

Advanced Algorithms for Compression of Videos with Ultra High Resolution

O. Soukenik, L. Polak, J. Kufa

Brno University of Technology, Faculty of Electrical Engineering and Communication,
Department of Radio Electronics, Technicka 12, 616 00 Brno, Czech Republic

E-mail: xsouke02@vut.cz, polakl@vut.cz, kufa@vutbr.cz

Abstract—In this paper, attention is devoted to advanced video codecs (HEVC, VVC, VP9 and AV1) and their utilization for compression of videos with Ultra High Resolution (UHD), also marked as 4K and 8K. A Python-based program employing FFmpeg multimedia framework is created to explore compression performances of such video codecs. It allows to automate encoding process and record all important information during the video compression (e.g. the required CPU time, time consumption, RAM usage). The performance of the video compression algorithms is evaluated in terms of objective quality metrics. First experimental results show that video codecs VVC and AV1 could replace HEVC and VP9 for 4K and 8K videos in the future.

Keywords— video codecs, objective metrics, FFmpeg, HEVC, VVC, VP9, AV1, Python

1. INTRODUCTION

Demand for videos in Ultra High Resolution (UHD), many times abbreviated as 4K and 8K, are gradually increasing. To broadcast video content in such a resolution with appropriate visual quality, especially in the case of Video on Demand (VoD) [1], efficient video compression algorithms (codecs) are required. The video compression tools need to be versatile and must excel with very flexible settings. The most established video codecs, High Efficiency Video Coding (HEVC) and VP9, can be utilized to compress videos with full HD resolution or 4K. However, their performances for videos with higher resolution are lower, especially at lower bitrates. Versatile Video Coding (VVC) [2] and AOMedia Video 1 (AV1) [3] are emerging compression algorithms developed for effective compression of videos in different formats. Compared to HEVC and VP9, their utilization for 4K and 8K videos can be more effective [4], [5].

In this paper, attention is devoted to compression performances of video codecs HEVC, VVC, VP9 and AV1. The visual quality of a test video compressed by these codecs is evaluated by different objective quality metrics. To automate and manage the encoding process of the video, a Python-based program employing FFmpeg¹ multimedia framework is developed and created.

This short paper is organized as follows. Video codecs used in this work are briefly introduced in Section 2. Objective metrics used to evaluate the quality of compressed videos are described in Section 3. Section 4 and Section 5 are devoted to the developed Python program to automate the process of video encoding and settings used to compression of a video, respectively. The obtained preliminary results are briefly evaluated in Section 6 while Section 7 concludes this paper.

2. ADVANCED VIDEO CODECS

The latest generation of video codecs supports computationally intensive algorithms used to improve intra frame compression by copying similar parts of a frame (intra prediction) or using a feedback with full decoding and analysis with objective metrics to improve the encoder's decision algorithm. Next, such video encoders have an additional option to make transmission more efficient for very small data stream (full prediction from surrounding blocks when the encoder assesses that the error will be small at using of an appropriate restoration filter). Significance of these improvements is visible mainly at transmission of synthetically generated contents such as game streaming, presentations and screen sharing.

In this work, we compare compression performance of video codecs HEVC, VVC, VP and AV1. HEVC and VVC have been developed within the family of H.26x video coding standards. Video codecs VP9 and AV1, compared to HEVC and VVC, are open and royalty-free, used mainly in the field of streaming (YouTube, Netflix). AV1 is developed by the Alliance for Open Media including such major companies like Apple, Facebook, Google, Microsoft, Netflix and Amazon.

¹<https://ffmpeg.org/>

3. OBJECTIVE VIDEO QUALITY METRICS

The visual quality of a compressed video can be evaluated either by subjective or objective way. Subjective quality evaluation is based on the scores obtained from people who evaluate the video quality under laboratory conditions. The outputs of so called subjective tests have high information value (scores directly obtained from people). On the other hand, they are costly and time consuming. Objective quality metrics utilize mathematical-based algorithms to reveal the visual quality of the compressed videos. Depending on the availability of reference (uncompressed) source, these metrics can be full, reduced or non-reference (FR, RR and NR). Objective metrics are cost effective, on the other hand, but, in general, are less punctual than subjective-based studies.

