

An Adaptive Deblocking Filter to Improve the Quality of the HEVC Standard

Alaa F. Eldeken

Department of Computer Engineering, Military Technical College, Cairo, Egypt
Email: eldeken @ieee.org

Gouda I. Salama

Department of Computer Engineering, Military Technical College, Cairo, Egypt
Email: drgouda @ieee.org

Abstract—In this paper, we present an adaptive deblocking filter to improve the video quality for high efficiency video coding (HEVC) scheme. The HEVC standard is a hybrid coding scheme using block-based prediction and transform encoding/decoding. At the decoding step, the boundary of any two adjacent blocks causes visual discontinuities called blocking artifacts that can be removed using deblocking filter. Conventional approaches, including the HEVC standard, tend to remove those artifacts using two offset parameters that are defaulted to 0. However, such a choice is not necessarily suitable to encode/decode all video sequences. The proposed approach reduces an exhaustive search among a set of candidate offsets to eventually select the best offsets adaptively (i.e., for each frame) according to some characteristics of the data sequences. Improvements are shown using the proposed approach in terms of rate-distortion (RD) performance as opposed to the HEVC standard without changing the compression ratio and with negligible change in the encoding/decoding time.

Index Terms—Boundary strength, high efficiency video coding, block boundary, deblocking filter.

I. INTRODUCTION

HEVC is a new video coding standard recently launched in order to save the channel bandwidth and disk space as opposed to the standard H.264/AVC. It is also known as H.265 or MPEG-H Part-2. HEVC has been designed to focus on two key issues: increasing video resolution and increasing the use of parallel processing architectures. HEVC provides 50% more bit-rate reduction and a higher degree of parallelism when compared to H.264/AVC by adopting a variety of coding efficiency enhancement and parallel processing tools [1].

Typically, the H.264/AVC [2, 3, 4] divides a frame into 16x16 fixed size of macroblocks. However, this fixed size limits the ability of the H.264/AVC to encode/decode the high resolution videos. Contrarily, in HEVC, a frame is divided into coding tree units (CTU) of 16x16, 32x32 or 64x64. Each CTU can be further divided into smaller blocks, called coding units (CUs), using a quadtree structure. Each CU can be further split into either prediction units (PUs) or transform unit (TUs).

Using the quadtree structure, the size of each TU, used in the prediction error coding, is ranged from 4x4 up to 32x32 leading to larger transformations than that of the H.264/AVC that only uses 4x4 and 8x8 transforms. In turn, the high resolution videos can be encoded/decoded using the HEVC more efficiently than that of the H.264/AVC standard [1].

In HEVC, the boundary of any two adjacent CUs causes visual discontinuities; called blocking artifacts. These artifacts can be removed using deblocking filtering (DBF). Although, the DBF in HEVC is similar to that in the H.264/AVC standard [5] and is also implemented in the inter-prediction process [6], the DBF design in HEVC is simpler in terms of its decision making process [7]. In HEVC, the DBF is followed by a sample adaptive offset (SAO) filter [8]. In the decoder loop, SAO is applied to the reconstructed samples before writing them into the decoded frame buffer.

The DBF is one of the highly computational parts in HEVC. It has two integer offset parameters that are set manually; O_{β} and O_{T_c} . These two offsets are ranged from -6 to 6 yielding 169 combinations (i.e., pairs). In the HEVC standard, the two offsets are manually set to 0 as a default value; however, such a choice is not necessarily suitable to encode/decode all video sequences. In this paper, an adaptive DBF approach is proposed using a set of candidate pairs that can be chosen from those 169 pairs for each frame. The offset range is condensed to only 4 values leading to a reduced searching process among only 16 pairs. Thus, a reduction by a factor of 10 in the preprocessing time is obtained to eventually enhance the objective video quality compared to that of the HEVC standard.

This paper is organized as follows. The HEVC deblocking filter and the proposed deblocking approach are shown in Section 2 and Section 3, respectively. Experimental results are shown in Section 4. Finally, conclusions are given in Section 5.

