Assessment of Different Coding Units Usage in VVC Inter-Frame Prediction

Ramiro Viana[1,2], Fernando Sagrilo[3], Marta Loose[1], Gustavo Sanchez[4], Guilherme Corrêa[1], Luciano Agostini[1]
[1]Video Technology Research Group (ViTech), UFPel, Pelotas/RS, Brazil
[2]Electronic Engineering, UFPel, Pelotas/RS, Brazil
[3]Graduate Program in Electrical Engineering (PPGEE), UNIPAMPA, Alegrete/RS, Brazil
[4]Federal Institute of Sertão Pernambucano (IFSertaoPE), Salgueiro/PE, Brazil
{rgsviana, marta.breunig, gcorrea, agostini}@inf.ufpel.edu.br,
fernandosagrilo.aluno@unipampa.edu.br, gustavo.sanchez@ifsertao-pe.edu.br

***Abstract*—With the increasing demand for video transmission through streaming platforms, as well as social media, video encoding has become something that increasingly needs new developments and improvements. This paper presents an investigation related to the different coding unit (CU) sizes that are encoded within the Versatile Video Coding (VVC) inter-frame prediction, specifically in the Unidirectional, Bidirectional and Affine Motion Estimation (ME), when multiple high-resolution videos were encoded and the reached results were evaluated. In this investigation, the number of samples encoded with each CU size, as well as the time necessary to encode each CU size, are shown and discussed, in order to support future developments targeting the efficiency increase or the computational effort decrease in the inter-frame prediction of the VVC standard. Considering the reached results, the main conclusion is that Affine is the ME tool that takes the longest time to be completed, even being used to encode much less samples than Unidirectional and Bidirectional ME together. The reached results also showed that the relative use of the Affine ME inside the VVC inter-frame prediction is inversely proportional to the video resolution. Finally, the relative use of different CU sizes to encode the video samples present a similar behavior for different resolutions and encoding tools.**

***Index Terms*—Video Encoding, Motion Estimation, Block Size, Unidirectional, Bidirectional, Affine.**

I. Introduction

In recent years, the demand for access to digital videos has intensified a lot, mainly due to the constant rise in the use of the Internet [1]. Another factor for this increasing demand was the Covid-19 pandemic which boosted the remote work and study, as well as the various streaming platforms available for entertainment. For these reasons, efficient ways to encode videos are been developed in order to support this increased demand for video transmission [2].

The video encoding process consists of several distinct steps and tools working together to encode each frame of a video to reduce the amount of redundant data in the representation of video information, exploring spatial redundancy, temporal redundancy, and entropy redundancy [3].

The state-of-the-art standard for video encoding is the Versatile Video Coding (VVC) [4]. Despite presenting a lot of required computational effort, VVC provides high compression ratios [5] [6], being superior to other encoders available on the market in terms of compression efficiency [2].

In VVC, the video to be encoded is firstly partitioned into blocks called Coding Tree Units (CTU) [7] and each CTU can have from 32x32 to 128x128 samples. Each CTU is the root of a Quadtree with a nested multi-type tree (QTMT), a complex structure used to further divide the CTU into blocks called Coding Units (CU). The CUs are used to better adapt the coding process to the image features [8]. The CTU can be split into quaternary, ternary, and binary partitions and the VVC allows the use of 28 different CU sizes when encoding each CTU. The CUs are the frame blocks that are, in fact, encoded.

VVC is a hybrid encoder, which means that each CU will be predicted through an inter-frame or an intra-frame prediction, and the prediction residues are then transformed and quantized and, finally, the quantized information is processed by the entropy encoder to finish the coding process [9]. The encoder also has inverse quantization and transforms steps, as well as a filter step and these steps are necessary to guarantee that encoder and decoder will use exactly the same references. Since quantization is a lossy process, then the encoder will not use the original frame as a reference for future encoding, but the reconstructed frame [10].

VVC brings a lot of novelties in each one of these steps when compared with previous standards. The very high computational cost of VVC is due to the high number of CU sizes and encoding tools which are available, since all combinations must be evaluated to define the best one to encode that frame region [10].

