Decoding-Complexity-Aware HEVC Encoding Using a Complexity-Rate-Distortion Model

Thanuja Mallikarachchi, *Student Member, IEEE*, Dumidu S. Talagala, *Member, IEEE*, Hemantha Kodikara Arachchi, *Member, IEEE*, and Anil Fernando, *Member, IEEE*

***Abstract*—The energy consumption of Consumer Electronic (CE) devices during media playback is inexorably linked to the computational complexity of decoding compressed video. Reducing a CE device's the energy consumption is therefore becoming ever more challenging with the increasing video resolutions and the complexity of the video coding algorithms. To this end, this paper proposes a framework that alters the video bit stream to reduce the decoding complexity and simultaneously limits the impact on the coding efficiency. In this context, this paper (i) first performs an analysis to determine the trade-off between the decoding complexity, video quality and bit rate with respect to a reference decoder implementation on a General Purpose Processor (GPP) architecture. Thereafter, (ii) a novel generic decoding complexity-aware video coding algorithm is proposed to generate decoding complexity-rate-distortion optimized High Efficiency Video Coding (HEVC) bit streams. The experimental results reveal that the bit streams generated by the proposed algorithm achieve 29.43% and 13.22% decoding complexity reductions for a similar video quality with minimal coding efficiency impact compared to the state-of-the-art approaches when applied to the HM16.0 and openHEVC decoder implementations, respectively. In addition, analysis of the energy consumption behavior for the same scenarios reveal up to 20% energy consumption reductions while achieving a similar video quality to that of HM 16.0 encoded HEVC bit streams.**

***Index Terms*—Complexity-rate-distortion, decoding complexity, decoding energy, energy minimization, HEVC**

I. Introduction

THE ever-increasing consumption of High Definition (HD) and Ultra High Definition (UHD) video contents and the proliferation of mobile media consumption habits in end-users, are making video playback on resource constrained Consumer Electronic (CE) devices (e.g., smartphones, tablets etc.,) increasingly necessary and challenging [1]. In fact, the actual resource consumption of video decoding is tightly coupled with the complexity of the video content as well as the compression format. Therefore, the adoption of high resolution video contents and complex video coding standards such as the High Efficiency Video Coding (HEVC) [2][3] substantially affect the energy usage of a CE device.

Traditionally, energy reductions in video decoding devices are achieved by either improving the efficiency of the radio receiver interface, modifying the decoder architecture and decoding operations, or by modifying the media content to reduce the complexity of the decoding process [1]. The latter being in the domain of video coding, consists of simplistic approaches that alter the basic coding parameters such as the Quantization Parameter (QP), frame resolution, frame rate. [4][5]. More state-of-the-art solutions manipulate the motion compensation and in-loop filtering operations in HEVC [6][7] or adopt Dynamic Voltage and Frequency Scaling (DVFS) [8] – [11] techniques to reduce the decoder's power consumption. However, the state-of-the-art methods in the literature do not exploit the variations of the computational complexity that exist between different decoding operations to determine the optimum coding parameters at the encoder itself.

In this context, this paper proposes a novel encoding algorithm that exploits the relationship between decoding complexity, rate and distortion to derive trade-off coefficients for the rate and decoding complexity at a given QP. The proposed algorithm advances the state-of-the-art by determining the whole spectrum of coding modes (i.e., HEVC quadtree structure, prediction modes, motion vectors and transform decisions etc.,) required to encode a given content by minimizing the decoding complexity, while balancing its impact on the coding efficiency. Thus, the experimental results reveal that the bit streams generated by the proposed algorithm achieve a significant decoding complexity and energy reduction for a similar video quality to that of the HM 16.0 encoded bit streams with a minimal bit rate increase, compared to the state-of-the-art methods.

The remainder of this paper is organized as follows. An overview of the state-of-the-art is presented in Sec. II, followed up by a comprehensive analysis on the decoding complexity, rate and distortion parameters and the proposed encoding algorithm in Sec. III. Finally, Sec. IV and V present the experimental results and the concluding remarks along with potential future work, respectively.

II. Background and Related Work

The relationship between a CE device's energy consumption and the many factors that affect it (e.g., the

The manuscript received on August 25, 2017. Revised December 4, 2017.

The authors T. Mallikarachchi, D.S. Talagala, and A. Fernando are with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom. (e-mail: d.mallikarachchi@surrey.ac.uk;d.talagala@surrey.ac.uk;w.fernando@surrey.ac.uk).

H. Kodikara Arachchi is with the Nottingham Trent University, 50 Shakespeare Street, Nottingham, NG1 4FQ, United Kingdom. (e-mail: hemantha.kodikaraarachchi@ntu.ac.uk).

complexity of the content, video coding algorithm, communication protocols and technologies, hardware architecture, etc.) have resulted in research focused on reducing the power consumption on all layers of the IP stack. Yet, they can be broadly categorized into two areas [1]; solutions that operate on the physical and link layer protocols, and those that operate in the application layer. The former attempt to reduce the energy consumed in communication activities, whereas the latter attempt to reduce the complexity of processing the content being consumed. The focus of this work is on the application layer, i.e., adapting the content during the encoding process, and thereby reducing the decoding complexity of the HEVC coded video bit stream. Hence, the following discussion focuses on the state-of-the-art approaches relevant to the second category.

The energy consumption in Complementary Metal-Oxide-Semiconductor (CMOS) circuits exhibits a linear relationship with the Central Processing Unit (CPU) clock frequency [12]. Therefore, exploiting the relationship between computational complexity, execution time and clock frequency, the energy consumed per decoding operation can be mapped to a quadratic relationship to the operation's computational complexity [12] for a given decoder architecture. Thus, simplifying the decoding operation, and thereby reducing the device's energy consumption has been attempted on numerous occasions. The Green-MPEG initiative by Moving Picture Experts group (MPEG) is one of the recent developments that standardizes green meta-data [13], which can be used to reduce the decoding complexity and tweak display parameters to reduce the device's energy consumption [13]. Other recent developments constitute structural modifications at the data- and task-level for parallelized decoder implementations to support real-time decoding of high resolution, high frame rate HEVC bit streams [14]. Furthermore, the utilization of Just-In-Time adaptive decoder engine [15], and OpenMP and actor-based dataflow models [16] have resulted in energy-aware HEVC decoder implementations over the last few years.

In a diverging approach, the work in [7] proposes the use of simplified in-loop and interpolation filters during motion compensation. Here, in-loop filtering is skipped to suit the desired level of complexity. In addition, 7- and 8-tap luma and 4-tap chroma filters in HEVC are reduced to 3-tap luma and 1-tap chroma filters, respectively. Along the same vein, the algorithmic level approximate computing applied for energy efficient HEVC decoding [17] reduces decoding complexity, but in common with [7], suffers from severely compromised video quality due to the modified interpolation filters. As such, in general, decoder modifications can lead to two drawbacks; incompatibility with or irrelevance to existing decoder implementations, and the degradation of video quality.