II. HEVC DEBLOCKING FILTER

In this section, we discuss the DBF process in HEVC, and its deblocking decisions with challenges and operation. In the DBF process, there are three types of boundaries: CU boundary, TU boundary and PU

boundary. The CU boundaries always include TU boundary and PU boundary [7]. In a DBF process, both vertical and horizontal edges are filtered in a row, and so does the decoding process. The DBF process is applied to 8x8 block boundaries for both luma and chroma components [9].

The DBF process consists of two main steps: determining boundary strength (BS) parameter and its deblocking decisions and operation [7]. In the BS step, the BS parameter is estimated using the information of intra or interceding as well as the motion vector difference.

Table 1 illustrates how the BS value is determined [1]. This implies that there is typically no filtering within the static areas. As well, this step helps avoid multiple subsequent filtering of the same areas where pixels are copied from one frame to another with a residual equal to zero leading to non-over smoothing [7].

Table 1. Definition Of Boundary Strength Values Between Two Adjacent Blocks [1].

Condition	BS
One of the blocks is Intra	2
One of the blocks has non zero coded residual coefficient	1
Differences between corresponding spatial motion vector	1
Motion-compensated prediction refers to different pictures	1
Otherwise	0

Note that the filtering operation ability is determined by both the BS parameter and the quantization parameter (QP). Additional conditions are applied to each luma component block edges to determine whether the DBF strength should be strong or normal to be applied to the block boundary. Before discussing the deblocking decisions step, it's worth noting to first describe the main challenges when designing a deblocking filter and the proposed solutions.

The first challenge arises when deciding whether to basically activate the filtering process or not (i.e., to apply a filter to a certain block boundary or not). One solution is to check whether the CU boundary is PU boundary or TU boundary, provided that the BS parameter exceeds 0 [10]. In case of activating the filtering process, another challenge arises when setting the DBF strength either to strong or normal. One remedy is to check whether the boundaries variation of two adjacent blocks satisfies some conditions or not [8].

For example, Table 2 shows two adjacent blocks X and Y that are nominated to the DBF process, where the deblocking decisions step are based on rows 0 and 3.

A blocking artifact is characterized by low spatial activity on both sides of the block boundary, whereas there is discontinuity at the block boundary [7, 11]. First, we should define two thresholds β and T_C . These parameters depend on the QP that is used to adjust the quantization step for quantizing the prediction error

coefficients [8]. Table 3 [12] shows a piecewise linear dependence between the thresholds β and T_C , and a quantization parameter QP that is ranged from 0 to 55.

Given two quantization parameters of two adjacent blocks X and Y, QPX and QPY, respectively, the parameter QP can be calculated as in (1) [12],

$$QP = ((QPX + QPY + 1) \gg 1) \quad (1)$$

Thus, the thresholds β and T_C can be directly derived from Table 3 [12]. Given two adjacent blocks X and Y as shown in Table 2, it can be shown that the values of β and T_C are first used in activating/deactivating the DBF process as in (2),

$$|X(2,0) - 2X(1,0) + X(0,0)| + |X(2,3) - 2X(1,3) + X(0,3)| + |Y(2,0) - 2Y(1,0) + Y(0,0)| + |Y(2,3) - 2Y(1,3) + Y(0,3)| > \beta \quad (2)$$

Where $X(\cdot)$ and $Y(\cdot)$ denote pixel values. The thresholds β and T_C can be used in deciding the filter strength; i.e., strong or normal in three phases [10]. The first phase checks that there is a low spatial activity on the side of block boundary; the same way shown in (2), but with a lower threshold, as in (3):

$$|X(2,j) - 2X(1,j) + X(0,j)| + |Y(2,j) - 2Y(1,j) - Y(0,j)| < \beta/8 \quad (3)$$

Where $\forall 0 \leq j \leq 3$. The second phase checks that the signal on both sides of the block boundary is flat, as in (4):

$$|X(3,j) - X(0,j)| + |Y(0,j) - Y(3,j)| < \beta/8 \quad (4)$$

Finally, the third phase checks that the difference in intensities of samples on two sides of the block boundary does not exceed a certain value, as in (5):

$$|X(0,j) - Y(0,j)| < 2.5 \times T_C \quad (5)$$

Note that the thresholds β and T_C have their own integer offset parameters, $O\beta$ and OT_C , respectively, where $-6 \leq O\beta, OT_C \leq 6$.

