International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 28711-28731
© Research India Publications
http://www.ripublication.com

Analysis and Evaluation of Proposed Algorithm For Advance Options of H.263 and H.264 Video Codec

M. A. Ansari[1]
Senior Member IEEE, Faculty of Engineering, Islamic University,
Madinah, KSA (on EOL from GBU, Gr. Noida, NCR, India)
Email: ma.ansari@ieee.org
Imran Ullah Khan[2]
Dept. Electronics & Communication Engineering
Integral University, Lucknow, India, Email: imranuk79@gmail.com

Abstract

This Paper deals with the Evaluation, Analysis and Algorithm of Advance options of H.263 and H.264/AVC Video Codec. Analysis and Evaluation of H.263 Video Codec is important because it enable us to determine output parameter, which is Peak Signal to Noise ratio (PSNR) or Quality of Picture. In this Paper, the performance of H.263 Video Codec is evaluated and analyzed for various Advance options such as Syntax based Arithmetic coding (SBAC), Advanced Prediction mode, Unrestricted Motion Vector mode etc. H.264/AVC is an international video coding standard, jointly developed by groups from ISO/IEC and ITU-T, which aims at achieving improved compression performance. Various applications of H.264/AVC include conversational, storage, and streaming. For H.264 Video Codec we proposed Algorithm for Advance options such as Context Adaptive Binary Arithmetic coding (CABAC), and Intraprediction. The improvement in video quality with the use of these advance options of H.263 is of the order of 0.8 dB or more. The compression efficiency of CABAC is 10-15% better as compared to context adaptive variable length coding (CAVLC) and it can be up to 32% better when compared to other entropy compression methods such as Huffman.

Keywords: CABAC, CAVLC, Intraprediction, PSNR and SBAC.

Introduction

The H.263 Video codec is the international telecommunication union (ITU-T) recommended standard for very low bit rate video compression. Currently, H.263 encoding and decoding is the predominant technology used in video conferencing across analog telephone lines, optimized at rates below 64 Kbits per sec. The primary goal in the H.263 standard codec was coding of video at low or very low bit rates [1-3] for applications such as mobile networks, Public switched telephone networks (PSTN) and the narrow band integrated service digital network (ISDN). This goal could only be achieve with small image sizes such as sub-Quarter Common Intermediate Format (SQCIF) and Quarter Common Intermediate Format (CIF) at low frame rates [4, 5].

Its predecessor H.261 has the same coding algorithm with some changes to improve its performance and error recovery. The H.264/AVC video coding standard explicitly defines all the syntax elements, such as motion vectors, block coefficients, picture numbers, and the order they appear in the video bitstream.

H.263 and Its Parameters

The H.263 video coder has many advanced features, which were not available in its predecessor H.261.

H.263 has following new features in its algorithm.

1. Half pixel motion compensation.
2. Improved variable length coding & reduced overhead.
3. Unrestricted motion vectors that are allowed to point outside the picture area.
4. Arithmetic coding in place of variable length (Huffman) coding.
5. Advanced motion prediction including overlapped block motion compensation.

This video codec supports five resolutions QCIF, CIF, SQCIF, 4CIF and 16CIF. It is observed generally that H.263 achieves the same quality as H.261 coders at about half the bit rate[6,7].

The block diagram of H.263 video codec is shown in figure1. Varying different parameters may control the features and functionalities of H.263, but for this work only those basic parameters that provide trade-off between video quality, transmission bandwidth and computational complexity are considered [8-10].

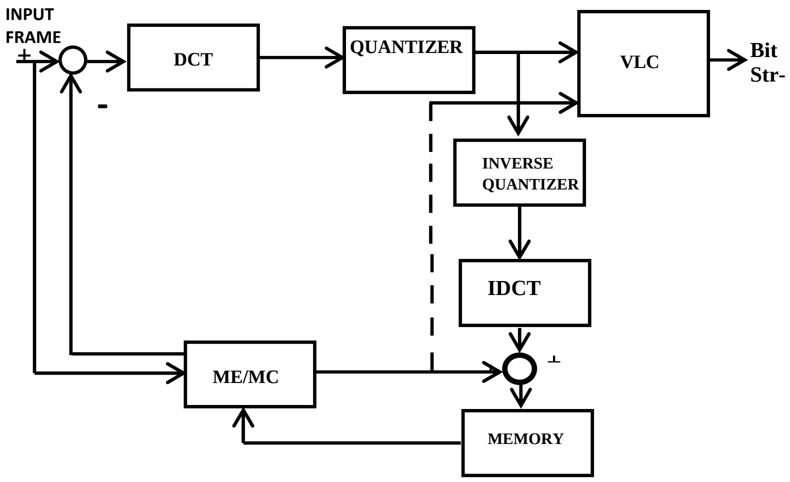


Figure 1: Block diagram of H.263 coder

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

a. Syntax Based Arithmetic Coding

In Arithmetic coding data is compressed in such a way that it encodes data on a number line between 0 and 1by creating a code string which represents a fractional value. There is clear separation between the encoding of information of a model and data representation of that model. There are two types of modeling used in arithmetic coding: fixed model and adaptive model [11, 12].

