RDO CANDIDATE SELECTION FOR MAXIMIZING CODING EFFICIENCY IN A PRACTICAL HEVC ENCODER

Joose Sainio, Alexandre Mercat, and Jarno Vanne
Ultra Video Group, *Tampere University*
Tampere, Finland
{joose.sainio, alexandre.mercat, jarno.vanne}@tuni.fi

ABSTRACT

High Efficiency Video Coding (HEVC) creates the conditions for economic video transmission and storage but making most of its compression potential calls for effective rate-distortion optimization (RDO) techniques in practical HEVC encoders. This paper explores the effectiveness of the following universally applicable RDO techniques: 1) rough mode decision for intra RDO candidate selection; 2) number of intra and inter RDO search candidates; and 3) accurate bit cost estimation in entropy coding. All these techniques are implemented into Kvazaar open-source HEVC encoder and altogether they improve the coding efficiency of Kvazaar veryslow preset by 5.7%, 4.0%, and 7.1% with PSNR, SSIM, and VMAF quality metrics, respectively. Even though the proposed techniques reduce the coding speed of Kvazaar to 0.64×, Kvazaar is still, on average, 2.16× as fast as the x265 encoder and attains 12.4%, 22.3%, and 4.7% better coding gain for the same PSNR, SSIM, and VMAF quality, respectively. These results let us conclude that Kvazaar is currently the leading practical open-source solution for high-quality HEVC encoding.

Index Terms— *High Efficiency Video Coding (HEVC), rate distortion optimization (RDO), practical encoding, Kvazaar HEVC encoder, video coding*

1. INTRODUCTION

Video is a ubiquitous part of our everyday life, where a plurality of trending media applications ranges from low-latency live streaming to high-fidelity *video on demand* (*VoD*) services. All these applications are behind the ever-increasing video volume, and ISO/IEC MPEG and ITU-T VCEG have released a series of international video coding standards to meet their transmission and storage needs. Currently, the landscape of these standards is dominated by the universal *Advanced Video Coding* (*AVC*/*H.264*) [1], well-established *High Efficiency Video Coding* (*HEVC*/*H.265*) [2], and emerging *Versatile Video Coding* (*VVC*/*H.266*) [3]. This work focuses on HEVC that is one of the most widespread video formats at the moment [4].

HEVC employs the classical block-based hybrid coding scheme, where the vast complexity stems from the *rate-distortion optimization* (*RDO*) process [5] that deals with the large number of block partitions. However, the HEVC standard only specifies the bitstream format, and it is up to the encoder to select the most efficient coding mode under the given complexity and bitrate restrictions. In practice, the encoder can control the coding complexity by limiting the available coding tools, the number of coding blocks, or the accuracy of the RDO. Since the coding efficiency depends on both the number of bits coded and visual quality of the compressed video, RDO has to take them both into account when selecting the optimal coding mode for each block. This is commonly solved using Lagrangian RD cost (*J*) [5] that is calculated as

$$J = D + \lambda R,$$

where *D* is the distortion, λ the Lagrangian multiplier, and *R* the bitrate. λ is used to control the ratio between bitrate and distortion. Since distortion needs to be calculated for potentially millions blocks per frame, its computational complexity needs to be controlled. The three popular methods, from least to most complex, are: 1) *sum of absolute differences* (*SAD*) or 2) *sum of absolute transformed differences* (*SATD)* between the prediction and original blocks; and 3) *sum of squared differences* (*SSD)* between the reconstruction and original blocks.

In HEVC, the video frame is divided into *coding tree units* (*CTUs*) that are usually 64×64 pixels in size [6]. The CTUs can be recursively divided down to 8×8 *coding units* (*CUs*). The CU is made up of one, two, or four *prediction units* (*PUs*) and a recursive transform tree of *transform units* (*TUs*) [6]. TUs are used to compress the residual remaining after the prediction [6]. The PUs can either be intra predicted [7] from the surrounding (already coded CUs) or inter predicted [2] from the previously coded frames. There are three inter modes in HEVC: 1) *merge mode* that selects candidates from the surrounding CUs to code the *motion vector* (*MV*) of the PU, 2) *skip mode*, which is a special case of the merge mode, where the whole CU is predicted using a single merge candidate without any residual, and 3) *MV difference*, where the MV is coded relative to a MV candidate [2], [8], [9]. Finally, HEVC employs *context adaptive binary arithmetic coding* (*CABAC*) for entropy coding [10].

