Intra/inter algorithm for B frame processing in H.264/AVC encoder

Zoran M. Milicevic and Zoran S. Bojkovic

Abstract— The H.264/AVC video coding standard aims to enable significantly improved compression performance compared to all existing video coding standards. In order to achieve this, a robust rate-distortion optimization (RDO) technique is employed to select best coding mode. This paper presents combined (intra and inter prediction) fast mode decision algorithm for H.264/AVC. The objective is to reduce the computational complexity of the encoder without significant rate-distortion performance degradation. The choose method reduces the computational time savings for B picture up to 42 %, when using modification H.264/AVC encoder with negligible loss of rate-distortion performance.

Keywords—Low complexity inter skip prediction, Rate-distortion optimization, Lagrange multiplier, Selective intra coding.

I. Introduction

H.264/AVC, the result of the collaboration between the ISO/IEC Moving Picture Experts Group and the ITU-T Video Coding Experts Group, is the latest standard for video coding [1]-[3]. The goals of this standardization effort were enhanced compression efficiency, network friendly video representation for interactive (video telephony) and non-interactive applications (broadcast, streaming, storage, video on demand). The H.264/AVC video coding standard can deliver significantly improved compression efficiency compared with previous standards. Due to this improved compression efficiency and increased flexibility of coding and transmission, H.264 has the potential to enable new services over different networks. However, the performance gains of H.264 come at a price of increased computational complexity.

The H.264/AVC video coding standard has the same basic functional elements as previous standards (MPEG-1, MPEG-2, MPEG-4 part 2, H.261, H.263) [4]-[6], i.e., transform for reduction of spatial correlation, quantization for bit rate control, motion compensated prediction for reduction of temporal correlation and entropy encoding for reduction of statistical correlation. However, in order to fulfill better coding performance, the important changes in H.264 occur in the details of each functional element by including intra-picture prediction, a new 4x4 integer transform, multiple reference pictures, variable block sizes, a quarter pel precision for motion compensation, a deblocking filter [7], and improved entropy coding (Context Adaptive Binary Arithmetic Coding (CABAC) and Context – Based Adaptive Variable Length Coding (CAVLC)) [8], [9]. Improved coding efficiency comes at the expense of added complexity to the coder/decoder [10]. H.264 utilizes some methods to reduce the implementation complexity [11].

Also, it is important to point out that the concept of B pictures generalization was realized through B frame processing. Until now the problem of B frames processing was not enough treated in H.264/AVC standard.

This paper is organized as follows. Section 2 represents our motivation for intra/inter B frame processing in H.264/AVC encoder. The importance of intra and inter prediction is presented in Section 3. The choose algorithm is described in Section 4. Experimental results followed by the discussion are shown in section 5. Section 6 concludes the work.

II. MOTIVATION

The important part of the H.264/AVC video coding standard is the analysis of computational complexity as a basis for further research effort in order to reduce it complexity.

Fig. 1 shows the run-time percentages of several major functional modules. As it can be seen, transform for cost generation (SATD computation) and mode decision take the largest portion of computation, and intra predictor generation is the second. These two functions take 77% of computation and obviously are the processing bottleneck of the H.264/AVC intra frame coder [12]. This result is quite reasonable because intra prediction has to generate 13 kinds of different predictors for each-luma sample and 4 kinds for each chroma sample. Also, the residues of each prediction mode also need to be 2-D Hadamard transformed into cost value for accumulation. After mode decision, only the decided modes and its corresponding residues are processed by DCT/Hadamard transform, quantization, inverse quantization, inverse DCT/Hadamard transform, and entropy coding. Exp-Golomb VLC and CAVLC are bit-level processing with complex controlling, but their computational load is not very large. Note that the run-time percentage of CAVLC will vary according to different QP values.

Zoran M. Milicevic is with the Telecommunication and IT Department GS of SAF, 2 Raska, Belgrade, Republic of Serbia (e-mail: mmilicko@eunet.yu).