In this work, the visual quality of videos is evaluated in terms of objective metrics. For this purpose, Peak Signal-to-Noise Ratio (PSNR) and Video Multimethod Assessment Fusion (VMAF) objective metrics are used. PSNR is a well-known metric used to assess initial visual quality of a video. VMAF is a combination of multiple objective quality metrics that have coefficients set using Mean Opinion Score (MOS) to make the final score closer to subjective rating.

4. TOOL FOR ENCODING OF VIDEO SEQUENCES

The process of video encoding, thanks to numerous options and settings of advanced compression algorithms, is not a straightforward task. Many times, reference implementation of these video codecs is command line based, and the definition and set of parameters can be not user friendly. Hence, within this work, a utility² for automate encoding of video sequences is created in program language Python. It allows to set and change encoding settings independently on the used video codec. In this way, a number of tasks can be easily created in a queue for processing. Because implementations of some video codecs are not scale to multiple cores of advanced processors, possibility of parallel processing to achieve higher efficiency is also considered. To process multiple video sequences, an option to batch process of them was implemented into the created utility.

The created tool employs FFmpeg library and its console utilities ffmpeg and ffprobe. FFmpeg provides an unified interface for various encoders with many options and settings. The tool was created under Linux operating system and allows to record data about CPU utilization, CPU time (from scheduler data) and memory required for encoding of a video sequence. Windows platform is supported as well.

5. EQUIPMENT AND SETTINGS

The compression performance of the used video codecs was tested on a short video sequence Chimera "Aerial" (420p 10 bit, 60fps, 1200 frames) from Netflix [6].

The compression of video has been realized on a regular desktop AMD Ryzen 5 1600AF (2. gen, Zen+) processor with a locked frequency of 3.6 GHz. It has 32 GB of 3200 MT/s and DDR4 RAM. Information about the used reference implementation of video codecs is summarized in Table I. VVenC [7] and VVdeC [8] were compiled using GCC with default settings.

Example of the used command to encode the video sequence by codec AV1 is follows:

```
ffmpeg -i <input> -c:v libaom-av1 -cpu-used 5 -b:v 0 -crf 39  
<output>.mkv.
```

The angle brackets indicate the parts of command that are variable. Argument "-c:v" is used to select the encoder (see Table I), "-cpu-used" is an encoder specific argument used for libvpx-vp9 and libaom-av1 to set the level of trade-off in quality for speed (similarly, for libx264 and libx265 it is used "-preset"). For libaom-av1 and libvpx-vp9 argument "-b:v 0" is needed for unrestricted maximal bitrate while "-crf" sets the constant Quantization Parameter (QP). Next, modes Constant Quality (CQ) and Constant Rate Factor (CRF) are used. Actual bitrate in these modes is largely dependent on that how much of temporal information is in the encoded video. Codec libvpx-vp9 was employed with "-deadline 0 -aq-mode 0", where "-deadline" is used to turn off limits for time spent on encoding of a frame and "-aq-mode" is utilized to turn off adaptive quantizer (needed for VMAF metric at high quantization levels [9]). VVenC was used with preset "fast", 12 threads and with PQPA on.

Ten values of evenly distributed QP (from 9 to 63) and CRF (from 9 to 45) were used.