Currently, there are three well-known open-source HEVC encoders: *HEVC test model* (*HM*) [11], *x265* [12], and *Kvazaar* [13], [14]. HM is the reference encoder that practically implements all HEVC coding tools. However, it does not include multithreading or comprehensive optimizations and is thereby too complex for practical deployment. Conversely, both x265 and Kvazaar are practical encoders that have several magnitudes lower complexity than HM [14], but their speedup comes at the cost of coding efficiency.

Most studies on practical encoders seek to reduce the encoding complexity of Kvazaar or x265 further [14]–[22]. However, their significant speedup over HM leaves room for pursuing coding gains even at a cost of additional complexity. In fact, some previous works have improved the coding efficiency of x265 [23]–[25] in its slowest configuration, i.e., the one that achieves the best coding efficiency. All these approaches focus on the quantization, or by extension, *rate*

This work was supported in part by the *AI for situational Awareness* (*AISA*) project led by Nokia and funded by Business Finland, and the Academy of Finland (decision no. 349216).

Table I. Kvazaar performance with different numbers of MPM and RMD candidates for 32×32 and 4×4 PUs.

		1 RMD candidate		2 RMD candidates		3 RMD candidates		4 RMD candidates	
		BD-rate	Encoding speed	BD-rate	Encoding speed	BD-rate	Encoding speed	BD-rate	Encoding speed
32×32 intra PUs									
MPM	0	2.3%	1.86×	1.3%	1.45×	0.9%	1.18×	0.6%	1.00×
	1	1.0%	1.55×	0.6%	1.29×	0.4%	1.08×	0.3%	0.94×
	2	0.3%	1.31×	0.1%	1.16×	0.1%	1.00×	0.0%	0.88×
	3	0.1%	1.09×	0.0%	1.00×	0.0%	0.90×	-0.1%	0.81×
4×4 intra PUs									
MPM	0	8.3%	1.89×	4.1%	1.48×	2.6%	1.22×	1.6%	1.03×
	1	3.8%	1.57×	1.7%	1.31×	0.8%	1.11×	0.3%	0.97×
	2	1.7%	1.29×	0.7%	1.15×	0.2%	1.01×	-0.1%	0.90×
	3	0.6%	1.07×	0.0%	1.00×	-0.3%	0.93×	-0.5%	0.83×

control (*RC*). Most notably, Bichon *et al.* [23] used temporal distortion propagation to achieve optimal *adaptive quantization* (*AQ*) with a lookahead encoder. Similarly, Liu *et al.* [24] employed lookahead to improve the RC of x265, but the complexity overhead was only reported for a single-threaded implementation. Tang *et al.* [25] used Hadamard transform based cost to estimate frame-level and block-level AQ, specifically for RC. However, their evaluations with x265 and HM showed that the AQ algorithm is heavily influenced by the underlying RC algorithm, which may not be suitable for all cases. Nevertheless, all these solutions were limited to a lookahead encoder [23], [24] or RC [25].

In this work, we propose universally applicable solution for maximizing the coding efficiency by exploring the optimal number of RDO search candidates. For intra search, we evaluate the effectiveness of *rough mode decision* (*RMD*) in RDO candidate selection and analyze the optimal number of RDO candidates depending on the size of the blocks for both luma and chroma. Similar exploration is performed on number of RDO candidates for different inter modes; most notably the significance of merge and skip modes for the overall coding efficiency. Additionally, the importance of accurate bit cost estimation is verified. To the best of our knowledge, this is the first paper that explores the number of RDO candidates in practical HEVC encoders.

The rest of the paper is structured as follows. Section 2 describes our framework for RDO parameter exploration. Section 3 introduces the proposed RDO techniques. Section 4 benchmarks the coding speed and efficiency of RD-optimized Kvazaar over those of HM and x265. Finally, Section 5 concludes the paper.

