# Rate-Distortion and Complexity Comparison of HEVC and VVC Video Encoders

†Ícaro Siqueira, ⋆Guilherme Correa, †Mateus Grellert
†Universidade Católica de Pelotas, Brazil
⋆Video Technology Research Group (ViTech) - Universidade Federal de Pelotas (UFPel), Brazil
icaro.siqueira@sou.ucpel.edu.br, gcorrea@inf.ufpel.edu.br, mateus.grellert@ucpel.edu.br

***Abstract*—Video-coding systems have presented significant improvements driven by the wide adoption of video streaming technologies combined with demands for better quality from users. The most recent video-coding standard from JCT-VC, named High Efficiency Video Coding, greatly improved the compression rate compared to its predecessor, H.264/AVC, but an even better performance must be pursued to accommodate future technologies. This article presents a comparison between the current state-of-art HEVC standard with the most recent project that is being conducted by the same group of experts, entitled Versatile Video Coding (VVC). According to experimental results obtained using similar configurations for both encoders, the VVC reference software provides significant bit-rate savings of 43% on average when compared to HEVC. However, this compression gains come with high computational costs: the VVC encoding time is on average 10.2 times higher when SIMD are used, and 15.9 times higher without such optimizations.**

***Index Terms*—video coding, HEVC, VVC, complexity**

## I. Introduction

Nowadays, an increase in video transmitted by means such as broadcast channels, social media and digital networks has been noticed. With the growth of this trend, demand for high-resolution video content is increasing, especially for high and ultra-high definition content (HD and UHD), as weel as for high frame rate content. As a result, the bandwidth required to transmit a video, as well as the amount of memory needed to store it, will grow proportionally with those features.

To provide high-quality videos to a large number of users, efficient video compression techniques need to be employed. High-Efficiency Video Coding (HEVC) is a video coding standard for video compression, developed by the Joint Collaborative Team on Video Coding (JCT-VC) [1]. HEVC is capable of increasing compression by 39% on average for the same image quality (measured in Bjontegaard Delta Bitrate – BD-BR) [2] when compared with its previous version, the H.264/AVC standard [3]. However, the cost for this significant coding efficiency gain is a substantial increase in the computational complexity at the encoder side. In fact, [4] states that HEVC requires from 20% to 50% more computations to compress data than H.264/AVC.

As higher resolution, higher frame rates, and immersive representations grow in popularity, it is reasonable to expect that HEVC will need to be replaced in the future. With that in mind, the Joint Video Experts Team (JVET) was established with the goal of exploring future video encoding technologies beyond HEVC. Since then, the most promising technologies have been integrated into Versatile Video Coding (VVC) [5].

VVC brings a handful of innovations compared to HEVC, such as larger block sizes, simpler partitioning structure, new inter modes, etc. However, computing resources are once again used to achieve better compression results. In fact, reports using the former JVET encoder (Joint Exploration Test Model 7 – JEM7) show that it is 11.3 times more complex than HEVC for a 25% BD-BR gain [6], i.e., an increase of 25% in compression efficiency. Such comparisons are very important, as they motivate the design of efficient implementations of VVC encoders, which will most likely be necessary in the coming years.

In this paper, we present a comparative analysis on the performance of HEVC and VVC encoders, including a discussion on the selected software implementations, the choice of parameters, and the evaluation configuration. A brief description of both encoders is also presented as theoretical background. The assessment focuses on both complexity, measured as processing time of the encoding task, as well as on compression/quality efficiency.

The paper is organized as follows: Section II presents a comparative discussion of the analyzed encoders. Section III describes the assessment methodology and the experimental setup. Experimental results are presented in Section IV and the work is concluded in Section V.

## II. HEVC and VVC encoders

As depicted in Fig. 1, the video coding process is composed of five steps: Prediction, Transform, Quantization, Filtering, and Entropy Coding. Prediction is divided into intra-frame and inter-frame prediction. Intra-frame prediction explores spatial redundancy and generates a predicted block based on neighboring samples from the same frame. Inter-frame prediction explores temporal redundancy by generating predicted samples from samples in previously encoded frames. Inter-frame prediction is divided into Motion Estimation (ME), which searches a block in reference frames and generates a Motion Vector (MV), and Motion Compensation (MC), which reconstructs the frame based on the obtained MV.