²<https://github.com/ondrasouk/encoders-comparison-tool>

Table I: Used software stack

Software	Version	Source
Distribution	OpenSUSE Tumbleweed	
GCC	11.2.1	
FFmpeg	4.4.1	https://ffmpeg.org/download.html#get-sources
libvpx	1.11.0	https://www.webmproject.org/code/
libaom	3.3.0	https://aomedia.google.com/aom/
libsvtav1	0.9.0	https://github.com/AOMediaCodec/SVT-AV1
libx264	0.161	https://code.videolan.org/videolan/x264
libx265	3.5	https://bitbucket.org/multicoreware/x265_git
vvenc	1.3.1	https://github.com/fraunhoferhhi/vvenc
vvdec	1.3.0	https://github.com/fraunhoferhhi/vvdec

6. RESULTS

First results (rate-distortion graphs and computational time) for video Chimera "Aerial" are shown in Fig. 1. It is visible that the best results are achieved at the using of video codecs AV1 and VVC. However, especially for AV1 and VVC, time needed for encoding (at best settings) is high, thus faster settings was chosen. The overall comparison of the used codecs using Bjontegaard-Delta (BD) method is presented in Table II. It is visible that reference libaom-av1 implementation outperforms other codecs in BD-PSNR and BD-rate, followed by VVenC and SVT-AV1.

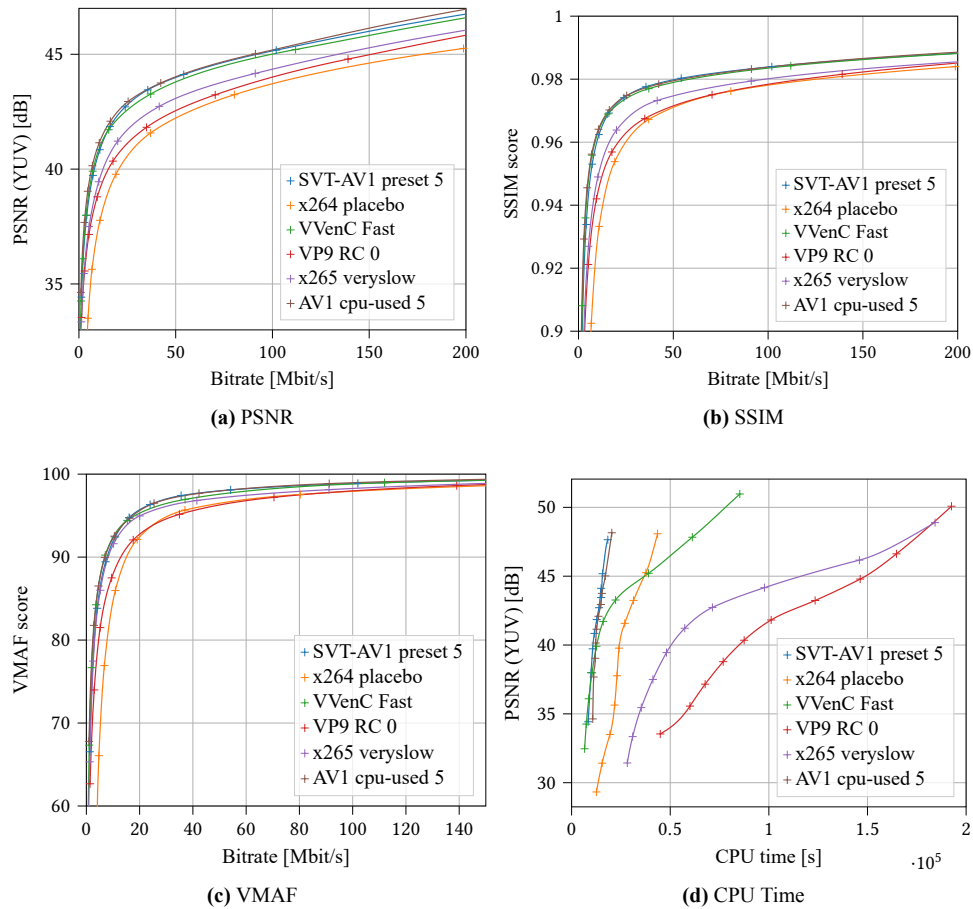
**Figure 1:** Rate-Distortion graphs and computation time for the video sequence Chimera "Aerial"

