Indonesian Journal of Electrical Engineering and Computer Science
Vol. 35, No. 2, August 2024, pp. 898~906
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v35.i2.pp898-906

Complexity analysis of the VVenC versus VVC encoder

Hajar Touzani[1], Fatima Errahimi[1], Anass Mansouri[2], Ali Ahaitouf[1]
[1]Laboratory Intelligent Systems, Georesources and Renewable Energies, Faculty of Sciences and Technologies,
Sidi Mohamed Ben Abdellah University, Fez, Morroco
[2]Laboratory Intelligent Systems, Georesources, and Renewable Energies, National School of Applied Sciences,
Sidi Mohamed Ben Abdellah University, Fez, Morroco

Article Info

Article history:

Received Dec 8, 2023
Revised Mar 8, 2024
Accepted Apr 6, 2024

Keywords:

Compression
GPROF
Profiling
Run time
VTM
VVC
VVenC

ABSTRACT

The joint video experts team (JVET) has recently finalized a next-generation open-source video codec, called versatile video coding (VVC). The new standard presents a higher gain in the run time and rate compressions. Based on the reference software (VTM, VVC Test Model) of VVC, an optimized encoder was developed these last years resulting in a fast and efficient encoder, called Fraunhofer versatile video encoder (VVenC). Based on the Bjontegaard methodology and the GPROF profiling tool, this paper presents a technical complexity analysis and comparison of both VVC and VVenC. The appropriate comparisons cover the percentage taken by each block in terms of processing time and the resulting whole encoding time. The peak signal-to-noise ratio (PSNR) and bit rate between VVC and VVenC encoders based on the common test conditions manual are also analyzed and compared. The profiling results show that the VVenC encoder presents a maximum gain of runtime and bit rate of 90% and 20% respectively in classes A and D test sequences, compared to the VVC encoder.

Corresponding Author:

Hajar Touzani
Laboratory Intelligent Systems, Georesources and Renewable Energies
Faculty of Sciences and Technologies, Sidi Mohamed Ben Abdellah University
Fez, Morroco
Email: hajar.touzani@usmba.ac.ma

1. INTRODUCTION

Nowadays, there is an increase in video consumption in HD and UHD formats through the internet, packaged media, and TVs. This has created a great demand for an efficient video compression standard that can trade between an efficient compression, with an accelerated encoding time and responding to the highest quality. To satisfy this need, we are experiencing a continuous evolution of the Video Coding Standards. In 2013, the H.265/high efficiency video coding (HEVC) was published, this standard presented a doubled gain of bit rate compared to its predecessor H.264/advanced video coding (AVC), 2003 for the same video quality, but with a higher coding complexity [1]. The coding complexity of the HEVC standard has been addressed in several works, dedicated to its analysis and profiling. For example, Bossen *et al.* [2] presented an overview of the complexity and implementation issues of the HM (HEVC test model), the software reference of HEVC both on Intel and ARM processors. The profiling results show that in the random access (RA) configuration, the RDCost function takes the higher amount of time (40%) where the computation of the sum of absolute differences (SAD) and other distortion metrics take place, then the motion estimation/compensation takes the second higher percentage (20%) compared to other blocks. Similar complexity and performance analysis of HEVC on ARM processors has been reported by research [3], [4] based on both random access and all intra-configurations. Using the sequences provided by the joint collaborative team on video coding (JCT-VC) common test conditions [5], the profiling results show that in the random access configuration, the inter-prediction and RDCost blocks consume the highest percentage (38% and 29% respectively) of encoding time where only the interpolation filter function takes 30% in the inter prediction block. In the All Intra configuration, the Transform/quantization and intra-prediction blocks take the most computing time (33.1% and 28.4% respectively), the highest percentage (33.1%) taken by the transform and quantization block is presented by the rate distortion optimization quantization (RDOQ) function that requires an exhaustive search over multiple candidates to determine the optimal quantized level through rate-distortion optimization (RDO), which leads to considerable complexity in practice.

Versatile video coding (VVC) is the successor video compression standard of HEVC that was finalized in July 2020, by the joint video experts team (JVET) [6] formed by the united video expert team of the MPEG working group of ISO/IEC JTC and the VCEG working group of ITU-T. VVC introduced new technics to provide a superior compression efficiency when compared to HEVC. More recently, a most powerful software encoder implementation of the VVC was published; this newly published version named VVenC (published as open source by the Fraunhofer Institute at the end of the 2020 year), presents an optimized version of VTM (VVC test model) and its encoder operates much faster than the VVC standard encoder. However, the cost for this significant coding efficiency gain is a substantial increase in the computational complexity, mainly in the encoder side. So, before starting any optimization on the encoder, a detailed complexity analysis and comparison of performances between the different published versions should be performed.