2. PARAMETER EXPLORATION FRAMEWORK

All our experiments were performed with *uvgVenctester* [26] testing framework. The coding efficiency was measured using *Bjøntegaard delta bitrate* (*BD-rate*) [27], [28] with PSNR quality metric. It yields average bitrate differences for the same PSNR so that negative BD-rate implies better coding efficiency. Computational complexity was measured as relative encoding speed, i.e., speedups less than one imply more complex configuration. The experiments were run on 22-core Intel Xeon E5-2699v4 processor, and the encoders were compiled using MSVC 14.29. The test set was composed of the 19 natural sequences in HEVC common test condition with YUV420 chroma subsampling [29].

The Kvazaar *veryslow* preset offers the highest coding efficiency, so it was used as an anchor for the parameter exploration. The experiments were carried out using *random access* (*RA*) configuration, except that *all intra* (*AI*) configuration was used in the analysis of intra PUs. *Wavefront parallel processing* and *overlapped wavefront* parallelization strategies were enabled in Kvazaar.

3. PROPOSED RDO TECHNIQUES

Currently, Kvazaar performs RDO search for two (or three for 4×4 PUs) best intra modes found in RMD, and all three *most probable modes* (*MPMs*), if they are not already among the best RMD candidates. For inter coding, only the best configuration for each PU split is checked. Most importantly, the RD cost of merge mode is merely computed if the merge mode has the best RMD cost of all 2N×2N PU modes. Furthermore, the skip mode is selected if the merge mode has no coefficients. Similarly, the RD cost of *symmetric motion partitions* (*SMP*) is computed only for the best combination of PUs. Overall, this means that in *veryslow* configuration, RDO search is carried out for up to five (six for 4×4 PUs) intra candidates and three inter candidates.

In Kvazaar, intra RDO search accounts for around 8–15% of total encoding time and inter RDO around 8–11% in RA configuration. In both cases, the majority of the computational overhead stems from prediction and quantization, whereas distortion and bit cost calculation take less than 10% of the search time in RDO.

3.1. CABAC Entropy Coding

Kvazaar only updates the CABAC contexts after completing a CTU instead of every CU. This optimization reduces complexity of faster encoder configurations and practically comes for free because their less accurate distortion calculation does not benefit from full precision in bit cost estimation. Instead, for *veryslow* preset, keeping all CABAC contexts up to date during the search has negligible effect on overall encoding complexity but it improves coding efficiency. With our test set, the average coding gain was 0.8%. Overall, this is also an important optimization to leverage impact of the other proposed techniques.

3.2. Intra Mode Decision

Table I tabulates the coding efficiency and complexity with different numbers of MPM and RMD candidates for 32×32 and 4×4 intra PUs. For 16×16 and 8×8 intra PUs, the values fall in between. The anchor configuration uses three MPM and two RMD candidates (highlighted in blue). The results are obtained in AI configuration, where only the listed PU sizes are allowed. The results indicate that larger number of RDO modes is more useful for smaller PUs, i.e., the rough search cost is less accurate with them. Overall, it is more beneficial to increase the number of MPM than RMD modes. The total number of tested modes is also reduced in cases, where the MPM mode is

Table II. Kvazaar performance with different number of intra chroma RDO candidates.

	Number of intra chroma RDO candidates			
	2	3	4	5
	Depth 1-3			
BD-rate	-0.11%	-0.11%	-0.11%	-0.12%
Encoding speed	0.94×	0.92×	0.89×	0.87×
	Depth 4			
BD-rate	-0.08%	-0.07%	-0.09%	-0.13%
Encoding speed	0.96×	0.94×	0.92×	0.91×

already included in the RDO candidates. To that end, all three MPM modes are always checked in our experiments.

Fig. 1. depicts the *rate-distortion-complexity* (*RDC*) tradeoff of selecting one to eight best RMD candidates for the RDO search plus either 1) one more for the 4×4 PUs; 2) one more for the 4×4 and 8×8 PUs; or 3) two more for the 4×4 PUs and one more for 8×8 PUs. Our results show that the cases 1) and 3) provide better RDC tradeoff over that of 2). The least complex configuration that reaches the maximum BD-rate gain of around 0.3% is selected for further experimentation; it is marked with a red line in Fig 1.

