Content-Adaptive Feature-Based CU Size Prediction for Fast Low-Delay Video Encoding in HEVC

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

***Abstract*—Determining the best partitioning structure of a Coding Tree Unit (CTU) is one of the most time consuming operations in HEVC encoding. Specifically, it is the evaluation of the quadtree hierarchy using the Rate-Distortion (RD) optimization that has the most significant impact on the encoding time, especially in the cases of High Definition (HD) and Ultra High Definition (UHD) videos. In order to expedite the encoding for low delay applications, this paper proposes a Coding Unit (CU) size selection and encoding algorithm for inter-prediction in the HEVC. To this end, it describes (i) two CU classification models based on *Inter N×N* mode motion features and RD cost thresholds to predict the CU split decision, (ii) an online training scheme for dynamic content adaptation, (iii) a motion vector reuse mechanism to expedite the motion estimation process, and finally introduces (iv) a computational complexity to coding efficiency trade-off process to enable flexible control of the algorithm. The experimental results reveal that the proposed algorithm achieves a consistent average encoding time performance ranging from 55% – 58% and 57% – 61% with average Bjøntegaard Delta Bit Rate (BDBR) increases of 1.93% – 2.26% and 2.14% – 2.33% compared to the HEVC 16.0 reference software for the *low delay P* and *low delay B* configurations, respectively, across a wide range of content types and bit rates.**

***Index Terms*—HEVC, CU size, low-delay, motion classification, RD optimization, video coding**

I. Introduction

THE recent developments in the Consumer Electronic (CE) technologies, and the content capturing capabilities of these devices have made multimedia the most frequently exchanged type of content over the modern communication networks. Moreover, the rapidly increasing mobile video data traffic, including the High Definition (HD) and Ultra High Definition (UHD) content captured by CE devices, (forecast to reach three-fourth of the mobile data traffic in 2019 [1]), present challenges such as the need for greater compression efficiency, energy efficiency and speed. In this context, the High Efficiency Video Coding (HEVC) [2] addresses the first of these requirements through the vastly superior performance exhibited over its predecessor H.264/AVC. However, the increased complexity of the features in the HEVC architecture significantly increase the demand for computational time and energy [5]; a non-trivial bottleneck for resource-constrained CE devices such as smart phones and camcorders. Efficient encoder designs that expedite the encoding process are therefore crucially important for the realization of high frame rate and real-time video communication applications in CE devices.

Although HEVC is essentially based on a hybrid coding architecture similar to that of H.264/AVC, it is accompanied by an assortment of novel coding features such as efficient prediction modes, filtering modes, parallelization tools, and flexible coding structures (e.g., Coding Units (CU), Prediction Units (PU), Transform Units (TU), etc.) [2]. The wide range of block sizes and combinations (i.e., 8×8 to 64×64) that this entails is one of the most important contributors towards the encoder's improved efficiency, yet at the same time, is also a major source of the complexity within the HEVC architecture [4], [5]. This is mainly due to the brute force Rate-Distortion (RD) optimization required for the combinations of coding modes that determines the best coding configuration. For example, an average encoding time increase of 43% is reported in [4], due to a simple increase of the maximum CU size from 16×16 to 64×64. Therefore, the recent literature has predominantly proposed mechanisms to reduce the complexity of the RD optimization that selects the best coding structure. In this context, the state-of-the-art fast encoding solutions generally utilize the depth correlation of spatial and temporal blocks, RD cost statistics of the CUs and the *Inter 2N×2N* prediction mode, feature-based offline and online training approaches, etc. [8]- [23], to determine the optimum CU size[1]. Hence, the selection of the CU size now becomes a prediction, whose effectiveness will determine the output quality and the bit rate of the encoded content. However, the vast differences in video characteristics, and the availability of additional information in the encoding chain itself, have not been fully investigated nor have they been exploited in these prediction approaches in order to realize a consistent encoding time saving across a wide range of content types and quality settings. Thus, the potential exists to develop implementation-friendly encoding algorithms that can effectively trade-off the coding efficiency in order to gain a reduction of the computational complexity.

To this end, this paper proposes a CU size prediction mechanism for low-delay HEVC video encoding based on the following contributions. First, (i) two independent content-adaptive decision making models are proposed to predict the optimal CU size; a dynamic motion feature-based model created by evaluating the *Inter N×N* mode motion and RD

T. Mallikarachchi, D. S. Talagala, H. Kodikara Arachchi, and A. Fernando are with the Centre for Vision Speech and Signal Processing, University of Surrey, Surrey, GU2 7XH, United Kingdom. e-mails: {d.mallikarachchi, d.talagala, h.kodikaraarachchi, w.fernando}@surrey.ac.uk

Manuscript received October 24, 2015; revised May 01, 2016, July 19, 2016 and September 19, 2016. This work was supported by the ACTION-TV project, which is funded under European Commission's 7^{th} Framework Program (Grant number: 611761).

[1]From the implementation perspective of the encoder, the CU size prediction boils down to a decision of whether a particular CU should be split into smaller CUs or not, i.e., the CU split decision referred to in the literature.

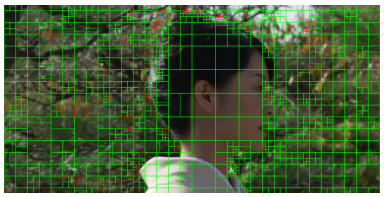


Fig. 1. The CU partitioning structure captured for a part of a frame in the "Kimono" sequence.

cost characteristics, and a heuristic RD cost threshold-based model. Thereafter, (ii) a window-based feature selection process is introduced to each model for independent, content-specific updating, which ensures the framework remains robust and adaptable to varying conditions such as scene changes. Finally, (iii) the *Inter N×N* mode motion vector reuse for further encoding complexity reduction, and (iv) the inclusion of a complexity control parameter to enable the flexibility of trading-off the proposed algorithm's complexity reduction for coding efficiency, are investigated. The simulation results reveal that the proposed algorithm achieves a significant and a consistent reduction of the encoding time compared to the HM 16.0 [26] implementation and the state-of-the-art algorithms with minimal impact on the RD performance for a variety of content types and quality levels.

The remainder of the paper is organized as follows. Section II provides an overview of the prior work found in the literature. The motion feature-based CU classification model and the RD cost threshold-based CU classification model are described in Section III. The resulting CU size prediction algorithm and the encoding framework are presented in Sections IV and V, respectively, and are followed by the results and discussion in Section VI. Finally, Section VII concludes and discusses the potential for future improvements.

II. Background and Related Work

A. Background

HEVC utilizes a block-based encoding architecture similar to H.264/AVC, albeit with a wider range of block sizes. In the main profile of HEVC, for example, a Coding Tree Unit (CTU) is partitioned into multiple CUs ranging from 8×8 to 64×64 in size (Fig. 1 illustrates an example partitioning structure for a typical video frame) [4], and contain multiple PUs and TUs that maintain prediction and transform information. Although this greatly enhances the flexibility of the architecture itself, the analysis of encoder implementations however reveal that the addition of these novel features produces a high coding gain to the detriment of computational complexity [5].

Rate-Distortion (RD) optimization, the process of determining the optimum coding modes and structure for a given content (which leads to the highest attainable coding efficiency in the encoder), uses a Lagrangian cost function to parameterize the encoding efficiency. This cost function is expressed as,

$$\underset{p}{\text{minimize}}\Big\{ D(p) + \lambda R(p) \Big\}, \quad p \in P \tag{1}$$

where $\lambda \geq 0$ is the Lagrange multiplier, p is a coding parameter combination from the set of all the possible coding options P, and $D(p)$, $R(p)$ are the distortion and rate associated with the selected set of coding parameters, respectively [6]. The process of selecting the optimum set of coding parameters, using the Lagrangian cost function in (1), is considered to be a major source of complexity in encoder implementations, due to the large number of possible combinations of the CU sizes, PU modes and other coding parameters [5]. The complexity increase in HEVC, with respect to both inter- and intra-prediction, therefore directly impacts the encoding performance; thus, the recent literature predominantly investigate mechanisms to minimize the number of RD evaluations and the encoding time, while retaining the coding performance. Moreover, minimizing the complexity associated with the coding structure determination and the motion estimation in inter-prediction is seen as a more pressing requirement as it encompasses a larger portion of the encoding time [5], [7].

B. Related Work

Reducing the computational complexity associated with inter-prediction in the HEVC architecture can be attempted for a range of operations, such as motion estimation, coding structure determination, filtering, etc. In fact, the complexity analysis presented in [7] suggests that optimizing the motion estimation alone only leads to a minimal reduction of the encoding time. However, the optimization of the different operations are not mutually exclusive, and could be utilized in conjunction with coding structure determination methods to achieve significant reductions of the encoding time. In this context, a summary of state-of-the-art is presented in the Table I to illustrate the operational basis of each algorithm.

