IJCTA, 8(4), 2015, pp. 1611-1620

Evaluation of Deblocking Filter for H.263 Video Codec & Proposed Algorithm for Entropy Coding for MPEG-4 Video Codec

Imran Ullah Khan[1], M IEEE, M.A. Ansari[2], SM IEEE, S. Hasan Saeed[3]

Abstract: In this Paper Performance of H.263 Video Codec is evaluated for Deblocking Filter for various Video sequences. Also Algorithm for MPEG-4/AVC for Deblocking Filter and Entropy Coding Scheme is proposed. These advanced options are evaluated and analyzed in terms of output parameter, which is Peak Signal to Noise ratio (PSNR) or Quality of Picture. The improvement in video quality with the use of this advance option Deblocking Filter of H.263 is of the order of 0.8 dB or more. Along with Proposal of Algorithm for Deblocking Filter and Entropy Coding Scheme Effect of Deblocking Filter on the Quality of Picture for MPEG-4/AVC is also shown. MPEG-4 is an ISO/IEC video coding standard which supports highly collaborative multimedia applications along with many traditional applications. It includes many advanced functionalities such as Error resilience & concealment, scalabilities and many more. Algorithm for Context Adaptive Binary Arithmetic coding (CABAC) for MPEG-4 Video Codec is proposed. The compression efficiency of CABAC is 10-15% better as compared to CAVLC and it can be up to 32% better when compared to other entropy compression methods such as Huffman.

Keywords: Boundary Strength, COC, Deblocking Filter, CABAC and PSNR.

1. INTRODUCTION

The H.263 Video Codec is the International Telecommunication union Standardization sector for Telecommunication (ITU-T) recommended video coding mechanism for low bit rate communication .Bit rate is upto 64 Kbits per sec .H.263 video codec (enCOding and DECoding) is used in video conferencing[1,2] across analog telephone lines. Networks are low bit rate General Switched telephone network (GSTN) [3-5].

MPEG-4/AVC video codec is jointly developed by groups from International Standardization Organization/ International Electrotechnical Commission (ISO/IEC) Moving Picture Expert Group [6] and ITU-T Video Coding Expert Group. MPEG-4 part 10, Advanced Video Coding and H.264/AVC are generally maintained jointly because of their identical technical content [7]. It is block-oriented motion compensation based video compression standard [8].

Deblocking Filtering is a post processing operation used at the decoder to remove blocking artifacts inherent in any block based video coder .Since, both Discrete Cosine Transform and Motion Estimation/Motion Compensation operations are performed on the block level independently, any incoherence (discontinuities) among the adjacent blocks will appear at the block boundaries, known as blocking artifacts [9].

Entropy Encoding scheme used for H.264/AVC, MPEG-4/AVC and High Efficiency Video Coding abbreviated as HEVC is Context Adaptive Binary Arithmetic Coding (CABAC) because of its better compression than most of the Algorithm used for Entropy encoding in video codecs. CABAC is a lossless compression technique provide better compression capability than all its predecessors[12].

[1,3] Dept. Electronics & Comm. Eng.,Integral University, India, *Email: imranuk79@gmail.com, s.saeed@rediffmail.com*

[2] Faculty of Engineering, Int. Islamic University, Madinah, KSA (on EOL from GBU, NCR, India), *Email: ma.ansari@ieee.org*