In contrast to decoder modifications, DVFS seeks to achieve energy reductions by maintaining the minimum required CPU frequency and voltage level that satisfies the decoding complexity demands. In these algorithms, energy use is balanced with respect to the video quality [8] – [10]. In general, these methods estimate and modify the operating frequency of the processor for the subsequent frames based on the decoding complexities of the preceding frames. Their operating principle is similar to that of Linux ondemand governor [18], but the frequency selection is solely governed by the decoder's operational complexity. The drawbacks of aggressive DVFS algorithms are frame drops and an impact on the overall system performance for which the general purpose devices may adversely affect the user's quality of experience.

A third approach to reducing the decoding energy consumption is dynamic content adaptation. This generally entails the adoption of scalable video coding architectures that use proxy servers [19], media transcoding [20], or dynamic adaptive streaming technologies [21]. However, these as well as device oriented [4] and battery-aware [5] adaptive multimedia delivery schemes are typically restricted to manipulating basic video coding parameters such as QP, spatial resolution, frame rate and scalable bit streams [22] to adapt video content to achieve energy savings. In fact, although energy-aware HEVC streaming solutions [6] do exist, they are limited to prediction mode and motion vector selection. As a result, diverse coding features available in the more modern coding standards remain unexploited, and the approach itself can suffer from variability of the perceived video quality with time. Overall, it can be observed that the state-of-the-art approaches (to reduce the energy consumption) do not alter how the bit stream itself is created, but instead focus on mitigating the effects of complex decoding operations after the fact. Furthermore, the preparation of a less complex bit stream at the encoder will retain the applicability of other decoder energy reduction strategies while allowing further energy consumption reductions.

III. Decoding Complexity – Rate – Distortion Analysis For Encoding

In order to consider the decoding complexity, together with rate and distortion during the encoding phase, the encoder must be aware of the decoding complexity of operations for all coding parameter combinations. Thus, detailed and accurate modelling of the decoding operation complexity is crucially important. To similar ends, the state-of-the-art techniques have exploited high-level complexity analysis of decoding operations [23], energy estimation based on decoding time [24], and mapping of decoding energy to the content and QP [25]. Yet, the level of details in these models is inadequate for a Coding Unit (CU) level decoding complexity estimation.

In general, the energy consumption of a decoder depends on a number of factors that are both architecture and implementation dependent (instruction set, memory management, CPU load balancing, voltage and frequency levels, etc.). Yet, with respect to a given architecture, the energy consumed when decoding the video bit stream is tightly coupled to the computational complexity [9]-[12] of the decoding operations. Hence, this coupling can be used to indirectly reduce the energy consumed to decode a bit stream during the video encoding process itself. To this end, a decoding complexity profiling for individual decoding operations relating to the HEVC coding modes and features has been carried out in our previous works [27][28] using open source instruction level profiling tools [34]. In this context, the computational complexities in terms of CPU cycles identified in [27][28] are embedded within HM16.0 implementation to make the encoder aware of the relative complexities of the decoding operations (The CU level decoding complexity estimation models in [27] and [28] for intra- and inter-prediction, respectively, have been verified to predict the decoding complexity within the encoder with only < 5% prediction error). The encoder now possesses the resulting bit rate, distortion and decoding complexity for a particular coding mode and QP of a given content which can then be used to form the decoding complexity, rate and distortion analysis, as described next.

A. Decoding complexity, rate and distortion analysis and trade-off

The selection of a coding mode and a structure that is appropriate in terms of decoding complexity and coding efficiency requires an in depth analysis of the impact of various coding parameters in a range of situations. In this context, prior investigations carried out in [29] and [30] define the relationship between the bit rate and distortion. A similar analysis and a defined relationship between the decoding complexity, rate and distortion with respect to HEVC coding parameters is crucial to develop a comprehensive model that facilitates decoding complexity-aware video encoding.

The HEVC encoder typically adopts a Rate-Distortion (RD) optimization process to determine the optimum coding modes for a given content. In this context, the minimization cost function used for coding mode selection can be expressed as,

$$\min_{p\in P} J_{RD} \;\Big|\; J_{RD} = D(p) + \lambda_r R(p), \tag{1}$$

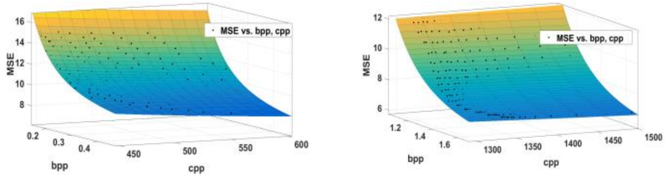
where *p* is a particular coding parameter combination in the set of all possible coding parameters *P*, and *D(p)* and *R(p)* are the distortion and rate associated with *p*, respectively. Here, $\lambda_r \geq 0$ denotes the Lagrangian multiplier that trades-off the distortion for the bit rate of a particular coding mode. The relationship between *D* and *R* has been extensively studied and a QP dependent relationship for λ_r is defined in [29].

In contrast, for the proposed encoding scheme, decoding complexity is introduced as another constraint requiring a modified Lagrangian cost function which constitute both bit rate and a decoding complexity as constraints. In this case, the modified cost function which is used for coding mode selection in the encoder is expressed as,

$$\min_{p\in P} J_{CRD} \mid J_{CRD} = D(p) + \lambda_\rho R(p) + \lambda_c C(p) \tag{2}$$

where *C(p)* is the decoding complexity associated with the coding parameter set *p*, and $\lambda_\rho \geq 0$ and $\lambda_c \geq 0$ are the bit rate and decoding complexity trade-off parameters, respectively. Determining the appropriate values for λ_ρ and λ_c in (2) now becomes crucial for the optimization of the encoding algorithm. To this end, the experimental approach adopted in this work is based on and builds upon the empirical observations presented in the following subsections.

B. Decoding complexity - rate - distortion space



(a) CRD space: inter-prediction (b) CRD space: intra-prediction

Fig. 1. An illustration of CRD space that exist between the decoding complexity (cpp), rate (bpp), and distortion (MSE) parameters, for different combinations of λ_ρ and λ_c at QP=25. Each point on the surface corresponds to a particular parameter combination of λ_ρ and λ_c.