By default, Kvazaar uses a *Progressive RMD* (*PRMD*) [30] for selecting the candidates for intra search. However, in the *veryslow* preset, the intra RMD accounts for less than 2% of total encoding time. The complexity of the PRMD is about half of the complexity of a complete RMD. Disabling the PRMD improves BD-rate by 0.2% and reduces encoding speed to 0.98×. When the complete RMD is enabled, the effect of mode count used for RDO search is similar to that reported in Table I. Thus, the RMD algorithm does not give any reason why smaller PUs benefit more from a higher number of RDO candidates, but rather the RMD cost is less accurate for smaller PUs.

In addition to the luma intra mode, the chroma intra mode is selected among five modes: 1) planar; 2) DC; 3) straight vertical; 4) straight horizontal; or 5) derived from the luma mode or bottom left angular mode if the luma mode is among the first four. The chroma mode selection is improved by including the chroma mode in the luma mode selection. This adds an overhead of around 15% to the luma mode selection. There is a risk that selecting the luma mode partially based on the chroma mode reduces the effectiveness of the chroma mode search, but according to our experiments, the overall improvement outweighs the loss from the chroma mode refining.

Table II tabulates the coding efficiency and complexity changes depending on the number of RDO candidates selected for chroma mode search. The results are obtained from AI configuration when only one size of PUs is enabled, i.e., the recursive search is performed only at one depth. Based on our experiments, selecting the derived mode is always beneficial, and the rest of the modes are selected based on RMD. The results for depths 1 to 3 are similar. The depth 4 differ because the chroma PU covers the area of four 4×4 luma PUs. Depth 4 is the only one to gain a non-negligible improvement when selecting more than two modes for the RDO search. Because the speed loss is less with depth 4, we propose to check all five candidates for it and two candidates for other depths.

To conclude, the proposed optimizations to intra mode decision include:

1) increasing the number of RMD RDO candidates to seven for 4×4 PUs, six for 8×8, and five for the others;
2) disabling PRMD;
3) including chroma for luma mode decision; and
4) including chroma mode selection with all five candidates search depth 4 and two for the other depths.

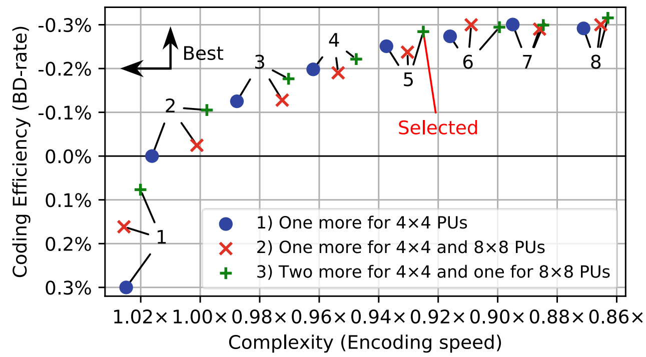

Fig. 1. RDC tradeoff for different number of RDO candidates for luma intra modes. The numbers linked to the symbols indicate the base number of best RMD candidates.

3.3. Inter Mode Decision

In general, Kvazaar splits the inter search into four stages: 1) merge candidate checking; 2) *integer motion estimation* (*IME*); 3) *fractional motion estimation* (*FME*); and 4) bidirectional prediction. All these stages use SATD as distortion metric, except for IME that employs SAD. Using SATD would increase the complexity of IME by almost tenfold for negligible to no RD gain. In general, IME compares blocks with larger differences than the other tools. For merge, FME, and bidirectional stages, the added complexity of SATD is tolerable since less than ten different candidates are checked as opposed to hundreds of candidates during IME. Therefore, no changes are proposed to distortion criteria in inter search.

The next step is to check the *zero-coefficient* (*ZC*), i.e., no residual, cost for all inter RDO candidates. In general, *rate-distortion optimized quantization* seeks to minimize the RD cost of quantized coefficients. However, it is not designed to check the cost of all zero coefficients, because inter coded CUs have a specific mechanism for signaling ZC CU. Checking the ZC cost only requires calculating the SSD between the prediction and the original block as well as calculating the cost of signaling skip mode or root coded block flag. Overall, this optimization improves BD-rate by 0.1% and reduces speed to 0.97×. More importantly, it has a compounding effect to all the following proposed optimizations. However, this is not proposed for intra PUs for two reasons: It is less likely that the intra coded PU has no residual data. Furthermore, there is no flag to indicate that the intra PU is with zero coefficients.