To evaluate the performances and the complexity of the published standard, the VTM encoder and decoder were implemented in many versions of software reference (VTM) based on Python and are available in the website of the VVC standard [7]. Based on this software reference many analysis works have been undertaken. Two studies present a comparison between the HEVC and VVC standard [8], [9]. In the first study [8], the obtained experimental results show that the VVC reference software provides significant bit-rate savings of 44.4% and a speed-up of 10.2 times on average when compared to HEVC. In the second [9], the authors present an analysis of both the complexity and memory requirements for the VVC codec. The results indicate that the VVC encoder and decoder are respectively, 5× and 1.5× more complex than their equivalent HEVC units and that the motion estimation and compensation consume 53% of the total encoding time in VVC and 22% is spent in the transformation and quantization block.

Concerning the comparison between VTM and VVenC performances, Wieckowski *et al.* [10] performed an analysis and identification of the main features and differences between the two versions of VVC (VTM vs VVenC) with the software reference of HEVC (HM). Results show that VVenC is 30× faster than VTM with a 12% bitrate increase, while still providing a 38% bitrate reduction compared to HM. In the faster encoding mode, VVenC runs 140× faster than VTM with a decrease of only 10% in bitrate compared to HM.

Then, in the light of these results and to pave the way for a better trade-off between an accelerated encoder with a reduced bitrate, it is more convenient, recommended and highly useful to deeply analyze the complexity of this VVenC encoder, in order to analyze and identify the most computational blocs that should be more optimized, particularly in the field of hardware implementation projects. In this framework, by the present work, based on the common test conditions and reference software configurations provided by the JVET and Fraunhofer Heinrich Hertz Institute (HHI) groups [6], we provide a technical overview of the performances and issues of the new VVC encoder: VVenC (Encoder version v0.2.1.0) [11] compared to VVC (VTM v10.0) [12]. By taking advantage of the GNU profiler tool: GPROF [13], the performed comparisons cover the percentage taken by each block in terms of the processing time and the resulting whole encoding time. The peak signal-to-noise ratio (PSNR) and bit rate between VVC and VVenC encoders, based on the common test conditions manual, are also analyzed and compared. The profiling results show that the VVenC encoder presents a maximum gain of runtime of 90% and 20% respectively in classes A and D test sequences when compared to the VVC encoder.

This rest of the paper is organized as follows, an overview of the main features of VVC and VVenC encoders is presented in the section 2. The section 3 describes the test and work environment used to analyze the performances and complexity of both VVenC and VVC standards. Finally, the experimental results, discussions, and comparisons are presented in section 4.

2. VVC AND VVenC DESIGN ASPECTS

VVC standard, also known as H.266 presents the basic hybrid coding architecture as the predecessor video compression standards, such as the H.265/high efficiency video coding (HEVC) and H.264/advanced video coding (AVC) standards [14]. In the VVC encoder, the block structure is designed using a highly flexible partitioning scheme, where each frame is divided into coding tree units (CTUs) partitions with sizes that can reach 128×128 and 4×4 compared to HEVC which was limited only to 64×64 and 4×4 block sizes.

The CTUs are then divided into coding units (CUs) using a recursive quaternary tree (RQT) followed by a binary or ternary tree (BT or TT), these partitioning configurations named multiple tree type (MTT) were newly introduced in the VVC encoder, whereas the HEVC encoder partitions are divided based only on one recursive quadtree: RQT. In the VVC encoder, the new flexible block partitioning with multiple tree types [15], provides an increase in partitioning numbers and can be useful for intra-prediction and transformation blocks. To address the problem of HEVC's limitations for square-shaped TUs, the VVC standard introduced rectangular transforms for the transform module and this way, the transforms are directly applied in the CUs, without the need to apply another division for the residual blocks. Similarly, the flexibility provided by the MTT partitioning mode includes the prediction module, where CUs can be directly used for prediction. In summary, the VVC standard avoids the use of transform unit (TU) and prediction unit (PU) partitions and CU partitions are directly used in both transformation and prediction.

The partitioned blocks are then intra or inter-predicted. In the intra-prediction, 65 directional modes, instead of the only 33 modes in HEVC, in addition to the DC and planar modes that are used to predict a PU. Compared to HEVC, VVC has approximately twice the number of spatial intra-prediction sizes and spatial intra-prediction directions. The major change in the inter-prediction of VVC compared to the HEVC encoder is the affine motion model (AMM) which represents complex motions such as zooming, rotation, or shearing. This model is adopted in VVC and uses up to 1/16 motion vector (MV) precision versus only 1/4 and 1/8 for luma and chroma MV precisions are respectively used in HEVC. On the other hand, adaptive motion vector resolution (AMVR) has also been adopted into the VVC standard due to a favorable trade-off between the motion vector (MV) precision and the bit consumption on MV differences (MVDs) [16].