Generally, these two offset parameters control the filtering operation performance. It's worth noting that, both $O\beta$ and OT_C are always set to a default value of 0 in the HEVC standard [12]. However, this choice is not necessary suitable for all video sequences during encoding/decoding process.

III. PROPOSED DEBLOCKING FILTER

In this section, we discuss the proposed DBF approach. To solve the problem arisen in the previous section, we propose an adaptive DBF to better select a suitable offset values for both $O\beta$ and OT_C . This selection considers to the video features to be encoded/decoded to eventually improve the video filtering quality.

First, we exhaustively search for the best offsets for each frame, f_n of a video sequence among all candidate values, $\forall 1 \leq n \leq N$, where N denotes the number of frames to be coded per a data set. Recall that both OT_C and $O\beta$ have 13 values leading to 169 combinations (i.e., pairs) (see Section 1). Each candidate pair is provided to each frame to be encoded/decoded. Then, the rate distortion (RD) performance is determined using the decoded frame, f_n , and its corresponding original one, f_0 . This procedure is iteratively, performed 169 times for all N frames of a video sequence. Given all PSNR and bit-rate values corresponding to all the 169 pairs during the decoding frame, f_n , only one pair, p_i^{n*} chosen corresponding to the maximum RD value at the i^{th} iteration for frame f_n .

Algorithm 1 THE PROPOSED DBF APPROACH

```

1: Given:
2:  $N$ : number of frames to be coded of a data set,
3:  $f^n$ : the  $n^{th}$  frame,  $\forall 1 \leq n \leq N$ ,
4:  $I$ : number of candidate offset pairs,
5:  $i$ : iteration no.,  $\forall 1 \leq i \leq I$ ,
6:  $f_i^n$ : the decoded  $n^{th}$  frame at the  $i^{th}$  iteration,
7:  $f_0^n$ : the corresponding original  $n^{th}$  frame,
8:  $p_i^n$ : candidate pair of  $n^{th}$  frame at  $i^{th}$  iteration, and
9:  $s_i^n$ : Rate-distortion value =  $\frac{PSNR(f_0^n, f_i^n)}{Bit-rate(f_0^n, f_i^n)}$ 
10: Required:  $p_i^{n*}$ : best offset pair for  $n^{th}$  frame.
11: Initialize:  $I=16$  and  $Max=s_0^n$ .
12: for  $n = 1$  to  $N$  do
13:   for  $i = 1$  to  $I$  do
14:     if Equations (2) through (5) are TRUE then
15:       Apply the DBF using  $p_i^n$  to  $f_i^n$ .
16:       Determine the corresponding  $s_i^n(f_0^n, f_i^n)$ .
17:       if ( $s_i^n \geq Max$ ) then
18:         Set  $Max = s_i^n$ .
19:         Set  $p_i^{n*}$  to  $p_i^n$ 
20:       end if
21:     end if
22:   end for
23: end for

```

Given the results of that exhaustive search, we noticed that the 13-value range of each offset parameter can be condensed to only 4 values: -6, -2, 0 and 6 (i.e., 16 pairs). Such a reduced range results in about 94% of the maximum RD values (i.e., better DBF performance) for each frame. Algorithm 1 shows the proposed DBF approach resulting in the best pair, p_i^{n*} that corresponds to the maximum RD value; however, it's chosen from only 16 pairs not 169 ones.

IV. EXPERIMENTS & RESULTS

In this section data sequences, the implementation setup of experiments and results are discussed. The data sets used in the experiments include five classes of real sequences [13].

Each class has three video sequences having different features with characteristics provided in Table 3.

Class-A has a resolution of 2560x1600 (e.g., PeopleOnStreet and Traffic sequences). Class-B has a resolution of 1920x1080 (e.g., BasketballDive and Tennis sequences). Class-C has a resolution of 832x480 (e.g., BasketballDrill and Keiba sequences). Class-D has a resolution of 416x240 (e.g., RaceHorses and BQSquare sequences). Finally, Class-E has a resolution of 1280x720 (e.g., Vidyo1 and Vidyo3 sequences).

