



# Low delay error resilience algorithm for H.265|HEVC video transmission

Taha T. Alfaqheri<sup>1</sup> · Abdul Hamid Sadka<sup>1</sup>

Received: 28 April 2019 / Accepted: 21 October 2019  
© The Author(s) 2019

## Abstract

Transmission of high-resolution compressed video on unreliable transmission channels with time-varying characteristics such as wireless channels can adversely affect the decoded visual quality at the decoder side. This task becomes more challenging when the video codec computational complexity is an essential factor for low delay video transmission. High-efficiency video coding (H.265|HEVC) standard is the most recent video coding standard produced by ITU-T and ISO/IEC organisations. In this paper, a robust error resilience algorithm is proposed to reduce the impact of erroneous H.265|HEVC bitstream on the perceptual video quality at the decoder side. The proposed work takes into consideration the compatibility of the algorithm implementations with and without feedback channel update. The proposed work identifies and locates the frame's most sensitive areas to errors and encodes them in intra mode. The intra-refresh map is generated at the encoder by utilising a grey projection method. The conducted experimental work includes testing the codec performance with the proposed work in error-free and error-prone conditions. The simulation results demonstrate that the proposed algorithm works effectively at high packet loss rates. These results come at the cost of a slight increase in the encoding bit rate overhead and computational processing time compared with the default HEVC HM16 reference software.

**Keywords** H.265|HEVC · Video error resilience · Low delay · Error-prone condition · Video transmission

## 1 Introduction

The advancement in the manufacturing of high-performance electronic devices and their display technologies, such as smart mobile phones, tablets, and televisions devices, resulted in increased demands of ultrahigh-resolution video content delivery with low processing delay. Furthermore, nowadays, most of commercially available displays support spatial resolution up to 4 K (7668 × 4320) display resolution [1]. Such high-resolution display capabilities can consume most of the available bandwidth in conventional networks. Hence, for delivering high-quality video effectively, it is necessary to use efficient video coding tools to support high-resolution video applications. The most recent video coding

standard is the high-efficiency video coding (H.265|HEVC) standard [2].

The main target of developing H.265|HEVC standard was to double the coding efficiency of MPEG-4 part 10, advanced video coding (H.264|AVC) standard. This means keeping the same video quality at half encoding bit rate [3]. Furthermore, we extend the H.264|AVC applications, which are already supported by H.264|AVC coding standard to include more efficient video coding tools for high-resolution video and parallel processing applications [4].

However, the increase in video coding efficiency comes at the expense of an increase in the amount of inter-prediction and motion compensation activity in the encoding process, which mainly involves increasing the coding efficiency by increasing the number of temporal and spatial redundant information [5]. Important to realise on the negative side, this high coding efficiency comes at the cost of high computation complexity requirements as well.

In other words, a highly compressed bitstream means more encoded redundant video information. Consequently, compressed video content becomes more sensitive to channel bit errors than encoded bitstream with previous standards.

✉ Abdul Hamid Sadka  
Abdul.Sadka@brunel.ac.uk

Taha T. Alfaqheri  
Taha.alfaqheri@brunel.ac.uk

<sup>1</sup> Electronic and Computer Engineering, Brunel University  
London, London, UK

Therefore, transmitting a highly compressed video bitstream in unreliable transmission channel leads to degrading the perceived visual quality at the video decoder or if errors hit the sensitive encoded data such as slice header data leads to failing in the decoding process for the whole video sequence [6]. Figure 1 demonstrates a video transmission issue on the received video quality when using an unreliable wireless channel.

The H.265/HEVC codec system is a hybrid video coding type, which means the compression techniques depends on removing the temporal information first and then spatial information.

Thus, the injected errors on the transmitted video bitstream are propagated spatially and temporally on the perceived video quality. Consequently, injecting single-bit error on the encoded bitstream can lead to severe visual quality degradation.

In the development of H.265/HEVC, the primary target of both standardisation organisations (i.e. ITU-T and ISO/IEC) was to increase the bit rate saving to more than 50% compared to the previous H.264/AVC video coding standard [7, 8]. The high bit rate saving was achieved by adding new coding features to support more efficient coding and make the video codec more friendly to parallel processing applications [7]. However, developing error resilience tools for video coding standard is out of the video standard scope [6].

There are two main error control categories to reduce the effects of transmission errors on perceived visual quality. The first one employs traditional data error control methods which use lossless channel coding tools in data recovery such as Automatic Retransmission reQuest (ARQ) schemes. However, implementing such error recovery tools in compressed video delivery is less efficient because the nature of the compressed bitstream is of a variable-length code, which makes error recovery process of corrupted video contents a very challenging task.

The second category is video error control techniques implemented within the video coding system. In this case, to minimise the effects of the transmission errors efficiently at the video decoder side, the video error control can be divided into three categories: forward error recovery, error concealment, and interactive error recovery.

In the forward error recovery approach, the video encoder takes the full responsibility to insert redundant error resilience codes and makes the coded bitstream more robust against errors.

The second error control approach is called the error concealment approach, in which the decoder is responsible to spatially and temporally conceal the errors. The spatial error concealment employs correctly received information using interpolation techniques on the surrounded macroblocks. In case the whole macroblock information is lost, the simplest and most common concealment technique consists of replacing the lost macroblock by the spatially corresponding macroblock in the previously decoded frame. The other error concealment approach at the decoder is called temporal error concealment techniques, which extrapolate the correctly received motion vectors of the current and previous frames [9].

The third video error control approach uses the joint encoder–decoder error resilience techniques. In this approach, the video encoder and decoder work in an interactive way to reduce the effects of the channel errors on the perceived visual quality. In this category, a backward feedback channel is used from the decoder to the encoder sides to keep the encoder updated.

All lossy error resilience techniques take advantage of the nature of the human visual system to tolerate the distorted visual quality. The design of video error resilience tools depends on the employed video coding tools.

The developed error resilience tools should keep a balance between error robustness and video encoding efficiency,

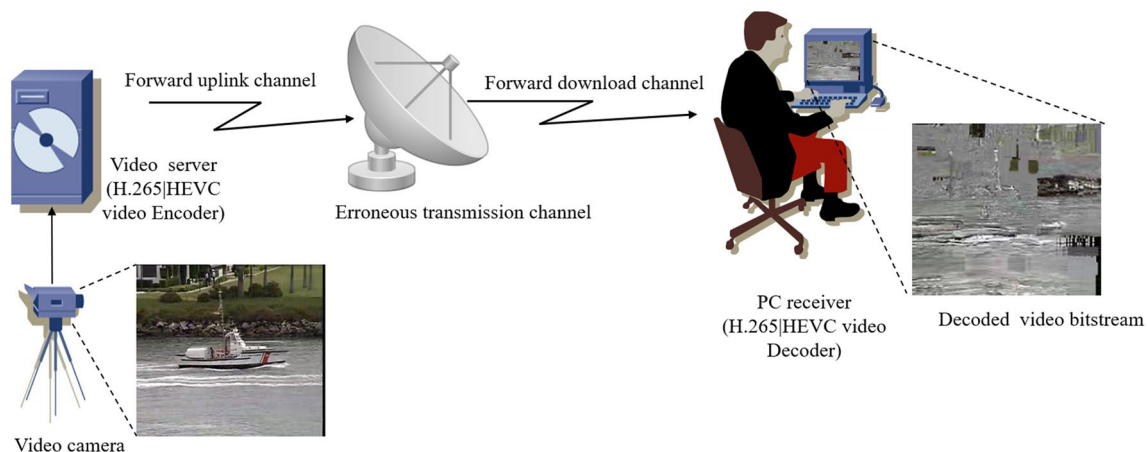


Fig. 1 Video transmission issue scenario

whilst aiming to maintain the video quality of service under error-prone conditions.

Developing error resilience tools at the encoder is one of the most efficient solutions to mitigate the effects of erroneous errors in real-time video transmission.

This paper presents an adaptive error resilience algorithm to increase perceived visual quality using H.265/HEVC coding system under error-prone conditions.

The proposed work is based on forward error recovery method (without using a feedback channel) and interactive error recovery (using feedback update from the decoder to the encoder sides).

The intra-encoding mode is used at slice level instead of intra-picture to support low delay delivery applications and keep a balance between coding efficiency and error resilience performance. Our proposed work considers during the design stage the bit rate overhead, time delay of computational complexity, and video start-up time delay. The evaluation work is further extended to investigate the effect of LTE network traffic load on perceived visual quality at the decoder.

This paper is organised as follows. An overview of relevant literature that supports the proposed error resilience algorithm is presented in Sect. 2. Section 3 describes the proposed adaptive slice algorithm. The evaluation process and encoding configuration for testing the proposed algorithm are reported in Sect. 4. Section 5 discusses the obtained objective and frame by frame quality assessment results, coupled with computational complexity, and processing time delay analysis. Finally, Sect. 6 presents the paper's conclusions and future work recommendations.

## 2 Background

### 2.1 Error resilience using region of interest extraction

This section presents a literature review of related work of the state-of-the-art video error resilience algorithms for low delay video delivery applications.

One of the main video encoding requirements at low delay or conversational video applications is to reduce the number of reference frames to a minimum level. These low delay video processing requirements can be achieved by reducing the number of previously used future reference frames at the motion process.

