FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 21, no. 1, April 2008, 107-119

An Approach to Selective Intra Coding and Early Inter Skip Prediction in H.264/AVC Standard

Zoran Miličević and Zoran Bojković

Abstract: This paper presents selective intra coding and early inter skip fast mode decision algorithm for H.264/AVC and compares performance of H.264/AVC with prior standards. Video coding standard H.264/AVC provides gains in compression efficiency of up to 50% over a wide range of bit rates and video resolutions compared to previous standards. In order to achieve this, a robust rate-distortion optimization (RDO) technique is employed to select best coding mode and reference frame for each macroblock. Also, the original and modification test models are compared for combined skip and intra prediction method in H.264/AVC encoder, when B pictures are analyzed. Experimental results show that the 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.

Keywords: Early inter skip prediction, Selective intra prediction mode decision, MPEG-2, MPEG-4 part 2, Windows Media Video 9, Audio Video Coding Standard (AVS).

1 Introduction

The H.264 video coding standard has the same basic functional elements as previous standards (MPEG-1, MPEG-2, MPEG-4 part 2, H.261, H.263) [1–3], 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

---

Manuscript received on September 9, 2007.

Z. Miličević is with the Telecommunication and IT Department GS of the SAF, Raska 2, 11000 Belgrade, Serbia (e-mail: `mmilicko@eunet.yu`). Z. Bojković is with the University of Belgrade, Bulevar Nikole Tesle 38, 11000 Belgrade, Serbia (e-mail: `zbojkovic@yahoo.com`).

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, and improved entropy coding. Improved coding efficiency comes at the expense of added complexity to the coder/decoder. H.264 utilizes some methods to reduce the implementation complexity.

The improvement in coding performance comes mainly from the prediction part. Intra prediction significantly improves the coding performance of H.264/AVC intra frame coder. On the other side, inter prediction is enhanced by motion estimation with quarter-pixel accuracy [4], variable block sizes, multiple reference frames [5] and improved spatial/temporal direct mode. Unlike the previous standards, prediction must be always performed before texture coding not only for inter macroblocks but also for intra macroblocks [6].

This paper is organized as follows. The short overview of H.264/AVC standard is presented in section II. Section III explains selective intra coding and early inter skip algorithm and gives experimental results. Section IV concludes the work.

2 Overview of H.264/AVC

The coded output bitstream of H.264/AVC has two layers, Network Abstraction Layer (NAL) and Video Coding Layer (VCL) [7]. NAL abstracts the VCL data in a manner that is appropriate for conveyance on a variety of communication channels or storage media. The VCL unit contains the core video coded data, which consists of video sequence, picture, slice, macroblock and block. H.264/AVC use follow coding tools [8]:

- Intra spatial (block based) prediction: Full-macroblock (16×16) luma or chroma prediction 4 modes (directions) for prediction and 8×8 (FRExt-only) or 4×4 luma prediction 9 modes (directions) for prediction;
- Inter temporal prediction block based motion estimation and compensation: Multi reference pictures, Reference B picture, Arbitrary referencing order, Variable block sizes for motion compensation (seven block sizes: 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4, $^{1}/_{4}$ sample luma interpolation ($^{1}/_{4}$ or $^{1}/_{8}$ th-sample chroma interpolation), Weighted prediction, Frame or Field based motion estimation for interlaced scanned video;
- Interlaced coding features: Frame-field adaptation (Picture Adaptive Frame/Field (PicAFF) and MacroBlock Adaptive Frame Field (MBAFF)) and Field scan;
- Lossless representation capability: Intra PCM raw sample-value macroblocks and Entropy-coded transform-bypass lossless macroblock (FRExt-only);