Our implementation runs on Intel Core i5 with 4GB of RAM. The proposed approach (i.e., referred to as Proposed) is compared to the HEVC standard (i.e., referred to as standard) [10]. We use the HEVC standard software (HM10) [14] for encoding/decoding the data sets mentioned above.

In this paper, the performance of competing approaches is evaluated by: i) the rate distortion (RD) (in dB/Kbps), ii) the Bjontegaard (BD) rate ratio [15], iii) the compression ratio between the decoded video sequence and its original using the two approaches for all data sets, and iv) the time consumed in encoding/decoding process. The quantization parameter (QP) is set to 22, 27, 32, and 37 [16]. The group of picture (GOP) is set to 8. The configuration module of the encoding/decoding process is set to random access configuration. It worth noting that our implementation first runs for all possible combinations of both offset parameters $O\beta$ and OT_C .

Each is ranged from -6 through 6, yielding 169 possible pairs (i.e., combinations) for each frame of a data set. Having such an exhaustive search, the aforementioned range can be condensed to only four selections for each offset parameter (i.e., -6, -2, 0 and 6) resulting in sharing about 94% of better selections. In other words, the possible 169 combinations of both offset parameters $O\beta$ and OT_C are reduced to only 16 combinations. Thus, the preprocessing time (i.e., searching time for best offsets) is accordingly reduced by nearly a factor of 10.

Note that both competing approaches provide the same compression ratios using all data sets described above.

Fig. 1 and Fig. 2 show the RD performance at different QPs (22, 27, 32, 37), using the whole number of frames of all video sequences described in Table 4 with the two competing approaches. The Proposed approach outperforms the standard in terms of RD on the basis that the higher the better. As well, in terms of the BD rate ratio [15], The Proposed approach surpasses the standard by a maximum increase of -0.50%, -0.33%, -0.31%, -0.23% and -0.42% for data sets of Classes-A, B, C, D and E, respectively, on the basis that the lower the better.

It's worth noting that the compression ratio has not been changed, however, a negligible increase in encoding/ decoding time has occurred by a maximum of 3%. These improvements are due to using better offset pairs for each frame of a data set instead of setting both offsets to 0 as shown in the HEVC standard [12].

V. CONCLUSIONS

In this paper, we modify the deblocking filter process of the HEVC standard. This modification is based on determining better offset parameters in the filtering process to remove the blocking artifacts on the decoded video sequences. The proposed approach uses an adaptive DBF providing an improvement in terms of RD performance without changing the compression ratio and with negligible change in encoding/decoding time compared to the HEVC standard. This improvement comes due to nearly tenfold reduction of the searching time for better offsets.

ACKNOWLEDGMENT

The authors wish to thank Dr. Mohamed M. Fouad, Department of Computer Engineering, Military Technical College, Cairo, Egypt, for his valuable ideas and guidance making a great impact to the completion of this work.

REFERENCES

- [1] G. 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, Dec. 2012.
- [2] J. Shang, W. Ding, Y. Shi, Y. Sun, "Fast Intra Mode Decision Algorithm Based on Texture Direction," *MECS International Journal on Education and Management Engineering*, pp. 384-391, Nov. 2011.
- [3] Z. Hu, T. Wang, K. Chen, Z. Xie and X. Wang, "Operator Design Methodology and Application in H.264 Entropy Coding," *MECS International Journal on Information Engineering and Electronic Business*, pp. 51-58, Nov. 2010.
- [4] 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, July 2003.
- [5] S.-C. Hsia, W.-C. Hsu, and S.-C. Lee, "Low-complexity high-quality adaptive deblocking filter for H.264/AVC system," *Signal Processing: Image Communication*, vol. 27, pp. 749-759, Aug. 2012.
- [6] J. Lou, A. Jagmohan, D. He, L. Lu, and M.-T. Sun, "H.264 deblocking speedup," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 8, Aug. 2009.
- [7] A. Norkin, G. Bjontegaard, A. Fuldseth, and M. Narroschke, "HEVC deblocking filter," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, Dec. 2012.
- [8] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, and C.-Y. Tsai, "Sample adaptive offset in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, Dec. 2012.
- [9] Y. Li, N. Han, and C. Chen, "A novel deblocking filter algorithm in h.264 for real time implementation," in *3rd Intern. Conf. on Multimedia and Ubiquitous Engineering*, March 2009.
- [10] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, Dec. 2012.
- [11] 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, July 2003.