The first conducted work to select a group of MBs to be encoded with intra-refresh in H.264/AVC video coding was by Hoaming Chen et al. [10]. Their proposed error resilience coding scheme is based on adaptive intra-refresh. The error resilience coding scheme selects the important regions depending on the received network conditions, i.e. different

packet loss rates (PLRs) and video motion information. To keep coding efficiency at acceptable level, the refresh cycle sizes range at (4, 8, 16). A selected area depends on the PLR value in the feedback channel. When receiving update of low PLR values, i.e. ( $10^{-4}$ ), the cycle size will be selected with smaller values, i.e. (4). In contrast to high error-prone conditions such as PLR with ( $10^{-1}$ ), the refreshing cycle sizes will be increased to obtain a balance between error resilience and H.264/AVC coding efficiency performance.

Two research studies have been conducted in the recent past to improve error resilience of H.265/HEVC coding standard which considers the moving areas as a region of interest. In 2015, the authors considered the moving area of the encoded video as a region of interest (ROI) and these extracted areas should be protected against transmission errors. Their proposed error resilience algorithm is based on generating activity map at frame level. The moving regions are segmented into blocks and, based on the maximum depth level of CTUs, they calculate and protect the moving region parts at the encoder [5]. In 2016, they improved their region of interest algorithm by utilising a rate control in a region of interest extraction process [6]. The obtained objective quality results presented by the authors show that a significant improvement of 0.88 dB was achieved compared with H.265/HEVC reference selection method for a PLR of 5% [6]. The moving region extraction methodologies of these studies are based on a work proposed by Hai-Miao Hu et al. [5]. This work (ROI-based rate control scheme) aimed to improve the coding efficiency of H.264/AVC coding standard by allocating more encoding bit budget to moving regions and improve the perceived quality for ROI area at the cost of non-ROI quality.

However, using the moving region as a region of interest does not provide precise selectivity of the selected moving areas because of the flexibility of block partitioning in the H.265/HEVC encoding process.

Under these circumstances, an adaptive slice encoding (ASE) algorithm is developed and proposed based on an understanding of the previous related work.

One of the most challenging tasks in using region of interest extraction process in H.265/HEVC coding system is how to keep computation complexity to a minimum level. Another challenge is how to implement accurate ROI in low delay transmission constraint such as conversational video applications.

Rate control tool in video coding is responsible for calculating the best trade-off between image quality and the required bit rate. Most video coding systems, in general, are lossy systems, so it is important to keep a bit rate saving at the highest level with maintaining the highest visual quality level-targeted video applications.

There is a great deal of research interest in proposing efficient algorithms for moving region extraction [11]. The

moving regions extraction process should take into consideration a computation analysis. Such computation analysis is based on the human visual system analysis. For example, it includes frame texture details, skin colour, and object motion speed. These requirements make rate control adaptation more challenging in real-time processing applications.

The moving extraction process starts after the motion estimation stage, in which quantisation parameters are adjusted accordingly. There is a dilemma between region segmentation and motion estimation priorities. Quantisation parameters (QP) need to be adjusted before the rate-distortion optimisation (RDO) process start. On the other hand, motion information is generated after RDO and QP are generated before the RDO process. However, in the moving region extraction process, the QP needs to be adjusted based on motion information as an ROI. To solve the moving region extraction and QP adjustments dilemma, researchers in [12] proposed the motion differencing method. In this method, the motion vector of each current macroblock is compared with the macroblock of the previous frame. However, this method does not give acceptable results when dealing with fast-moving objects. In [13], the researchers proposed a method to distinguish the importance of each macroblock using the mean absolute difference (MAD) method between the current and previous macroblock (MB) [14]. Also, the researchers in [11] and [15] use the same region segmentation methodology which gives very accurate results when the background of the moving objects is stable. However, a slight movement in the object background area (e.g. temporal changing in lighting conditions or camera zooming) can adversely affect extracting the region of interest areas [16].

## 2.2 H.265|HEVC encoding with feedback channel

In general, low delay video application such as conference video application requires special attention in reference picture management. In unreliable networks, the encoder with feedback capabilities usually receives acknowledgement signal from the decoder with a delay channel (in milliseconds). Basically, in video codec systems, there are two types of acknowledgement signals; at slice level, these signals are transmitted from the decoder to the encoder [17]. The first type is called a positive acknowledgement (ACK) signal which is responsible for sending acknowledgement signal to the encoder, indicating the correctly received slice. The reference frame is chosen depending on the ACK signal update received from the decoder. If the encoder did not receive an ACK signal at a predefined interval, the encoder assumes that an error occurred and an intra-coding has to be applied.

The second feedback signal type sends a negative acknowledgement (NACK) by the decoder to notify the encoder that an error or loss has occurred in the currently decoded bitstream. In addition to the acknowledgement of

the correctly received slice, the addresses of the corrupted parts are signalled back to the encoder. On the other hand, the reference picture buffer is updated each time accordingly on receiving the acknowledgement signal. At the same time, the decoder can apply an error concealment technique to reduce temporal–spatial error propagation in case error occurred.

## 2.3 H.265|HEVC video coding standard

The H.265|HEVC video coding standard is the result of continuous hard work aimed at enhancing the coding efficiency of the previous video codec standards [18].

Several changes have been added to the H.265|HEVC coding tools such as video coding control signals and bitstream structure [4]. A new coding unit design concept of flexible coding unit sizes has been added to the standard. A basic unit in H.265|HEVC standard is called a coding tree unit (CTU). The encoding size of CTU ranges ( $8 \times 8 - 64 \times 64$ ) block sizes, which help to increase the coding efficiency with video resolution higher than high definition (HD) resolution [7]. All of the mentioned previous points make H.265|HEVC coding standard an attractive candidate to meet the main targets of wireless and multimedia applications for high visual quality requirements.

### 2.3.1 Slice structure in H.265|HEVC system

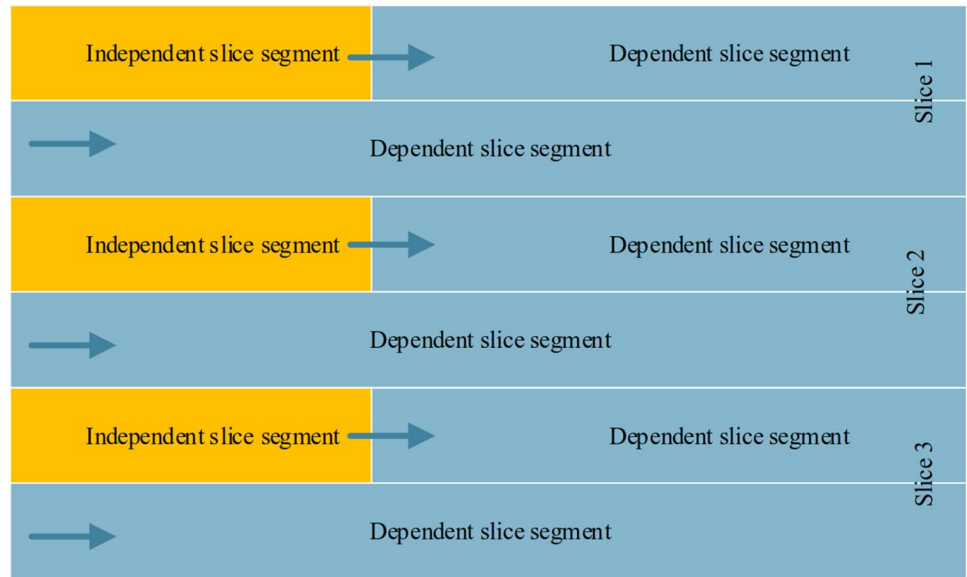
As discussed earlier, each frame can be represented by one or more slices and each slice contains a group dependent and independent slice segments. As can be seen in Fig. 2, a first slice segment is encoded independently from previously encoded slices. The remaining slice segments are encoded depending on the first slice segment with the same encoding mode of coding units.

In lossy packet networks, the packetized slices should not exceed the maximum transmission unit (MTU) size. On the other hand, the slicing structure helps decrease the transmitted packet length, and this is achieved by decreasing the number of encoded CTUs in each slice; hence, the length of transmitted packets becomes shorter, as a result, reducing error propagation at the decoder [19].

One main usage of the slicing segmentation concept in H.265|HEVC coding system is that it can reduce the effect of errors in corrupted video samples. Only the affected slice will be discarded or recovered from another correctly received slice segments. Moreover, the slicing segment concept helps to expedite the resynchronisation process using the correctly received independent slice segment header.

However, increasing the number of encoded slices adversely affects the coding efficiency. For example, at slice boundaries, the intra- and motion predictions are not

**Fig. 2** Slice segment process sequence



allowed, which reduces the spatial frame prediction. Furthermore, it is a slice structure increased overhead.

There are three slice types used in H.265/HEVC coding system: intra mode slice, uni-prediction slice (P-slice), and bi-prediction slice (B-slice). Each slice header contains a complete reference picture list update. A reference picture set concept for H.265/HEVC coding standard will be further explained in detail later.