The relationship that exists among the decoding complexity, rate and distortion (i.e., the CRD space) is both complex and content dependent. In order to visualize and understand this parameter space, the behavior of *C*, *R* and *D* for multiple video contents was analyzed (50 frames of 3 HD and 3 Common Intermediate Format (CIF) sequences that exhibit diverse motion and texture characteristics were encoded and analyzed) for multiple $QP \in \{0,\Lambda,51\}$ and combinations of $\lambda_\rho \in [0,\infty)$ and $\lambda_c \in [0,\infty)$. As an example, the resulting decoding complexity, rate and distortion values in terms of complexity per pixel (cpp), bits per pixel (bpp) and Mean Square Error (MSE) are illustrated for a particular video content in Fig. 1. Here, each observed point in the CRD space corresponds to a set of coding modes, selected for a particular content and QP, for an arbitrary combination of λ_ρ and λ_c values. As such, the selection of combinations of λ_ρ and λ_c to be used in the encoder's optimization function in (2) boils down to an engineering decision; i.e., selecting an appropriate trade-off between decoding complexity, rate and distortion for a specific requirement. The decision criteria and the process adopted for selecting generic values for λ_ρ and λ_c that were deemed appropriate in this work are described next.

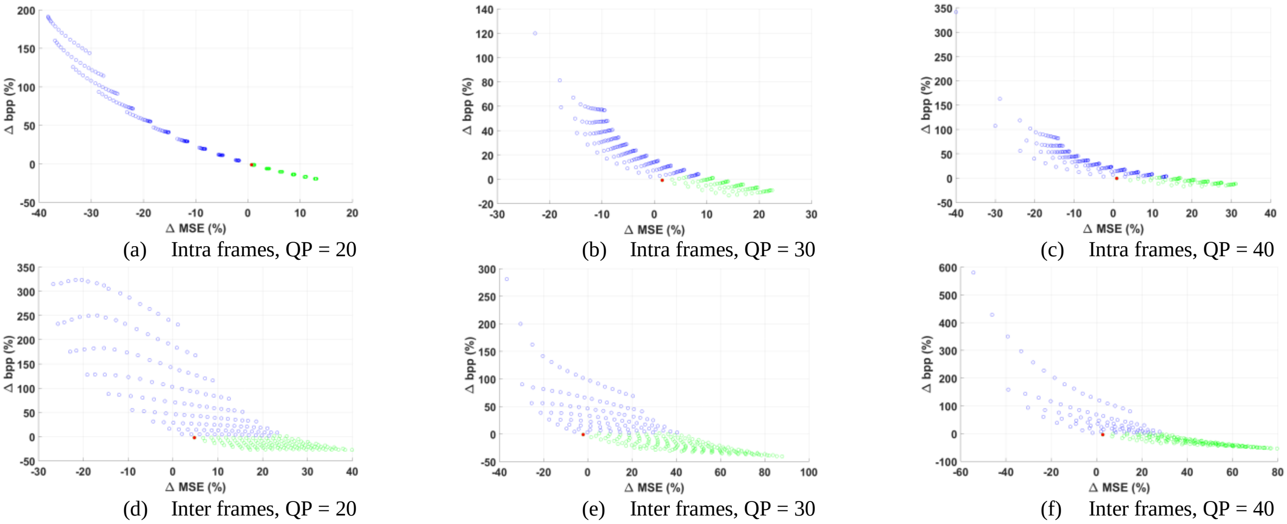
C. Determining an appropriate and generic λ_ρ and λ_c

In order to determine a suitable operating point in the CRD parameter space, the rate, distortion and decoding complexity obtained using different combinations of λ_ρ and λ_c are compared with those values obtained when using the traditional Lagrangian cost function in (1). To facilitate this analysis, first the percentage differences of each parameter, i.e., ΔR, ΔD and ΔC given by,

$$\Delta\Gamma = 100 \times \frac{\Gamma_{CRD} - \Gamma_{RD}}{\Gamma_{RD}} \tag{3}$$

is computed. Here, Γ represents the distortion *D*, bit rate *R* and decoding complexity *C*, while Γ_{RD} and Γ_{CRD} correspond to those same parameters obtained when using the cost functions in (1) and (2), respectively.

Fig. 2 illustrates the distribution of ΔR and ΔD for a particular sequence and a selection of QPs. Each data point corresponds to the deviation of the operating point in CRD space (when using (2) as the mode selection cost function), i.e., a unique λ_ρ and λ_c combination, with respect to the traditional RD optimized operating point (when using (1) as the mode selection cost function). Here, the differences in the behavior for different frame types and λ_ρ and λ_c pairings can be observed. Moreover, it can also be observed that distortion,



(a) Intra frames, QP = 20 (b) Intra frames, QP = 30 (c) Intra frames, QP = 40

(d) Inter frames, QP = 20 (e) Inter frames, QP = 30 (f) Inter frames, QP = 40

Fig. 2. The distribution of ΔD and ΔR for different combinations of λ_ρ and λ_c value pairs for "Kimono 1080p" sequence at three sample QP values. Each point represent the deviation (%) of rate, and distortion of the proposed algorithm from that of the RD optimized mode selection. The "green" points represent the subset of operational points that satisfy the criteria ΔR ≤ 1%. The "red" highlighted point corresponds to the selected operational point that gives the minimum ΔD within the subset of "green" highlighted data points.

for example deviates significantly from the RD optimized value for some λ_ρ and λ_c combinations. Therefore, selecting an appropriate λ_ρ and λ_c becomes a matter of the preferred trade-off of each parameter against the other.

The following approach is adopted in this work to constrain the impact on the coding efficiency and to achieve a decoding complexity reduction, as per the objectives outlined in the introduction. To this end, in the empirical analysis of the data obtained for the many combination of λ_ρ and λ_c in Sec. III-B, the appropriate design constraints are enforced to obtain the relevant Lagrangian multiplier parameter combination. Thus, a constraint is first placed on the bit rate such that $\Delta R \leq 1\%$. Thereafter, from the subset of λ_ρ and λ_c combinations that satisfy this criteria, the operating point, i.e., λ_ρ and λ_c combination, that minimizes ΔD is derived (the operating point selected in this manner for the Kimono 1080p sequence in Fig. 2 is highlighted in red). It should be noted that the reduction in decoding complexity achieved here is governed by the coding efficiency trade-off defined above, and a different set of constraints will naturally result in another λ_ρ and λ_c combination and different performance. Moreover, it is observed that the λ_ρ and λ_c combination that satisfy the aforementioned criteria is both QP and content dependent; thus, a set of generic values for λ_ρ and λ_c are obtained by averaging of the individual optimized parameters of the 6 test sequences in Sec. III-B. The final generic values for λ_ρ and λ_c are given by:

$$\lambda_c\big|_{generic} = \begin{cases} 2.53431\times10^{-5}.e^{0.1327QP} & Intra-frame \\ 0.34825\times10^{-3}.e^{0.03029QP} & Inter-frame \end{cases} \tag{4}$$

and

$$\lambda_\rho\big|_{generic} = \begin{cases} 0.95\times\lambda_r & Intra-frame \\ 1.01\times\lambda_r & Inter-frame \end{cases}. \tag{5}$$

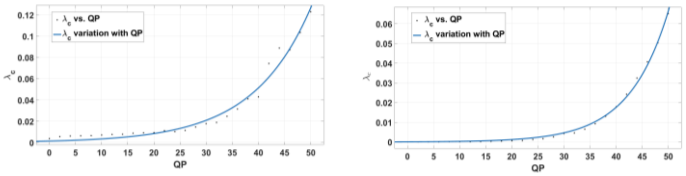
In this case, λ_r is the QP dependent Lagrangian multiplier defined in [29]. Fig. 3 graphically illustrates the variation in λ_c across the range of QPs 0 – 51 for both inter- and intra-predicted frames.