- 8×8 (FRExt-only) or 4x4 Integer Inverse Transform (conceptually similar to the well-known DCT);
- Residual color transform for efficient RGB coding without conversion loss or bit expansion (FRExt-only);
- Scalar quantization;
- Encoder-specified perceptually weighted quantization scaling matrices (FRExt-only);
- Logarithmic control of quantization step size as a function of quantization control parameter;
- Deblocking filter (within motion compensation loop);
- Coefficient scanning: Zig-Zag (Frame) and Field;
- Lossless Entropy coding: Universal Variable Length Coding (UVLAC) using Exp-Golomb codes, Context Adaptive VCL (CAVLC), Context-based Adaptive Binary Arithmetic Coding (CABAC);
- Error Resilience Tools: Flexible Macroblock Ordering (FMO), Arbitrary Slice Order and Redundant Slices;
- SP and SI synchronization pictures for streaming and other uses;
- Various color spaces supported (YCbCr of various types, YCgCo, RGB, etc. especially in FRExt); 4:2:0, 4:2:2 (FRExt-only), and 4:4:4 (FRExt-only) color formats.

H.264/AVC defines a set of seven Profiles, each supporting a particular set of coding functions and each specifying what is required of an encoder or decoder that complies with the Profile. Also, profiles are defined to cover the various applications from the wireless networks to digital cinema. There are three Profiles in the first version: Baseline, Main and Extended. Baseline profile is to be applicable to real-time conversational services such as video conferencing and videophone. Suitable application for the Main Profile includes broadcast media applications such as digital television and stored digital video. Extended profile is aimed at multimedia services over Internet [7]. Also there are four High Profile defined in the Fidelity range extensions for applications such as content-contribution, content-distribution, and studio editing and post-processing: High, High 10, High 4:2:2 and High 4:4:4. Since a profile defines a set of coding tools or algorithms that can be used in generating a compliant bitstream, a level places constraints on certain key parameters of the bitstream.

In 2004 Joint Video Team (JVT) added new extensions known as the Fidelity Range Extensions (FRExt) [8], which provide a number of enhanced capabilities relative to the base specification. Specifically, these include: supporting an adaptive block-size for the residual spatial frequency transform, supporting encoder-specified perceptual-based quantization scaling matrices, and supporting efficient lossless representation of specific regions in video content.

Also, the Scalable H.264/AVC extension is applied to extend the hybrid video coding approach of H.264/AVC in a way that a wide range of spatial-temporal and quality scalability is achieved [9]. It refers to scalability as a functionality that allows the removal of parts of the bit-stream while achieving a reasonable coding efficiency of the decoded video at reduced temporal, SNR, or spatial resolution.

H.264/MPEG 4-AVC has received a great deal of recent attention from industry. Besides the classical application areas of videoconferencing and broadcasting of TV content (satellite, cable, and terrestrial), the improved compression capability of H.264/MPEG 4-AVC enables new services and thus opens new markets and opportunities for the industry. Another area that has attached a lot of near-term industry implementation interest is the transmission and storage of HD content [10]. H.264 can be applied to consumer equipment such as digital cameras, cell phones, video-over-IP networks, and high-definition digital broadcasting through terrestrial or satellite channels, and digital storage system such as high-definition DVD [7]. Today H.264/AVC enable IPTV over DSL, because the H.264/AVC standard lets telephone companies (telcos) deploy broadcast-and DVD-quality video content over their existing DSL-based IP access networks to help them effectively compete with cable and wireless operators [11].

Table 1 compares the coding algorithms of H.264 with MPEG-2 and MPEG-4 part 2. As it can be seen the advantage of using H.264/MPEG-4 part 10 is evident when coding tools are taken into account.

Other coding standards such as Windows Media Video 9 (WMV-9) [12] and AVS China [13] are briefly compared too (Table 2).

3 Selective Intra Coding and Early Inter Skip Algorithm

In the open literature the optimal mode (mode decision) for a macroblock is selected as that which produces the least RD cost. The H.264 standard employs a brute force algorithm to search through all possible block-sizes to find a motion vector for each macroblock. Thus, the computational burden of the searching process is far more demanding than any existing video coding algorithm.

A fast intra mode decision algorithm for H.264/AVC intraprediction by using local edge information is proposed in [14]. It is observed that the pixel along the direction of local edge are normally of the similar values (this is true for both luma and chroma components). A good prediction could be achieved if the pixels are

Table 1. Comparison of standards MPEG-2, MPEG-4 part 2, and H.264 / MPEG-4 part 10 [7].

