Luciano Agostini, Bruno Zatt, Sergio Bampi, and Daniel Palomino. 2022. FastInter360: A Fast Inter Mode Decision for HEVC 360 Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 5 (2022), 3235–3249. https://doi.org/10.1109/TCSVT.2021.3096752

[26] Iago Storch, Luciano Agostini, Bruno Zatt, and Daniel Palomino. 2021. Exploring ERP Distortions to Reduce the Encoding Time of 360 Videos. In *Anais Estendidos da XXXIV Conference on Graphics, Patterns and Images* (Online). SBC, Porto Alegre, RS, Brasil, 139–145. https://doi.org/10.5753/sibgrapi.est.2021.20026

[27] G.J. Sullivan and T. Wiegand. 1998. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine* 15, 6 (1998), 74–90. https://doi.org/10.1109/79.733497

[28] Ei Ei Tun, Supavadee Aramvith, and Takao Onoye. 2022. Low complexity mode selection for H.266/VVC intra coding. *ICT Express* 8, 1 (2022), 83–90. https://doi.org/10.1016/j.icte.2021.08.018

[29] Mai Xu, Chen Li, Shanyi Zhang, and Patrick Le Callet. 2020. State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression. *IEEE Journal of Selected Topics in Signal Processing* 14, 1 (2020), 5–26. https://doi.org/10.1109/JSTSP.2020.2966864

[30] Mengmeng Zhang, Yan Hou, and Zhi Liu. 2023. An early CU partition mode decision algorithm in VVC based on variogram for virtual reality 360 degree videos. *EURASIP Journal on Image and Video Processing* 2023, 1 (2023), 9.

[31] Wenjun Zheng, Chao Yang, Ping An, Xinpeng Huang, and Liquan Shen. 2024. Learning-based CU partition prediction for fast panoramic video intra coding. *Expert Systems with Applications* 258 (2024), 125187. https://doi.org/10.1016/j.eswa.2024.125187