b. Advanced Prediction Mode

The optional advanced prediction mode is explained in terms of overlapped block motion compensation and four motion vectors per macroblock.

c. Four Motion Vectors Per Macroblock

In conventional H.263, there is one motion vector designated for each macroblock but in the advanced prediction mode there are four motion vectors are used per macroblock

d. Overlapped Motion Compensation

The overlapping motion compensation is used only for the 8x8 luma blocks. In an 8x8 luma prediction block each pixel is find by taking the weighted sum of three prediction values and then this weighted sum is divided by 8(with rounding). Upto three three motion vectors can be used to find the prediction values. [13].

e. Unrestricted Motion Vector Mode

In the conventional mode, there are restriction on motion vectors thereby all pixel referenced by the motion vectors are lies inside the coded picture area but in the optional mode motion vectors are unrestricted. For this purpose an edge pixel is used [13].

Overview of H.264 Encoder and Decoder

Additional Features of H.264/AVC video Codec which distinguish it from the previous video compression standards are listed below [14, 15]:

1. Two context adaptive coding schemes.
2. B-frame can be used as reference frame.
3. Multiple reference frame motion compensation is allowed, which also improves the prediction accuracy. The restriction that only the immediate previous frame can be used as reference frame is thus removed.
4. More motion compensation block sizes and shapes, such as 8x4, 4x8 and 4x4 are supported. The minimum luma motion compensation block size can be as small as 4x4.
5. ¼ pixel motion estimation improves prediction accuracy. It has the same prediction accuracy as in MPEG-4 but with lower interpolation complexity.
6. Directional spatial prediction is applied in intra-coded macroblocks (MBs) of pictures to reduce the amount of information before their block transform.
7. In-loop deblocking filtering removes the blocking artifacts caused by transform and quantization [15].
8. Small block-size transform of 4x4 is used rather than 8x8 used in earlier standards, which results in less ringing artifacts.
9. The Discrete Cosine Transform (DCT) is replaced by integer transform that is exact- match inverse transform, thus avoiding drift during inverse transform.
10. Parameter sets are used between the encoder and decoder to achieve synchronization in terms of syntax.
11. Flexible macroblock ordering (FMO) partitions a frame into different slice groups.
12. Data partitioning groups a slice in up to three packets by their importance.
13. Encoder can send redundant representation of some regions of a frame to enhance robustness to data loss. The Fig.2 shows a typical H.264/AVC video Encoder.

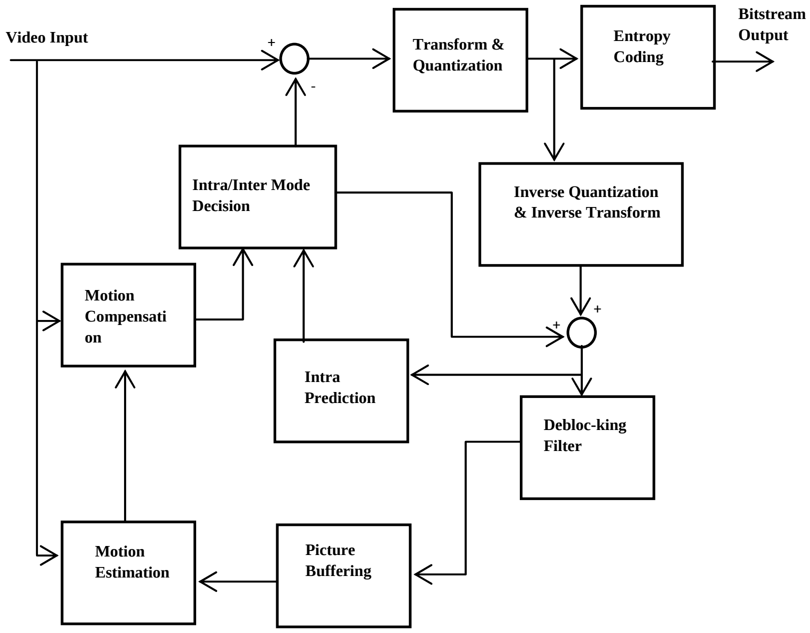


Figure 2: Block Diagram of H.264/AVC Encoder

The Syntax actually is the most important tool for ensuring compliance and error detection. Like other video coding standards, H.264/AVC only defines the syntax of the decoder in order to allow flexibility in specific implementations at the encoder.

The decoder receives a compressed bit stream from the Network Abstraction Layer (NAL). These are rescaled and inverse transformed and using the header information decoded from the bitstream [16] the decoder creates a prediction macro block identical to the original prediction formed in the encoder. Figure3 shows a typical H.264/AVC video Decoder.

Technical Overview of H.264

The H.264 design supports the coding of video (in 4:2:0 **chroma** format) that contains either progressive or interlaced frames. Generally a frame of video contains two interleaved field.

a. Network Abstraction Layer (NAL)

The Video Coding Layer (VCL), which is described in the following section, is specified to represent, efficiently, the content of the video data. The NAL is specified to format that data and be responsible for header.

b. Video coding layers

The video coding layer of H.264 is similar in script to other standards such as MPEG-2 video. It consists of a hybrid temporal and spatial predictions, in conjunction with transform coding. Figure-2 shows the H.264 encoder. In common with earlier coding standards, H.264 does not explicitly define a Codec but rather defines the syntax of an encoded video bit stream together with the method of decoding this bitstream [17].