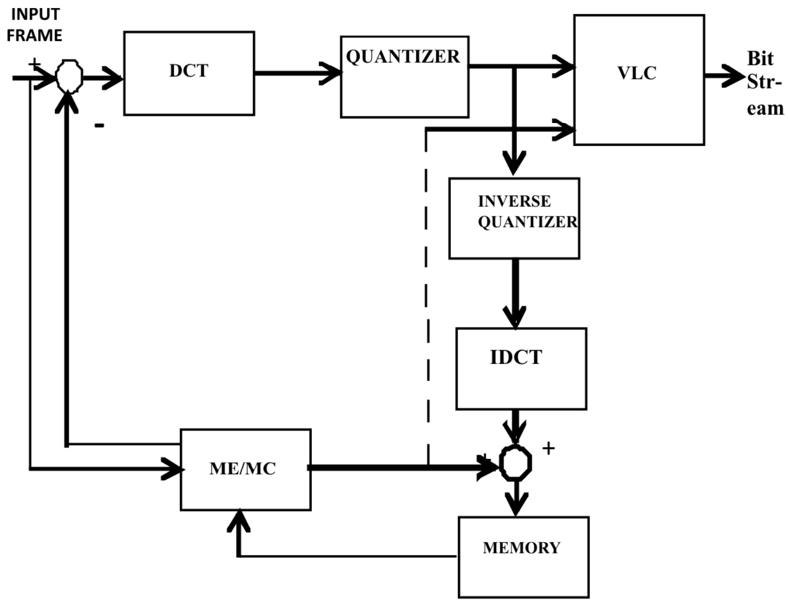


Figure 1: Block diagram of H.263 coder

Where,

DCT	–	**Discrete Cosine Transform**
VLC	–	**Variable Length Coding**
IDCT	–	**Inverse Discrete Cosine Transform**
ME/MC	–	**Motion Estimation/Motion Compensation**

2. DEBLOKING FILTER

Since H.263 and MPEG-4/AVC are very low bit rate video codec (H.263 upto 64 Kbps and MPEG-4 from few Kbps to tens of Mbps) and at very low bit rates, these artifacts are very disturbing and necessary to be eliminated.H.263 and MPEG-4/AVC have the option of de-blocking filters to eliminate the blocking artifacts which is not available in conventional Video codec[10,11].The filtering operation is performed at the reconstructed frames along the 8x8 DCT block edges at the decoder. It is applied separately for luminance (Y-intensity of light radiated from a surface per unit area in a specified direction) as well as chrominance (U & V-signal used to convey color info of the picture $U =$ blue – Y and $V =$ red – Y). The filtering operation is performed for all block boundaries in two steps first along the horizontal edges than it is followed by the vertical edges.

There are abrupt change is observe in frequency components near the block boundaries. In the Uncompressed image and video these block boundaries can be identified by Human Visual System. It is known as a blocking artifact. Filter is used to improve the image quality, remove distortion, and increase Quality of picture (PSNR). [11].

An Algorithm is proposed for Deblocking filter in the figure 1.The procedure carried out on macroblocks. First fast boundary strength decision is applied that for boundary strength low values as 1 a weak filter is sufficient but for BS 3 and 4 strong and special filter is recommended.

3. CABAC AND ITS PROPOSED ALGORITHM

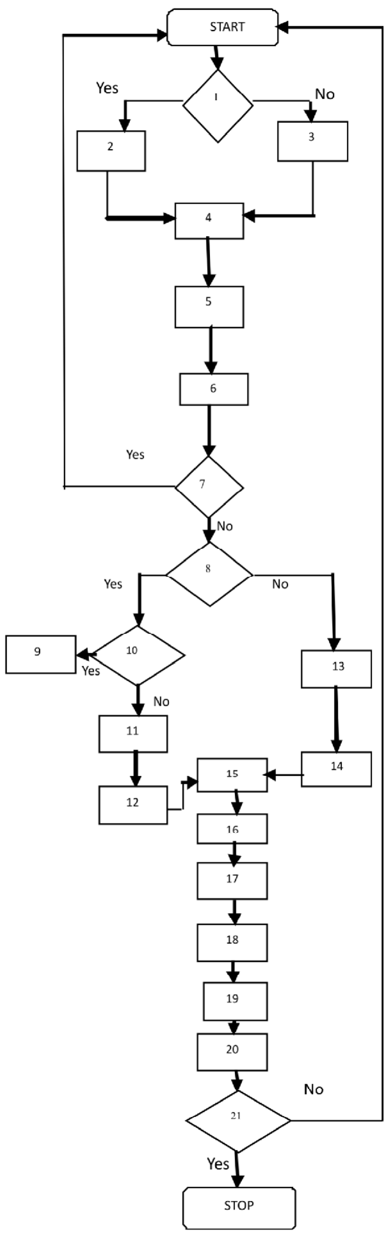
Figure 2 shows the graphical representation of the CABAC decoding algorithm. Once system receives the “Start” signal from the ‘Host’ or driver it begins decoding the bit stream where it initializes the Context Memory only before the first macroblock of the slice[13,14].

