

# End User Video Quality Prediction and Coding Parameters Selection at the Encoder for Robust HEVC Video Transmission

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

**Abstract**—Along with the rapid increase in the availability for high quality video formats such as HD (High Definition), UHD (Ultra HD) and HDR (High Dynamic Range), a huge demand for data rates during their transmission has become inevitable. Consequently, the role of video compression techniques has become crucially important in the process of mitigating the data rate requirements. Even though the latest video codec HEVC (High Efficiency Video Coding) has succeeded in significantly reducing the data rate compared to its immediate predecessor H.264/AVC (Advanced Video Coding), the HEVC coded videos in the meantime have become even more vulnerable to network impairments. Therefore, it is equally important to assess the consumers' perceived quality degradation prior to transmitting HEVC coded videos over an error prone network, and to include error resilient features so as to minimize the adverse effects those impairments. To this end, this paper proposes a probabilistic model which accurately predicts the overall distortion of the decoded video at the encoder followed by an accurate QP- $\lambda$  relationship which can be used in the RDO (Rate Distortion Optimization) process. During the derivation process of the probabilistic model, the impacts from the motion vectors, the pixels in the reference frames and the clipping operations are accounted and consequently the model is capable of minimizing the prediction error as low as 3.11% whereas the state-of-the-art methods can't reach below 20.08% under identical conditions. Furthermore, the enhanced RDO process has resulted in 21.41%-43.59% improvement in the BD-rate compared to the state-of-the-art error resilient algorithms.

**Index Terms**—HEVC, error resilient video coding, AMVP, error concealment.

## I. INTRODUCTION

VIDEO content production has been rapidly evolving since the past decade, enabling the end user to access high quality video formats such as High Definition (HD), Ultra HD (UHD) and HDR (High Dynamic Range). Along with this growth in the video formats, the need for higher bandwidths during their transmission is also unavoidable. As an example, an uncompressed UHD video can occupy several gigabytes of volume to store a few second video clip. Therefore, reducing the bandwidth consumption during the video transmission, with a minimal impact to the end user's Quality of Experience (QoE) has become a burning issue for video service providers.

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

Manuscript received MMMM DD, YYYY. This work was supported by the CONTENT4ALL project, which is funded under European Commissions H2020 Framework Program (Grant number: 762021).

To this end, video compression techniques such as H.264/AVC (Advanced Video Coding), H.265/HEVC (High Efficiency Video Coding) have drastically reduced the storage requirement and transmission bandwidths of these high resolution video formats [1]. However, as a result of the spatial and temporal correlation among the coded video frames, a loss of a packet during a video transmission which is fairly common in wireless media, could affect a number of frames which are dependent on the lost information. Nevertheless, as a consequence of the higher correlation of HEVC video frames, the quality degradation during a packet loss is much worse in the HEVC compared to the H.264 [2].

Therefore, it is crucially important to incorporate redundant information (i.e., error resilient features), so that the lost information during a video transmission can be recovered using already received information. In response to this, the solution proposed in this paper is two folds. First, the paper proposes an end user video quality prediction algorithm which is compatible with the Advanced Motion Vector Prediction (AMVP) feature available in the HEVC codec. The proposed algorithm which operates at the video encoder relies on a tree-based approach to compute all possible motion vectors for a received block. Based on these motion vectors, the expected transmission distortion at the decoder is predicted using a mathematical model presented in the paper. Furthermore, the clipping operation which occurs at the decoder due to erroneous reference pixels is also accounted in the algorithm where the clipping term is first expressed using a mathematical model and later simplified using a set of statistical properties of the video sequence. The second contribution of this work makes use of the aforementioned distortion values to determine robust motion vectors and coding modes. Here the conventional Rate Distortion Optimization (RDO) function in the HEVC encoder is amended to suit with the video transmission over error prone channels. Consequently, the relation between the Lagrangian parameter ( $\lambda$ ) and the Quantization Parameter (QP) is re-expressed in terms of the Packet Error Rate (PER) and a set of video content specific parameters. The proposed algorithm is implemented in HEVC HM reference software and the performance of the proposed method is compared against the state-of-the-art techniques for different PERs, bit rates and content types. The results demonstrate a significant gain in performance in terms of B-D rate for all the scenarios tested.

The remainder of this paper is organized as below. Section II discusses the state-of-the-art video quality prediction and

error resilient algorithms and their limitations in the context of HEVC. Section III illustrates the error propagation scenarios in HEVC and their implications on the user perceived video quality. Mathematical formulations of the video quality prediction algorithms and the refined RDO process is described in Section IV. Later, the simulation environment and the characteristics of the test video sequences are discussed in Section V. Finally, the performance gain of the proposed method is compared against the state-of-the-art techniques in Section VI followed by the concluding remarks in Section VII.

## II. RELATED WORK

Deriving video coding parameters which are robust against network impairments is a popular application layer error resilient technique and the work proposed in this paper also adopts the same technique. In general, the whole process constitutes of three major inter-related sub-processes; namely, the distortion estimation of the decoded video inside the encoder (i.e., end user video quality prediction), deriving the set of robust video coding parameters at the encoder which maximizes the decoded video quality (i.e., redundancy inclusion) and the error concealment at the decoder. Out of these, this section discusses the state-of-the-art in encoder side operations (i.e., first two sub processes) which are relevant to this work with a little emphasis on decoder error concealment.

Assuming the decoder to perform slice copying during a packet loss, Zhihai et al. [3] expressed the overall distortion and thereby the quality of an H.263 decoded video as a combination of the source distortion (i.e., due to lossy video compression) and the channel caused distortion which is introduced by the packet losses in the channel. Later the most robust coding modes are selected by applying these computed distortion terms inside the RDO cost function. Later, the method proposed in [3] was further enhanced in [4] by additionally incorporating the error propagation through the Unconstrained Intra Prediction (UIP) mode available in H.264, where UIP refers to the prediction of I MBs (Macroblocks) using already encoded neighboring P MBs which lie in adjacent video packets. Furthermore, as a consequence of propagated channel error, certain pixel values at the decoder could exceed their nominal range (i.e., for 8 bit precise videos, nominal range of pixels values are between 0 and 255) in which case the decoder performs a clipping operation to retain them within the nominal range. This behavior had not been foreseen by the previous quality prediction methods and was incorporated in the algorithm proposed in [5]. However, the applicability of these algorithms in the context of HEVC is limited due to the fact that these methods do not possess the intelligence to capture the error propagation behavior introduced by the novel components of HEVC such as AMVP. A content-aware distortion model to predict the decoded video quality of an HEVC coded video is proposed by Anekeuh and Ifeachor in [6]. In this model, the video quality is predicted by considering the PER, content type and the QP, and is subsequently claimed to be 92% accurate in general. However the method proposed in [6] has to be deployed after the video encoding process and therefore resilience addition shall take place in a different

operational point (i.e., at the network layer) than at the encoder (i.e., at the application layer) which is fundamentally different from the proposed approach in this work.

In the context of error resilient video encoding, the rational for decoded video quality prediction is to incorporate redundant information at the encoder so that the end user perceived video quality is improved. Out of different types of redundancy inclusion techniques that can be employed, encoding in intra mode is one way of terminating the temporal error propagation [7]. In response to this, Zhihai et al. in [3] and Yang et al. in [8] proposed error resilient video encoding algorithms by incorporating the predicted distortion values inside the RDO function, while assuming slice-copying and motion-copying at the decoder respectively. Furthermore, Crutz et al. in [9] illustrated a mechanism to determine the intra-refresh block addition frequency. There the authors have found that the source distortion does not remain the same for different intra-refresh addition frequencies and therefore, used the mean-absolute-difference of the residual energy to determine the source distortion. The proposed method demonstrates a 1.19 - 1.54 dB PSNR gain with respect to the state-of-the-art methods. Furthermore, Cote et al. in [10] described a mechanism to effectively determine the size of each slice according to the PER in addition to the RDO based intra block addition described above. However, unlike the traditional works which re-used the error free encoding parameters of the RDO process, Zhang et al. in [11] formulated a mathematical relationship to determine the optimal Lagrangian parameter to be used during the video transmission over error prone channels. Consequently the method is claimed to exhibit a noticeable gain over [8]. Moving in the same direction, [12] first utilized the channel caused distortion prediction algorithm in [3] to determine the ideal set of intra-refresh blocks and later to derive the motion vectors so as to identify the locations within the reference frame which carry minimal error propagation. The authors in [12] have incorporated the refined Lagrangian model proposed in [11] during the RDO process. However, we have found that these Lagrangian parameter derivation is still sub-optimal in the context of hyperbolic rate-distortion function used in HEVC and presented a novel QP- $\lambda$  relationship.

With respect to the inclusion of redundancy information in HEVC coded videos, limited works can be found in the literature. From what is available, a temporal domain error propagation analysis and mitigation approach for HEVC coded videos has been presented in [13], and a similar approach which effectively toggles the Temporal Motion Vector Prediction (TMVP) feature in HEVC has been presented in [14], [15]. However, these approaches have not considered the pixel domain error propagation, and as a consequence are ill-suited for the task of terminating the temporal propagation of errors. More recently Carreira et al. in [16] proposed an approach to preserve the motion vectors in an HEVC coded video, again without considering the impact from the pixel errors in the reference frames. Even though the method is fairly simple, it affects severe performance degradation during the transmission over relatively high packet error rate channels as depicted in the results section in this paper.

Therefore, it can be concluded that the existing error resilient algorithms suffer from two major drawbacks making them ill-suited for HEVC based video transmission. Firstly, the existing distortion estimation methods do not have the capacity to identify the errors that would have caused by novel HEVC features such as AMVP. Secondly, the RDO parameters used by the existing methods during the video transmission over error prone channels are sub-optimal in the context of HEVC.

### III. BACKGROUND: SOURCES OF ERRORS AND THEIR PROPAGATION IN HEVC

The near 50% improvement in the coding efficiency of HEVC over its predecessor H.264/AVC (Advanced Video Coding) [17] is achieved via a series of key features such as the Quad-tree Coding Tree Unit (CTU) structure, Advanced Motion Vector Prediction (AMVP), additional intra/inter coding modes and sophisticated filtering techniques (e.g., Sample Adaptive Offset filter (SAO), 8 PAL line filter) [1]. However, this additional complexity of HEVC coded bit streams make them more susceptible to network packet errors when compared to H.264 bit streams [2]. Specifically, AMVP which being dependant on information external to the current decoding unit, can carry forward any prior transmission impairments from previous frames causing significant quality degradation in the subsequent decoding blocks. This section focuses on the background relevant to the AMVP feature in HEVC and the sources of the errors associated with it<sup>1</sup> namely: (i) errors related to motion vectors, (ii) pixel level errors and (iii) errors related to clipping of pixels values.