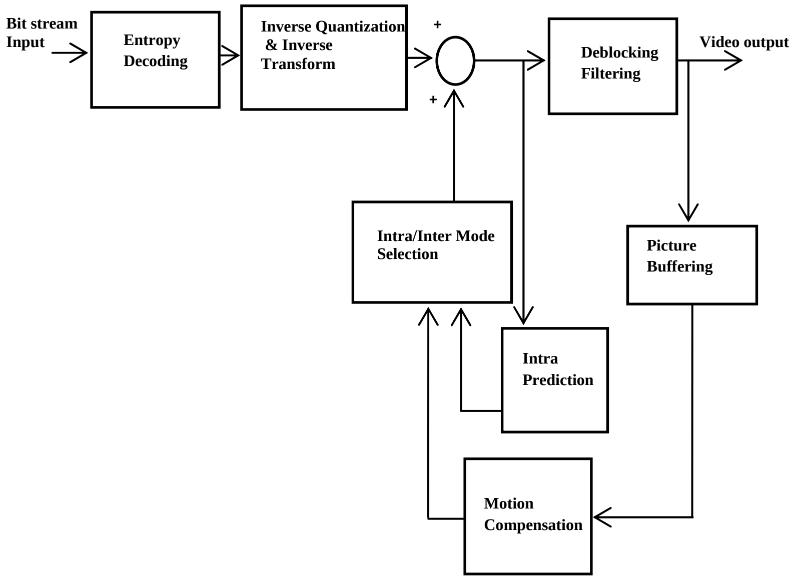


Figure 3: Block diagram of H.264/AVC Decoder

c. Transformation and Quantization

Discrete cosine transform (DCT) is a popular block based transform for image and video compression. It transforms the residual data from time domain representation to frequency domain presentation [18, 19]. Since most image and video are low frequency data, DCT can centralize the coding information.

Entropy Coding

There are two popular entropy coding methods; variable length and arithmetic coding. The H.264 standard defines two coding methods: Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC).For baseline profile, only CAVLC is employed. [20, 21]

A. Proposed Algorithm- Context Adaptive Binary Arithmetic coding (CABAC) Algorithmic View

Figures 4&5 shows the graphical representation of the CABAC decoding algorithm as per the H.264 Video Standard. System begins decoding the bitstream as it get indication of the “Start” signal from the ‘Host’ or driver where it initializes the Context Memory only before the first macro block of the slice.

If current Macro block is not the first, the system prefetches the information from the Top/Left Neighbors required by the current macro block, and stores it into the local memories of the system. After parameter initializations, the system begins decoding the various Syntax Elements from the H.264 coded bit stream in the order as specified by the standard. For 'Skip' macro block the system bypasses all decoding processes and begins decoding the next macro block [22, 23]. Once the decoding of all present syntax elements is completed for the required macro block. The macro block information is then packaged according to required output format and sent to an output buffer. This process of decoding the Syntax Elements for each macro block continues till the end of slice is reached.