If current Macroblock is not the first, the system prefixes the information from the Top/Left neighbors required by the current macroblock, and this information is stored into the local memory of the system. After fifth block (Load info from global neighbor memory to Local memories) of algorithm as shown in figure next step is parameter initializations, after this the system begins decoding the various Syntax Elements from the MPEG-4/AVC coded bitstream in the order as specified by the standard. For 'Skip' macroblock means the system bypasses all decoding processes and begins decoding the next macroblock [15].

Once the decoding of all present syntax elements is completed for the required macroblock, this information is stored for future use as neighbor information. The macroblock information is also then parceled according to required output format and sent to an output buffer. This process of decoding the Syntax Elements for each macroblock continues till the end of slice is reached [15,16].

Despite high compression efficiency, the usage of CABAC is limited because of the fact that it is more computation-intensive than the other entropy coding schemes currently employed in video domain. Moreover, CABAC decoding is essentially a bit-serial process and has strong data dependency, because the decoding result of one bit always has a direct effect on the decoding process of the following bit. This feature makes it difficult for parallel and pipeline implementation of decoding process and hence it is extremely arduous for General Purpose Processor (GPP) and Digital Signal Processor (DSP) to perform CABAC real time encoding/decoding.

CABAC also uses four basic types of context models based on conditional probability. The first type uses a context template that includes up to two past neighbors to the syntax element currently being encoded. For instance



Where
1 IS First MB of Slice
2 Initialization context memory & other CABAC parameters
3 Determine Top/Left neighbor MBs for Current MB
4 MB parameters initialize
5 Load info from global neighbor memory to Local memories
6 Decode MB type and 8X8 type
7 IS skip MB?
8 IS intra?
9 Decode IPCM MB
10 IS IPCM?
11 Read intra prediction mode
12 Read chroma intra prediction mode
13 Read reference frame information
14 Read motion vector differential info
15 Read CBP info
16 Read transform flag info
17 Read Delta quantization info
18 Read coefficient info
19 Store current MB info in global neighbor memory
20 Package current MB info in global neighbor memory
21 IS last MB of slice

Figure 2: Proposed Algorithmic View of CABAC Decoding

modeling may use a neighbor immediately before and an immediately above the current element, and further, the modeling function may be based on bin-wise comparison with neighbors.

4. SIMULATION, IMPLEMENTATION DETAILS AND RESULTS OBTAINED

The H.263 video codec software used in this work was developed by university of British Columbia Canada; Version 3.0 is used for this work. This coder can accept input video of various formats and includes almost all options including that for advanced mode defined for H.263 standard.

Important parameters in the analysis are as follows (Numbers in bracket are default values).

-i <filename> original sequence (required parameter)

-a <n> image to start at [0]

-b <n> image to stop at [0]

-x <n> (<pels> <lines>) coding format

n = 1 SQCIF n = 2; QCIF n = 3: CIF n = 4: 4CIF

n = 5 16 CIF n = 6; custom [12 : 11 PAR]

SQCIF (128 × 96) QCIF(176 × 144) CIF (352 × 288)

4 CIF (704 × 576) 16 CIF (1408 × 1152)

-k <n> framer to skip between each encoded frame

-r <n> target bit rate in bits/sec default is variable bit rate.