Zoran S. Bojkovic is with University of Belgrade, 38 Nikola Tesla Boulevard, Belgrade, Republic of Serbia (e-mail: zbojkovic@yahoo.com).

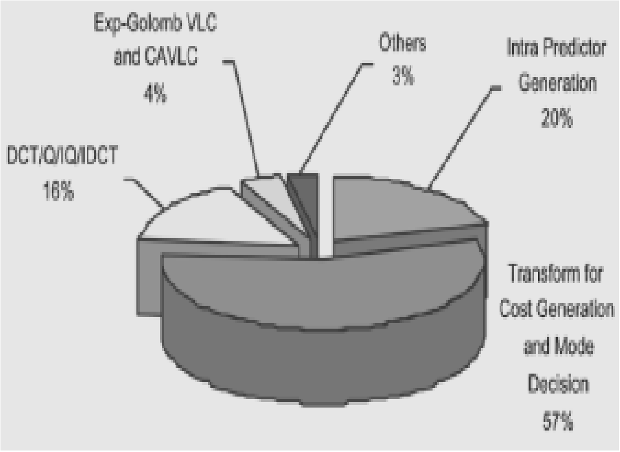


Fig. 1. Run-time percentages of various functional modules [12].

A comparative view of the computational requirements of MPEG-2 and H.264 broken down by the steps involved is shown in Fig. 2. The processing requirements are in terms of a high-end PC CPU available today. The bar drawn in the image shows the point at which the CPU is 100% loaded [13].

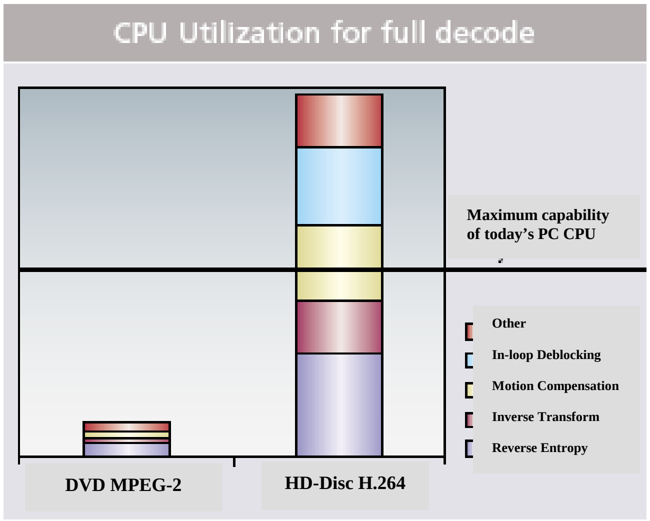


Fig. 2. Comparative view of the computational requirements of MPEG-2 and H.264 [13].

III. INTER AND INTRA PREDICTION

The intra prediction reduces spatial redundancies by exploiting the spatial correlation between adjacent blocks in a given picture. Each picture is divided into 16×16 pixel macroblock and each macroblock is formed with luma components and chroma components. For luma components, the 16×16 pixel macroblock can be partitioned into sixteen 4×4 blocks. The chroma components are predicted by 8×8 blocks with a similar prediction technique as the 16×16 luma prediction. There are 9 prediction modes for 4×4 luma blocks and 4 prediction modes for 16×16 luma blocks. For the chroma components, there are 4 prediction modes that are applied to the two 8×8 chroma blocks (U and V) [14].

Inter-prediction is to reduce the temporal correlation with help of motion estimation and compensation. In H.264/AVC, the current picture can be partitioned into the macroblocks or the smaller blocks. A macroblok of 16x16 luma samples can be partitioned into smaller blocks sizes up to 4x4. For 16x16 macroblock mode, there are 4 cases: 16x16, 16x8, 8x16 or 8x8, also four cases: 8x8, 8x4, 4x8 or 4x4 for 8x8 mode. The smaller blocks size requires larger number of bits to signal the motion vectors and extra data of the type of partition, however the motion compensated residual data can be reduced. Therefore, the choice of partition size depends on the input video characteristics. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas. Each partition or sub-macroblock partition in an inter-coded macroblock is predicted from an area of the same size in a reference picture. The offset between the two areas (the motion vector) has quarter-sample resolution for the luma component and one-eight-sample resolution for the chroma components. The process for inter-prediction of a sample block can also involve the selection of the pictures to be used as the reference pictures from a number of stored previously decoded pictures. Reference pictures for motion compensation are stored in the picture buffer. They are classified as “short-term”and “long-term” reference pictures. Memory management is required to take care of making some stored pictures as “unused” and deciding which pictures to delete from the buffer for efficient memory management [14].

