IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 23, 2020, accepted July 2, 2020, date of publication July 10, 2020, date of current version July 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008516

Efficient Unequal Error Protection Techniques for Tile-Based Transmission of HEVC Videos

BIJAYA PAUDEL, (Member, IEEE), AND SINA VAFI, (Senior Member, IEEE)
College of Engineering, IT, and Environment, Charles Darwin University, Casuarina, NT 0810, Australia
Corresponding author: Bijaya Paudel (bijaya.paudel@cdu.edu.au)

ABSTRACT The video bitstream compressed by the efficient high efficiency video coding (HEVC) standard is extremely vulnerable to the channel error. For robust transmission of such compressed videos, techniques can be formed based on the specific characteristics of the compression standard. With the introduction of new coding features in HEVC, such as flexible block partitioning and tiles, unequal error protection (UEP) schemes are proposed in this paper aiming to enhance the quality of the important regions. The proposed algorithms are implemented in two and three-levels. For the two-level UEP, tiles are prioritized based on their motion density, which is defined as the ratio of motion vector magnitudes to the block size in a compressed video frame. Furthermore, a three-level UEP is proposed to improve the protection of low-important tiles, which may include moving objects. For this purpose, clustering algorithms, utilizing kernel density estimation (KDE) and density-based spatial clustering of applications with noise (DBSCAN), are modified based on the motion density of coding tree units (CTUs). Indeed, this represents implementation of an object detection algorithm based in compressed domain. Simulation results confirm that proposed UEP schemes achieve better objective quality compared to conventional UEP and equal error protection (EEP) approaches.

INDEX TERMS HEVC, motion density, object extraction, tiles, unequal error protection.

I. INTRODUCTION

The usage of services such as video conferencing, multimedia messaging, video sharing and Internet TV has unprecedentedly grown in recent years. During transmission, the probability of error derived from the network congestion and interference presents significant challenges to achieve high video quality at the receivers' end. Therefore, adopting an efficient strategy in the transmission system is crucial to deliver video data at a high quality for an available bitrate. In this role, compression and error correction are essential techniques to maintain an adequate quality of the delivered video stream. In compression, redundancies are reduced so that the compressed video can be transmitted over the band-limited channel. High efficiency video coding (HEVC) is represented as the latest compression standard, whose efficiency is 50 % more than its predecessor, i.e. H.264/ advance video coding (AVC). However, due to its high compression rate, the resulting bitstream becomes highly vulnerable to errors occurred during the transmission [1].

Unequal error protection (UEP) is recognized as one of the most effective techniques for protection of the compressed video bitstream against the channel error. In this method, different parts of the bitstream are unequally protected based on their importance. Application of UEP in the data transmission over the noisy channel was discussed in [2]. Since then, the concept has been widely investigated for different applications. The early works relating to UEP scheme in video communications are presented in [3]–[5]. From the last decade, a number of UEP techniques were proposed to enhance protection of the compressed video bitstream. These techniques are based on specific characteristics of the compressed video such as scalability [1], group of pictures [5], flexible macroblock ordering [6] and data partitioning [7]. Despite the effectiveness of these techniques for videos compressed by H.264 standard, some of them including flexible macroblock ordering and data partitioning are not applicable to HEVC. Recently, some studies investigated UEP techniques suitable for HEVC bitstream, which are based on the scalable video coding [1], [8]. Scalable video coding results loss in compression efficiency compared to non-scalable coding. In addition, due to the strong spatio-temporal dependencies between scalable video layers, any error occurred in the

The associate editor coordinating the review of this manuscript and approving it for publication was Nilanjan Dey.

hierarchical encoding significantly deteriorates quality of the received video.

In HEVC, video frames can be partitioned into a number of tiles [9]. Tiles are a rectangular-shaped group of coding tree units (CTUs), which are separated by vertical and/or horizontal boundaries. They are designed to break prediction dependencies across the boundaries from each other. This is done by disabling both intra-prediction and motion vector prediction across tiles boundaries. Therefore, each tile can be independently processed. This feature improves video coding efficiency as multiple tiles can be processed at the same time. Video applications, such as video conferencing and video surveillance, benefit from region-of-interest functionality of tiles. Region based prioritized encoding of tiles can be done to achieve better visual quality in the specific region of frames. This idea is also extended for transmission of tiles, which are prioritized based on the region-of-interest. Furthermore, flexibility in defining tiles inside a picture increases their suitability in video applications.

In a conducted work, a content aware UEP scheme was presented for the transmission of HEVC frames [10]. A motion density based scheme was applied to identify important tiles in video frames. This paper extends this UEP scheme presented in [10] to provide a low-complex and a more efficient protection technique for HEVC bitstream. This is done based on the high temporal correlation between consecutive frames. Due to this correlation, temporally neighbouring tiles can be used to determine importance of the current tile. As a result, the newly proposed technique requires less process in determining the importance of tiles in the current frame. The importance of tiles is expressed in two levels, where motion density of each tile is compared with a threshold value.

Improvement on the video quality can be achieved by increasing protection of high-important tiles, whose motion densities are greater than the threshold value. There is usually a non-uniform distribution of motion density around the threshold value. Tiles containing moving objects tend to have very high values of motion densities compared to other tiles. These large value of motion densities affect the threshold value. As a result, most of the tiles in a frame are classified as the low-important one. In many situations, tiles that are in the neighbourhood of a high-important tile may contain part of a moving object. When such tiles are transmitted as low-important ones, that part of the object can be decoded with an error. Although it may not significantly impact the objective video quality, but results in degradation of the quality of video received by the viewer. Therefore, it is expected that by allocating better protection to such neighbouring tiles, the video quality will be improved. For this purpose, a three level UEP scheme is proposed, where low-important tiles in the video frame are further divided into low-high and low-low sub-levels. Simulation results indicate improvement on the video performance compared to the previous UEP and equal error protection (EEP) techniques.

The paper is organized as follows. Section II discusses motion density, which is applied to detect important regions in a video frame. An analysis of the correlation between tiles in consecutive frames is done in section III. The proposed two-level UEP scheme is presented in section IV. Section V discusses simulation results for two-level UEP method. Section VI discusses the implementation of three-level UEP scheme. Simulation results for three-level UEP are given in Section VII. Finally, conclusions are provided in Section VIII.

II. PRELIMINARIES

A. CORRELATION BETWEEN COLLOCATED TILES IN CONSECUTIVE FRAMES

In video sequences, the correlation coefficient between blocks[1] is calculated as follows:

$$\rho = \frac{\sum_{i=1}^{N_P}(X_i - X') \times \sum_{i=1}^{N_P}(Y_i - Y')}{\sqrt{\left[\sum_{i=1}^{N_P}(X_i - X')^2 \times \sum_{i=1}^{N_P}(Y_i - Y')^2\right]}}, \qquad (1)$$

where N_P is the total number of pixels inside one block. X_i and Y_i are luminance values of i^{th} pixel in blocks of consecutive frames. X' and Y' are the mean values of X_is and Y_is, respectively.

As expected, the above-mentioned correlation coefficient is between -1 and 1. In this paper, if $|\rho| < 0.5$, a low correlation between blocks is concluded. Otherwise, considered blocks have a high correlation. A high value of the correlation coefficient means that there is a high similarity between considered blocks. In this case, blocks either have consistent motion activity or belong to the same object.

Such this definition can be applied for determining the correlation between collocated tiles in two consecutive frames. For this purpose, in equation (1) N_P is represented as the number of pixels inside a tile. Figure 1 shows the correlation coefficients calculated for tiles in *Sunflower.yuv* and *Pedestrian.yuv* video sequences. In the obtained results, correlation coefficients fluctuate between 0.5 and 0.98 for 10^{th}

[1] A block is being considered as a group of pixels inside the frame.

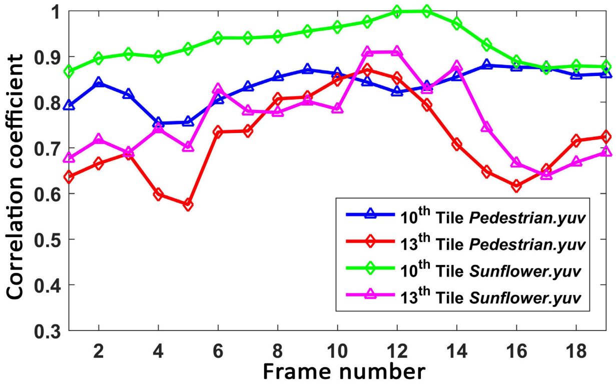


FIGURE 1. Correlation between collocated tiles in consecutive frames.

and 13th tiles of the given video sequences, respectively. For all frames, correlation coefficient of the 10th tile of the *Sunflower.yuv* video is greater than or equal to 0.85. This is because the tile had the same object throughout the video sequence. On the other hand, the correlation coefficients for 13th tile is comparatively lower than those of the 10th. These are due to the presence of motion activity in the tile.

Collocated tiles may not share similar motion information. A tile in a frame may have higher or lower motion activity compared to the collocated tile of the previous frame. Therefore, measurement of motion activity of tiles is essential.

B. MOTION DENSITY

In HEVC, compressed video frames constitute a number of coding tree units (CTUs), which are divided into coding units (CUs). CUs may be of variable sizes, which are determined based on the size of CTU and coding tree depth. These CUs represent the basic processing unit, in which a coding mode is assigned. When a CU is encoded in inter-prediction mode, it splits into one, two or four prediction units (PUs). The motion vectors constituted by these PUs represent the displacement of CUs between two frames. Therefore, it is obvious that the motion vector magnitude, calculated as an absolute value of horizontal and vertical motion vectors, and size of a CU are important to determine the motion activity of a CU. This concludes importance of motion density (MD) in determining motion activity of CUs, which is defined as follows:

$$\mathbf{MD}_{CU} \triangleq \frac{\mathbf{MV}}{\mathbf{S}}, \tag{2}$$

where **MV** is the total magnitude of motion vectors inside a CU and **S** is the index determined based on the size of CU. For a CU including $M \times M$ pixels, where $M = 2^n$ and $3 \leq n \leq 6$, S is given by:

$$\mathbf{S} = \frac{M}{8}, \quad 8 \leq M \leq 64. \tag{3}$$

The motion density of a CTU is determined as the average of motion densities of all of its CUs. That is:

$$\mathbf{MD}_{CTU} = \frac{1}{N_{CU}} \sum_{k=1}^{N_{CU}} \mathbf{MD}_{CU}^{k}, \tag{4}$$

where N_{CU} is the number of CUs and $\mathbf{MD}_{CU}^{k}$ is the motion density of k^{th} CU inside the CTU, calculated by equation (2).

Similarly, motion density of a tile can be defined on the basis of motion densities of its CTUs, as follows:

$$\mathbf{MD}_{tile} = \frac{1}{N_{CTU}} \sum_{k=1}^{N_{CTU}} \mathbf{MD}_{CTU}^{k}, \tag{5}$$

where N_{CTU} is the number of CTUs inside the tile and $\mathbf{MD}_{CTU}^{k}$ is the motion density of k^{th} CTU obtained from equation (4).

Figure 2 shows the motion density of tiles of Sunflower.yuv and Pedestrian.yuv videos. It is observed that a tile has different motion densities in different frames. Even when a tile shares visual similarities in consecutive frames, as indicated by Figure 1, they may have different motion activities.

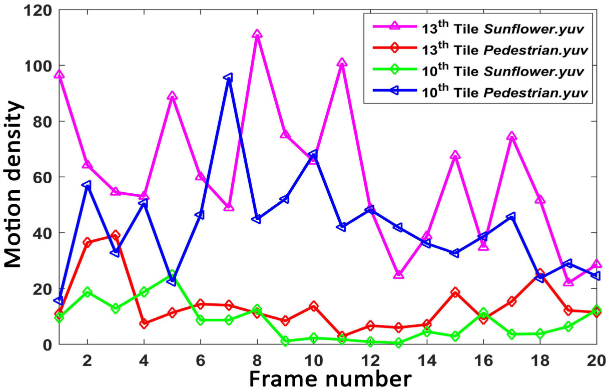


FIGURE 2. Motion densities of tiles in different frames.

III. ANALYSIS OF MOTION DENSITY OF TILES IN CONSECUTIVE FRAMES

Although collocated tiles in consecutive frames are visually similar, they may not have similar motion activity. It has been realized that collocated tiles in consecutive frames may have different values of motion density. This is mainly evident when motion information of blocks in consecutive frames are not highly correlated. Such this low correlation is realized when position of a fast moving object is different in two consecutive frames. In this case, the motion information of blocks in the same position of adjacent frames may not be consistent. As a result, motion density of collocated tile in the previous frame may not directly represent motion activity of collocated tile in the current frame.

To overcome the above-mentioned problem, motion activity can be measured for a larger area. This helps to compensate for the difference in motion activity that arises due to fast motion of small objects. Instead of calculating correlation between collocated tiles, the correlation between frames based on the motion density of tiles is considered. For this purpose, equation (1) can be modified, as follows:

$$\rho_{\mathbf{MD}}^{n} = \frac{\sum_{i=1}^{N_T} (\mathbf{MD}_{tile,i}^{n} - \mathbf{MD}_{tile}^{\prime n})}{\sqrt{\left[\sum_{i=1}^{N_T} (\mathbf{MD}_{tile,i}^{n} - \mathbf{MD}_{tile}^{\prime n})^2\right]}} \times \frac{\sum_{i=1}^{N_T} (\mathbf{MD}_{tile,i}^{(n-1)} - \mathbf{MD}_{tile}^{\prime (n-1)})}{\sqrt{\left[\sum_{i=1}^{N_T} (\mathbf{MD}_{tile,i}^{(n-1)} - \mathbf{MD}_{tile}^{\prime (n-1)})^2\right]}}, \tag{6}$$

where N_T is the total number of tiles in a frame. Similarly, $\mathbf{MD}_{tile,i}^{n}$ and $\mathbf{MD}_{tile,i}^{(n-1)}$ are the motion densities of i^{th} tile in n^{th} and $(n-1)^{th}$ frames, respectively.

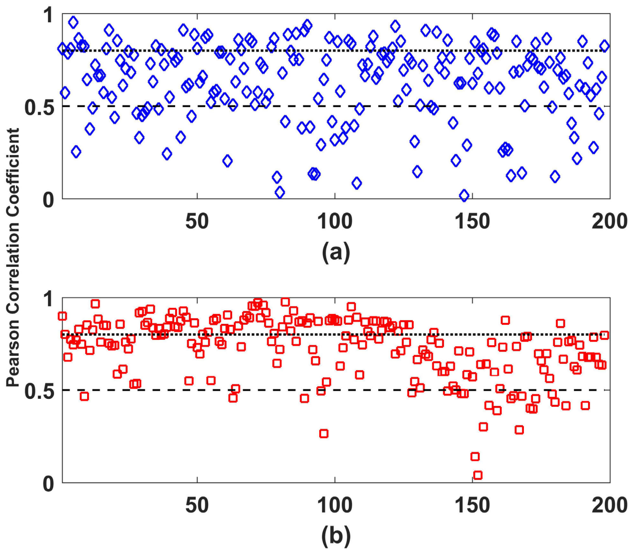


FIGURE 3. Correlation between frames based on motion density of tiles. (a) *Sunflower.yuv* (b) *Pedestrian.yuv.*

Figure 3 shows motion density based correlation of sunflower.yuv and pedestrian.yuv videos. The values are shown for 200 frames. In sunflower.yuv video, correlation coefficients greater than 0.5 and 0.8 are realized for a total of 47 and 149 frames, respectively. Similarly, in pedestrian.yuv video, 78 and 173 frames have correlations higher than 0.5 and 0.8, respectively. Considering such this high correlation between number of frames, it is concluded that the motion density of a tile positioned in a previous frame can be utilized to estimate motion activity of the collocated tile in the current frame. This leads to evaluate the importance of a tile in the entire bitstream on the basis of the motion density of tiles from its previous frame.

IV. PROPOSED TWO-LEVEL UEP APPROACH

In the previous section, a high correlation between consecutive frames based on the motion density of tiles was observed. This means that motion densities of tiles in the previous frame can be utilized for determining the importance of tiles in the current frame. This is accomplished by defining a difference between motion densities of considered tiles, as follows:

$$\Delta\mathbf{MD}_k^i = \mathbf{MD}_k^i - \mathbf{MD}_k^{(i-1)}, \quad 2 \le i \le N_f, \tag{7}$$

where N_f is the total number of video frames. $\mathbf{MD}_k^i$ and $\mathbf{MD}_k^{i-1}$ are the motion densities of k^{th} tiles in i^{th} and $(i-1)^{th}$ frames, respectively.

Let us consider the threshold for evaluating priority of tiles based on the average of motion densities of tiles in the same frame. Then, for k^{th} tile of $(i-1)^{th}$ frame, the threshold is represented by:

$$Th_k^{(i-1)} = \frac{1}{N_T}\sum_{k=0}^{N_T}\mathbf{MD}_k^{(i-1)}. \tag{8}$$

The difference between motion density of the k^{th} tile in $(i-1)^{th}$ frame and its threshold value is given by:

$$\mathbf{MD}_k^{(i-1)} - Th_k^{(i-1)}$$

$$= \mathbf{MD}_k^{(i-1)} - \frac{1}{N_T}\mathbf{MD}_k^{(i-1)} - \frac{1}{N_T}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{9}$$

$$= \frac{N_T-1}{N_T}\mathbf{MD}_k^{(i-1)} - \frac{1}{N_T}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{10}$$

Assume $\mathbf{MD}_k^{(i-1)} < Th_k^{(i-1)}$. Then,

$$\mathbf{MD}_k^{(i-1)} < \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{11}$$

From (7) and (11),

$$\Delta\mathbf{MD}_k^i < \mathbf{MD}_k^i - \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{12}$$

If $\Delta\mathbf{MD}_k^i \le 0$,

$$\mathbf{MD}_k^i < \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{13}$$

Thus, k^{th} tile in i^{th} frame $(\mathbf{T}_k^i)$ is transmitted as a low-important tile.

On the other hand, when $\Delta\mathbf{MD}_k^i > 0$, $\mathbf{T}_k^i$ is transmitted as a low-important tile if,

$$\mathbf{MD}_k^i \le \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{14}$$

Otherwise, $\mathbf{T}_k^i$ is transmitted as a high-important tile.

Alternatively, consider $\mathbf{MD}_k^{(i-1)} > Th_k^{(i-1)}$. Then,

$$\mathbf{MD}_k^{(i-1)} > \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{15}$$

From (7) and (15), it is understood that,

$$\Delta\mathbf{MD}_k^i > \mathbf{MD}_k^i - \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{16}$$

If $\Delta\mathbf{MD}_k^i \ge 0$,

$$\mathbf{MD}_k^i > \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{17}$$

Hence, $\mathbf{T}_k^i$ is represented as a high-important tile.

On the other hand, when $\Delta\mathbf{MD}_k^i < 0$, $\mathbf{T}_k^i$ is a high-important tile if,

$$\mathbf{MD}_k^i > \frac{1}{N_T-1}\sum_{j=1,j\neq k}^{N_T}\mathbf{MD}_j^{(i-1)}. \tag{18}$$

Otherwise, it is a low-important tile.

The common term appearing in equations (14) and (18) can be represented as the threshold value for determining the priority of k^{th} tile in the i^{th} frame. This is given by:

$$Th_k^i = \frac{1}{N_T - 1} \sum_{j=1, j \neq k}^{N_T} \mathbf{MD}_k^{i-1}, \quad i \geq 2. \tag{19}$$

To determine the priority of tiles in the first frame of the video, the average value of motion densities of tiles in the same frame is considered. If the value of motion density of a tile is higher than the average, it is considered as the high-important tile. Otherwise, the tile is a low-important one. For all other frames, $2 \leq i \leq N_f$, UEP method is implemented based on the following algorithm:

Algorithm 1

```
1) Set $i = 2$
2) Calculate $\Delta\mathbf{MD}_k^i$, $1 \leq k \leq N_T$.
3) If $\Delta\mathbf{MD}_k^i \geq 0$,
    a) $T_k^i$ is evaluated as a high-important tile, if $T_k^{i-1}$ is a high-important tile.
    b) If $T_k^{i-1}$ is a low-important tile, calculate $Th_k^i$.
    c) If $\mathbf{MD}_k^i > Th_k^i$, $T_k^i$ is evaluated as a high-important tile.
    d) If $\mathbf{MD}_k^i < Th_k^i$, $T_k^i$ is evaluated as a low-important tile.
4) If $\Delta\mathbf{MD}_k^i < 0$,
    a) $T_k^i$ is evaluated as a low-important tile, if $T_k^{i-1}$ is a low-important tile.
    b) If $T_k^{i-1}$ is a high-important tile, calculate $Th_k^i$.
    c) If $\mathbf{MD}_k^i > Th_k^i$, $T_k^i$ is evaluated as a high-important tile.
    d) If $\mathbf{MD}_k^i < Th_k^i$, $T_k^i$ is evaluated as a low-important tile.
5) If $i \neq N_f$, $i = i + 1$ and repeat steps 2 to 5. Otherwise, stop the algorithm.
```

In order to implement this motion density based UEP scheme, the HM encoder can be modified to calculate the motion density and prioritize tiles during the encoding process. Let M be the number of cores used in encoding of a video bitstream. The configuration for HEVC encoder is set to generate N_T tiles per frame. For simplicity of the analysis, it is assumed that all tiles have a similar computational complexity. Let t be the total processing time for each tile by a core, which includes calculation of the motion density and evaluation of the tile's priority. Then, the processing time required to prioritize a tile based on the motion density of tiles at the same frame is approximately given by:

$$T \approx \frac{N_T}{M} \times t. \tag{20}$$

When a tile's priority is evaluated based on the motion density of tiles from the previous frame, it can be prioritized without the knowledge of other tiles of the current frame. Therefore, the processing time of a tile is reduced to t. For one frame, the difference in encoding time is given by:

$$\Delta T = \frac{N_T}{M} \times t - t. \tag{21}$$

For F_n frames, the amount of time saved in the encoding process is given by:

$$\Delta T = \left[\frac{N_T}{M} - 1 \right] \times t \times F_n. \tag{22}$$

V. SIMULATION ANALYSIS

In this section, performance of the proposed UEP technique (UEP_NEW) is evaluated based on conducted simulations. The evaluation is done according to objective video quality measurement and visual comparison of uncompressed and received video frames.

A. SIMULATION SETUP

In all simulations, video sequences are compressed based on same configuration, including frame sizes, frame rate, specifications of tiles and number of frames. Two 1920×1080 video sequences including 200 frames, in *IBBBPBBBP...* format are encoded by HM reference software [11]. The encoding is done at the rate of 24 frames per second.

The results are compared with previous schemes (UEP_OLD) proposed in [10] and Equal Error Protection (EEP). Another technique of prioritizing of tiles based on the number of PUs (UEP_PU) is also considered. In [8], it has been mentioned that the number of PUs inside tiles can be used to determine the processing complexity of tiles. Hence, tiles are prioritized based on their processing complexity.

The error protection is done by a multi-rate (3730, 2238) quasi-cyclic low-density parity check (QC-LDPC) code [12], which is effective to combat bit errors occurred in the physical layer. Out of 2238 message bits, 746 bits are protected with the high priority at rate 0.5, while remaining 1492 bits are protected with the low priority at rate 0.67. In this case, the average of code rate is approximately 0.6. For EEP, a code with the rate of $\frac{2}{3}$ is formed based on the method described in [13]. To construct block of bits suitable for specifications of the channel code, zero padding is conducted for each tile. Furthermore, size of message bits in chosen error protection codes matches the size of tiles to significantly reduce the number of zero bits. In the provided simulations, the number of padded zero bits is maintained approximately at less than 2% of the overall number of transmitted bits. Codewords are modulated by binary phase-shift-keying (BPSK) and transmitted over additive white gaussian noise (AWGN) channel.

For the convenience of video quality assessment, peak-signal-to-noise-ratio (PSNR) is a most commonly used method [14]. There are other objective based methods including structural similarity index (SSIM) and video quality metric (VQM), but they are not used as frequently as PSNR [15]. Indeed, PSNR has the limitations of disregarding viewing conditions and characteristics of the human visual system. However, for a particular video content and similar encoding configurations, PSNR values provide reliable interpretations of the video quality [14], [15]. In order to illustrate subjective benefits of the technique, visual comparison of original and received video frames is provided. Therefore, this paper provides comparisons of different UEP techniques based on PSNR and visual comparison of video frames.

1) CONFIGURATION OF TILES CONSIDERED FOR SIMULATIONS

Due to partitioning flexibility, tiles can be partitioned in different ways for video transmission. In general, there are two different tile partitioning configurations, uniform and non-uniform. Non-uniform tile partitioning methods are aimed to improve the speed of HEVC encoding by balancing spatial load across different tiles [16], [17]. On the other hand, uniform methods have been equally effective in videos with balanced workload distribution. Uniform tile partitioning also avoids the complexity of determining optimal tile boundaries at each frames. In this paper, uniform tile partitioning is considered.

It is well known that increasing number of tiles improves encoding and decoding time. However, high number of tiles may affect coding efficiency and visual quality [16]. By contrast, lowering number of tiles may improve visual quality, but it may significantly increase the load imbalance between tiles and the encoding time. Table 1 presents PSNR values for video sequences encoded with different of tiles. These results indicate that the received video quality doesn't significantly vary, when number of tiles are kept at less than or equal to 5 × 5. In this paper, 4 × 4 and 5 × 5 tile partitionings are applied. Furthermore, in simulations, each tile is encapsulated into a slice with a distinct header. This is because at any bitrate, a portion of bits is used to carry header information of slices. As the inclusion of slice headers may result reduction of video quality at any given bitrate, in order to minimize the impact of slice headers, the simulations are conducted with higher bit-rate, 2000 kbps. In this case, headers only consume about 0.5% of total compressed video bits. It should also be considered that encapsulating tiles within slices helps to avoid transmission of extra bits to the decoder. These bits indicate row and column locations of tiles, loop filter control and bit-stream location information of all but first tile in each frame.

TABLE 1. PSNR (dB) values for different configuration of tiles.

Video	(kbps)	3 × 3	4 × 4	4 × 5	5 × 5	6 × 6
Sunflower	1000	42.27	42.21	42.07	42.02	41.71
	1500	43.31	43.22	43.13	43.01	42.68
	2000	44.12	44.04	43.99	43.89	43.62
	2500	44.81	44.75	44.69	44.64	44.31
	3000	45.04	44.98	44.92	44.86	44.52
Pedestrian	1000	38.88	38.84	38.81	38.77	38.50
	1500	40.23	40.18	40.11	40.02	39.71
	2000	41.40	41.33	41.27	41.16	40.87
	2500	41.99	41.94	41.88	41.82	41.44
	3000	42.38	42.32	42.29	42.25	41.91

B. SIMULATION RESULTS FOR TWO-LEVEL UEP TECHNIQUE

Figure 4 shows the rate-distortion performance of considered *Sunflower.yuv* and *Pedestrian.yuv* videos, when proposed UEP technique is applied. At $\frac{E_b}{N_0} = 3.2$ dB, it is noticeable that the performance of different configurations are close to one another. Although the difference in PSNR value is higher for the lower bitrates, they perform more or less the same at higher bitrate (>2000 kbps). This is because at lower bitrates the video quality is compromised by proportion of bits used to carry header information. At higher bitrates, the proportion of bits carrying header information is significantly reduced and the video quality is improved. Considering the achieved result, the simulations henceforth are conducted for 5 × 5 tiles per frame at 2000 kbps.

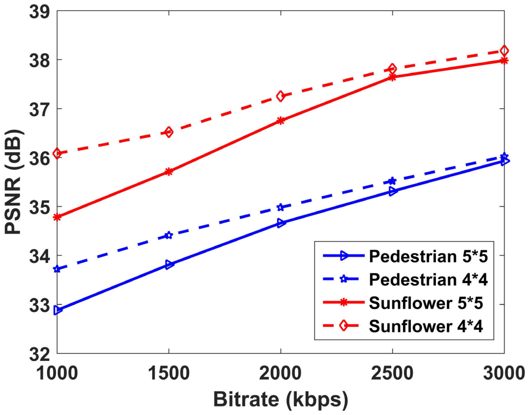
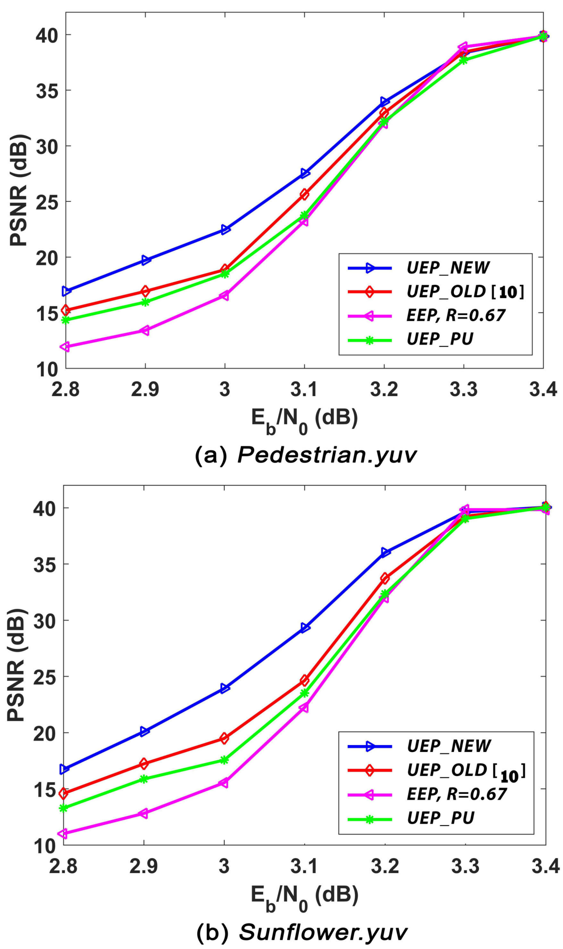


FIGURE 4. PSNR results for different tile partitioning configurations.

Figure 5 shows PSNR results of 1080p Sunflower.yuv and Pedestrian.yuv video sequences, respectively. Overall, the video performance provided by UEP schemes outperform EEP technique. This is because UEP schemes allocate higher protection to selected tiles in a video frame. For Sunflower.yuv video, at $\frac{E_b}{N_0} = 3.1$ dB, the performance of the video protected based on the new technique is improved by 3.9, 4.8 and 5.6 dBs compared to UEP_OLD, UEP_PU and EEP techniques, respectively. A similar result is obtained from pedestrian video, at $\frac{E_b}{N_0} = 3.1$ dB, the proposed method outperforms other UEP techniques by 7 dB.

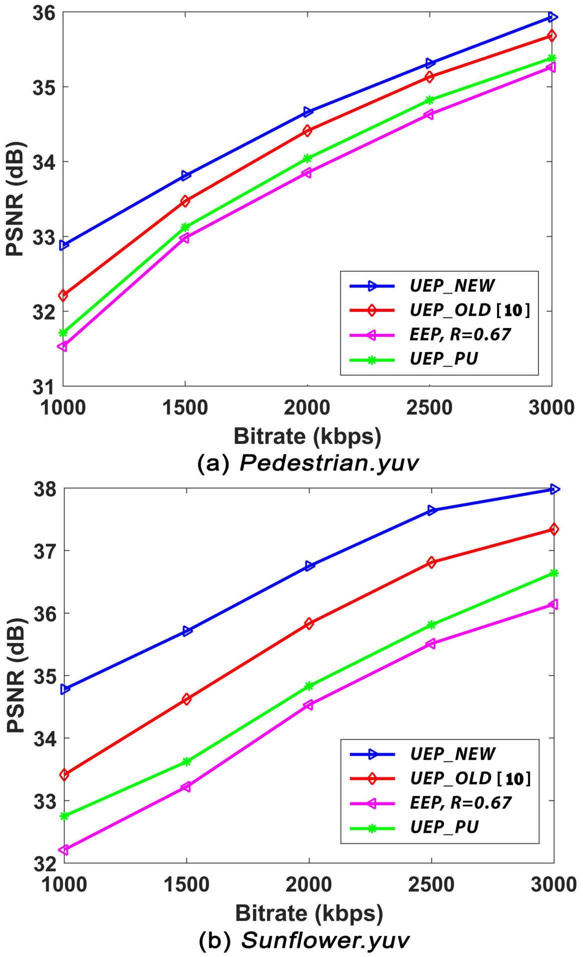
Figure 6 shows rate-distortion performance of the considered videos for UEP and EEP techniques. At $\frac{E_b}{N_0} = 3.2$ dB, the proposed UEP scheme shows the best performance. At a bitrate of 2000 kbps, it outperforms UEP_OLD by 0.92 and 0.25 dBs in *Sunflower.yuv* and *Pedestrian.yuv* videos, respectively. To demonstrate subjective comparison at the same bitrate, Figure 7 shows reconstructed frames of *Sunflower.yuv* and *Pedestrian.yuv* video sequences at



(a) *Pedestrian.yuv*

(b) *Sunflower.yuv*

FIGURE 5. PSNR results for videos at different $\frac{E_b}{N_0}$s.



(a) *Pedestrian.yuv*

(b) *Sunflower.yuv*

FIGURE 6. PSNR results for videos at different bitrates.

$\frac{E_b}{N_0}$ = 3.2 dB. The regions marked by blue and red ellipses indicate the degradation in visual quality compared to the encoded video frame. For 22^{nd} frame of *Sunflower.yuv* sequence, it is obvious that the proposed method provides the least degradation in the video frame. The same conclusion is achieved for the 43^{rd} frame of *Pedestrian.yuv* video sequence. These results confirm the subjective performance of applied techniques.

VI. PROPOSED THREE-LEVEL UEP METHOD

The three-level UEP scheme aims to increase the protection of those tiles, which are positioned in the neighbourhood of high-important tiles. As these neighbouring tiles may contain some parts of moving objects, their protection is necessary to improve overall video performance. Therefore, in order to identify tiles that contain moving objects, an object detection algorithm is presented, which is implemented in the compressed domain. The algorithm consists of two main steps, which are segmentation and clustering of CTUs based on their motion density values. In the first step, CTUs in the frame are classified into two groups based on the value of their motion density, which is calculated by equation (2). In the second step, density based spatial clustering is done to detect clusters of CTUs with the high motion density.

A. SEGMENTATION OF CTUs BASED ON MOTION DENSITY

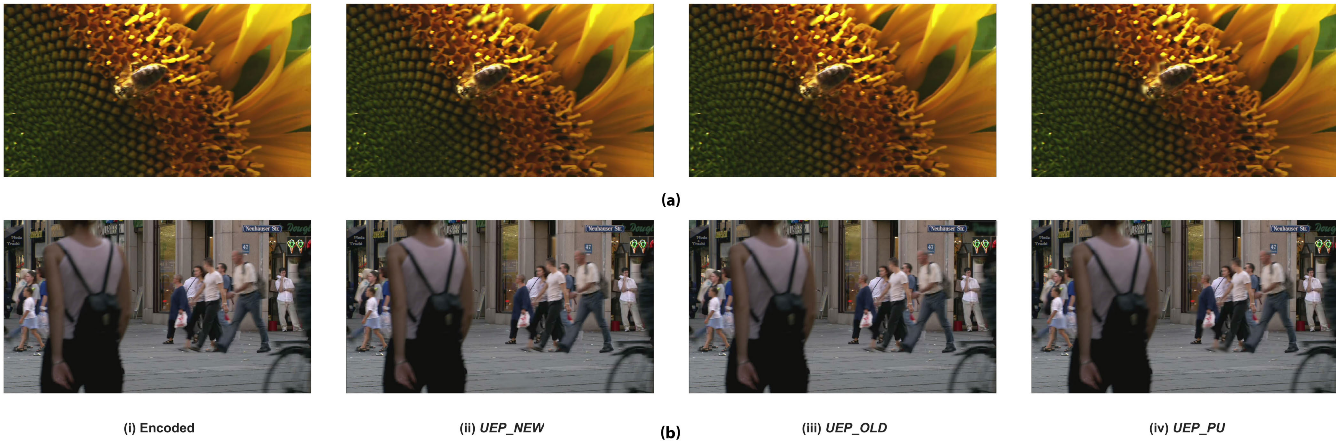
For segmentation of CTUs, kernel density estimation (KDE) based technique is applied to classify them into two groups, G^1_{CTU} and G^2_{CTU}. To perform KDE, a kernel function centered at motion density value is formed for every CTU, which is given by [18]:

$$K_f = \frac{1}{h} K\left[\frac{x - MD_{CTU}}{h}\right], \tag{23}$$

where $K[x]$ is a gaussian kernel function defined as: $K[x] = \frac{1}{\sqrt{2\pi}}(e^{-\frac{1}{2}x^2})$. In this equation, h is called the bandwidth. For a Gaussian kernel, it is given by:

$$h = 1.06 \times \sigma_{MD_{CTU}} \times N^{-\frac{1}{5}}, \tag{24}$$

where N is the number of CTUs inside a frame and $\sigma_{MD_{CTU}}$ is the standard deviation of motion densities of CTUs.



(a)

(i) Encoded (ii) *UEP_NEW* (b) (iii) *UEP_OLD* (iv) *UEP_PU*

FIGURE 7. (a) The 22nd frame of *Sunflower.yuv* sequence (b) The 43rd frame of *Pedestrian.yuv* sequence.

After determining K_fs, the probability density function (PDF) is estimated as follows:

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} K_f^i, \tag{25}$$

where K_f^i is the kernel function centered at motion density of i^{th} CTU.

The threshold for classifying CTUs is calculated as the first local minima of $f(x)$. This threshold is represented as β. If motion density of a CTU is higher than β, it is included in G^1_{CTU}. By contrast, CTUs, whose motion densities are lower than β, they belong to G^2_{CTU}.

To study the effectiveness of this technique, motion density of CTUs in different frames of Hall monitor.yuv video is evaluated. For 37th frame, it is realized that most of the CTUs have motion density in the range of $0-5$. By contrast, motion density of some CTUs are larger than 40. The threshold for classifying CTUs (β) is calculated as 7.6. This classification of CTUs is shown in Figure 9 (b). In the Figure, CTUs belonging to G^1_{CTU} are indicated in white, and other CTUs are represented in black.

B. SPATIAL CLUSTERING OF CTUs WITH HIGH MOTION DENSITY

This process applies density-based spatial clustering of applications with noise (DBSCAN) technique to identify and extract the clusters of CTUs belonging to G^1_{CTU} [19].

Let m be the number of CTUs in G^1_{CTU}. A bivariate data is then formed, which is a set of m data points, $\mathbf{X} = \{\mathbf{D}_i | i = 1 \cdots m\}$. Here, each data point ($\mathbf{D}_i$) is a two-dimensional vector ($\mathbf{D}_i^1, \mathbf{D}_i^2$), which represents x- and y- axis locations of the corresponding CTU.

Let $\mathbf{D}_p$ and $\mathbf{D}_q$, $1 \leq p \leq m$, $1 \leq q \leq m$, be two data points in $\mathbf{X}$. Then, the $\varepsilon-$ neighbourhood of data point $\mathbf{D}_p$ includes $\mathbf{D}_q$ if $d(\mathbf{D}_p, \mathbf{D}_q) \leq \varepsilon$. Here, ε is the radius of the neighbourhood region. Similarly, the parameter $d(\mathbf{D}_p, \mathbf{D}_q)$ is the euclidean distance between two data points $\mathbf{D}_p$ and $\mathbf{D}_q$. It is calculated as follows:

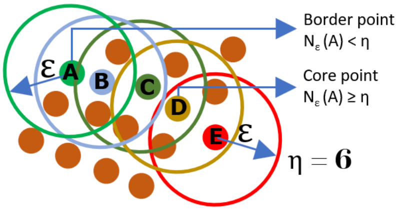


FIGURE 8. Density reachability and connectivity.

$$d(\mathbf{D}_p, \mathbf{D}_q) = \sqrt{(\mathbf{D}_p^1 - \mathbf{D}_q^1)^2 + (\mathbf{D}_p^2 - \mathbf{D}_q^2)^2}. \tag{26}$$

Thus, the $\varepsilon-$ neighbourhood of data point $\mathbf{D}_p$ is represented as follows:

$$N_\varepsilon(\mathbf{D}_p) = \{\mathbf{D}_q \in \mathbf{X} | d(\mathbf{D}_p, \mathbf{D}_q) \leq \varepsilon\}. \tag{27}$$

Let η be the minimum number of data points required to form a cluster. An arbitrary data point ($\mathbf{D}_a$), $\mathbf{D}_a \in \mathbf{X}$, is called direct density reachable from $\mathbf{D}_p$ when following conditions are satisfied:

1) D_a is a part of ε-neighbourhood of $\mathbf{D}_p$, $\mathbf{D}_a \in N_\varepsilon(\mathbf{D}_p)$.
2) The number of points in $N_\varepsilon(\mathbf{D}_p)$, $|N_\varepsilon(\mathbf{D}_p)|$, is greater than or equal to η. That is, $|N_\varepsilon(\mathbf{D}_p)| \geq \eta$.

In this case, $\mathbf{D}_p$ is called a core point. $\mathbf{D}_a$ is also called a core point if $|N_\varepsilon(\mathbf{D}_a)| \geq \eta$. Otherwise, it is considered a border point.

Another arbitrary point ($\mathbf{D}_b$), $1 \leq b \leq m$, $b \neq p$, $\mathbf{D}_b \in \mathbf{X}$, is called density reachable from $\mathbf{D}_p$ if there exists a sequence of points $\mathbf{D}_j = \{\mathbf{D}_1, \cdots, \mathbf{D}_n\}$ with $\mathbf{D}_1 = \mathbf{D}_p$ and $\mathbf{D}_n = \mathbf{D}_b$, such that each of $\mathbf{D}_j$s, $2 \leq j \leq n$, is directly density reachable from $\mathbf{D}_{j-1}$. Lastly, any two points ($\mathbf{D}_c$) and ($\mathbf{D}_d$), ($\mathbf{D}_c, \mathbf{D}_d$) $\in \mathbf{X}$, are called density-connected if there is a point $\mathbf{D}_p$ such that $\mathbf{D}_p$ and $\mathbf{D}_c$ as well as $\mathbf{D}_p$ and $\mathbf{D}_d$ are density reachable. Figure 8 provides an illustration of these concepts. For better understanding, five points are labelled and indicated in distinct colour and a circle of the same colour represents neighbourhood of each point. In the figure, point A is direct density reachable from point B. Similarly, points A and E are density reachable from point C through points B and D, respectively. Thus, they are all density connected. In this case, points B, C and D are called core points and points A and E are called border points.

Based on DBSCAN algorithm [19], a cluster of data points can be formed, when they satisfy the following properties:

1) All the data points must be mutually density-connected.
2) If a data point is density-connected to any data point of the cluster, it is part of the cluster as well.

The parameters ε and η influence the performance of DBSCAN algorithm. In different datasets, different values of the ε and η provide the optimal clustering results. Generally, η is determined based on the number of dimensions and presence of noise in the data set. As a rule of thumb, $\eta = 2 \times \textit{dimensions of data points}$, can be used [20]. However, larger values are usually preferred for data sets with noise to yield more significant clusters.

As previously stated, m data points represent the location of CTUs. Therefore, there is a limit to the maximum number of data points that can occur within a given neighbourhood. Let η_{max} be the maximum number of data points in a neighbourhood with radius ε. Table 2 shows different values of η_{max} for different values of ε. To form a cluster, minimum value of ε has to be 1, because the closest CTUs are at a distance of 1 unit from another. If ε is set at 1, the minimum number of points required to form the densest cluster becomes 5, as seen from the table.

TABLE 2. Maximum number of CTUs in a neighbourhood.

Neighbourhood sizes	η_{max}
$0 < \varepsilon < 1$	0
$1 \leq \varepsilon < \sqrt{2}$	5
$\sqrt{2} \leq \varepsilon < 2$	9
$2 \leq \varepsilon < \sqrt{5}$	13
$\sqrt{5} \leq \varepsilon < \sqrt{8}$	21

If $\eta = \eta_{max}$, the DBSCAN algorithm only searches for densest clusters, which may not always present in the data. Therefore, to choose an optimum value of η, neighbourhood density is defined as follows:

$$\rho_{nh} = \frac{\eta}{\eta_{max}}. \tag{28}$$

Let $\mathbf{Dist}_i = \{d(\mathbf{D}_i, \mathbf{D}_j)|j = \{1, 2, \cdots, m-1\}, j \neq i\}$, $1 \leq i \leq m$, be the distances from i^{th} data point to other $m-1$ data points sorted in ascending order. In this case, $\mathbf{A} = \{d(\mathbf{D}_i, \mathbf{D}_{\eta-1})|1 \leq i \leq m\}, \mathbf{A} \in \mathbf{Dist}_i$, is a set of distances from each data point to its $(\eta-1)^{th}$ nearest neighbour. The smallest of these distances (γ) is defined by:

$$\gamma = min(\mathbf{A}). \tag{29}$$

This γ represents radius of the densest neighbourhood that includes η data points. Based on values of γ and η, ρ_{nh} is calculated from equation (28).

If $\rho_{nh} \geq 0.5$, $\varepsilon = \gamma$. On the other hand, if $\rho_{nh} < 0.5$, η is increased by 1 and the process is repeated until a suitable combination of ε and η are obtained.

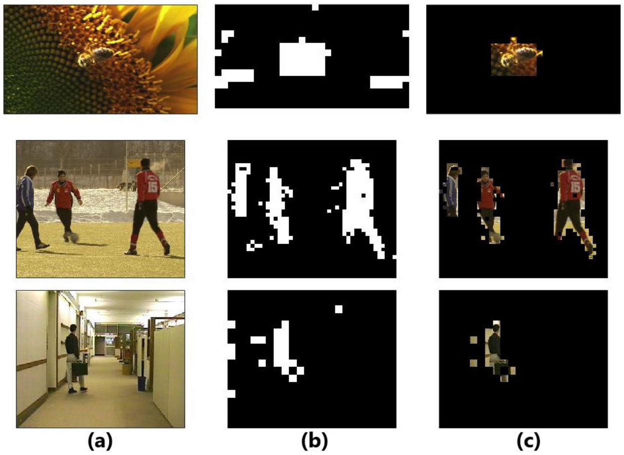


FIGURE 9. (a) 3rd frame of *Sunflower.yuv*. 19th frame of *Soccer.yuv*. 39th frame of *Hall monitor.yuv*. (b) MD based classification of CTUs (c) Detected regions.

Figure 9 shows the application of above-mentioned steps in three different videos. The three videos shown above are 1080p Sunflower.yuv, 4CIF Soccer.yuv and CIF Hall monitor.yuv. During encoding, the size of CTU is set as 64 × 64 pixels for 1080p Sunflower.yuv video. By contrast, the CTU size is set as 16 × 16 pixels for the low resolution 4CIF Soccer.yuv and CIF Hall monitor.yuv videos. In Figure 9 (b), two groups of CTUs can be noticed. CTUs, whose motion densities are higher than the threshold are indicated in white, whereas other CTUs are black. It is also realized that there may be CTUs with high motion densities even in stationary areas. Such CTUs can be interpreted as noise. To filter them, density based clustering algorithm was applied. Figure 9 (c) shows the cluster of CTUs formed based on DBSCAN technique, which represents the detected region. These regions are spread across tile boundaries. Thus, tiles that encapsulate the detected region are prioritized.

Based on the moving object detection technique mentioned in above, a three-level UEP of HEVC compressed frames can be formed. The method is summarized in the following algorithm, which is implemented for each frame.

VII. SIMULATION RESULTS FOR THREE-LEVEL PROTECTION

Simulations are performed to evaluate the performance of proposed three-level UEP technique. In this case, the error protection is achieved by a (4716, 3144) UEP QC-LDPC. Out of 3144 bits, 524, 1048, 1572 bits are transmitted as a high, low-high and low-low priority, respectively. The average code rate is maintained at 0.67 [12]. Other settings including

Algorithm 2

```
1) Evaluate tiles $T_l$, $1 \le l \le N_{tile}$, in a frame as low- or high-importance based on Algorithm 1.
2) Calculate motion density of all CTUs in a frame. ($\mathbf{MD}^k_{CTU}$, $1 \le k \le N$)
3) Perform segmentation of CTUs.
    a) Initialize two empty groups of CTUs. $G^1_{CTU} = \varnothing$, $G^2_{CTU} = \varnothing$.
    b) Calculate $f(x)$ from equation (25).
    c) Set: $\beta$ = first minima of $f(x)$.
    d) $\begin{cases} CTU^k \rightarrow G^1_{CTU} & \text{if } \mathbf{MD}^k_{CTU} > \beta \\ CTU^k \rightarrow G^2_{CTU} & \text{otherwise} \end{cases}$
4) Start DBSCAN algorithm.
    a) Set: $X = \{CTU^k \mid CTU^k \in G^1_{CTU}\}$.
       $m$ = length of $X$.
    b) Initialize an empty cluster $C$, $C = \varnothing$.
    c) Calculate optimal values of $\epsilon$ and $\eta$.
        i) Initialize: $\eta = 4$.
        ii) Obtain $\mathbf{Dist}_i$, $\mathbf{A}$ and $\gamma$.
        iii) Calculate $\rho_{nh}$.
        iv) If $\rho_{nh} < 0.5$, $\eta = \eta + 1$, go to step (ii).
        v) If $\rho_{nh} \ge 0.5$, $\varepsilon = \gamma$.
    d) For each of $CTU^k$s $\in X$,
        i) If $CTU^k$ belongs a core point or a border point, it belongs to the cluster. That is, $CTU^k \rightarrow C$.
5) If a tile ($T_l$) is evaluated as low-important,
    a) If $CTU^l \in C$, $1 \le l \le L_c$,[2] is inside $T_l$, it is evaluated as low-high important tile.
    b) If the tile doesn't include any $CTU^l \in C$, it is evaluated as low-low important tile.
```

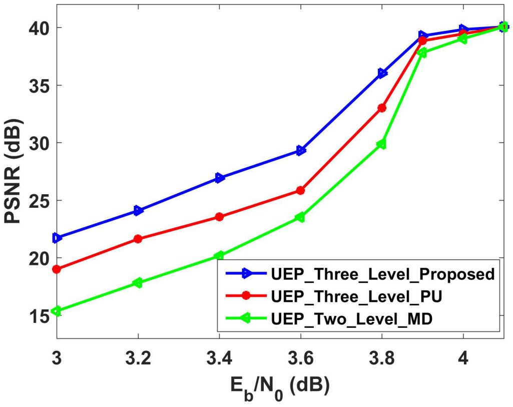


FIGURE 10. PSNR results for "Sunflower.yuv" video.

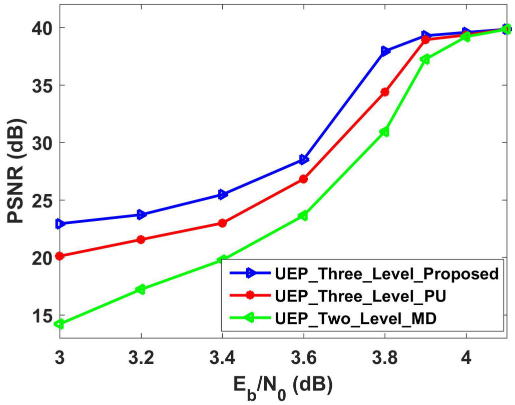


FIGURE 11. PSNR results for "Pedestrian.yuv" video.

modulation and channel type are kept the same as the previous simulations mentioned in section V.

The performance of proposed three level UEP technique is compared with the previously proposed two-level UEP method. For this purpose, the error protection for two-level UEP is achieved by (5222, 3730) UEP QC-LDPC code, which has a code rate of 0.71. Out of 3730 message bits, 1492 bits are transmitted with a high priority, whereas other 2238 bits are transmitted with a low priority.

Figures 10 and 11 show PSNR values for Sunflower.yuv and Pedestrian.yuv protected by two different UEP schemes in terms of $\frac{E_b}{N_0}$. UEP_Three_LEVEL_Proposed provides better performance than UEP_Three_LEVEL_CU implemented based on the numbers of CU partitioning inside tiles. It is also noticed that the peak value of PSNR is attained at an $\frac{E_b}{N_0} = 4$ dB, which is 0.6 dB higher than the previously proposed two level UEP. This is due to utilizing a higher code rate on the UEP QC-LDPC code.

VIII. CONCLUSION AND FUTURE WORK

In this paper, efficient UEP techniques were proposed for robust transmission of HEVC compressed videos. Analyses were conducted to verify the correlation between collocated tiles in consecutive frames. Based on the correlation existed between consecutive frames, a two-level UEP was proposed. Simulation results showed that the proposed two-level UEP scheme provides a significant improvement compared to EEP technique and other UEP methods. In addition, a three-level UEP technique was proposed to optimize the overall code rate. The three-level UEP was applied to improve the performance of tiles which had lower motion density, despite including part of a moving object. A low-complexity DBSCAN based object extraction method was implemented to identify such tiles. Results show that in terms of average PSNR, the three-level UEP method outperforms previous UEP scheme by 3.5 dB. Furthermore, improving performance of the tiles, which includes parts of a moving object, is expected to improve the subjective video quality as well.

As a future work, a detailed analysis and comparison of uniform and non-uniform tile partitioning schemes will be done to investigate the best tile partitioning configuration. Similarly, the research will focus on extending proposed scheme in application layer, which requires protection against packet/tile losses occurred during transmission. In addition, investigating a more effective technique for identifying moving objects in video frames will also be considered.

REFERENCES

[1] Y. Huo, C. Zhou, J. Jiang, and L. Hanzo, "Historical information aware unequal error protection of scalable HEVC/H.265 streaming over free space optical channels," *IEEE Access*, vol. 4, pp. 5659–5672, 2016.

[2] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 4, pp. 600–607, Oct. 1967.

[3] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 819–828, Jun. 2000.

[4] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the Internet using unequal error protection," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 121–131, Feb. 2003.

[5] Y. Chang, S. Lee, and R. Komyia, "A fast forward error correction allocation algorithm for unequal error protection of video transmission over wireless channels," *IEEE Trans. Consum. Electron.*, vol. 54, no. 3, pp. 1066–1073, Aug. 2008.

[6] Y. Wang and S. Sun, "An improved uep based on the importance of MacroBlock," in *Proc. 4th IEEE Int. Conf. Broadband Netw. Multimedia Technol.*, Oct. 2011, pp. 291–295.

[7] N. Nasruminallah and L. Hanzo, "Near-capacity H.264 multimedia communications using iterative joint source-channel decoding," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 538–564, 2nd Quart., 2012.

[8] E.-S. Ryu and S. Ryu, "Robust real-time UHD video streaming system using scalable high efficiency video coding," *Multimedia Tools Appl.*, vol. 76, no. 23, pp. 25511–25527, Dec. 2017, doi: 10.1007/s11042-017-4835-2.

[9] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, "An overview of tiles in HEVC," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 969–977, Dec. 2013.

[10] B. Paudel, S. Vafi, and P. Bhattarai, "An adaptive ROI based UEP scheme for HEVC compressed video bitstreams," in *Proc. IEEE 8th Global Conf. Consum. Electron. (GCCE)*, Oct. 2019, pp. 373–374.

[11] Fraunhofer Heinrich Hertz Institute. *High Efficiency Video Coding*. [Online]. Available: https://hevc.hhi.fraunhofer.de/

[12] B. Paudel and S. Vafi, "An unequal error protection of quasi-cyclic low density parity check (QC-LDPC) codes based on combinatorial designs," in *Proc. IEEE 4th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Feb. 2019, pp. 488–492.

[13] S. Vafi and N. Rezvani Majid, "A new scheme of high performance quasi-cyclic LDPC codes with girth 6," *IEEE Commun. Lett.*, vol. 19, no. 10, pp. 1666–1669, Oct. 2015.

[14] Q. Huynh-Thu and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," *Telecommun. Syst.*, vol. 49, no. 1, pp. 35–48, Jan. 2012.

[15] 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 Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 76–90, Jan. 2016.

[16] L. Dragić, I. Piljić, and M. Kovać, "Dynamic load balancing algorithm based on HEVC tiles for just-in-time video encoding for heterogeneous architectures," *Automatika*, vol. 60, no. 2, pp. 239–244, Apr. 2019, doi: 10.1080/00051144.2019.1618527.

[17] C. Blumenberg, D. Palomino, S. Bampi, and B. Zatt, "Adaptive content-based tile partitioning algorithm for the HEVC standard," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2013, pp. 185–188.

[18] L. C. Matioli, S. R. Santos, M. Kleina, and E. A. Leite, "A new algorithm for clustering based on kernel density estimation," *J. Appl. Statist.*, vol. 45, no. 2, pp. 347–366, Jan. 2018, doi: 10.1080/02664763.2016.1277191.

[19] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A new DBSCAN parameters determination method based on improved MVO," *IEEE Access*, vol. 7, pp. 104085–104095, 2019.

[20] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 169–194, 1998, doi: 10.1023/A:1009745219419.

BIJAYA PAUDEL (Member, IEEE) received the B.Eng. degree in electrical engineering from the Institute of Engineering, Tribhuvan University, Nepal, in 2013, and the M.Eng. degree from Charles Darwin University, Australia, in 2016. He is currently pursuing the Ph.D. degree with the College of Engineering, IT, and Environment, Charles Darwin University. His research interests include channel coding, robust wireless video transmission, machine learning, and video processing in compressed domain.

SINA VAFI (Senior Member, IEEE) received the Ph.D. degree from the University of Wollongong (UoW), Wollongong, Australia. He is currently a Senior Lecturer of electrical and electronics engineering with Charles Darwin University (CDU) and get-chartered of Engineers Australia (EA). He is the author of over 40 articles in algebraic coding theory and broadband wireless video transmission systems and currently leads number of research projects in the abovementioned fields. He is the General Chair of the 2021 Australian Communication Theory Workshop (2021 AusCTW). He was an invited Speaker in number of workshops, Technical Committee Member of international conferences, and a Reviewer of several scholarly journals.