Optimizing the PU level mode decision is one of the more popular complexity reduction methods found in the recent literature. Thus, in addition to the approaches summarized in Table I, this can also encompass methods that use spatio-temporal correlations and inter-level mode information [10], RD cost prediction [12], and the dynamic encoder parameter selection [15]. Furthermore, in the recent past, the SKIP mode decision has been used extensively to terminate the PU mode evaluation (Table I). However, the common theme in these works is the fact that they focus on the PU mode decision and do not consider the CU size decision (it should be noted that they can still be used in conjunction with a CU size selection algorithm); thus, every CU depth level must be evaluated in order to determine the optimum CU size. In practice, experimental results [13] suggest that the potential exists for further reduction of the encoding complexity than what has already been achieved.

Crucial to this problem is the optimal selection of the CU size in the coding hierarchy; hence, Table I also summarizes the relevant state-of-the-art CU size selection algorithms and discusses their merits and demerits. Here, it is observed that these CU size selection algorithms which predominantly rely on the statistics gathered from the evaluation of the *Inter 2N×2N*, SKIP, or other PU modes, can be enhanced further. This is especially true in the case of textured sequences with complex motion where high quality is required; an area where existing implementations show some deficiency, and

TABLE I
A SUMMARY OF THE STATE-OF-THE-ART QUADTREE STRUCTURE OPTIMIZATION SCHEMES USED FOR FAST HEVC ENCODING

Reference	Algorithm Description	Comments
Gweon *et al.* [8]	Uses the Coded Block Flag (CBF) to skip the evaluation of remaining PU modes.	The complexity reduction achieved is limited since only PUs are being considered and the lower likelihood of the CBF condition being satisfied in complex sequences.
Vanne *et al.* [9]	Optimizes the PU mode selection between Symmetric (SMP) and Asymmetric Partitions (AMP). The method selectively skips PU mode evaluations based on the merge and SKIP modes.	Complexity is reduced with respect to [8] due to the early determination of PU modes that can be skipped. However, the complexity reduction is less significant when the number of skipped PU modes are lower. The performance is seen to degrade for complex and textured content which generally results in fewer merge and SKIP mode selections. A similar behavior is observed at lower QPs.
Sampaio *et al.* [11]	A Motion Vector Merge (MVM) algorithm that evaluates the *Inter N×N* mode to determine which SMP PU modes to evaluate.	The computational complexity reduction is limited due to the smaller number of PU evaluations that are skipped.
Yang *et al.* [14]	Early Skip Mode Detection (ESD) assesses the SKIP mode selection of a CU and determines when to skip the remaining modes.	The approach is more suited for less complex sequences encoded at higher QPs where the SKIP mode is more likely to be the preferred coding mode.
Shen *et al.* [16]	Utilizes the neighboring and co-located CU information to determine and skip unnecessary CU depth levels.	RD performance experiences a drastic degradation for highly textured and complex motion sequences, due to sub-optimum decisions derived from the spatially adjacent CUs. Hsu *et al.* [22] therefore suggest encoding intermediate frames using the traditional RD optimization, which reduces the coding losses and error propagation, albeit increasing the encoding time.
Lee *et al.* [13]	Employs RD cost statistics of *Inter 2N×2N* mode evaluations, and the status of SKIP and merge modes, to determine the CU early termination depths. A SKIP Mode Decision (SMD) algorithm makes use of the SKIP mode selection statistics to skip both PU modes and CU depth levels.	The algorithm demonstrates good RD performance and computational complexity reductions. However, a large variance in the achievable complexity reduction is observed with respect to the QP and content types. Use of higher QPs and encoding of less complex contents results in the best performance due to a greater prevalence of SKIP and merge modes. Moreover, the evaluation of the current depth level becomes futile in the event the CU requires further splitting, such as in the case of low QP encoding of complex content.
Shen *et al.* [17]	CU split decisions are calculated using a Bayesian decision rule based on a feature set collected from the *Inter 2N×2N* mode.	The threshold comparison values (which are not adapted to the QP or to the content) and decision tree topology derived from offline training is less adaptable to changing content; thus, more dynamic video sequences and sequences with scene changes exhibit degraded RD performance. Furthermore, the early termination conditions evaluated at the end of each CU depth results in a reduction of the encoding time performance at high bit rates.
Shen *et al.* [18]	CU early termination decisions based on an offline Support Vector Machine (SVM) trained algorithm.	
Correa *et al.* [19]	Early CU termination using the offline trained decision trees and threshold values.	
Xiong *et al.* [20]	Uses Pyramid Motion Divergence (PMD) features calculated from the optical flow of downsampled frames for CU size selection.	The optical flow calculation within the encoding loop, that is used to extract motion, can be both time consuming and resource intensive [21].

is a scenario that must be addressed to operate at a wide range of bit rates. Moreover, a content adaptive operation is crucially important to cater for the diversity of the content; thus, the capacity for dynamic training and content-specific feature extraction also becomes a necessity to improve the general performance of the algorithms.

III. CU CLASSIFICATION FOR SPLIT LIKELIHOOD MODELLING

The CU sizes resulting from the splitting decisions attained through the RD optimization are highly dependent on the nature and the complexity of the content. Therefore, predicting the CU size beforehand, using a set of pre-determined features, becomes challenging due to this dynamic nature of the problem. In this context, it becomes evident that the early determination of the CU size requires a modelling of the CU split likelihoods of that particular content using a set of content-specific features. The selection of appropriate features that accurately model the CU split decision, and are also easy to extract from the encoding chain, is therefore crucially important. Two dynamic content-specific techniques that can be used for this purpose are described next.

A. Motion Feature-Based CU Classification

The first model, based on the motion feature-based CU classification approach [23], [24], attempts to represent the CU split likelihood as a function of three parameters given by

$$f(\boldsymbol{F}) := f(\alpha, \beta, \omega), \tag{2}$$

where α, β, and ω represent a motion classification, an *Inter N×N* RD cost category and the CU size, respectively, and f denotes a probabilistic model that determines the outcome of the CU split decision and is described in Section IV.

First, when observing the partitioning behavior of CUs during inter-prediction, it can be noticed that blocks with similar motion tend to utilize larger CUs, whereas blocks with complex motion tend to utilize smaller CUs [11], [20], [25]. Attempting to classify these characteristics, from the information available within the encoding chain itself, therefore becomes attractive due to both its simplicity and its minimal impact on the complexity. To this end, this paper proposes an initial *Inter N×N* mode evaluation (skipping the traditional PU evaluation order [3]) to collect the necessary motion information for each CU. The motion of the CU is

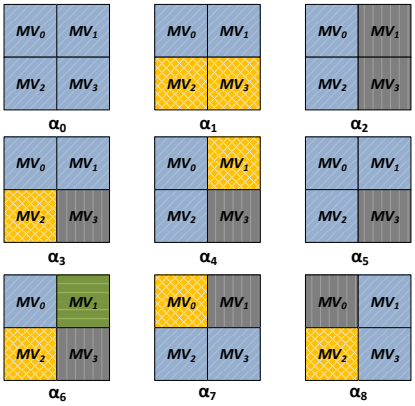


Fig. 2. CU categorization based on motion characteristics. Equal motion vectors are identified by the same colour/pattern. Here, α_5 denotes all orientations of the category where three motion vectors are equal and one differs.

TABLE II
AVERAGE SPLIT LIKELIHOOD (%) OF CUS IN MOTION CATEGORIES†

Sequence	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
Hall CIF	43	49	48	51	58	55	39	52	59
Highway CIF	18	35	34	47	49	41	59	44	45
Container CIF	30	55	56	47	48	55	50	47	58
Kimono HD	7	17	21	27	32	19	38	26	30
ParkScene HD	52	48	50	50	55	54	56	49	55
City HD	47	52	52	49	60	58	64	54	61

† A typical HD sequence would on average result in approximately 100 occurrences of each motion category per frame.

classified thereafter based on the similarity[2] of the resulting four motion vectors of the constituent blocks [23]. This results in a CU being classified into one of nine categories depicted in Fig. 2 and denoted by α_i $i \in \{0, 1, ..., 8\}$. Here, the motion vectors of the $k \in \{0, 1, 2, 3\}$ blocks are given by

$$MV_k = (mv_k, rPOC_k), \tag{3}$$

where $rPOC_k$ is the reference Picture Order Count (POC) number of the reference frame of the motion vector mv_k.