IV. Experimental Results and Discussion

This section presents the performance of the proposed algorithm, in terms of the decoding complexity, power consumption and the impact on the coding efficiency.

A. Simulation environment

The proposed algorithm is implemented in the HM 16.0 reference software [31], where the complexity models presented in [27][28] perform the decoding complexity estimations and the proposed Lagrangian cost function in (2) determines the coding modes for both inter- and intra-prediction (Fig. 4). The resulting bit streams are decoded using the HM 16.0 [31] and openHEVC [32] software decoders on a system (8GB RAM, with 9 CPU frequency steps ranging from 759 MHz – 1600MHz) running the Linux kernel 4.10, and the system's inbuilt Graphics Processing Unit (GPU) based hardware decoder. The algorithm's performance is compared



(a) Inter-frames (b) Intra-frames

Fig. 3. The generic λ_c behavior with respect to the QP and frame type.

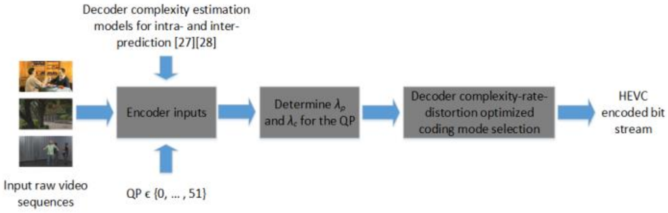

Decoder complexity estimation models for intra- and inter-prediction [27][28]
Input raw video sequences
Encoder inputs
QP ϵ {0, ... , 51}
Determine λρ and λc for the QP
Decoder complexity-rate-distortion optimized coding mode selection
HEVC encoded bit stream

Fig. 4. A high level illustration of the proposed encoding algorithm. Once λ_ρ and λ_c are determined, the coding mode selection is performed using (2).

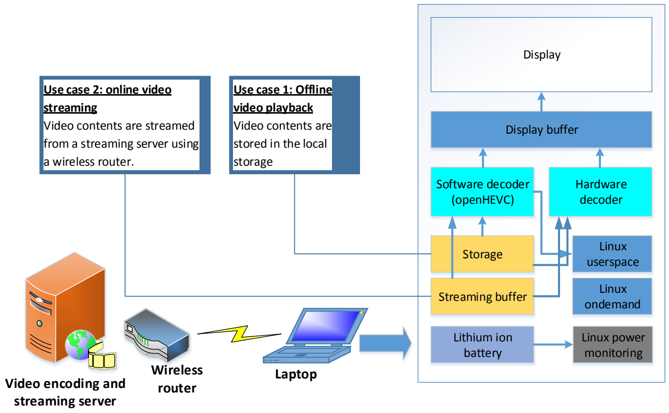

Display
Display buffer
Software decoder (openHEVC)
Hardware decoder
Storage
Linux userspace
Streaming buffer
Linux ondemand
Lithium ion battery
Linux power monitoring
Use case 2: online video streaming
Video contents are streamed from a streaming server using a wireless router.
Use case 1: Offline video playback
Video contents are stored in the local storage
Video encoding and streaming server
Wireless router
Laptop

Fig. 5. A graphical illustration of the simulation environment. The two use cases considered in this work are indicated as use case 1 (offline video playback) and use case 2 (online video streaming).

with two state-of-the-art approaches; a tunable HEVC decoder proposed by Nogues *et al.* [7] and the power-aware encoding algorithm proposed by He *et al.* [6]. The CIF and HD video sequences used in this experiments are encoded using random access configuration with QPs 22, 27, 32, and 37. The complexities of the decoding processes are measured using the open source instruction-level analysis tools callgrind/valgrind [33]. Finally, the decoding energy consumption is determined by measuring the energy dissipated by the system during playback. A high level overview of this experimental setup is graphically illustrated in the Fig. 5.

B. *Evaluation metrics*

The performance of the proposed and state-of-the-art algorithms is evaluated by the measuring the decoding complexity reduction achieved by the different bit streams. To this end, percentage decoding complexity reduction achieved for the same video quality to that of the reference given by BD-C is evaluated by utilizing the Bjøntegarrd Delta-Bit Rate (BD-BR) calculation specified in [34] and by considering the area under the decoding complexity, distortion curve [35][36]. Similarly, the device's percentage energy consumption reduction for the same video quality given by BD-E is evaluated by utilizing the energy dissipated when decoding the bit streams generated by the HM reference encoder and any other algorithm, with Peak Signal-to-Noise Ratio (PSNR) as the quality metric. Finally, the impact on the coding efficiency is measured in terms of BD-BR [34], which illustrates the impact on the bit rate for the same resultant video quality.

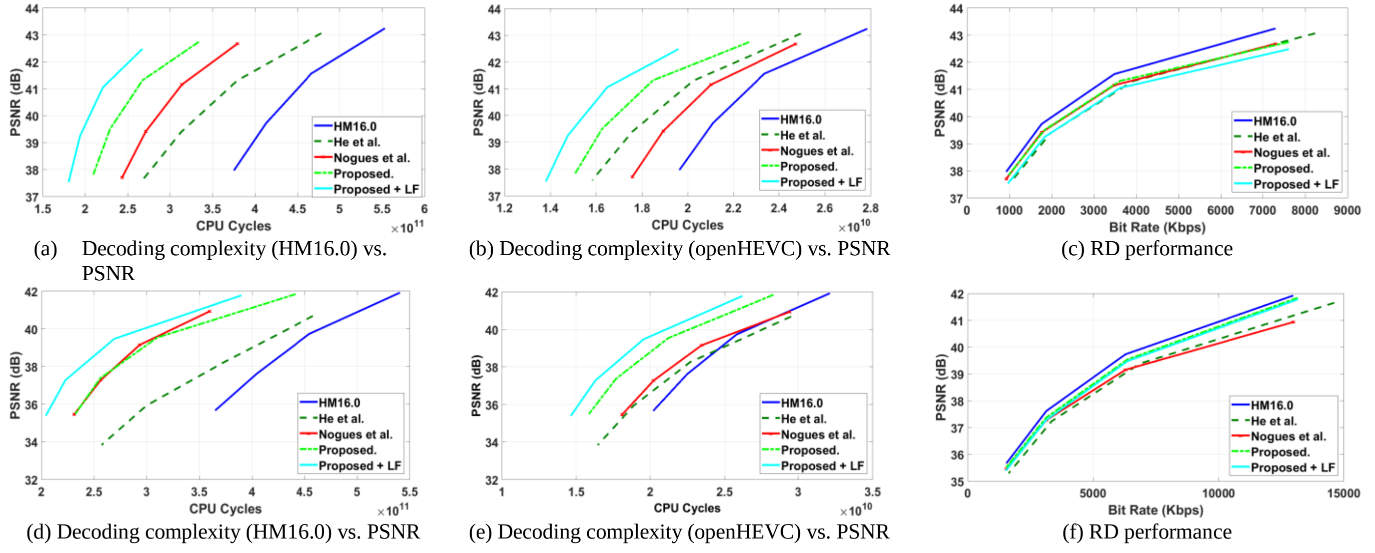
C. *Performance evaluation and analysis*