IV. THE INTRA- AND INTER- MODE SELECTION ALGORITHM

To find the best coding parameters for each macroblock, H.264/AVC reference software encodes all possible combination of parameters and calculates the rate and distortion of a given macroblock for each combination. This means that the encoder calculates the R-D costs of all possible coding options and chooses the coding mode of a given macroblocks which has minimum R-D cost [15].

The modification software version with algorithm applied reduces computational processing through early identification of macroblocks to be skipped [16]. Prior to coding each macroblock, the encoder estimates the rate-distortion cost of coding or skipping the macroblock. Based on these estimates, the macroblock is either coded as normal or skipped. Skip prediction model aims to reduce computation whilst maintaining or improving rate-distortion performance.

To generate the problem we have implemented one known skip prediction algorithm [17], [18] for complex calculations. Our research includes:

- Main profile with correspondence tools,
- The skip algorithm only for B pictures,
- Rate distortion optimization mode selection disabled.

We used the skip algorithm only for B pictures because, in comparison to prior video-coding standards, the concept of B pictures is generalized in H.264/AVC. Generally, B pictures utilize two distinct reference picture buffers, which are referred to as the *first* and *second* reference picture buffer, respectively. In B pictures, four different types of inter-picture prediction are supported: list 0, list 1, bi-predictive, and direct prediction. If no prediction error signal is transmitted for a direct macroblock mode, it is also referred to as *B slice SKIP mode* and can be coded very efficiently, similar to the SKIP mode in P slices [19].

Taking into account previous discussion, the goal of a rate-distortion optimized mode selection algorithm is to find the coding mode (out of K available coding modes) that minimizes distortion subject to a rate constraint. Let $X=(X_1, X_2,\ldots, X_N)$ denote a group of N macroblocks. For a vector of N macroblock coding mode allocations $M = (M_1, M_2,\ldots, M_N)$ and a rate constraint R_T, this optimization problem can be expressed as:

$$M^* = \arg\min_{M} D(X,M) \tag{1}$$

subject to

$$R(X,M) \le R_T$$

where M* is a vector of optimal mode allocations, $D(X,M)$ is a distortion metric calculated for the group of coded and decoded macroblocks X, $R(X, M)$ is the coded rate of the group of macrobloks X. This may be written as an unconstrained problem using a Lagrange multiplier method as:

$$M^* = \arg\min_{M} \sum_{i=1}^{N} J(X_i) \tag{2}$$

where $J(Xi)$, is the rate-distortion cost for macroblock i and is given as:

$$J(X_i) = D(X_i, M) + \lambda R(X_i, M) \tag{3}$$

The Lagrange multiplier λ is chosen, considering all N macroblocks, such that the rate constraint holds [20].

This is typically a complex problem due to the interdependencies between macroblocks. The complexity can be reduced by assuming macroblock independence [20], typically leading to a sub-optimal solution:

$$M^* = \sum_{i=1}^{N} \arg\min_{M_i} J(X_i) \tag{4}$$

where:

$$J(X_i) = D(X_i, M_i) + \lambda R(X_i, M_i) \tag{5}$$

Let M_i be the coding mode (one of K possible modes) chosen by the encoder for macroblock X_i and let $M_i=K$ denotes the “skip” mode. The rate-distortion cost of coding a macroblok is given in (5) whereas costs of skipping a macroblock is:

$$J(X_i, K) = D(X_i, K) \tag{6}$$

respectively, where λ is a weighting parameter (Lagrange multiplier). Note that the rate associated with a skipped macroblock is effectively zero. Macroblock X_i should be skipped (not coded) if:

$$D(X_i, M_i) + \lambda R(X_i, M_i) \ge D(X_i, K) \tag{7}$$

The skip prediction algorithm proceeds as follows [18]:

- For every macroblock, calculate $D^n(X_i, K)$ and read previously stored values $D^{(n-1)}\left(X_i^{(n-1)}, M_i^{(n-1)}\right)$ and $R^{(n-1)}\left(X_i^{(n-1)}, M_i^{(n-1)}\right)$
- Calculate the activity factor for the current macroblock:

$$F_i = D^{(n-1)}\left(X_i^{(n-1)}, M_i^{(n-1)}\right) \cdot R^{(n-1)}\left(X_i^{(n-1)}, M_i^{(n-1)}\right) \tag{8}$$

- Calculate $\hat{\lambda}$ using equation (by substituting F_i for F):

$$\hat{\lambda} = \left(7.374\cdot10^{-8}F + 5.239\cdot10^{-5}\right) \times \exp\left[\left(-3.688\cdot10^{-5}F + 0.3203\right)\cdot QP\right] \tag{9}$$

- Choose “skip” mode if the following expression is true:

$$D^{(n-1)}\left(X_i^{(n-1)}, M_i^{(n-1)}\right) + 0.5\hat{\lambda} R^{(n-1)}\left(X_i^{(n-1)}, M_i^{(n-1)}\right) \ge D^{(n)}(X_i, K) \tag{10}$$

- If “skip” mode is chosen, no further processing is carried out and the macroblock is marked as “skipped”. If “code” mode is chosen, process the macroblock as normal.

Selective intra prediction mode decision schemes [21] for fast intra mode decision stems from the fact that the dominating direction of a bigger block is similar to that of smaller block. The best prediction mode of 4×4 luma block within 16×16 block has the same direction as that of 16×16 luma block. The computation of the intra prediction and the chroma prediction can be reduced on the base of the overall edge information from the 16×16 intra prediction result.

The number of candidate modes from selective intra prediction mode decision is taken to be as shown in Table 1. The number of mode combinations for an MB is only 1× (4×16 +4) = 68 at the best case, whereas the current RDO calculation in H.264/AVC requires 592.

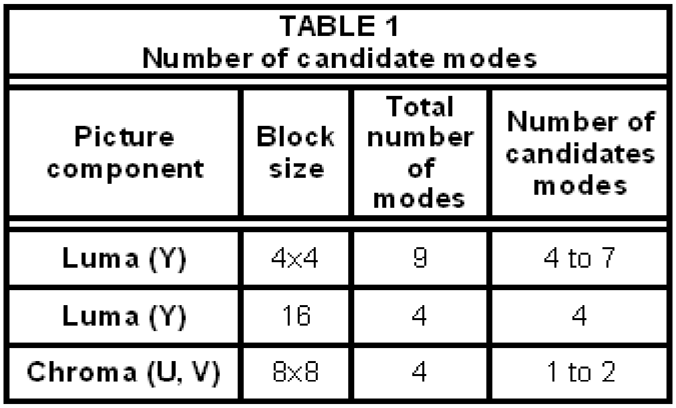


TABLE 1
Number of candidate modes

Picture component	Block size	Total number of modes	Number of candidates modes
Luma (Y)	4×4	9	4 to 7
Luma (Y)	16	4	4
Chroma (U, V)	8×8	4	1 to 2

V. EXPERIMENTAL RESULTS AND DISCUSSION

Our experiment environment is based on modification of H.264 reference encoder of JM 10.2 (modification version - JM10.2M), which was developed by JVT [22]. Experimental results are tested with the conditions indicated in the Table 2, specifying and covering 10 video sequences of the different activity, resolutions (QCIF and CIF). Also, we used for test 4:2:0 Main profile, suitable frame rate/frame skipping and quantization parameter. All tests in the experiment are run on the Pentium Intel 2.53 GHz, with 512MB RAM and the OS

Microsoft Windows XP.

