Exploitation of Approximation Storage at VVC Inter-Frame Prediction Memories

Yasmin Camargo, Daniel Palomino, Bruno Zatt
Video Technology Group (ViTech) - PPGC - CDTec - Universidade Federal de Pelotas (UFPel)
Pelotas, Brasil
{yasmin.sc, dpalomino, zatt}@inf.ufpel.edu.br

Felipe Sampaio
Grupo de Tecnologias em Sistema de Computação (SisTech) Federal Institute of Rio Grande do Sul (IFRS)
Campus Farroupilha, Brasil
felipe.sampaio@farroupilha.ifrs.edu.br

***Abstract*— This paper presents an error tolerance (aka. resilicent) profiling, focusing on the inter-frame prediction at Versatile Video Coding (VVC) encoders, for approximate storage exploitation. Although next-generation VVC encoders have great video compression, they are still concerned about their high energy consumption. This problem occurs with greater impact at specific moments of coding, mainly in the inter-frame prediction stage, which demands greater complexity. In this context, the goal of this work is to investigate the possibility of using approximate storage in four specific memory regions during the inter-frame prediction step using VVC: Reconstructed, Original, Filtered and Prediction Samples Buffer The study evaluated the resulting losses in terms of coding efficiency by inserting errors in four video sequences and four inter-frame prediction memories. The results indicate that an increase in the error rate results in higher BD-Rate values. In general, the results indicate a great potential for exploring this technique, since the increase in BD-Rate did not exceed 0.6% and the technique proved to be more efficient on average was the Prediction Samples Buffer memory (maximum BD-Rate of 0.14%), followed by the Reconstructed Samples Buffer (0.39%), Original Samples Buffer (maximum BD-Rate of 0.41%) and Filtered Samples Buffer (0.51%).**

Keywords — Video Encoding; Versatile Video Coding (VVC); Approximate Storage.

I. Introduction

The flow of digital videos on the internet is expanding more and more. In 2022, for example, video usage grew by 24% compared to 2021, equivalent to 65.93% of all internet traffic [1]. For this type of media to be efficiently transmitted and stored, there are tools known as video compressors (encoders/codecs), which are responsible for significantly reducing the size of a video representation. According to The 6th Annual Bitmovin Video Developer Report [2] the most consolidated encoders currently are H.264/AVC and HEVC reaching a large share of the market.

However, some standards were recently released,gaining more and more prominence, as it is the case of Versatile Video Coding (H.266/VVC), released in July 2020, which is the current state of the art in video compression. This trend can be seen in the data presented in Fig. 1 [2], in which VVC is already the third most used codec, with an adoption perspective of 29% still in 2023.

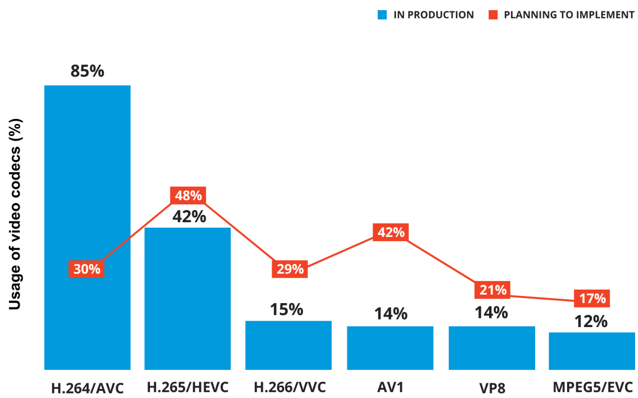

Fig. 1. Current use of video codecs in production and implementation planning for the next 12-24 months

VVC stands out for its coding efficiency, as it offers a 50% saving in bit compression compared to its predecessor, High-Efficiency Video Coding (HEVC) while maintaining the same visual quality [3]. However, it is necessary to point out that this codec requires a significantly higher computational cost since it improves and brings a series of new coding tools. Among the encoder steps, it is possible to emphasize the inter-frame prediction, more specifically the motion estimation module, which represents up to 57% of the total VVC coding time, according to the experiments carried out in [4].

As a consequence of its high computational complexity, applications that implement the VVC codec suffer with a high energy consumption. This energy consumption represents a key challenge when mobile devices are the targets, as they have limited batteries and they are not connected to a power source all the time. Allied with this, it is expected that, by 2023, approximately 70% of the global population will have mobile devices with connectivity [5] and 67.60% of the traffic on these devices is destined for videos [2]. Thus, studies are needed to increase the energy efficiency of the VVC video encoder.