The output (i.e., the difference between the input and predicted samples) of the intra-inter prediction block is then processed by the transform and quantization blocks that can convert the residue (the output signal) to the frequency domain and eliminate the coefficients that cannot be perceived by the human visual system. In the VVC encoder, in addition to the deblocking filter (DBF) and the sample adaptive offset (SAO) filters designed for the HEVC standard, the filtering step specifies another in-loop filter adaptive loop filter (ALF) [17]. The goal of the filtering process is to remove the coding artifacts resulting from the previous encoding techniques (prediction, transform, and quantization) using smoothing techniques. Finally, in the entropy coding, the context-adaptive binary arithmetic coding (CABAC) is still unchanged and it is used to compress the residual data using variable-length or arithmetic coding techniques, generating the final coded bitstream of the video.

VVC encoder targets a reduction of 50% in the bit rate with the same visual quality when compared to the HEVC standard. The Fraunhofer versatile video encoder (VVenC) was recently initiated [11] to provide a fast and efficient video compression standard with a real-world example of the VVC encoder and decoder standard. These new improvements are based on the adaptation of extensive single instruction multiple data (SIMD) optimizations, the improved encoder search algorithms, and the multi-threading to support the parallelization. To better understand and integrate the innovations and improvements by these new standards, it is necessary to go through a deep comparison of their performances and an analysis of their complexity, which is the main aim of the present work, in order to pave the way for more and methodical optimizations. Hence, the next section is dedicated to the implementation assessment performed to analyze the complexity and the performance of both VVC and VVenC standards on Intel processors.

3. TEST ENVIRONMENT

3.1. Software encoders test model

All standard-based technologies are implemented as a "test model" to evaluate and compare the codec efficiency and performance. The HEVC standard is implemented as HM (HEVC test Model) reference software coded in C++ language and available online on the Fraunhofer HEVC website [18]. VVC encoder is based on the VTM (VVC test model), the open reference software implementation of VVC encoder and decoder standard. The latest version (the version VTM v10.0) is used in comparisons since this version already integrate single instruction/multiple data (SIMD) optimization kernels for Intel architecture. According to the Fraunhofer Heinrich Hertz Institute (HHI) document [19], the VVenC is based on an optimized VTM reference software providing optimizations on SIMD and multi-thread processes. This reference software encoder and decoder are fully available on the GitHub websites [19], [20] with the different steps needed to build and execute the optimized VTM to have a complete evaluation and verification of the proposed technologies during standardization.

3.2. Test environment

The purpose of the VTM encoder is mainly to provide a common test bed implementation of the VVC encoder proving a set of test sequences with, in addition, a list of encoder configurations according to the JVET common test conditions, these configurations include the following:

- All intra (AI): In this mode, all frames are encoded as I-frames, which means that the motion estimation/compensation is disabled. This mode is too fast compared to other modes but it gets lower compression rates.
- Random access (RA): In this setup, I-frames and B-frames are combined in a group of pictures (GOPs) to be used as references to encode the current frames. Each block of a B-frame can use up to two reference frames, that's why this mode takes more time compared to the All Intra mode but achieves a good compression rate compared to the other modes.
- Low delay (LD): In this configuration, the first frame is encoded as an I-frame then only P-frames (or B-frames) are used for the rest of the video sequence. This configuration can trade-off between a higher compression rate compared to AI mode and is faster than the RA setup.

Concerning the VVenC standard, the encoder is adapted to five presets (randomaccess_faster.cfg, randomaccess_fast.cfg, randomaccess_medium.cfg, randomaccess_slow.cfg, and randomaccess_slower.cfg) [21], each preset presents a different trade-off between the running time and the compression rate. In the preset "randomaccess_faster.cfg", the encoder is much faster than the "randomaccess_slower.cfg" preset, but the quality and the compression rate of the encoded video are slightly lower than those achieved by the slower preset.