The following step is to check the RD cost for all non-duplicate merge candidates. As per the previous optimization, both merge and skip mode can be checked simultaneously. Since the merge mode candidate is calculated using SSD from the reconstruction, the cost for best non-merge candidate has to be calculated similarly to compare with the best merge mode. Overall, this improves BD-rate by 4.1%, but encoding speed is reduced to 0.81×.

Calculating the RD cost for the best candidate of both reference lists and the best bidirectional candidate improves BD-rate by 1.0% and slows encoding speed down to 0.92×. Checking two best candidates from both lists improves BD-rate further by 0.3%, but the encoding speed is reduced by further 0.91×. Checking the third best candidate does not further improve coding efficiency. For bidirectional prediction, adding more candidates has negligible effect on coding efficiency.

For SMP, the search is performed by finding the best candidate for both PUs and then calculating RD cost for the best PU combination. Table III tabulates the coding gain and the corresponding speed decrement when two to four best PU

Table III. Kvazaar performance with different number of SMP RDO candidates.

SMP RDO candidates	BD-rate	Encoding speed
2	-0.03%	0.94×
3	-0.07%	0.88×
4	-0.09%	0.84×

Table IV. Benchmarked encoders and command line options.

Encoder	Version	Command line options
Kvazaar anchor	5c16b50	--preset veryslow
RD-optimized Kvazaar	ddf26e6	--preset veryslow --rd 4 --intra-chroma-search --full-intra-search
HM	16.22	-c encoder_randomaccess_main.cfg
x265	3.4	--preset veryslow --tune <none/psnr/ssim>

combinations are checked. Considering the miniscule BD-rate improvement, it is not worthwhile to increase the number of RDO candidates for SMP modes.

In inter coding, chroma is always derived from the luma, i.e., the MVs for luma and chroma are identical. Indeed, chroma and luma movements are practically always identical in natural video. Therefore, chroma is excluded in rough search phases, but included in the cost calculation of the RDO candidate to maximize the coding efficiency.

To conclude, the proposed optimizations to inter mode decision include:

1) increasing the number of RDO candidates for square PUs from one to ten;
2) keeping the number of RDO candidates one for SMP; and
3) checking ZC cost for all RDO candidates.

4. COMPARISON WITH HM AND X265 ENCODERS

Table IV tabulates the encoding configurations of HM [11], x265 [12], and Kvazaar [14] used in our benchmarking. The intra period in RA condition should be one second, since both HM and Kvazaar only support static GOPs, it is set to nearest multiple of sixteen for all tested encoders. For x265, the tune setting that was most beneficial for x265 was used with each quality metric.

Table V shows the overall improvement of the proposed optimizations for PSNR, SSIM, and VMAF BD-rates and the corresponding complexity overhead over the Kvazaar anchor. The anchor tends to perform well with SSIM, so SSIM gain remains the smallest. The greatest improvement is obtained with VMAF, which is promising from the perspective of human visual system.

The encoding speed degradation is between 0.55× and 0.71×. The most significant reduction in speed takes place in *Class A* because the large resolution reduces the number of duplicate merge candidates. As the resolution shrinks, the number of duplicate merge candidates increases. However, the complexity reduction with *class D* is still higher because of the small resolution, the search is performed to large depth more often. The low complexity increase with *Class E* is because the static background allows terminating the search early, thus reducing the effect of increasing the number of RDO candidates.

Tables VI and VII report the coding efficiency and speed of the proposed RD-optimized Kvazaar over HM and x265, respectively. Even with the proposed optimizations, the coding efficiency of Kvazaar falls more than 10% behind that of HM, but Kvazaar is still up to two magnitudes faster. Using average coding speeds with *Class A* sequences as an example, encoding a two-hour 4Kp30 movie on our 22-core processor would approximately take 600 days with HM, almost 6 days with x265, and only 2.8 days with Kvazaar. The lowest speedup is obtained with *class D*, since the small resolution limits the parallelization. Compared with x265, Kvazaar is twice as fast and provides better coding efficiency in all cases except for *class A* when measured with VMAF.