With this in mind, *approximate storage* techniques can be explored to reduce the energy consumption in VVC, when looking at the perspective of data memory infrastructure. This strategy can lead to significant energy savings [6], especially for more computationally intensive applications, such as video encoding. This concept is based on the fact that some data can be removed (or changed) without the human being noticing. When approximate storage is applied, this reduction might be achieved by several techniques. For instance, some strategies downscale the memory supply voltage, which leads to energy savings in transmission and storage, but incurring on lossy memory operations. As a side effect, there is a degradation in the final quality of the video. The research challenge that arises is related to performe proper evaluations with regardin to the fault tolerance profile (resilience), focusing on analyzing the resulting losses in terms of coding efficiency.

The tolerance of approximations in data storaging and data computation (*aka.* approximate computing) have been exploited in recent years by the video coding and circuits and systems design research community. At the perspective of approximte computing, several works have analysed the impact of error occurrence during the computation of different modules within a video codecs [7][8]. Further, the approximate computing concept was directly applied to modify video coding algorithms or design imprecise mathematical hardware operators to increase the hardware energy efficiency in exchange of minimal loss at the compression efficiency [9][10][11]. Concerning the approximate storage exploitation at resilient data, error resilience profiles were built and extensively analyzed for inter-frame [12] and intra-frame [13] predictions for the VVC decoders. Such works endorse the relevance of approximate storage exploitation to increase the energy efficiency of state-of-the-art VVC applications. However, the cited studies focused on the decoder side, which have considerably less computational complexity when compared to the encoder part, which reunites all the high-complexity decision-based coding tools inserted at VVC codecs.

Therefore, the goal of this work is to evaluate the levels of resilience in VVC video encoders, considering scenarios where approximate storage techniques are explored in specific memory regions of the inter-frame prediction module. It is intended, in this way, to trace resilience profiles and define adequate levels of loss of coding efficiency, in exchange for the reduction of energy consumption.

II. Adopted Memory Model for VVC Inter-Frame Prediction

As seen before, the inter-frame prediction presents one of the greatest complexities of the VVC video encoder, since a huge number of operations, tests, calculations, and memory accesses are performed in this module. It is important to emphasize that, in this step, the encoder is trying to predict the motion behavior between the frames. The goal is to identify the temporal redundancy, because the sequence of images that make up a video tends to be very similar for frames close to each other. To perform this prediction, the VVC inter-frame prediction adopts three motion estimation steps: Integer Motion Estimation (IME), Fractional Motion Estimation (FME), and Affine Motion Estimation (AME). This operation can be seen in Fig. 2, highlighting each of the memories that are most accessed (represented in blue).

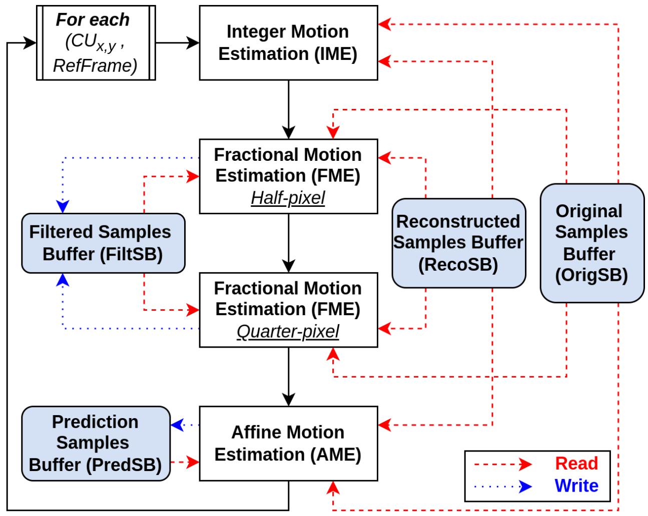

Fig. 2. Adopted VVC inter-frame prediction memory model.

The original frame that is being currently encoded is stored in the Original Samples Buffer (OrigSB). This table is used as a basis to try to find the best prediction [14], so this memory will be accessed by all stages of the inter-prediction since it is always necessary to compare candidate blocks searched by any ME modules, aiming to find the most similar one.

In the memory called Reconstructed Samples Buffer (RecoSB), the temporally past reference frames and the temporally future reference frames (from L0 and L1 reference frame lists) are stored [14]. These frames have already been previously encoded, reconstructed and selected by the encoder. Therefore, they must be stored so that in the next steps it is possible to find the motion, that is, they are the frames that are potentially closest to the originals.