These configurations were used to analyze the performances of the encoder and decoder in different modes. However, further and in-depth studies may be needed to analyze the computational complexity of different functions in VVC. In this paper, the analysis of the encoders (VVC and VVenC), four Qp values (22, 27, 32, and 37) are tested with the different configuration files presented above, by the common test conditions. The test sequences adopted in this work are presented in Table 1. All sequences are represented in a YUV 4:2:0 subsampling, with 8 bits per sample. Since 4 of the test sequences were encoded with the four Qp values tested and the different configurations presented in the VTM encoder, a total of 38 VTM10.0 were tested for the VVC encoder and a total of 64 VVenC v0.2.1.0 were tested in the VVenC encoder. The tests were performed on a standard platform (Intel Core i7, 3.6 GHz, 4 GB of RAM) under the Linux (Ubuntu 18.04) operating system.

Table 1. Test sequences

Classes	Sequences	Resolution	Frame rate [Hz]
A	People on street	2,560×1,600	30
B	Basketball drive	1,920×1,080	50
C	Vidyo1	1,280×720	60
D	RaceHorses	832×480	30
E	BQSquare	416×240	60

The reference software implementations used here are the VTM10.0 [12], for VVC, and the Encoder version v0.2.1.0 [19], for VVenC. The profiling was performed on the Linux operating system based on the GPROF tool [13], and results of the spent time on each function for the two encoders are highlighted. GPROF, is an open-source profiling tool, it is used to learn where the encoder spent its time. Additionally, this tool is most often used on the Linux operating system, and it can be used for future works on different platforms, especially on ARM processors. This procedure leads to identifying the more time-consuming parts in the program code, which then need to and can be optimized, to make the encoder execution faster, which is always desired. Based on the Bjontgaard methodology, both bitrate and PSNR are compared in the different presets to have a detailed analysis of the different bitrates and compressed video qualities when compressed with VVC and VVenC.

4. IMPLEMENTATION RESULTS AND DISCUSSION

4.1. Encoder profiling results on random access-mode

The results presented in this section focus on the most relevant HD (1,920×1,080) and UHD (3,840×2,160) resolution use cases with random access encoding as defined in the JVET common test conditions. In that regard, Figures 1 and 2 highlight respectively the profiling results of the VTM10.0 in random access (RA) configuration and the v0.2.1.0 (VVenC) in the randomaccess_slower preset for the PeopleonStreet (class A) sequence. Based on Gprof profiling results, Table 2 reports the time percentage taken by the various C++ classes for both VVC versions for more details on the most computational functionalities. Results show that the RDCost and prediction blocks take the great amount of time in VVC. RDCost which corresponds to the decision block of VVC and which is responsible for the control and decisions of the sizes and modes used for each partition, takes 18% and 26% respectively for both VTM and VVenC as shown in Figures 1 and 2.

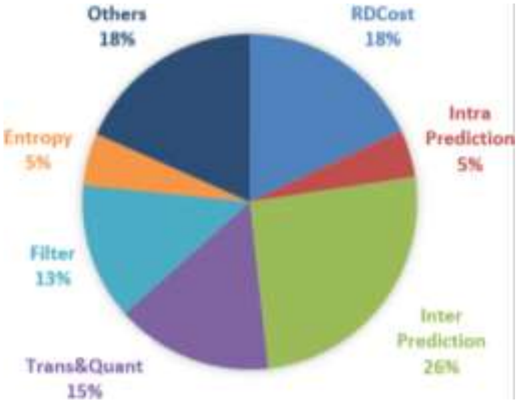

Figure 1. Profiling results of VTM encoder

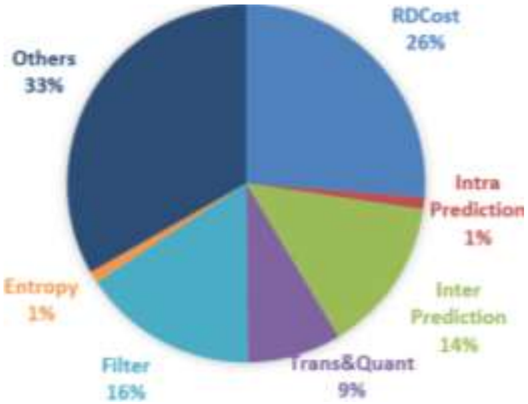

Figure 2. Profiling results of VVenC encoder

As presented in Table 2, the functions intra/inter search account for about 6% in VTM and the function calcHAD (Hadamard operations) takes to 11.43% in VVenC (see Table 2). In the random access configuration, the inter-prediction block presents the most computational block in previous standards as reported in the state-of-the-art [22]-[24]. In VTM, the inter prediction accounts for 26% and only 14 % in VVenC. This can be explained by the adaptation of the extensive SIMD optimizations (simdInterpolate presented in Table 2) in VVenC to accelerate the inter prediction. The filter block takes a higher percentage in VVenC (16%) in comparison with the percentage taken on VTM (13%), this block includes the Adaptive Loop Filter (1.67% in VTM, 8.57% in VVenC), the AlfCovariance (0.4% in VTM, 2.95% in VVEnC) and the SIMDFilter as presented in Table 2. The transformation/quantization block is also optimized in the VVenC encoder by the adaptation of the multithreading technique, this can be explained by the difference of consumed time in this block from 15% in VTM to 9% in VVenC which includes the DQIntern, DCT, DST, and TrQuant functions as shown in Table 2. Based on these results, our study demonstrates that the inter prediction and RDCost still the most computational blocks, that for future studies, must be explored and optimized using different optimized implementation technics (parallelism, hardware implementation) to accelerate more the encoding time.