Next, analyzing the split likelihood (i.e., the ratio between the number of CUs that are split and the total number of CUs) in Table II of a CU classified as described above, it can be seen that textural diversity becomes important. The motion category alone no longer sufficiently models the split likelihood of a CU; thus, additional features are necessary to realize a more robust model. The *Inter* N×N RD cost γ can be a second parameter that describes the split likelihood. However, in practice, the range of γ is quite large, and may make the statistical analysis of individual RD costs less useful especially in the case of very rare large γ values. Therefore, the square-root of the RD cost is adopted instead, and is consolidated into one of 200 bins with a bin size Δ of 5. This results in a RD cost category β given by,

$$\beta = \begin{cases} \left\lfloor \frac{\sqrt{\gamma}}{\Delta} + \frac{1}{2} \right\rfloor & \sqrt{\gamma} \leq 200\Delta \\ 200 & \text{otherwise.} \end{cases} \tag{4}$$

[2]Two motion vectors are considered to be equal when each others horizontal and vertical components are equal and point to the same reference picture.

Finally, the CU size ω can be used as a third parameter that describes the CU split likelihood. The relationship between α_i, β and ω, and the motion-based feature $\boldsymbol{F}$ in (2), however remains complex and content dependent as illustrated in Fig. 3. For example, it can be observed that each α_i exhibits diverse CU split likelihoods for different β in Fig. 3(a) and ω in Fig. 3(b), while larger β favours CUs with larger ω to be split in Fig. 3(c). Thus, although the motion feature-based approach possesses the necessary flexibility to model the dynamic nature of the content, a separate decision making process (described in Section IV-A) is still required.

B. RD Cost Threshold-Based CU Classification

In contrast to the motion feature-based approach, the CU split likelihood can also be modelled in a partly heuristic fashion. This second model investigates its relationship with respect to a general distribution of the *Inter* N×N mode RD cost γ, the CU size ω and the Quantization Parameter QP.

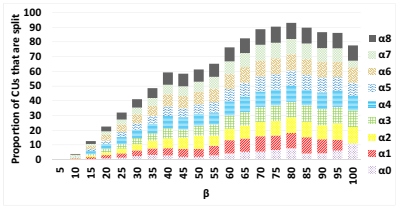
First, observing multiple video sequences reveals that the CU splitting behavior can be modelled by two Gaussian distributions; a CU split and non-split likelihood distribution [13], [28]. Fig. 4(a) illustrates an example of these with respect to γ for a particular CU size and QP. Crucially, these distributions reveal the existence of three regions within the range of γ that demarcate the CUs that are split, the CUs that are not-split and a third region where the decision is ambiguous. The generalized behavior of these distributions is illustrated in Fig. 4(b), and can be used to classify the CUs based on two RD cost thresholds $\gamma \geq HTh_{spt}$ and $\gamma \leq HTh_{nspt}$, where HTh_{spt} and HTh_{nspt} are the CU split and non-split thresholds, respectively (adaptively determining these thresholds using the mean γ values of the CU split and non-split Gaussian distributions is described in Section IV-B).

To this end, the distribution of HTh_{spt} and HTh_{nspt} is analyzed for sequences encoded using the *low delay P* configuration and $QP \in \{22, 27, 32, 37\}$ in HM16.0. Despite some variations, from the results observed in Fig. 5, it is evident that an exponential curve can parameterize the behavior of these two RD cost thresholds. Thus, generalized RD cost thresholds can be obtained, which are given by

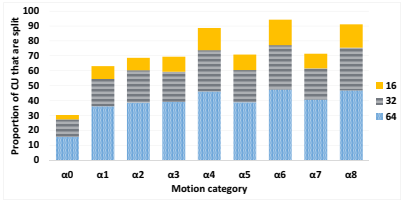
$$HTh_{spt} = \begin{cases} 2347 \times e^{0.1248 \times QP}, & \omega = 64 \\ 851.2 \times e^{0.1228 \times QP}, & \omega = 32 \\ 279.9 \times e^{0.1227 \times QP}, & \omega = 16 \end{cases} \tag{5}$$

$$HTh_{nspt} = \begin{cases} 736.2 \times e^{0.1378 \times QP}, & \omega = 64 \\ 225.4 \times e^{0.1468 \times QP}, & \omega = 32 \\ 57.72 \times e^{0.1607 \times QP}, & \omega = 16 \end{cases}. \tag{6}$$

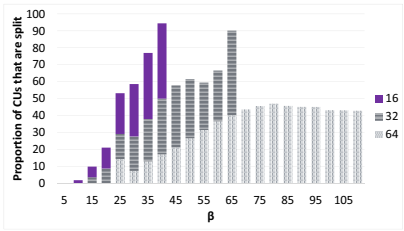
Table III summarizes the R-squared measure of the goodness-of-fit obtained for HTh_{spt} and HTh_{nspt} in (5) and (6). The modelled curves and the data illustrated in Fig. 5 suggest that the proposed models for HTh_{spt} and HTh_{nspt} are good representations in general, yet, are not particularly accurate, especially at higher QP values given the content specificities. Obtaining these content specific thresholds is however crucially important for an accurate optimal CU size prediction. In essence, threshold values in (5) and (6) could be



(a) Normalized proportion of CUs that are split in each motion category (α_i) for each β. (Averaged over CU sizes 64, 32 and 16).

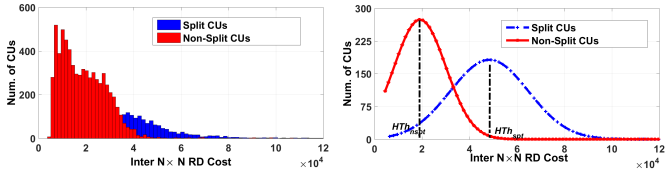


(b) Normalized proportion of CUs that are split in each CU size (ω) for each motion category (α_i). (Averaged over entire range of β).



(c) Normalized proportion of CUs that are split in each CU size (ω) for each β. (Averaged over all motion categories).

Fig. 3. Distribution of normalized proportion (%) of CUs that are split across β, motion category (α) and CU size (ω) for 200 frames in "City (720p)" video sequence when encoded with QP=27 using *low delay P main* configuration in HM 16.0. Results depict the CU split likelihoods for the different combinations of the parameters α, β, and ω in the feature vector $\boldsymbol{F}$.



(a) (b)

Fig. 4. (a) CU split and non-split likelihood distributions of the "ParkScene HD" video sequence for QP=32 using the *low delay P* configuration in HM 16.0. (b) A representation of the HTh_{spt} and HTh_{nspt} RD cost thresholds that identifies the CU split, CU non-split and ambiguous regions.

TABLE III
R^2 GOODNESS-OF-FIT OF THE SPLIT AND NON-SPLIT THRESHOLDS

CU Size	HTh_{spt}	HTh_{nspt}
64	0.827	0.782
32	0.703	0.906
16	0.780	0.882

used as initialization parameters during the preliminary stages of the encoding, and can later be adapted with content specific data as described in Section V.

IV. Fast CU Size Selection

This section describes how the two independent CU split likelihood models in Section III can be used in a complementary fashion, to determine if a CU should be split or not, all the while adapting to the specific content being encoded.

A. Motion Feature-Based CU Size Selection

Applying the feature-based CU classification model, the split probability of a CU in the n^{th} frame can be defined as

$$P_{s,n}(\boldsymbol{F}) = \frac{D^1_{act}(\boldsymbol{F})\big|_n}{D^1_{act}(\boldsymbol{F})\big|_n + D^0_{act}(\boldsymbol{F})\big|_n}, \tag{7}$$

where $D^{\eta}_{act}(\boldsymbol{F})\big|_n$ is the number of CUs with a feature vector $\boldsymbol{F}$, within the given frame that are either split ($\eta = 1$) or not-split ($\eta = 0$), based on the actual split decision obtained for the CUs by the RD optimization. Thus, $P_{s,n}$ can be considered a frame-wise statistic (calculated at the end of each encoded frame) of the optimal split decision computed from the statistics accumulated in $D^{\eta}_{act}(\boldsymbol{F})\big|_n$ that is obtained for each feature vector during the training phases described in Section V-B[3]. However, since this statistic can vary over time due to changes in the underlying content, a snapshot of the actual split probability is obtained through a windowed averaging process across the split probabilities calculated for the individual frames (i.e., $P_{s,n}(\boldsymbol{F})$). Mathematically, this can be expressed as

$$P_s(\boldsymbol{F})\big|_n = \frac{1}{\widetilde{W}} \sum_{t=0}^{n-1} P_{s,t}(\boldsymbol{F}) H(t-n), \tag{8}$$

where the window function $H(t)$ is given by

$$H(t) = \begin{cases} 1 + \frac{t}{W} & 0 \geq t \geq -W \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