The IME step, for example, highly accesses this memory since it will try to find the candidate block that presents the greatest similarity. As the block is a portion of the frame it is necessary to perform tests for several different partition sizes with all the reference frames that are stored in the RecoSB memory.

As the next step, the FME aims to deepen this search even further, since the smaller the region, the more accurate the prediction will be. Therefore, it is necessary to use fractional pixels, since the motion can be very subtle and does not vary more from one pixel to another, but from half or quarter of a pixel. As these samples do not exist, they need to be generated from VVC-defined interpolation filters [15]. Thus, it is necessary to store the data that are being generated in the Filtered Samples Buffer (FiltSB) memory.

The last step is the AME, which is one of the novelties that was implemented in VVC, being responsible for predicting non-translational movements, such as rotation and zoom. This prediction has two variations that are defined according to the number of parameters established during coding. The first variation is the transformation block to be generated by two vectors (4 parameters) or the case to be generated by three vectors (6 parameters) [16]. Therefore, this step reunites complex operations and, during such calculations and tests performed in this process, it is necessary to store some local information, so it accesses the Prediction Samples Buffer (PredSB) memory.

This work evaluates the VVC encoder resilience behavior when errors occurred during the inter-frame prediction step, according to the approximate storage model depicted in Fig. 2, where the dashed lines represent the approximated read/write memory operations where errors are injected. The adopted error insertion model at inter-frame prediction leads to non-optimal choices to IME, FME and AME modules. As the frame reconstruction path is not affected by the inserted approximations, there is no involved drift between the encoding and decoding parts.

III. Experimental Setup

As mentioned, approximate storage allows you to reduce power and increase memory performance. In this sense, this work intends to analyze how video coding behaves when error are assumed during read and write memory operations. Therefore, the memory resilience for the inter-frame prediction module will be verified, more specifically in the Reconstructed Samples Buffer (RecoSB), Original Samples Buffer (OrigSB), Filtered Samples Buffer (FiltSB), and Prediction Sample Buffer (PredSB) memories.

Therefore, at such inter-frame prediction steps of encoding, approximate storage techniques will be applied and their effects analyzed. It is noteworthy that to insert the different error rates, both in reading and writing, a PinTool tool was developed to be used with Intel Pin instrumentation tool, [17]. The following error rates were analyzed: 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, and 0 (no loss). An error rate of 10^{-7}, for example, means that there is a probability of an error occurring for every 10 million write or read operations, while an error rate of 10^{-3} means that there is a probability of an error occurring for every thousand memory operations.

To carry out the experiments, the VVenC software was used, which is an implementation based on the VVC standard, the VTM, with a series of optimizations so that its execution time is smaller. The justification for choosing VVenC is because the execution of experiments with VTM ends up becoming unfeasible, since it has a very high simulation time, while VVenC is up to 30x faster for a cost of 12 % increase in bit rate [18]. In addition, the use of the PinTool tool considerably increases the execution time, which ends up aggravating this problem even more. VVenC works on one among five available profiles for coding, in which one of which must be chosen: faster, fast, medium, slow, and slower. For this work, medium profile was chosen, as it strikes a balance between time and coding efficiency, enabling a large part of VVC tools.

VVC encoder executions were performed for the first 17 frames in four video sequences that make up class C (832x480 pixels) of the videos defined as a test for the VVC, which are: BasketballDrill, BQMall, RaceHorses, and PartyScene [19]. Ten repetitions of each experiment were performed, with the goal of having more accurate results since the average of these values will be used to evaluate them. The Quantization Parameter was also defined following the evaluation recommendations defined by the VVC, four values of QPs were used: 22, 27, 32, and 37 [19].

After encoding, log files are generated that contain various information with encoding data, representing a total of 3.840 files (variations for 4 videos, 4 QPs, 4 memories, 6 error rates, and 10 repetitions). Thus, to facilitate the analysis of the data obtained, python scripts were developed to extract the necessary information, such as exporting the data to a CSV file, calculating the average of repetitions, and generating graphs comparing the results obtained. Furthermore, to verify the results, the calculation of the BD-Rate [20] is performed, which is an objective metric to represent the rate-distortion efficiency losses achieved in the executed experiments.