-m write repeated reconstructed frame to disk [off]

In this Paper the performance of H.263 video codec is evaluated for Deblocking Filter

-J Deblocking Filter

The performance are evaluated in terms of subjective quality of reconstructed video, measured in terms of peak signal to noise ratio defined in eq. 2 for different target bit rates.

$$\mathrm{COC} = \frac{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} i(x,y) \quad e(x,y)}{\sqrt{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} i(x,y)^2} \quad \sqrt{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} e(x,y)^2}} \ldots\ldots\ldots\ldots \quad (1)$$

The PSNR for each frame is defined as:

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{255}{\mathrm{MSE}}\right)^2 = 20\log_{10}\left(\frac{255}{\mathrm{MSE}}\right) \text{ (for each Y,U,V)}\ldots.. \quad (2)$$

Where MSE is Mean square error

$$\mathrm{MSE} = \frac{1}{XY}\sum_{x=1}^{X}\sum_{y=1}^{Y}[i(x,y) - e(x,y)]^2 \ldots\ldots\ldots\ldots\ldots\ldots.. \quad (3)$$

Where, $i(x, y)$ = Intensity of input pixel (for each Y, U, V) e (x, y) = Intensity of output pixel (for each Y, U, V)

The PSNR for entire video sequence is defined in terms of average PSNR

$$\text{Average PSNR} = \frac{i}{t}\sum_{i=1}^{t}\text{PSNR(i)} \quad \text{for each(Y,U \& V)......} \tag{4}$$

Where t is total number of frames in video and PSNR (i) is the PSNR value for i^{th} frame.

Table 1 gives the summary of results obtained in this work regarding the performance of H.263 Video codec for Deblocking filter for Angiography video sequence at different target bit rates.

It should be noted here that improvement in the subjective quality of decoded video is obtained at the cost of increased computational complexity.

The SSIM is a structural similarity index(similarity measuring full reference metric between two images, means the measurement of image quality based on a reference initial uncompressed or distortion-free image). It can also be understood as an improved version of traditional methods like PSNR and MSE, generally which is inconsistent with human eye perception.

$$\text{SSIM(x,y)} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}\text{............} \tag{5}$$

Where

μ_x the average of x μ_y the average of y;

σ_x, σ_y and σ_{xy}, the variance of x, y and convariance of xy;

$c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$

c_1 and c_2 are two variables to stabilize the division

with weak denominator;

In the equation of c_1 and c_2 L is the dynamic range of the pixel-values and $k_1 = 0.01$, $k_2 = 0.03$ by default.

Equation (5) of SSIM is applicable only on Y ie. luma. Its value ranges between-1 and 1, where 1 is only possible if two sets of data are identical. Generally it is considered on window sizes of 8 × 8.

Table 1
Angiography Video Sequence Simulation Results For Simple, Deblocking Filter

Angiography OBR (Kbps)	*Simple*		*DF*	
	COC	*PSNR (dB)*	*COC*	*PSNR (dB)*
120.7	0.99111	29.87	0.99113	30.15
203.47	0.99848	32.58	0.99852	32.66
304.26	0.99971	34.31	0.99978	34.55
404.32	1.00004	35.63	1.00006	35.78
507.65	1.00029	36.71	1.00034	36.89
607.9	1.00066	37.51	1.0007	37.71
710.06	1.00108	38.19	1.00112	38.32
810.02	1.00117	38.88	1.00126	39.14
913.65	1.00132	39.55	1.00145	39.72
1018.13	1.00139	39.98	1.00152	40.09

Table 2
Heart Video Sequence Simulation Results For Simple, Deblocking Filter

HEART OBR (Kbps)	*Simple*		*DF*	
	COC	*PSNR (dB)*	*COC*	*PSNR (dB)*
20.05	0.99746	30.18	0.99766	30.81
40.1	0.99848	32.71	0.99854	33.03
60.14	0.99969	34.91	0.99969	34.99
80.18	1.00009	36.25	1.00076	36.31
100.23	1.00068	37.47	1.00132	37.77
120.23	1.00112	38.5	1.00179	38.71
140.24	1.00151	39.23	1.00156	39.55
160.25	1.00186	40.27	1.00169	40.6
180.27	1.00237	41.41	1.00238	41.67
200.29	1.00277	42.42	1.00284	42.72