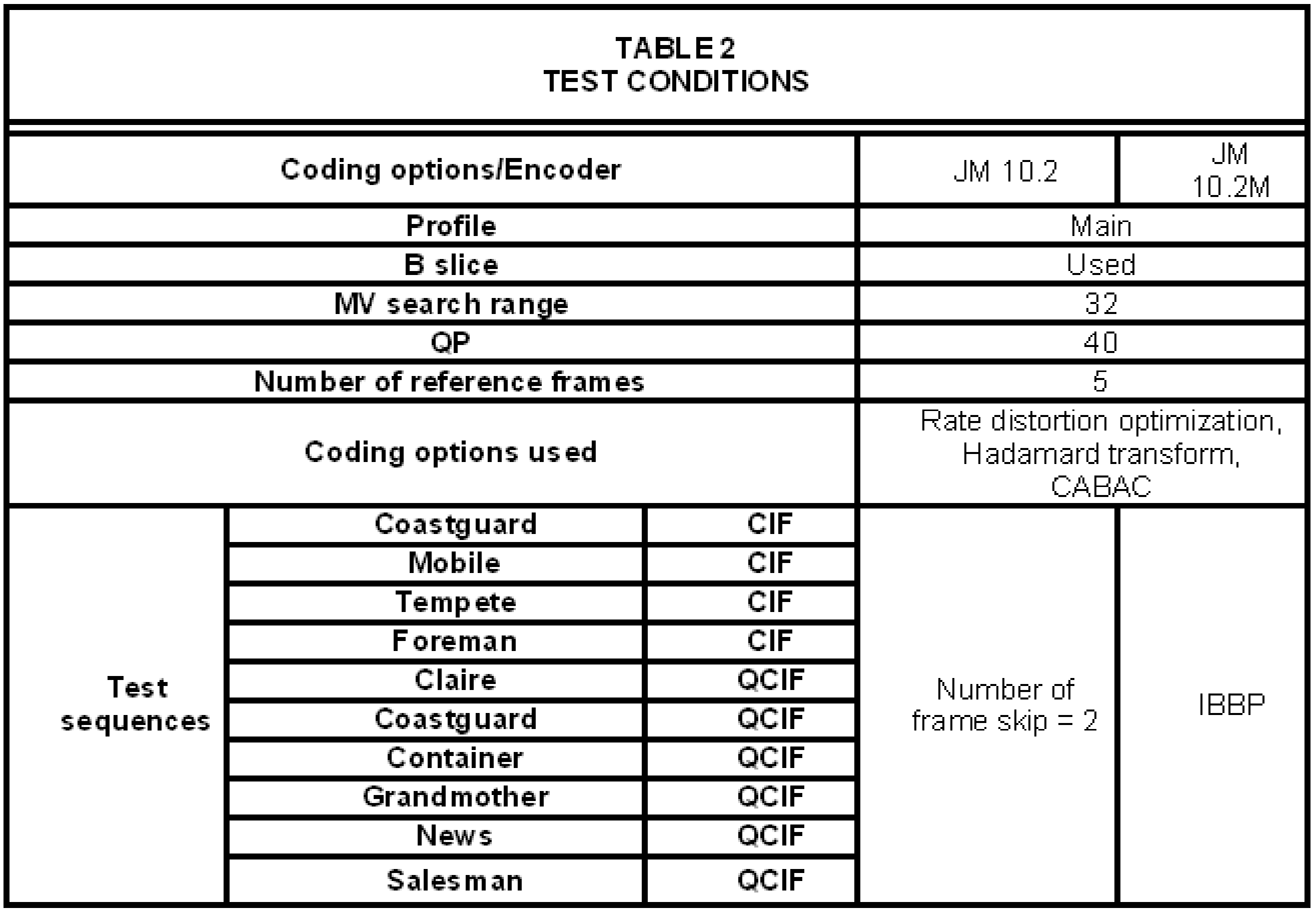


TABLE 2
TEST CONDITIONS

Coding options/Encoder			JM 10.2	JM 10.2M
Profile			Main	
B slice			Used	
MV search range			32	
QP			40	
Number of reference frames			5	
Coding options used			Rate distortion optimization, Hadamard transform, CABAC	
Test sequences	Coastguard	CIF	Number of frame skip = 2	IBBP
	Mobile	CIF		
	Tempete	CIF		
	Foreman	CIF		
	Claire	QCIF		
	Coastguard	QCIF		
	Container	QCIF		
	Grandmother	QCIF		
	News	QCIF		
	Salesman	QCIF		