This paper is focused on the VVC inter-frame prediction, more specifically on the three most important encoding tools of this step: Unidirectional, Bidirectional, and Affine Motion Estimation. This paper presents an investigation of the different CU sizes use during these inter-frame prediction tools, including the number of samples encoded with each CU size for each one of these three encoding tools and also the time spent to encode each CU size.

II. VVC Inter-Frame Prediction

The inter-frame prediction exploits temporal redundancy in the video sequence by predicting the current frame CUs using other frames previously encoded frames from the video [5]

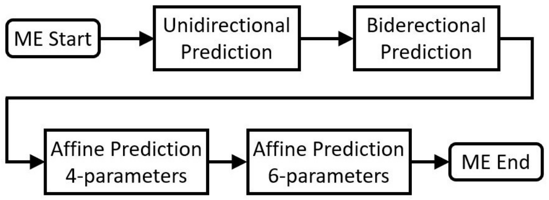
ME Start
Unidirectional Prediction
Biderectional Prediction
Affine Prediction 4-parameters
Affine Prediction 6-parameters
ME End

Fig. 1. VVC ME Flowchart

[11]. The most important step inside the inter-frame prediction is the Motion Estimation (ME), focus of this work. ME tries to find, inside the reference frames, which block is the best one to be used to predict the current CU, called "the best match" [12]. Naturally, this process requires a high level of computational effort. VVC Motion Estimation is composed of three main tools: Unidirectional ME, Bidirectional ME, and Affine ME [13].

The Unidirectional Motion Estimation is the basic ME mode and is present in all previous video encoders with inter-frame prediction. This tool can use one or more reference frames to find the best match, but in a direct way, then, the blocks inside of each reference frame are compared with the current CU and the best one is selected. To reduce the complexity of this process, a search area is defined around the collocated block inside the reference frame (the one in the same position as the current CU) [14]. The VTM implements the Test Zone Search (TZS) ME algorithm [15] in this step. The Bidirectional Motion Estimation can use two blocks in two different frames to generate the prediction. But all other processes are similar to those ones used in the Unidirectional ME. The Bidirectional ME is effective for video sequences where there are fast movements, panning cameras, zooming and scene changes [13].

Both, Unidirectional and Bidirectional ME are focused on translational movements in the objects inside a scene, and all available CU sizes can be used in both ME tools. The Affine Motion Estimation (AME) is one of the main innovations of VVC in relation to other video encoders. It maps the non-translational movements of the objects in scenes, such as scaling, rotation, and zooming, improving the coding efficiency [14].

The VVC AME is applied in two configurations: with 4-parameters or with 6-parameters. The first one is used to model simpler movements, like scaling and rotation. The second one is used to map more complex movements, like shearing [6] [7]. Considering the higher computational effort required in the Affine ME, only CUs equal to or larger than 16 × 16 samples can be used in this ME tool. Fig. 1 shows the sequence of use of these ME tools inside a VVC encoder.

III. Experimental Methodology

The experiments were done using the VVC reference software, called VVC Test Model (VTM) [16], version 14.0. This work used the Random-Access configuration, since the main goal is to evaluate the inter-frame prediction. The Quantization Parameters (QPs) used were 22, 27, 32 and 37, as defined by the Common Test Conditions (CTCs) [17].

Fifteen videos with three different resolutions were used and the first 32 frames of these videos were considered. These videos are form the UVG dataset [18] and the NETVC dataset [19]. The videos were divided in three resolutions: HD (1280×720), Full HD (1920×1080) and 4K Ultra HD (3840×2160). The HD sequences were *Dark*, *Netflix DrivingPOV*, *Vidyo4*, *Netflix DinnerScene*, and *KristenAndSara*. The Full HD sequences were *Netflix TunnelFlags*, *Jockey*, *Beauty*, *Touchdown Pass*, and *Rush Field Cuts*. Finally, 4K Ultra HD sequences were *ToddlerFountain*, *SunBath*, *Lips*, *BuildingHall2* and *Netflix Dancers*. All videos were encoded four times, once for each QP value.

The VVC VTM was modified to allow the extraction of the required information necessary for the assessment presented in this paper. These insertions were done inside the *EncCu.cpp* VTM file. The information was extracted for the four ME stages: Unidirectional, Bidirectional, 4-parameters Affine and 6-parameters Affine.