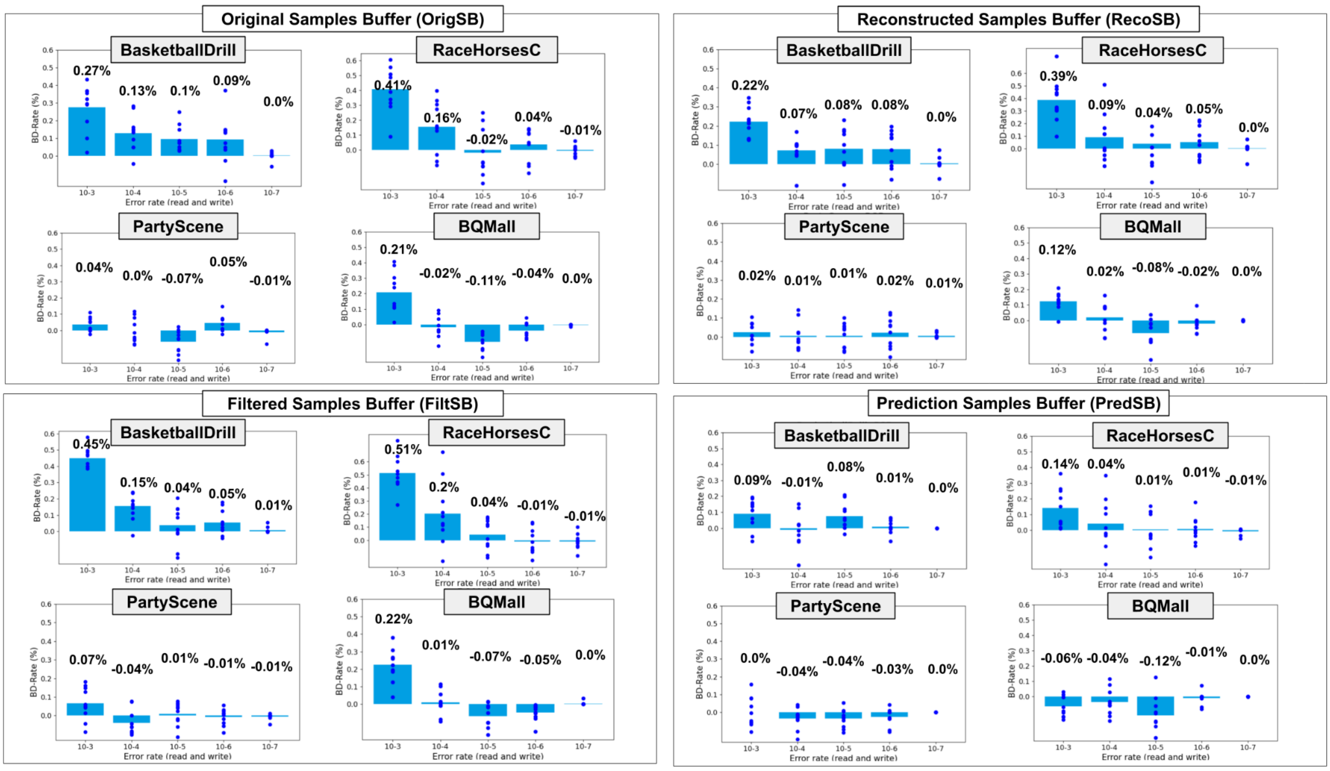

Fig. 3. BD-Rate (%) results consideing error injection at each inter-frame prediction memory for each evaluated test sequence.

IV. Experimental Results

The approximate storage exploitation results for OrigSB, RecoSB, FiltSB and PredSB inter-frame prediction related memories are shown in Fig. 3. Besides, the charts of Fig.3 also compares the behavior for the four coded video sequences (BasketballDrill, BQMall, RaceHorses, and PartyScene). In each of them, a correspondence was established between the adopted error rate (varying between 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, and 10^{-3} for both memory reading and writing operations) and the average BD-rate obtained for all experiment repetitions (depicted by the blue bar). In addition, the individual results of each experiment repetition are presented by the dark blue dots in Fig. 3.

In general, the results show that the higher the adopted error rate, the higher the BD-Rate values, which represent more expressive losses in coding efficiency (ratio between coded video quality and achieved compression rate). As this work focused on class C videos (832x480 pixels), we can note quite lower coding efficiency losses (BD-Rate results), even when 10^{-3} is set as the evaluated error rate. It is important to emphasize that, in this work, the evaluation was performed individually for each selected memory. In a complete inter-frame prediction resilience evaluation for approximate storage exploitation, the coding efficiency losses for each memory part shall be combined. Further, higher BD-Rate results might occur for higher-resolution videos, such as HD1080p and 4K.