Table II: Comparison of codecs to x264 placebo

Codec and settings	BD-PSNR [dB]	BD-rate [%]
AV1 cpu-used 5	2.89	-66.00
VVenC Fast	2.46	-64.34
SVT-AV1 preset 5	2.67	-61.29
x265 veryslow	1.56	-49.17
VP9 RC 0	1.17	-32.81

7. CONCLUSION

In this short paper, attention was devoted to performance study of advanced compression algorithms HEVC, VVC, VP9 and AV1 for 4K videos. A Python-based tool employing FFmpeg multimedia framework was created to manage the whole encoding more efficiently. The visual quality of video sequences was evaluated by different objective metrics. All the results, presented in this paper, are a part of preliminary bachelor thesis.

Among others, future work plans include the extension of this study. More video sequences with different spatial and temporal information (SI and TI) will be considered. The number of the used objective quality metrics to evaluate the performance compression of video codecs will be increased. There is also planned a creation of graphical user interface for the developed tool to manage the video compression more easier.

ACKNOWLEDGMENT

This work was supported by the BUT internal project no. FEKT-S-20-6325.

REFERENCES

- [1] C. Evans, J. Issa, and S. Forrest, “The sustainable future of video entertainment, From creation to consumption,” Aug. 2020, p. 35. [Online]. Available: <https://www.interdigital.com/download/5fa0694a8934bfdf5f00596a>.
- [2] B. Bross, Y.-K. Wang, Y. Ye, *et al.*, “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, Oct. 2021, ISSN: 1558-2205. DOI: [10.1109/TCSVT.2021.3101953](https://doi.org/10.1109/TCSVT.2021.3101953). [Online]. Available: <https://ieeexplore.ieee.org/iel7/76/9556114/09503377.pdf>.
- [3] J. Han, B. Li, D. Mukherjee, *et al.*, “A technical overview of av1,” *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1435–1462, Sep. 2021, ISSN: 1558-2256. DOI: [10.1109/JPROC.2021.3058584](https://doi.org/10.1109/JPROC.2021.3058584). [Online]. Available: <https://ieeexplore.ieee.org/iel7/5/4357935/09363937.pdf>.
- [4] C. Bonnineau, W. Hamidouche, J. Fournier, N. Sidaty, J.-F. Travers, and O. Déforges, “Perceptual quality assessment of hevc and vvc standards for 8k video,” *IEEE Transactions on Broadcasting*, vol. 68, no. 1, pp. 246–253, 2022. DOI: [10.1109/TBC.2022.3140710](https://doi.org/10.1109/TBC.2022.3140710).
- [5] P. Akyazi and T. Ebrahimi, “Comparison of compression efficiency between hevc/h.265, vp9 and av1 based on subjective quality assessments,” in *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, 2018, pp. 1–6. DOI: [10.1109/QoMEX.2018.8463294](https://doi.org/10.1109/QoMEX.2018.8463294).
- [6] NETFLIX INC. “Digital video sequence ”aerial”.” (2015), [Online]. Available: https://media.xiph.org/video/derf/Chimera/Netflix_Aerial_4096x2160_60fps_10bit_420.zip. CC BY-NC-ND 4.0.
- [7] A. Wieckowski, J. Brandenburg, T. Hinz, *et al.*, “Vvenc: An open and optimized vvc encoder implementation,” in *Proc. IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2021, pp. 1–2. DOI: [10.1109/ICMEW53276.2021.9455944](https://doi.org/10.1109/ICMEW53276.2021.9455944).
- [8] A. Wieckowski, G. Hege, C. Bartnik, *et al.*, “Towards a live software decoder implementation for the upcoming versatile video coding (vvc) codec,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3124–3128. DOI: [10.1109/ICIP40778.2020.9191199](https://doi.org/10.1109/ICIP40778.2020.9191199).
- [9] I. Katsavounidis, “Dynamic optimizer-a perceptual video encoding optimization framework,” *The NETFLIX tech blog*, 2018.