Table 3
Salesman Video Sequence Simulation Results For Simple, Deblocking Filter

SALESMAN OBR (Kbps)	*Simple*		*DF*	
	COC	*PSNR (dB)*	*COC*	*PSNR (dB)*
148.85	0.98988	28.93	0.98996	28.99
200.73	0.99751	30.35	0.99756	30.45
300.86	0.9984	31.96	0.99843	32.14
401.12	0.9985	33.04	0.99859	33.3
501.42	0.99862	33.86	0.99966	34.06
601.67	0.99976	34.52	0.99968	34.74
702.08	0.99989	35.08	0.99994	35.33
802.04	1.00003	35.6	1.00006	35.83
902.36	1.00011	36.05	1.0002	36.29
1002.99	1.00021	36.48	1.00024	36.68

Table 4
Shields Video Sequence Simulation Results For Base Profile And Cabac Enabled

Data Rate (Mbps)	*PSNR(dB) Base Profile*	*PSNR(dB) CABAC Enabled*	*SSIM Base Profile*	*SSIM CABAC Enabled*
0	31.9	32.3	0.88	0.88
0.2	37.3	37.6	0.93	0.932
0.5	42	42.4	0.96	0.963
0.8	44.5	45	0.972	0.972
1	45.9	46.5	0.974	0.974
1.2	46.9	47.4	0.974	0.9755
1.5	47.9	48.5	0.976	0.9764
1.8	48.8	49.7	0.977	0.978
2	49.7	50.2	0.979	0.9798
2.2	50.5	50.9	0.98	0.982
2.5	51.1	51.6	0.981	0.984

Sequence title	:	Angiography
Resolution	:	176 × 144
Number of frames	:	300
Color space	:	YUV 4:2:0
Frames per Second	:	30
Source	:	Uncompressed progressive

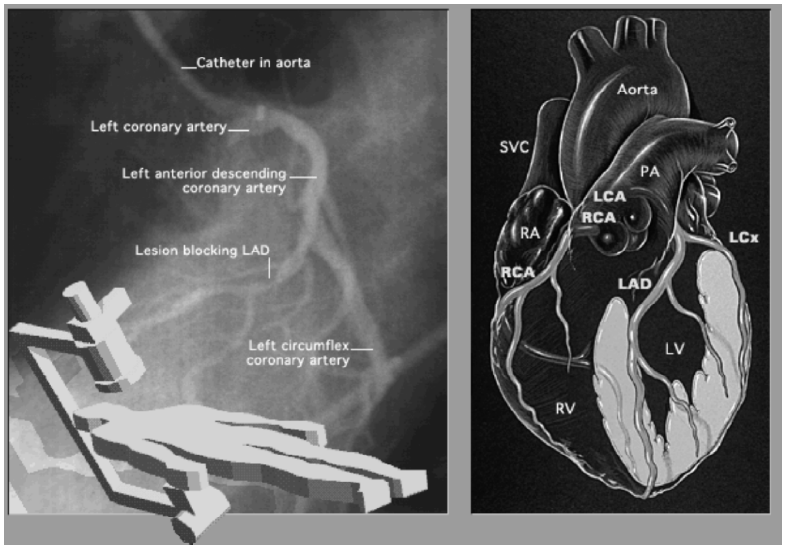


Figure 3: Snapshot of "Angiography" video sequence, frame 30

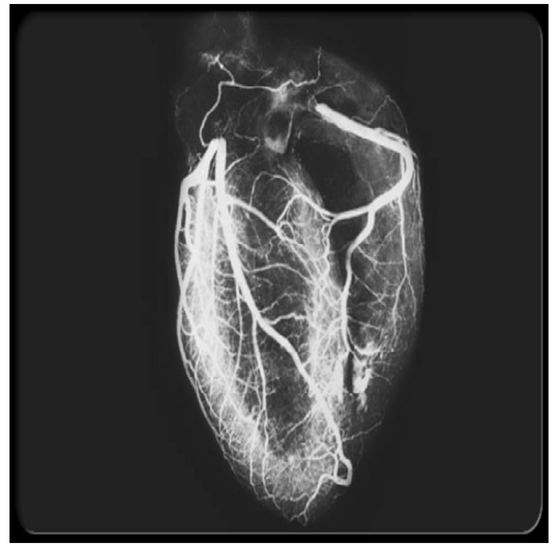
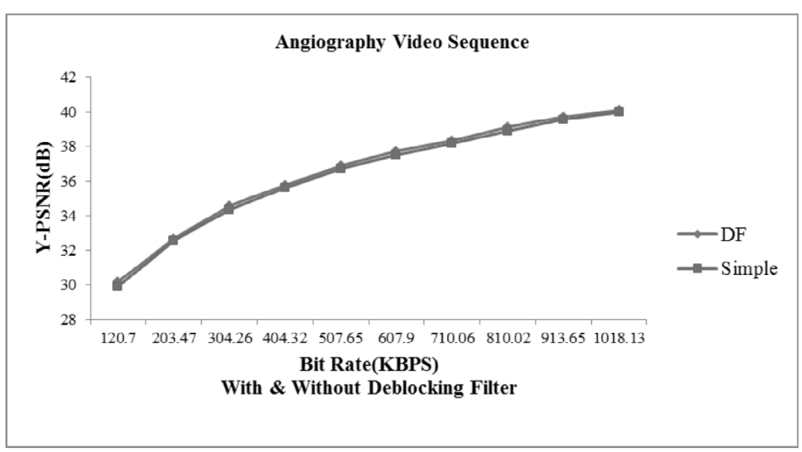