Regarding the memory modules, the PredSB memory had a lower BD-Rate rate compared to the other modules, with an average BD-Rate of 0.04% for the 10^{-3} error rate. This suggests that this memory is more resilient to errors, that is, the impact of applying approximate storage on this memory is smaller. As mentioned earlier, this is the local memory of the affine prediction, being a specific module for predicting non-translational movements. Our insight is that the AME prediction ends up being less used compared to IME and FME, resulting in less errors that are propagated alon the next VVC encoder modules. For the other modules, the average losses for an error rate of 10^{-3} reach 0.19%, 0.23%, and 0.31% for the RecoSB, OrigSB and FiltSB modules, respectively. Thus, in terms of error tolerance to approximate storage exploitation, for the tested scenarios, the FiltSB memory, used to store the FME interpolated samples, represents the least resilient memory, followed by the OrigSB (current original frame storage) and, then, by RecoSB (which keeps the reconstructed reference frame samples).

When examining each video individually, it is possible to notice that the results obtained vary from one video to another, with some being more sensitive to certain encoding parameters than others. This difference becomes evident when we compare the PartyScene video, which presented an average BD-Rate rate of 0.03% for an error rate of 10^{-3}, to the RaceHorses and BasketballDrill sequences, which present rates of 0.36% and 0.26%, respectively. Such variation is considered normal, since each video has its own characteristics, such as different textures and motion properties, which lead to different coding stages. It is assumed, for example, that the PartyScene video (which presented the lowest BD-Rate rates) presents scenes with less movement and therefore can be coded mostly with intra-frame prediction, while the others may have been coded predominantly with inter-frame prediction, whereas the errors are being inserted, justifying the higher rates.

Although some unexpected behaviors have been verified, as in the case of the RecoSB memory for the RaceHorses video, in which the BD-Rate decreased from 0.05% to 0.04% (between error rates of 10^{-6} and 10^{-5}) instead of increasing, or, for example, in the OrigSB memory for the PartyScene video between rates 10^{-6} and 10^{-5} in which the reduction was from 0.05% to -0.07%. Such results can be justified by the small difference in the BD-Rate values between the error rates, which did not exceed 0.6 between the highest and lowest value found. It is also observed that there is a significant variation in the values obtained from the BD-Rate for each repetition of the coding (data dispersion), demonstrating that the randomness of the inserted errors is quite varied. Therefore, unexpected results can be caused by limitations in the number of simulation repetitions, given the extremely high simulation time required. We plan to extend the number of experiments repetitions to have more solid results.

V. Conclusions

This work carried out an initial evaluation of the resilience profiles of inter-frame prediction module at VVC video encoder with a focus on the use of approximate storage techniques. Error tolerance levels were surveyed, in terms of coding efficiency losses (BD-Rate metric) considering the occurrence of errors in reading and writing operations in four specific memory regions: Reconstructed Samples Buffer, Filtered Samples Buffer, Original Samples Buffer and Prediction Samples Buffer. As preliminary results, there was a general trend towards an increase in coding efficiency losses with the increase in applied error rates. The memory with greater resilience to applied errors was PredSB, then OrigSB, RecoSB and FfiltSB. In addition, it was found that the behavior of videos for inserting errors is quite varied, depending on their individual characteristics. The performed analysis allowed identifying the possibility of using errors during the coding process, since the values found were significantly low, not exceeding a 0.6% increase in the BD-Rate, which could lead to a reduction in energy consumption. However, it is worth noting that this work is at an early stage and analyzes of actual impacts on energy savings have not yet been carried out. A study on the magnitude of this gain is planned for future work. Furthermore, it is planned to expand the simulations to analyze unexpected behaviors in the profiles traced in this study, with a focus on increasing the number of repetitions performed to obtain more accurate results. In addition, deeper analyzes will be carried out on each of the memories and encoded videos to obtain more detailed resilience profiles, which will allow the effective use of approximate storage techniques and the evaluation of the impact of these techniques in reducing the energy consumption of the codecs VVC.

References

[1] Sandvine, "Global Internet Phenomena Report," Waterloo, ON, Canada, Sandvine Incorporated, 2023. [Online]. Available:https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2023/reports/Sandvine%20GIPR%202023.pdf

[2] Bitmovin. "Video Developer Report 2021", Bitmovin Inc., 2021. [Online]. Available: https://bitmovin.com/video-developer-report/?utm_source=hellobar&utm_medium=banner&utm_campaign=report&utm_content=Video_Developer_Report.