Feature/Standard	MPEG-2	MPEG-4 part 2	MPEG-4 part 10/H.264
Macroblock size	16×16 (frame mode) 16×8 (field mode)	16×16	16×16
Block Size	8×8	16×16, 16×8, 8×8	16×16, 8×16, 16×8, 8×8 , 4×8, 8×4,4×4
Intra-prediction	No	Transform domain	Spatial domain
Transform	8×8 DCT	8×8 DCT/Wavelet transform	8×8, 4×4 integer DCT 4×4, 2×2 Hadamard
Quantization	Scalar quantization with step size of constant increment	Vector quantization	Scalar quantization with step sizes of increase at the rate of 12.5%
Entropy coding	VLC	VLC	VLC, CAVLC and CABAC
Pel accuracy	$^{1}/_{2}$-pel	$^{1}/_{4}$-pel	$^{1}/_{4}$ -pel
Reference picture	One picture	One picture	Multiple pictures Forward/forward
Bidirectional prediction mode	Forward/backward	forward/backward	Forward/backward forward/forward
Weighted prediction	No	No	Yes
Deblocking filter	No	No	Yes
Picture Types	I, P, B	I, P, B	I, P, B, SI, SP
Profiles	5 profiles	8 profiles	7 profiles
Playback & Random Access	Yes	Yes	Yes
Error robustness	Data partitioning FEC for important packet transmission	Synchronization, data partitioning, header extension, reversible VLCs	Data partitioning, parameter setting, flexibile macroblock ordering, redundant slice, SP and SI slices
Transmission rate	2-15Mbps	64kbps - 2Mbps	64kbps150Mbps
Encoder complexity	Medium	Medium	High
Compatible with previous standards	Yes	Yes	No

predicted using those neighboring pixels that are in the same direction of the edge. Therefore, an edge map which represents the local edge orientation and strength is created, and a local edge direction histogram is then established for each subblock. Based on the distribution of the edge direction histogram, only a small number of prediction modes are chosen for RDO calculation during intraprediction. Experimental results show that the fast mode decision algorithm increase the speed of intra coding significantly with negligible loss of the quality.

In [15] a fast intermode decision algorithm to decide the best mode in intra-coding is proposed. It makes use of the spatial homogeneity and the temporal stationarity characteristics of the video objects. Specifically, spatial homogeneity

Table 2. Comparison of WMV-9 and AVS with H.264/MPEG-4 part 10 [7].

Feature/Standard	MPEG-4 part 10/H.264	WMV-9	AVS
Prediction block size	16×16, 8×16, 16×8, 8×8, 4×8, 8×4, 4×4	16×16, 8×8	16×16,8×8
Intra-prediction	4×4, 8×8: 9 modes 16×16: 4 modes	No	8×8: 5 modes
Transform	8×8, 4×4 integer DCT 4×4, 2×2 Hadamard	8×8, 4×8, 8×4, 4×4 integer DCT	Asymmetric 8×8 integer DCT
Quantization	Scalar quantization	Dead zone, uniform scalar quantization	Scalar quantization
Entropy coding	VLC, CAVLC and CABAC	Multiple VLC tables	VLC
Sub-pel filter	$^1/_2$-pel: 6-tap, $^1/_4$-pel: 2-tap	$^1/_2$-pel: 4-tap, $^1/_4$-pel: 4-tap	$^1/_4$-pel: 4-tap, $^1/_4$-pel: 4-tap
Reference picture	Multiple pictures	One picture	Two pictures
Bidirectional prediction mode	Forward/backward forward/forward backward/backward, 2 motion vectors	Forward/backward, 2 motion vectors	Forward/backward, symmetric 1 motion vector
Weighted prediction	Yes	Yes	Yes
Deblocking filter	Yes	Yes	Yes

of a MB is decided based on the MBs edge intensity, and temporal stationarity is decided by the difference of the current MB and it collocated counterpart in the reference frame. Based on the homogeneity and stationarity of the video objects, only a small number of intermodes are selected in the the RDO process. The experimental results show that the fast intermode decision algorithm is able to reduce on the average 30% encoding time, with a negligible peak signal-to-noise ration loss of 0,03 dB or, equivalently, a bit rate increment of 0,6%.