This section initially discusses the complexity reduction performance of the proposed method with respect to state-of-the-art decoding complexity reduction techniques, and thereafter investigates its potential energy savings with respect to the voltage-frequency scaling approaches in different application scenarios.

1) *Comparison with modified decoder implementations:*

Modifications of the motion compensation filters in the decoder (MC) and the intermittent skipping of the loop filter (LF), proposed by Nogues *et al.* [7] contributes significantly to reduce the complexity of the decoding operations (ref. BD-C results in Table I). However, this impacts visual quality considerably due to the distortions introduced by the modified motion compensation filtering operations. For example, the reduced filter sizes in [7] results in a different predicted image than that is used by the encoder to calculate the motion vectors for the Prediction Unit (PU). Hence, this partially filtered PU now gets compensated with a somewhat incorrect residual, which in turn distorts the reconstructed PU. Furthermore, the propagation of these errors to future frames further impacts the visual quality of the video as a whole. Although, the intra-refresh in the random access configuration marginally limits the impact of error propagation, these distortions nevertheless result in an increased BD-BR (ref. Fig. 6(c), 6(f)).

Moreover, as illustrated in the Table I, the impact on quality would be content dependent when the decoding operations are altered in this fashion, especially since the distortions would be significant in complex video sequences with high motion



(a) Decoding complexity (HM16.0) vs. PSNR

(b) Decoding complexity (openHEVC) vs. PSNR

(c) RD performance

(d) Decoding complexity (HM16.0) vs. PSNR

(e) Decoding complexity (openHEVC) vs. PSNR

(f) RD performance

Fig. 6. The decoding complexity (in CPU cycles) – distortion curves and rate-distortion curves for the "kimono" (top row), and "parkscene" (bottom row) sequences illustrating the relative performance of the proposed and state-of-the-art techniques.

TABLE I
DECODING COMPLEXITY REDUCTION PERFORMANCE IN THE RANDOM ACCESS CONFIGURATION

Sequence	Proposed (model only)			Proposed (model + LF [7])			He *et al.* [6] (PUM + DBLK)			Nogues *et al.* [7] (MC + LF)		
	BD-C* (%)	BD-C† (%)	BD-BR (%)	BD-C* (%)	BD-C† (%)	BD-BR (%)	BD-C* (%)	BD-C† (%)	BD-BR (%)	BD-C* (%)	BD-C† (%)	BD-BR (%)
Akiyo	-27.3	-9.8	5.5	-38.0	-15.7	11.7	-7.0	-6.6	9.5	-16.5	-9.5	6.4
Waterfall	-19.6	-12.6	5.6	-25.1	-16.6	7.2	-15.9	-9.1	14.9	-29.4	-6.4	18.4
Container	-21.7	-8.7	2.7	-29.0	-12.7	11.2	-10.2	-6.8	5.9	-16.0	-8.4	10.9
Coastguard	-17.2	-9.9	4.5	-23.3	-13.2	15.0	-12.3	-4.9	21.3	-25.1	-5.7	14.6
Band	-34.0	-12.3	7.7	-43.3	-18.9	18.2	-3.6	-4.5	40.1	-20.5	-9.6	16.2
Beergarden	-25.8	-11.4	2.5	-35.7	-16.3	11.1	-7.1	-6.4	7.0	-21.4	-9.3	11.1
Café	-36.6	-12.2	5.8	-46.6	-18.2	13.3	-13.4	-9.4	9.9	-20.6	-10.4	12.4
Dancer	-34.9	-17.9	9.7	-43.7	-23.7	13.0	-18.3	-10.7	15.4	-33.1	-7.1	15.5
GT Fly	-40.0	-18.5	9.1	-51.1	-26.1	12.9	-17.5	-6.3	50.5	-34.8	-8.7	35.0
Kimono	-38.6	-20.3	6.7	-45.4	-26.1	14.0	-19.8	-13.5	16.5	-32.2	-8.9	15.9
Musicians	-34.5	-16.5	9.9	-44.1	-23.4	17.4	-16.3	-6.4	40.8	-33.5	-7.0	25.9
Parkscene	-31.4	-17.3	7.3	-40.0	-23.5	10.3	-19.6	-11.3	14.4	-34.3	-7.2	18.4
Poznan street	-32.6	-12.4	2.7	-33.7	-18.6	11.3	-12.6	-9.6	6.5	-20.4	-10.9	11.8
BasketDrill	-26.7	-10.7	9.1	-37.1	-16.9	17.0	-2.7	-1.2	26.6	-22.9	-7.6	21.0
BasketPass	-20.6	-7.9	8.3	-31.4	-13.4	13.1	-7.8	-0.2	38.3	25.1	-5.7	29.7
Average	**-29.43**	**-13.22**	**6.47**	**-37.83**	**-18.88**	**13.11**	**-12.27**	**-7.12**	**21.17**	**-25.72**	**-8.16**	**17.54**

* BD-C (%) achieved when using HM 16.0 reference decoder.
† BD-C (%) achieved when using openHEVC decoder.

and textured content (e.g., "musicians" and "coastguard" vs. "container" and "Poznan street"). However, the proposed method in contrast, shows a negligible change in BD-BR compared to the method proposed by Nogues *et al.* [7]. This is due to the proposed algorithm operating at the encoder-side which determines the type of the motion vector (integer-pel vs. fractional-pel) based on the optimization cost function in (2); thus, requiring no changes to the decoding process itself.

Skipping the loop filter (LF) on the other hand, as in [7], reduces the decoding complexity with minimal impact on video quality and can also be implemented when decoding the proposed bit stream. For example, the experimental results presented in the Table I illustrate the BD-C improvements that can be achieved for the proposed algorithm in this manner. Here, the de-blocking and the Sample Adaptive Offset (SAO) filters are skipped by the decoder based on the complexity level specified. In this case, the performance of the proposed method can be improved, albeit for an additional BD-BR increase of 6.47%.