#### A. Impact of Packet Errors on the Motion Vectors

A consequence of AMVP-based motion vector encoding in HEVC is that motion vectors seen by the decoder may become erroneous due to multiple reasons. As an example, the motion vectors at the decoder can be erroneous although the coding information of the corresponding block is correctly received. The three such scenarios relevant to video transmission over lossy networks are briefly discussed next in decreasing order of their impact.

1) *Erroneous Candidate List Construction*: Motion vectors in HEVC, instead of being sent directly and individually, are derived indirectly from the motion vectors of neighboring blocks using a candidate motion vector and a relative difference to this candidate. Yet, to minimize the number of bits that must be transmitted, HEVC assumes that the decoder can derive the candidate block using a predefined algorithm (known as candidate list generation), and therefore only encodes an index and the relative difference into the bit stream. Naturally, losses of previous coding blocks can result in a different candidate list (to what the encoder expects) being generated by the decoder; thus, the computed motion vector could become significantly different to what was intended.

This behavior, relevant to both the Merge and Motion Vector Prediction modes, is illustrated for a hypothetical scenario in

<sup>1</sup>It is assumed that errors are concealed by the decoder using motion copying throughout the course of this work.

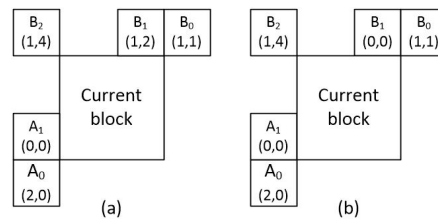


Fig. 1. Spatial candidates and motion vectors at the (a) encoder and (b) decoder after lossy transmission: a candidate list truncation scenario.

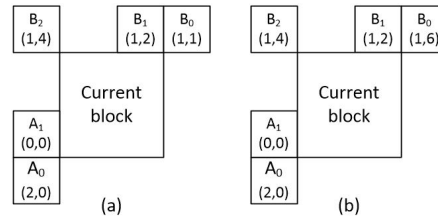


Fig. 2. Spatial candidates and motion vectors at the (a) encoder and (b) decoder after lossy transmission: an erroneous motion vector scenario.

Fig. 1 where block  $B_1$  is incorrectly received. The candidate list generation follows the following process. First, the candidate list must be derived. To this end, the neighboring blocks are traversed in the order  $A_1 \rightarrow B_1 \rightarrow B_0 \rightarrow A_0 \rightarrow B_2$ , and blocks with the same motion vector as a prior block in the candidate list are discarded. Next, the encoder evaluates the least cost motion vector from all members in the candidate list, and encodes its index into the bit stream. Finally, the decoder reverses this process. Yet, it becomes apparent that the member motion vectors in the candidate list (in the initial list generation and due to the truncation of the initial candidate list in particular) may differ from what the encoder intended due to prior errors in the bit stream and their concealment by the decoder. For example since  $A_1 = B_1$  in Fig. 1(b), the previously 5 member candidate list becomes truncated to 4 and the indices of  $B_0$ ,  $A_0$  and  $B_2$  are shifted by one at the decoder. Thus, had the encoder selected  $A_0$ , the decoder will assume  $B_2$  to be correct, potentially resulting in a large error of the motion vector in what was essentially a correctly received coding block at the decoder.

2) *Erroneous Motion Vectors in the Candidate List*: Fig. 2 illustrates a less severe case of motion vector errors to that described previously. Here, the candidate list remains unaltered, but individual motion vectors may be incorrect (e.g.,  $B_0$  differs at the decoder). In this scenario,  $B_0$  may have been derived for example by concealing a lost block using its co-located block in the previous frame. Thus, the motion vector error may be less significant in comparison to the previous scenario, yet once more a moderate error will be observable for an otherwise correctly received coding block at the decoder.

3) *Errors Introduced by Concealment Operations*: In contrast to the previous scenarios where motion vectors became incorrect in otherwise correctly received blocks, the arguably most prevalent and detectable (by the decoder) is the loss of the decoding information for the current block. Typically, in these circumstances decoders apply a concealment operation such as motion-copy concealment [18] (motion vectors of co-

located blocks are reused) and boundary matching algorithms [19] (motion vectors are estimated such that the discrepancy at the boundaries of the lost block are minimized) to reconstruct the lost regions of the frame. Although in general these motion vectors do not exactly match the motion vectors intended by the encoder, the resulting error in an individual frame may not be very significant to the viewer. Crucially however, they can result in significant impact on subsequent coding block, including those correctly received, due to the temporal propagation of the motion vector error, leading to the motion vector error scenarios described previously.

### B. Impact of Packet Errors on Pixel Information

Similar to the motion vector errors, pixel level errors may also occur in both correctly and incorrectly received blocks due to two factors. In the case of the former, they correspond to different motion vectors being observed by the decoder to what was intended by the encoder at any time in the past. Here, the pixels pointed to may differ from the intended, resulting in an erroneously recreated block. In the case of the latter, error concealment techniques can only predict what was intended to be received; thus, the missing residuals will almost certainly be irrecoverable. As a result, for complex content the resulting pixel level errors could lead to significant, visible propagated errors even when errors in the motion vectors are negligible.

### C. Impact of Clipping in Reconstructed Pixels

The decoder clipping operation, which manifests due to a combination of the erroneous motion vectors and pixel information can add errors to the decoded video. Clipping error is merely a product of how decoders are implemented to remain within the valid range of pixel values (e.g., between 0 and 255 for an 8-bit decoder). In fact, clipping operation at the decoder remains identical to that of the encoder in error-free transmissions. This however no longer remains true in lossy transmissions and the incorrect motion vectors and missing residual pixel values can result in inconsistent clipping operations at the encoder and the decoder. Thus, the decoder will clip individual pixels giving rise to what is known as clipping errors [5]. Unlike the previous sources of errors, clipping occurs only in correctly received blocks, yet it too will result in errors being propagated to subsequently decoded blocks.

## IV. END-TO-END DISTORTION MODELLING AND CODING

Section III described how errors due to transmission losses in HEVC can be categorized. Hypothetically, if the encoder could be made aware of where these errors occurred and their impact on the decoded sequence, changes to the coding structure can be envisaged that mitigate the impact of errors. To this end, it is necessary to model the distortion introduced by errors during transmission from the encoder to the decoder, i.e., the channel distortion, in addition to the source distortion (the differences in the compressed and original pixel values) already known by the encoder. Furthermore, the channel itself behaves randomly, leading to a varying severity of the channel

TABLE I  
TERMS AND NOTATIONS

Definition	Notation
Packet Error Rate (PER)	$p$
Current frame	$n$
Location of an arbitrary pixel in the current frame	$i$
Original pixel value of pixel $i$	$X_n^i$
Error-free pixel value of pixel $i$	$\hat{X}_n^i$
Erroneous pixel value of pixel $i$	$\tilde{X}_n^i$
Residual of pixel $i$	$r_n^i$
Reference frame of pixel $i$	$n_0$
Location of the reference pixel of pixel $i$	$j$
Squared error component in the channel: $D_{CH}^{SE}(\cdot)$	$E \left[ \left( \hat{X}_n^i - \tilde{X}_n^i \right)^2 \right]$
Difference component in the channel: $D_{CH}^D(\cdot)$	$E \left[ \hat{X}_n^i - \tilde{X}_n^i \right]$
Squared error component at the source: $D_{SRC}^{SE}(\cdot)$	$\left( X_n^i - \hat{X}_n^i \right)^2$
Difference component at the source: $D_{SRC}^D(\cdot)$	$X_n^i - \hat{X}_n^i$
Expectation operator: Expected value for a PER $p$	$E[\cdot]$
Clipping operation performed on pixel $i$	$\Gamma(X_n^i)$
Motion compensation operator of motion vector $u$	$k(u)$

induced distortion. This section describes how this channel distortion can be modelled such that the encoding process can be modified to mitigate the effects of transmission errors.

Table I summarizes the terms and notations used in this work. For a pixel  $i$  in frame  $n$ , the reference pixel and reference frame are denoted by  $j$  and  $n_0$ , respectively. Furthermore, for some packet error rate  $p$ , the original, error-free (at the encoder) and erroneous (i.e., decoded and concealed) pixel values at the location  $i$  are given by  $X_n^i$ ,  $\hat{X}_n^i$  and  $\tilde{X}_n^i$ , respectively. Thus, the error-free pixel value can also be expressed as  $\hat{X}_n^i = \hat{X}_{n_0}^j + r_n^i$ , where  $r_n^i$  is the residual contribution to the pixel location  $i$ .

### A. Modelling Distortion Observed by the Decoder

The total distortion observed at the decoder can be expressed as Squared Error (SE) between the original and displayed (including error concealment after decoding where needed) pixel values. Here, the contribution of the individual pixels can be evaluated separately; thus, the contribution of an arbitrary pixel to the squared error can be defined as  $D^{SE}(X_n^i)$ , in (1), using the notation in Table I.

The observed distortion at the decoder consists of three components; a source distortion (a squared error) corresponding to the quantization error at the encoder, channel distortion (a squared error) due to packet errors in the transmission, and a product of the difference terms of the source and channel components. The random nature of packet errors and their impact on the decoded pixels are modelled using the expectation operation  $E(\cdot)$ ; thus,  $D_{SRC}^D$  being channel independent, the difference components  $D_{SRC}^D$  and  $D_{CH}^D$  become independent and are expressed as such to obtain (2).

### B. Channel Distortion from Motion Vector and Pixel Errors

HEVC, as with its predecessors, carries motion vectors, coding parameters and residual information of CTUs (Coding

$$D^{\text{SE}}(X_n^i) \triangleq \text{E} \left[ \left( X_n^i - \tilde{X}_n^i \right)^2 \right] = \text{E} \left[ \left( X_n^i - \hat{X}_n^i + \hat{X}_n^i - \tilde{X}_n^i \right)^2 \right] \quad (1)$$

$$\begin{aligned} &= \left( X_n^i - \hat{X}_n^i \right)^2 + \text{E} \left[ \left( \hat{X}_n^i - \tilde{X}_n^i \right)^2 \right] + 2 \left( X_n^i - \hat{X}_n^i \right) \text{E} \left[ \hat{X}_n^i - \tilde{X}_n^i \right] \\ &= D_{\text{SRC}}^{\text{SE}}(X_n^i) + D_{\text{CH}}^{\text{SE}}(X_n^i) + 2 D_{\text{SRC}}^{\text{D}}(X_n^i) D_{\text{CH}}^{\text{D}}(X_n^i) \end{aligned} \quad (2)$$

$$D_{\text{CH}}^{\text{SE}}(X_n^i) \triangleq \text{E} \left[ \left( \hat{X}_n^i - \tilde{X}_n^i \right)^2 \right] = (1-p) \text{E} \left[ \left( \hat{X}_{n_0}^j + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^j + r_n^i \right\} \right)^2 \right] + p \text{E} \left[ \left( \hat{X}_{n_0}^j + r_n^i - \tilde{X}_{n-1}^j \right)^2 \right] \quad (3)$$

$$D_{\text{CH}}^{\text{D}}(X_n^i) \triangleq \text{E} \left[ \hat{X}_n^i - \tilde{X}_n^i \right] = (1-p) \text{E} \left[ \hat{X}_{n_0}^j + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^j + r_n^i \right\} \right] + p \text{E} \left[ \hat{X}_{n_0}^j + r_n^i - \tilde{X}_{n-1}^j \right] \quad (4)$$