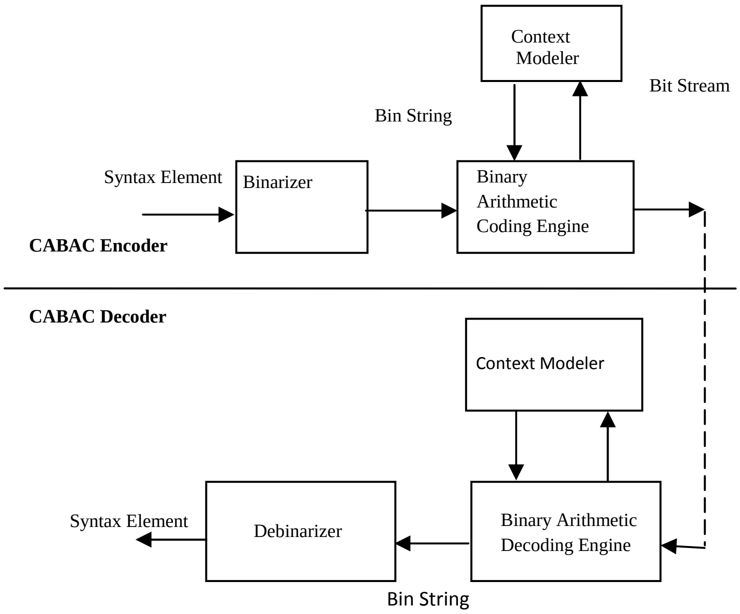


Figure 4: Block diagrams of CABAC codec of H.264/AVC

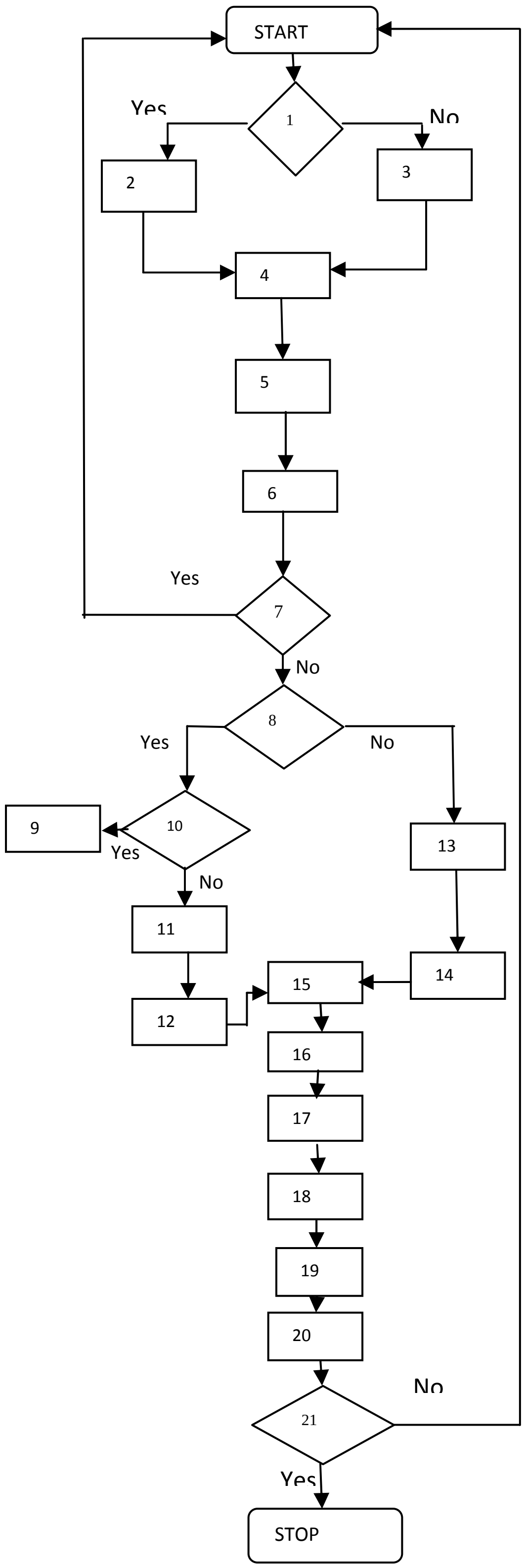


Figure 5: Proposed Algorithmic View of CABAC Decoding

Where; with reference to Fig.5

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 infromation
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

Prediction

Prediction tries to fine a reference MB that is similar to the current MB under processing so that, instead of the whole current MB, only their (hopefully small) difference needs to be coded. Depending on their reference MB comes from, prediction is classified into inter-frame prediction and intra-frame prediction. In an interpredict (P & B) mode, the reference MB is somewhere in a frame before or after the current frame, where the current MBs resides. It could also be some weighted function of MBs from multiple frames. In an intra-predict (I) mode, the reference MB is usually calculated with the mathematical functions of neighboring pixels of current MB. The difference between the current MB and its prediction is called residual error data (residual). It is transformed from spatial domain to frequency domain by means of discrete cosine transform. Because human visual system is more sensitive to low frequency image and less sensitive to high frequency image, quantization is applied such that more low information is retained while more frequency information is discarded. The third and final type of compression coding is entropy coding. A variable length coding gives shorter codes to more probable symbols and long codes to less probable ones such that the total bit count is minimized. After this phase, the output bit stream is ready for transmission or storage. There is also a decoding path in the encoder. One has to use the reconstructed frame as the reference for prediction since in the decoder side only the reconstructed frame is available. The restored residual data is obtained by performing inverse transformation. Adding the restored residual to the predicted MB, the reconstructed MB is obtained that is then inserted to the reconstructed frame. Now the reconstructed frame can be referred to either by the current I type compression or P-type or B-type prediction. Prediction exploits the spatial or temporal redundancy of a video sequence so that only the difference between the actual and predict instead of the whole image data need to be encoded.

There are two types of prediction intra prediction for I-type frame and inter prediction for P-type (predictive) and B-type (bidirectional predictive) frame[24,25].

a. Intra Prediction

There exist high similarity among neighboring blocks in video frame. Consequently, a block can be predicted from its neighboring blocks. The prediction is carried out by means of a set of mathematical functions. In H.264, an I-type 16x16, 4:2:0 MB has its luminance component (one 16x16) and chrominance (two 8x8) separately predicted. There are many ways to predict a macro block. The luminance component may be intra predicted as one single Intra 16x16 blocks or 16 intra 4x4 blocks. When using the Intra 4x4 case, each 4x4 utilizes one of nine prediction modes (one DC mode and eight directional prediction modes). When using the Intra 16x16 case, which is well suited for smooth image area, a uniform prediction is performed for the whole luminance component of a macro block. Four prediction modes are defined. Each chrominance component is predicted as a single 8x8 block using one of the four modes [26, 27].

b. Inter Prediction

High quality video sequences usually have high frame rate at 30 or 60 frames per second. Therefore, two successive frame in a video sequences are very likely to be similar. The goal of the inter prediction is to utilize this temporal redundancy to reduce data needed to be encoded. When encoding frame ***t***, we
only need to encode the difference between frame (t-1) and frame t, instead of whole frame t, this is called motion estimated inter frame prediction. In most video coding standards, the block based motion estimation (BME) is used to estimate for movement of a rectangular block from the current frame. For each MxN pixel current block in the current frame, BME compares it with some or all possible MxN candidate blocks in the search area in the reference for the best match. The reference frame may be a previous frame or the next frame in P-type coding or both in B type coding. A proper matching criteria is to measure the residual calculate by subtracting the current block from the candidate block , so that the candidate block that minimizes the residual is chosen as best match. The cost function is called sum of absolute difference (SAD), which is sum of pixel by pixel absolute difference between predicted and actual image.