These experiments were run in a computer with a Intel Xeon E5-2650v4 2.20 GHz processor, 96 GB of DDR4 memory and a 2 TB SSD and required a total of 168 hours of execution. In a total, more than one hundred million CUs were registered to support the assessment presented in this paper.

The results for Unidirectional and Bidirectional ME were grouped and analyzed together since these tools have similar behavior and are present in other encoders. The two modes of Affine ME, with 4 and 6-parameters, were also grouped and analyzed together, since their behavior are similar.

To make the assessment more understandable, the CU sizes were grouped according to the number of samples they have. CU sizes grouping is shown in Table I, where the column "Weight" was defined as the number of samples inside each CU divided by 16, only to simplify the analysis presented in the next section of this paper.

After the encoding of all sequences with the four QPs, the features were extracted and the final results were reached following the steps below:

- The averages of usage as the best match and of the time spent to encode all CU sizes for each video and for the four QPs were calculated, considering the two groups of

TABLE I
CU Size Grouping

Group	CU Sizes	Weight
1	4×8 8×4	2
2	4×16 8×8 16×4	4
3	4×32 8×16 16×8 32×4	8
4	4×64 8×32 16×16 32×8 64×4	16
5	8×64 16×32 32×16 64×8	32
6	16×64 32×32 64×16	64
7	32×64 64×32	128
8	64×64	256
9	64×128 128×64	512
10	128×128	1024

tools: (i) Unidirectional and Bidirectional ME and (ii) Affine ME;

- Then, these averages of usage and time were grouped according to the video resolutions and new averages were calculated, again considering the two group of ME tools;
- Then, the averages of each CU size were added according to their weight, considering the 10 groups defined in Table I, for each resolution and for each ME tool group;
- Finally, the number of samples inside each CU were multiplied by the average usage of each group of CUs, to generate the average number of samples encoded with that group of CUs.

IV. Results and Discussion

The reached results of encoded samples using each group of CU sizes, considering the method presented in the previous section, are summarized in Fig. 2, with one graph for each group of sequences in each one of the three resolutions focused in this investigation: HD, Full HD and Ultra HD. The X-axis of these graphs presents ”Weight” defined in Table I, indicating the groups of CU sizes. The Y-axis presents the average number of samples encoded in each group of CUs for each execution (considering the average for the four QPs for each sequence, and the average of the results for all sequences with each resolution, as presented in the previous section). These graphs present, in blue, the results for Unidirectional and Bidirectional ME and, in red, the results for the two Affine ME modes.

The first observation related to the results presented in Fig. 2, only confirms that Affine ME is not available for CU groups with weights up to the value 8 (CU sizes smaller than 16x16), since this tool does not appear in the graphs for weights up to 8.

One important conclusion is that Unidirectional and Bidirectional ME have a similar behavior when considering the relative distribution among the different resolutions, with the highest number of samples being encoded using CU group with weights 32 in all cases. Other important conclusion is that the Affine ME is less used for higher resolutions than for lower resolutions, in terms of relative use.

Table II shows the average results of samples encoded with the two groups of ME tools for the three resolutions. These results are an average for all sequences and all QPs used to encode these sequences for the three resolutions. Observing Table II one can conclude that the Affine ME and the Unidirectional and Bidirectional ME have a balanced use for HD resolutions, with the Affine being a little bit more used. But for Full HD and Ultra HD resolutions, the Affine ME is much less used than the Unidirectional and Bidirectional ME. Another observation is that as the resolution grows, both tools tend to be more used and this is an expected result, since there are more samples to be encoded. But an important conclusion is that the relative relevance of the Affine ME is as smaller as higher is the resolution.

Another important analysis is related to the time spent to encode the CUs for the two group of tools for the three resolutions. Fig. 3 present graphs with the average results of encoding time spent for each CU groups weight, in a similar way of the graphs presented in Fig. 2.

Observing these graphs one can conclude that different resolutions cause important modification in the distribution of coding effort for the Unidirectional and Bidirectional ME among the different groups of CUs, where, as higher is the resolution, as higher is the effort spent with these tools to encode the smaller CUs.