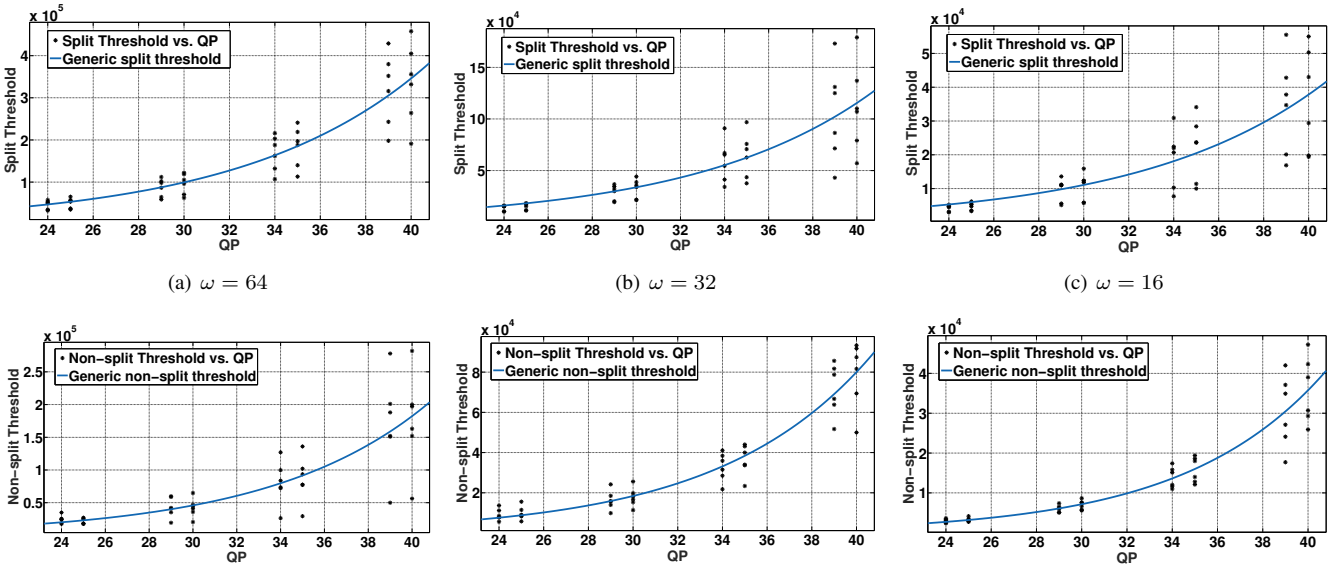
and $\widetilde{W}$ represents the area under the curve of $H(t)$. The window function and its effective length W are crucial in terms of the content adaptability, as it enables the predicted CU split decision to be biased to the most recent W frames. Therefore, by appropriately selecting W, the prediction can be made content adaptive and less susceptible to scene dynamics.

The outcome of the decision of whether to split or not-split a CU is now obtained by comparing (8) with an empirically determined threshold T, such that the decision for the n^{th} frame is given by

$$D_{fs}\big|_n = \begin{cases} 1 & P_s(\boldsymbol{F})\big|_n \geq T \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

The threshold T therefore acts as a switch that either splits or does not split the CUs. Empirical observations reveal that the value of T impacts both the bit rate and quality, where a smaller value of T generally results in more CUs being split, while a larger T results in less splitting. In this context, T and the window length W can be considered as design parameters that need to be empirically determined and preset for a desired trade-off of the quality and the bit rate.

[3]It should be noted that, in order to gather the statistics of the actual splitting behavior of the content being encoded, the RD optimization in (1) can not be completely bypassed. Thus, the outcome of the split decision must be evaluated using an appropriate balance of either the two models presented in this paper or the traditional RD optimization approach. Precisely how this is implemented to achieve the fast coding objective, is described in Section V-B.



(a) $\omega = 64$ (b) $\omega = 32$ (c) $\omega = 16$

(d) $\omega = 64$ (e) $\omega = 32$ (f) $\omega = 16$

Fig. 5. CU split and non-split thresholds and the fitted exponential curves with respect to QP of 6 different HD video sequences. The top and bottom rows correspond to the CU split and non-split thresholds, respectively. The y-axis depicts each threshold with respect to the *Inter* $N\times N$ RD cost, (γ).

B. RD Cost Threshold-Based CU Size Selection

In the RD cost threshold-based CU classification approach, the *Inter* $N\times N$ RD cost γ is compared to the HTh_{spt} and HTh_{nspt} thresholds for the ω and QP relevant to each CU. The resulting split decision D_{hs} of the n^{th} frame can therefore be expressed as

$$D_{hs}\big|_n = \begin{cases} 1 & \gamma \geq HTh_{spt} \\ 0 & \gamma \leq HTh_{nspt} \end{cases}. \tag{11}$$

However, as described in the previous section, HTh_{spt} and HTh_{nspt} need to be made adaptive in order to become content-aware, and a heuristic process can be used to adapt the generic HTh_{spt} and HTh_{nspt} thresholds in (5) and (6).

First, in order to adapt the split and not-split thresholds to the content, the RD cost statistics of the actual split decision (similar to the feature-based model in (7)) are analyzed in terms of the mean and standard deviation of these thresholds for a particular CU size ω and Quantization Parameter QP. Adopting a window-based approach to maintain the content adaptability as before, the mean and standard deviation statistics during the n^{th} frame for the split RD cost threshold can be expressed as

$$\mu^{spt}\big|_{n,\omega,QP} = \frac{1}{W}\sum_{t=0}^{W-1} E\left[\gamma^{spt}(n-t)\big|_{\omega,QP}\right] \tag{12}$$

and

$$\sigma^{spt}\big|_{n,\omega,QP} = \sqrt{\frac{1}{W}\sum_{t=0}^{W-1} E\left[\left(\gamma^{spt}(n-t)\big|_{\omega,QP} - \mu^{spt}\right)^2\right]}, \tag{13}$$

respectively. $E(\cdot)$ represents the expectation operating on the applicable CUs in that frame, and $\gamma^{spt}(\cdot)\big|_{\omega,QP}$ represents the RD cost of the CUs that are split with a CU size of ω and a Quantization Parameter QP. The RD cost threshold statistics for the not-split scenario, $\mu^{nspt}\big|_{n,\omega,QP}$ and $\sigma^{nspt}\big|_{n,\omega,QP}$, can be obtained in a similar fashion.

Thus, the two thresholds themselves are made content adaptive by applying the following;

$$HTh_{spt}\big|_{n,\omega,QP} = \left\{\mu^{spt} + \tau \times \sigma^{spt}\right\}\big|_{n,\omega,QP} \tag{14}$$

and

$$HTh_{nspt}\big|_{n,\omega,QP} = \left\{\mu^{nspt} - \tau \times \sigma^{nspt}\right\}\big|_{n,\omega,QP}. \tag{15}$$

The parameter τ acts as a governor that controls the adaptation of the model via the adaptation and training process described in Section V-B, and is an empirical design parameter that can be used to trade-off the computational complexity for the coding efficiency in the proposed encoding algorithm.

V. Proposed Fast Encoding Framework

In this section, computing the ultimate CU split decision, using the two independent decisions in the previous section, is described. This is followed by a mechanism to exploit the *Inter* $N\times N$ mode motion characteristics obtained during the CU split likelihood modelling, to supplement the CU size selection algorithm and further expedite the encoding process.

A. Joint CU Split Decision Prediction

The approach to using the two independent decisions described in Section IV forms the basis of the proposed fast encoding algorithm. Thus, when obtaining the joint decision, two distinct categories exist; one where both independent split decisions concur, and another where they differ. Hence, for the former category, the joint split decision during the n^{th} frame predicted by the encoding framework can be expressed as

$$CU_{\eta}\big|_n = \begin{cases} 1 & D_{fs}\big|_n = 1 \wedge D_{hs}\big|_n = 1 \\ 0 & D_{fs}\big|_n = 0 \wedge D_{hs}\big|_n = 0, \end{cases} \tag{16}$$

where $D_{fs}\big|_n$ and $D_{hs}\big|_n$ are the two independent decisions obtained for the applicable $\boldsymbol{F}$, γ, ω and QP of each CU. The second category of decisions, where the models differ, can now be used to initiate the adaptation of the framework to enhance its robustness to different contents.

B. Model Adaptation and Training

Following from the discussion of the joint split decision prediction, it is crucial that the models are able to adapt; thus, some RD evaluation becomes essential to calculate the actual CU split statistics in Section III. The proposed algorithm therefore consists of multiple training phases that facilitate the gathering of these statistics, such that the expected performance gains are not compromised. These are described in detail below.

1) Initial Training: Both split likelihood models described in Section III require some initial training[4] to gather content specific data at the beginning of a video sequence. In this work, the first four frames, i.e., $n = 1, \ldots, 4$, are used for this statistical information gathering. During this phase, the CU split decisions are obtained via the traditional RD optimization; thus, the two models are initialized with sufficient content specific data. However, these models may diverge from the actual content due to changes in the scene, making the accumulated statistics less relevant with time. Hence, the models must be continually refreshed, via intermediate training, as described next.

2) Intermediate Training: Intermediate training of the proposed framework can be split into three categories; training where no data exists for the features associated with a CU, training where the two models' decisions differ, and training for modelling efficiency improvements.

