A Feature Based Complexity Model for Decoder Complexity Optimized HEVC Video Encoding

Thanuja Mallikarachchi, Dumidu S. Talagala, Hemantha Kodikara Arachchi, and Anil Fernando
Centre for Vision Speech and Signal Processing, University of Surrey, United Kingdom.
Email: {d.mallikarachchi,d.talagala,h.kodikaraarachchi,w.fernando}@surrey.ac.uk

***Abstract*—The complexity of the novel video compression algorithms is a major contributor for the increased demand of processing and energy resources for video playback in consumer electronic devices. Therefore, a decoder complexity reduction mechanism is proposed which constitutes a model that predicts the decoder's complexity requirements to decode the HEVC encoded bit streams with a 4.2% average prediction error and a decoder complexity optimized encoding algorithm, which reduces the decoding complexity by an average of 28.06% and 41.19% with a -1.91 dB and -2.46 dB impact to the BD-PSNR for the *low delay P* and *random access* configurations, respectively.**

I. Introduction

The proliferation of High Definition (HD) and Ultra-High Definition (UHD) video formats, increased complexity of the state-of-the-art video compression algorithms such as High Efficiency Video Coding (HEVC) [1], [2] and the unprecedented growth in mobile video data (expected to reach three-fourth of the overall mobile data traffic by 2019 [3]), have become crucial factors in making video playback in Consumer Electronic (CE) devices a resource intensive operation. In this context, the limited availability of processing and energy resources in hand-held CE devices (smart phones, tablets, etc.,) present a bottleneck when it comes to multimedia playback. Thus, improving the energy efficiency of these devices has become a compelling challenge that must be overcome. That said, the energy consumed by the multimedia playback is highly correlated with both the complexity of the codec as well as the content. Therefore, a video encoding framework that minimizes the resource utilization of the decoders could positively contribute towards enhancing the energy efficiency of these CE devices.

Reducing the complexity of the decoder, and by extension the device's energy consumption, has received significant attention in the recent literature. However, the energy-aware media delivery mechanisms that involve proxy servers [4], and media transcoding techniques [5] are only limited to the dynamic variation of the Quantization Parameter (QP), spatial resolution, and frame rate. A more complex power-aware HEVC streaming solution which generates bit streams with various complexity levels has been proposed by He *et al.* [6]. Moreover, Nogues *et al.* [7] introduced modifications to the in-loop filtering process and the interpolation filters at the decoder to further reduce the decoder's complexity. In general, these methods, as well as Dynamic Voltage and Frequency Scaling (DVFS) [8] solutions typically suffer from video quality artifacts due to the skipping of decoder operations and frame drop encounters, respectively. Therefore, mechanisms to generate encoded bit streams which minimize the extensive resource utilization at the decoders are envisioned to achieve both complexity reductions while maintaining the perceived video quality intact. Crucially, such frameworks require an accurate and detailed complexity estimation model that predicts the processing requirements of the decoder. The state-of-the-art models for the complexity estimation include approaches such as predicting the decoding energy of a bit stream using the decoder processing time [9], and the mapping of the relationships between the decoder complexity, content and the QP [10]. Moreover, Herglotz *et al.* introduce two such models [11], [12] that estimate the HEVC decoding energy for intra- and inter-coded frames, respectively. However, the level of detail considered in these models is found to be inadequate when estimating the decoding complexity at the Coding Unit (CU) level within the traditional Rate-Distortion (RD) optimized encoding chain.

To this end, this paper introduces a novel feature based decoder complexity model that predicts the decoding complexity of an inter-coded CU. Thereafter, a decoder complexity optimized coding mode selection algorithm, which leverages the proposed complexity model to exploit the diverse decoder complexities associated with the novel coding modes and features in HEVC, is introduced to generate video bit streams that minimize the decoder's processing and energy requirements.

This work was supported by the ACTION-TV project, which is funded under European Commission's 7th Framework Program (Grant number: 611761).

II. HEVC Decoder Complexity Modeling

This section first describes the process of formulating the proposed decoder complexity model and is followed by a description of the process of profiling the decoder complexity.