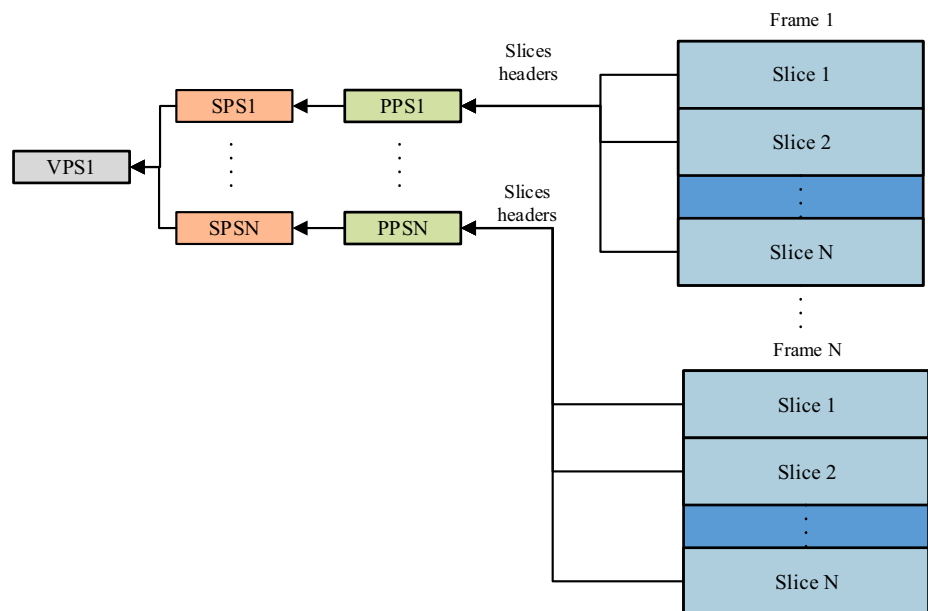
A slice header contains shared information between slice segments. This shared information differs from slice to slice at frame level. A reference selection list is updated in each slice header and signals explicitly. The slice header information of each frame is stored in picture parameter set (PPS). The PPS data is stored in a sequence parameter set

(SPS). Finally, a video parameter set (VPS) contains shared information of the PPS and SPS. Further demonstration of the interconnection of the three parameters set is shown in Fig. 3.

### 3 Proposed work

The aim of the proposed algorithm is to reduce error propagation at slice level. An adaptive encoding algorithm is introduced at the video encoder to encode and protect the most active slices. A coded video sequence is represented as a series of access units (AUs) in sequential order with shared sequence parameter. Each access unit is represented

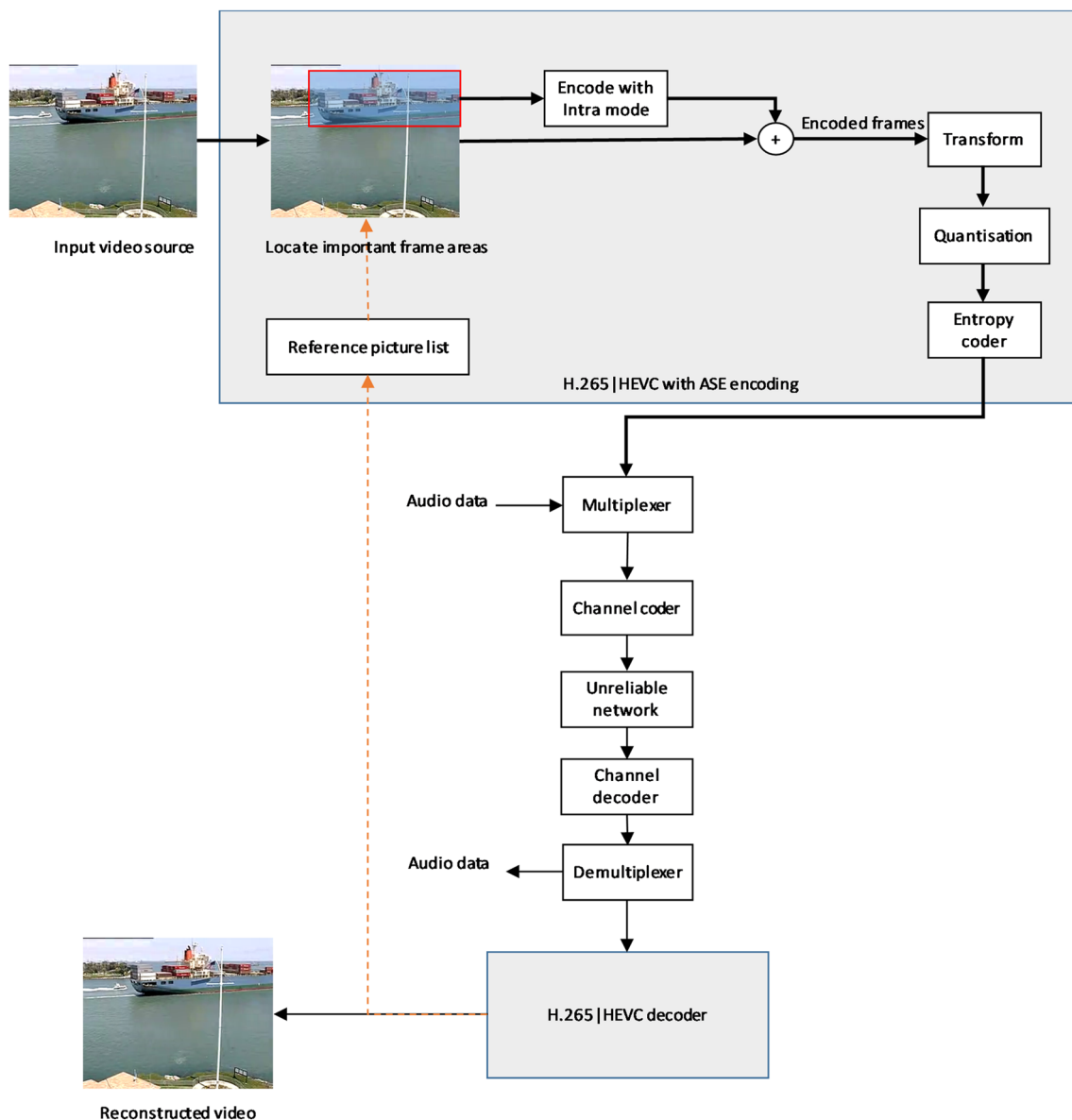
**Fig. 3** Activation of parameter sets in H.265/HEVC slice headers



by a group of NAL units. A prefix code of access unit delimiter is used to identify the start of new AU in NAL unit bitstream. A primary encoded AU contains a group of VCL NAL units, which includes one or multiple slices. These slices contain video sample data. A redundant coded picture is encoded as additional VCL NAL units. These additional VCL samples are used for error recovery when the primary video samples are lost or corrupted. In this case, the decoder will parse the contents of the correctly received data to recover the corrupted video samples. In error-free conditions, the decoder will discard the additional redundant video data. At the end of each video coded sequence, a non-VAL NAL unit is encoded to indicate the end of the NAL units bitstream.

The concept of the proposed algorithm is illustrated in (Fig. 4). For instance, suppose the ship in the video sequence is the most important area that requires protection against errors. This algorithm will extract the active slice, i.e. the ship, and the active slices are then encoded in intra mode.

As discussed earlier in the slice structure section, the independent slice header segment address identifies the exact location of a slice segment at picture level by using count number (ctb) identification in fixed scan order. The objective of the ASE algorithm is to reduce the temporal error propagation by encoding the most sensitive and important coding units (CUs) using intra-coding mode at slice level. In the following subsections, further details are provided as to how the activation map is generated to indicate the active slices that should then



**Fig. 4** Encoding the slice of the most important areas in the ASE algorithm

be encoded with intra mode. A rate control mechanism for the subdivided frame regions is presented as well.

### 3.1 Important area protection

The proposed algorithm is described as follows, At the first stage, a slice level differencing method is implemented on the current and previous frames. The active area consists of a change in content with new texture information. A moving slice is a highly important slice that must be protected to reduce error propagation. Each slice is mapped with its greyscale representation. The current and previous encoded slices are mapped into a row and column projection curves with  $(CV_n^x)$  and  $(CV_n^y)$ , respectively. Each slice in the current frame is projected into a one-dimensional vector.

The greyscale representation for the slice row  $L_n(x)$  and slice column  $L_n(y)$  can be calculated in Eqs. (1) and (2), respectively:

$$L_n(x) = \sum_x L(x, y), \tag{1}$$

$$L_n(y) = \sum_y L(x, y), \tag{2}$$

where  $L_n$  are the greyscale values for frame number (P). The average values for  $L_n(x)$  and  $L_n(y)$  are calculated based on the number of the calculated grey samples rows (r) and columns (c), respectively, as defined in Eqs. (3) and (4):

$$L_{avn}(x) = \frac{\sum_x L_n(x)}{r}, \tag{3}$$

$$L_{avn}(y) = \frac{\sum_y L_n(y)}{c}. \tag{4}$$

Then, the averaged 1-D projected curves are normalised using Eqs. (5) and (6):

$$CV_n^x = L_n(x) - L_{avn}(x), \tag{5}$$

$$CV_n^y = L_n(y) - L_{avn}(y), \tag{6}$$

where  $(CV_n^x)$  and  $(CV_n^y)$  represent the 1-D projected curves for the slice number (n).

For better coding efficiency performance, the extraction process uses intra-refresh map which is a calculated grey-scale projection method (GPM) [20]. For this reason, the extracted moving objects with unstable object background give more accurate selectivity results. The GPM extraction method is used in image stabilities applications because of its implementation simplicity and moving object selection accuracy [20].

A cross-correlation for the current and previous slice is then calculated [20]. The difference vector  $DV_n(p)$  or each slice is then calculated as in Eq. (7):

$$DV_n(p) = \frac{1}{256} \sum_{(i,j) \in p}^{TS} |L_n(i, j) - L_{n-1}(i + CV_n^x, j + CV_n^y)|, \tag{7}$$

where  $L_n(i, j)$  and  $L_{n-1}(i, j)$  are the luma samples representation for the current (n) and previous slice (n - 1) and (TS) is the total number of encoded slices per current frame.

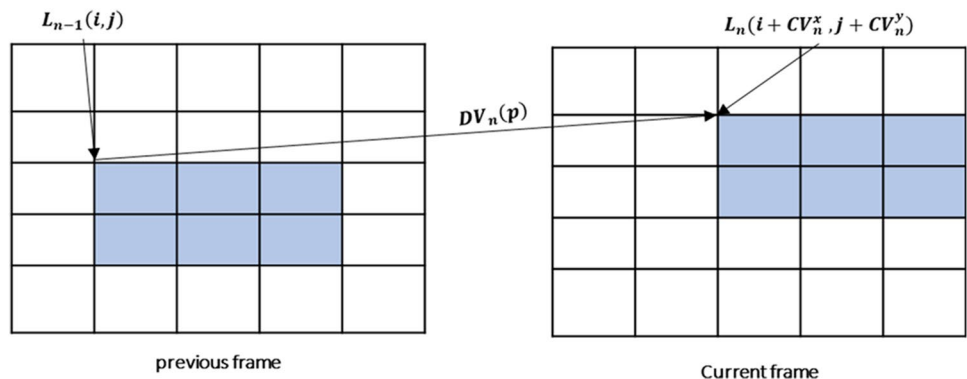
The searching area to find the maximum cross-correlation of the normalised projection curves between the slices of the current and the previous frames can be calculated as in Eq. (8):

$$\text{Searching area} = \frac{\text{number of } (CU_{\text{level1}}^p) + \text{number of } (CU_{\text{level1}}^{p-1})}{2}, \tag{8}$$

where  $(CU_{\text{level1}}^p)$  and  $(CU_{\text{level1}}^{p-1})$  are the encoded units of  $(32 \times 32)$  block at coding level 1 for the current and previous frames, respectively. Equation (8) is optimised between the encoding processing delay (an additional computational cost which resulted from the motion estimation calculations) and error resilience performance.

The difference vector  $DV_n(p)$  calculations are shown in Fig. 5.

Fig. 5 Difference vector calculation



### 3.2 Subdivision of non-active area

In general, people pay more attention to moving objects in the foreground due to the nature of the human visual system. Additionally, people focus more on the middle area in visualisation. To get the best trade-off between the coding efficiency and perceived quality, a non-active area is further subdivided into a high textured area which contains a high stationary spatial detail, and a passive (or flat) area which includes a fixed background area with lowest texture details. A simple subdivision example is shown in Fig. 6. The decoded video quality is reduced in a gradual way from high important areas to textured and passive areas, respectively.

An adaptive modified grey projection ( $AMGP_w$ ) based on the grey projection method in [20] is implemented to achieve a more accurate selection of motion-active slices.

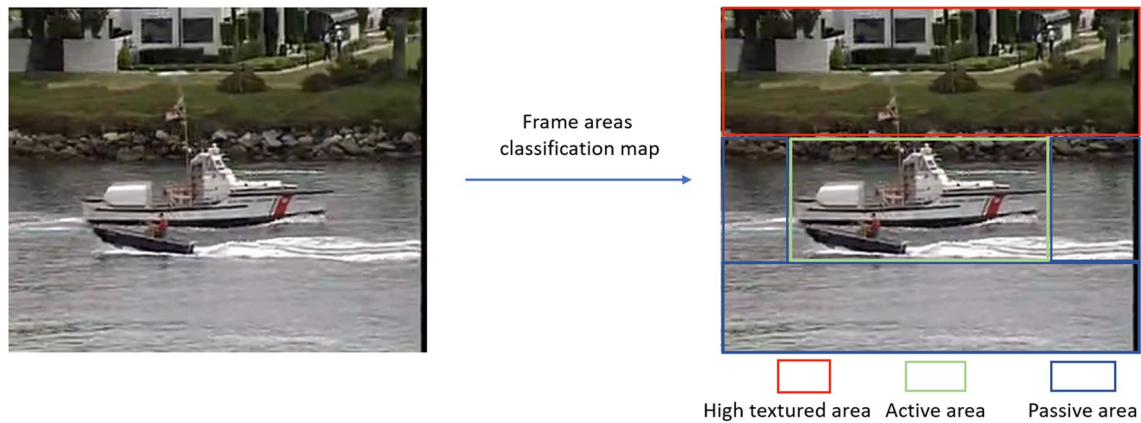
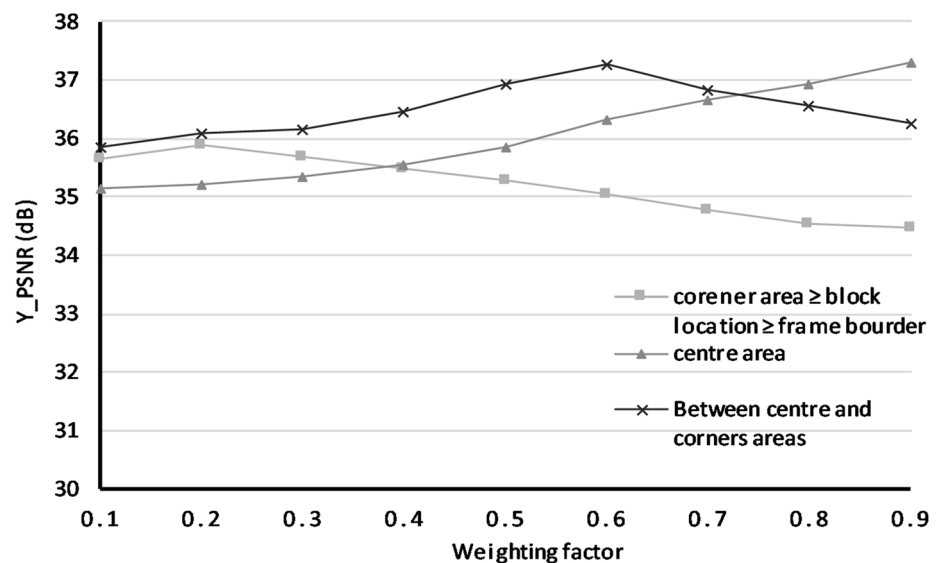


Fig. 6 Division areas in the ASE algorithm

Fig. 7 Weighting factors three for three ASE areas



To find out the relation between the encoded slice location within the frame and adaptive modified grey projection (AMGP) value, different weighting factors ranging 0.1–0.9 were objectively evaluated to obtain the best rate control optimisation with ASE algorithm implementation. The selected weighting factors were achieved during the trial and error optimisation experimental work on modified HM16.06 +ASE encoder. Due to limited space, one selected test result is presented in Fig. 7. The figure presents the obtained results for Akiyo video test sequence encoded with the frame rate (25 fps).

The idea of allocating different weighting factors at this stage is to optimise the proposed algorithm with the video coding efficiency. In the proposed algorithm, the frame content complexity is divided into three main areas. The decision of allocating weighting factor depends on the encoded



frame area’s sizes, which is proportional to the frame dimensions.

As the natural human visual system focuses more on the area of the central frame, this means the probability of the active areas to be encoded in the frame area (0.9). Thus, the condition of encoding with a higher weighting factor will be in the central area. A less probability of moving area will be the area between the central and the corner areas (0.6). The corner area will be allocated with the lowest weighting factor (0.1). The predefined weighting factors are selected based on the trial and errors experiments to be optimised with the intra-coding refresh.

A weighting factor is assigned for each frame region according to Eq. (9):

$$AMGP(W) = \begin{cases} 0.9, & \text{if the block location} \leq \text{bounding box of centre frame area} \\ 0.2, & \text{if the block location} \geq \text{corners frame area} \\ 0.6, & \text{otherwise} \end{cases} \tag{9}$$

The active area extraction process mainly depends on the calculated difference vector  $DV_n(p)$  and the weighting factor of the current frame. Finally, the most active slice areas in the current frame ( $p$ ) are encoded.

The encoding unit in the active map is encoded with intra mode as defined in Eq. (10):

$$AMGP_n(p) = \begin{cases} 1, & \text{if } AMGP_w \times DV_n(p) / \text{average} [DV_n(p)] > AMGP_{th} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where  $DV_n(p)$  is the calculated difference vector and  $AMGP_w$  is the calculated weighting factor for the currently encoded frame.

### 3.3 Non-active area selection

As discussed earlier, a non-active area for each frame is further divided into two regions according to the content features. A further subdivision region contributes to ensuring a perceived visual quality at frame transition level from high-quality regions (active areas) to lower quality (passive or high flat region areas). Furthermore, it helps to allocate larger weighting factors for active areas and lower bit budget to non-active regions, which results in the assignment of more bits to important frame areas. The extraction process of high textured areas from non-active areas is done using mean absolute difference (MAD) calculations between the current and previous frames. In this work, a (0.35) value is selected as a threshold point for generating high textured map as defined in the following equation, Eq. (11).

$$H_n(p) = \begin{cases} 1, & \text{if } H_{n-1}(p) < \text{threshold} \\ 0, & \text{elsewhere} \end{cases} \tag{11}$$

where  $H_{n-1}(p)$  refers to macroblock in the previous frame. Then, the remaining map areas are extracted and encoded as the lowest complex areas (refer to passive areas).

At each slice header, a full set of reference picture list is extracted at the decoded picture buffer (DPB). To identify whether the current slice is suitable to be used in the prediction process or not, a reference picture set (RPS) data at the slice header is compared with the referenced pictures at DPB.

For error detection and recovery purposes, a feedback channel from the decoder is used to notify the encoder about the occurred errors. The H.265/HEVC codec use a flag named (used-by-currng-pic-X-flag). The encoder parses the slice header and checks the flag activation [21].

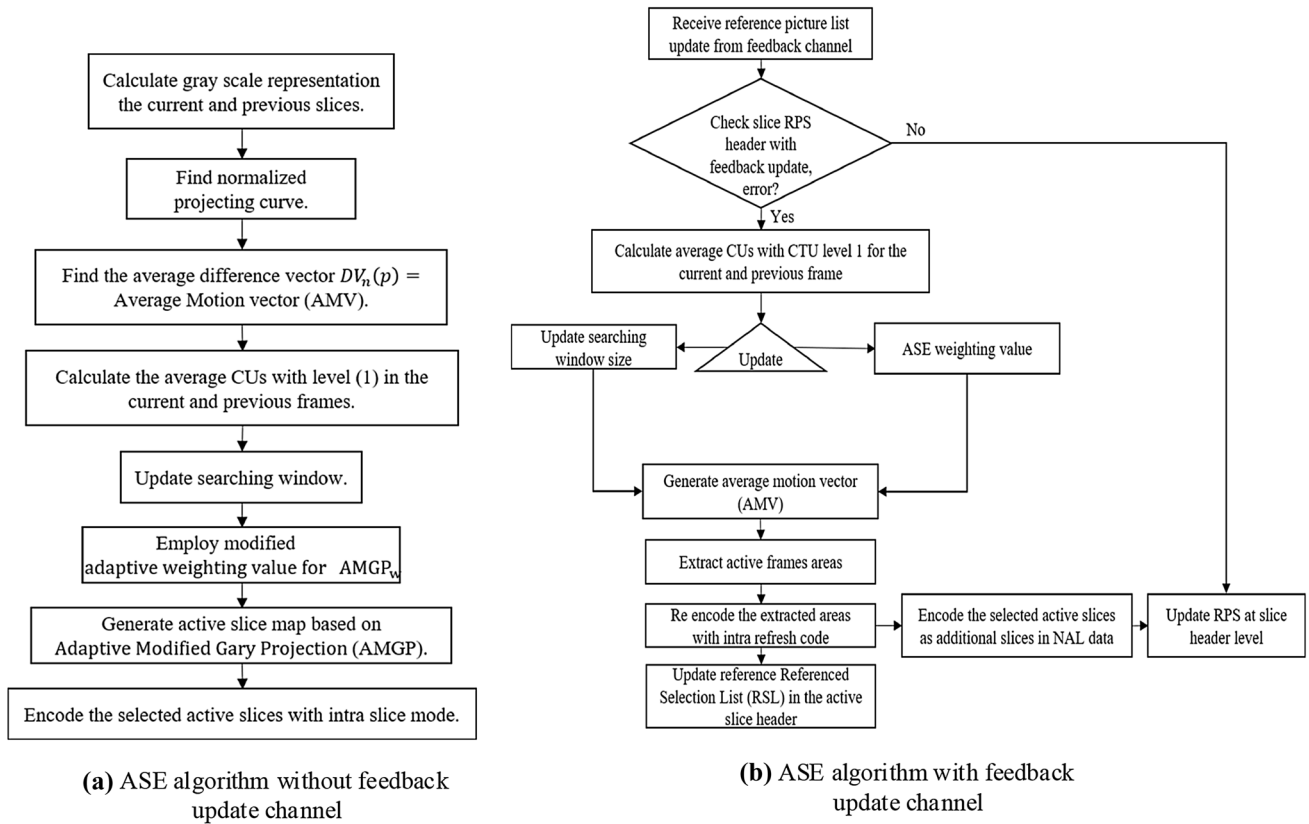
At the decoder side, a slice header RPS is checked against the available reference pictures at the DPB. If there is an update from RPS at the slice header, but there is no update available at DPB, it will consider this slice as not used in the current prediction process. However, if the flag is activated, then the encoded current slice is intended to be used

in the prediction process, but there is loss or corruption in the reference pictures at the decoder.

A flowchart illustrating the proposed ASE algorithm with feedback channel implementations is depicted in Fig. 8a.

#### 3.3.1 Error resilience algorithm with feedback update

The proposed error resilience algorithm is further extended to work with receiving feedback update the acknowledgement (ACK) signals to enhance perceived visual quality at the decoder. The H.265/HEVC coding system requires a feedback channel to locate a damaged slice. To get a more accurate error localisation, the segment header information of the corrupted slices is sent back via a feedback channel to the encoder. This header information contains the most recent update about the reference picture list which includes the address of the most recent erroneous slice. A flowchart of the proposed algorithm with feedback channel implementation is shown in Fig. 8b.



**Fig. 8** Adaptive slice encoding flowchart. **a** ASE algorithm without feedback. **b** ASE algorithm with feedback

### 3.4 Rate control adaptation of the proposed algorithm

The challenging tasks in the region of interest extraction implementation are how to keep the computation complexity of the extraction process to the minimum level and how to implement an accurate ROI process in low delay transmission.

In this algorithm, the encoder optimises the best trade-off between the number of intra-coded slices per frame and the coding efficiency target. A frame is divided into a passive or flat areas and high texture or complex areas. In HM16 reference software, a lambda rate control is used to optimise the encoding bit rate (bit allocation budget) and video quality (target quantisation parameters) [22]. The encoding bit rate is adjusted based on a target bit rate and picture buffer size for each group of pictures (GOP). Then, the encoder allocates the required encoding bit budget at the LCU level. Depending on the calculated target bit rate, the number of bits per pixel (bpp) is measured depending on the rate distortion Eq. (12):

$$\lambda = \alpha \cdot \text{bpp}^\beta, \quad (12)$$

where bpp is bit per pixel and  $\alpha$  and  $\beta$  are predefined parameters values. Once  $\lambda$  is calculated, a QP (delta quant) value can be obtained using Eq. (13) and quantisation step size using Eq. (14).

$$\text{QP} = 4.2 \ln \lambda + 13.7, \quad (13)$$

$$\text{Q\_step} = 1 + 2^{(1/6)}. \quad (14)$$

#### 3.4.1 Network testbed setup

The NS3 is chosen to be installed on Linux operating system. The long-term evolution (LTE) module is embedded within the NS3 environment. The LTE module simulates the core LTE network. To integrate the NS3 simulator embedded with LTE network module with a real physical Ethernet interface, a hardware in loop (HIL) platform in [24] is employed. Each node in NS3/LTE network is connected using carrier-sense multiple access (CSMA) scheme. LTE serving gateway (SGW)/packet data network gateway (PGW) uses a point to point internet connection. The NS3/LTE network simulator is configured with network configuration parameters reported in Table 1.

**Table 1** LTE network parameters

Network parameter	Value
Physical layer profile	Orthogonal frequency-division multiple access (OFDMA), full frequency-division duplexing (FDD) Downlink: 5 MHz Uplink: 1.4 MHz
Path loss model	Friis free space model
User equipment (UE) noise figure	8 db
Antenna type	Isotropic antenna
Transmission mode	Single input single output
Base station (eNB) operating power/noise figure	50 dbm/4 db

### 3.4.2 Hardware and software testbed setup

Three PCs are used in the experimental work. Two PCs are implemented as PC video server (Dell T410 Power Edge server).

CPU: Quad-core 2.35 GHz, RAM: 16 GB, operating system: Microsoft Windows 10 and video receiver (Dell XPS, CPU: Intel Core i5-7200 @2.5 GHz, RAM: 8 GB, operating system: Microsoft Windows 10). Open-source network simulator version 3 (NS3) is installed on separated PC (HP Compac 8200, CPU: Core i5-2500 s, RAM: 8 GB, operating system: Ubuntu server 15.04). The open-source cross-platform multimedia player (VLC) is used to stream the video test sequences at the sender side, and at the receiver used to visualise the perceived visual quality.

### 3.4.3 Error-prone environment setup

For evaluation the performance of the proposed algorithm under error-prone conditions, various packet loss rates are injected into encoded video bitstreams. A modified version of NAL unit loss software produced in [25] is used to generate different packet rates to be injected on the encoded bitstream. It is utilised to support the NAL unit structure of H.265/HEVC coding standard.

## 4 Experimental setup

In this section, the hardware and software experimental setup and encoding video configurations are presented. The main objectives are to test the performance of the proposed algorithm with different error-prone conditions, encoding bit rates, in addition to the computational complexity of the modified video reference HM16 software.

Pre-selected video test sequences are chosen in the experiments. The input video test sequences are in raw format (YUV) with colour space format 4:2:0. They are classified into two groups; the classification is done according to their texture video information and motion activity speed. Class

A: represents video sequences with low texture details and slow-motion activity. Class B: represents video sequences with high texture details and high-motion activity. The video sequences characteristics are presented in Table 2.

## 5 Experimental results and discussion

For each video test sequence, 30 test times with different seeds are tested and the averaged Y\_PSNR values are recorded. The process involves comparing the ASE algorithm performance with default reference software (HM16.06), region-based error-resilient scheme (ROI)

**Table 2** Characteristics of the test video sequence [23]

No.	Sequence name	Resolution	Number of frames	Video class
1	Hall monitor	QCIF, CIF	300	A
2	Mobile	QCIF, CIF	300	A
3	Bus	CIF	150	B
3	Container	QCIF, CIF	300	A
4	Grandma	QCIF	870	A
5	Akiyo	QCIF, CIF	300	A
6	Miss America	QCIF	150	A
7	Bridge—close (far distance)	QCIF, CIF	2001	A
8	Mother and daughter	QCIF, CIF	300	A
9	News	QCIF, CIF	300	
10	Bridge (far)	QCIF, CIF	2101	A
11	Bridge—close (near distance)	QCIF, CIF	2001	A
12	Coastguard	QCIF, CIF	300	B
13	Claire	QCIF	494	A
14	Carphone	QCIF	382	B
15	Highway	QCIF, CIF	2000	A
16	Salesman	QCIF	449	B
17	Silent	QCIF, CIF	300	B
18	Suzie	QCIF	150	B

algorithm in [26], and improved region of interest (IROI) algorithm in [27]. All video sequences were randomly injected with packet loss rate PLRs ranging 2–18% using packet loss rate generator software and encoding setting as in Table 3. The experimental work results are shown in Fig. 9a.

**Table 3** Encoding parameters of modified HM 16.06

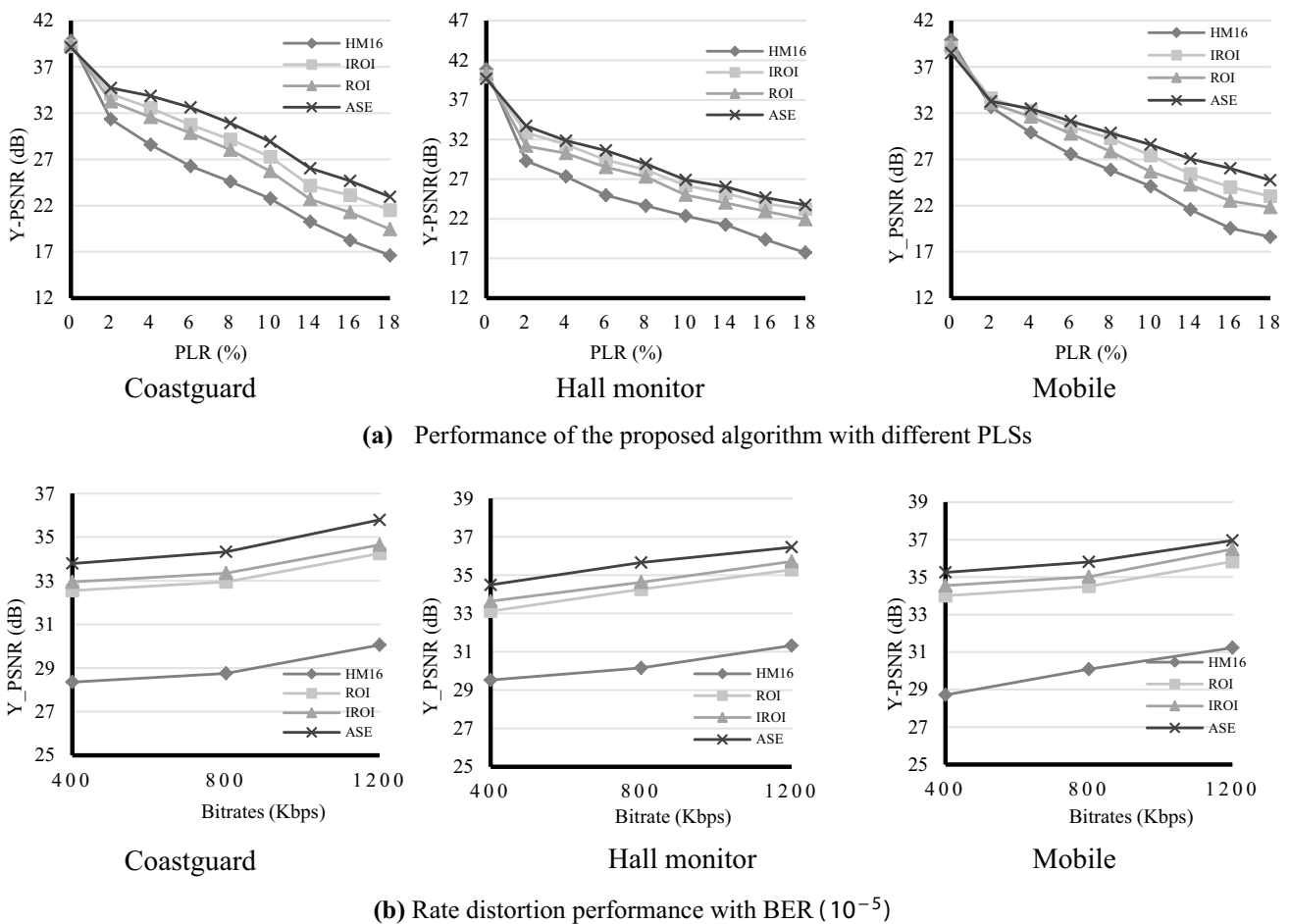
Encoding parameter	Configuration
Profile	Main
Encoding GOP size	8 Frames
Rate control	Disabled
Filtering	Enabled
Search range	64
bit depth	8
Intra mode	1st frame in each GOP
(Max/min) transform unit size	32/4
Largest coding unit (LCU) size	64×64
Frames to be encoded (-f)	300

The evaluation includes both error-free and error-prone condition. In error-prone condition, video test sequences are injected with random bit error generated with  $BER = (1 \times 10^{-5})$ . All the tested video sequences are in CIF resolution.

Further evaluation tests are performed to measure the effectiveness of the proposed ASE algorithm under erroneous packet loss conditions with different encoding bit rates. The objective quality evaluation for ASE algorithm with different encoding bit rates is shown in Fig. 9b.

Table 4 reports the obtained Y-PSNR gain with and without feedback update for the three video sequences (coastguard, hall, and mobile sequences) in CIF resolution. Notably, the average (Y-PSNR) of ASE algorithm with different PLRs has improved by (4.521 db), (2.283 db), and (1.076 db) compared to HM16 for H.265/HEVC codec, ROI, and IROI algorithms, respectively.

In error-free conditions (PLR = 0%), Y-PSNR of the ASE algorithm is reduced by (−1.096 db; HM16), (−0.605 db; ROI), and (−0.318 db; IROI).



**Fig. 9** Video quality evaluation of the proposed ASE algorithm. **a** Performance of the proposed algorithm with different PLRs. **b** Rate distortion performance with BER ( $10^{-5}$ )

**Table 4** ASE performance comparison in terms of Y-PSNR (db)

Video sequence	Error-free condition Y-PSNR values (db)			Error-prone condition Y-PSNR values (db)		
	ASE vs HM16.06	ASE vs ROI	ASE vs IROI	ASE vs HM16.06	ASE vs ROI	ASE vs IROI
Coastguard	-0.702	-0.323	-0.126	5.754	2.861	1.514
Hall	-1.196	-0.623	-0.274	3.636	1.913	0.756
Mobile	-1.391	-0.871	-0.555	4.173	2.076	0.958
Average	-1.096	-0.605	-0.318	4.521	2.283	1.076

The obtained test results in Table 4 indicate that the performance of the proposed algorithm is less effective in error-free conditions. The complex processing part in the proposed algorithm is in the frame partitioning depending on complexity of the frame content. These three transitions areas contribute to obtaining the best balance between coding deficiency and error resilience performance.

### 5.1 Frame by frame video quality assessment

The proposed ASE algorithm is further evaluated using subjective quality assessment. Pre-selected frames are extracted from the raw video test sequence for video test assessment. The video sequence named Coastguard sequence with CIF resolution was encoded at 30 fps. Packet errors at a mean rate of 2% are injected into the video test sequence. The obtained results are shown in Fig. 10, using frame by frame visual quality assessment. It can be seen from the decoded frames that ASE algorithm produces a better perceptual

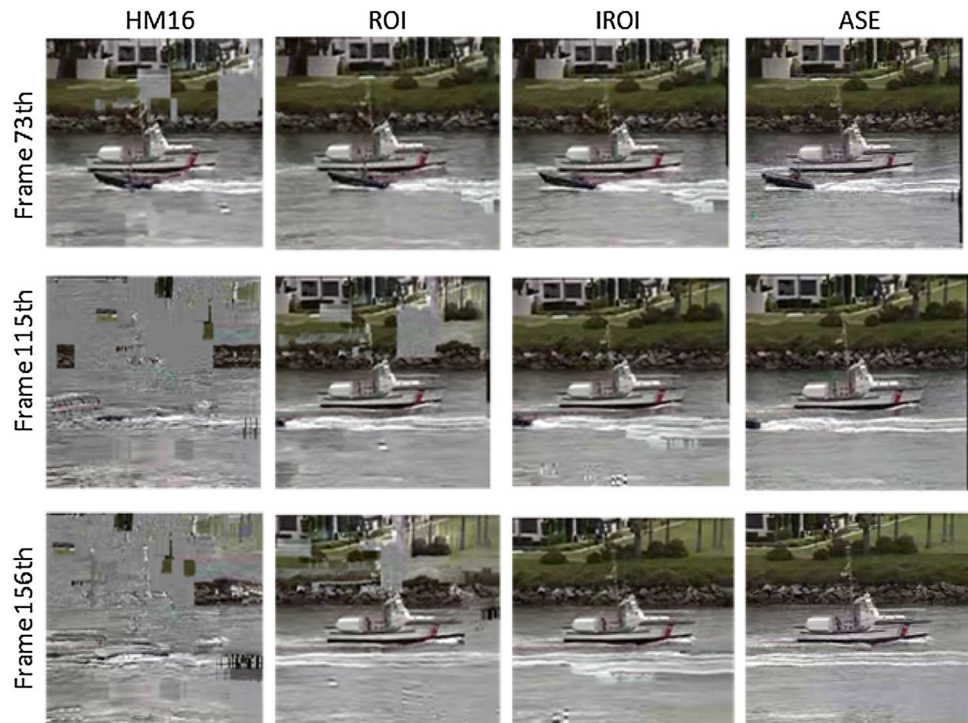
visual quality compared to the reference error resilience algorithms.

### 5.2 Network congestion and time delay

The proposed algorithm is further evaluated by streaming the encoded video in LTE network in terms of network load. The number of the end users is ranging from 10 to 30 users per base station. The experimental work is conducted depending on the settings described in the network testbed setup section. A frame copy concealment is used at the decoder to avoid failure in the decoding process. The two main objectives are targeted in this experimental work. The first one is to show the effect of a different number of LTE network clients over a shared bandwidth, on objective decoded video quality.

The second one is the start-up time, as it is a critical factor for meeting the user's quality of experience requirements [28].

**Fig. 10** Frame by frame visual quality assessment



The start-up time is defined as the required time that the decoder buffer needs to process before displaying the decoded pictures.

The authors in Ref. [28] recommended that the start-up video delay in video streaming applications should not be more than 2 s. In this experimental work, we chose 1000 ms and 500 ms as realistic use cases.

Based on the network configuration parameters and the network testbed which was described earlier, the encoding configuration settings are reported in Table 3. Eighteen video sequences were selected and the average for each tested number of clients group (10, 20, 30 end users) is recorded. A network load is categorised into three levels: light (10 users), medium (20 users), and heavy load (30 users).

Each test is repeated ten times to get more reliable verifications.

Our algorithm is integrated with the video evaluation platform.

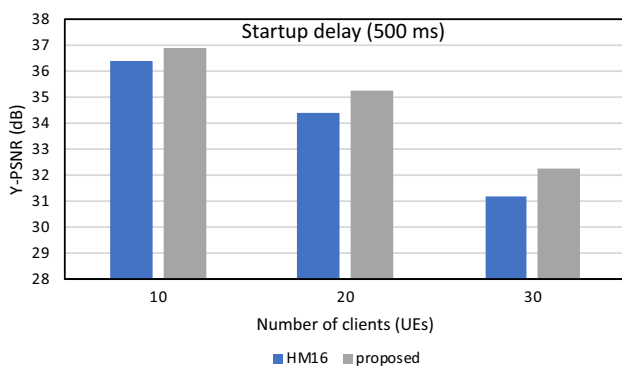
The average Y-PSNR results for the pre-selected test sequences are recorded for network load behaviour.

Figures 11 and 12 show the effect of increasing the number of users on the perceptual visual quality in terms of the Y-PSNR at the decoder.

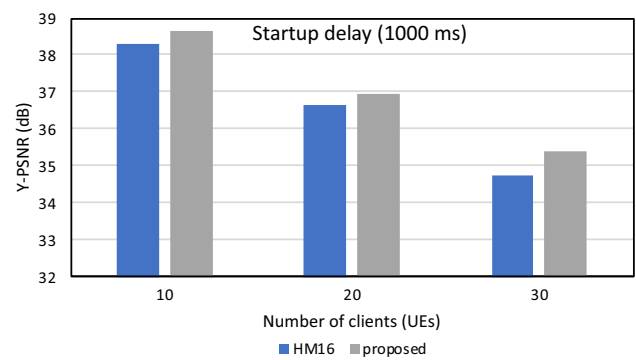
It is obvious when we increase the number of clients, the proposed algorithm outperforms the default reference software.

When the encoded video sequences is streamed with a high network load, the dropped packets are increased significantly due to network congestion. Therefore, the objective video quality is deteriorated further at a higher network load with a shared bandwidth network.

It is noted from Figs. 11 and 12 results that the start-up delay takes longer, the number of decoded redundant intra slices increased, in return the probability of recovered the damaged slices increased due to using the encoded redundant slices to resynchronise the corrupted areas. Hence,



**Fig. 11** Average Y-PSNR for video streaming with start-up delay (500 ms)



**Fig. 12** Average Y-PSNR for video streaming with start-up delay (1000 ms)

in real-time video streaming, the decoded video quality is affected by two factors in the proposed algorithm: the encoding bit rate and the GOP structure used.

In general, for a GOP with longer encoded length, the coding efficiency will be higher due to less encoding bit rate required. However, a high GOP size means more dependent frames are encoded between I-frame interval; this longer interval lowering the encoding error resilience. This is in addition to the increase in the picture decoding delay. The acceptable total delay could be between 500 ms and 1000 ms before the picture is ready for display.

### 5.3 Computational complexity

The aim of this work is to determine the impact of encoding/decoding computational complexity from a processing time perspective. It is worth noting that the reference software (HM) is mainly used for developing H.265/HEVC video coding algorithms. Moreover, it does not practically support real-time video encoding applications. Although the HM suffers from slow speed of execution of encoding and decoding, some attempts have been made during the regular reference development process, i.e. HM versions.

In the experimental work, we measured the encoding/decoding processing time for the proposed algorithm, and then compared the obtained results with the standard default reference software HM16.

The video test sequences are encoded using the same encoding setting reported in Table 3. For video test sequences, the experiment was implemented with the same video test sequences reported in Table 2.

As stated in the JCT-VC common test conditions [29], there are mainly three encoding configuration settings: low delay-B slices, all intra (AI), random access (RA). In low delay-B configuration, the first frame is encoded with intra frame type and the following frames are encoded as redundant frames with bi-prediction B-frames which give higher

coding efficiency and coding delay than redundant frames with uni-prediction P-frames [21, 30].

For (AI) configuration, intra mode is used to encode the whole video sequence. This encoding type gives low encoding time, but requires very high encoding rates.

For (RA) encoding configuration, the encoded video frames are organised in hierarchical B structure. This mode gives higher compression efficiency than other encoding modes. However, it is not suitable for low delay applications, because it requires more processing for reorganising the decoding pictures order at the far end decoder.

In this paper, we evaluated the proposed algorithm with low delay-B configuration mode and encoded with QP at (32). The execution time of encoding/decoding process for each video test sequence without feedback channel is reported in Table 5.

The table presents the encoding and decoding run times as an indication of algorithmic complexity compared with the default reference (HM 16).

To examine the processing time of the proposed error resilience algorithm, the average processing time of enc/decoding 18 video test sequences is obtained. Figure 13 shows the required increase in the enc/decoding processing time percentage compared with HM 16 software without the error resilience tool implementation.

It is noted from the obtained results that the encoder consumes more time than that at the decoder (Fig. 13). The additional computation time in the modified HM16 encoder has arisen from rate control adaptation for

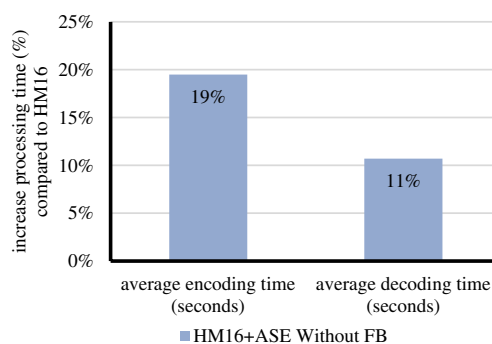


Fig. 13 Average increase in encoding/decoding processing time compared to HM16

encoding different areas with different encoding bit rates, and also allocating different quantisation parameters at different largest coding units levels. In addition to the picture segmentation processing into different subregion areas, this process includes differencing method as additional computations between the current and previous frames.

For additional computations at the decoder, the high amount of time spent on parsing the redundant slices leads to an increase in reference sample generation at the decoded picture buffer. Furthermore, an additional part of the decoding process is spent on the scanning process at the slice boundaries level, in addition to the reference sample generation for the intra slices prediction. The increasing time in both encoding and decoding process

Table 5 Encoding and decoding time of the ASE algorithm compared to the HM16 reference software

Video test sequence	Number of frames	Encoding time (s)		Decoding time (s)	
		HM16	HM16+ASE	HM16	HM16+ASE
1 Hall monitor	300	575	664	6.5	7.4
2 Mobile	300	491	646	6.8	7.8
3 Container	150	275	298	4.3	5.2
4 Grandma	300	548	686	5.8	6.6
5 Akiyo	870	956	1232	9.3	10.2
6 Miss America	300	507	653	5.8	6.3
7 Bridge—close (far distance)	150	249	367	4.2	5.4
8 Mother—daughter	2001	1283	1492	16.3	17.4
9 News	300	501	647	6.2	7.1
10 Bridge—far	300	498	622	5.7	6.4
11 Bridge—close (near distance)	2101	1355	1498	15.2	16.4
12 Coastguard	2001	1294	1435	14.3	15.1
13 Claire	300	485	588	5.8	6.6
14 Carphone	494	684	809	7.3	8
15 Highway	382	538	653	6.6	7.6
16 Salesman	2000	1385	1509	16.5	17.5
17 Silent	449	649	844	8.4	9.2
18 Suzie	300	473	586	6.5	7.3
Average (s)		708	846	8.4	9.3

arises from adding a set of C++ classes for error resilience implementation.

## 6 Conclusions and recommendation

This paper presents an efficient H.265/HEVC error resilience algorithm to support low delay video delivery applications. The novelty of this algorithm lies in automatically selecting the most active frame regions and protecting them against transmission errors at the cost of an increase in the encoding bit rate overhead and encoding/decoding computational complexity. The proposed work also took into consideration the coding efficiency by subdividing the non-active regions into flat and high textured areas. Hence, the saving of the bit budget in non-active areas is achieved by spending larger portions of the available bit budget on active frame areas and obtaining the best trade-off between the coding efficiency and error resilience performance.

We conducted several simulation scenarios for evaluating the proposed algorithm. Firstly, we presented different network testbeds used for a modified video codec performance evaluation. The experimental work was conducted in error-prone and error-free environments, with averaging packet loss rates ranging from 2 to 18%. The obtained results show the proposed algorithm yields a Y-PSNR gain of 4.52 db over the HM16 reference software, and outperforms the state-of-the-art algorithms ROI and IROI by 2.28 db and 1.07 db, respectively. However, in error-free conditions, the proposed algorithm suffered the highest db loss of  $-1.09$  against the default software HM16.

Furthermore, the encoding and decoding processing time of the tested video sequences is analysed and reported in terms of computational complexity. The processing time results of the proposed algorithm showed that the encoding and decoding time increased by 19% and 11%, respectively.

The algorithm is further investigated with start-up video play delay (0.5 s and 1 s), in long-term evolution LTE network. The obtained results showed that when the start-up delay increases (0.5–1 s) at the decoder, the objective decoded video quality remarkably increases (1 db on average). These results indicate that the proposed algorithm, without a feedback channel, can be used in low delay video applications. Our future work includes implementing a Gilbert–Elliott model with the proposed algorithm for providing real-time quality service estimation. The model will enable the automatic control and adjustment of the encoding parameters.

**Acknowledgements** This publication was made possible by NPRP grant 9-181-1-036 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors ([www.ceproqha.qa](http://www.ceproqha.qa)). The authors would

also like to thank Mr. Marc Pelletreau, the MIA Multimedia team, the Art Curators and the management staff of the Museum of Islamic art, Doha, Qatar, for their support in using their Media Lab facilities during the experimental work.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Nightingale, J., Wang, Q., Grecos, C.: HEVStream: a framework for streaming and evaluation of high efficiency video coding (HEVC) content in loss-prone networks. *IEEE Trans Consum Electron* **58**(2), 404–412 (2012)
2. International Telecommunication Union Telecommunication Standardization Sector (ITU-T) and International Organization for standardization/international electrotechnical commission (ISO/IEC), High efficiency video coding, Rec. H.265/ISO/IEC 23008–2 (2013)
3. Garcia, R., Kalva, H.: Subjective evaluation of HEVC and AVC/H.264 in mobile environments. *IEEE Trans Consum Electron* **60**(1), 116–123 (2014)
4. Sjöberg, R., Boyce, J.: HEVC high-level syntax. In: High efficiency video coding (HEVC): algorithms and architectures, pp 13–48 (2014)
5. Flynn, D., et al.: Overview of the range extensions for the HEVC standard: tools, profiles, and performance. *IEEE Trans. Circuits Syst. Video Technol.* **26**(1), 4–19 (2016)
6. Psannis, K.E.: HEVC in wireless environments. *J Real-Time Image Process.* **12**(2), 509–516 (2015)
7. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012)
8. Yousfi, R., Ben Omor, M., Damak, T., Ben Ayed, M.A., Masmoudi, N.: JEM-post HEVC vs. HM-H265/HEVC performance and subjective quality comparison based on QVA metric. In: 2018 4th International Conference Advances Technology Signal Image Process. ATSIIP 2018, pp. 1–4, (2018)
9. Usman, M., He, X., Lam, K.M., Xu, M., Bokhari, S.M.M., Chen, J.: Frame interpolation for cloud-based mobile video streaming. *IEEE Trans. Multimed.* **18**(5), 831–839 (2016)
10. Chen, H., Zhao, C., Sun, M.T., Drake, A.: Adaptive intra-refresh for low-delay error-resilient video coding. *J. Vis. Commun. Image Represent.* **31**, 294–304 (2015)
11. Wang, D.Y., Sun, S.X.: Region-based rate control and bit allocation for video coding. In: 2008 International Conference Apperceiving Computing Intelligence Analysis. ICACIA 2008, vol. 8, no. 1, pp. 147–151, (2008)
12. Li, H., Wang, Z., Cui, H., Tang, K.: An improved ROI-based rate control algorithm for H.264/AVC. In: International Conference Signal Processing Proceedings, ICSP, vol. 2, (2007)
13. Liu, Y., Li, Z.G., Soh, Y.C.: Region-of-interest based resource allocation for conversational video communication of H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **18**(1), 134–139 (2008)
14. Liu, Y., Li, Z.G., Soh, Y.C.: A novel rate control scheme for low delay video communication of H.264/AVC standard. *IEEE Trans Circuits Syst Video Technol* **17**(1), 68–78 (2007)



15. Song, H., Kuo, C.-C.J.: A region-based H.263+ codec and its rate control for low VBR video. *IEEE Trans Multimed* **6**(3), 489–500 (2004)
16. Hu, H.M., Lin, W., Li, B., Sun, M.T.: A region-based rate-control scheme using inter-layer information for H.264/SVC. *J. Vis. Commun. Image Represent.* **22**(7), 615–626 (2011)
17. Wang, F., Wu, W., Lou, Y., Yang, A.: An adaptive H264 video protection scheme for video conferencing. *IEEE Vis. Commun. Image Proces*, pp 1–4, (2011)
18. Ohm, J., Sullivan, G.J.J.: High efficiency video coding: the next frontier in video compression. *IEEE Signal Process. Mag.* **30**(1), 152–158 (2013)
19. Stockhammer, T., Hannuksela, M.M., Wiegand, T.: H. 264/AVC in wireless environments. *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 657–673 (2003)
20. Ren, G., Li, P., Wang, G.: A novel hybrid coarse-to-fine digital image stabilization algorithm. *Inform. Technol J* **9**(7), 1390–1396 (2010)
21. Sjöberg, R., et al.: Overview of HEVC high-level syntax and reference picture management. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1858–1870 (2012)
22. Li, B., Li, H., Li, L., Zhang, J.:  $\lambda$  domain rate control algorithm for high efficiency video coding. *IEEE Trans. Image Process.* **23**(9), 3841–3854 (2014)
23. <http://trace.eas.asu.edu/yuv/yuv.html>. Accessed 12 Mar 2019
24. Fouda, A., et al.: Real-time video streaming over NS3-based emulated LTE networks. *Int J Electron Commun Comput Technol* **4**(3), 659–663 (2014)
25. Wenger, S. NAL Unit Loss Software. Document JCTVC-H0072, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). (2012). [http://phenix.intevry.fr/jct/doc\\_end\\_user/current\\_document.php?id=4373](http://phenix.intevry.fr/jct/doc_end_user/current_document.php?id=4373). Accessed 3 July 2018
26. Maung, H.M., Aramvith, S., Miyanaga, Y.: Region-of-interest based error resilient method for HEVC video transmission. In: 2015 15th International symposium on information and communication technology, ISC 2015, pp. 241–244, (2016)
27. Maung H., Aramvith, S., Miyanaga, Y.: Improved region-of-interest based rate control for error resilient HEVC framework. In: International conference on digital signal processing. DSP, pp. 286–290, (2017)
28. Krishnan, S.S., Sitaraman, R.K.: Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *IEEE/ACM Trans. Netw.* **21**(6), 2001–2014 (2013)
29. Bossen, F.: Common test conditions and software reference configurations, document JCTVC-L1100 of JCT-VC, Geneva (2013)
30. Zhao, T., Wang, Z., Kwong, S.: Flexible mode selection and complexity allocation in high efficiency video coding. *IEEE J Sel. Top. Signal Process.* **7**(6), 1135–1144 (2013)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Taha Alfaqheri** is a PhD student and research assistant in the Electronic and Computer Engineering Department at Brunel University London, UK. His main research interests are developing video coding standard algorithms and 2D/3D visual processing and communication systems.



**Abdul Hamid Sadka** has nearly 25 years' worth of academic leadership and research excellence in the area of signal processing and visual communications systems. He is the Director of "Brunel Digital Science and Technology" Research Hub that comprises over 150 research active academics working in 7 research clusters on a wide range of areas in digital science and technology. He is also the Founding Director of the Centre for Media Communications Research (CMCR) and Director

of Masters of Engineering studies. He is the former Head of the Department of Electronic and Computer Engineering (ECE) at Brunel University (Oct 06–Jul 12). He has a very research and management-focused academic career underlined by a commitment to industry engagement and high profile academic output. He has 200+ publications in refereed journals and conferences, three patents and a flagship textbook entitled "Compressed video communications" published by J. Wiley in 2002. He has thus far managed to attract circa £13M worth of research grants and contracts and has supervised nearly 50 research assistants and PhD students to full conclusion. He frequently serves on influential advisory boards and international evaluation panels, and provides expert consultancy services to the Telecom/ICT industry as well as corporate law firms in the area of 2D/3D video compression and visual media processing. He served as elected member (2011–2014) on the Steering Board of the NEM European technology platform. He is a chartered engineer (CEng), Fellow of HEA, Fellow of IET and Senior Member of IEEE.