On the other side a fast inter mode selection algorithm was proposed in [16] to alleviate the encoder complexity due to mode decision, while maintaining picture quality and other coding efficiency. The FInterms algorithm makes use of a spatial content measure to assign different levels of inter-modes to each macroblock depending on its complexity. The lower the complexity of the macroblock content, the fewer inter-modes the encoder has to check, and vice versa.

The paper [17] seeks to provide an algorithm for fast coding mode selection in H.264/AVC encoders by reducing the number of candidate modes. A rate estimation method for further reduction of Lagrangian cost calculation is presented, too. It is concluded that total computation for mode selection can be reduced signicantly if the cost calculations for less probable modes, e.g. intra modes in very low motion videos, can be skipped. Also the computation can be reduced further if

R is estimated in the cost calculation.

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 [18].

The modification software version with algorithm applied reduces computational processing through early identification of macroblocks to be skipped [19]. 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 [20] for complex calculations. Our research includes: *Main profile with correspondence tools, the skip algorithm only for B pictures and 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 [10].

Let Mi be the coding mode (one of K possible modes) chosen by the encoder for macroblock X_i and let $M_i = K$ denote the skip mode. The rate-distortion cost of coding an macroblok is given in

$$J(X_i) = D(X_i, M_i) + \lambda R(X_i, M_i) \tag{1}$$

while costs of skipping a macroblock is [20]:

$$J(X_i, K) = D(X_i K) \tag{2}$$

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

$$D(X_i, M_i) + \lambda R(X_i, M_i) \geq D(X)i, K) \tag{3}$$

$D(X_i, K)$ is the distortion between the current macroblok and the motion-compensated macroblok from the reference picture with zero displacement from the predicted vector MVp. The distortion measure is selected to be mean-squared error (MSE). $D(X_i,K)$ may be calculated prior to coding the current macroblock, i.e. its calculation does not depend on any outputs of the coding process. $D(X_i,Mi)$ is the MSE between the current macroblock and the decoded, reconstructed macroblock while $R(X_i,M_i)$ is the number of bits required to code the current macroblock using coding mode $M_i \in \{1,\ldots,K-1\}$. The actual values of $D(X_i,M_i)$ and $R(X_i,M_i)$ are not available prior to coding, and thus these parameters are estimated for each macroblock in the current frame (n) using the following models

$$\hat{D}^{(n)}(X_i^{(n)},M_i^{(n)}) = \alpha_d D^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)}) \quad (4)$$

$$\hat{R}^{(n)}(X_i^{(n)},M_i^{(n)}) = \alpha_r R^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)}) \quad (5)$$

$D^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)})$ is the MSE between the original and reconstructed macroblocks in the same position in the previous frame $(n-1)$, while $R^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)})$ is the number of bits required to code the macroblock in the same position in the previous frame and α_d and α_r are fixed parameters. The skip prediction algorithm proceeds as follows [20]:

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

$$F_i = D^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)}) \cdot R^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)}) \quad (6)$$

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

$$\hat{\lambda} = (7.734 \cdot 10^{-8}F + 5.239 \cdot 10^{-5})xe^{(-3.688 \cdot 10^{-5}F + 0.3203)QP} \quad (7)$$

Choose skip mode if the following expression is true:

$$D^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)}) + 0.5\hat{\lambda} R^{(n-1)}(X_i^{(n-1)},M_i^{(n-1)}) \geq D^{(n)}(X_i,K) \quad (8)$$

- 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.