A. *Formulating the Decoder Complexity Model*

The process of reconstructing a CU at the decoder consists of two phases; the decoding phase (i.e., entropy decoding of syntax and residual coefficients) and the decompression phase (i.e., predicting and reconstructing the block based on decoded information). In this context, considering the most significant functions (illustrated in Fig. 1), the decompression complexity

TABLE I
ESTIMATED CPU CYCLES OF THE DECOMPRESSION OPERATIONS IN THE HEVC DECODER.

e_{fc} (per 8×8 PU)		e_{lf} (per 8×8 PU)	e_{cf} (per 8×8 PU)	e_{wp} (per 8×8 PU)	e_{rc} (per 8×8 CU)	e_{it} (per 4×4 TU)	$e_{overhead}$ (per 8×8 CU)	
Uni	Bi						$2N \times 2N$	Other
626	1389	14069	1957	16365	11552	6029	160645	259782

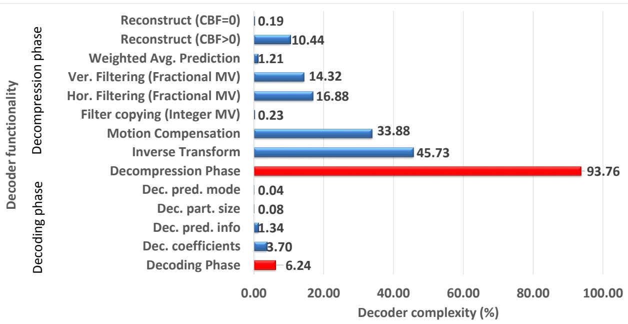

Fig. 1. Distribution of decoder complexity of the *low delay P* encoded Kimono HD at QP=22 using the HM reference decoder.

for an inter predicted CU can be expressed as,

$$E_d = \sum_{i=1}^{N} \Big\{ e_{fc}(i) + e_{lf}(i) + 2 \times e_{cf}(i) + e_{wp}(i) \Big\} + \sum_{j=1}^{M} e_{it}(j) + e_r, \tag{1}$$

where e_{fc}, e_{lf}, e_{cf}, e_{wp}, e_{it} and e_r correspond to the complexities in the decoder for the filter copying (i.e., for integer-pel motion vectors), luma filtering, chroma filtering, weighted average prediction, inverse transform and reconstruction operations, respectively. Moreover, N and M identify the number of Prediction Units (PU) and Transform Units (TU) that constitute the CU.[1]

The complexity of the decoding phase is dominated by the resources consumed for the residual coefficient decoding. Thus, the decoding complexity of processing the j^{th} CU is given by,

$$E_{CU}(j) = E_d + e_{coeff} + \frac{e_{overhead}(j)}{nCU(j)}. \tag{2}$$

Here, e_{coeff} denotes the complexity incurred when decoding the transform coefficients, and $e_{overhead}$ encompasses the overhead generated due to the quadtree structure of the CTU that results from the j^{th} CU for the selected PU size. nCU represents the total number of CUs of the same size as the j^{th} CU that could reside within a CTU. Hence, the decoding complexity of a CTU, which comprises of inter predicted CUs, (E_{CTU}) becomes the summation of $E_{CU}(j)$ values and can be expressed as

$$E_{CTU} = \sum_{j=1}^{K} E_{CU}(j), \tag{3}$$

where K is the total number of CUs that make up the CTU.

[1]The e_{lf} and e_{cf} terms need to be considered twice when both vertical and horizontal filtering are used in a PU. In addition, e_{it} becomes non-zero only when a TU comprises of non-zero coefficients.

B. *Profiling the Complexity of the HEVC Decoder*

In order to estimate the complexity of the decoding operations, this work profiles the decoder for a range of test sequences (see Table II for details) with the commonly used instruction level profiling tools [13] that have been used extensively for the purpose of complexity analysis [14]. Here, the decoding complexity of an inter-coded CU with respect to the HM reference decoder is analyzed, and is presented in terms of the number of CPU cycles (with respect to an Intel x86 reference system with a 3.4 GHz CPU, 8GB RAM) consumed, which represent the relative complexity of each operation.

The decoder complexity of the decompression operations predominantly depend on the CU, PU and TU sizes. Thus, the estimated CPU cycles are averaged and presented in Table I as a base value with respect to the smallest unit of operation considered (i.e., the smallest CU, PU, and TU size). The decoder complexity of e_{fc}, e_{lf}, e_{wp} and e_{cf} are presented as a function of PU size. In this context, the complexity of an arbitrary PU in a given CU can be derived by appropriately scaling the given CPU cycles to the required PU size.

Furthermore, the complexity associated with the transform coefficient decoding, e_{coeff}, can be modeled as a linear function of the number of non-zero transform coefficients [10]. In addition, the decoder complexity model and the profiling details for intra-predicted CUs are derived based on the information presented in [15].