For the experiments, we use the first 50 frames of the different test sequences. The test sequences have been selected to emphasize different kind of motions and contents, such as low-to-high amount of movement, camera zooming and panning motion and context with complicated texture. We used for test 4 sequences in CIF format (Foreman, Coastguard, Mobile, Tempete) and 6 sequences in QCIF (Claire, Coastguard, Container, Grandmother, News, Salesman).

Our idea is to perform tests and compare different test versions in order to show which improvements are obtained by JM 10.2M.

In order to compare and to analyze the output results we choice the following the key factors: signal - to – noise ratio (SNR) for luma (Y) picture component, the bit rate (kbps) and the computational time (ms).

We measured SNR only for Y because human visual system is better sensitive to luma then to chroma components of pictures.

Starting from the fact that video coding standard H.264/AVC holds two methods of entropy coding, i. e., CABAC and CAVLC; we have used CABAC because this method gives better results than CAVLC in the sense of video context coding [2].

Also, we have applied Hadamard transformation because it improves the encoder performance comparing to other transformations [2].

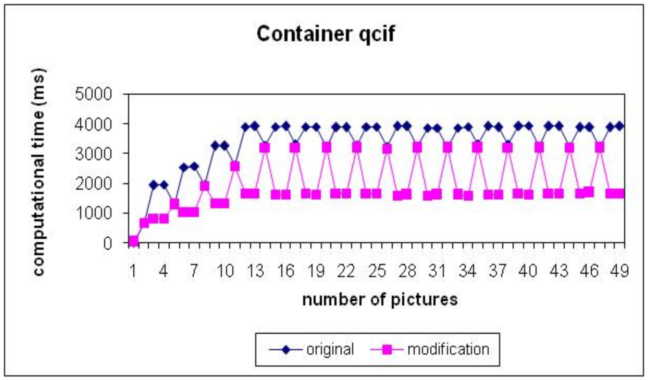
In our experiment a number of CIF and QCIF test sequences were encoded using the original encoder. The same sequences were encoded using a modified version of encoder, incorporating skip prediction algorithm and selective intra coding algorithm.

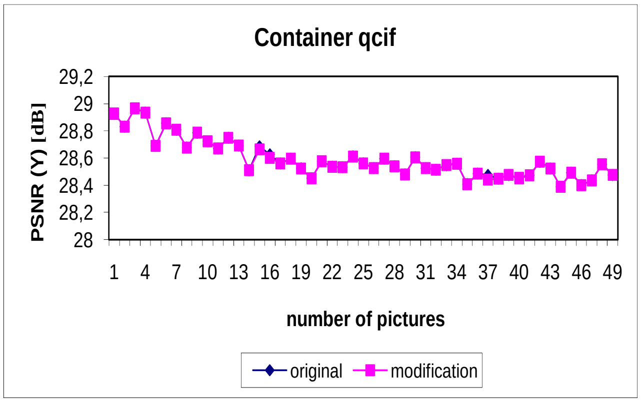
TABLE 3
TEST RESULTS

Test sequences	Computational time (%)	SNR (dB)	Bit rate (%)
Coastguard	37,29	-0,02	-6,11
Mobile	38,62	-0,06	-7,45
Tempete	36,05	-0,05	-7,36
Foreman	37,35	-0,22	-6,84
Claire	35,41	-0,03	-0,24
Coastguard	38,01	-0,04	-2,87
Container	42,26	0,00	0,53
Grandmother	39,88	0,00	-0,89
News	39,00	-0,07	-2,52
Salesman	34,53	-0,04	-1,71
Average	**37,84**	**-0,05**	**-3,55**

Table 3 shows experimental results when both test software version are used. Table 3 shows the performance of the combined method (skip mode prediction and selective intra prediction decision) in B slices in the IBBP structure. When the number of reference frames is 5, the proposed method gives encoding time saving from about 35% to over 42 %. The encoding time in average is reduced by about 38 %. This means that the modified H.264/AVC encoder is faster than reference software JM 10.2.