2) *Comparison with power-aware encoding mechanisms:*

The encoding algorithm proposed by He *et al.* [6] attempts to reduce the complexity of the filtering operations during motion compensation and the de-blocking operation performed by the decoder. In this context, the energy optimized motion vector selection algorithm (PUM) and the de-blocking filter disabling algorithm (DBLK) produce a bit stream which demonstrates a moderate complexity reduction as seen in the Table I and decoding complexity, distortion curves in Fig. 6. In comparison to [7], a much higher BD-BR loss is observed, especially for the sequences with high motion and complex texture properties. Here, although the motion vector and PU mode decisions are made at the encoder, the selection of the trade-off factors do not consider the impact of both rate and distortion which significantly affects the coding efficiency. Hence, despite the 12% and 7% BD-C reduction achieved by the algorithm, its applicability is limited due to the bit rate increase required to achieve similar quality to the HM 16.0 encoded bit streams. Furthermore, the lack of the detailed decoding complexity model and the QP agnostic trade-off factor selection in [6] results in a poor BD-C reduction and increased loss of coding efficiency. In addition, the algorithm by He *et al.* [6] requires the communication of the de-blocking filter decisions to the decoder, which requires additional overhead as either multiple Picture Parameter Set (PPS) Network Abstraction Layer (NAL) units or metadata must be exchanged between the encoder and decoder.

In contrast, the proposed algorithm demonstrates considerable improvements in decoding complexity reduction with the minimal impact of a BD-BR increase to 6.47% on average and delivers BD-C reductions of 29.43% and 13.22% for the HM16.0 and openHEVC decoders, respectively, for a similar video quality compared to HM encoded bit streams. This is aided by the use of more detailed and accurate HEVC decoding complexity estimation models [27][28] which yield more accurate decoding complexity estimations for the decoding complexity rate distortion optimization in (2).

3) *Energy consumption behavior-offline video playback:*

Next, the overall energy consumed when decoding the bit streams generated by the proposed method is investigated and compared to those of the HM 16.0 encoder. In this case, the bit streams are stored within the mobile device and are decoded in real time using openHEVC [32] software decoder (Use case 1 in Fig. 5). They are displayed on screen for 20 minutes simulating an offline video playback use case on a mobile device. The energy consumption during the whole decoding and playback process is measured in terms of the reduction in battery capacity via Linux's power measurement tools. The energy consumed for each QP (22, 27, 32, 37) is recorded and together with decoded stream's PSNR is used to calculate BD-E which represents the energy consumed to achieve the same video quality as the reference HM 16.0 generated bit streams. The energy reduction under these conditions with different

TABLE II
ENERGY CONSUMPTION BEHAVIOR OF THE ENCODING ALGORITHMS IN THE RANDOM ACCESS CONFIGURATIONS

Sequence	Proposed + ondemand	HM 16.0 + DVFS [11]	Proposed + DVFS [11]	He *et al.* [6] + DVFS [11]	Nogues *et al.* [7] + DVFS [11]	Proposed + LF + ondemand	Proposed + LF + DVFS [11]
	BD-E (%)	BD-E (%)	BD-E (%)	BD-E (%)	BD-E (%)	BD-E (%)	BD-E (%)
Band	-9.2	-10.3	-15.4	-17.3	-4.7	-13.0	-22.0
Beergarden	-5.1	-12.5	-15.5	-18.7	-3.7	-8.5	-22.6
Café	-4.5	-5.9	-12.8	-14.0	-1.6	-8.0	-17.0
Dancer	-3.5	-9.7	-13.5	-16.0	-1.7	-8.5	-21.0
GT Fly	-2.7	-8.4	-11.6	-12.6	-1.3	-8.0	-19.1
Kimono	-5.3	-7.1	-13.8	-16.7	-1.9	-7.0	-21.2
Musicians	-3.8	-8.8	-16.0	-14.5	-2.1	-7.8	-20.7
Parkscene	-7.1	-10.4	-17.8	-18.4	-4.0	-14.1	-22.3
Poznan street	-2.3	-8.7	-10.7	-10.3	-4.0	-7.7	-18.2
Average	**-4.83**	**-9.08**	**-14.12**	**-15.38**	**-2.77**	**-9.17**	**-20.45**

BD-E is expressed with respect to the energy consumed to decode HM 16.0 encoded bit streams using openHEVC software decoder and Linux's ondemand frequency governor.

DVFS schemes and bit streams is reported in Table II.

Next, the impact of using a more sophisticated DVFS scheme is investigated. In this context, the dynamic frequency selection algorithm proposed by Raffin *et al.* [11] has been integrated in the openHEVC decoder. Here, the operating frequency of the processor is controlled based on the estimated complexity of the subsequent frame (assessed using the moving weighted average of the complexities of previously decoded frames). Therefore, the selection of the CPU frequency becomes application and content specific, i.e., in this context the decoder and the current bit stream. The energy consumption behavior shown in Table II and graphical illustrations in Fig. 7 for the bit streams emphasize how an application specific DVFS governor can indeed outperform a generic DVFS governor. However, as was the case before, the increased complexity of the HM 16.0 bit streams limits the potential energy savings that can be achieved. In fact, the complexity reduction by the proposed algorithm's bit stream allows the DVFS algorithm to select much lower CPU frequencies that lead to greater energy savings. Hence, an improvement of -5.04% (-14.12% vs -9.08%) BD-E reduction when compared to the HM 16.0 bit stream with DVFS is observed for the proposed algorithm. Moreover, as illustrated in Fig. 7 and Table II, the BD-E reduction of the proposed algorithm when utilized with a decoder that skips the loop filtering process (similar to the algorithm proposed in [7]), is on average -20.45%. This suggests that the video playback devices can reduce the energy consumption by approximately 20% by decoding the bit streams generated by the proposed algorithm and by skipping the de-blocking filters; a significant decoding energy reduction for a similar quality to that of the HM encoded bit streams, when considering the software based HEVC decoder implementations.