The prediction residue (i.e., the difference between the input and predicted samples) is then processed by the Transform and Quantization steps. The first one converts the residue to the frequency domain, and the latter eliminates coefficients

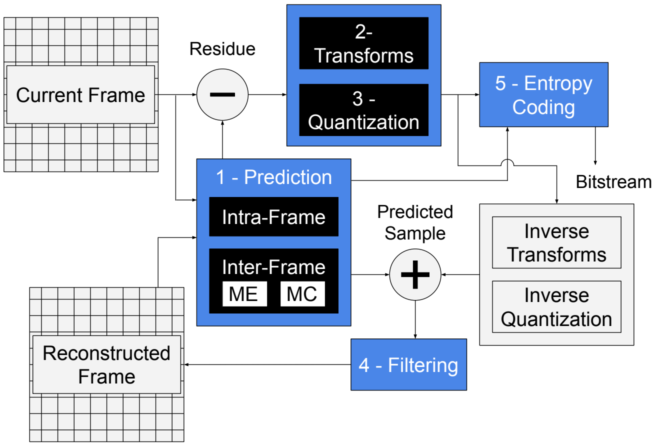

Fig. 1. Block diagram of a generic block-based video encoder.

TABLE I
HEVC VS VVC CODING TOOLS

| | HEVC | VVC |
|---|---|---|
| Block sizes | 64x64 - 4x4 | 128x128 - 4x4 |
| Partitioning Structures | CU, PU, TU in a CTU quadtree structure | CTU quadtree with nested multi-type binary/ternary tree |
| Intra-frame Prediction | 33 directional modes, DC and planar modes | 65 directional modes, DC and planar modes |
| Inter-frame Prediction | 1/4-pixel luma MV precision, 1/8-pixel chroma MV precision, merge mode | affine motion compensation, up to 1/16 MV precision (in affine mode), adaptive motion vector resolution (AMVR) |
| Transforms | 32x32 - 4x4 DCT, 4x4 DST | 64x64 - 2x2 DCT, DST |
| Filtering | SAO, Deblocking filter | SAO, Deblocking filter, ALF |

which will likely not be perceived by the human visual system. Quantization is controlled by a Quantization Parameter (QP), which is directly proportional to the strength of the coding loss, and controls the quality of the reconstructed video.

The filtering step is responsible for removing coding artifacts using smoothing techniques specially designed for block-based encoders (such as the Deblocking Filter). Finally, the Entropy Coding is responsible for compressing the residual data using variable-length or arithmetic coding techniques, generating the final coded bitstream of the video.

Table I shows the main differences between HEVC and VVC observed in their main modules. As new versions of the VVC standard draft are still in development, some coding tools may be included or removed until the conclusion of the standard [7], [8].

The HEVC frame partitioning scheme is presented in Fig. 3. Coding Tree Units (CTU) are partitioned in a recursive quadtree scheme, where each Coding Unit (CU) can assume a maximum size of 64x64 samples and a minimum size of 8x8 samples. CUs can also be partitioned into Prediction Units (PU) according to the partitioning modes shown in Figure 2

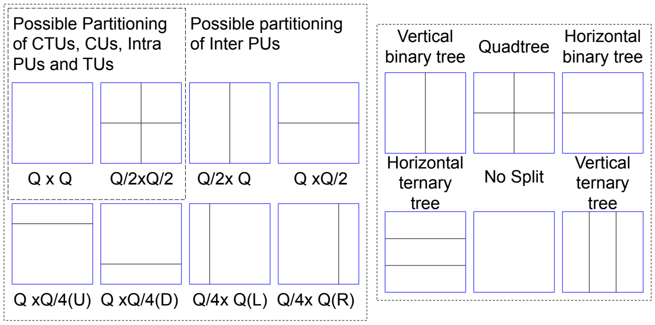

Fig. 2. Prediction block partitioning for HEVC (left) and VVC (right).