Another important conclusion is that even being less used than the Unidirectional and Bidirectional ME, Affine ME has a very high computational effort. The relation between resolution and encoding time is consistent, since the relative Affine computational effort is higher for lower resolutions, where this tool is more used to define the best match, as presented in the previous discussion. But the surprising result is that the Affine computational effort is higher than the Unidirectional and Bidirectional effort in many cases. Table III shows the average encoding time of the two groups of ME tools for the three resolutions. This table clearly shows that the time spent to process Affine ME is more than double the time spent to process Unidirectional and Bidirectional ME for HD videos, even with the Affine being used to encode three times fewer CUs, as presented in Table II. The time spent to encoding the Affine ME is also higher than Unidirectional and Bidirectional ME for Full HD sequences, and similar for Ultra HD sequences. In this last case, the times spent are similar, but the Affine is used to encode only one-eighth of the CUs if compared with Unidirectional and Bidirectional tools.

V. Conclusions

This paper presented an investigation about the Unidirectional, Bidirectional and Affine ME of the VVC standard, when processing three different video resolutions.

Through the results and discussions presented in this paper, it is possible to conclude that the Unidirectional and Bidirectional ME are used to encode much more samples than the Affine ME, but require a smaller computational effort, regardless the video resolution. The presented results also allowed a conclusion that as higher is the video resolution, as

TABLE II
Average Samples encoded with the two groups of ME tools for the three resolutions

Resolution	Unidirectional and Bidirectional	Affine
HD	71,747,066	75,778,708
Full HD	660,475,022	498,660,097
Ultra HD	1,329,027,202	772,402,364

TABLE III
Average Encoding times of the two groups of ME tools for the three resolutions

Resolution	Unidirectional and Bidirectional (s)	Affine (s)
HD	33.37	79.70
Full HD	383.03	500.96
Ultra HD	778.42	738.24

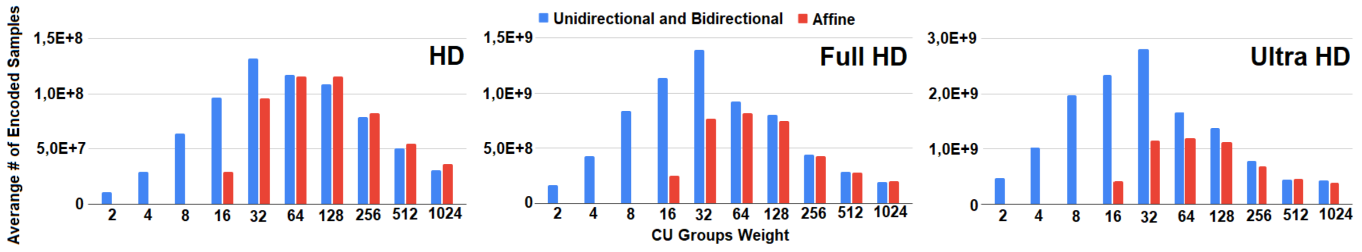

Fig. 2. Average number of samples encoded with each group of CU sizes in HD, Full HD and Ultra HD videos

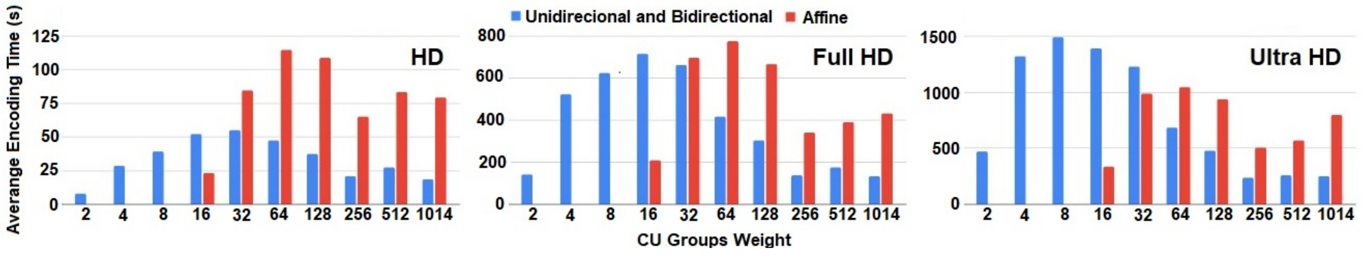

Fig. 3. Time spent to encode the groups of CU sizes in HD, Full HD and Ultra HD videos

lower is the relative use of the Affine ME in comparison with Unidirectional and Bidirectional ME. On the other hand, the relative use of different CU sizes does not present a significant change for different resolutions and encoding tools.