TABLE III
ENERGY CONSUMPTION† BEHAVIOR OF THE PROPOSED ALGORITHM

sequence	Software decoding		Hardware decoding
	Offline playback BD-E (%)	Online streaming BD-E (%)	BD-E (%)
Band	-9.2	-5.8	-1.6
Beergarden	-5.1	-2.1	-1.6
Café	-4.5	-1.8	-1.5
Dancer	-3.5	-5.0	-3.2
GT Fly	-2.7	-2.5	-1.2
Kimono	-5.3	-4.2	-1.3
Musicians	-3.8	-3.4	-1.1
Parkscene	-7.1	-2.9	-1.9
Poznan street	-2.3	-4.9	-3.3
Average	**-4.83**	**-3.62**	**-1.85**

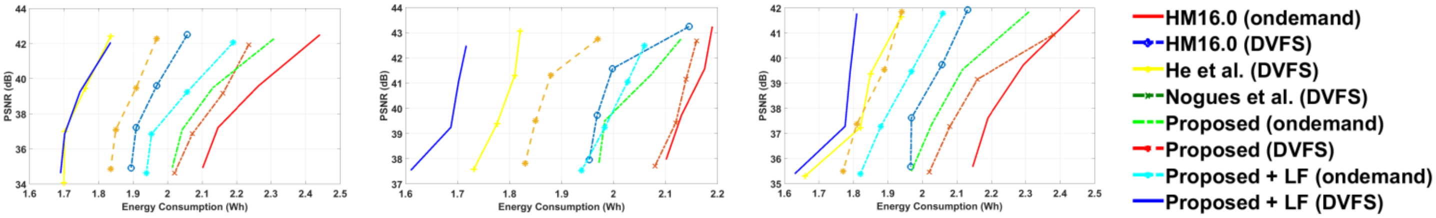
†BD-E is expressed with respect to the energy consumed to decode the HM16.0 reference encoder's bit streams when using the openHEVC decoder and Linux's ondemand frequency governor.

4) *Energy consumption behavior-other use cases:*

In this subsection, we discuss the energy consumption behaviour of the proposed algorithm for two other video playback scenarios in addition to the off-line playback scenario analyzed in the previous subsections. First, consider the decoding energy consumption behaviour in Table III when using the proposed algorithm and a GPU based hardware HEVC decoder for the off-line video playback scenario. In this case, CPU and GPU clock frequencies can be maintained at their minimums while the GPU is managed by the hardware itself. Crucially, the results demonstrate that the proposed algorithm outperforms HM 16.0 in terms of the decoding energy consumption by -1.85% to achieve a similar video quality. This is attributed to the reduced complexity of the bit streams generated by the proposed algorithm, which increases the GPU's idle time, which in turn causes the BD-E reduction observed[1]. That being said, the absolute power consumption will still depend on the power management policy of the GPU driver, inter-process communications, etc., and since the efficient management of these resources are outside the scope of the this work, the results presented here correspond to system's default GPU power management settings in the processor architecture.

Finally, the proposed algorithm exhibited a BD-BR increase of 6.47% (in Table I) due to no attempt being made to strictly control the bit rate. Naturally, this raises the question of what impact the increased bit rate would have on the energy consumption of the proposed method in a use case such as on-line video streaming (use case 2 in Fig. 5). Thus, this too was investigated for QPs 22, 27, 32, and 37 and the results are presented in Table III. Here, the bit stream is streamed over a 802.11n wireless link to be decoded by the openHEVC software decoder using the Linux ondemand DVFS governor.

[1] The remaining state-of-the-art algorithms require modifications to the decoder implementations which are not feasible for hardware implementation. Hence, the experimental results for the hardware HEVC decoder is presented solely for the proposed algorithm.



(a) Beergarden (b) Kimono (c) Parkscene

Fig. 7. The decoding energy consumption with respect to distortion for the proposed and state-of-the-art algorithms. Here, the energy consumed to decode the streams is presented as the total reduction of the device's battery capacity during video playback.

The total energy consumed in the process (including consumption by the wireless interface) is measured and reported in terms of BD-E in Table III. The results illustrate a -3.62% BD-E decoding energy reduction is achieved by the proposed algorithm for a similar video quality to that of HM 16.0 encoded streams. It is therefore evident that the decoding energy reduction via proposed method exceeds the increased energy consumed during transmission for the somewhat greater bit rate that results from the proposed approach. In this context, the results suggest that the proposed encoding algorithm has potential benefits in content preparation for both off-line and on-line video playback and streaming scenarios; a crucial improvement compared to the state-of-the-art in energy-efficient multimedia content preparation and distribution mechanisms.

V. Conclusion

Reducing the complexity of the encoded bit streams is seen as a potential application layer solution for the increased energy demands in mobile video playback. In this context, this paper proposes a decoding complexity-aware video coding algorithm which makes use of a comprehensive decoding complexity, rate and distortion analysis to determine the QP dependent generic trade-off factors for the three parameters involved in the new mode selection cost function. The proposed encoding algorithm considers the overall impact of the three parameters to determine the optimum trade-off between the coding efficiency and decoding complexity, when selecting a particular coding mode. Thus, the HEVC bit streams generated by the proposed algorithm results in a higher decoding complexity and energy reduction (up to 20.45%) for a similar video quality to that of HM16.0 encoded bit streams with minimal coding efficiency impact compared to state-of-the-art approaches.

The future work will focus on developing a joint energy and rate controlled video encoding algorithm for video streaming applications that serve resource constrained mobile devices.

References

[1] M. A. Hoque, M. Siekkinen, and J. K. Nurminen, "Energy efficient multimedia streaming to mobile devices a survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 579–597, Feb. 2014.

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

[3] 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.

[4] Z. Longhao, R. Trestian, and G. –M. Muntean, "eDOAS: Energy-aware device-oriented adaptive multimedia scheme for wi-fi offload," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC),*Istanbul, Turkey, Apr. 2014, pp. 2916–2921.

[5] M. Kennedy, H. Venkataraman, and G. Muntean, "Battery and streamaware adaptive multimedia delivery for wireless devices," in *Proc. IEEE Conference on Local Computer Networks (LCN),* Denver, CO, USA, Oct. 2010, pp. 843–846.

[6] Y. He, M. Kunstner, S. Gudumasu, E. S. Ryu, Y. Ye, and X. Xiu, "Power aware HEVC streaming for mobile," in *Proc. IEEE International Conference on Visual Communications and Image Processing*, Kuching, Malaysia, Nov. 2013, pp. 2-6.

[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, North Ireland, Oct. 2014, pp. 1-6.

[8] W. Y. Liang, M. F. Chang, Y. L. Chen, C. F. Lai, Y. Ye, and X. Xiu, "Energy efficient video decoding for the android operating system," in *Proc. IEEE International Conference on Consumer Electronics.,* Las Vegas, USA, Jan. 2013, pp. 344–345.

[9] 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.

[10] Y. L. Chen, M. F. Chang, and W. Y. Liang, "Energy-efficient video decoding schemes for embedded handheld devices," *Multimedia Tools and Applications.*, pp. 1-20, 2015.

[11] E. Raffin, E. Nogues, W. Hamidouche, S. Tomperi, M. Pelcat and D. Menard, "Low power HEVC software decoder for mobile devices," *Journal of Real-Time Image Processing.*, vol. 22, no. 3, pp. 495–507, 2016.

[12] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 645–657, May. 2005.

[13] MPEG, "Call for proposals on green MPEG," Joint Collaboration Team on Video Coding (JCT-VC), Apr. 2013.