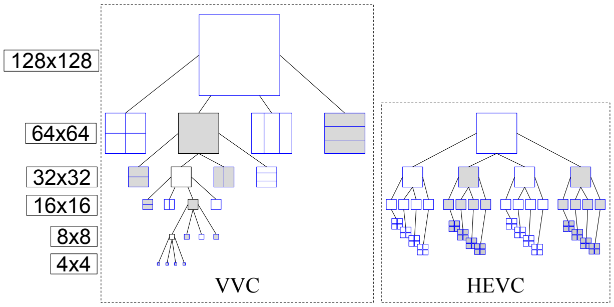

Fig. 3. Partitioning tree example for VVC (left) and HEVC (right).

and into Transformation Units (TU) in a recursive quadtree scheme with sizes ranging from 32x32 to 4x4 samples.

The VVC quadtree, also seen in Figure 3, is based on a nested multi-type tree structure. Besides the quadtree, CUs can also be partitioned into binary or ternary partitions, as shown in Figure 2. Thus, VVC has increased the number of partitioning types because binary and ternary partitions are not restricted to inter blocks, but can be used for intra prediction and transformation (restricted to the maximum and minimum transform sizes of 64x64 and 4x4 samples, respectively).

## III. Assessment Methodology

In the analysis presented in this work, both encoders were tested using QP values of 22, 27, 32, and 37, in accordance with the common test conditions (CTC) manual [9]. The Random Access (RA) temporal configuration was selected, motivated by the facts that it is typically chosen for broadcasting and streaming applications and provides in most cases the best results in coding efficiency when compared to the other options [10].

For the VVC reference software(VTM), parallelism tools were disabled (single threaded mode). As well, since the latest VTM versions contain Single Instruction/Multiple Data (SIMD) optimization kernels for Intel and ARM architectures, two binaries were compiled for testing: with and without SIMD.

The list of test sequences used is presented in Table II. All sequences are represented in a YUV 4:2:0 subsampling, with 8 bits per sample. Since 19 sequences were encoded under 4 QP values for each binary (with and without SIMD), a total of 228 encodings were conducted, so the number of encoded frames was limited to the first 60 ones. In addition, the tests

TABLE II
ANALYZED YUV SEQUENCES

| Sequence Name | Class | Resolution | Frame Rate | SI | TI |
|---|---|---|---|---|---|
| PeopleOnStreet | Class A | 2560x1600 | 30fps | 8.5 | 66.7 |
| Traffic | Class A | 2560x1600 | 30fps | 6.6 | 57.8 |
| BasketballDrive | Class B | 1920x1080 | 50fps | 5.5 | 39.7 |
| BQTerrace | Class B | 1920x1080 | 60fps | 11.4 | 81.4 |
| Cactus | Class B | 1920x1080 | 50fps | 6.9 | 70.7 |
| Kimono | Class B | 1920x1080 | 24fps | 2.8 | 54.1 |
| ParkScene | Class B | 1920x1080 | 24fps | 5.4 | 54.2 |
| Tennis | Class B | 1920x1080 | 24fps | 4.0 | 58.7 |
| BasketballDrill | Class C | 832x480 | 50fps | 7.4 | 40.1 |
| BQMall | Class C | 832x480 | 60fps | 9.8 | 55.7 |
| PartyScene | Class C | 832x480 | 50fps | 11.4 | 54.1 |
| RaceHorses | Class C | 832x480 | 30fps | 8.8 | 52.9 |
| BasketballPass | Class D | 416x240 | 50fps | 7.8 | 36.8 |
| BlowingBubbles | Class D | 416x240 | 50fps | 8.3 | 56.5 |
| BQSquare | Class D | 416x240 | 60fps | 16.9 | 83.9 |
| RaceHorses | Class D | 416x240 | 30fps | 10.6 | 51.9 |
| FourPeople | Class E | 1280x720 | 60fps | 7.4 | 61.6 |
| Johnny | Class E | 1280x720 | 60fps | 6.0 | 59.1 |
| SlideEditing | Class F | 1280x720 | 30fps | 21.8 | 83.9 |

were performed on an Intel Xeon octo-core CPU@3.60GHz, with 32 GB RAM using the GNU Parallel [11] to run at most 4 encoding tasks simultaneously (each task was single-core).