The first type of intermediate training is triggered when the actual information required to compute $P_s(\boldsymbol{F})\big|_n$ does not exist (e.g., a situation where the feature $\boldsymbol{F}$ has not been encountered within the window length W). Similarly, the RD optimization is followed in the event that $D_{fs}\big|_n$ in (10) and $D_{hs}\big|_n$ in (11) contradict each other. In both cases, the additional information obtained regarding the actual splitting behaviour will result in the refinement of both split likelihood models, thereby improving the accuracy of the subsequent decisions of similar CUs.

The third and most important intermediate training phase is triggered by RD costs where $HTh_{nspt} < \gamma < HTh_{spt}$ (see Fig. 4(b)), and the CU split decision of the RD cost threshold-based model $D_{hs}\big|_n$ is undefined as per (11). Controlling the size of this region will result in a trade-off of the output quality for the reduction of the computational complexity. Hence, in addition to facilitating the statistics gathering, the complexity control parameter τ (introduced for this purpose in (14) and (15)) affords a degree of flexibility when implementing the overall algorithm. Here, a larger τ will expand this region, allowing more decisions to be taken by the RD optimization, resulting in better quality. A smaller τ on the other hand will reduce the number of the RD optimizations and will result in improvements to the encoding time performance.

[4]The RD cost threshold-based model can still utilize the generic values of the HTh_{spt} and HTh_{nspt} thresholds in (5) and (6), respectively, until such content specific statistics are accumulated.

C. Motion Vector Reuse in Motion Estimation

This subsection describes how the motion vectors computed during the *Inter N×N* mode evaluation can be reused to further expedite the encoding process during the subsequent PU level evaluations that repeatedly occur for each CU.

Consider the four constituent motion vectors and reference frames returned by the initial *Inter N×N* mode evaluation in (3). These motion vectors identify the motion category of a CU, α_i, which interestingly has a structural relationship with the PU modes as illustrated in the Fig. 6. This relationship can be exploited to skip the motion estimation when a particular PU mode is being evaluated for a CU. For example, when a CU possesses the motion category α_0 (i.e., all four motion vectors are equal and point to the same reference picture), the motion vector MV_0 available in (3) can be reused for the *Inter 2N×2N* mode, thereby skipping the motion estimation phase for that PU mode. However, not all PUs can be identified in this fashion; thus, some PUs require motion estimation, e.g., the PUs denoted by "ME" and the PU modes that are not illustrated in Fig. 6 will require the usual motion estimation.

Hence, this capability to reuse the motion information extracted during the CU size prediction emerges as a direct secondary benefit of the initial *Inter N×N* mode evaluation. As a result, the proposed framework is supplemented with this feature to further expedite the encoding process.

D. The Overall Fast Encoding Algorithm

The performance improvements of the fast encoding algorithm proposed in this paper can be described as a result of two distinct operations; the content-adaptive CU size prediction in Section V-A and the motion reuse operation in Section V-C, that both exploit the initial *Inter N×N* mode evaluation. A high level flow diagram of the resulting algorithm, identifying the major decision making components and the operations of the individual blocks, is summarized in Fig. 7.

At an implementation level, if the CU split decision is in the negative, the CU is encoded at the selected depth level. However, if it is positive, the encoding cycle evaluates the next depth level of the CU. During the first N frames for the sequence, and whenever the CU split decision can not be predicted, the traditional RD optimization is triggered via the initial and intermediate training processes described in Section V-B. The statistics calculated during these periods are simultaneously used to update and refine the split likelihood models as described earlier. The shaded area in the Fig. 7 depicts the PU mode selection operation. Here, the available PU modes of the CU are evaluated in the traditional evaluation order [26], and the best PU mode is selected using an RD optimization. However, the motion estimation phase for a subset of PUs is skipped and the *Inter N×N* motion vectors are reused where appropriate as described in Section V-C.

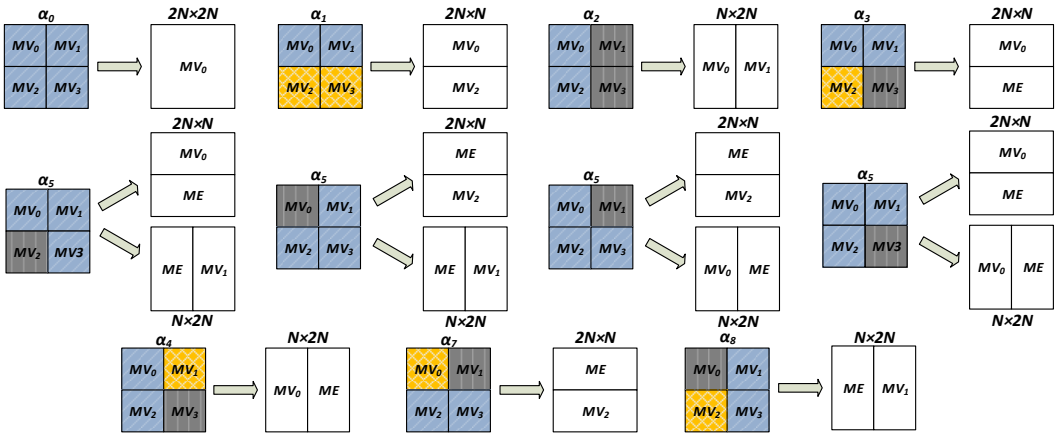


Fig. 6. The relationship between motion categories α_i and motion vector reuse for respective PU instances. Similar motion vectors in the *Inter* $N\times N$ mode are identified by the same colour and pattern. The PU instances denoted by "ME", and the PU modes that are not indicated, follow the traditional motion estimation process to determine the motion vectors.

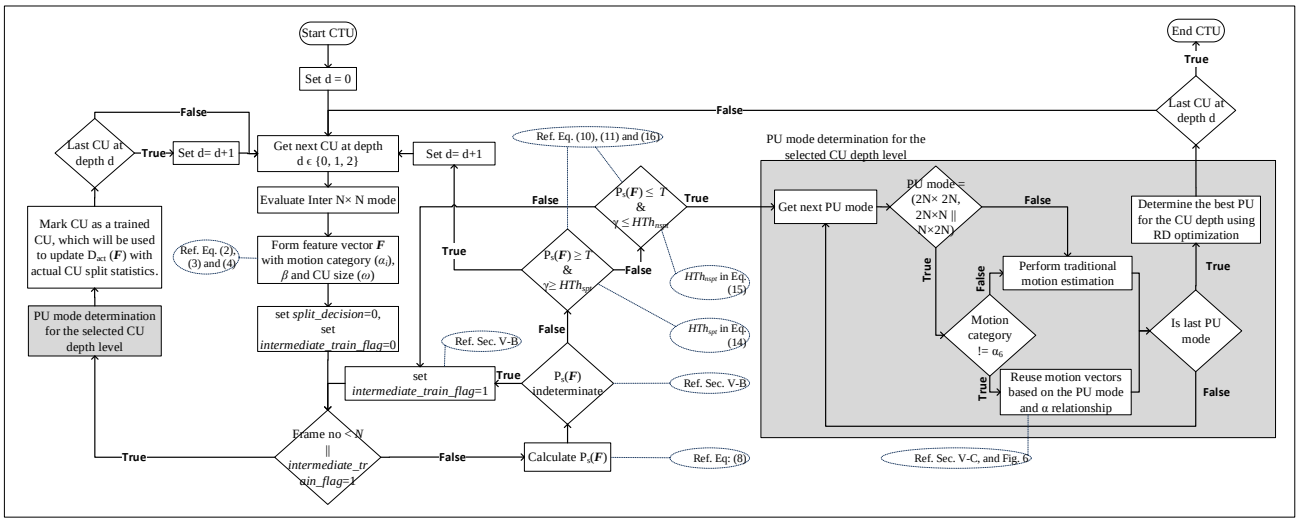


Fig. 7. The proposed fast encoding algorithm for HEVC based low delay video encoding. The flowchart describes the process of making the CU split/non-split decision during the compression phase. The model adaptation will take place during the encoding of the CTU with the selected CU structure. The dashed circles link the decision making blocks to the corresponding explanations and equations introduced in the previous sections.

VI. Experimental Results and Discussion

The following section presents the experimental results of the proposed content-adaptive fast CU size selection and encoding algorithm for low delay HEVC video encoding. The RD and encoding time performance of the proposed algorithm are compared with several state-of-the-art algorithms in the literature. These include the HM 16.0 [26] reference implementation, the CU size selection method proposed by Shen *et al.* [16], the fast encoding algorithms proposed by Lee *et al.* [13], the fast block partitioning algorithm proposed by Lu *et al.* [29], two versions of PU mode decision algorithms proposed by Vanne *et al.* [9], and the offline data mining approach to CU early termination proposed by Correa *et al.* [19].

A. Simulation Configurations and Performance Metrics

The algorithms are evaluated for a range of HD and UHD video sequences composed of both natural and synthetic content ranging from simple to highly complex motion with diverse spatial and temporal characteristics. Table IV summarizes the experimental setup and encoding configurations[5].