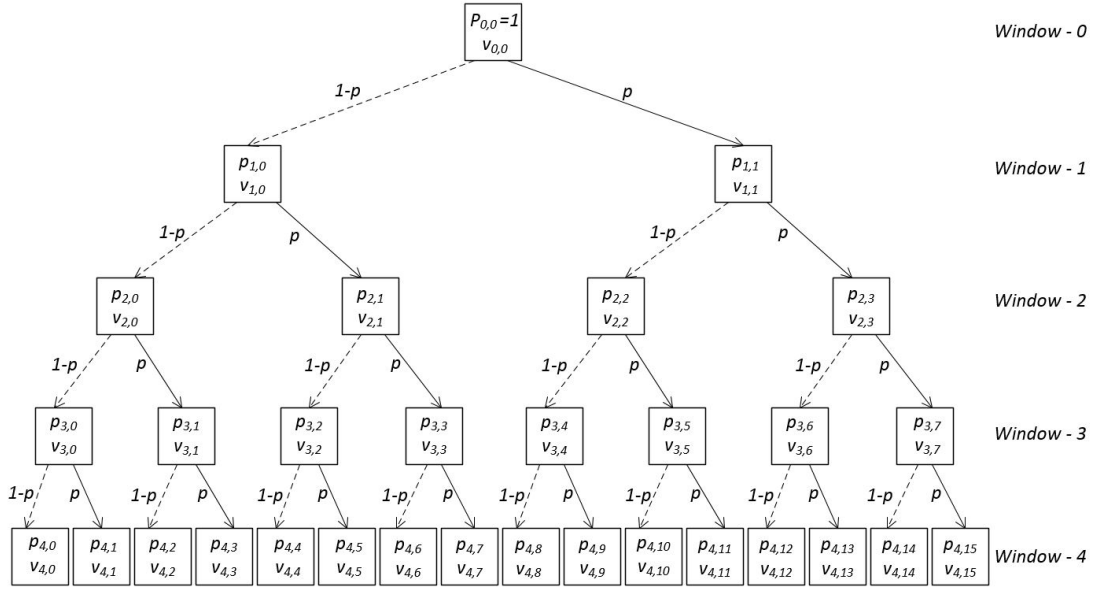


Fig. 3. A tree of potential motion vectors and their likelihoods with respect to a known reference at the 0<sup>th</sup> window level.

Tree Units) in a single abstracted unit known as a NAL (Network Abstraction Layer) unit. Hence, the loss of a transport layer packet during transmission compromises each element. Bearing this in mind, if a single NAL unit can be encapsulated in one transport layer packet, the channel component of the distortion observed at the decoder can be defined as in (3). Expending on the initial definition therein, for a PER  $p$  (if the state of the reference frame  $n_0$  is known), the likelihoods of the NAL unit containing  $X_n^i$  being correctly received or lost is  $(1-p)$  and  $p$ , respectively. Thus, since  $\tilde{X}_n^i \triangleq \hat{X}_{n_0}^j + r_n^i$ , the resultant channel distortion in each case can be computed, and the expected channel distortion can be expressed with respect to its reference frame  $n_0$  as the probabilistically weighted sum in (3). Note that the residual  $r_n^i$  is only applied to the correctly received scenario and is subsequently clipped by  $\Gamma$ , and  $\tilde{X}_{n_0}^j$ ,  $\tilde{X}_{n-1}^j$  denote the motion-compensated and concealed<sup>2</sup> pixel values relevant to the perfectly received and lossy scenarios,

<sup>2</sup>This work applies motion coping as its preferred means of error concealment due to its relatively greater capability of approximating both lost pixel and motion vector information at a low computational complexity. However, it should be noted that the proposed distortion estimation framework is independent of the error concealment technique applied here, and that any error commencement operation can be incorporated in its place in (3).

respectively. Next, how these can also be incorporated into the end-to-end distortion model is discussed.

First, in order to visualize the manner errors propagate and to aid the following discussions, consider the diagram in Fig. 3 where a tree diagram illustrates all possible states of motion vectors and the likelihood of each state with respect to a known reference (i.e., the 0<sup>th</sup> level)<sup>3</sup>, over an interval of 4 video frames. The dashed lines here represent correctly received paths and the solid lines represent the paths where NAL units were lost. Further, the motion vectors in each node may refer to any prior frame in the decoding order; hence, the nodes in the diagram do not reflect any frame ordering as such. For example, when connected by a solid line, the parent node corresponds to the reference frame of the child node, whereas for a dashed line, it corresponds to the immediately previous frame (in fact, this may not have been its reference frame in a

<sup>3</sup>Being that the transmission channel conditions are subject to change and that the proposed distortion model is used for error resilience inclusion in real-time applications, the existence of a feedback channel as used in RTSP [20] that notifies the encoder of its current state, e.g., which NAL units were not received, is assumed. As such, the 0<sup>th</sup> window level corresponds to this last known state and the window length is determined by the latency of the feedback channel.

$$D_{\text{CH}}^{\text{SE}}(X_n^i) = \sum_{\forall u \in \mathcal{U}_1} P(u) \mathbb{E} \left[ \left( \hat{X}_{n_0}^j + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right)^2 \right] + \sum_{\forall u \in \mathcal{U}_2} P(u) \mathbb{E} \left[ \left( \hat{X}_{n_0}^j + r_n^i - \tilde{X}_{n-1}^{k(u)} \right)^2 \right] \quad (5)$$

$$= \sum_{\forall u \in \mathcal{U}_1} P(u) \mathbb{E} \left[ \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} + \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right)^2 \right] \\ + \sum_{\forall u \in \mathcal{U}_2} P(u) \mathbb{E} \left[ \left( \hat{X}_{n_0}^j + r_n^i - \hat{X}_{n-1}^{k(u)} + \hat{X}_{n-1}^{k(u)} - \tilde{X}_{n-1}^{k(u)} \right)^2 \right] \quad (6)$$

$$D_{\text{CH}}^{\text{SE}}(X_n^i) = \sum_{\forall u \in \mathcal{U}_1} P(u) \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right)^2 + \sum_{\forall u \in \mathcal{U}_1} P(u) \mathbb{E} \left[ \left( \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right)^2 \right] \\ + \sum_{\forall u \in \mathcal{U}_1} 2P(u) \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) \mathbb{E} \left[ \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right] + \sum_{\forall u \in \mathcal{U}_2} P(u) \left( \hat{X}_{n_0}^j + r_n^i - \hat{X}_{n-1}^{k(u)} \right)^2 \\ + \sum_{\forall u \in \mathcal{U}_2} P(u) \mathbb{E} \left[ \left( \hat{X}_{n-1}^{k(u)} - \tilde{X}_{n-1}^{k(u)} \right)^2 \right] + \sum_{\forall u \in \mathcal{U}_2} 2P(u) \left( \hat{X}_{n_0}^j + r_n^i - \hat{X}_{n-1}^{k(u)} \right) \mathbb{E} \left[ \hat{X}_{n-1}^{k(u)} - \tilde{X}_{n-1}^{k(u)} \right] \quad (7)$$

$$D_{\text{CH}}^{\text{SE}}(X_n^i) = D_{\text{Refer}}^{\text{SE}}(X_n^i) + DP_{\text{Refer}}^{\text{SE},\Gamma}(X_n^i) + DX_{\text{Refer}}^{\text{D},\Gamma}(X_n^i) + D_{\text{Conceal}}^{\text{SE}}(X_n^i) + DP_{\text{Conceal}}^{\text{SE}}(X_n^i) + DX_{\text{Conceal}}^{\text{D}}(X_n^i) \quad (8)$$

$$D_{\text{CH}}^{\text{D}}(X_n^i) = \sum_{\forall u \in \mathcal{U}_1} P(u) \mathbb{E} \left[ \hat{X}_{n_0}^j + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right] + \sum_{\forall u \in \mathcal{U}_2} P(u) \mathbb{E} \left[ \hat{X}_{n_0}^j + r_n^i - \tilde{X}_{n-1}^{k(u)} \right] \quad (9)$$

$$= \sum_{\forall u \in \mathcal{U}_1} P(u) \mathbb{E} \left[ \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} + \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right] \\ + \sum_{\forall u \in \mathcal{U}_2} P(u) \mathbb{E} \left[ \hat{X}_{n_0}^j + r_n^i - \hat{X}_{n-1}^{k(u)} + \hat{X}_{n-1}^{k(u)} - \tilde{X}_{n-1}^{k(u)} \right] \quad (10)$$

$$= \sum_{\forall u \in \mathcal{U}_1} P(u) \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) + \sum_{\forall u \in \mathcal{U}_1} P(u) \mathbb{E} \left[ \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right] \\ + \sum_{\forall u \in \mathcal{U}_2} P(u) \left( \hat{X}_{n_0}^j + r_n^i - \hat{X}_{n-1}^{k(u)} \right) + \sum_{\forall u \in \mathcal{U}_2} P(u) \mathbb{E} \left[ \hat{X}_{n-1}^{k(u)} - \tilde{X}_{n-1}^{k(u)} \right] \quad (11)$$

$$D_{\text{CH}}^{\text{D}}(X_n^i) = D_{\text{Refer}}^{\text{D}}(X_n^i) + DP_{\text{Refer}}^{\text{D},\Gamma}(X_n^i) + D_{\text{Conceal}}^{\text{D}}(X_n^i) + DP_{\text{Conceal}}^{\text{D}}(X_n^i) \quad (12)$$

$$DP_{\text{Refer}}^{\text{SE},\Gamma} + DX_{\text{Refer}}^{\text{D},\Gamma} = DP_{\text{Refer}}^{\text{SE}} + DX_{\text{Refer}}^{\text{D}} + \sum_{\forall u \in \mathcal{U}_1} P(u) \left\{ \mathbb{E} \left[ \mathcal{A} \left( r_n^i + 2\hat{X}_{n_0}^j - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} - \tilde{X}_{n_0}^{k(u)} \right) \right] \right\} \quad (13)$$

non-erroneous scenario, but the concealment operation applied here ensures that a lost region always refers to its immediately previous frame). Now, if the set of leaf nodes in the tree at the window length  $L$  are defined as  $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$ , where

$$\mathcal{U}_1 = \{u_{L,1}, u_{L,3}, \dots, u_{L,W-1}\} \mid W = 2^L \quad (14)$$

$$\mathcal{U}_2 = \{u_{L,2}, u_{L,4}, \dots, u_{L,W}\} \mid W = 2^L, \quad (15)$$

the motion vectors and probability of each leaf occurring become functions of the leaf location  $u \in \mathcal{U}$  given by

$$V(u) \in \{V(u_{L,1}), V(u_{L,2}), \dots, V(u_{L,W})\}$$

and

$$P(u) \in \{P(u_{L,1}), P(u_{L,2}), \dots, P(u_{L,W})\},$$

respectively. For notational simplicity in the derivations:

**Definition:** Let  $k(u)$  refer to a pixel location indicated by the motion vector  $V(u)$  and  $P(u) = (1-p)^a p^b$  for  $a, b \in \{0, \dots, L\} \mid a + b = L$ .