Table V. Impact of the proposed RDO techniques on Kvazaar.

Class	BD-rate PSNR	BD-rate SSIM	BD-rate VMAF	Encoding Speed
hevc-A	-5.1%	-2.5%	-7.8%	0.55×
hevc-B	-6.2%	-4.4%	-8.6%	0.62×
hevc-C	-5.4%	-4.0%	-6.3%	0.71×
hevc-D	-5.9%	-4.2%	-6.4%	0.58×
hevc-E	-5.1%	-4.1%	-6.4%	0.71×
Average	**-5.7%**	**-4.0%**	**-7.1%**	**0.64×**

Table VI. Performance of RD-optimized Kvazaar over HM.

Class	BD-rate PSNR	BD-rate SSIM	BD-rate VMAF	Encoding Speed
hevc-A	13.7%	12.0%	11.4%	208.75×
hevc-B	15.6%	12.6%	12.3%	169.67×
hevc-C	13.6%	11.3%	9.9 %	71.38×
hevc-D	15.3%	10.9%	11.1%	19.89×
hevc-E	19.3%	17.5%	15.8%	174.49×
Average	**15.5%**	**12.7%**	**12.0%**	**119.69×**

Table VII. Performance of RD-optimized Kvazaar over x265.

Class	BD-rate PSNR	BD-rate SSIM	BD-rate VMAF	Encoding Speed
hevc-A	-7.9%	-20.1%	3.8%	1.91×
hevc-B	-11.3%	-21.5%	-2.7%	2.10×
hevc-C	-13.5%	-23.6%	-3.4%	2.70×
hevc-D	-9.8%	-23.4%	-1.2%	1.71×
hevc-E	-19.1%	-21.7%	-20.2%	2.33×
Average	**-12.4%**	**-22.3%**	**-4.7%**	**2.16×**

5. CONCLUSION

In this paper, we explored several RDO techniques and implemented them into practical Kvazaar HEVC encoder. Overall, the proposed techniques improved the coding efficiency of Kvazaar by 5.7%, 4.0%, and 7.1%, for the same PSNR, SSIM, and VMAF quality, respectively. Although the encoding speed slowed down to 0.64× due to these optimizations, Kvazaar remains magnitudes faster than HM and is superior to x265.

These results indicate that the proposed RD-optimized Kvazaar is able to provide the best RD tradeoff among open-source HEVC encoders. Furthermore, we believe that implementing adaptive quantization into Kvazaar could potentially increase its coding efficiency even beyond that of HM.

ACKNOWLEDGMENT

The authors would like to thank all the contributors of the Kvazaar open-source project.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Aug. 2003.

[2] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] "New 'Versatile Video Coding' Standard to Enable Next-Generation Video Compression." ITU. Sep. 2020. [Online]. Available: https://www.itu.int/en/mediacentre/Pages/pr13-2020-New-Versatile-Video-coding-standard-video-compression.aspx

[4] "Bitmovin Video Developer Report 2019." Bitmovin. 2019. [Online]. Available: https://cdn2.hubspot.net/hubfs/3411032/Bitmovin%20Magazine/Video%20Developer%20Report%202019/bitmovin-video-developer-report-2019.pdf

[5] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Proc. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[6] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697–1706, Dec. 2012.

[7] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[8] Y. Yuan *et al.*, "Quadtree based nonsquare block structure for inter frame coding in high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1707–1719, Dec. 2012.

[9] P. Helle *et al.*, "Block merging for quadtree-based partitioning in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1720–1731, Dec. 2012.

[10] V. Sze and M. Budagavi, "High throughput CABAC entropy coding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1778–1791, Dec. 2012.

[11] "HEVC Reference Software Version 16.22." [Online]. Available: https://vcgit.hhi.fraunhofer.de/jct-vc/HM/-/tags/HM-16.22 (accessed Sep. 22, 2022).