Table 2. Profiling results for VVC and VVenC encoders; distribution by class

Blocks	Functions	VTM	VVenC
RDCost	XCalcHAD	0,01%	11,43%
	InterSearch	4.57%	1.08%
	RDO	12.07%	13.46%
	IntraSearch	1.38%	0.29%
Intra Prediction	IntraPrediction	4.52%	1.06%
Inter prediction	Interpolation Filter	3.48%	0.39%
	SIMDInterpolate	0.2%	4.37%
	Inter Prediction	21.95%	9.28%
Trans&Quant	TRQuant	2.03%	0.93%
	DQIntern	9.65%	7.09%
	DCT & DST	3.5%	0.47%
Filter	AlfCovariance	0.4%	2.95%
	Adaptive Loop Filter	1,67%	8,57%
	SIMDFilter	11.13%	4,35%
Entropy	CABACWriter	5.02%	1.04%

4.2. Encoder profiling results on all intra mode

The VTM10.0 has also been profiled in the case of the all intra mode to determine the most time-consuming functions presented in the intra-prediction mode as shown in Figure 3. In the AI configuration compared to the RA configuration, it is evident that motion estimation takes 0% because of disabling the motion compensation coding in this setup. The transformation/quantization and the intra-prediction blocks take a significant amount of the encoding time (34.21%, as shown in Figure 3 for the TrQuant and 17% for the intra-prediction), where the calculation of angular and planar modes are strongly presented. Computation of the sum of absolute differences (SAD) and other distortion metrics takes place in the TComRdCost class, which accounts for about 12% of the encoding time. Furthermore, partitioning functionalities account for 14%, this high percentage can be explained by the new partitioning methods that were newly introduced on VVC and that are based on binary, ternary, and quaternary trees as described in section 2.

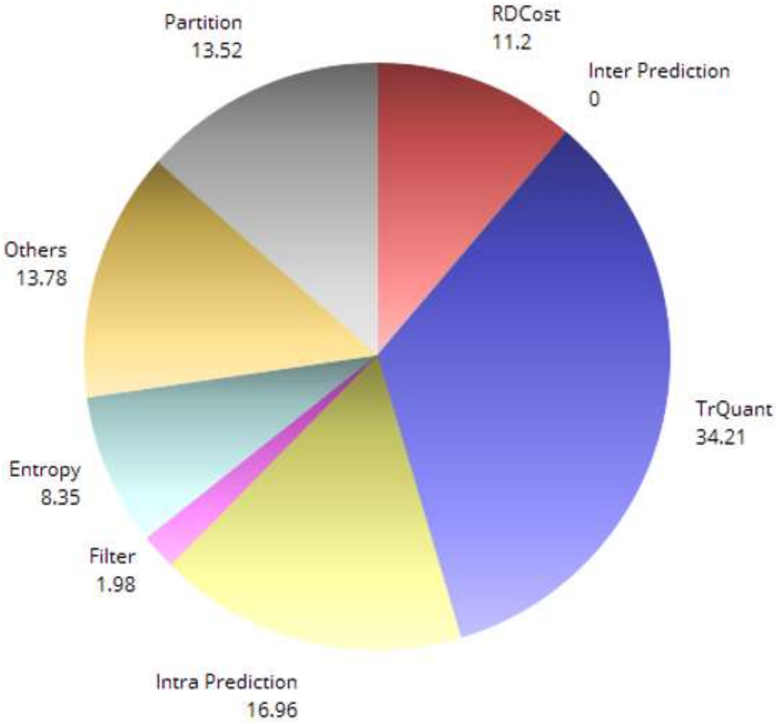

Figure 3. Profiling results of VVC encoder using AI configuration

4.3. VVC vs VVenC compression and quality analysis results

Table 3 shows the encoding times, the PSNR, and the bit rates for the classes A, C, D, and E. Results present the approximate average comparisons for the four predefined preset modes (randomaccess_slower.cfg, randomaccess_Medium.cfg, randomaccess_Fast.cfg and randomaccess_Faster.cfg). The quality and compression efficiency were measured based on the Bjontegaard-Delta PSNR (BD-PSNR) and bitrate (BD-BR) respectively [25]. The BD-PSNR represents the average difference in PSNR between the resulting bitstream quality of the VTM encoder and the VVenC encoder. The BD-BR is the difference between the bit rates of VTM and VVenC encoders. Additionally, the performances of VVenC is evaluated in comparison with the VTM encoder in terms of encoding time that is calculated using the following:

$$\Delta T = (T1 - T2)/T1$$

Where T1 and T2 are the running time of the VTM 10.0 and v0.2.1.0 respectively.

In the analysis results presented in the Table 3, we found that the encoding time in VVenC was optimized by a maximum percentage of 90% in class A in the faster-preset mode (randomaccess_Faster.cfg), and a minimum of 28.22% in class C in the slower preset mode (randomaccess_Slower.cfg). On the other hand, the bitrate was increased by around 20% in the slower mode and a minimum of 2% in fast and faster presets.

Table 3. VTM vs VVenC timing, PSNR, and bit rate comparison

	Faster			Fast			Medium			Slower		
	ΔT (%)	BD-PSNR (%)	BD-BR (%)	ΔT (%)	BD-PSNR (%)	BD-BR (%)	ΔT (%)	BD-PSNR (%)	BD-BR (%)	ΔT (%)	BD-PSNR (%)	BD-BR (%)
A	90.06	-0.77	5.78	80.88	-0.23	9.54	62.55	-0.09	12.13	44.5	-0.35	13.08
C	87.1	-0.9	7.02	86.43	-0.53	10.2	73.99	-0.07	11.87	28.22	-0.04	14.96
D	78.65	-1.24	9.07	78.31	-1.55	11.79	58.67	-2.3	12.2	40	-1.08	19.2
E	87.05	-2.06	2.43	86.06	-1.4	2.52	74.89	-0.81	6.57	63.8	-0.44	8.02

4.4. Discussion

Figures 4 and 5 show respectively the gain in runtime and the bit rates for classes A, C, D, and E concerning the slower, medium, fast, and faster-preset modes for the VVenC encoder. It can be seen that this encoder is faster in the Faster mode when compared to the medium and slower modes for classes A, C, D, and E. The bit rate is more increased in the slowest presets (19.2% in the slower mode, 12.2% in the medium mode, 11.79% in the fast mode and 9.07% in the faster mode as presented in the Table 3). Wieckowski *et al.* [10], the encoding time has also been accelerated in VVenC, results show that VVenC is 140× and 30× faster than VTM with a 12% and 22% increases in the bitrate, respectively for faster and slower presets. In conclusion, to have a good tradeoff between encoder runtime and BD-rate performance in the VVenC encoder, the medium preset (randomaccess_medium.cfg) is recommended.

Based on the analysis and profiling results performed in this work and presented in the previous section, Figure 6 presents the results of the comparison between the algorithmic complexity per module for the two encoders: VTM and VVenC. For the sake of visibility, the Y-axis displayed the average running time for both VTM and VVenC encoders for all blocks where the green color refers to the VVenC encoder and the blue color refers to the VTM encoder. In VVenC, the most complex module is the filter followed by the RDCost functions that is also more complex than in the VTM encoder. The other blocks present a higher complexity in VTM in comparison to the VVenC encoder (entropy, motion compensation, intra and transform/quantization are more complex in VTM compared to VVenC by factors of x5, x2.5, x2, and x2 as highlighted in Figure 6). As a result, the more optimized block in VVenC compared to VTM version is the entropy block that presents a factor of 5 times in terms of running time compared to VVenC.