From Fig. 3, it becomes clear that the combination of motion vectors that can be received at the decoder increase rapidly with the window length  $L$ . Naturally, it is also apparent that (3) only applies to the simplest case where  $L = 1$ .

The two scenarios where the currently encoded block is received or lost in (3) can be generalized to an arbitrary window length  $L$ , such that the squared error component in the channel can be expressed as shown in (5)-(8). Equation (8) illustrates how the channel distortion  $D_{\text{CH}}^{\text{SE}}$  can be expressed as six individual distortion terms. In the order that they appear, each summation term corresponds to the following:

- $D_{\text{Refer}}^{\text{SE}}$  - Distortion from referring to incorrect reference pixels; occurs due to accumulated motion vector errors.
- $DP_{\text{Refer}}^{\text{SE},\Gamma}$  - Propagated distortions via the reference pixel;

- occur due to the combination of pixel errors and clipping.
- $DX_{\text{Refer}}^{\text{D},\Gamma}$  - Cross component of the reference pixel and propagated errors above, including the impact of clipping.
- $D_{\text{Conceal}}^{\text{SE}}$  - Distortion due to the error concealment operation; the magnitude of which is algorithm dependent.
- $DP_{\text{Conceal}}^{\text{SE}}$  - Propagated distortion via the concealment pixels in the  $(n-1)$ <sup>th</sup> frame. Note that  $DP_{\text{Conceal}}^{\text{SE}}(X_n^i) \triangleq \sum_{\forall u \in \mathcal{U}_2} P(u) D_{\text{CH}}^{\text{SE}}(X_{n-1}^{k(u)})$ .
- $DX_{\text{Conceal}}^{\text{D}}$  - Cross component of the concealment and the propagated concealment errors above.

Similarly, the difference component in (4) can be generalized as shown in (9)-(12). Equation (12) illustrates how  $D_{\text{CH}}^{\text{D}}$  can now be expressed as the sum of four difference terms. Once again, in the order that they appear, summation each term corresponds to the following:

- $D_{\text{Refer}}^{\text{D}}$  - Difference error caused by erroneous reference pixel locations due to motion vector errors.
- $DP_{\text{Refer}}^{\text{D},\Gamma}$  - Propagated difference error via the reference pixels due to pixel errors and clipping.
- $D_{\text{Conceal}}^{\text{D}}$  - Difference error due to the error concealment operation; magnitude of this error is algorithm dependent.
- $DP_{\text{Conceal}}^{\text{D}}$  - Propagated difference error via the concealment pixels in the  $(n-1)$ <sup>th</sup> frame. This is also equivalent to  $DP_{\text{Conceal}}^{\text{D}}(X_n^i) \triangleq \sum_{\forall u \in \mathcal{U}_2} P(u) D_{\text{CH}}^{\text{D}}(X_{n-1}^{k(u)})$ .

Consequently, it becomes clear that the overall channel induced distortion is broadly comprised of motion vector errors in the current frame, concealment errors in the current frame and propagated errors from previous frames. For example, in (8) and (12),  $D_{\text{Refer}}^{\text{SE}}$ ,  $D_{\text{Conceal}}^{\text{SE}}$ ,  $D_{\text{Refer}}^{\text{D}}$  and  $D_{\text{Conceal}}^{\text{D}}$  can be derived directly for the  $n$ <sup>th</sup> frame, while  $DP_{\text{Conceal}}^{\text{SE}}$ ,  $DX_{\text{Conceal}}^{\text{D}}$  and  $DP_{\text{Conceal}}^{\text{D}}$  can be computed iteratively from the  $(n-L)$ <sup>th</sup> frame. However, the propagated errors via the clipped pixel, the terms corresponding to  $DP_{\text{Refer}}^{\text{SE},\Gamma}$ ,  $DX_{\text{Refer}}^{\text{D},\Gamma}$  and  $DP_{\text{Refer}}^{\text{D},\Gamma}$ , can have a significant impact on the overall distortion; hence, modelling the impact of those terms is essential. How these terms can be computed and approximated is described next.

### C. Channel Distortion due to Clipped Pixels

One approach to assessing the impact of clipping on  $DP_{\text{Refer}}^{\text{SE},\Gamma}$ ,  $DX_{\text{Refer}}^{\text{D},\Gamma}$  and  $DP_{\text{Refer}}^{\text{D},\Gamma}$  would be to compute its impact at each node of the motion vector tree in Fig. 3 using a brute-force approach to the problem. However, this quickly becomes impractical with increasing  $L$ , and a simpler model of its behavior becomes indispensable. To this end, these terms must first be simplified further as described in Appendix A.

Following from (31) in Appendix A, the summation of the clipped terms in (8) can be expressed as given in (13), where

$$\mathcal{A} \triangleq \tilde{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\}$$

$$DX_{\text{Refer}}^{\text{D}}(X_n^i) \triangleq \sum_{\forall u \in \mathcal{U}_1} 2P(u) \left( \hat{X}_{n_0}^j - \tilde{X}_{n_0}^{k(u)} \right) D_{\text{CH}}^{\text{D}}(X_{n_0}^{k(u)})$$

and

$$DP_{\text{Refer}}^{\text{SE}}(X_n^i) \triangleq \sum_{\forall u \in \mathcal{U}_1} P(u) D_{\text{CH}}^{\text{SE}}(X_{n_0}^{k(u)}).$$

Since the error pattern will always be unknown, a statistical approach that considers the properties of the input video

TABLE II  
MODEL PARAMETERS FOR DISTORTION ESTIMATION

Format	$\alpha_1$	$\alpha_2$
CIF	0.0459	2.8890
HD	0.0116	0.7216

sequence is adopted to estimate and quantify the impact of clipping. Intuitively, for a general case, the absolute magnitude of the residual signal can be seen to influence the likelihood of clipping, i.e., large residuals are more likely to clip. Further, at the upper and lower extremes of the pixel value range, relatively smaller residuals are also likely to clip, but the error itself will be proportional and of a similar order to the residual. Hence, in this work, the following approximation is assumed to be an analogue of the clipping behavior;

$$\mathcal{A} \triangleq \tilde{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \approx \gamma_n r_n^i, \quad (16)$$

where  $\gamma_n$  is an experimentally derived constant.  $\gamma_n$  itself is modelled by analyzing multiple videos with different motion characteristics, and the following model was found to be sufficiently accurate. Here,  $\gamma_n = \frac{\alpha_1}{\sigma_n} (\mu_n + \alpha_2)$ , where  $\alpha_1$ ,  $\alpha_2$  are resolution-dependent constants, and  $\mu_n$  and  $\sigma_n$  are the mean and the standard deviation of original motion vectors in the  $n$ <sup>th</sup> frame. Table II summarizes the model parameters obtained for different video formats. The remaining clipping term in (13) can be approximated in a similar fashion by incorporating the definition in (12) as,

$$r_n^i + 2\hat{X}_{n_0}^j - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} - \tilde{X}_{n_0}^{k(u)} \approx \gamma_n r_n^i + 2 \left( \hat{X}_{n_0}^j - \tilde{X}_{n_0}^{k(u)} \right). \quad (17)$$

Substituting (16) and (17) where appropriate, (13) can be simplified as shown in (18)-(19).

In a similar fashion to the clipped squared error components in (8), from (37) in Appendix A, the clipped term of the difference error in (12) can be simplified further using the assumption in (16). As a result,

$$DP_{\text{Refer}}^{\text{D},\Gamma} = DP_{\text{Refer}}^{\text{D}} + (1-p) \gamma_n r_n^i, \quad (20)$$

where

$$DP_{\text{Refer}}^{\text{D}} \triangleq \sum_{\forall u \in \mathcal{U}_1} P(u) D_{\text{CH}}^{\text{D}}(X_{n_0}^{k(u)}) \quad (21)$$

and  $\sum_{\forall u \in \mathcal{U}_1} P(u) = (1-p)$ . At this point, each of the components in (8) and (12) can be computed with the aid of the approximations in (19) and (20). However, some components must still be computed recursively over the window length  $L$ .

### D. Error-Compensated Rate-Distortion Optimization

In the error-free HM based HEVC encoding [21], encoding parameters (Quantization Parameter (QP), coding tree structure, transform block structure, prediction block structure, motion vectors) are selected by minimizing the block-level Rate-Distortion (R-D) cost function expressed in (22),

$$\min_{m \in \mathcal{M}} J_{\text{RD}} \left| J_{\text{RD}} \triangleq D(m) + \lambda_0 R(m), \quad (22)$$

$$DP_{\text{Refer}}^{\text{SE},\Gamma}(X_n^i) + DX_{\text{Refer}}^{\text{D},\Gamma}(X_n^i) = DP_{\text{Refer}}^{\text{SE}}(X_n^i) + DX_{\text{Refer}}^{\text{D}}(X_n^i) + \gamma_n r_n^i \sum_{\forall u \in \mathcal{U}_1} P(u) \left\{ \gamma_n r_n^i + 2 \mathbb{E} \left[ \hat{X}_{n_0}^j - \tilde{X}_{n_0}^{k(u)} \right] \right\} \quad (18)$$

$$= DP_{\text{Refer}}^{\text{SE}}(X_n^i) + DX_{\text{Refer}}^{\text{SE}}(X_n^i) + (1-p)(\gamma_n r_n^i)^2 + 2(\gamma_n r_n^i) \{D_{\text{Refer}}^{\text{D}}(X_n^i) + DP_{\text{Refer}}^{\text{D}}(X_n^i)\} \quad (19)$$