[12] "X265 HEVC Encoder/H.265 Video Codec." [Online]. Available: https://bitbucket.org/multicoreware/x265_git/src/master/ (accessed Sep. 22, 2022).

[13] "Kvazaar Open-Source HEVC Encoder." Ultra Video Group. [Online]. Available: https://github.com/ultravideo/kvazaar (accessed Sep. 22, 2022).

[14] A. Lemmetti, M. Viitanen, A. Mercat, and J. Vanne, "Kvazaar 2.0: fast and efficient open-source HEVC inter encoder," in *Proc. ACM Multimedia Syst. Conf.*, Istanbul, Turkey, May 2020.

[15] A. Lemmetti, A. Koivula, M. Viitanen, J. Vanne, and T. D. Hämäläinen, "AVX2-optimized Kvazaar HEVC intra encoder," in *Proc. IEEE Int. Conf. Image Process.*, Phoenix, Arizona, USA, 2016.

[16] A. Mercat, A. Lemmetti, M. Viitanen, and J. Vanne, "Acceleration of Kvazaar HEVC intra encoder with machine learning," in *Proc. IEEE Int. Conf. Image Process.*, Taipei, Taiwan, 2019.

[17] Q. Hu, X. Zhang, Z. Gao, and J. Sun, "Analysis and optimization of x265 encoder," in *Proc. IEEE Visual Commun. Image Process. Conf.*, Valletta, Malta, Dec. 2014 .

[18] S. Yin, X. Zhang, and Z. Gao, "Efficient SAO coding algorithm for x265 encoder," in *Proc. IEEE Visual Commun. Image Process. Conf.*, Singapore, Dec. 2015.

[19] Y. C. Lin, J. J. Wu, and K. H. Chen, "CU partition prediction scheme for x265 intra coding using neural networks," in *Proc. Int. Conf. Control Robot. Cybern.*, Tokyo, Japan, Sep. 2019.

[20] F. Mu, H. Zhang, X. Liu, and Z. Zhang, "Fast inter mode decision algorithms for x265," in *Proc. IEEE Inf. Technol. Mechatronics Eng. Conf.*, Chongqing, China, Dec. 2018.

[21] F. Takano, H. Igarashi, and T. Moriyoshi, "4K-UHD real-time HEVC encoder with GPU accelerated motion estimation," in *Proc. IEEE Int. Conf. Image Process.*, Beijing, China, Sep. 2017.

[22] Y. Zhao, L. Song, X. Wang, M. Chen, and J. Wang, "Efficient realization of parallel HEVC intra encoding," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, San Jose, California, USA, Oct. 2013.

[23] M. Bichon, J. Le Tanou, M. Ropert, W. Hamidouche, and L. Morin, "Optimal adaptive quantization based on temporal distortion propagation Mmodel for HEVC," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5419–5434, Nov. 2019.

[24] Z. Liu, L. Wang, X. Li, and X. Ji, "Optimize x265 rate control: an exploration of lookahead in frame bit allocation and slice type decision," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2558–2573, May 2019.

[25] M. Tang, X. Chen, J. Wen, and Y. Han, "Hadamard transform-based optimized HEVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 827–839, Mar. 2019.

[26] J. Sainio, A. Mercat, and J. Vanne, "uvgVenctester: open-source test automation framework for comprehensive video encoder benchmarking," in *Proc. ACM Multimedia Syst. Conf.*, Istanbul, Turkey, Jun. 2021.

[27] G. Bjøntegaard, *Improvements of the BD-PSNR Model*, document VCEG-AI11, TU-T SG16/Q6, Berlin, Germany, Jul. 2008.

[28] *Working Practices Using Objective Metrics for Evaluation of Video Coding Efficiency Experiments*, document ITU-T HSTP-VID-WPOM and ISO/IEC DTR 23002-8, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 2020.

[29] F. Bossen, *Common test conditions and software reference configurations*, document JCTVC-L1100, Geneva, Switzerland, Jan. 2013.

[30] H. Zhang and Z. Ma, "Fast intra mode decision for high efficiency video coding (HEVC)," *IEEE Trans. on Circuits Syst. Video Technol*, vol. 24, no. 4, pp. 660–668, Apr. 2014.