Figure 4: Snapshot of "Heart" video sequence, frame 30

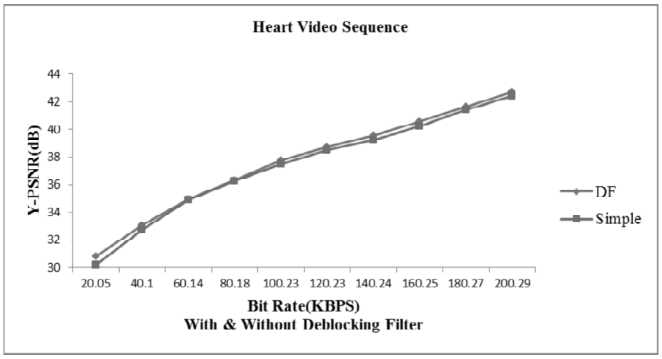
Figure 5: Snapshot of "Salesman" video sequence, frame 20

Sequence title	:	Shields
Resolution	:	High Definition (1280 × 720)
Number of frames	:	252

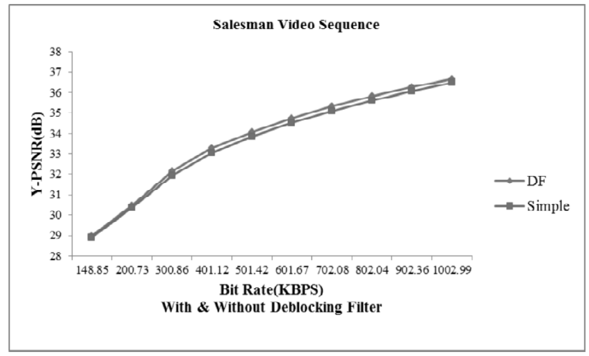
Fig. 6 Snapshot of "Shields" video sequence, frame 252



(a)

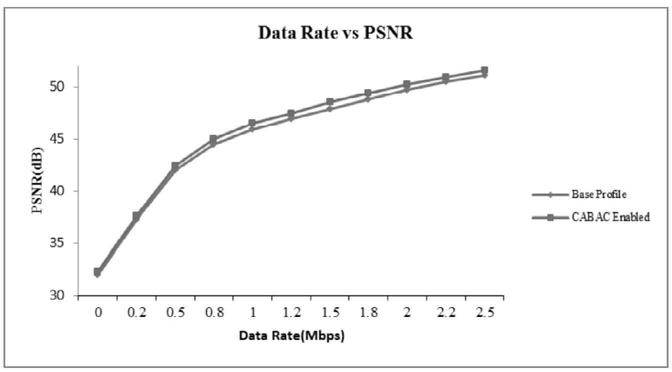


(b)