- [12] I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, and W.J. Han, "High efficiency video coding (HEVC) test model draft 10 (HM 10) encoder description," Technical Report Document JCTVC-L1002, JCT-VC, Geneva, Switzerland, Jan. 2013.
- [13] "ftp://hvc:US88Hula@ftp.tnt.uniannover.de/testsequenc-s," 2003.
- [14] "http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches," 2012.
- [15] Gisle Bjontegaard, *Calculation of average PSNR differences between RD-curves*, VCEG-M33, Texas, USA, April 2001.
- [16] Frank Bossen, *Common test conditions and software reference configurations*, JCTVC-D600, Daegu, KR, U.S.A., Jan. 2011.

Authors' Profiles

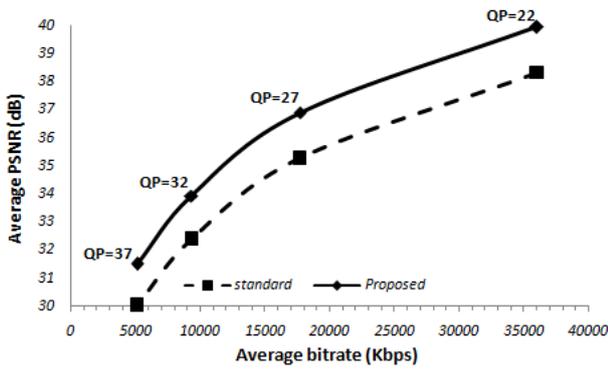


Alaa F. Eldeken, received the Bachelor engineering and Masters' engineering degrees from the Military Technical College (MTC), and Cairo University, respectively, Cairo, Egypt, in 1999 and 2008, respectively. He is currently a PhD student in the Computer Engineering Dept. at MTC in the field of video coding. His research interests include computer simulation, multimedia, computer graphics, image and video processing.

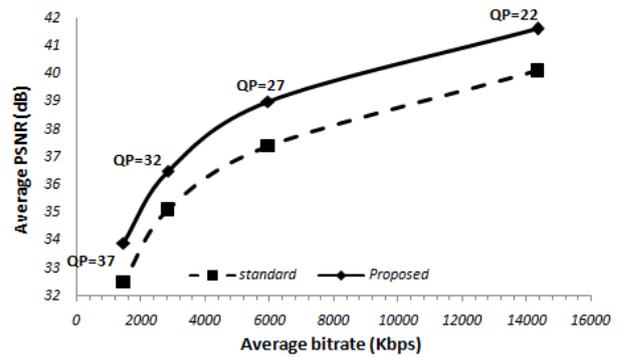


Gouda I. Salama, received the Bachelor engineering and Masters' engineering degrees from MTC, Cairo, Egypt, in 1988 and 1992, respectively. As well, he received the Ph.D. degree in Electrical engineering from Virginia Tech. University, U.S.A., in 2002.

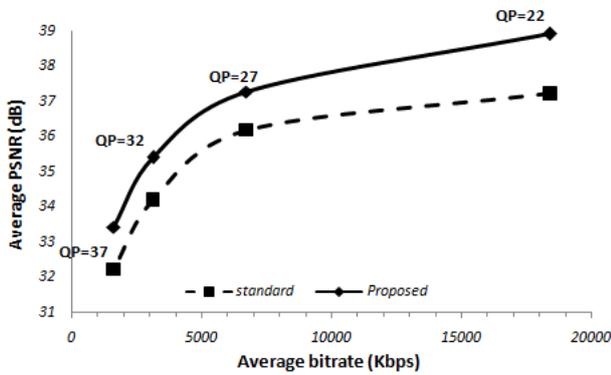
He is currently a faculty member with the Department of Computer Engineering, MTC. His research interests are in image processing, pattern recognition, image and video compression.