The impact on the RD performance is evaluated using the Bjøntegaard Delta Bit Rate (BDBR) [27] and the average percentage encoding time saving, ΔT, is evaluated for the proposed and state-of-the-art algorithms by comparing the implementations of the respective algorithms with the HM 16.0 reference software [26]. In this context, ΔT is given by,

$$\Delta T = 100 \times \frac{T_{HM} - T_{\rho}}{T_{HM}}, \tag{17}$$

where T_{HM}, is the encoding time of HM16.0 and T_{ρ} is the encoding time required for each fast encoding approach.

[5]The CU classification models proposed in this paper are based on the motion features extracted from the preceding video frames. Adopting the same approach for the *Random Access* configuration must also consider the impact of future frames and is therefore outside the scope of this work.

TABLE IV
SIMULATION SETUP AND CONFIGURATIONS

Configuration Parameter	Value
QPs	22, 27, 32, 37
Encoding Configurations	Low Delay P Main, Low Delay B Main
HEVC software version	16.0
Video sequence types	HD, UHD
Frame rates	HD: 30 fps, UHD: 50 fps
Number of frames	200
Machine	Intel Core i5, 8GB RAM, Ubuntu 14.04 LTS

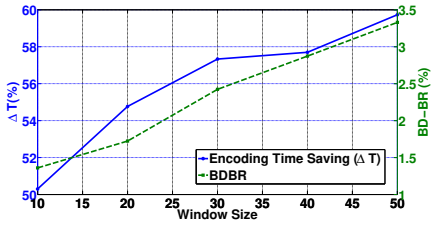


Fig. 8. The effect of the window length W (when $\tau = 0$) on the coding efficiency and the encoding time reduction of the proposed encoding algorithm.

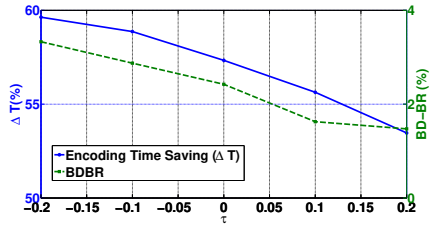


Fig. 9. The effect of the complexity control parameter τ (when $W = 30$) on the coding efficiency and the encoding time reduction of the proposed encoding algorithm.

B. Results and Performance Analysis

The window length W and the complexity control τ parameters defined in Section IV will naturally affect the performance of the proposed algorithm. This section first discusses how they can be selected and their the impact, and is followed by an analysis of the overall performance and implications of using the proposed fast coding framework.

1) Window Length and Complexity Control Parameter Selection: The experimental results illustrate the performance impact of W on ΔT and BDBR in Fig. 8 for HD sequences. Here, the value of the complexity control parameter is set as $\tau = 0$, which results in HTh_{spt} and HTh_{nspt} being exactly similar to the mean γ of the CU split, and non-split Gaussian distributions (as seen in Fig. 4 the mean values are already good approximations for these thresholds). It can be observed that the BDBR improves with smaller window sizes, while ΔT increases with increasing window size and vice-versa. Intuitively, this is due to less training being required for long W, yet longer window lengths also suggest less adaptability and sub-optimal quality. Evidently, the opposite is true when the W is shorter, where a smaller window size eventually increases the amount of training required, thus, resulting in a decrease in ΔT as well as BDBR. An empirically determined window size of $W = 30$ that both provides comparable average BDBR increases to that of the state-of-the-art algorithms and also facilitates the adequate accumulation of statistical data (in general, an average of approximately 25 training occurrences are observed for a typical feature vector), is used in the analysis in the remainder of this discussion.

The experimental results illustrated in Fig. 9 for different τ show that ΔT tends to increase with a corresponding increase of the BDBR. This is due to the fact that when the middle region is smaller, split decisions of more CUs become incorrect due to fewer RD optimization occurrences, which negatively impacts BDBR but improves ΔT. Naturally, when the region becomes larger the opposite is true, and is reflected in Fig. 9. Hence, the performance results of the proposed algorithm discussed in the remaining sections use $\tau = 0$, which together with the previously selected W, corresponds to BDBR increases comparable to the state-of-the-art solutions.

In addition, as discussed in Section IV-A, the threshold value T in (10) decides the portion of CUs that will be decided to split, for a given feature vector $\boldsymbol{F}$. In this context, an empirically determined value of T ($T = 0.6$) is used as the threshold, by considering it's impact on the RD efficiency.

2) Overall Performance of the Proposed Algorithm: The performance of the proposed algorithm is presented in the Tables V - VII for the *low delay P* and *low delay B* configurations. These results first address the impact of only the CU size selection aspect (described in Section V-A and V-B) denoted by SI. Then in SII the impact of including the motion vector reuse from the *Inter N×N* mode evaluation (described in Section V-C) is evaluated. The following discussion further analyzes these results in terms of the variations seen for different content types, QPs and other relevant attributes.

a) Performance Variation with QP: Examining the encoding time performance results illustrated in the Table V for a subset of sequences representing diverse content types and QPs ($QP = 22, 27, 32, 37$), a variation in ΔT with the corresponding bit rate of the video sequences can be observed. For example, ΔT tends to increase with the decreasing bit rates (i.e., increasing QP) for the proposed as well as the state-of-the-art algorithms. In general, this behaviour can be explained as follows. Typically, when encoding a CU at larger QPs, larger CUs and prediction modes such as SKIP and merge modes are favoured; thus, the algorithms that early terminate a CU at smaller depths and early detect SKIP/merge modes, demonstrate an increased ΔT compared to smaller QPs that yield smaller CUs and fewer SKIP mode PUs.

However, interestingly, the variation of ΔT with the QP is relatively large for the methods proposed by Lee *et al.* [13], Shen *et al.* [16] as well as Correa *et al.* [19]. This is due to the evaluation of the CU Early Termination (ECUT) and CU Skip Estimation (CUSE) conditions [13] that require the encoder to evaluate the current CU prior to determining whether the CU should be split further. Similarly, the verification of the decision trees introduced in [19] is performed as the final operation at a particular depth level, and results in a similar behavior. Here, video sequences that tend to use smaller CUs when using a lower QP will result in the algorithm

TABLE V
ENCODING TIME SAVING WITH RESPECT TO THE QUANTIZATION PARAMETER AND THE CONTENT

Algorithm	Quantization Parameter (QP)																			
	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37	22	27	32	37
	Kimono				Musicians				Dancer (Synthetic)				Traffic				Poznan			
	AM, HT (ΔT%)				LM, HT (ΔT%)				AM, LT (ΔT%)				HM, HT (ΔT%)				LM, LT (ΔT%)			
Proposed SI	53	52	53	52	50	50	52	52	47	51	55	55	51	52	55	58	50	61	63	64
Proposed SII	55	57	55	56	52	53	55	57	50	53	58	59	53	55	58	62	53	64	67	69
Lee *et al.* [13]	27	35	44	53	30	39	47	55	32	45	55	62	31	45	56	63	35	60	68	72
Shen *et al.* [16]	39	41	43	46	44	44	45	47	44	47	49	53	56	56	55	57	66	67	68	69
Lu *et al.* [29]	29	29	27	26	33	32	32	32	27	28	29	30	22	22	23	24	32	30	30	31
Vanne *et al.* S_{14} [9]	25	33	40	46	31	36	41	47	33	41	48	53	34	43	49	56	37	47	58	59
Vanne *et al.* S_{25} [9]	46	51	49	52	42	46	50	61	49	53	59	63	43	49	53	57	49	60	64	62
Correa *et al.* [19]	24	48	60	62	27	42	46	58	33	57	64	65	23	50	61	65	19	61	64	68

The sequence categories (i.e., LM, AM, HM, LT, HT) are defined as follows. LM: Low Motion, AM: Average Motion, HM: High Motion, LT: Low Texture, HT: High Texture.