Fig. 3 shows computational time reduction (curve with cycle dots marks original, whereas curve with square dots marks modification computational time in relations with number of pictures) for Container test sequence. This sequence was chosen because it gives the best performance of proposed scheme in sense of computational saving time.





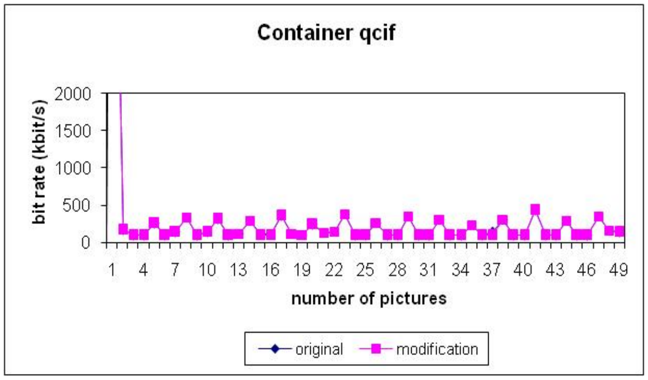


Fig. 3. Computational time, PSNR (Y), bit rate reduction for Container (QCIF) test sequence.

However, there is very negligible loss in term SNR: it is only -0,05 (dB) in average. On the other hand, our values concerned the bit rate are slightly increased (from 0.53 to -7,45 %).

VI. CONCLUSION

This paper compares the two test models (original and modification) for combined skip and inter prediction method in H.264 encoder, when B pictures are analyzed. The rate-distortion costs of coding or skipping a macroblock are estimated prior to processing. A decision whether to code the macroblock or stop further processing is made based on a Lagrangian cost function. Also, selective intra fast mode decision in encoders is proposed by reducing the number of candidate modes using the directional information.

Our experimental results indicate that coding time is reduced by 35-42 % through early identification of macroblocks that are likely to be skipped during the coding process and through reducing the number of candidate modes. There is not significant loss of rate-distortion performance. Coding time is substantially reduced because a significant number of macroblocks are not processed by the encoder. The computational saving depends on the activity of video sequences.

REFERENCES

[1] T. Weigand, G. Sullivan, G. Bjontegaard and A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” *IEEE Transactions on*

Circuits and System for Video Technology, vol.13, no.7, pp.560-576, July 2003.

[2] A. Puri, H. Chen, A. Luthra, “Video Coding using the H.264/MPEG-4 AVC compression standard,” *Signal Processing: Image Communication*, issue 9, vol. 19, pp.793-846, October 2004.

[3] Iain E. G. Richardson, *H.264 and MPEG-4 video compression video coding for next-generation multimedia*, John Viley & Sons Ltd., The Atrium, Southern Gate, Chichester, West Sussex PO 19 8SQ, England, 2003.

[4] K.R.Rao, Z.S.Bojkovic, D.A.Milovanovic, *Multimedia communication systems: techniques, standards and networks*, Prentice Hall PTR, 2002.

[5] K.R.Rao, J.J.Hwang, Techniques and Standards for images, video and audio coding, Prentice Hall PTR, 1996.

[6] T. Stockhammer, M. Hannuksela and S. Wenger, “H.26L/JVT coding network abstraction layer and IP based transport,” *IEEE ICIP 2002*, Rochester, New York, vol. 2, pp. 485-488, September 2002.

[7] P. List, A. Joch, J. Lainema, G. Bjontegaard, M. Karczewicz, “Adaptive Deblocking Filter,” *IEEE Transactions on Circuits and System for Video Technology*, vol.13, no.7, pp. 614-619, July 2003.