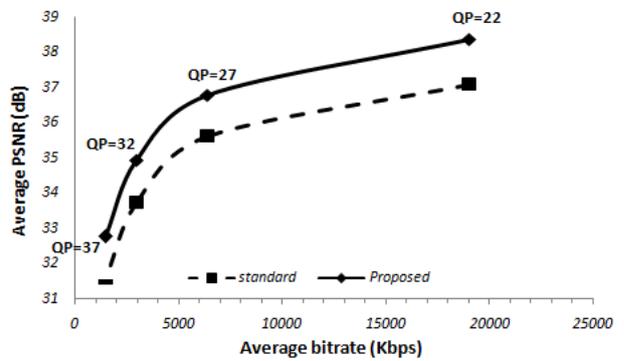
(a) PeopleOnStreet



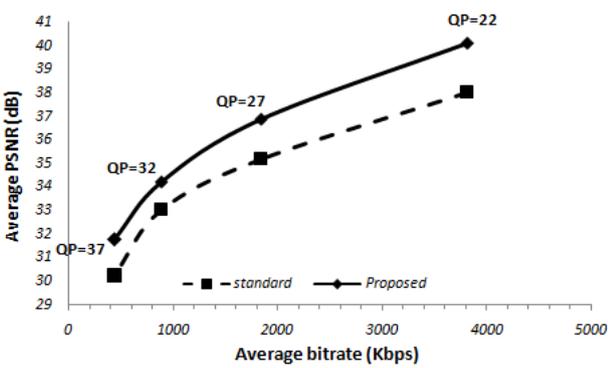
(b) Traffic



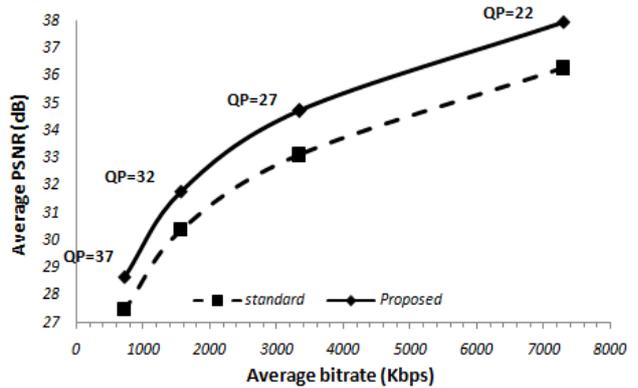
(c) BasketBallDrive



(d) Tennis

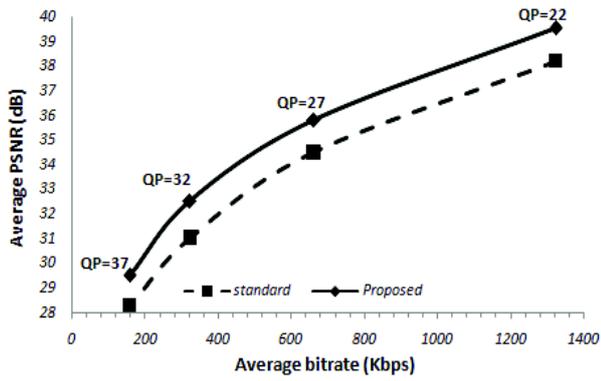


(e) BasketBallDrill

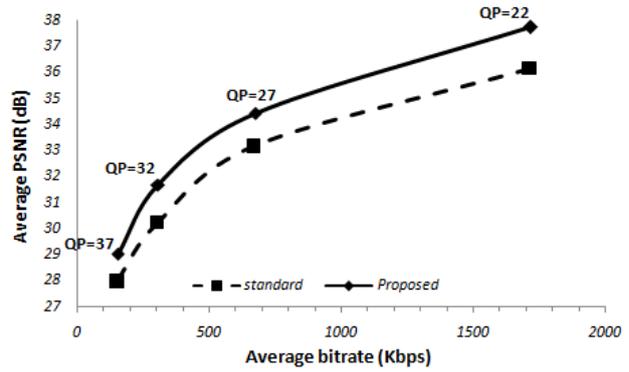


(f) Keiba

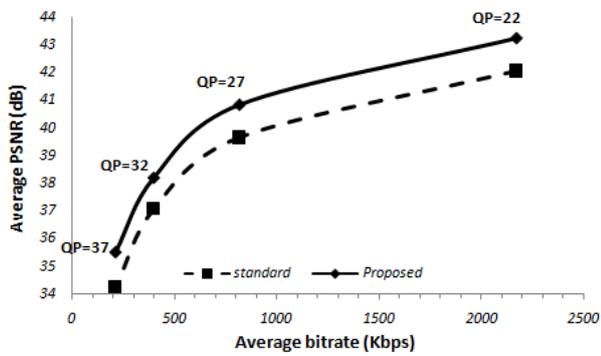
Fig. 1. (a-f) RD performance for different video sequences (PeopleOnStreet, Traffic, BasketBallDrive, Tennis, BasketBallDrill, and Keiba) using the whole number of frames with different QPs (22, 27, 32, 37), the higher the better.



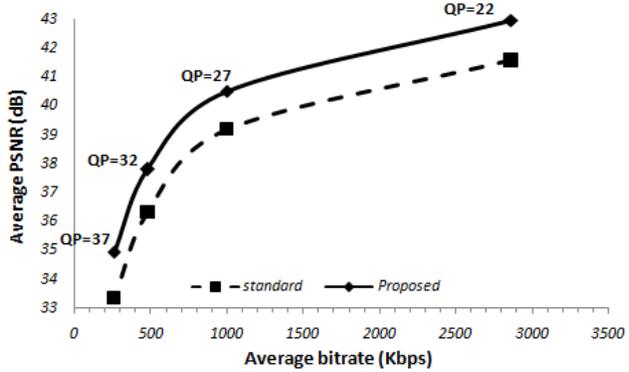
(a) RaceHorses



(b) BQSquare



(c) Vidyol



(d) Vidyol3

Fig. 2. (a-d) RD performance for different video sequences (RaceHorses , BQSquare, Vidyol , and Vidyol3) using the whole number of frames with different QPs (22, 27, 32, 37), the higher the better.

Table 2. Four-Pixel Long Vertical Block Boundary Formed By Two Adjacent Blocks X And Y [1, 11]

x				Y			
X(3,0)	X(2,0)	X(1,0)	X(0,0)	Y(0,0)	Y(1,0)	Y(2,0)	Y(3,0)
X(3,1)	X(2,1)	X(1,1)	X(0,1)	Y(0,1)	Y(1,1)	Y(2,1)	Y(3,1)
X(3,2)	X(2,2)	X(1,2)	X(0,2)	Y(0,2)	Y(1,2)	Y(2,2)	Y(3,2)
X(3,3)	X(2,3)	X(1,3)	X(0,3)	Y(0,3)	Y(1,3)	Y(2,3)	Y(3,3)