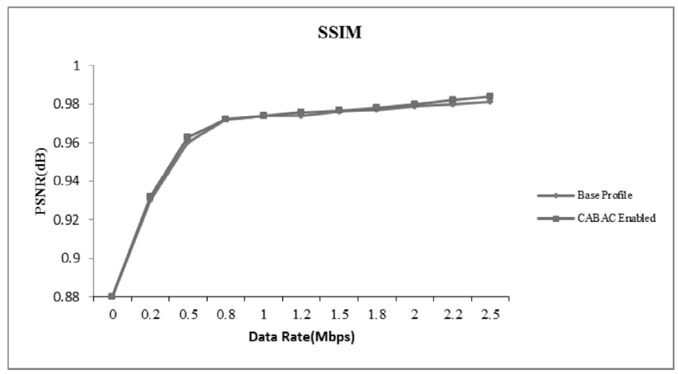


(c)

Fig. 7. Comparative Performance of H.263 Video Coder for Angiography, Heart and salesman video sequence with and without Deblocking Filter



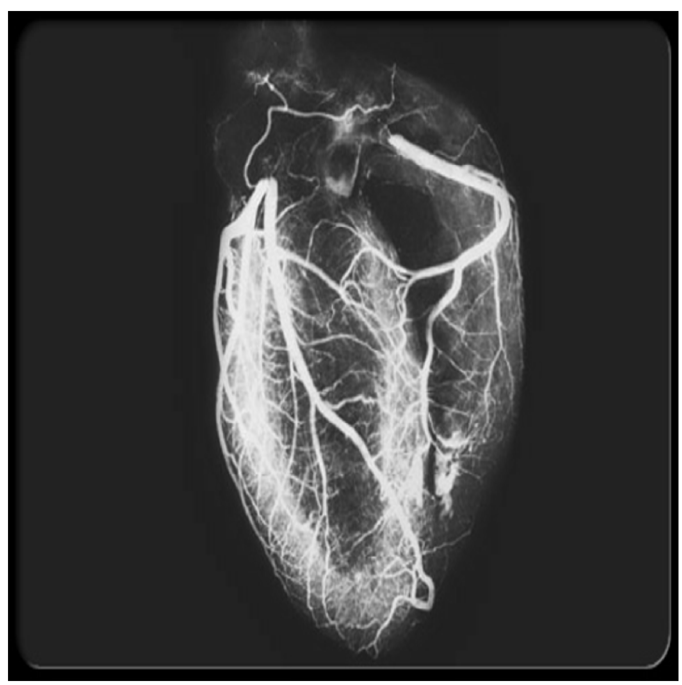
(a)



(b)

Fig.8 Comparative Performance of MPEG-4 Video Coder for Shields video sequence for Base profile and CABAC enabled (a) Data rate vs PSNR (b) SSIM profile

(a)



(b)

Fig. 9 Frame 34 of QCIF HEART sequence encoded at 40Kbps (a) Without filter (b) With loop filter

5. CONCLUSION

The performance of H.263 codec with deblocking filter is as follows:

H.263 has option to use deblocking filter (DBF) (-J option) to reduce the blocking artifacts in the decoded frames. Figure 5 show the graph of Y-PSNR vs bit rate for Angiography sequence with and without Deblocking Filter.

(1) On comparing the graphs it is observed that the use of Deblocking Filter in H.263 video codec improve its performance by 0.06-0.7 dB at the same bit rate.

(2) The simulation result for MPEG-4 is shown in figure 6. The reconstructed image without Deblocking Filter is shown in figure 7(a). The important information is distorted in reconstructed image without Deblocking Filter. The reconstructed image with Deblocking Filter is shown in figure 7 (b). The reconstructed image with Deblocking Filter is more closely matched to the original image. Deblocking filter algorithm was proposed for video real time encoding or decoding in H.264. According to the results, it was verified that computational cost can be decreased considerably with better subjective video quality, especially, in low rate coding or low activity sequence.

(3) The improvement in COC is upto 12 E-05.

The performance of MPEG-4 codec with CABAC shown in figure 6 (a) & (b)fig is as follows:

1. The improvement in video quality with the use of CABAC is of the order of 0.3-0.9 dB.

2. With CABAC enabled profile SSIM improved upto 0 to 0.003.

References

[1] M. Ghanbari, "Video Coding: an introduction to standard codes;" IEE press, London 1999.