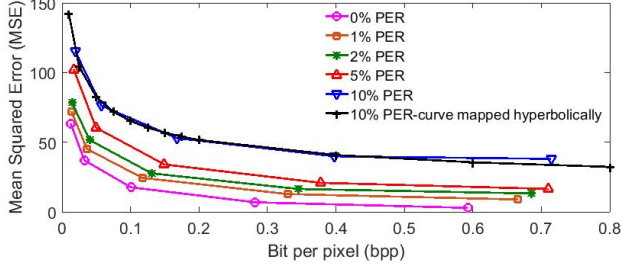


Fig. 4. R-D curves for *Musicians* sequence

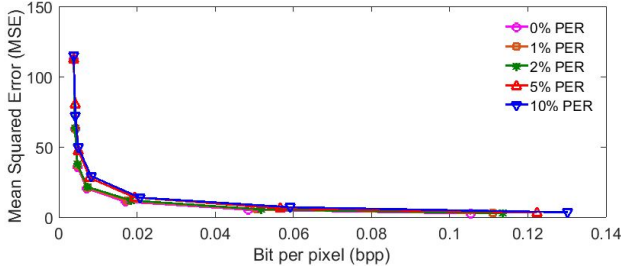


Fig. 5. R-D curves for *Akiyo* sequence

where  $m$  is a coding mode decision from the set of all possible coding modes  $\mathcal{M}$ ,  $D$  is the distortion of the block expressed in terms of Mean Squared Error (MSE),  $R$  is the bit rate consumed by that particular block and the Lagrangian parameter  $\lambda_0 \triangleq -\frac{\partial D}{\partial R}$ . During error free video transmissions,  $R$  and  $D$  forms a hyperbolic relationship as in (23),

$$D(R) = CR^{-K} \quad (23)$$

where  $C$  and  $K$  are model parameters which reflect the characteristics of the video sequence and are determined on the fly using a content learning approach [22]. Interestingly the same hyperbolic relationship between the rate and the distortion holds true even under erroneous HEVC transmission. For illustration purposes the particular behavior is depicted in Fig. 4 for a 10% packet error rate (PER) whereas the behaviour is similar for any other PER as well. Fig. 4 is obtained by plotting R-D curves for 5 different QP values in the range 20-40 and, 4 different packet error rates in the range 1%-10% for *Musicians* video sequence. Therefore unlike in previous codecs,  $\lambda$  determination does not require additional modifications during the constant rate HEVC coding even under erroneous transmissions (Since the relationship between  $R$  and  $D$  is still hyperbolic).

However, the QP determines the quantization step size used to encode Discrete Cosine Transform (DCT) coefficients and therefore has a significant bearings on the total bit rate consumption and the output video quality. Furthermore unless multiple QP optimization is used, the standard HM encoder

assumes a generic exponential relationship between the QP and the  $\lambda$  irrespective of the video characteristics [23]. i.e.,

$$QP = 4.2005 \times \ln(\lambda_0) + 14.0122 \quad (24)$$

Multiple experiments are conducted to evaluate the validity of (24) during the video transmission over error-prone channels, where  $D$  in (22) is replaced by  $D^{\text{SE}}$  in (1) to select robust coding parameters in an R-D sense. These parameters include coding mode decisions (adding intra blocks in place of inter blocks), motion vectors pointing to reference blocks which carry minimal channel errors and the block boundaries (hence the block sizes) which are easily concealable during packet losses. The experiments suggest that (24) is sub-optimal in those conditions. Following observations were made during the conducted experiments. First, in general for a larger QP value, a lower  $\lambda$  value than the one that is suggested by (24) demonstrates better R-D performances. The explanations for the above behaviors can be summarized as below. First as illustrated in Fig. 4, it can be observed that for a given QP value, as the packet error rate increases, R-D curves shifts horizontally in the R-D plane urging more bits for the same quality. Furthermore it can also be observed that the difference in  $\lambda$ s (change in slopes of the R-D curve) between the error free scenario and the erroneous scenarios, gradually decreases along with the increase in bit rate. Consequently, for larger QPs (corresponding to lower bit rates), lower  $\lambda$  values than the one that are given by (24) should be used in the Rate Distortion Optimization (RDO) process during erroneous scenarios. Second, for low motion video sequences, a higher  $\lambda$  value than the one given by (24) should be used during the RDO. The particular behaviour is illustrated in Fig. 5 for *Akiyo* video sequence using 5 QPs and 4 PERs. The change in slopes between the error free R-D curves and erroneous R-D curves are much larger in Fig. 5 specifically for low bit rates. Therefore, higher  $\lambda$  values should be selected for such video sequences for much larger range of bit rates.

Therefore, in order to reflect the above behaviors during the QP derivation process, (25) is proposed in place of (24) for the video compression targeting the error-prone channels.

$$QP = a \times \ln(\lambda_0) + 14.0122 \quad (25)$$

where,

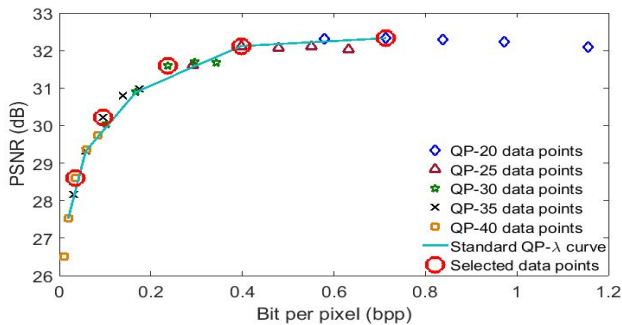
$$a = m \times p + c \quad (26)$$

$$m = \beta_1 / (\sigma_n + \beta_2) + \beta_3 \times \mu_n + \beta_4 \quad (27)$$

$$c = \nu_1 / (\sigma_n + \nu_2) + \nu_3 \times \mu_n + \nu_4 \quad (28)$$

$p$  is the PER,  $\mu_n$ ,  $\sigma_n$  are the mean length and the standard deviation of the motion vectors in frames 0 to n,  $\beta_1, \beta_2, \beta_3, \beta_4$  and  $\nu_1, \nu_2, \nu_3, \nu_4$  are resolution dependent parameters.



Fig. 6. Best QP- $\lambda$  data point selection for *Musicians* at 10%

The derivation process of the model parameters in (25) - (28) is as follow. For each QP value, a set of  $\lambda$  values which are spanned around the original  $\lambda$  value given by (24) are plotted. Later the best  $\lambda$  value for a given QP is selected by identifying the data point which is furthestmost above the standard QP- $\lambda$  relationship (i.e., the curve corresponding to (24)). In the meantime, the selected data points each corresponding to a unique QP value, should also span the whole operational bit per pixel (bpp) range. The behavior for the *Musicians* sequence for 10% PER is shown in Fig. 6. Finally the selected data points are mapped to appropriate QP values using the mapping function in (25). The content dependant variable  $a$  in (25) is derived by employing a curve fitting approach. Motion vectors statistics and PER are used as the input to the fitting criteria. Table III summarizes the model parameters obtained for different video formats. Therefore, the mapping function in (25) is used throughout this work in place of (24).

### V. SIMULATION ENVIRONMENT

The proposed distortion estimation algorithm and RDO algorithms are implemented in HM16.2 HEVC reference software [21]. Additionally three state-of-the-art error resilience and distortion estimation algorithms ([16], [3] and [5]) are also implemented within the same simulation environment. Video sequences with different texture characteristics, motion characteristics and spatial resolutions are tested during the performance evaluation as described here. Out of the Full HD ( $1920 \times 1080$ ) video sequences tested, both the background and the objects are moving in *BMX\_2*, *Musicians*, *Park\_Scene* and *Basketball\_Drive*; Objects are relatively stationary while the background is moving (camera panning) in *Kimono* and *BQTerrace* sequences; Objects are moving in a stationary background in *Cafe* and *Cactus* video sequences while *Beer\_garden* is a slow moving video sequence with large amount of texture information. Finally *Dancer* and *GT\_Fly* are two synthetic Full HD sequences with camera zooming. Similarly, out of the CIF ( $352 \times 288$ ) video

TABLE III  
MODEL PARAMETERS FOR PROPOSED QP - $\lambda$  RELATIONSHIP

	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\nu_1$	$\nu_2$	$\nu_3$	$\nu_4$
CIF	-28.7	76.22	0.002	0.308	0.740	-2.6	0.000	3.728
HD	1.112	0.409	0.006	-0.2	45.860	5.359	0.104	0.315

TABLE IV  
HEVC SIMULATION PARAMATERS

Configuration Parameter	Value
Group of Picture (GOP) Size	20
Number of encoded frames (HD)	100
Number of encoded frames (CIF)	300
Frame rate	25 fps
HEVC configuration	<i>encoder_lowdelay_P_main</i>
HEVC profile	<i>main</i>
Number of CTUs per slice (HD)	30
Number of CTUs per slice (CIF)	6

sequences tested, *Stefan* and *Football* are two fast sequences having both moving background and moving objects; *Bus*, *Coastguard* and *Flower* are fast camera panning sequences; *Foreman* and *Carphone* are averagely fast sequences with relatively stationary objects and finally *Akiyo*, *Highway* and *Container* are slow moving sequences. Table IV summarizes the simulation configurations used in these comparisons. All the videos are encoded such that a row of CTUs occupies the payload of a single slice; therefore, 30 CTUs per slice in Full HD video sequences and 6 CTUs per slice in CIF video sequences. During the experiment, the compressed video is assumed to be transmitted over an erro prone wireless channel having 1%, 2%, 5% and 10% packet error rates.

### VI. RESULTS AND DISCUSSION

The accuracy of the proposed distortion estimation algorithm and the effectiveness of the overall error resilient framework is discussed separately in below two subsections.

#### A. Accuracy of Distortion Estimation

The accuracy of the proposed distortion estimation algorithm is compared against two state-of-the-art methods [3], [5] which do not account for the AMVP features available in HEVC. Furthermore, in order to visualize the ability of the proposed method to reach the theoretically best solution, a brute-force distortion estimation algorithm is also incorporated. Here, the brute-force algorithm is implemented by first computing all probable pixel maps in each leaf in every branch in the probability tree and then taking the weighted average of the maps to compute the final predicted picture and finally calculating the distortion values between the original and the above predicted pictures. Fig. 7 illustrates the absolute difference of predicted distortion values and the brute-force distortion values obtained for the proposed method and the state-of-the-art methods at 10% packet error rate for *GT\_Fly* video sequence. Furthermore, the same measurements are plotted for the actual distortion values (i.e, as seen by the decoder) in Fig. 7. The corresponding measurement can be expressed as,  $abs(DIST_1 - DIST_2) / DIST_1$ . Where  $DIST_1$  is the per CTU average distortion value predicted by the brute-force method and  $DIST_2$  is the per CTU average distortion value predicted by the method of interest. It should be noted that, here the CTU based measurements are taken mainly as these values are re-used inside the CTU based RDO cost function as will be described in the next sub-section.