III. DECODER COMPLEXITY OPTIMIZED ENCODING

Traditionally, the HEVC reference encoder has followed a RD optimization process in order to select the coding modes and coding structures for a given content. The Lagrangian cost function evaluated therein can be expressed as

$$\min \; D(p) + \lambda R(p) \;\Big|\; p \in P_k, \tag{4}$$

where $\lambda \geq 0$ is the Lagrange multiplier, p is a particular coding parameter combination in the set of all the possible coding options P_k, and $D(p)$, $R(p)$ are the distortion and rate associated with the selected set of coding parameters, respectively. The coding mode and structure that returns the minimum cost for a given content is selected as the coding parameter combination that gives the best coding efficiency. However, due to the varying distribution of decoder complexities, the coding parameters selected via (4), may still utilize a significant amount of resources during decoding. Therefore, generating a decoder complexity optimized bit stream requires the selection of less complex coding modes during the RD optimization phase.

TABLE II
ACCURACY OF THE COMPLEXITY ESTIMATION MODEL

	Low Delay P (LDP)		*Low Delay B* (LDB)	
	Sequence	$P_e(\%)$	Sequence	$P_e(\%)$
Training Set	Bridge-far	6.13	Bridge-far	4.58
	Waterfall	0.44	Waterfall	3.60
	Band HD	5.96	Band HD	4.33
	Kimono HD	4.84	Kimono HD	4.25
	GT Fly	3.52	GT Fly	2.94
	Average	**4.17**	**Average**	**3.94**
Validation Set	Coastguard	3.30	Coastguard	2.43
	Container	6.86	Container	1.36
	Beergarden HD	4.84	Beergarden HD	4.55
	Dancer HD	4.75	Dancer HD	4.32
	Cafe HD	5.04	Cafe HD	5.88
	Musicians HD	2.40	Musicians HD	8.47
	Average	**4.53**	**Average**	**4.50**

In this context, the complexity models for inter predicted and intra predicted CUs introduced in Sec. II and in [15], respectively, are capable of estimating the required decoding complexity for a particular coding mode, which could be used in the Lagrangian optimization. Therefore, the modified cost function in the proposed algorithm can be expressed as,

$$\min\ E(p)\ \big|\ p \in P_k, \tag{5}$$

where, $E(p)$ represents the decoder's complexity cost for the coding parameter set p. In this context, the minimum decoding complexity becomes the selection criteria for the PU mode, CU size, TU size, motion vectors (e.g., integer-pel vs fractional-pel motion vectors), etc. for a particular QP.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results are first presented for the validation of the proposed complexity estimation model, followed by a discussion of the performance of the proposed complexity optimized coding mode selection algorithm.

A. Validating the Proposed Complexity Model

The decoding complexity predicted by the proposed model (E_p), is compared with the complexity that is measured directly from the profiling tools (E_a) [13], and the prediction error, P_e is calculated as

$$P_e = 100 \times \frac{|E_a - E_p|}{E_a}. \tag{6}$$

The experimental results illustrated in the Table II reveal that the proposed model, which demonstrates an average prediction error less than 5% for both uni- and bi-predicted video frames (for 50 frames averaged over the QPs 22, 27, 32, and 37), is essentially a suitable complexity estimation model to be used in decoder complexity optimized video coding applications.

B. Performance of the Proposed Encoding Algorithm

The performance of the proposed algorithm is compared with two similar state-of-the-art algorithms in terms of the complexity reduction and the quality impact. In this context, the decoder's complexity reduction is computed using

$$\Delta E = 100 \times \frac{|E_\rho - E_{HM}|}{E_{HM}}, \tag{7}$$

where E_ρ and E_{HM} are the estimated decoder complexities for the bit streams generated by the proposed and other state-of-the-art algorithms, and the bit streams generated by the HM reference encoder [16], respectively. Moreover, the quality impact is measured using the Bjøntegaard Delta-Peak Signal-to-Noise Ratio (BD-PSNR) [17]. The performance is analyzed and presented for the test sequences encoded using the *low delay P* and *random access* configurations for 50 frames averaged over the QPs 22, 27, 32, and 37. All the bit streams are decoded using the HM reference decoder [16] in an Intel x86 system with a 3.4 GHz CPU and 8GB RAM.

The algorithm proposed by Nogues *et al.* [7] constitutes a process that alters the interpolation filters (MC), and an algorithm that skips the loop filtering process (LF) based on the required complexity level. The experimental results presented in the Tables III and IV correspond to the lowest complexity level for which the decoder applies the respective algorithms for all the video frames in the sequence. However, despite the average 28% decoder complexity reduction, the reconstructed sequences demonstrate a considerable loss in BD-PSNR. The changes to the interpolation filters at the decoder, which are unaware to the encoder, causes prediction artifacts resulting in the aforementioned quality drop. The algorithm proposed by He *at al.* [6] utilizes an energy optimized PU mode selection (PUM), and a distortion-energy optimized de-blocking filter decision algorithm (DBLK) to achieve bit streams with various complexity levels (low, medium, high), where the presented results correspond to the lowest level. The PUM mode is constrained to the PU mode decision and motion estimation vs merge mode decision, hence, the level of complexity reduction achieved is in the range of 16.4%.