There are three new features of motion estimation in H.264: variable block size, multiple reference frames and quarter pixel accuracy.

Variable block size – Block size determine tradeoff between the residual error and the number of motion estimation spends the same efforts when estimating the motion of moving objects and background (no motion). This method causes low coding efficiency. Variable block size motion estimation (VBSME) uses smaller block size for moving objects and larger block size for background, to increase the video quality and the coding efficiency. The H.264 specifies multiple block sizes starting from 16x16 going down to 4x4.

c. Proposed Algorithm

The input image is in Tag Index File (TIF) format as shown in Figure 6. The algorithm computes PSNR and Compression for different values of Quantization steps with and without Intraprediction [28].The user has the choice of entering desired Quantization step for the computation. Further, it displays the Menu for the selection of Intraprediction, and the mode of Intraprediction. After the Intraprediciton mode is selected, the prediction matrix is constructed from current frame samples that have been previously encoded [28]. A matrix P is subtracted from the present macroblock to produce remaining macroblock 'X' of 4x4 size. Transformation, Quantization and their inverses are performed on 'X'. The Prediction matrix is updated according to the mode of Intra-prediction to process the next macroblock.

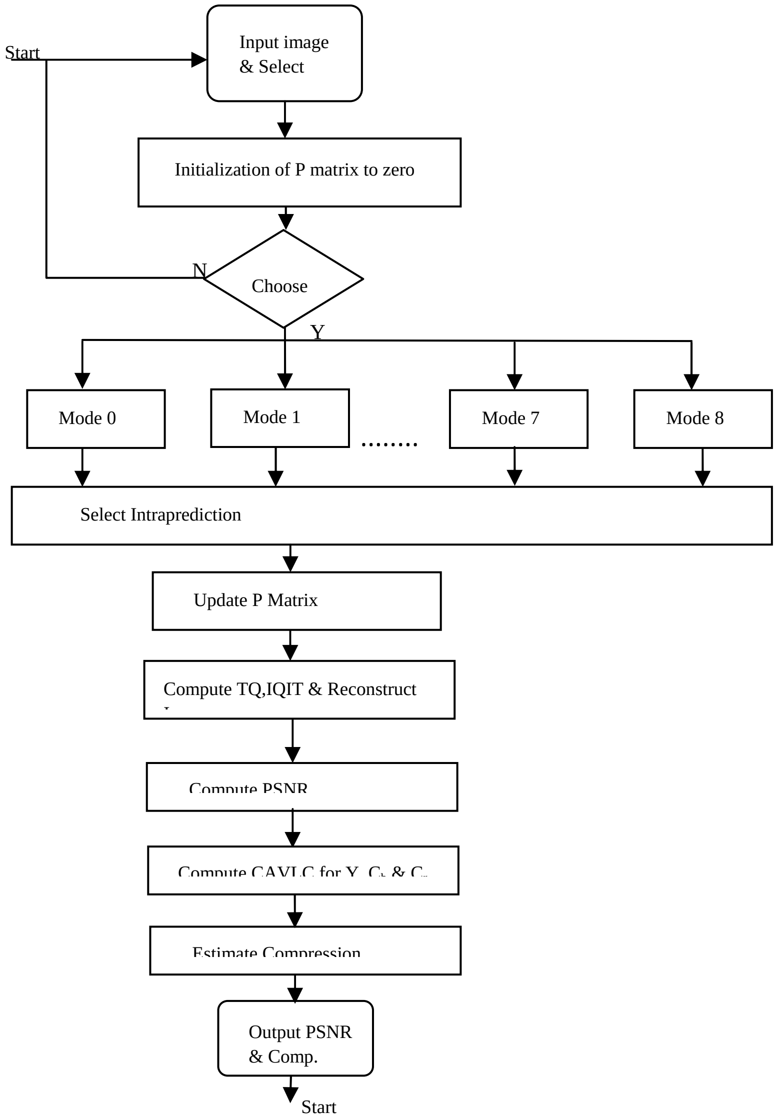


Figure 6: Algorithm of H.264/AVC Encoder Implementing all Intraprediction modes

PSNR is computed for the reconstructed image by computing root mean square error. CAVLC is applied to Y, C_b and C_r components.

Simulation, Implementation Details And Results Discussion

The software for H.263 video codec 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.