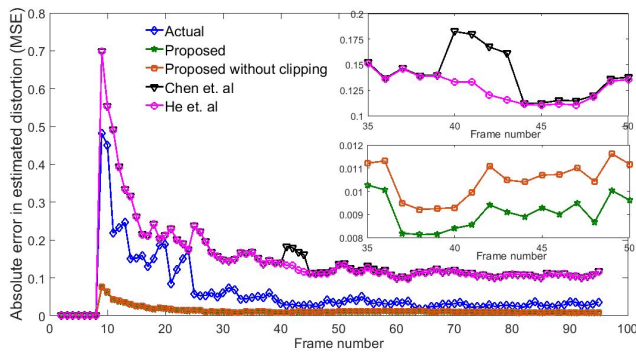


Fig. 7. Absolute error in estimated distortion with respect to the brute-force distortion for *GT\_Fly* sequence at 10% PER

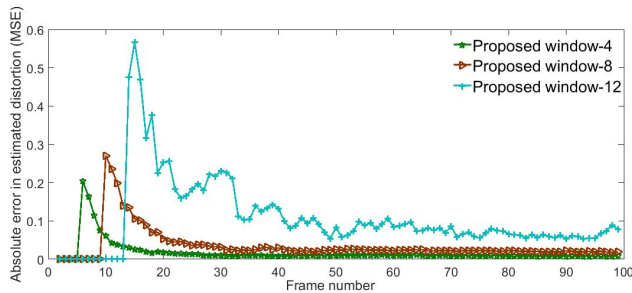


Fig. 8. Impact of window-length on the estimated distortion for *GT\_Fly* sequence

As can be seen in Fig. 7, the distortion values of the proposed method has been very closer to the brute force values and as more frames are encoded, the accuracy of the proposed method has become practically similar to the corresponding brute-force distortion values. The main reason for the superior performances of the proposed method can be considered as the incorporation of the AMVP techniques available in HEVC during the distortion prediction stage. As further illustrated in Fig. 7, the impact of incorporating the clipping compensation term has further improved the accuracy of the distortion estimation. In the meantime, the state-of-the-art methods have not been sufficiently accurate during the distortion estimation. Out of the two state-of-the-art methods presented, [3] has been more accurate compared to [5] specially at higher frame numbers. The clipping compensation term which is introduced in [5] for a context different from the HEVC, has negatively impacted.

Fig. 8 illustrates the performance of the proposed method for different window lengths for *GT\_Fly* video sequence. As expected, as the window-length increases the performance of the proposed method has decreased. This is simply due to the fact that the encoder becomes less informative as the window length increases and consequently, the prediction also becomes less accurate. Interestingly as the frame number increases, the prediction error gets reduced and finally becomes stable for each window-length. This can be explained by the fact that, as many number of frames are encoded, the content dependent parameters become more accurate and consequently the prediction model becomes more and more intelligent. Therefore, the overall distortion values become much closer to

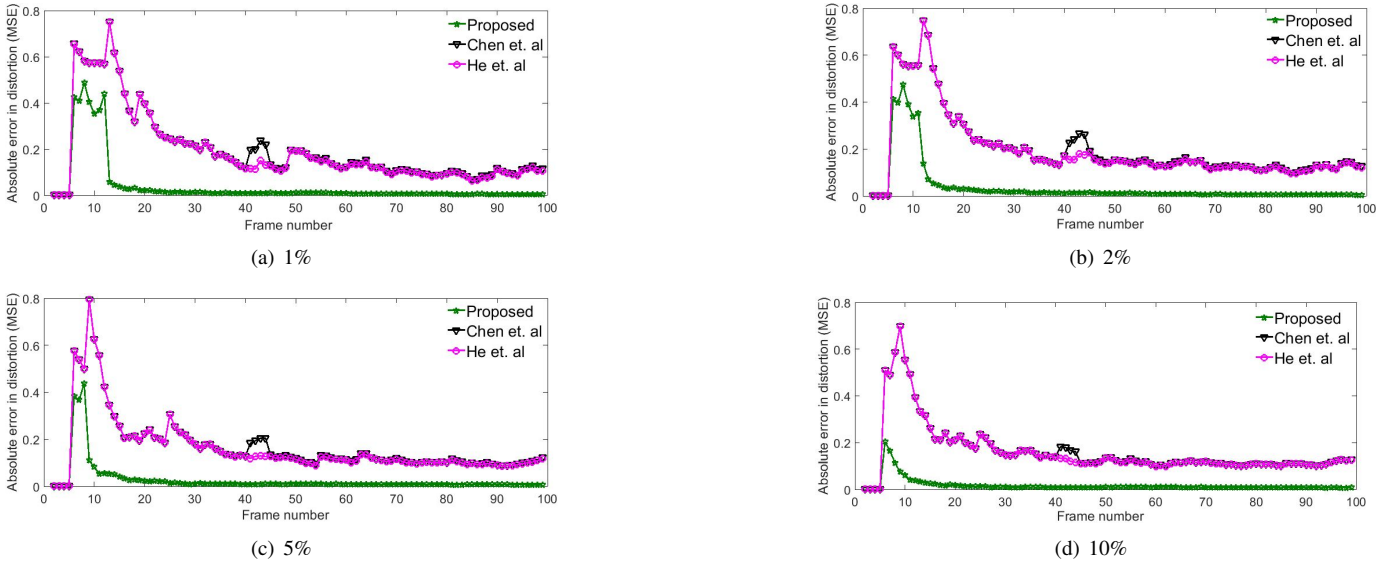
the brute force measurements making the proposed distortion model suitable for long duration videos.

Fig. 9 describes the behaviors of the proposed method and the two state-of-the-art methods for different packet error rates. *GT\_Fly* video sequence is tested for window-length 4 in all of these simulations. As can be seen in Fig. 9, the two state-of-the-art methods have demonstrated similar performances for all the four packet error rates tested. However, two important observations can be made in the proposed distortion prediction technique. First, as the frame number increases the percentage error between the predicted and the brute-force approach get reduced and eventually become stable. Secondly, as the packet error rate increases, the above stabilizing point occurs more rapidly. The first observation follows the criteria described in the previous paragraph (As more frames are encoded, the prediction model becomes more knowledgeable). The second observation can be explained by the fact that as the packet error rate increases, the convergence of the model parameters occur more rapidly and consequently the stabilizing point. This eventually results in overall lesser error percentage for high packet error rate scenarios.

Finally, Table V describes the measured block-wise distortion estimation performances for different content types. Firstly, similar behaviors as with Fig. 9 can be observed for all the video contents (i.e., the performance gain over the state-of-the-art methods, percentage error being reduced along with the increase in packet error rates). Secondly, the dependency of the content type on distortion estimation can be observed in Table V. As is expected, for relatively low motion video sequences such as *Beer\_garden*, *Cafe* the distortion estimation has become more accurate. This is simply, for low motion video sequences, motion vectors are relatively smaller and uniform and hence they can be recovered accurately. For high motion sequence, the discrepancies of the actual and recovered motion vectors during the error concealment operation become larger and consequently algorithms become less accurate.

### B. Effectiveness of Overall Error Resilience

This section describes the overall performance of the proposed method during the video transmission over error prone channels. Three state-of-the-art methods [3], [5] and [16] are incorporated during the performance comparison for 1%, 2%, 5% and 10% random packet errors. Fig. 10 illustrates the PSNR measurements at the decoder for different error resilience methods for *Musicians* video sequence. The horizontal axis in each graph is represented using bit-per-pixel (bpp). As it can be seen in Fig. 10, the proposed method has outperformed the state-of-the-art methods with a significant margin for all the packet error rates considered. The percentage gain of the proposed method over the state-of-the-art methods [3], [5] is more prominent in low bit rates for all the considered packet error rates. This behavior indicates that the redundant information inclusion which becomes much crucial at low bit rates, has been much optimum in the proposed method compared to [3], [5]. The performance gain of the proposed over [3], [5] is relatively consistent for all the tested packet error rates, making it equally suitable for channels ranging from highly lossy channels to not-so lossy channels.

Fig. 9. Prediction accuracy for different PERs for *GT\_Fly* sequenceTABLE V  
AVERAGE OF BLOCKWISE PERCENTAGE ERROR IN DISTORTION ESTIMATION (%)

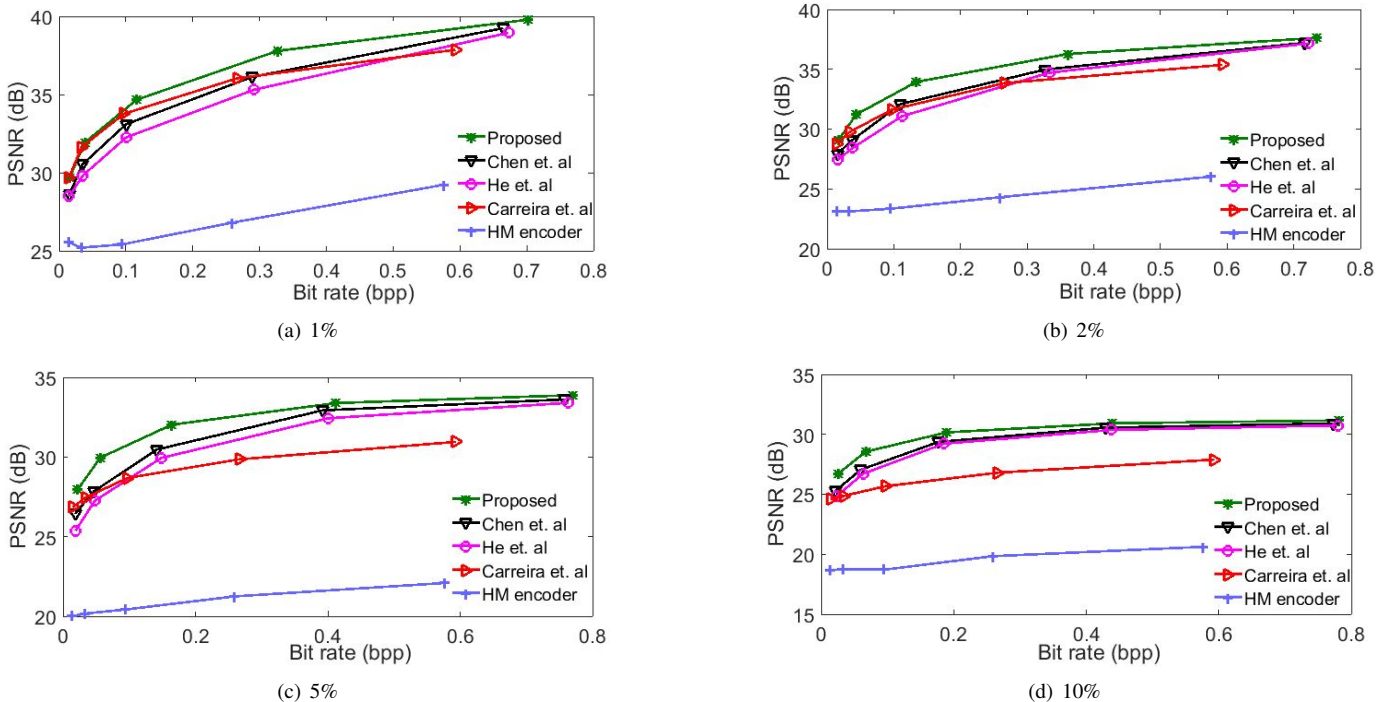
	Resl.	1088p		1080p										CIF							Average	
		Dancer	GT_Fly	Band	Beer_garden	Cafe	Park_scene	Musicians	Start_race	Water_splash	Basketball_drive	BQTerrace	Cactus	Akiyo	Bus	Carphone	Coastguard	Football	Foreman	Highway		Stefan
1%	Proposed	4.3	3.9	9.6	4.0	5.1	4.7	3.6	1.3	17.3	4.8	11.3	7.8	6.1	8.5	4.0	7.2	16.9	5.9	8.8	6.1	<b>7.06</b>
	Chen [5]	20.0	28.2	36.2	22.3	31.0	21.7	24.7	22.2	37.9	45.1	39.4	53.7	13.3	25.4	21.9	25.5	53.0	26.3	51.9	25.7	<b>31.27</b>
	Zhihai He [3]	26.2	19.4	23.7	16.7	18.3	19.5	20.4	4.1	32.7	42.8	39.1	53.5	13.0	21.4	11.2	17.5	32.9	20.4	16.4	18.6	<b>23.39</b>
2%	Proposed	3.6	3.8	8.4	4.2	4.5	4.4	3.4	0.8	11.2	4.2	10.2	7.4	6.3	7.9	3.0	4.4	15.8	5.8	8.0	6.4	<b>6.19</b>
	Chen [5]	26.3	20.3	35.7	22.9	32.3	19.9	25.2	30.2	56.6	50.9	36.8	54.3	12.3	25.2	18.6	64.1	53.1	22.4	50.6	23.2	<b>34.05</b>
	Zhihai He [3]	24.4	19.6	23.2	17.6	20.1	17.9	21.1	13.7	39.7	48.7	36.6	54.1	12.0	21.2	8.1	65.5	31.9	16.7	16.0	16.9	<b>26.25</b>
5%	Proposed	2.8	2.6	8.1	3.3	2.8	3.1	2.8	2.2	9.2	2.2	7.8	5.4	8.7	6.5	2.6	4.2	10.9	5.1	7.0	4.3	<b>5.08</b>
	Chen [5]	22.8	16.8	34.2	20.7	28.6	17.8	23.1	29.8	53.3	46.9	32.4	42.9	14.1	23.2	17.4	71.9	51.2	22.4	48.9	19.2	<b>31.88</b>
	Zhihai He [3]	21.0	16.3	22.8	15.4	16.7	16.4	19.6	15.0	34.3	43.8	32.2	42.7	13.8	20.6	7.9	77.8	28.5	17.3	17.7	13.6	<b>24.67</b>
10%	Proposed	2.2	1.7	6.5	1.8	2.4	1.9	2.2	0.8	7.3	1.4	6.2	3.4	1.2	3.8	1.1	2.2	8.0	1.5	3.4	3.1	<b>3.11</b>
	Chen [5]	21.9	16.4	31.5	20.9	31.0	17.4	20.6	27.3	51.4	35.0	31.6	40.3	5.4	17.2	14.0	38.5	45.2	16.9	43.9	16.0	<b>27.12</b>
	Zhihai He [3]	20.6	16.1	21.8	16.5	20.4	16.6	18.5	12.0	34.1	31.0	31.3	40.0	5.2	14.6	6.3	35.7	24.0	11.8	14.1	10.9	<b>20.08</b>