H.264/AVC encodes the macroblock by iterating all the luma intra decisions for each possible chroma intra prediction mode for the best coding efficiency. Number of mode combinations for luma and chroma components in an macroblock is C8×(L4×16 + L16), where C8, L4, and L16 represent the number of modes for chroma prediction, 4×4 luma prediction and 16×16 prediction, respectively. It means that, for an macroblock, it has to perform 4×(9×16 + 4) = 592 different RDO calculations before a best RDO mode is determined.

The key idea of the chose *selective intra prediction mode decision* methods for fast intra mode decision stems from the fact that the dominating direction of a bigger block is similar to that of smaller block [21]. 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 9 modes luma intra predictions modes is assorted of the 4×4 block to make four candidate groups according to the directional information of the 16×16 prediction mode. Therefore the unlikely modes are filtered out prior to the RD cost computation based on the directional correlation between 16×16 luma block and 4×4 luma block.

Selective intra prediction mode decision method for 8×8 chroma intra prediction modes is proposed for the enhancement of chroma intra prediction. Even though the luma block and the chroma block are from luminance signals and chrominance signals separately, they encode the same section of image, 16×16 pixel macroblock, and share overall directional information. The number of chroma prediction modes is narrow down from 4 modes to 2 modes, according to the best prediction mode of the 16×16 luma block.

In accordance with previous, the number of candidate modes for 4×4 luma block size is decreased from 9 to 4-7 modes, while 16×16 luma block size has same value (4 modes). The number of candidate modes for 8×8 chroma block size is decreased from 4 to 1-2 modes. The number of mode combinations for an macroblock is only 1× (4×16 + 4) = 68 at the best case, whereas the current RDO calculation in H.264/AVC requires 4× (9×16 + 4) = 592.

3.1 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 follow conditions: Main profile; MV search range has value 32; Quantization parameter has value 40; Number of reference frames is 5; Hadamard transform; CABAC; IBBP format; specifying and covering 6 video sequences of the different activity, same resolutions (QCIF). All tests in the experiment are run on the Pentium Intel 2.53 GHz, with 512 MB RAM and the OS Microsoft Windows XP.

In our 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 6 sequences in QCIF format (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 our version JM 10.2M.

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

Table 3 shows experimental results when both test software version are used.

Table 3. Experimental results.

Test sequences	Computational Time (%)	SNR (dB)	Bit rate (%)
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	−38.18	0.03	1.28

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% until over 42%. The encoding time in average is reduced over 38%. This means that the modified H.264/AVC encoder is faster than reference software JM 10.2. Figure 1 shows computational time reduction (curve with romb dots marks original, while curve with square dots marks modification results for computational time depending on the number of pictures) for Container test sequence. This sequence was chosen because it gives the best performance of the

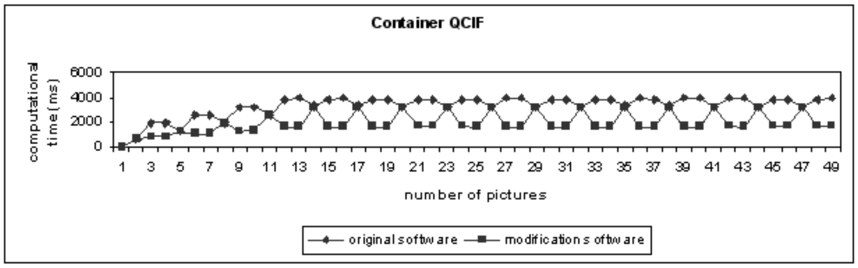

Fig. 1. Computational time for Container (QCIF) test sequence.

proposed scheme in sense of computational saving time.

However, there is very negligible loss in term SNR: it is only 0.03 (dB) in average. On the other hand, our values for the bit rate are slightly increased (from -0.53 to 2,87%).

Table 4 summarizes the simulation results for the several different fast coding algorithms compared to our the best test sequence Container in terms of the computational time, the bit rate and PSNR. It can be seen that the proposed algorithm give the better results especially in term of computational saving time. The obtained results show that our JM10.2M algorithm achieves values over 60% in computational time reduction in comparison to algorithms in [14] and [17], while that differences are less for another two algorithms [15] and [16], but still better.