Table II also displays the Spatial and Temporal Indices (SI, TI) of each sequence. The SI and TI values estimate the level of texture and motion complexity of a scene, respectively, as described in [12]. The values were computed for the luminance (Y) component which is evaluated in the prediction.

Complexity was measured in two ways: (1) in terms of CPU running time; and (2) using to the GPROF profiling tool, which allows comparing the time spent on each function of both encoders. The first complexity measurements included all sequences from Table II, 60 frames, and both VVC versions (with and without SIMD optimizations). Due to the highly time-consuming nature of the profiling tool, only the SIMD VVC version was used in the second timing analysis. In addition, frame count was reduced to 30, and a subset of 9 video sequences was used: all from classes D and C and one from Class B (ParkScene).

Compression efficiency and quality were measured objectively. Specifically, the Bjontegaard-Delta bitrate (BD-BR) and PSNR (BD-PSNR) are used. The BD-BR represents the average difference in bitrate between a reference and a test encoder for the same image quality, when the test encoder is more efficient than the reference, the resulting BD-BR is negative [13]. Similarly, the BD-PSNR represents the difference in quality when the same bitrate is considered.

To compare both encoders in terms of complexity, the encoding time ratio (TR) showed on table III was computed $TR = T_{VTM}/T_{HM}$, where $T_{VTM}$ and $T_{HM}$ respectively stand for the average encoding time required by the HEVC reference software(HM) and the VTM encoders for the four tested QP values. This ratio can be interpreted as how many times more a sequence takes to be encoded under VTM compared to HM.

## IV. EXPERIMENTAL RESULTS

The reference software implementations used in this work were the VTM 5.0 [8], for VVC, and the HM 16.9 [7], for HEVC. A second VVC implementation is also available on the JVET repository, called JVET Experimental Module (JEM) [14]. JEM is a software that includes additional tools still under consideration (not yet elected for the final VVC draft). Thus, VTM is more recommended for performance comparisons that include processing time.

Table III presents detailed experimental results, including the calculated BD-BR and BD-PSNR. The results show that the VVC encoder is able to provide 44.4% bitrate savings for the same image quality, on average, which is a significant coding improvement. In terms of quality under the same bitrate, the average improvement is 2.1 dB, which is also significant. This means that VVC will be an important upgrade when larger resolutions hit the market, encoding sequences with high quality while requiring less storage resources and network bandwidth for transmission.

Table III also presents comparative encoding time values, calculated as the ratio between the VVC and the HEVC encoding times (considering both VTM with and without SIMD optimizations). The results indicate that VTM is more than 10 times slower in comparison to HM, even when SIMD optimizations are enabled. When the accelerating kernels are disabled, the average value increases to 15.9 times. Since HEVC is still considered a highly complex encoder, this increase raises an important concern for future VVC encoders, showing that it is essential to pursue solutions to cope with the complexity burden required by this emerging technology.

Profiling results are presented in Fig. 4, which shows complexity results per module for HEVC and VVC. For the sake of visibility, the Y-axis is displayed on logarithmic scale. In VVC, the most complex step is the Motion Compensation (MC), which is 19 times more complex than the HEVC MC. This is explained by the new inter-frame prediction tools implemented