As it can be further seen in Fig. 10, [16] has performed reasonably well at relatively lower packet error rates while operating at a narrow range of bit rates towards the lower end. The lack of a probabilistic model, which could have evaluated the impact of certain coding decisions along the temporal domain, has been the major reason for poor performance in [16] for higher packet error rates. For lower packet error rates the impact is minimal simply because the temporal error propagation too is minimal. Furthermore, at lower bit rates the amount of redundant information that can be allocated for error resilient purposes are minimal irrespective of the type of the error resilient algorithm, which eventually leads to similar performances for all the methods.

As is further evident from Fig. 10, the HM encoder with motion-copying at the decoder has performed significantly poor compared to the rest of the methods. This is mainly due to the fact that the HM based encoding assumes an error free transmission and hence no error resilience is introduced. Finally, the performance gain in [5] compared to [3] is mainly

attributed to its consideration of error proration impact for both the coding mode selection process as well as the motion vector estimation process where as [3] only evaluates the error propagation impact during the coding mode selection.

The computational complexities of different methods are illustrated in Table VI and VII. All the simulations in the tables are conducted in a PC with a 2.8 GHz CPU and a 9 GB RAM. In Table VI, the percentage increase in encoding time compared to HM16.2 [21] is measured for a window length 4. Here, the proposed method, [3] and [5] have exhibited high complexities owing to their sophisticated probabilistic models whereas [16] without such a model, has consumed a lower time. Furthermore, the processing time required for the proposed method at different window lengths are illustrated in Table VII. As expected, for higher window lengths, the encoding time has significantly increased due to the growing number of branches in the probability tree (i.e., Fig. 3). However, it should also be noted that the distortion calculations in each branch of a given window length can be implemented as

Fig. 10. PSNR for *Musicians* sequence at different PERsTABLE VI  
ENCODING TIME COMPARED TO HM16.2 FOR WINDOW LENGTH 4

	Beer_garden	GT_Fly	BMX_2	Park_scene	Akiyo	Bus	Foreman	Coastguard	Average
Proposed	274%	324%	340%	338%	174%	143%	185%	180%	<b>245%</b>
Chen [5]	266%	396%	345%	377%	162%	178%	177%	157%	<b>257%</b>
Zhihai He [3]	271%	389%	335%	369%	159%	143%	173%	153%	<b>249%</b>
Carreira [16]	197%	194%	217%	213%	103%	104%	104%	123%	<b>157%</b>

TABLE VII  
AVERAGE ENCODING TIME FOR DIFFERENT WINDOW LENGTHS

	Win. 4	Win. 4 Parallel Proce.	Win. 8	Win. 8 Parallel Proce.	Win. 12	Win. 12 Parallel Proce.	Avg. for Parallel Proce.
Proposed	249%	<b>111%</b>	1,236%	<b>120%</b>	16,767%	<b>126%</b>	<b>119%</b>

independent threads if parallel processing support is available (e.g., in software encoding). Therefore, had the proposed method been implemented to support multiple threads, the encoding time could have been reduced approximately up to 119% as shown in the table.

Table VIII summarizes the BD-rate gain observed for the proposed method for 5 equally spaced QP values in the range 20-40. The HM encoder with motion-copying has not been incorporated during the performance evaluation due to the fact that it doesn't follow a hyperbolic curve during the packet losses and a meaningful number for the BD-rate therefore cannot be attained. Among the noticeable behaviors, the proposed method has demonstrated better performances for most of the videos tested and has failed in few instances. For *Akiyo*

and *Container*, [3] and [5] has outperformed the proposed method. They are very slow moving video sequences with little variation over time. Therefore, it can be concluded that for very slow motion sequences with stationary objects, even the models without AMVP produces good results. Furthermore, [16] has outperformed the proposed method for *GT\_Fly* video sequence under low packet error rates. *GT\_Fly* has a uniform movement and consequently motion vectors too become fairly uniform. For such sequences, motion vector protection alone as done in [16] is sufficient and therefore having an advanced error propagation model is not essential. However, as the PER increases the necessity for a probabilistic model emerges as the propagated error cannot be concealed only by preserving the motion vectors [16]. In general it can be concluded that for most of the video types the proposed method has outperformed the state-of-the-art methods with a significant margin.

## VII. CONCLUSIONS

This paper proposes an error resilient framework that can be incorporated during the HEVC video compression. The contribution of the paper is two fold. First a probabilistic model is presented to estimate the distortion experienced by the HEVC decoder, when the coded video is received over a lossy channel. During its design, the probabilistic model has considered the quality degradation in three inter-related aspects namely, errors caused by the incorrect motion vectors, errors in the pixels in the reference frames and the errors attributed to the clipping operation. Consequently the model has been capable of minimizing the discrepancy as low as 3.11% compared to the brute-force approach while state-of-the-art methods couldn't reach below 20.08%. The second contribution describes a novel Rate Distortion Optimization

TABLE VIII  
AVERAGE BD-RATE GAIN COMPARED TO STATE-OF-THE-ART ERROR RESILIENT METHODS (%)

		Dancer	GT_Fly	Beer_garden	Bmx_2	Cafe	Musicians	Park_scene	Basketball_drive	BQTerrace	Cactus	Akiyo	Bus	Coastguard	Stefan	Foreman	Football	Highway	Carphone	Container	Flower	Average		
	Resl.	1088p			1080p								CIF											
1%	Carreira [16]	-45.1	140	-2.6	-8.8	-44.0	-14.3	-17.3	-89.4	-16.5	-48.7	-3.3	-38.2	-47.5	-28.2	-46.5	-27.9	-85.7	-82.1	39.7	-92.3	<b>-27.93</b>		
	Chen [5]	-58.1	19.0	-28.7	-57.8	-46.8	-34.0	-27.9	-34.9	-58.5	-26.2	15.8	-12.9	-8.0	-22.9	-31.7	-23.0	-27.7	-12.3	50.8	-2.5	<b>-21.41</b>		
	Zhihai He [3]	-67.9	-26.1	-20.7	-59.1	-66.3	-49.6	-39.0	-44.9	-51.2	-45.6	30.0	-34.4	-30.5	-35.2	-29.5	-9.5	-19.8	-1.2	54.6	-71.6	<b>-30.88</b>		
2%	Carreira [16]	-53.1	176	-2.5	-77.0	-41.8	-44.5	-31.3	-93.6	-13.6	-43.0	-17.1	-51.5	-50.0	-26.6	-80.1	-34.7	-90.0	-99.8	26.5	-99.0	<b>-37.34</b>		
	Chen [5]	-69.4	14.2	-23.3	-62.9	-61.5	-41.6	-36.0	-44.3	-55.1	-53.1	5.9	-29.7	-43.5	-19.7	-33.4	-13.8	-5.0	-25.2	50.2	-87.6	<b>-31.74</b>		
	Zhihai He [3]	-70.6	-41.3	-37.5	-68.0	-84.4	-53.7	-42.7	-44.9	-55.2	-50.5	19.0	-39.2	-28.3	-41.6	-23.3	-23.4	12.8	14.1	48.1	-70.7	<b>-34.06</b>		
5%	Carreira [16]	-71.3	-96.5	-45.9	-97.0	126	-75.0	-61.7	-95.9	-6.3	-39.5	-24.2	-81.0	-96.6	-90.0	-96.4	-72.3	-94.4	-100	-64.5	-100	<b>-64.12</b>		
	Chen [5]	-80.5	-76.1	-38.9	-51.8	-22.8	-44.7	-41.3	-40.5	-60.0	-51.9	9.4	-39.4	-62.1	-34.6	-43.1	-26.1	9.0	-34.8	-63.2	-93.9	<b>-44.37</b>		
	Zhihai He [3]	-65.2	-66.0	-61.6	-57.8	-56.0	-57.8	-36.1	-46.9	-48.0	-41.5	37.9	-37.6	-66.0	-32.0	-33.9	-19.1	23.0	-28.9	-89.1	-89.3	<b>-43.59</b>		
10%	Carreira [16]	-100	-90.0	-33.3	-96.9	4.2	-89.2	-80.2	-98.4	12.6	-25.7	-19.9	-95.0	-100	-80.0	-99.4	-89.7	-99.9	-100	-20.8	-100	<b>-70.08</b>		
	Chen [5]	-87.6	-23.5	-37.7	-59.0	-53.5	-40.8	-28.7	-9.6	-64.3	-65.2	29.1	-37.0	-60.0	-34.3	-40.0	-34.0	-3.7	-45.0	-54.4	-93.4	<b>-42.13</b>		
	Zhihai He [3]	-53.7	-65.4	-44.9	-54.0	-58.7	-49.7	-21.4	-53.8	-54.7	-43.2	78.5	-33.0	-62.0	-32.4	-34.2	-22.1	25.9	-3.1	-77.9	-83.0	<b>-37.14</b>		

(RDO) algorithm which is employable at the HEVC encoder during the coding parameters selection process in an erroneous environment. The proposed RDO algorithm which is based on the probabilistic model derived in the first contribution, incorporates packet error rate and the content dependant parameters to derive an accurate QP- $\lambda$  relationship. Consequently, the proposed method has demonstrated 21.41%-43.59% BD-rate improvement compared to the existing methods. Future works focuses on reducing the complexity of the distortion estimation algorithm through an effective tree-pruning algorithm.