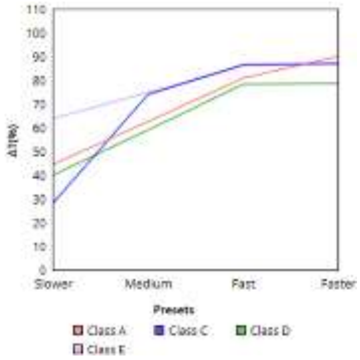

Figure 4. Comparison of the runtime gain

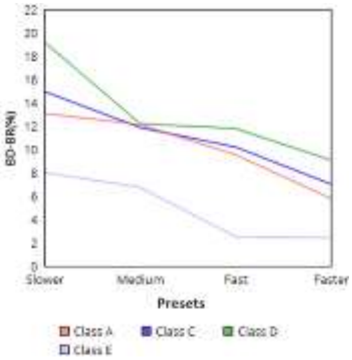

Figure 5. Comparison of the gain in bit rate

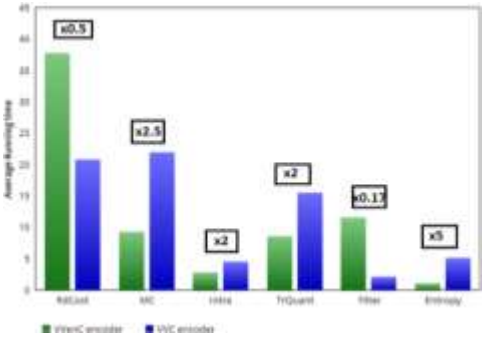

Figure 6. Complexity comparison by module using GPROF

5. CONCLUSION

In this paper, a profiling analysis between the current state-of-the-art VTM encoder and its optimized version VVenC encoder is presented. Based on the GPROF tool the profiling results show that the RDCost and motion compensation (MC) are the most computational blocks both for VVenC and VTM encoders in the random access mode. When in terms of complexity the entropy functionalities are 5 times more optimized in VVenC compared to VTM encoder. A comparison of the execution time, PSNR, and bit rate between VTM and VVenC encoders proves that the VVenC is too fast compared to VTM with a percentage that can reach 90% in PeopleOnStreet test sequence, and more optimized in terms of bit rate by a percentage of 20% in RaceHorses class. These statistical results are helpful for hardware implementation projects and provide valuable insight for researchers in the field of multimedia systems that focus on developing solutions to enable this emerging technology in next-generation devices. Users can then easily have a clear idea of which block to focus on to accelerate the optimization and hardware implementation of this new standard.

REFERENCES

[1] N. Minallah, S. Gul, and M. M. Bokhari, "Performance Analysis of H.265/HEVC (High-Efficiency Video Coding) with Reference to Other Codecs," in *2015 13th International Conference on Frontiers of Information Technology (FIT)*, IEEE, Dec. 2015, pp. 216–221. doi: 10.1109/FIT.2015.46.

[2] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012, doi: 10.1109/TCSVT.2012.2221255.

[3] H. Touzani, F. Errahimi, A. Mansouri, and A. Ahaitouf, "Implementation analysis of HEVC encoding on zynq platform under embedded linux," in *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, IEEE, Apr. 2017, pp. 1–5. doi: 10.1109/WITS.2017.7934607.

[4] H. Touzani, A. Mansouri, F. Errahimi, and A. Ahaitouf, "Implementation analysis of HEVC encoding on dual ARM Cortex A15 architecture," in *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, IEEE, Mar. 2016, pp. 1–5. doi: 10.1109/IT4OD.2016.7479325.

[5] F. Bossen, "Common test conditions and software reference configurations." Document JCTVC-H1100, 2012.

[6] "JVET - joint video experts team," Committed to connecting the world. [Online]. Available: https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jvet.aspx.

[7] "Svn_VVCSoftware_VTM." [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_VVCSoftware_VTM.

[8] I. Siqueira, G. Correa, and M. Grellert, "Rate-Distortion and Complexity Comparison of HEVC and VVC Video Encoders," in *2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)*, IEEE, Feb. 2020, pp. 1–4. doi: 10.1109/LASCAS45839.2020.9069036.

[9] F. Pakdaman, M. A. Adelimanesh, M. Gabbouj, and M. R. Hashemi, "Complexity Analysis Of Next-Generation VVC Encoding And Decoding," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, Oct. 2020, pp. 3134–3138. doi: 10.1109/ICIP40778.2020.9190983.

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

[11] "VVenC, the Fraunhofer Versatile Video Encoder." [Online]. Available: https://github.com/fraunhoferhhi/vvenc.

[12] "Milestone VTM-10.0." [Online]. Available: https://jvet.hhi.fraunhofer.de/trac/vvc/milestone/VTM-10.0?by=version.

[13] S. L. Graham, P. B. Kessler, and M. K. Mckusick, "Gprof: a call graph execution profiler," in *Proceedings of the 1982 SIGPLAN symposium on Compiler construction - SIGPLAN '82*, New York, New York, USA: ACM Press, 1982, pp. 120–126. doi: 10.1145/800230.806987.

[14] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, "Versatile video coding editorial refinements on draft 10," *Fraunhofer Heinrich Hertz Institute*. [Online]. Available: https://jvet.hhi.fraunhofer.de.

[15] Y.-W. Huang *et al.*, "A VVC Proposal With Quaternary Tree Plus Binary-Ternary Tree Coding Block Structure and Advanced Coding Techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1311–1325, May 2020, doi: 10.1109/TCSVT.2019.2945048.