For example some of the options which are used in this work 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 16CIF, n=6 custom [12: 11 PAR]

SQCIF (128 X 96) QCIF (176 X 144) CIF (352 X 288) 4 CIF (704 X 576) 16 CIF (1408 X 1152)

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

-c-<n> rate control method

-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 following advanced options:

-E Syntax based arithmetic coding

-F Advanced prediction mode

-J Deblocking filter

-D Unrestricted motion vector

The performance are evaluated in terms of subjective quality of reconstructed video, measured in terms of peak signal to noise ratio defined in eq. 3 & 4 for different target bit rates. Mean square error (MSE) and SNR are important parameters. The Coefficient of correlation (COC) and PSNR for each frame is defined as:

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

Where MSE is Mean square error

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

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)

$$PSNR = 10\log_{10}\left(\frac{255}{MSE}\right)^2 = 20\log_{10}\left(\frac{255}{MSE}\right)(\text{for each } Y, U, V) \cdots \quad (3)$$

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

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

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

For CIF sequences Target bit rate is varied from 100 kbps to 1 Mbps at the step of 100 Kbps.

Similarly for QCIF sequence target bit rate is varied from 20
Kbps to 200 Kbps at the step of 20 Kbps.

Sequence title : Angiography
Resolution : 176x144
Number of frames : 300
Color space : YUV 4:2:0
Frames per Second : 30
Source : Uncompressed progressive

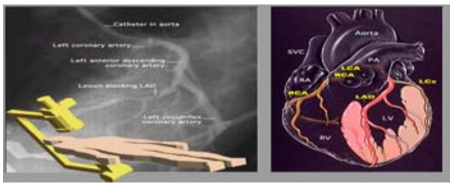


Figure 7: Snapshot of “Angiography” video sequence, frame 30

Sequence title : Acquisition sequence of rotational angiography
Resolution : High Definition (1280× 720)
Number of frames : 252

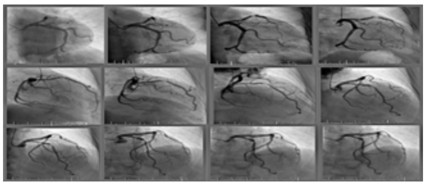


Figure 8: Snapshot of “Acquisition sequence of rotational angiography” video sequence, frame 252

Table 1: Angiography Video Sequence Simulation results for Simple, SBAC

	Simple		**SBAC**	
OBR (Kbps)	COC	PSNR (dB)	COC	PSNR (dB)
120.7	0.99111	29.87	0.99111	29.87
203.47	0.99848	32.58	0.99848	32.85
304.26	0.99971	34.31	0.99978	34.55
404.32	1.00004	35.63	1.00008	35.83
507.65	1.00029	36.71	1.00033	36.87
607.9	1.00066	37.51	1.00068	37.68
710.06	1.00108	38.19	1.00111	38.29
810.02	1.00117	38.88	1.00123	39.04
913.65	1.00132	39.55	1.00143	39.69
1018.13	1.00139	39.98	1.00151	40.07

Table 2: Angiography Video Sequence Simulation results for Simple, APM

	Simple		**APM**	
OBR (Kbps)	COC	PSNR (dB)	COC	PSNR (dB)
120.7	0.99111	29.87	0.99113	30.16
203.47	0.99848	32.58	0.99858	32.78
304.26	0.99971	34.31	0.99979	34.6
404.32	1.00004	35.63	1.00011	35.92
507.65	1.00029	36.71	1.0005	37.05
607.9	1.00066	37.51	1.00076	37.81
710.06	1.00108	38.19	1.00114	38.37
810.02	1.00117	38.88	1.00128	39.21
913.65	1.00132	39.55	1.00147	39.77
1018.13	1.00139	39.98	1.00154	40.19

Table 3: Angiography Video Sequence Simulation results for Simple, UMVM

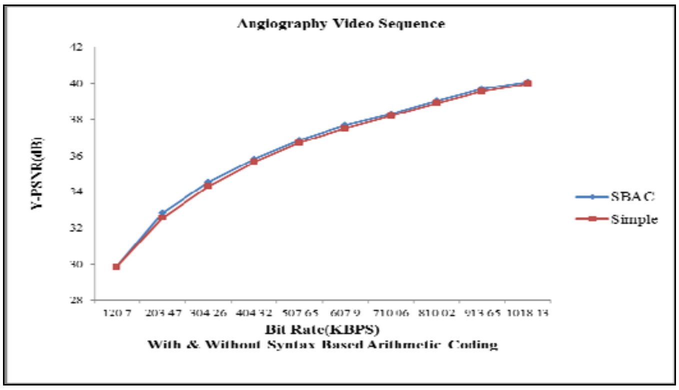
	Simple		**UMVM**	
OBR (Kbps)	COC	PSNR (dB)	COC	PSNR (dB)
120.7	0.99111	29.87	0.99114	29.96
203.47	0.99848	32.58	0.99851	32.64
304.26	0.99971	34.31	0.99972	34.39
404.32	1.00004	35.63	1.00003	35.67
507.65	1.00029	36.71	1.00031	36.76
607.9	1.00066	37.51	1.00067	37.54
710.06	1.00108	38.19	1.0011	38.23
810.02	1.00117	38.88	1.00119	38.91
913.65	1.00132	39.55	1.00134	39.58
1018.13	1.00139	39.98	1.00139	39.98