TABLE III
HEVC VS VVC (BD-BR/PSNR AND RUNNING TIME)

| Sequence Name | BD-BR (%) | BD-PSNR (db) | TR SIMD | TR no SIMD |
|---|---|---|---|---|
| PeopleOnStreet | -38.85 | 1.97 | 11.07 | 15.18 |
| Traffic | -49.35 | 1.91 | 9.18 | 15.34 |
| BasketballDrive | -46.06 | 1.21 | 9.31 | 13.10 |
| BQTerrace | -59.71 | 1.57 | 11.18 | 16.41 |
| Cactus | -47.01 | 1.23 | 13.00 | 18.92 |
| Kimono | -28.91 | 1.00 | 8.68 | 15.07 |
| ParkScene | -43.02 | 1.53 | 14.54 | 25.79 |
| Tennis | -41.05 | 1.48 | 11.36 | 15.71 |
| BasketballDrill | -46.22 | 2.31 | 5.52 | 8.19 |
| BQMall | -42.77 | 2.17 | 4.88 | 8.25 |
| PartyScene | -48.53 | 2.77 | 13.16 | 16.87 |
| RaceHorses | -40.60 | 1.80 | 8.48 | 13.27 |
| BasketballPass | -40.96 | 2.37 | 11.78 | 18.20 |
| BlowingBubbles | -40.73 | 1.87 | 10.87 | 14.17 |
| BQSquare | -60.92 | 4.10 | 13.24 | 19.04 |
| RaceHorses | -41.00 | 2.19 | 13.15 | 24.06 |
| FourPeople | -41.68 | 1.78 | 3.12 | 4.66 |
| Johnny | -50.11 | 1.45 | 12.41 | 24.68 |
| SlideEditing | -34.47 | 5.53 | 8.28 | 14.81 |
| **Average** | **-44.40** | **2.07** | **10.17** | **15.88** |

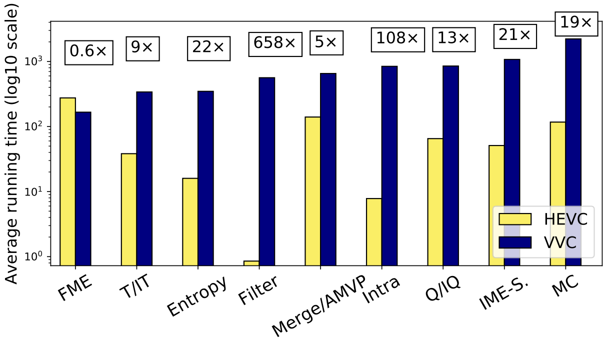

Fig. 4. Complexity comparison by module using a profiling tool

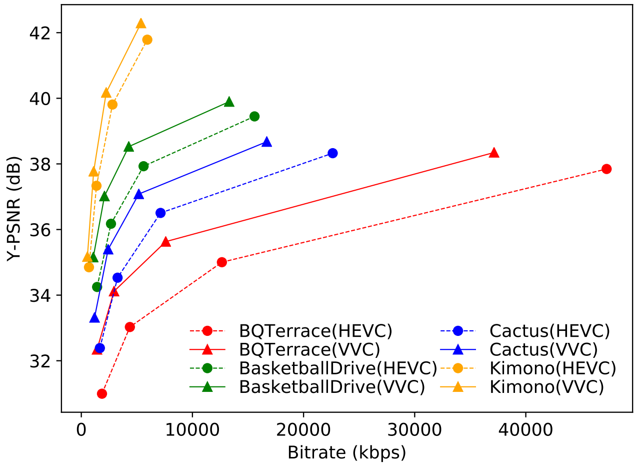

Fig. 5. Rate-distortion curves on four 1080p sequences.

in VVC, like the Affine Motion Vectors and Triangle Mode. The module that presents the largest complexity difference between VVC and HEVC is the Filter, which is 658 times more complex in VVC than in HEVC, especially due to the new Adaptive Loop Filter (ALF) tool. The analysis also shows that the most time-consuming step in HEVC is the Fractional Motion Estimation (FME), which is the only module that requires more time in HEVC than in VVC. However, the SIMD optimizations in VVC cause a large decrease in the processing time of this step.

Figure 5 presents a bitrate versus quality analysis for each QP comparing both encoders, considering four 1080p (1920x1080) input sequences. It is possible to see that the VVC encoder is superior in all curves both in terms of bitrate and PSNR. Note that the curve gap is larger when the bitrate range from QP 22 to 37 is higher. This is also reflected in larger BD-BR savings, as presented in the table IV.