TABLE VI
OVERALL PERFORMANCE OF THE PROPOSED ALGORITHM (LOW DELAY P)

Sequence	Proposed SI		Proposed SII		Lu *et al.* [29]		Lee *et al.* [13]		Shen *et al.* [16]		Vanne *et al.* S_{14} [9]		Vanne *et al.* S_{25} [9]		Correa *et al.* [19]	
	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)
Musicians 1080p	51	2.60	55	2.90	32	2.13	43	2.34	45	2.56	39	0.05	50	1.20	43	5.65
Band 1080p	56	1.78	59	1.82	40	0.68	47	1.10	51	3.60	42	1.08	57	1.51	57	10.2
Kimono 1080p	52	1.27	56	1.37	28	0.47	40	1.06	42	1.27	36	0.48	49	1.13	49	4.05
Parkscene 1080p	49	3.00	53	3.10	26	2.33	45	2.34	49	3.94	42	0.78	47	1.38	53	16.86
Dancer 1088p	52	1.32	55	1.69	29	3.90	49	0.56	48	1.49	44	0.39	56	1.10	55	12.64
GT Fly 1088p	52	1.68	54	1.74	34	3.65	50	1.03	42	2.43	41	2.29	57	4.38	50	7.84
Beergarden 1080p	55	1.00	58	1.01	16	0.54	48	1.40	58	5.65	43	0.94	46	1.53	47	3.89
Poznan 1088p	59	1.05	63	1.20	31	0.73	59	0.89	68	6.26	50	1.32	59	1.91	53	6.50
City 720p	54	1.76	58	2.38	21	1.11	52	1.30	59	1.71	48	1.01	51	2.38	58	18.33
Traffic 1600p	54	4.02	57	4.20	23	4.11	49	2.50	56	6.87	44	0.79	51	2.01	50	8.24
Men-Plants 2160p	59	2.08	62	2.48	41	2.03	51	1.80	64	2.84	45	0.72	50	1.85	56	5.01
Park-Buildings 2160p	60	2.59	62	2.60	29	1.10	55	1.69	58	3.23	49	1.78	52	2.81	54	1.76
Men-calendar	60	1.08	63	2.89	50	3.30	54	1.05	58	2.88	50	1.83	56	1.96	56	6.45
Average	**55**	**1.93**	**58**	**2.26**	**30**	**2.00**	**49**	**1.46**	**53**	**3.44**	**44**	**1.03**	**52**	**1.93**	**52**	**8.26**

unnecessarily evaluating the upper CU depth levels before early termination. In contrast, the proposed algorithm predicts the CU split decision prior to the encoding of a CU; thus, the unnecessary evaluations of larger CUs are avoided. This leads to far less performance variation between QPs, and suggests that the proposed algorithm is suitable for encoding videos across a wider range of bit rates unlike the existing approaches.

b) Performance Variation with the Content: It can be observed that all the state-of-the-art algorithms demonstrate a relatively large encoding time reduction for less textured sequences such as "Poznan Street" (Table VI and VII). This increase in ΔT is mainly due to the skipping of rarely used CU depth levels [16] and the ECUT methods that have been employed [13], [29]. Hence, the less complex the content (less textured and simple motions), the more likely it becomes that they will be encoded with larger CUs, causing the ECUT to exhibit an increased ΔT. In addition, static or content with simple motions can exploit the CU depth range estimation algorithms in [16] to skip the unnecessary CU depth levels.

However, in the case of more textured sequences with average or low motions (e.g., "Kimono", "Musicians" etc.,) much smaller CU sizes are required; thus, the ECUT checks at upper depth levels becomes ineffectual, leading to a much lower ΔT. Moreover, the skipping of the rarely used CU depth levels in [16] eventually leads to relatively high coding losses, especially in the case of sequences that exhibit multiple localized motions (i.e.,"Poznan Street"). The depth range estimations in this case often become less accurate, and the errors in these are propagated across the frame to further deteriorate the coding efficiency.

That said, the method proposed by Shen *et al.* performs reasonably well even with sequences that generally exhibit uniform motion across the frame (e.g., "Kimono"), to the detriment of the encoding time performance. In contrast to this approach, the proposed framework exhibits a performance that varies much less with the content. The effect of the content's complexity therefore appears trivial to the proposed algorithm, due to its early prediction of the CU split decision prior to the actual encoding of the CU. Furthermore, the proposed algorithm only evaluates the selected CU depth level, thus, the encoding time consumed for unnecessary CU depth level evaluation is avoided, and proves to be an advantage over the prevailing state-of-the-art encoding solutions when encoding a wider range of video sequences.

However, for sequences such as "Traffic", which has fast moving objects through out the sequence, the proposed method

TABLE VII
OVERALL PERFORMANCE OF THE PROPOSED ALGORITHM (LOW DELAY B)

Sequence	Proposed SI		Proposed SII		Lu *et al.* [29]		Lee *et al.* [13]		Shen *et al.* [16]		Vanne *et al.* S_{14} [9]		Vanne *et al.* S_{25} [9]		Correa *et al.* [19]	
	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)
Musicians 1080p	53	1.96	57	2.03	29	0.90	43	1.71	44	1.21	37	0.68	51	1.89	48	19.54
Band 1080p	57	2.08	61	2.20	37	1.10	49	1.20	53	1.35	40	0.76	52	1.18	49	9.11
Kimono 1080p	55	2.73	59	2.81	24	0.96	40	4.55	43	1.42	38	0.82	51	1.72	33	4.56
Parkscene 1080p	52	2.34	55	2.48	26	0.67	46	1.78	49	1.29	44	0.68	54	0.72	46	15.16
Dancer 1080p	54	1.87	57	2.13	29	3.56	50	1.31	48	1.20	42	0.20	55	2.16	49	11.8
GT Fly 1080p	53	1.35	57	1.72	37	5.48	46	3.46	42	0.98	32	0.41	53	2.90	36	7.09
Beergarden 1080p	56	1.71	61	1.18	12	0.23	48	1.10	55	4.98	36	0.61	52	1.08	48	12.22
Poznan 1088p	62	1.18	67	1.27	30	1.01	60	1.02	66	5.74	51	0.37	57	0.99	51	7.24
City 720p	52	1.44	56	2.16	20	1.00	48	1.59	48	0.97	30	0.53	52	3.19	28	6.65
Traffic 1600p	57	3.50	60	3.87	22	1.00	50	1.97	52	4.47	29	0.44	48	1.48	42	7.50
Men-Plants 2160p	62	3.00	66	3.10	32	2.01	54	2.27	50	1.37	33	0.10	48	1.66	58	15.52
Park-Buildings 2160p	63	1.49	66	2.02	21	1.16	59	2.58	54	2.37	31	0.74	56	1.77	58	5.1
Men-calendar 2160p	63	3.17	67	3.40	43	4.93	57	1.50	58	1.37	32	0.62	54	1.98	59	19.14
Average	**57**	**2.14**	**61**	**2.33**	**28**	**1.84**	**50**	**2.00**	**51**	**2.20**	**36**	**0.53**	**52**	**1.74**	**47**	**10.81**

and the state-of-the-art algorithms demonstrate a relatively large BDBR in the range of 4%. This suggests that the complex motions and rapidly changing content have caused the decision making algorithms to make fewer efficient decisions compared to the sequences with average motion complexity. One of the solutions envisioned in this case is to reduce the window size W to increase the content adaptability of the proposed algorithm. In this context, the proposed algorithm is sufficiently flexible in its design parameters to cater for the diversities of the video sequences. In addition to the joint utilization of the two independent models, the complexity control parameter coupled with the intermediate training phases provide a more attractive training solution for the proposed framework in contrast to the use of fully RD optimized training frames at pre-defined intervals as proposed by Lee *et al.* in [13].

c) PU Level Complexity Reduction: The impact of skipping PU modes is illustrated in the Tables VI and VII, for two algorithms proposed by Vanne *et al.* [9]. The algorithm S_{14} which skips a limited number of PU modes in each CU depth level based on the selection of SKIP or merge modes, results in a smaller complexity reduction, whereas the algorithm S_{25} produces greater complexity reductions to the detriment of the coding efficiency. Generally, the BDBR increase is relatively small in PU level optimization algorithms that skip the evaluation of asymmetric or certain symmetric partitions, as is the case here. This is due to the higher tendency of the RD optimization based PU mode selection preferring the selection of the *Inter* $2N \times 2N$ mode over the remaining PU modes at each CU depth level. However, the intermittent increases in the BDBR seen in the tables with respect to certain sequences stem from the ignorance of the remaining PU modes which eventually lead to the selection of less efficient CU and TU structures. Moreover, PU skip decisions derived based on the selection of SKIP and merge modes in [9], results in an increased ΔT for less complex sequences encoded at lower quality levels (ref. Table V), whereas the opposite is true when the sequences are encoded at higher quality levels. In contrast, the proposed algorithm maintains a consistent encoding time reduction ($\approx 50\%$) across all quality levels due to its SKIP/merge mode agnostic decision making models which perform the CU split decision prediction.

d) Content Adaptability and Online Training: The content adaptability and the effectiveness of the online training used in the proposed algorithm are further corroborated when comparing its performance with that of the offline trained algorithm proposed by Correa *et al.* in [19], whose BDBR exceeds 10% for some sequences due to its fixed thresholds and decision trees [19], [30]. These coding losses emphasize the inherent drawback of using offline trained algorithms, QP and content agnostic RD cost thresholds, and rigid decision tree topologies [30] on previously unseen data sets. Furthermore, the disregard for the ambiguous regions, which often exist in the CU split/non-split statistical distributions [30] (see Fig. 4), eventually lead to the less efficient split decisions in [19]; thus, degrades the RD performance. Consequently, retraining the algorithm to the new data sets is a potential solution, yet this may become a time consuming and tedious process.