The results presented in this paper clearly point to new research opportunities related to the high complexity of the Affine ME. Then, novel solutions focused on reducing the encoding effort for this encoding tool tends to generate important impacts on the whole encoder.

Acknowledgment

The authors of this work would like to thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq for funding this research.

References

[1] Kai Zhang, Li Zhang, Hongbin Liu, Jizheng Xu, and Yue Wang, "Interweaved prediction for affine motion compensation," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 3158–3161.

[2] Seishi Takamura, "Versatile video coding: a next-generation video coding standard," *NTT Technical Review*, vol. 19, no. 6, 2019.

[3] Iain E. Richardson, *The H.264 Advanced Video Compression Standard*, WILEY, 2010.

[4] Ticao Zhang and Shiwen Mao, "An overview of emerging video coding standards," *GetMobile: Mobile Comp. and Comm.*, vol. 22, no. 4, pp. 13–20, May 2019.

[5] Mário Saldanha, Marcel Corrêa, Guilherme Corrêa, Daniel Palomino, Marcelo Porto, Bruno Zatt, and Luciano Agostini, "An overview of dedicated hardware designs for state-of-the-art av1 and h. 266/vvc video codecs," in *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2020, pp. 1–4.

[6] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[7] Jens Brandenburg, Adam Wieckowski, Tobias Hinz, Anastasia Henkel, Valeri George, Ivan Zupancic, Christian Stoffers, Benjamin Bross, Heiko Schwarz, and Detlev Marpe, "Towards fast and efficient vvc encoding," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6.

[8] Seongwon Jung and Dongsan Jun, "Context-based inter mode decision method for fast affine prediction in versatile video coding," *Electronics*, vol. 10, no. 11, 2021.

[9] Heiko Schwarz, Muhammed Coban, Marta Karczewicz, Tzu-Der Chuang, Frank Bossen, Alexander Alshin, Jani Lainema, Christian R. Helmrich, and Thomas Wiegand, "Quantization and entropy coding in the versatile video coding (vvc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3891–3906, 2021.

[10] Mathias Wien, *High Efficiency Video Coding – Coding Tools and Specification*, 10 2014.

[11] Thomas Amestoy, Alexandre Mercat, Wassim Hamidouche, Daniel Menard, and Cyril Bergeron, "Tunable vvc frame partitioning based on lightweight machine learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 1313–1328, 2020.

[12] Adson Duarte; Paulo Gonçalves; Luciano Agostini; Bruno Zatt; Guilherme Correa; Marcelo Porto; Daniel Palomino, "Fast affine motion estimation for vvc using machine-learning-based early search termination," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2022, (accepted paper).

[13] Paulo Henrik Ribeiro Gonçalves, "Um esquema rápido baseado em aprendizado de máquina para a predição interquadros do codificador de vídeo vvc.," Dissertação (mestrado em ciência da computação), Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2021.

[14] Sang-Hyo Park and Je-Won Kang, "Fast affine motion estimation for versatile video coding (vvc) encoding," *IEEE Access*, vol. 7, pp. 158075–158084, 2019.

[15] Xufeng Li, Ronggang Wang, Wenmin Wang, Zhenyu Wang, and Shengfu Dong, "Fast motion estimation methods for hevc," in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2014, pp. 1–4.

[16] Karsten Suehring, "Vtm-14.0," $https : //vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/ - /releases/VTM - 14.0$, Aug. 2021, Accessed: 2021-12-10.

[17] Frank Bossen, Jill Boyce, Karsten Suehring, Xiang Li, and Vadim Seregin, "Vtm common test conditions and software reference configurations for sdr video," 10 2020.

[18] Alexandre Mercat, Marko Viitanen, and Jarno Vanne, "UVG dataset," in *Proceedings of the 11th ACM Multimedia Systems Conference*. may 2020, ACM.

[19] Thomas Daede; Andrey Norkin; Ilya Brailovskiy, "Video codec testing and quality measurement," https://tools.ietf.org/id/draft-ietf-netvc-testing-08.html, Jan. 2019, Accessed: 2022-02-22.