[14] Y. Duan, J. Sun, L. Yan, K. Chen, and Z. Guo, "Novel efficient HEVC decoding solution on general-purpose process," *IEEE Trans. On Multimedia.*, vol. 16, no. 7, pp. 1915–1928, Jul. 2014.

[15] R. Ren, E. Juarez, C. Sanz, M. Raulet, and F. Pescador, "Energy-aware decoder management: A case study on RVC-CAL specification based on just-in-time adaptive decoder engine," *IEEE Trans. Consumer Electronics.*, vol. 60, no. 3, pp. 499 – 507, Aug. 2014.

[16] M. Chavarrias, F. Pescador, M. J. Garrido, E. Juarez, C. Sanz, "A multicore DSP HEVC decoder using an actor-based dataflow model and openMP," *IEEE Trans. Consumer Electronics.*, vol. 61, no. 2, pp. 236-244, May. 2015.

[17] E. Nogues, D. Menard, and M. Pelcat, "Algorithmic-level approximate computing applied to energy efficient HEVC decoding," *IEEE Trans. On Emerging Topics in Computing.*, vol. PP, no. PP, pp. 1-1, 2016.

[18] D. Brodowski, N. Golde, R. J. Wysocki, and V. Kumar "CPU frequency and voltage scaling code in the Linux (TM) kernel," [Online] Available: https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

[19] 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.

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

[21] T. C. Thang, Q. –D. Ho, J. W. Kang, and A. T. Pham, “Adaptive streaming of audiovisual content using MPEG DASH, ” *IEEE Trans. on Consumer Electronics*., vol. 58, no. 1, pp. 78 – 85, Feb. 2012.

[22] Y. Andreopoulos, and M. van der Schaar, “Complexity-constrained video bit stream shaping, ” *IEEE. Trans. on Signal Processing,* vol. 55, no. 5, pp. 1967 – 1974, May. 2007.

[23] M. Viitanen, J. Vanne, T. Hamalainen, M. Gabbouj, and J. Lainema, “Complexity analysis of next-generation HEVC decoder,” in *Proc. International Symposium on Circuits and Systems (ISCAS).*, Seoul, South Korea, May. 2012, pp. 882-885.

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

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

[26] C. Herglotz, D. Springer, A. Eichenseer, and A. Kaup, “Modelling the energy consumption of HEVC intra decoding,” in *Proc. International Conference on Systems, Signals, and Image Processing.*, Bucharest, Romania, Jul. 2013, pp. 91-94.

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

[28] T. Mallikarachchi, H. K. Arachchi, D. Talagala, and A. Fernando, “A feature based complexity model for decoder complexity optimized HEVC video encoding”, in *Proc. International Conference on Consumer Electronics.*, Las Vegas, USA, Jan. 2017.

[29] B. Li, H. Li, L. Li, and J. Zhang, “Lambda-domain rate control algorithm for High Efficiency Video Coding,” *IEEE Trans. On Image Processing.*, vol. 23, no. 9, pp. 3841-3854, Jul. 2014.

[30] G. J. Sullivan, and T. Weigand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine.*, vol. 15, no. 6, pp.74-90, Nov. 1998.

[31] JCT-VC, “HEVC reference software – HM16.0,” [online] Available: https://hevc.hhi.fraunhofer.de/svn/svn HEVCSoftware/tags/HM-16.0/

[32] "Open source HEVC decoder (OpenHEVC)", [online] Available: https:://github.com/OpenHEVC/

[33] Valgrind Developers, “The valgrind quick start guide,” [online] Available: http://valgrind.org/docs/manual/QuickStart.html.

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

[35] C. Herglotz, and A. Kaup, “Joint optimization of rate, distortion and decoding energy for HEVC intra-frame coding,” in *Proc. International Conference on Image Processing*., Arizona, USA, Sept. 2016.

[36] E. Monteiro, M. Grellert, S. Bampi, and B. Zatt, “Rate-distortion and energy performance of HEVC and H.264/AVC encoders: A comparative analysis,” in *Proc. IEEE International Symposium on Circuits and Systems(ISCAS).*, Lisbon, Portugal, May. 2015, pp. 1278-1281.

Thanuja Mallikarachchi (S’12) received his B.Sc. (Eng.) degree with honors in Electronic and Telecommunication Engineering from University of Moratuwa, Sri Lanka in 2011. From 2011 to 2013, he was a Senior Engineer at Virtusa (pvt) Ltd., Colombo, Sri Lanka. He is currently pursuing his Ph.D. in the Centre for Vision, Speech and Signal Processing (CVSSP) at the University of Surrey, United Kingdom.

His research interests are in the areas of video coding, video communication and video processing.

Dumidu S. Talagala (S’11-M’14) received the B.Sc. Eng (Hons) in Electronic and Telecommunication Engineering from the University of Moratuwa, Sri Lanka, in 2007. From 2007 to 2009, he was an Engineer at Dialog Axiata PLC, Sri Lanka. He completed his Ph.D. degree within the Applied Signal Processing Group, College of Engineering and Computer Science, at the Australian National University, in Canberra, Australia, in 2013. He is currently a research fellow in the Centre for Vision, Speech and Signal Processing at the University of Surrey, United Kingdom.

His research interests are in the areas of video processing and coding, sound source localization, spatial soundfield reproduction,array signal processing, audio-visual signal processing, sparse sensing and convex optimization techniques.

Hemantha Kodikara Arachchi (M’02) received his B.Sc. (Eng.) degree with honors and M.Phil. degree in Electronic and Telecommunication Engineering from University of Moratuwa, Sri Lanka in. 1997 and 2000 and the Ph.D. degree in Telecommunications from AIT, 2004.

At present, he is a Senior Research Fellow at the Centre for Vision, Speech and Signal Processing group of the University of Surrey, UK. Before joining University of Surrey, he was with Brunel University from 2004 to 2006. In 2003, he was with the Imaging Research Group, Loughborough University, UK as an academic visitor. His research interests are in video coding, video communication, QoE and context-aware content adaptation. He has published over 70 peer reviewed journal and conference papers

Anil Fernando (MS98-M01-SM03) received the B.Sc. Engineering degree (First class) in Electronic and Telecommunications Engineering from the University of Moratuwa, Sri Lanka in 1995 and the MEng degree (Distinction) in Telecommunications from Asian Institute of Technology (AIT), Bangkok, Thailand in 1997. He completed his PhD in video coding at the Department of Electrical and Electronic Engineering, University of Bristol, UK in February 2001.

Currently, he is a reader in signal processing at the University of Surrey, UK. Prior to that, he was a senior lecturer in Brunel University, UK and an assistant professor in AIT. His current research interests include cloud communications, video coding, Quality of Experience (QoE), intelligent video encoding for wireless systems and video communication in LTE. He has published more than 290 international publications on these areas.