[8] D. Marple, H. Schwrts, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” *IEEE Transactions on Circuits and System for Video Technology*, vol.13, no.7, pp.620-636, July 2003.

[9] G. Bjontegaard and K. Lillevold, *Context-adaptive VLC coding of coefficients*, JVT document C028, Fairfax, May 2002.

[10] J. Ribas-Corbera, P. A. Chou, and S. Regunathan: “Generalized Hypothetical Reference Decoder for H.264/AVC,” *IEEE Transactions on Circuits and System for Video Technology*, vol.13, no.7, pp. 674-687, July 2003.

[11] Z. M. Milicevic, Z. S. Bojkovic, RD optimization and skip prediction for H.264/AVC standard, *Conference EUROCON’05*, Belgrade, 22-24 November 2005.

[12] Y. W. Huang, B. Y. Hsieh, T. Ch. Chen and L. G. Clan, “Analysis fast algorithm, and VLSI architecture design for H.264/AVC intra frame coding, ” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp. 378-401, March 2005.

[13] www.ati.de/products/pdf/H264_Whitepaper.pdf

[14] S. Kwon, A. Tamhankar, K. R. Rao, “Overview of the H.264/MPEG-4 part 10”, *Journal of Visual Communication and Image Representation*, issue 2, vol. 17, pp.186-216, April 2006.

[15] I. Choi, J. Lee and B. Jeon, “Fast Coding Mode Selection With Rate-Distortion Optimization for MPEG-4 part-10 AVC/H.264,” *IEEE Transactions on Circuits and System for Video Technology*, vol. 16, no. 12, pp. 1557-1561, December 2006

[16] Z. Milicevic, Z. Bojkovic, “Computational time reduction using low complexity skip prediction for H.264/AVC standard,” *WSEAS Transactions on Information Science and Applications,* issue 1, vol. 14, pp. 59-63, January 2007.

[17] K. P. Lim, G. Sullivan, T. Wiegard, *Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods*, Document JVT-K049, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Munich, Germany, March, 2004.

[18] C. S. Kannangara, I. E. G. Richardson, M. Bystrom, J. R. Solera, Y. Zhao, A. MacLennan, and R. Cooney, “Low-complexity skip prediction for H.264 through lagrangian cost estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 202-208, February 2006.

[19] D. Marple, T. Weingard, G. J. Sullivan, “The H.264/MPEG4 Advaced Video Coding Standard and its Applications,” *IEEE Communication Magazin*e, issue 8, vol. 44, pp. 134-143, August 2006.

[20] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 1, pp. 23-50, 1998.

[21] J. S. Park, and H. J. Song, "Selective Intra Prediction Mode Decision for H.264/AVC Encoders" *Transactions on Engineering, Computing and Technology*, vol. 13, pp. 51-55, May 2006.

[22] http://bs.hhi.de/~suehring/tml/download/JM10.2.

Zoran M. Milicevic received the B. S. degree in 1996 and M. Sc. degree from the University of Belgrade, Republic of Serbia in 2003. He is currently working toward the Ph. D. degree. He is the author and co-author of more then 10 scientific journal and conference papers. His current research interest includes video and image coding compression algorithms and implementation.

Zoran S. Bojkovic received the B. S, M.Sc and Ph.D degree, all in electrical engineering, from the University of Belgrade, Republic of Serbia, in 1964, 1974 and 1978, respectively. Since 1969 he has been with the University of Belgrade, where he is a Professor of Electrical Engineering. He was a visiting professor in USA, Korea, Germany, Norway and many other countries all over the world.

Prof. Bojkovic has published more than 400 invited, regular and tutorial papers in international books, journals and conference proceedings. He is also an active reviewer and a member of scientific committees of numerous journals and conferences. He is co-authors of 4 books together with professor K.R. Rao in the area of multimedia communications published by Prentice Hall Wiley and CRC Press. His areas of interest are image and video processing as well as multimedia communication.