[3] Fraunhofer Institute. "Fraunhofer HHI is proud to present the new state-of-the-art in global video coding: H.266/VVC brings video transmission to new spee.", Fraunhofer Institute, 2021. [Online]. Available: https://newsletter.fraunhofer.de/-viewonline2/17386/465/11/14SHcBTt/V44RELLZBp/1

[4] F. Pakdaman, M. A. Adelimanesh, M. Gabbouj and M. R. Hashemi, "Complexity Analysis Of Next-Generation VVC Encoding And Decoding," 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 3134-3138, doi: 10.1109/ICIP40778.2020.9190983.

[5] Cisco. "Cisco Annual Internet Report," Cisco Systems Inc., 2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[6] T. G. Viegas."Técnicas de Computação Aproximada para Implementação de Filtros FIR", 2016. [Online]. Available: https://fenix.tecnico.ulisboa.pt/downloadFile/1407770020544962/Thesis.pdf

[7] Palumbo, Francesca, and Carlo Sau. "Reconfigurable and approximate computing for video coding." arXiv preprint arXiv:2103.03712 (2021).

[8] W. El-Harouni, S. Rehman, B. S. Prabakaran, A. Kumar, R. Hafiz and M. Shafique, "Embracing approximate computing for energy-efficient motion estimation in high efficiency video coding," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, Switzerland, 2017, pp. 1384-1389, doi: 10.23919/DATE.2017.7927209.

[9] R. Domanski, W. Kolodziejski, G. Correa, M. Porto, B. Zatt and L. Agostini, "Low-Power and High-Throughput Approximated Architecture for AV1 FME Interpolation," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401224.

[10] Silva, Giovane Gomes, et al. "Approximate Hardware Architecture for Interpolation Filter of Versatile Video Coding." Journal of Integrated Circuits and Systems 16.2 (2021): 1-8.

[11] H. Mahdavi, H. Azgin and I. Hamzaoglu, "Approximate Versatile Video Coding Fractional Interpolation Filters and Their Hardware Implementations," in IEEE Transactions on Consumer Electronics, vol. 67, no. 3, pp. 186-194, Aug. 2021, doi: 10.1109/TCE.2021.3107460.

[12] R. Soares, et al. “Error Resilience Evaluation of Approximate Storage in the Motion Compensation of VVC Decoders”, IEEE 14th Latin American Symposium on Circuits and Systems (LASCAS 2023), Quito, Ecuador, 2023.

[13] M. Isquierdo, R. Soares, F. Sampaio, B. Zatt and D. Palomino, "Error Resilience Evaluation of Approximate Storage in the Intra Prediction of VVC Decoders," 2022 35th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI), Porto Alegre, Brazil, 2022, pp. 1-6, doi: 10.1109/SBCCI55532.2022.9893263.

[14] Gonçalves, Paulo Henrik Ribeiro. Um esquema rápido baseado em aprendizado de máquina para a predição interquadros do codificador de vídeo VVC. MS thesis. Universidade Federal de Pelotas, 2021.

[15] V. R. Filho, M. Monteiro, I. Seidel, M. Grellert and J. L. Güntzel, "Hardware-Friendly Search Patterns for the Versatile Video Coding Fractional Motion Estimation," 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), Tampere, Finland, 2021, pp. 1-6, doi: 10.1109/MMSP53017.2021.9733603.

[16] S. -H. Park and J. -W. Kang, "Fast Affine Motion Estimation for Versatile Video Coding (VVC) Encoding," in IEEE Access, vol. 7, pp. 158075-158084, 2019, doi: 10.1109/ACCESS.2019.2950388.

[17] O. Levi. "PIN: A Dynamic Binary Instrumentation Tool," Intel Corporation, 2005. [Online]. Available: https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool.

[18] A. Wieckowski et al., "Vvenc: An Open And Optimized Vvc Encoder Implementation," 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shenzhen, China, 2021, pp. 1-2, doi: 10.1109/ICMEW53276.2021.9455944.

[19] Bossen, Frank. "Common test conditions and software reference configurations." JCTVC-L1100 12.7 (2013).

[20] BJONTEGAARD, g. Calculation of average PSNR differences between RD-curves (VCEG-M33). 2001. [Online]. Available: https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc