Research Article

Optimized Sample Adaptive Offset Filter in HEVC

N. Archana ⓘ,[1] R. Menaka,[1] P. T. Vasanth Raj ⓘ,[2] G. Keerthivasan,[3] and N. Kumar ⓘ[4]

[1]*Department of Electronics and Communication Engineering, Chennai Institute of Technology, Chennai, India*
[2]*Center for System Design, Chennai Institute of Technology, Chennai, India*
[3]*Tata Consultancy Services, Chennai, India*
[4]*Jimma University, Jimma, Ethiopia*

Correspondence should be addressed to N. Kumar; kumar.nachimuthu@ju.edu.et

The high-efficiency video coding (HEVC) standard incorporates sample adaptive offset (SAO) as an in-loop filtering technique to reduce distortion. The fundamental tenet of SAO is to classify every reconstructed pixel and then simply add the offset to each group of pixels. The previous existing technique employs 32 bands in band offset (BO) calculation to determine the proper offset value to minimize the distortion in the SAO estimate algorithm for statistics collection and parametric determination procedure. The suggested method will reduce the necessary band count by 16 bands. It does not alter how edge offset is determined in any way. It offers enough and efficient room for creating superior band group predictions and selections, as well as an ideal decrease in the implemented area. Thus, reducing the number of bands to 16 also decreases the amount of storage needed to hold each pixel's information in binary files by up to 0.65%.

1. Introduction

The initial video processing and storing schemes use magnetic tapes and then it was submitted with compact discs, it is used to store the data in digital format. Due to the drastic changes in technology the necessity for lower storage required to store the data and the relatively reduced bandwidth required to transmit video also increased. To meet the need for reduction of storage space, the series of video coding standards are developed with regular updates over the previous standard used. One such video coding standard is the high-efficiency video coding (HEVC) standard otherwise known as H.265 is the successor of H.264 or commonly called AVC (Advanced Video Coding) [1].

H.264 [2] is the block-oriented motion compensation for processing each video frame. H.264 was introduced in the year of 2003. The process of motion compensation involves deriving the transformation between the current picture and the reference picture which may be considered from the earlier frame or from the next frame.

This video codec was widely used in high-definition DVDs such as Blu-Ray, many Apple products, and many Internet sources such as YouTube, US defense video applications, and so on. The process of compression in H.264 involves prediction, transformation, and encoding.

The encoder of the H.264 used to process the individual macroblock to form the prediction depends upon the previously processed data from the same frame or with the previous one to form a residual. The initial prediction process will find the difference between the current frame data with that of previously coded data to derive such residuals where the coded data may be well-thought-out from the present frame (intraframe prediction) or any frame which was by now encoded and transmitted (interframe prediction).

The mean for the block of residuals was derived for the further transformation process to define a set of coefficients. This process of deriving the mean residual will add the same coefficient value to all the macroblocks thus further processing may be more flexible than the previous standard but the output which is obtained was not so effective.

H.264 considers a macroblock of the maximum of 16×16 pixel sample for processing, whereas in H.265 it takes about 64×64 pixel sample for further processing. In

intraframe prediction it used to consider the motion in about 9 directions, to overcome these disadvantages the new video codec called H.265 was formulated.

H.265 [3] mainly aims to reduce the bit rate by half of the H.264 while maintaining more video quality than H.264. The HEVC standard is drafted by the JCT-VC (Joint Collaborative Team on Video Coding) which is formed and well-known by the ISO/IEC Moving Picture Experts Group formally known as MPEG and ITU-T Video Coding Experts Group (VCEG) in the year 2013 [4–6].

H.265 is used in various fields which requires reduced video traffic for transmission. It is widely used for many applications, including real-time conversational applications such as video chatting, telepresence systems and video conferencing and editing systems, terrestrial transmission systems, camcorders, video content acquisition, Internet and mobile network video, broadcast of high definition video(HD) TV signals over satellite, cable, online video streaming platforms and Blu-ray Discs.

Today, 1080P high definition (HD) and 4K ultra-HD are ruling the world, but the studies illustrate that the future will be dominated by the next-generation video specification called the 8K ultra-HD. The 8K is about to provide a high visual experience with a definition of 7680×4320 pixels per frame and 120 frames per second [7].

The basic HEVC video encoder works like this first, it divides each sample image into many units, then it predicts each unit using inter- or intra-unit prediction, and finally, it subtracts the prediction from the unit. The residual is quantized and changed. The discrepancy between the predicted value and the original image unit is known as a residual. The transform output, prediction information, mode information, and headers are all encoded using entropy.

One of the remarkable overcomes or changes to H.264 in H.265 is the introduction of the SAO (sample adaptive offset) filter and the introduction of CTU (coding tree units). HEVC uses a maximum of 64×64 macroblocks. Nowadays, the video business has rapidly developed while the demand for higher resolution and higher definition has continuously improved.

H.265 uses In-loop filtering is used to remove the artifacts and residuals caused during video decoding, which occurs in H.264. The process of removing such an artifact in H.264 was one of the most complicated processes. But the complications in removing artifacts were reduced by adding this In-loop filter. The basic In-loop filtering that has been introduced and adopted in video coding standards is the deblocking filter (DBF), which is essentially used to reduce blocking artifacts [8].

The H.265 has many advanced technologies in which the in-loop filtering consists of two stages, such as deblocking filtering and sample adaptive offset filtering, which are cascaded together as shown in Figure 1. Sample adaptive offset (SAO) is adapted in the HEVC standard as an in-loop filtering method to reduce distortion.

In Figure 1, the transform block (scaling and inverse transform), deblocking and SAO filters, intrapicture prediction, and motion compensation (comes along with motion prediction block) are decoder modeling [9].

A number of novel intracode approaches are used by the versatile video coding (VVC) standard [10]. Nonpromising modes are suggested to be removed using adaptive mode pruning (AMP). Using the ideal mode, mode-dependent termination (MDT) proposes to choose an acceptable model and eliminate pointless intrapredictions.

VVC is effectively enhanced by the quadtree with nested multi-type tree (QTMT) partition structure [11]. According to simulation findings, the suggested technique has a complexity reduction of up to 70% when compared to VVC reference software.

In this paper, detailed work on the SAO filter is explained in Section 2 which is the current model of the SAO filter in HEVC codec. The proposed work and models are given in Section 3. Detailed analysis on various parameters such as PSNR is done in Section 4. In Sections 5 and 6, the conclusion on the work and reference are presented, respectively.

2. SAO Filter

SAO filter improves the signal processing on the encoder sideways, which reduces computational complications on the decoder side. Random access (RA), high efficiency (HE), low complexity (LC), and low delay (LD) are the main benefits of using SAO filter H.265.

The actual process of the SAO filter is that it modifies the decoded samples by adding offsets conditionally to each sample after the application of the deblocking filter in which the SAO filter is found in the HEVC encoder.

Figure 2(a), shows a 16×16 block A having two edges, one with horizontal and another one with a vertical edge indicated with the blue color. Initially, 2-D DCT transform and quantization process are applied to this block A. Then inverse transform and inverse quantization are takes place to obtain the reconstructed output block A' as shown in Figure 2(b). Quantization errors were labeled with different colors: red for negative error, green for positive error, and purple for very positive error.

Ringing artifacts have been seen often appear along and close to edges. SAO is intended to eliminate such artifacts, and the results are shown in Figures 2(c)–2(e). Different colors are used to calibrate SAO's performance on block A' yellow denotes decreased quantization error, purple denotes an increase in error, and pink denotes no change in absolute error. The significance of the pixels that have the same blue and green hue as before the SAO procedure remains unchanged.

The results, which are shown in Figure 2(c), show that when the horizontal class is used over the reconstructed block A', the majority of errors with respect to the horizontal edges and some errors nearby also get removed. However, the results do not remove the ringing artifacts along the vertical edges. The vertical edges are improved in the next step of Figure 2(d), but the ringing artifacts associated with the diagonal edges are still present, as illustrated in Figures 2(e) and 2(f).

The 2 edges usually cause the pixel values to fluctuate. This example shows how each edge offset class may efficiently get rid of the artifacts related to the relevant edge directions and certain mistakes near those edges. A single-direction edge class cannot successfully eliminate artifacts in all other directions, as shown by this example.
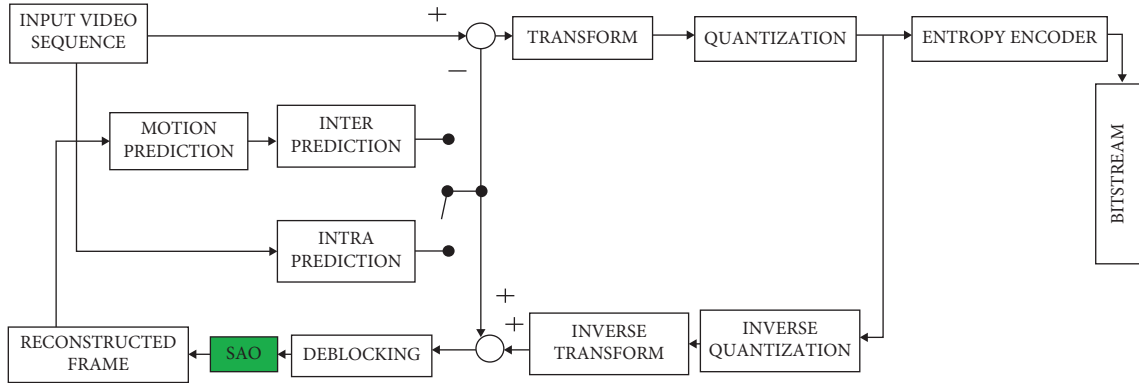
FIGURE 1: HEVC encoder block diagram.

Therefore, a method was introduced that provides combined edge classes which gives a new edge offset class, for this only one CTB was chosen depending upon rate-distortion performance from the 4 edge offset classes along with the new edge class, as a result of this the ringing artifacts reduced, as shown in Figure 2(g) and finally reached to zero. This shows the SAO process that improves the quality of video by improving the distortion that occurs due to artifacts.

In the early phases of its development, SAO will segment an image into LCU-aligned sections in order to collect local statistical data [12]. Figure 3 illustrates an example of the process of subdividing each image into LCU-aligned sections, where the dashed lines signify the borders of LCUs.

If it does not belong to either BO or EO then it is the region marked by "OF."

Either BO or EO may improve each area (Edge Offset). Splits the pixel intensity to 16 predefined bands and sends the band offsets to a decoder in H.264.

The quantity of side information and the no. of bands are trade-offs in HEVC. Each band's intensity interval shrinks as there are more bands present, drastically reducing the no. of pixels in the separate band and raising the nonzero offset, which might lessen distortion. First, the band offset divides the region's pixels into many bands, each of which has pixels with the same intensity range.

The range is uniformly split into 32 intervals, and each of those intervals includes an offset. The range starts at zero and goes all the way up to the greatest possible intensity value, which for 8-bit pixels is 255. When there are a large number of pixels in a band, especially in the center bands, the offset has a tendency to become zero. As a direct consequence of this, the 32 band offsets are separated into two distinct groups, namely, the middle 16 bands and the remaining 16 bands, as shown in Figure 4.

As a result of more investigation, it was revealed that it is sufficient to signal the decoder with only the starting band position and the offsets of four bands in sequence, as shown in Figure 5. As a consequence of this, it was decided that the no. of signaled offsets in BO should be compacted from 16 to 4, bringing it up to the same level as the no. of indicated offsets in EO [13].

SAO estimate procedure comprises 2 phases static collection and parameters determination phase as in Figure 6.

Each block is responsible for each phase which includes 2 phases of SAO.

In statistic collection [14], there are 4 classes for edge offset such as EO_0: horizontal; EO_1: vertical; EO_2: diagonal 135; EO_3: diagonal 45, and 5 types for each edge offset class of individual samples, this categorization depends upon the relationship of current sample c with that of the neighboring sample a and b as shown in Figure 7, where Table 1 shows the five categories.

The statistical data [15–17], shows that the widely held offset values for categories 1 and 2 are positive and categories 3 and 4 are negative. If the sample may not fall under any of these categories, then it is considered category zero and so the SAO process was not necessary to be applied.

The four offsets with respect to the selected direction are encoded for every region [18], this states that the EO trying to decrease the distance between the current and neighboring samples which helps to reduce the bit required to encode the sign bit of offset value.

Currently, the band offset has 32 bands. The 48 classes are referred to as 16 EO categories and 32 BO bands. Information count (C) and total (S) will be gathered for each categorization. Count refers to the quantity of samples in a CTB that fall within a certain categorization. The total of differences between the original and the reconstructed samples that fall within the designated categorization within one CTB is known as the sum. Later, bitmaps are used to accumulate statistics from 16 samples in 4×4 blocks at a time.

In parameter determination [14], for every classification, given data C and S, the initial procedure of parameter determination is toward obtaining 3 parameters, offset (Of), distortion (Dist), and cost (CO). Figure 8 shows the S and C generation bitmaps.

The distortion and cost are premeditated with the following equations:

$$\text{Dist} = Of * Of * C - Of * S * 2, \tag{1}$$

$$\text{CO} = D + \text{rate} * \text{lambda}. \tag{2}$$

Table 1: EO categories and conditions.

Category	Condition
1	c < a && c < b
2	(c < a && c == b) || (c == a && c < b)
3	(c > a && c == b) || (c == a && c > b)
4	c > a && c > b
0	None of the above

The band group with the lowest CO successes and their position is referred to as the band. Comparing the CO of the 4 EO classes, the BO, and the SAO that have not been applied, the minimum one is chosen as the kind of present CTB. The total of the Ds for each EO and BO is called *D*.

Figure 9 shows the total categories for EO and bands for BO after the offsets of respective bands are done. Thus, the existing system has 16 categories for EO and 32 bands for BO which are collectively called 48 classifications of offsets.

These offsets are used for the parameter determination where the parameter which is to be determinate causes better compression of a given video. Then the process is lead to band position determination.

Figure 10 shows the band position determination process. In-band position determination, CO plays a major role. CO of a single-band group is the summation of costs (CO) of 4 bands within the same band group. 32 band offsets are calculated and consecutive four bands are only taken into account for better band position determination [13]. The band group with the lowest cost will be chosen, and its first band will be referred to as band position after 29 COs of these band groups have been compared.

The next stage in the parametric determination phase is a type and EO class determination. The CO of 4 edge offset, band offset, and the raw class where SAO is not applied were compared, and the least value is processed as the type of present CTB. These features are shown in Figure 11. Each EO class's and BO's distortion (*D*) is the result of adding the Ds of 4 separate classifications. The *D* for an SAO that has not been useful is null or 0. With the aid of CABAC, the rates for each of the 4 EO classes, BO, and SAO not applied, remain determined.

Figure 11 shows the function of the parameter determination model. Therefore, it considers the four classifications from the edge offset, band position determination, and function obtained by not applying SAO which leads to the compression with better parameters.

The final stage in the parametric determination phase is mode determination. Figure 12 shows that the parameters of the upper CTB merge, and left CTB merge are compared with that of the current CTB merge and the superlative one is declared as the current CTB. Figure 13 shows that each CTU consists of CTBs with three different color components that include luma and chroma (Cr, Cb). One of the main and very important in this process is the criteria that have to be followed in the comparison is a transform of cost, hence name it to cost transformed (TCO).

$$TCO = TCO\_y + TCO\_c, \tag{3}$$

$$TCO\_y = \frac{Dis\_y}{Lu\_Y + Rt\_y}, \tag{4}$$

$$TCO\_c = \frac{(Dis\_cb + Dis\_cr)}{Lu\_c + Rt\_c}, \tag{5}$$

where Rt_*y* and Rt_*c* are rates of luma and chroma, which may obtain from CABAC, Dis_*y*, Dis_*cb*, and Dis_*cr* are a distortion of luma, *cb*, and the component of particular CTB. Lu_*Y* and/Lu_*c* is lambda of luma and chroma.

Figure 14 is an example to illustrate the SC process. SAcc and CAcc are the abbreviations for accumulators of sum and count. E.g., BO classifications are performed to the 2×2 reconstruction samples (0×93, 0×96, $0 \times 9b$, and 0×99).

Since all these samples belong to band 4, the differences (Org.-Rec.) belonging to band 4 are summed up [19]. The SAcc and the CAcc of band 4 add to −5 and 4, respectively.

3. Proposed System

In the proposed system, on the statistic collection phase before the sum and count determination, we introduce a block that will preselect the pixel intensity for a particular part of the sum and count generation defined for 16 band classifications for the band offset side.

Hence in the statistical collection (SC) phase, the sum and the count can be done by 16 band classification for band position determination (the existing system uses 32 bands for every sample block).

The process of SAO takes place normally from the initial stage of calculating the sum and the count value of the current data with that of the reference and derives different categories of offsets depending upon this sum and count that is obtained for each EO class, BO, and the sample without applying SAO. By finding the sum and counting the process of the statistical collection gets terminated.

The parametric determination (PD) phase followed by the statistical collection phase is used for predicting proper offset (EO, BO, and no offset) and respective numerical value for that offset that has to be along with that of the encoded data, from this we can understand that parametric determination phase plays a complicated process in SAO filtering process.

Many criteria increase the complexity of this parametric determination phase, one of such is to process 32 band classifications for the band position determination stage in the PD phase. In the proposed system, a block has an algorithm to separate the band into three subregions of bands which contains 16 band values in each region. Based on the pixel intensity of a sample the preselector will determine one of the sub-band regions with 16 band values. Instead of processing 32 band classifications, these 16 band classifications are more than enough to calculate the band offset.

The complications associated with processing 16 band classifications were two times less than the complication involved in processing in the existing system that requires 32
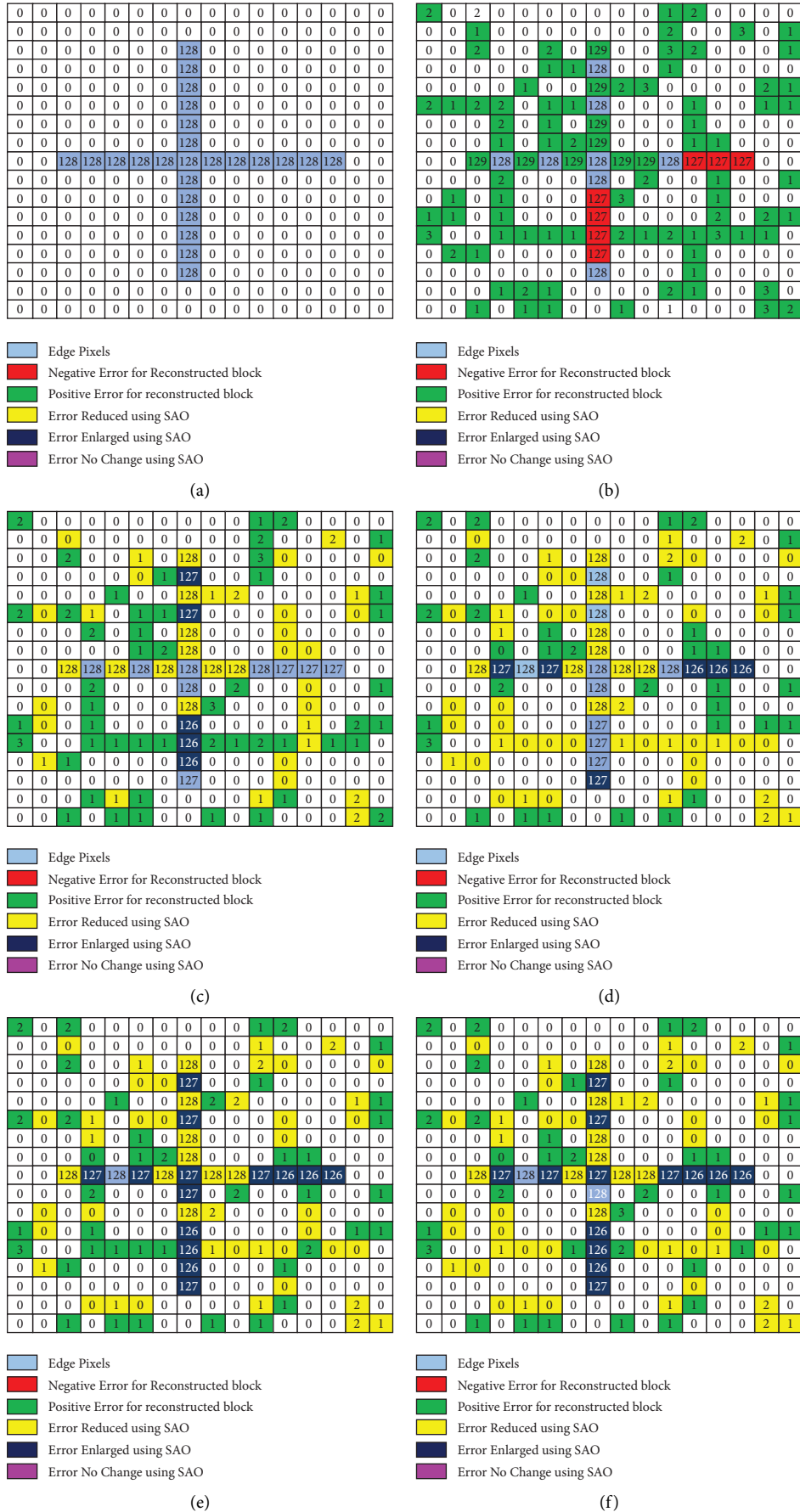
Edge Pixels
Negative Error for Reconstructed block
Positive Error for reconstructed block
Error Reduced using SAO
Error Enlarged using SAO
Error No Change using SAO

(a)

Edge Pixels
Negative Error for Reconstructed block
Positive Error for reconstructed block
Error Reduced using SAO
Error Enlarged using SAO
Error No Change using SAO

(b)

Edge Pixels
Negative Error for Reconstructed block
Positive Error for reconstructed block
Error Reduced using SAO
Error Enlarged using SAO
Error No Change using SAO

(c)

Edge Pixels
Negative Error for Reconstructed block
Positive Error for reconstructed block
Error Reduced using SAO
Error Enlarged using SAO
Error No Change using SAO

(d)

Edge Pixels
Negative Error for Reconstructed block
Positive Error for reconstructed block
Error Reduced using SAO
Error Enlarged using SAO
Error No Change using SAO

(e)

Edge Pixels
Negative Error for Reconstructed block
Positive Error for reconstructed block
Error Reduced using SAO
Error Enlarged using SAO
Error No Change using SAO

(f)

Figure 2: Continued.

2	0	2	0	0	0	0	0	0	0	0	1	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	1
0	0	1	0	0	1	0	127	0	0	2	0	0	0	0	0	0
0	0	0	0	0	0	0	127	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	127	0	0	0	0	0	0	1	1	
2	0	0	0	0	0	0	127	0	0	0	0	0	0	0	1	
0	0	0	0	0	0	0	127	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	127	0	0	0	0	0	0	0	0	
0	0	127	127	127	127	127	127	127	127	128	126	126	126	0	0	
0	0	0	0	0	0	0	128	0	0	0	0	0	0	0	1	
0	0	0	0	0	0	0	128	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	126	0	0	0	0	0	0	0	1	
3	0	0	1	0	0	0	126	0	0	1	0	1	0	0	0	
0	1	0	0	0	0	0	126	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	127	0	0	0	0	0	0	0	0	
0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	2	0
0	0	1	0	1	1	0	0	1	0	1	0	0	0	2	1	

- ▢ Edge Pixels
- ▮ Negative Error for Reconstructed block
- ▮ Positive Error for reconstructed block
- ▮ Error Reduced using SAO
- ▮ Error Enlarged using SAO
- ▮ Error No Change using SAO

(g)

Figure 2: (a) Test block (16 × 16, size), (b) reconstructed block, (c–f) reconstructed block with edge offset class 0, 1, 2, and 3, and (g) reconstructed block with mutual edge offset class (0 and 1).
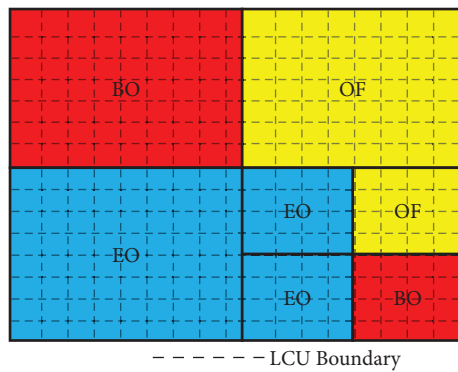
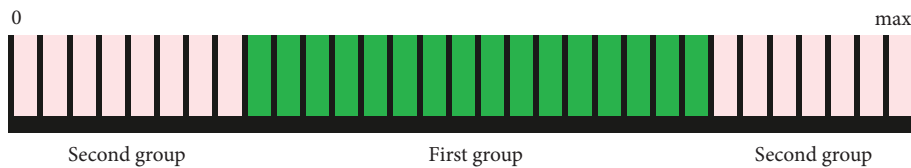Figure 3: Sample of subdividing a picture to LCU-aligned regions.

Figure 4: Classification of band offsets.

band classification to calculate appropriate offsets with the same encoding time without affecting the actual video quality.

Therefore, it reduces the hardware architecture from 48 classifications that include EO and BO to 32 classifications for sum and count generation Figure 15.

The proposed block consists of an algorithm that will sample the input sample block based on the grayscale values. Therefore, it acts like a preselector.

The preselector consists of three band values in common: Band Region I (0–15 band), Band Region II (8–23 band), and Band Region III (24–31 band) as shown in Figure 16.

The band regions are divided in a manner in which mostly the offsets are high for bands between 8 and 23. So, we divided it as a separate region and represented it as Band Region II, while the other bands are represented as Band Region I and Band Region II with every 16 bands. This shows that the block we introduced before the parametric

determination phase will give more priority to Band Region II where a higher number of band values falls in.

The preselector is an algorithm that is added to the existing system to find the appropriate offsets by choosing the relevant band region that directs each pixel to 16 band classifications. Each sample will have 16 individual band values defining the intensity of that pixel.

So, in the existing system, the sum and count for band offset are calculated for all 32 bands. Hence with the proposed system, we can do the calculation of band offset with 16 bands sum and count without any change in the BD rate and minor improvement in the bit rate. Though it produces some markable changes in the bit rate, the output compressed video that we obtained maintains the quality as same as the existing system. Figure 17 shows the modified block diagram.

In the band position determination, the most important process in the parametric determination phase, for the proposed system takes only 12 bands which are 28 bands in the existing system. The total band that has been selected and processed in band position determination improves the detection of active details in the sample.

Therefore, the accumulator for sum, count, and the consecutive band values for band offset is notably optimized. Figure 18 shows the band position determination for the proposed system.

4. Parameter Analysis

The experimental results are obtained and compared with HM 16.20 + SCM 8.8 which is the latest HM reference software release by the JCT-VC [20] under common test conditions [21, 22].

Two test conditions Lossy and mathematical lossless are defined, Lossy compression in which compressed content may not be numerically equivalent to that of the uncompressed data. Mathematically lossless in which decoded data is numerically similar to the uncompressed data.

Three different coding configurations or constraint conditions should be followed for each sample given by all intra (AI), low delay (LD), and random access (RA). AI is the condition with all pictures are coded as intrapictures. LD makes its first frame as an intraframe, with no backward reference frame for intraprediction (biprediction can be applied, but it may be possible only without reordering the frames).

For RA the intraframe may be taken in the random frame for a particular interval of time (Example: I frame may be taken as 16, 32, and 64 frames for every 20 fps, 30 fps, and 60 fps, respectively).

JCT-VC defines that H.265 allows both the RGB and YCbCr color formats which can be processed with the combinations of the above test conditions and coding configurations. Internal bit depth is chosen as the same value as input bit depth.

All the processed results can be reported with help of an Excel spreadsheet that includes file size, average bit rate, and encode and decode time. For each and every bitstream, the total number of bits used in each coding frame can also be reported separately.

The Bjøntegaard model was a popular evaluating tool to define coding efficiency which was used here to calculate average bit-rate saving [23, 24].

We test the Y-PSNR of the proposed system and HM 16.20 + SCM 8.8 on BasketballPass_416 × 240_50 used QPs are 31,33,39,40 and plotted as a graph in Figure 19.

HM 16.20 + SCM 8.8 software package is used to generate various configurations as

Encoder_intra_main.cfg

Encoder_lowdelay_main.cfg

Encoder_lowdelay_P_main.cfg

Encoder_randomaccess_main.cfg

Encoder_intra_main.cfg is a configuration file for AI mode, encode_lowdelay_main.cfg is a configuration file for LDB mode, encode_lowdelay_P_main.cfg is a configuration file for LDP mode, and encoder_randomaccess_main.cfg is a configuration file for RA mode.

Therefore, the results are obtained in each of the modes and stated in tables as follows with detailed discussion on the table.

From Figure 19, we can conclude that at nominal bit rates, the quality of the video increases by the proposed system than the HM 16.20 + SCM 8.8.

Table 2 gives the Bitrate by HM 16.20 + SCM 8.8 for input files from class C (BasketballDrill) and class D (BasketballPass) on four configurations such as AI, RA, LDB, and low delay P (LDP).

It also gives the bitrate by the proposed system for input files from class C (BasketballDrill) and class D (BasketballPass) on four configurations such as AI, RA, LDB, and LDP.

From Table 2, we can observe that the bit rate is reduced for four configurations such as AI, RA, LDB, and LDP for both the sample files from class C and class D.

The encoded BIN file size after encoding by the HM 16.20 + SCM 8.8 on the given sample files from class C and class D are tabulated in Table 3. The file size in the brackets is the actual bytes value of the respective file.

For example, from BasketballDrill the configuration of AI is taken. The BIN file size is approximately 9332 kB and its actual bytes size is 9,555,682 which is mentioned in the brackets.

Similarly, the encoded BIN file size after encoding by the proposed system on the given sample files from class C and class D are tabulated in Table 3. The file size in the brackets is the actual bytes value of the respective file.

For example, from BasketballPass in the configuration of LDP is here taken. The BIN file size is approximately 511 kB and its actual bytes size is 522,640 which is mentioned in the brackets.

From Table 3, the encoded BIN file size of the proposed system is reduced. For example, the BIN file size of BasketballDrill by HM 16.20 + SCM 8.8 in AI configuration is about 9332 kB (9,555,682 bytes) but the BIN file size of the same sample file BasketballDrill by the proposed system is
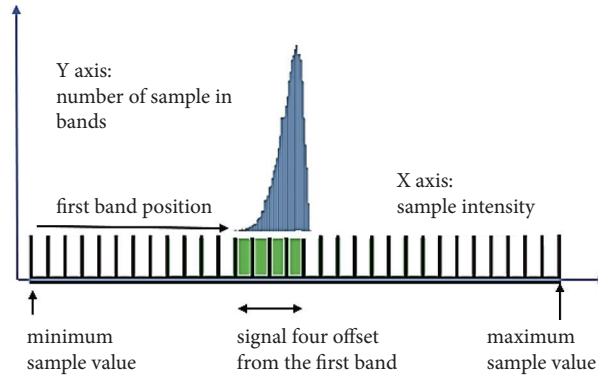
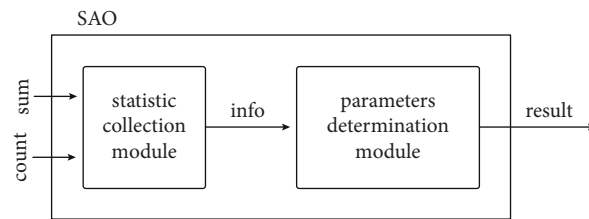FIGURE 5: Sample distribution in a CTB, with 4 consecutive bands.

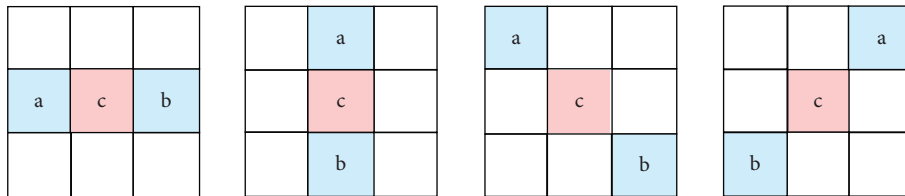FIGURE 6: SAO estimate block diagram.
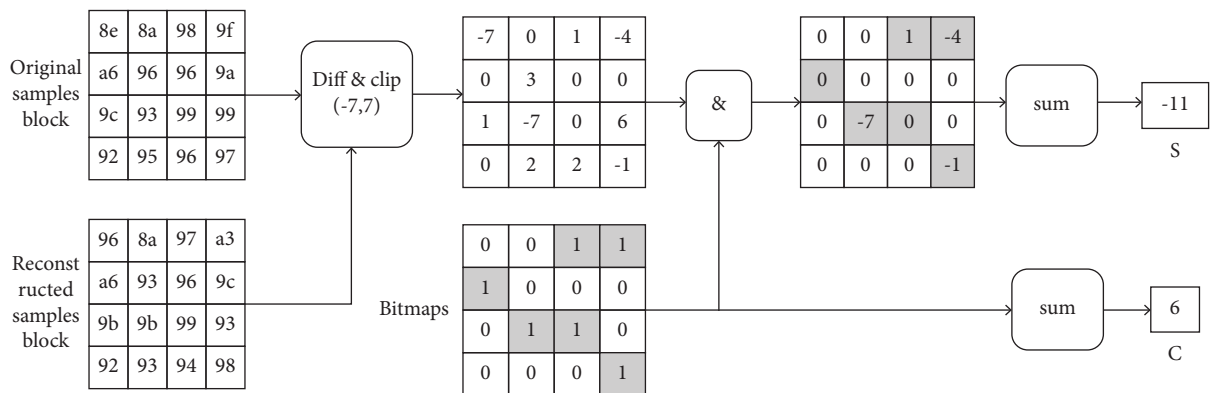

FIGURE 7: Types of EO classification 1-D patterns.

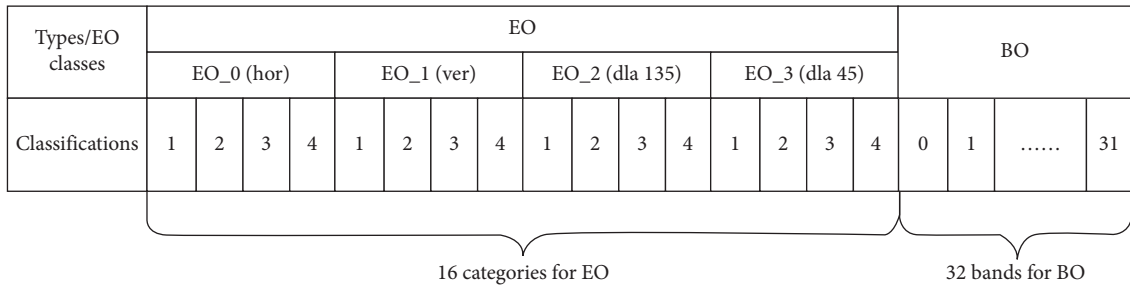
FIGURE 8: Example for S and C generation complete bitmaps.

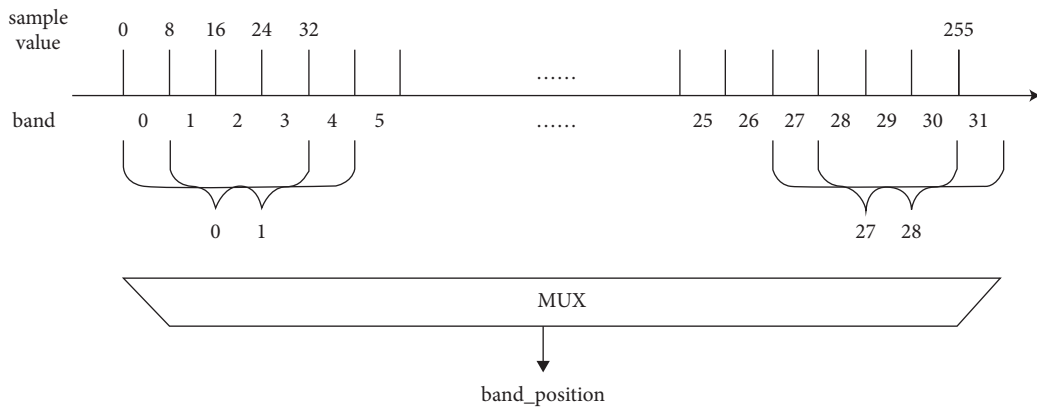| Types/EO classes	EO																BO			
	EO_0 (hor)				EO_1 (ver)				EO_2 (dla 135)				EO_3 (dla 45)							
Classifications	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	0	1	……	31

16 categories for EO   32 bands for BO

FIGURE 9: EO and BO bands categories.

FIGURE 10: The band position determination.
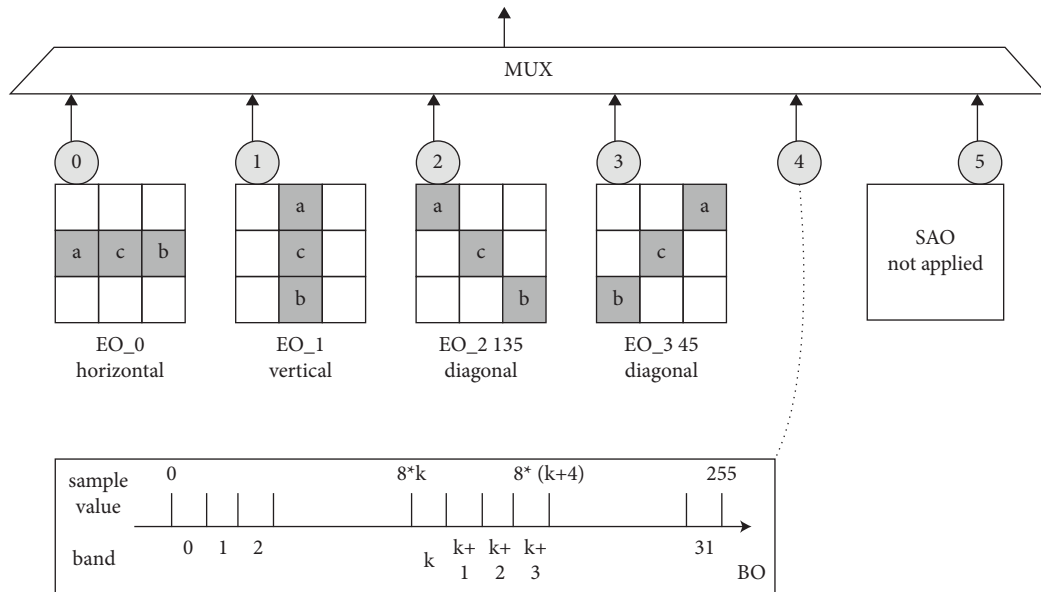
FIGURE 11: Types of determination.

the same. AI configuration is about 9271 kB (9,493,050 bytes) which is reduced by 0.65% in the Proposed System.

Table 4 illustrates the degradation of both HM 16.20 + SCM 8.8 vs. the proposed system. From the result, it is observed that the proposed system outperformance the existing system. Table 4, shows the experimental findings demonstrate that the suggested strategy may reduce parameter estimation time by an average of 75% while
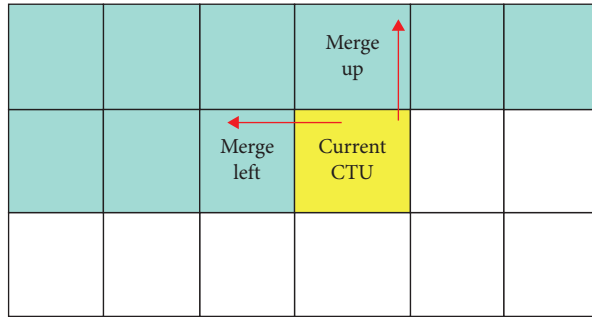
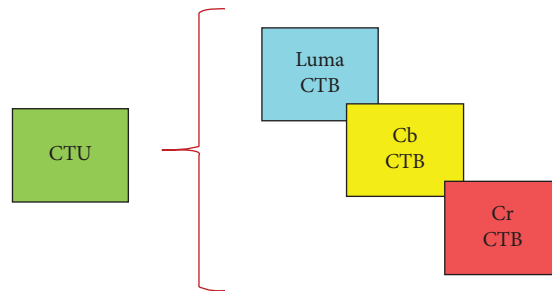Figure 12: The current CTU can reuse the SAO parameter of the left or above CTU.

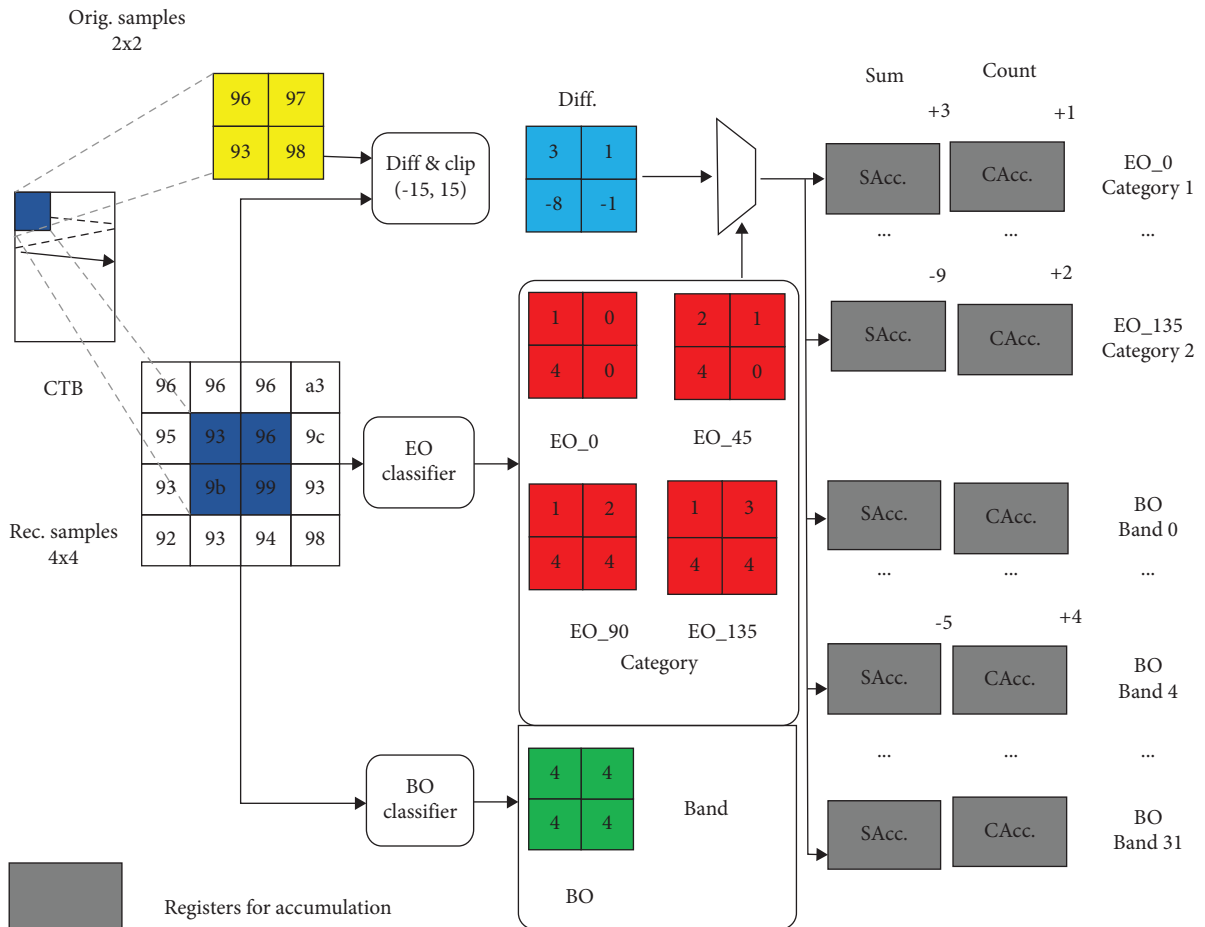
Figure 13: CTBs of three color components in CTU.

Figure 14: SC process.

Types/EO classes	EO																BO			
	EO_0 (hor)				EO_1 (ver)				EO_2 (dia 135)				EO_3 (dia 45)							
Classificat -ions	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	0	1	...	15

16 categories for EO   16 bands BO

FIGURE 15: The categories for EO and bands for BO of the proposed system.

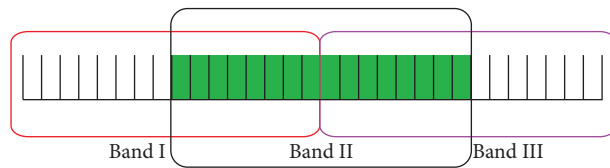Band I   Band II   Band III
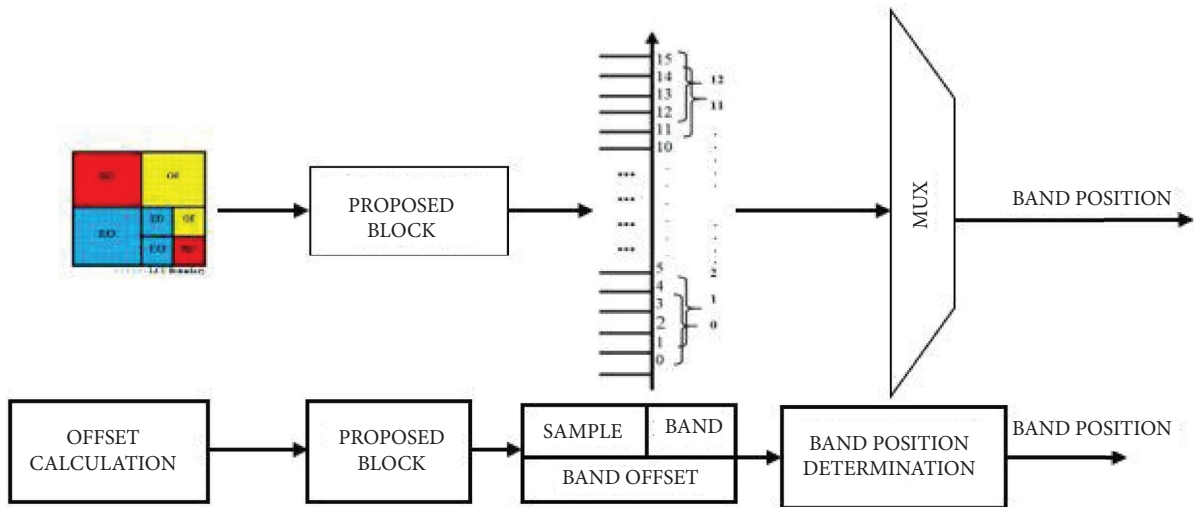
FIGURE 16: Proposed system band offset regions.

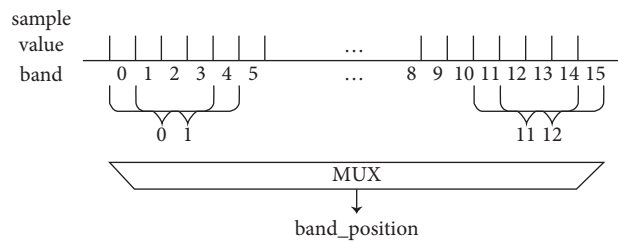FIGURE 17: Modified block diagram.

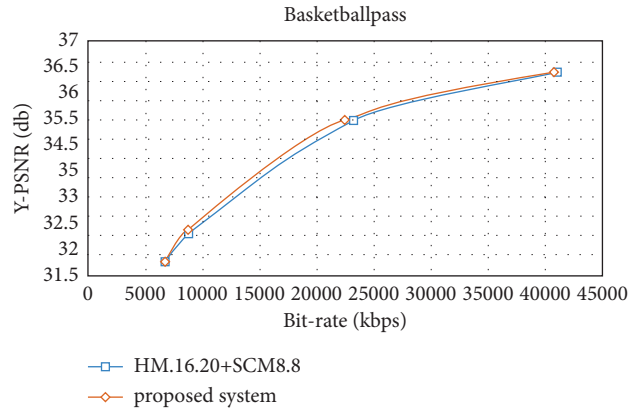FIGURE 18: The band position determination for the proposed system.

FIGURE 19: Y-PSNR of HM 16.20 + SCM 8.8 and proposed system for BasketballPass_416 × 240_50.

TABLE 2: Bitrate: HM 16.20 + SCM 8.8 vs. proposed system.

		Bitrate (in kbps)			
Test set		AI	RA	LDB	LDP
Class C WVGA	BasketballDrill	7644.54	936.946	869.402	916.835
Class D WQVGA	BasketballPass	1847.31	362.878	408.93	418.378
Proposed system					
Class C WVGA	BasketballDrill	7594.44	936.946	866.706	914.652
Class D WQVGA	BasketballPass	1843.87	362.504	408.522	418.112

HM 16.20 + SCM 8.8

TABLE 3: BIN file size: HM 16.20 + SCM 8.8 vs. proposed system.

		BIN file size (in kB/in bytes)			
Test set		AI	RA	LDB	LDP
Class C WVGA	BasketballDrill	9332 (9,555,682)	1144 (1,171,183)	1062 (1,086,753)	1120 (1,146,044)
Class D WQVGA	BasketballPass	2256 (2,309,146)	443(453,598)	500 (511,163)	511 (522,973)
Proposed system					
Class C WVGA	BasketballDrill	9271 (9,493,050)	1142 (1,168,844)	1058 (1,083,382)	1117 (1,143,316)
Class D WQVGA	BasketballPass	2251 (2,304,838)	443(453,130)	499 (510,652)	511 (522,640)

HM 16.20 + SCM 8.8

degrading coding efficiency by 0.5%, 0.5%, 0.9%, and 1.2% for AI, RA, LDB, and LDP, respectively, and by 0.8% on average.

Table 5 shows the experimental findings demonstrate that the suggested strategy may reduce parameter estimation time by an average of 50% while degrading coding efficiency by 0.18%, 0.38%, 0.5%, and 0.94% for AI, RA, LDB, and LDP.
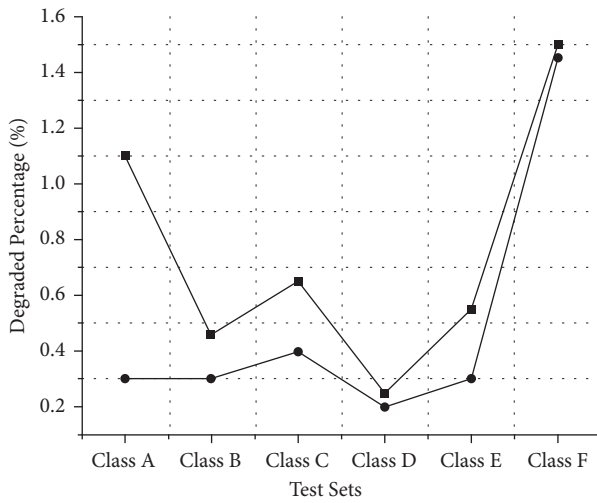
Figure 20 shows the comparison of the HM 16.20 + SCM 8.8 vs. the proposed system with different parameters.

TABLE 4: BD degradation: HM 16.20 + SCM 8.8 vs. proposed system.

| Test set		AI	RA	LDB	LDP
HM 16.20 + SCM 8.8					
Class A (4K × 2K)	Traffic	1.2	1.3	—	—
	People on street	1.3	2.2	—	—
Class B (1080P)	Cactus	0.6	2.5	2.7	10.2
	Park scene	0.6	0.9	1.6	9
	Basketball drive	0.3	1.5	1.3	8
Class C (WVGA)	Party scene	0.3	−0.2	0.7	4.3
	BasketballDrill	1.2	1.6	2.7	6.2
Class D (WQVGA)	Blowing bubbles	0.3	−0.6	0	2.8
	BasketballPass	0.5	0.8	1.3	4.9
Class E (720P)	Kristena and Sara	0.8	—	2.4	11
	Johnny	0.5	—	1.8	13.4
	Four people	0.8	—	2.5	8.9
Class F	Slide show	1.7	1.8	6.1	12.5
	BasketballDrill text	1.4	3	3.7	22.5
Proposed system					
Class A (4K × 2K)	Traffic	0.1	0	—	—
	People on street	0.2	0.3	—	—
Class B (1080P)	Cactus	0.2	0.2	0.4	2.1
	Park scene	0.2	0.1	0.4	1.2
	Basketball drive	0.3	0.4	0.3	1.5
Class C (WVGA)	Party scene	0.2	−0.4	0.5	1
	BasketballDrill	0.3	0.4	0.7	1.6
Class D (WQVGA)	Blowing bubbles	0.1	−0.1	0.1	0.6
	BasketballPass	0.1	0.2	0.7	0.9
Class E (720P)	Kristen and Sara	0.2	—	0.1	0.4
	Johnny	0.1	—	0.1	0.7
	Four people	0.2	—	0.1	0.3
Class F	Slide show	2.1	1	1.4	0.3
	BasketballDrill text	0.4	1.3	2.1	1
	Test set	AI	RA	LDB	LDP
Summary	**HM 16.20 + SCM 8.8**				
	Class A	1.1	1.65	—	—
	Class B	0.46	1.56	1.8	9.5
	Class C	0.65	0.8	1.7	5.2
	Class D	0.25	−0.05	0.55	3.7
	Class E	0.55	—	2.15	11
	Class F	1.5	1.95	5.1	17.85
	Proposed system				
	Class A	0.3	0.3	—	—
	Class B	0.3	0.35	0.75	1.7
	Class C	0.4	0.05	0.85	1.6
	Class D	0.2	0.15	0.6	1.05
	Class E	0.3	—	0.15	0.7
	Class F	1.45	1.4	2.05	0.9

TABLE 5: BD degradation: HM 16.20 + SCM 8.8 vs. proposed system.

| Test set | Y BD-rate degradation (%) $S = 8$ frames | | | | | | | |
| | AI | | RA | | LDB | | LDP | |
	HM 16.20 + SCM 8.8	Proposed	HM 16.20 + SCM 8.8	Proposed	HM 16.20 + SCM 8.8	Proposed	HM 16.20 + SCM 8.8	Proposed
Class A	0.6	0.1	2.5	0.3	—	—	—	—
Class B	0.5	0.1	2.3	0.3	2.5	0.6	11.5	1.3
Class C	0.7	0.1	1.2	0.5	2.1	0.5	7.5	1.1
Class D	0.4	0.1	0.5	0.2	0.9	0.4	4.7	0.7
Class E	0.6	0.2	—	—	2.1	0.2	11.5	0.3
Class F	1.3	0.5	2.8	0.6	5.5	0.8	12.5	1.3
Avg	0.68	0.18	1.86	0.38	2.62	0.5	9.54	0.94
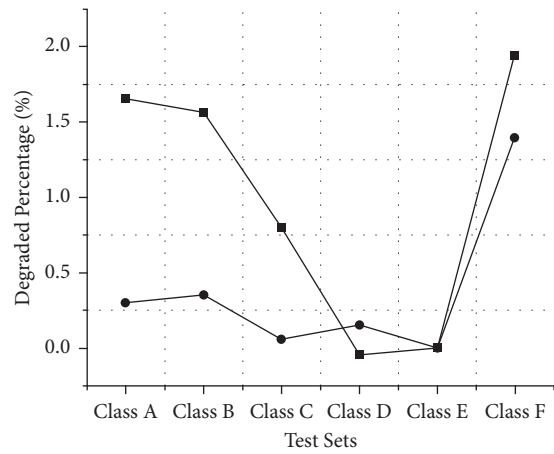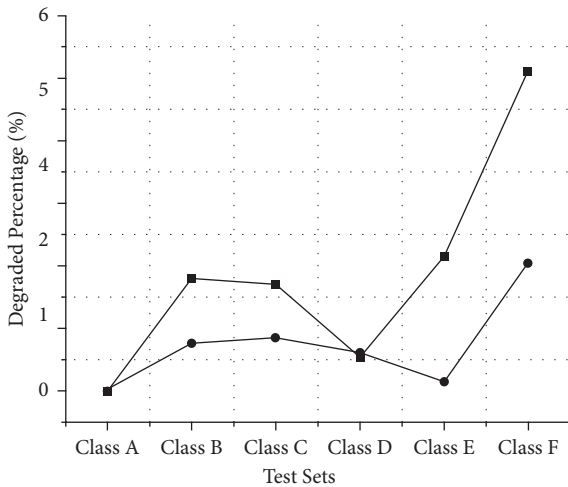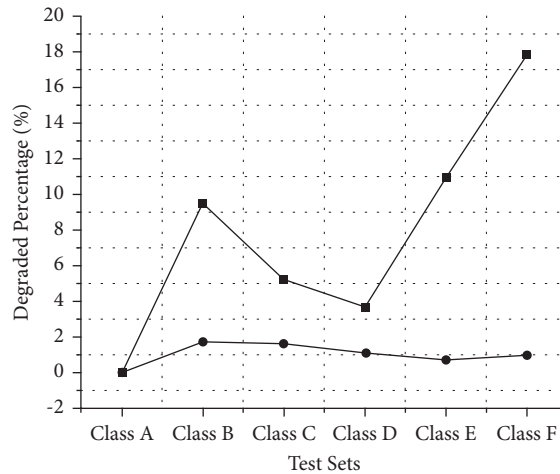
FIGURE 20: Comparison of the HM 16.20 + SCM 8.8 vs. proposed system with different parameters. (a) AI, (b) RA, (c) LDB, (d) LDP

5. Conclusion

In this paper, we present a proposed system with a pre-selector algorithm at the SAO filter for which the parameter is determined for a better distortion rate. The proposed system changes the band-offset determination with the use of a preselector and does not affect the edge offset calculation because it changes only the band selection method in the SAO filter. The approach is better than the previous model in achieving a better BD-rate from Table 4 for BasketballDrill in AI configuration, better video quality, and better BIN file size which is reduced up to 0.65% from Table 3 for BasketballDrill in AI configuration. Meanwhile, the proposed system not only improves the quality of the video but also may improve the physical area optimization in VLSI implementation and memory buffer in processing. In the future, we plan to apply the proposed system to design an Integrated Chip for HEVC which has enormous gate reduction and better memory allocation for accumulators when compared with existing HEVC ICs as hardware encoder-decoder.

Data Availability

The data are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] E. Alshina, G. Sullivan, J. R. Ohm, J. Boyce, and J. Chen, "JVET-D1001: Algorithm Description of Joint Exploration Test Model," in *In Proceedings of the 114th MPEG meeting*, San Diego, CA, USA, February 2018.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[3] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] https://www.itu.int/rec/T-REC-H.265.

[5] J. R. Ohm and G. J. Sullivan, "High efficiency video coding: the next Frontier in video compression [standards in a nutshell]," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 152–158, 2013.

[6] S. Shilpa Kamath, P. Aparna, and A. Antony, "Pixelwise improvised blend of predictors in HEVC lossless mode," *AEU - International Journal of Electronics and Communications*, vol. 114, p. 153000, 2020.

[7] T. Ito, "Future television—super hi-vision and beyond," in *In Proceedings of the 2010 IEEE Asian Solid-State Circuits Conference IEEE*, pp. 1–4, Beijing, China, 2010, November.

[8] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, 2003.

[9] H. Zhang, O. C. Au, Y. Shi, W. Zhu, V. Jakhetiya, and L. Jia, "Improved sample adaptive offset for HEVC," in *Proceedings of the 2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference IEEE*, pp. 1–4, Kaohsiung, Taiwan, November 2013.

[10] X. Dong, L. Shen, M. Yu, and H. Yang, "Fast intra mode decision algorithm for versatile video coding," *IEEE Transactions on Multimedia*, vol. 24, pp. 400–414, 2022.

[11] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, and G. Jiang, "Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1668–1682, 2020.

[12] C. M. Fu, C. Y. Chen, Y. W. Huang, and S. Lei, "Sample adaptive offset for HEVC," in *In Proceedings of the 2011 IEEE 13th International Workshop on Multimedia Signal Processing*, pp. 1–5, Hangzhou, China, 2011, October.

[13] C. M. Fu, E. Alshina, A. Alshin et al., "Sample adaptive offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1755–1764, 2012.

[14] J. Zhu, D. Zhou, S. Kimura, and S. Goto, "Fast SAO estimation algorithm and its VLSI architecture," in *In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP)*, pp. 1278–1282, IEEE, Paris, France, 2014, October.

[15] J. Archana, D. Keerthivasan, S. Janakiraman, and S. Goto, "SAO filter in HEVC: A survey," in *Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 974–979, Coimbatore, India, 2020.

[16] W. S. Kim and D. K. Kwon, "Non-CE8: method of visual coding artifact removal for SAO," in *Proceedings of the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 7th Meeting*, pp. 2–8, Geneva, Switzerland, 2011, November.

[17] W.-S. Kim and D.-K. Kwon, *Method of Visual Coding Artifact Removal for SAO*, p. 14, JCTVCG680, Nov. 2011.

[18] V. C. Jct, *High Efficiency Video Coding (HEVC) Test Model 12 (HM12) Encoder Description*, p. 19, JCTVC-N1002, Aug 2013.

[19] J. Zhou, D. Zhou, S. Wang, S. Zhang, T. Yoshimura, and S. Goto, "A dual-clock VLSI design of H. 265 sample adaptive offset estimation for 8k ultra-HD TV encoding," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 25, no. 2, pp. 714–724, 2017.

[20] https://hevc.hhi.fraunhofer.de/trac/hevc.

[21] F. Bossen, "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, no. 7, 2013.

[22] T. K. Tan, R. Weerakkody, M. Mrak et al., "Video quality evaluation methodology and verification testing of HEVC compression performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 76–90, 2016.

[23] G. Bjontegaard, *Calculation of average PSNR differences between RD-curves*, 2001.

[24] Z. Zhengyong, C. Zhiyun, and P. Peng, "A fast SAO algorithm based on coding unit partition for HEVC," in *In Proceedings of the 2015 6th IEEE International Conference on Software Engineering And Service Science (ICSESS)*, pp. 392–395, Beijing, China, 2015, September.