[16] Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Design of Perspective Affine Motion Compensation for Versatile Video Coding (VVC)," 2020, pp. 384–395. doi: 10.1007/978-3-030-40605-9_33.

[17] M.-Z. Wang, S. Wan, H. Gong, and M.-Y. Ma, "Attention-Based Dual-Scale CNN In-Loop Filter for Versatile Video Coding," *IEEE Access*, vol. 7, pp. 145214–145226, 2019, doi: 10.1109/ACCESS.2019.2944473.

[18] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, Y.-K. Wang, and T. Wiegand, "High-Efficiency Video Coding (HEVC)," Fraunhofer Heinrich Hertz Institute. [Online]. Available: https://hevc.hhi.fraunhofer.de.

[19] J. Brandenburg, A. Wieckowski, T. Hinz, I. Zupancic, and B. Bross, "VVenC Fraunhofer Versatile Video Encoder," Fraunhofer Heinrich Hertz Institute. [Online]. Available: https://www.hhi.fraunhofer.de/fileadmin/Departments/VCA/MC/VVC/vvenc-v1.2.0-v1.pdf.

[20] "VVdeC.Software repository." [Online]. Available: https://github.com/fraunhoferhhi/vvdec.

[21] "Working practices using objective metrics for evaluation of video coding efficiency experiments." ITU-T and ISO/IEC JTC 1, 2020.

[22] F. Saab, I. H. Elhajj, A. Kayssi, and A. Chehab, "Profiling of HEVC encoder," *Electronics Letters*, vol. 50, no. 15, pp. 1061–1063, Jul. 2014, doi: 10.1049/el.2014.1010.

[23] A. Mercat, A. Makinen, J. Sainio, A. Lemmetti, M. Viitanen, and J. Vanne, "Comparative Rate-Distortion-Complexity Analysis of VVC and HEVC Video Codecs," *IEEE Access*, vol. 9, pp. 67813–67828, 2021, doi: 10.1109/ACCESS.2021.3077116.

[24] D. García-Lucas, G. Cebrián-Márquez, and P. Cuenca, "Rate-distortion/complexity analysis of HEVC, VVC and AV1 video codecs," *Multimedia Tools and Applications*, vol. 79, no. 39–40, pp. 29621–29638, Oct. 2020, doi: 10.1007/s11042-020-09453-w.

[25] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves." ITU SG16, 2001.

BIOGRAPHIES OF AUTHORS

Hajar Touzani is a teacher and researcher in Sidi Mohammed Ben Abdellah University of Fez. She received her embedded system engineering degree from the National School of Applied Sciences, Fez, Morocco in 2015. She is currently a member of research lobaratory: Laboratory Intelligent systems, georesources and renewable energies, faculty of sciences and technologies, University Sidi Mohamed Ben Abdellah of Fez. Her PhD work focuses on the real time embedded Implementations, video coding and image processing, software/hardware implementations. She can be contacted at email: hajar.touzani@usmba.ac.ma.

Fatima Errahimi is a teacher and researcher at the Faculty of Sciences and Technologies in Sidi Mohammed Ben Abdellah University of Fez. She got PhD degree on Automatic Control Systems and robotics in 2004. Her special fields of interest include the integration of renewable energy sources in building energy management, optimization control for demand side units in households and smart grid, signal and image processing. She participated in many research projects among which the IRESEN one on low cost CPV in Morocco. She supervised many PhD thesis and several post graduate students. She can be contacted at email: errahimifatima@yahoo.fr.

Anass Mansouri received Ph.D degrees in Microelectronics and Embedded System from Faculty of Sciences and Technologies of Fez, Morocco, in 2009. He is a Professor in National School of Applied Sciences of Fez. His major research interests include VLSI and embedded architectures design, video and image processing, as well as Software/Hardware design and optimization. He is member of the Intelligent systems, Georesources and renewable Energies and head of the team Embedeed systems, electronics and telecommunication. He can be contacted at email: anass.mansouri@usmba.ac.ma.

Ali Ahaitouf teacher and researcher in the University of Sidi Mohammed Ben Abdellah of Fez. He obtained his PhD in electronics in 1998. He was the ex-director of the Intelligent systems, Georesources and renewable Energies. His research field covers microelectronics and solar compounds, digital and analog design of the integrated circuits, Image and data ompression. He managed many research multilateral projects, related to analog design optimization, electronics component characterization and optimization as well as solar energy under concentration (CPV). He supervised a dozen of PhD thesis and published more than fifteen papers in renowned journals. He can be contacted at email: alihas@upm.edu.my