Table 4: Simulation Results of Images: PSNR and Compression Achieved with Intraprediction

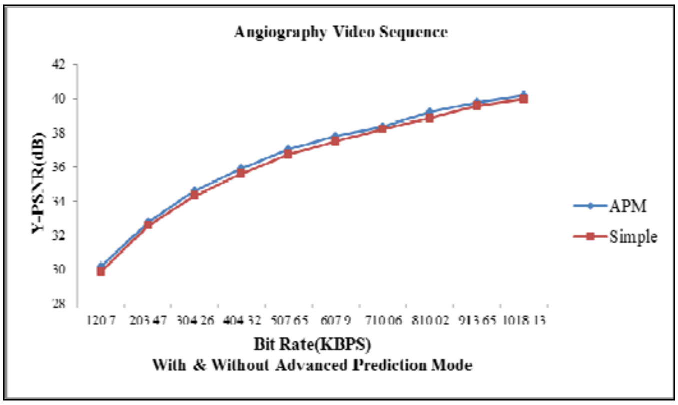
Intra Pre-diction Mode	Blue hills 800 X 600 pixels		Clock Geneva 1024 X 768 pixels		Stefan 512 X 512 pixels	
QP = 16	PSNR (dB)	Compre-ssion Achieved	PSNR (dB)	Compre-ssion Achieved	PSNR (dB)	Compre-ssion Achieved
0	34.1	11.7	39.1	29.6	35.7	15.8
1	34.2	12.1	38.6	29.4	35.2	15.1
2	34.1	7.6	38.9	25.8	35.4	10.7
3	34.3	10.8	39.8	27.4	35	14.2
4	32.6	8.4	37	19.4	33.5	10.9
5	32.8	8.6	36.1	19	34	11.7
6	33.1	9.1	38.2	23.4	34.1	12
7	34.5	11.1	40	28.4	35.6	14.3
8	34.6	11	40.4	29.7	36.9	15.9
Without IntraPre-diction	34.8	7.9	40.3	27.8	37.1	11.9

Table 5: Acquisition sequence of rotational angiography Video Sequence Simulation results for CABAC

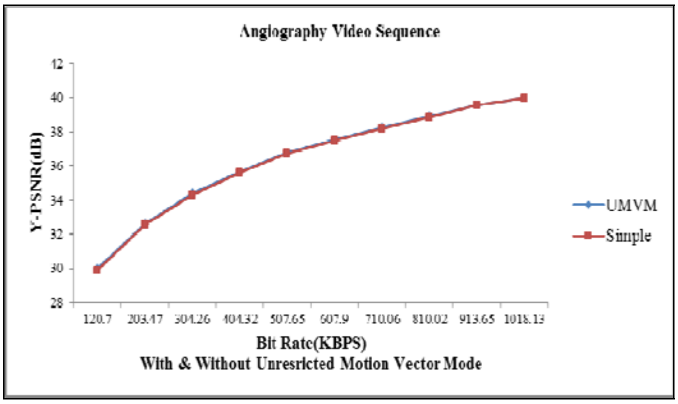
Data Rate (Mbps)	COC	PSNR(dB)	COC	PSNR(dB)
	Base Profile		**CABAC Enabled**	
0	0.99787	31.9	0.99851	32.3
0.2	1.00062	37.3	1.00072	37.6
0.5	1.00212	42	1.00274	42.4
0.8	1.00532	44.5	1.0062	45
1	1.0078	45.9	1.0099	46.5
1.2	1.0123	46.9	1.0165	47.4
1.5	1.0148	47.9	1.0167	48.5
1.8	1.0173	48.8	1.0199	49.4
2	1.0197	49.7	1.0201	50.2
2.2	1.0211	50.5	1.0229	50.9
2.5	1.0234	51.1	1.0261	51.6



(a) SBAC



(b) APM



(c) UMVM

Figure 9: Comparative Performance of H.263 Video Coder for Angiography video sequence with and without (a) SBAC (b) APM (c) UMVM

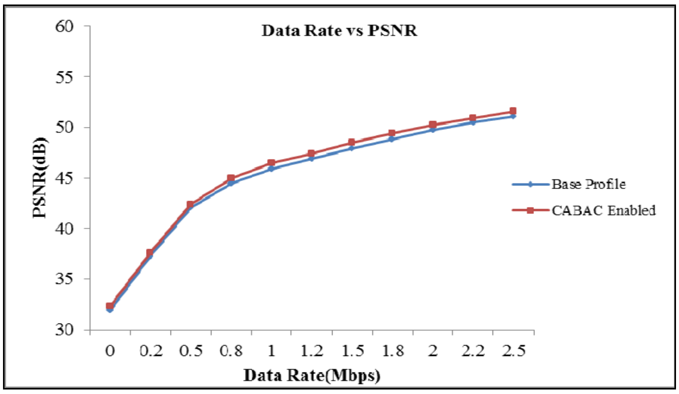


Figure 10: Comparative Performance of H.264 Video Coder for Acquisition sequence of rotational angiography video sequence with base profile and CABAC Enabled