In contrast, the proposed complexity optimized encoding algorithm utilizes the decoding complexity estimated from a detailed and accurate model which results in a less complex coding structure while the motion estimation yield a motion vector for the PU that reduces the overall complexity of the CU. Hence, the resulting bit streams are shown to consume less amount of decoder complexity compared to the state-of-the-art algorithms. Moreover, the usage of the LF in [7], which is mutually exclusive to the proposed CU level mode decisions, further increases the decoder's complexity reduction. However, the proposed algorithm only considers the decoding complexity for the mode selection, hence results in a less RD optimized bit stream.

V. CONCLUSION

In conclusion, it is evident that encoders could generate decoder resource optimized video bit streams by exploiting the diversities of the decoder complexity requirements of the HEVC coding modes. In this context, the proposed complexity model for HEVC inter-frame decoding predicts the decoding complexity with an average prediction error less than 5% for both uni- and bi-predicted frames. Furthermore, the proposed encoding algorithm is capable of generating HEVC bit streams that can achieve an average decoder complexity reduction of 28.06% and 41.19% with a BD-PSNR loss of -1.91 dB and

TABLE III

COMPLEXITY REDUCTION PERFORMANCE OF THE PROPOSED METHOD (LOW DELAY P).

	Sequence	Proposed (Model)		Proposed (Model + LF [7])		He *et al.* [6] (PUM)		He *et al.* [6] (PUM+DBLK)		Nogues *et al.* [7] (MC+LF)	
		$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR
Training Set	Bridge-far	12.90	-0.42	31.54	-1.96	3.28	-0.02	12.10	-0.08	19.16	-1.30
	Waterfall	26.97	-2.59	44.74	-3.61	10.42	-1.72	16.67	-2.12	22.42	-14.05
	Band HD	26.86	-1.46	49.46	-3.31	8.67	-0.82	11.21	-2.74	28.55	-7.47
	Kimono HD	39.68	-1.89	60.00	-5.14	17.35	-0.61	17.35	-0.64	31.31	-12.63
	GT Fly	46.48	-2.15	62.4	-3.66	29.07	-1.2	34.09	-1.24	43.63	-11.47
Validation Set	Coastguard	18.15	-1.64	43.29	-5.16	12.59	-0.51	14.70	-0.54	25.34	-12.50
	Container	18.99	-2.06	38.99	-3.22	8.41	-0.77	17.65	-0.84	16.77	-9.20
	Beergarden	24.14	-1.58	45.19	-3.83	9.66	0.44	11.66	-2.05	26.80	-12.90
	Dancer	40.66	-2.51	59.52	-3.73	21.96	-1.02	25.33	-1.13	40.76	-11.58
	Cafe	17.32	-1.55	42.59	-3.09	7.52	-0.62	14.30	-0.69	27.90	-10.68
	Musicians	36.54	-3.19	62.00	-6.57	15.85	-1.90	15.90	-1.98	36.74	-14.90
	Average	**28.06**	**-1.91**	**49.06**	**-3.93**	**13.16**	**-0.79**	**17.36**	**-1.27**	**29.03**	**-10.78**

TABLE IV

COMPLEXITY REDUCTION PERFORMANCE OF THE PROPOSED METHOD (RANDOM ACCESS).