#### APPENDIX A

##### THE IMPACT OF CLIPPING ON THE PROPAGATED ERRORS

As described in (8), the impact of pixel clipping at the decoder on the squared error is quantified by two terms;  $DP_{Refer}^{SE,\Gamma}$  and  $DX_{Refer}^{D,\Gamma}$ . Re-written as a single summation,

$$DP_{Refer}^{SE,\Gamma} + DX_{Refer}^{D,\Gamma} \triangleq \sum_{\forall u \in \mathcal{U}_1} P(u) \left\{ E \left[ \left( \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right)^2 \right] + 2 \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) E \left[ \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right] \right\}.$$

In order to simplify the notation in the following derivations, let  $\mathcal{A} \triangleq \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\}$ . The above now simplifies to the following:

$$DP_{Refer}^{SE,\Gamma} + DX_{Refer}^{D,\Gamma} = \sum_{\forall u \in \mathcal{U}_1} \left\{ E \left[ \left( \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} + \mathcal{A} \right)^2 \right] + 2 \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) E \left[ \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} + \mathcal{A} \right] \right\} P(u) \quad (29)$$

$$DP_{Refer}^{SE,\Gamma} + DX_{Refer}^{D,\Gamma} = \sum_{\forall u \in \mathcal{U}_1} P(u) \left\{ E \left[ \left( \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} \right)^2 \right] + 2 \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) E \left[ \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} \right] + E \left[ \mathcal{A}^2 \right] + 2 E \left[ \mathcal{A} \left( \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} \right) \right] + 2 \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) E \left[ \mathcal{A} \right] \right\} \quad (30)$$

Note that  $E \left[ \left( \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} \right)^2 \right] \triangleq D_{CH}^{SE} \left( X_{n_0}^{k(u)} \right)$  from (3) and  $E \left[ \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} \right] \triangleq D_{CH}^D \left( X_{n_0}^{k(u)} \right)$  from (4). Hence, the first two terms of (30) can be redefined in terms of these known quantities as,

$$DX_{Refer}^D \left( X_n^i \right) \triangleq \sum_{\forall u \in \mathcal{U}_1} 2 P(u) \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) D_{CH}^D \left( X_{n_0}^{k(u)} \right) \\ DP_{Refer}^{SE} \left( X_n^i \right) \triangleq \sum_{\forall u \in \mathcal{U}_1} P(u) D_{CH}^{SE} \left( X_{n_0}^{k(u)} \right). \quad (31)$$

Thus, (30) becomes

$$DP_{Refer}^{SE,\Gamma} + DX_{Refer}^{D,\Gamma} = DP_{Refer}^{SE} + DX_{Refer}^D + \sum_{\forall u \in \mathcal{U}_1} P(u) \left\{ E \left[ \mathcal{A} \left\{ \mathcal{A} + 2 \left( \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} \right) + 2 \left( \hat{X}_{n_0}^j - \hat{X}_{n_0}^{k(u)} \right) \right\} \right] \right\}. \quad (32)$$

Expanding  $\mathcal{A}$  further, (32) further simplifies to

$$DP_{Refer}^{SE,\Gamma} + DX_{Refer}^{D,\Gamma} = DP_{Refer}^{SE} + DX_{Refer}^D + \sum_{\forall u \in \mathcal{U}_1} P(u) \left\{ E \left[ \mathcal{A} \left( r_n^i + 2 \hat{X}_{n_0}^j - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} - \tilde{X}_{n_0}^{k(u)} \right) \right] \right\}. \quad (33)$$

In a similar fashion, the clipped term  $DP_{Refer}^{D,\Gamma}$  of the difference error in (12) can also be simplified further. Therefore,

$$DP_{Refer}^{D,\Gamma} \triangleq \sum_{\forall u \in \mathcal{U}_1} P(u) E \left[ \hat{X}_{n_0}^{k(u)} + r_n^i - \Gamma \left\{ \tilde{X}_{n_0}^{k(u)} + r_n^i \right\} \right] \quad (34) \\ = \sum_{\forall u \in \mathcal{U}_1} P(u) E \left[ \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} + \mathcal{A} \right]. \quad (35)$$

Since  $E \left[ \hat{X}_{n_0}^{k(u)} - \tilde{X}_{n_0}^{k(u)} \right] \triangleq D_{CH}^D \left( X_{n_0}^{k(u)} \right)$ , if

$$DP_{Refer}^D \triangleq \sum_{\forall u \in \mathcal{U}_1} P(u) D_{CH}^D \left( X_{n_0}^{k(u)} \right), \quad (36)$$

$$DP_{\text{Refer}}^{\text{D},\Gamma} = DP_{\text{Refer}}^{\text{D}} + \sum_{\forall u \in \mathcal{U}_1} P(u) E[\mathcal{A}]. \quad (37)$$

## REFERENCES

- [1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 12 2012.
- [2] B. Oztas and M. Pourazad, "A study on the HEVC performance over lossy networks," *IEEE International Conference on Electronics, Circuits and Systems*, pp. 785–788, 2012.
- [3] Zhihai He, Jianfei Cai, and Chang Wen Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 511–523, 6 2002.
- [4] D. Zhang, J. Liang, and I. Singh, "End-to-end distortion estimation for H.264 with unconstrained intra prediction," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 9 2012, pp. 697–700.
- [5] Z. Chen, P. V. Pahalawatta, A. M. Tourapis, and D. Wu, "Improved Estimation of Transmission Distortion for Error-Resilient Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 636–647, 4 2012.
- [6] L. Anegekuh, L. Sun, E. Jammeh, I.-H. Mkwawa, and E. Ifeachor, "Content-Based Video Quality Prediction for HEVC Encoded Videos Streamed Over Packet Networks," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1323–1334, 8 2015.
- [7] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos, "Error resilient video coding techniques," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, 7 2000.
- [8] Hua Yang and K. Rose, "Advances in Recursive Per-Pixel End-to-End Distortion Estimation for Robust Video Coding in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 845–856, 7 2007.
- [9] A. R. dela Cruz and R. D. Cajote, "Error resilient joint source-channel adaptive intra-refresh rate for wireless video transmission," in *2014 19th International Conference on Digital Signal Processing*. IEEE, 8 2014, pp. 509–514.
- [10] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 952–965, 6 2000.
- [11] Y. Zhang, W. Gao, Y. Lu, Q. Huang, and D. Zhao, "Joint Source-Channel Rate-Distortion Optimization for H.264 Video Coding Over Error-Prone Networks," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 445–454, 4 2007.
- [12] C.-H. Lu and Y. Lin, "Robust error resilient H.264/AVC video coding," in *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*. IEEE, 10 2013, pp. 1–4.
- [13] B. Li, J. Xu, and H. Li, "Parsing robustness in High Efficiency Video Coding - analysis and improvement," in *2011 Visual Communications and Image Processing (VCIP)*. IEEE, 11 2011, pp. 1–4.
- [14] J. Carreira, V. De Silva, E. Ekmekcioglu, A. Kondoz, P. Assuncao, and S. Faria, "Dynamic motion vector refreshing for enhanced error resilience in HEVC," pp. 281–285, 2014.
- [15] J. Carreira and V. Silva, "Dynamic motion vector refreshing for enhanced error resilience in HEVC," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European*, 2014, pp. 281–285.
- [16] J. Carreira, P. A. Assuncao, S. M. M. de Faria, E. Ekmekcioglu, and A. Kondoz, "A Two-stage Approach for Robust HEVC Coding and Streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2017.
- [17] T. K. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J.-R. Ohm, and G. J. Sullivan, "Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 76–90, 1 2016.
- [18] Y.-L. Chang, Y. A. Reznik, Z. Chen, and P. C. Cosman, "Motion Compensated Error Concealment for HEVC Based on Block-Merging and Residual Energy," in *2013 20th International Packet Video Workshop*. IEEE, 12 2013, pp. 1–6.
- [19] W. Lam, A. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, 1993, pp. 417–420.
- [20] H. Schulzrinne, "Real Time Streaming Protocol (RTSP)." [Online]. Available: <https://tools.ietf.org/html/rfc2326>
- [21] "HM software manual." [Online]. Available: [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/)
- [22] B. Li, H. Li, L. Li, and J. Zhang, " $\lambda$  Domain Rate Control Algorithm for High Efficiency Video Coding," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3841–3854, 9 2014.
- [23] Bin Li, Jizheng Xu, Dong Zhang, and Houqiang Li, "QP refinement according to Lagrange multiplier for High Efficiency Video Coding," in *2013 IEEE International Symposium on Circuits and Systems (IS-CAS2013)*. IEEE, 5 2013, pp. 477–480.



**Gosala Kulupana** (M'14-S15) received his B.Sc. (Eng.) degree with honours from University of Moratuwa, Sri Lanka in 2011. From 2011 to 2014, he was an Engineer at Mobitel (pvt) Ltd., Sri Lanka. He completed his PhD in the video coding domain in 2017 and currently working as 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 resource optimization, distributed processing and error resilience video coding.



**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, Canberra, 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 sound source localization, spatial sound field reproduction and convex optimization techniques.



**Hemantha Kodikara Arachchi** (M'02) received his B.Sc. (Eng.) degree with honours 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 CVSSP Multimedia Communications Research group of the University of Surrey, UK. 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 at the Department of Electrical and Electronic Engineering, University of Bristol, UK in February 2001. Currently, he is a reader in signal processing at University of Surrey, UK. His current research interests include cloud communication, Quality of Experience, intelligent video encoding for wireless systems and video communication in LTE.