Table 4. Comparison results for the several different fast coding algorithms

Sequence	Performance	Algorithm [14]	Algorithm [15]	Algorithm [16]	Algorithm [17]	JM10.2M
Container	Time (%)	−16.460	−39.05	−30.85	−15.6	−42.26
	PSNR (dB)	−0.104	−0.027	− 0.02	0.05	0.00
	Bit rate (%)	2.077	0.45	0.10	–	−0.53

4 Conclusion

This paper describes the main features of H.264/AVC standard and compare those features with features of another video coding standard (MPEG-2, MPEG-4 part 2) and especially with Windows Media Video 9 and AVS standards. Comparison is shown that H.264/AVC provides better compression, efficient coding of video contents and appropriate bit rates for equivalent perceptula quality. Also, this paper compares the two test models (original and modification) for combined skip and intra prediction method in H.264 encoder, when B pictures are analyzed. 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.

References

[1] T. Weigand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. on Circuits and System for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

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

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

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

[5] Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma, and L. G. Chen, “Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 507–522, Apr. 2006.

[6] M. Flierl and B. Girod, “Generalized B pictures and the draft JVT/H.264 video compression standard,” *IEEE Trans. on Circuits and System for Video Technology*, vol. 13, no. 7, pp. 587–597, 2003.

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

[8] G. Sullivan, P. Topiwala, and A. Luthra, “Overview of the H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions,” in *SPIE Conference on Applications of Digital Image Processing XXVII, Special session on andvanced in the new emerging standard: H.264/AVC*, Aug. 2004, pp. 53–74.

[9] in *Scalable video coding WD 1*, M. W. J.Reichel, H.Schwarz, Ed., Hong Kong, Jan. 2005, vol. Doc. VT-N020, Doc. VT-N020.

[10] D. Marple, T. Weingard, and G. J. Sullivan, “The H.264/MPEG4 advaced video coding standard and its applications,” *IEEE Communication Magazin*, vol. 44, pp. 134–143, Aug. 2006.

[11] [Online]. Available: www.envivio.com/pdf/whitepaper_H264_IPTV_Over_DSL.pdf

[12] S. Srinivasan *et al.*, “Windows media video 9: Overview and applications,” *Signal Processing: Image Communication*, vol. 19, pp. 851–875, Oct. 2004.

[13] W. Gao *et al.*, *AVS The Chinese Next-Generation Video Coding Standard*, National Association of Broadcasters Std., Apr. 2004.

[14] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, “Fast mode decision algorithm for intraprediction in H.264/AVC video coding,” *IEEE Trans. on Circuits and System for Video Technology*, vol. 15, no. 7, pp. 813–822, July 2005.

[15] D. Wu, F. Pan, K. P. Lim, S. Wuand, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, “Fast intermode decision in H.264/AVC video coding,” *IEEE Trans. on Circuits and System for Video Technology*, no. 7, pp. 953–958, July 2005.

[16] A. Yu, “Efficient block-size selection algorithm for interframe coding in H.264/MPEG-4 AVC,” in *Proc. of IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 3, Montreal, Canada, May 2004, pp. 169–172.

[17] H. Kim and Y. Altunbasak, “Low-compexity macroblok mode selection for H.264/AVC encoders,” in *Proc. of International Conference on Image Processing*, vol. 2, Oct. 2004, pp. 765–768.

[18] I. Choi, J. Lee, and B. Jeon, “Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264,” *IEEE Trans. on Circuits and System for Video Technology*, vol. 16, no. 12, pp. 1557–1561, Dec. 2006.

[19] Z. Milicević and Z. Bojković, “Computational time reduction using low complexity skip prediction for H.264/AVC standard,” *WSEAS Trans. on Information Science and Applications, Issue 1*, vol. 14, pp. 59–63, Jan. 2007.

[20] 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 Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 202–208, 2006.

[21] J. S. Park and H. J. Song, “Selective intra prediction mode decision for H.264/AVC encoders,” *Trans. on Engineering, Computing and Technology*, vol. 13, pp. 51–55, 2006.

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