Finally, Table IV shows detailed QP-wise encoding results for the ParkScene (1920x1080) sequence. It can be seen that VVC provides more bitrate savings on smaller QP values, ranging from 27% on QP=22 down to 20% on QP=37. Also note that the SIMD accelerators have a higher impact on larger QPs, showing that the accelerated kernels are more representative in the overall complexity as QP increases.

TABLE IV
QP-WISE COMPARISON OF BOTH ENCODERS (SEQUENCE: PARKSCENE)

| | **HM** | | **VTM** | | | |
|---|---|---|---|---|---|---|
| QPs | PSNR (db) | Bitrate (kbps) | PSNR (db) | Bitrate (kbps) | TR SIMD | TR no SIMD |
| 22 | 40.2 | 9404 | 41.0 | 6884 | 13.1 | 18.5 |
| 27 | 37.8 | 3719 | 38.7 | 2808 | 9.4 | 14.6 |
| 32 | 35.5 | 1599 | 36.4 | 1272 | 6.9 | 11.6 |
| 37 | 33.4 | 716 | 34.1 | 573 | 4.6 | 8.4 |

## V. CONCLUSION

In this paper, a rate-distortion and complexity comparison between the current state-of-the-art HEVC encoder and its emerging successor, VVC, has been presented. A comparison of the tools supported in each encoder indicated that VVC is much more computing-intensive than HEVC, and this was confirmed in the experimental analysis that followed. According to the results, the coding efficiency of VVC surpasses that of to HEVC, with average bitrate savings of 44.4% for the same image quality. At same bitrate, an average quality improvement of more than 2 dB is achieved by VVC. However, the experimental results have also shown that VVC encoding is 10 to almost 16 times more complex than HEVC. Finally, a profiling analysis has pointed out the main sources of such complexity in VVC, thus providing valuable insight for researchers in the field of multimedia systems that focus on developing solutions to enable this emerging technology in next-generation devices.

## ACKNOWLEDGMENT

This work was supported by the CNPq Brazilian research funding agency.

## REFERENCES

[1] ITU-T, "Recommendation H.265: High Efficiency Video Coding," 2013.

[2] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of h.265/mpeg-hevc, VP9, and h.264/mpeg-avc encoders," pp. 394–397, 2013.

[3] ITU-T, "Recommendation H.264: Advanced Video Coding," 2003.

[4] J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1885–1898, 2012.

[5] B. Bross, J. Chen, and S. L., "Versatile video coding (draft 5)," *14th Meeting of Joint Video Experts Team (JVET)*, 2019.

[6] M. Karczewicz and E. Alshina, "Jvet ahg report: Tool evaluation (ahg1)," *Joint Video Exploration Team (JVET) of ITU-T SG*, 2016.

[7] C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. J. Sullivan, "HM16.2: High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description Update 2," in *27th JCT-VC Meeting*, 2017.

[8] J. Chen, Y. Ye, and S. Kim, "Algorithm description for versatile video coding and test model 5 (vtm 5)," *14th Meeting of Joint Video Experts Team (JVET): Doc. JVET-J1001-v2*, 2019.

[9] F. Bossen *et al.*, "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, 2013.

[10] D. Grois, D. Marpe, T. Nguyen, and O. Hadar, "Comparative assessment of h.265/mpeg-hevc, vp9, and h.264/mpeg-avc encoders for low-delay video applications," *Proc. SPIE Vol. 9217*, 2014.

[11] O. Tange, "Gnu parallel - the command-line power tool," *The USENIX Magazine*, vol. 36, no. 1, pp. 42,47, Feb 2011.

[12] I.-T. R. ITU-R, "P910,"," *Subjective video quality assessment methods for multimedia applications*, 1999.

[13] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *VCEG-M33*, 2001.

[14] J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 7 (jem 7)," *Joint Video Exploration Team (JVET) of ITU-T SG*, vol. 16, 2017.