In contrast, a relatively constant BDBR increase is observed for the proposed algorithm as a result of its content adaptive nature. For example, the content adaptability features of the proposed algorithm ensure that the data upon which the CU split decisions are made always correspond to the content being encoded. Moreover, the experimental results further reveal that the content adaptive approaches introduced in the proposed algorithm are far more effective in capturing the content specific information, as oppose to the state-of-the-art approaches which typically rely on spatial and co-located CU statistics [16] for content adaptation.

e) Motion Estimation Complexity Reduction: The effect of the proposed fast CU selection method when supplemented by the motion vector reuse to optimize the PU level motion estimation, is presented under "Proposed SII" in Tables VI and VII. In this case, the experimental results demonstrate an additional average encoding time saving of 3% and 4% for the *low delay P* and *low delay B* configurations, respectively. However, BDBR increases of 0.37% and 0.19% for the two configurations are exhibited in comparison to the "Proposed SI", due to the skipping of certain motion estimations.

Finally, it should be noted that the computational cost of the training phases and the decision making stages are all included in the encoding performance results that are reported in this paper. Therefore, it is evident that the additional complexities introduced by the proposed algorithm are negligible in comparison with the significant time saving that can be achieved by incorporating these algorithms into the encoding cycle.

VII. Conclusion

This paper proposes a content adaptive fast CU size selection algorithm for HEVC based low delay video encoding. In this context, two CU split likelihood models (based on a motion feature-based and a RD cost threshold-based CU classification approaches) are introduced to model the CU split and non-split decisions. These models are dynamically generated and are continuously adapted using initial and intermediate training phases, such that they independently predict the split decision for a given CU. Moreover, the possibility of reusing motion vectors identified during the modeling stage, for motion estimation in the remaining PU modes, is also investigated to supplement the proposed algorithm.

One major conclusion to be drawn from this analysis is that the initial evaluation of the *Inter* $N\times N$ mode provides motion and complexity properties of the underlying CU, which can be used to classify a CU, in order to model the split likelihood. Furthermore, the use of two independent models facilitates the split decision refinement as well as the identification of when the models require training dynamically during the encoding cycle. The window based approach used in the model adaptation and decision making ensures that the resultant split decisions are content-adaptive and less susceptible to the dynamic variations such as scene changes; a non-trivial advantage over the state-of-the-art methods.

In conclusion, the simulation results for the proposed CU size selection and encoding algorithm reveal an average encoding time saving of 58% and 61% for the *low delay P* and *low delay B* configurations, respectively. Moreover, the experimental results reveal that the proposed encoding algorithms can achieve a relatively uniform average encoding time saving across a wide range of QPs and content ranging from low to highly complex textures and motion characteristics, due to its SKIP/merge mode agnostic early CU size prediction. The capacity of the proposed algorithm to maintain a consistent performance, in terms of both the encoding time saving as well as BDBR increase (which is 2.29 % on average), across diverse content types and QPs is especially notable when considering the performance fluctuations observed in the state-of-the-art solutions. The future work will focus on extending algorithm to the remaining coding structures (i.e., PUs and TUs) and other configurations in order to further expedite the encoding process with minimal impact on the coding efficiency.

References

[1] Cisco, "Global Mobile Data Traffic Forecast Update 2010-2015", *Cisco Visual Networking Index*, Feb. 2015.

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

[3] I. Kim, K. D. McCann, K. Sugimoto, B. Bross, W. Han, and G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM16) Encoder Description," *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T,* Geneva, CH, 2013.

[4] 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.

[5] F. Bossen, B. Bross, S. Member, S. Karsten, and D. Flynn, "HEVC Complexity and Implementation Analysis," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 22, no. 12, pp. 1685-1696, Oct. 2013.

[6] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 22, no. 12, pp. 1669-1684, Dec. 2012.

[7] K.-Y. Kim, H.-Y. Kim, J.-S. Choi, and G.-H. Park, "MC Complexity Reduction for Generalized P and B (GPB) Pictures in HEVC," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 24, no. 10, pp. 1723-1728, Oct. 2014.

[8] R. H. Gweon and Y.-L. Lee, "Early Termination of CU Encoding to Reduce HEVC Complexity," *document JCTVC-F045 - Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T*, Torino, Italy, Jul. 2011.

[9] J. Vanne, M. Viitanen, and T. Hamalainen, "Efficient Mode Decision Schemes for HEVC Inter Prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1579-1593, Sept. 2014.

[10] L. Shen, Z. Zhang, and Z. Liu, "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatio-temporal correlations," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 24, no. 10, pp. 1709-1722, Oct. 2014.

[11] F. Sampaio, S. Bampi, M. Grellert, L. Agostini, and J. Mattos, "Motion Vectors Merging: Low Complexity Prediction Unit Decision Heuristic for the Inter-prediction of HEVC Encoders," in *Proc. IEEE International Conference on Multimedia and Expo.,* Melbourne, Australia, Jul. 2012, pp. 657-662.

[12] S. Jung, and H. Park, "A Fast Mode Decision Method in HEVC using Adaptive Ordering of Modes," *IEEE Trans. Circuits Syst. Video Technol.,* vol. PP, no. 99, pp. 1, Aug. 2015.

[13] J. Lee, S. Kim, K. Lim, and S. Lee, "A Fast CU Size Decision Algorithm for HEVC," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 25, no. 3, pp. 411-421, Mar. 2015.

[14] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon, "Early SKIP Detection for HEVC," *document JCTVC-F045 - Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T,* Geneva, Switzerland, Nov. 2011.

[15] G. Correa, P. Assuncao, L. Agostini, and C da Silva, "Pareto-Based Method for High Efficiency Video Coding with Limited Encoding Time," *IEEE Trans. Circuits Syst. Video Technol.,* vol. PP, no. 99, pp. 1, Aug. 2015.

[16] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An Effective CU Size Decision Method for HEVC Encoders," *IEEE Trans. Multimed.,* vol. 15, no. 2, pp. 465-470, Feb. 2013.

[17] X. Shen, L. Yu, and J. Chen, "Fast Coding Unit Size Selection for HEVC based on Bayesian Decision Rule," *in Picture Coding Symposium (PCS).,* May. 2012, no. 2010, pp. 2010-2013.

[18] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP J. Image Video Process.,* vol. 2013, no. 1, pp. 1-11, Jan. 2013.

[19] G. Correa, P. Assuncao, L. Agostini, and L. a. da Silva Cruz, "Fast HEVC Encoding Decisions Using Data Mining," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 25, no. 4, pp. 660-673, Apr. 2014.

[20] J. Xiong, H. Li, Q. Wu, and F. Meng, "A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence," *IEEE Trans. Multimed.,* vol. 16, no. 2, pp. 559-564, Feb. 2014.

[21] S. Ahn, B. Lee, and M. Kim, "A Novel Fast CU Encoding Scheme based on Spatio - Temporal Encoding Parameters for HEVC Inter Coding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 25, no. 3, pp. 422-435, Mar. 2014.

[22] W.-J. Hsu and H.-M. Hang, "Fast coding unit decision algorithm for HEVC," *in Asia-Pacific Signal and Information Processing Association Annual Summit and Conference.,* Kaohsiung, Taiwan, Oct. 2013, pp. 1-5.

[23] T. Mallikarachchi, A. Fernando, and H. K. Arachchi, "Fast Coding Unit Size Selection for HEVC Inter Prediction," *in Proc. IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, Jan. 2015, pp. 457-458.

[24] T. Mallikarachchi, A. Fernando, and H. K. Arachchi, "Effective coding unit size decision based on motion homogeneity classification for HEVC inter prediction," *in Proc. IEEE International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014, pp. 3691-3695.

[25] Z. Liu, L. Shen, and Z. Zhang, "An Efficient Inter Mode Decision Algorithm Based on Motion Homogeneity for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 19, no. 1, pp. 128-132, Jan. 2009.
[26] HEVC Reference Software - HM-16.0 [Online] $https$: $//hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0/$.
[27] G. Bjontegarrd, "Calculation of average PSNR differences between RD-curves," *ITU - Telecommunications Standardization Sector STUDY GROUP 16 Video Coding Experts Group (VCEG)*, Austin, Texas, USA, 2001.
[28] S. Cho, and M. Kim, "Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 23, no. 9, pp. 1555 - 1564, Feb. 2013.
[29] J. Lu, F. Liang, L. Xie, and Y. Luo, "A Fast Block Partition Algorithm For HEVC," *in Proc. International Conference on Information, Communications and Signal Processing (ICICS),,* Tainan, Dec. 2013, pp. 1-5.
[30] G. R. Correa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz "Complexity-Aware High Efficiency Video Coding," *Springer International Publishing,* 2016, pp. 125 - 158.

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. Prior to that, 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 (S98-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 with more than 290 international publications on these areas. He is a senior member of IEEE and a fellow of the HEA, UK. He is also a member of the EPSRC College.