	Sequence	Proposed (Model)		Proposed (Model + LF [7])		He *et al.* [6] (PUM)		He *et al.* [6] (PUM+DBLK)		Nogues *et al.* [7] (MC+LF)	
		$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR	$\Delta E(\%)$	BD-PSNR
Training Set	Bridge-far	33.22	-1.69	45.80	-1.42	2.79	-0.01	7.18	-0.03	15.64	-0.22
	Waterfall	39.10	-2.89	48.68	-3.19	13.89	-0.54	18.36	-0.56	30.09	-6.53
	Band HD	41.59	-2.65	55.32	-3.00	6.03	-0.54	9.47	-0.56	21.72	-1.81
	Kimono HD	48.20	-2.32	62.16	-5.10	18.07	-0.58	18.60	-0.59	32.71	-2.69
	GT Fly	52.80	-3.20	64.20	-4.20	22.48	-0.75	27.40	-0.75	37.26	-5.50
Validation Set	Coastguard	34.94	-4.13	47.33	-4.51	12.53	-0.49	13.76	-0.50	24.66	-4.68
	Container	36.54	-1.91	46.40	-2.29	8.08	-0.28	12.95	-0.31	19.42	-5.67
	Beergarden	36.60	-2.13	49.30	-2.68	7.11	-0.33	9.59	-0.34	23.08	-3.91
	Dancer	49.50	-0.54	61.10	-3.25	18.75	-0.49	22.50	-0.58	35.60	-5.66
	Cafe	36.00	-1.99	49.80	-3.60	4.63	-0.23	9.55	-0.22	18.54	-1.49
	Musicians	44.60	-3.71	58.26	-5.40	18.12	-3.17	19.48	-3.18	37.44	-9.38
	Average	**41.19**	**-2.46**	**53.48**	**-3.51**	**12.04**	**-0.67**	**15.34**	**-0.69**	**26.92**	**-4.32**

-2.46 dB for *low delay P* and *random access* configurations, respectively, compared to the bit streams generated by the HM reference encoder. The future work will focus on extending the framework to consider both the rate and the distortion, along side the decoder's complexity, to generate more optimized HEVC video bit streams.

REFERENCES

[1] G.J.Sullivan, J. 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.

[2] F. Bossen, B. Bross, S. Karsten, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[3] Cisco, "Cisco visual networking index: global mobile data traffic forecast update 2014-2019," cisco, White Paper.

[4] R. Sjoberg, Y. Chen, A. Fujibayashi, M. M. Hannuksela, J. Samuelsson, T. K. Tan, Y. K. Wang, and S. Wenger, "Overview of HEVC high-level syntax and reference picture management," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1858–1870, Dec 2012.

[5] P. Shenoy and P. Radkov, "Proxy-assisted power-friendly streaming to mobile devices," in *Proc. Multimedia Computing and Networking Conference.*, 2003, pp. 177–191.

[6] Y. He, M. Kunstner, S. Gudumasu, E.-S. Ryu, Y. Ye, and X. Xiu, "Power aware HEVC streaming for mobile," in *Proc. IEEE Visual Communications and Image Processing (VCIP)*, Kuching, Malaysia, Nov. 2013, pp. 1–5.

[7] E. Nogues, S. Holmbacka, M. Pelcat, D. Menard, and J. Lilius, "Power-aware HEVC decoding with tunable image quality," in *Proc. IEEE workshop on Signal Processing Systems.*, Belfast, Noth. Ireland, Oct 2014, pp. 1–6.

[8] E. Kim, H. Jeong, J. Yang, and M. Song, "Balancing energy use against video quality in mobile devices," *IEEE Trans. Consumer Electron.*, vol. 60, no. 3, pp. 517–524, Aug 2014.

[9] C. Herglotz, Walencik, and A. Kaup, "Estimating the HEVC decoding energy using the decoder processing time," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, May 2015, pp. 513 – 516.

[10] T. Mallikarachchi, H. K. Arachchi, D. Talagala, and A. Fernando, "CTU level decoder energy consumption modelling for decoder energy-aware hevc encoding," in *Proc. IEEE International Conference on Consumer Electronics*, Las Vegas, USA, Jan. 2016.

[11] C. Herglotz, D. Springer, A. Eichenseer, and A. Kaup, "Modeling the energy consumption of HEVC intra decoding," in *Proc. IEEE International Conference on Systems, Signals, and Image Processing*, Bucharest, Romania, Jul. 2013, pp. 91–94.

[12] C. Herglotz, D. Springer, and A. Kaup, "Modeling the energy consumption of HEVC P- and B-frame decoding," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014, pp. 3661–3665.

[13] "The valgrind quick start guide," valgrind Documentation.

[14] F. Saab, I. H. Elhajj, A. Kayssi, and A. Chehab, "Profiling of HEVC encoder," *Electronics Letters.*, vol. 50, no. 15, pp. 1061–1063, July 2014.

[15] T. Mallikarachchi, H. K. Arachchi, D. Talagala, and A. Fernando, "Decoder energy-aware intra-coded HEVC bit stream generation," in *Proc. IEEE International Conference on Multimedia and Expo*, Seattle, USA, Jul. 2016.

[16] "HEVC reference software- HM-16.0," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0/.

[17] G. Bjontegarrd, "Calculation of average PSNR differences between RD-curves," *ITU - Telecommunications Standardization Sector STUDY GROUP 16 Video Coding Experts Group (VCEG)*, 2001.