(a) Original Image, 512x512 pixels

(b) Without Intraprediction PSNR = 37.1 dB Compression = 11.9 pixels

(c) With Horizontal Intraprediction PSNR = 35. 2 dB Compression = 15.1

(d) With Vertical Intra prediction PSNR=35.7dB Compression = 15. 8

(e) With Horizontal Up Intraprediction PSNR=36.9dB Compression=15. 9

Figure 11: Reconstructed Stefan Image Without and With Intraprediction

Conclusion

The performance of H.263 codec is with Syntax based arithmetic coding, Advanced prediction mode, Unrestricted motion vector mode etc. The following results are obtained:

1. The improvement in video quality with the use of Syntax based arithmetic coding is of the order of 0.02-0.5 dB
2. The improvement in video quality with the use of Advance prediction mode of the order of 0.01-0.8 dB
3. Unrestricted motion vector mode has almost the same performance as restricted motion vector mode the improvement of 0.1 to 0.2 dB because of the computational complexity.

With the proposed algorithm of CABAC improvement in PSNR is upto.6dB or more. The simulation results for various resolutions of pictures show better compression of about 25% when compared to the Codec without Intraprediction, without sacrificing on the quality of the reconstructed picture (PSNR achieved is of about 34 dB to 40 dB). Also algorithm of CABAC and Intraprediction are proposed for H.264 video codec.

References

[1] “Video coding for low bit rate communication ITU-T, Recommendation H.263, ver.1, 1995.

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

[3] 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.

[4] http://spmg.ece.ubc.co/h.263pleas/h.263 plus.html

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

[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] Brown, M., D. Bushmitch, K. Kerpez, D. Waring & Y. Wang, (2009) "Low-bit rate video codec parameter evaluation and optimization," in Military Communications Conference, MILCOM 2009, IEEE, pp. 1-20.
[8] www.mobile.ecs.soton.ac.uk/peter/h263/h263.html.
[9] A. Garzelli A. Andreadis, G. Benelli and S. Susini, \FEC coding for H.263 compatible video transmission," in International Conference on Image Processing, Santa Barbara, CA, Oct. 1997.
[10] M. Ghanbari," Video Coding: an introduction to standard codes;"IEE press, London 1999.
[11] G. G. Langdon, "An introduction to arithmetic coding", IBM Journal of Research and Development, Vol. 28, No. 2, pp.135-149 Mar. 1984.
[12] 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.
[13] P. Topiwala, G. sullivan, A.Joch, and F. Kossentini, "Performance evaluation of H.26L, TML8 vs. H.263++ and MPEG-4,"ITU-T Q.6/SG6 (VCEG) Tech. Rep. N8, sept.2001.
[14] http:\\www.ubvideo.com
[15] 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.
[16] T.T. Shih, C.L. Yang, Y.S. Tung, "Workload Characterization of the H.264/AVC Decoder," Proc. 5th IEEE Pacific-Rim Conf. Multimedia, Japan, November 2004, pp. 957-966.
[17] V. Lappalainen, A. Hallapuro, T. D. Hämäläinen, "Complexity of Optimized H.26L Video Decoder Implementation," IEEE Trans. Circ and Syst. for Video Technol., vol. 13, pp. 717-725, July 2003.
[18] http://www.vcodex.com/h264.html - Overview of H.264
[19] ---"Video Coding for Very Low Bit Rate Communication", ITU-T Recommendation H.264, May 2003.
[20] 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.
[21] D. Marpe, H. Schwarz, and T. Wiegand, "Context-adaptive binary arithmetic coding for H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 620–636, Jul. 2003.
[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).
[23] Won-jinkim, koelcho and ki-seokchung " Multi-Thread syntax element portioning for parallel entropy decoding" Department of Electronic and

Computer Engineering International conference,Hayang university, seoul, korea (2011).

[24] D. Marpe, T. Wiegand and G. J. Sullivan, "The H.264/MPEG4 Advanced Video Coding standard and its applications", IEEE Communications Magazine, vol.44, no.8, pp.134-144, Aug.2006.

[25] C. W. Ku, C. C. Cheng, G. S. Yu, M. C. Tsai, and T. S. Chang. "A High-Definition H.264/AVC Intra-Fame Codec IP for Digital Video and Still Camera Applications". In IEEE Transactions on Circuit and Systems for Video Technology (CSVT), vol. 16, no. 8, pp. 917–928, 2006.

[26] E. Sahin, and I. Hamzaoglu. "An Efficient Intra Prediction Hardware Architecture for H.264 Video Decoding". In Euromicro Conference on Digital System Design Architectures, Methods and Tool, pp. 448–454, 2007.

[27] M. Shafique, L. Bauer, and J. Henkel. "A Parallel Approach for High Performance Hardware Design of Intra Prediction in H.264/AVC Video Codec". In Design, Automation and Test in Europe Conference (DATE),pp. 1434–1439, 2009.

[28] Manjanaik.N, Manjunath.R " Intra Frame Coding for Advanced Video Coding Standard to Control PSNR and Bitrate Using Gaussian Pulse" International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-2, Issue-5, March 2014.