Table 3. Derivation of threshold variables β and T_c from input Q [12]

Q	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7	8
T_c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
β	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32	34	36
T_c	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4
Q	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	
β	38	40	42	44	46	48	50	52	54	56	58	60	62	64	64	64	64	64	
T_c	5	5	6	6	7	8	9	9	10	10	11	11	12	12	13	13	14	14	

Table 4. Video sequences description used in the experiments [13]

Seq. #	Sequence Name	File size (MB)	Frame rate (fps)	# of frames	Class	Resolution	Color format
1	PeopleOnStreet	555	30	150	A	2560 x 1600	4:2:0
2	Traffic	563	30	150			
3	Train	1714	60	300			
4	BasketBallDrive	989	50	300	B	1920 x 1080	
5	Tennis	389	50	240			
6	ParkScene	457	24	240			
7	BasketBallDrill	20	50	200	C	832 x 480	
8	Keiba	97	30	500			
9	BQMall	25	60	200			
10	RaceHorses	32	60	300	D	416 x 240	
11	BQSquare	64	60	600			
12	BasketBallPass	46	50	500			
13	Vidyo1	449	60	600	E	1280 x 720	
14	SlideShow	88	20	500			
15	Vidyo3	416	60	600			

How to cite this paper: Alaa F. Eldeken, Gouda I. Salama, "An Adaptive Deblocking Filter to Improve the Quality of the HEVC Standard", IJIGSP, vol.7, no.3, pp.9-15, 2015. DOI: 10.5815/ijigsp.2015.03.02