[2] "Video coding for low bit rate communication," Int. Telecommun.Union-Telecommun. (ITU-T), Recommendation H.263, version 1,1995; version 2, 1998; version 3, 2000.

[3] Draft ITU-T Recommendation H.263+ "Video coding for very low bit rate communication," Sep 1997.

[4] "ITU-T H.263 Encoder, version 2",Signal Processing and Multimedia Group, University of British Columbia, Canada.

[5] ITU-T, "Video coding for low bit rate communication," ITU-T Recommendation H.263, Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov. 2000.

[6] Frank H. P. Fitzek "MPEG–4 and H.263 Video Traces for Network Performance Evaluation," IEEE Transactions on Circuits and Systems, Vol.2 (2001).

[7] S. Kwon, A. Tamhankar, K. R. Rao, "Overview of H. 264/MPEG-4 Part 10", IEEE Transactions on Circuits and Systems for Video Technology, 2003.

[8] "D.Marpe, T.Wiegand and G.J.Sullivan, "The H.264/MPEG-4 Advance video coding standard and its applications", IEEE Communications Magazine, vol.44, no.8, pp.134-144, Aug.2006.

[9] S.Vijay,C.Chakrabarti,L.J.Karam "Parallel Deblocking Filter For H.264 AVC/SVC", IEEE international conference, 2010.

[10] F. Pan, X. Lin, S. Rahardja, W. Lin, E. Ong, S. Yao, Z. Lu, and X. Yang, " A locally adaptive algorithm for measuring blocking artifacts in images and videos," Signal Processing: Image Communication, 19,2004, pp. 499–506.

[11] Brian Dickey, "Hardware Implementation of a High Speed Deblocking Filter for the H.264 Video Codec" MS thesis 2012.

[12] Langdon G.G.An introduction to arithmetic coding IBM journal, res. Develop, 28:2, pp.135-149, 1984.

[13] P. G. Howard, and J. S. Vitter, "Practical implementations of arithmetic coding", Image and Text Compression, J. A. Storer, Boston, MA: Kluwer, pp. 85-112, 1992.

[14] D. Marpe, H. Schwarz, and T. Wiegand, "Context-adaptive binary arithmetic coding for H.264/MPEG-4 AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 620–636, Jul. 2003.

[15] Langdon G.G.An introduction to arithmetic coding IBM journal, res. Develop, 28:2, pp.135-149, 1984.

[16] P. G. Howard, and J. S. Vitter, "Practical implementations of arithmetic coding", Image and Text Compression, J. A. Storer, Boston, MA: Kluwer, pp. 85-112, 1992.

[17] Martinus Johannes Pieter Berkho "Analysis and Implementation of the H.264 CABAC entropy decoding engine", Computer Engineering Mekelweg 4, Thesis Master of Science, The Netherlands 2010.

[18] Eeckhaut, H. "Optimizing the critical loop in the h.264/AVC cabac decoder" IEEE International Conference on FPT 2006, December, Department of electronic and information systems, Ghent University, Belgium 2006.

[19] N. Keshaveni, S. Ramachandran, K.S. Gurumurthy, "Implementation of Context Adaptive Variable Length Coder for H.264 Video Encoder", International Journal of Recent Trends in Engineering [ISSN: 1797-9617] by the Academy Publishers, Finland, Jan. 2010.

[20] Ha. V.H.S, W. S. Shim, and J. W. Kim, "Real-time MPEG-4 AVC/H.264 CABAC entropy coder," in International Conference on Consumer Electronics Digest of Technical Papers, p.p 255-256, Jan. 8-12, 2005.

[21] R. R. Osorio and J. D. Bruguera, "High-Throughput Architecture for H.264/AVC CABAC Compression System", IEEE Transactions on Circuits and Systems for Video Technology, Vol.16, No.11, pp.1376-1384, Nov. 2006.

[22] Martinus Johannes Pieter Berkho "Analysis and Implementation of the H.264 CABAC entropy decoding engine", Computer Engineering Mekelweg 4,Thesis Master of Science, The Netherlands 2010.