Video Compression by Using H.264/MPEG-4 Advance Video Coding (AVC)

D. Lakshmi Bharathi[1], K. Prasanthi Jasmine[2]

[1]M.Tech Student, Andra Loyola Institute of Engineering & Technology, Vijayawada, Andhra Pradesh, India

[2] Associate Professor, Andra Loyola Institute of Engineering & Technology, Vijayawada, Andhra Pradesh, India

Abstract: ***This paper presents a novel approach to Video Compression in H.264/MPEG-4 Advanced Video Coding (AVC) compressed video streams for Spatial Correlation. We present a new video compression approach which tends to hard exploit the pertinent temporal redundancy in the video frames to improve compression efficiency with minimum processing complexity. It consists on a 3D to 2D transformation of the video frames that allows exploring the temporal redundancy of the video using 2D transforms and avoiding the computationally demanding motion compensation step. This transformation turns the spatial temporal correlation of the video into high spatial correlation.Indeed, this technique transforms each group of pictures to one picture eventually with high spatial correlation. Thus, the decorrelation of the resulting pictures by the DCT makes efficient energy compaction, and therefore produces a high video compression ratio. Many experimental tests had been conducted to prove the method efficiency especially in high bit rate and with slow motion video. The proposed method seems to be well suitable for video surveillance applications and for embedded video compression systems.***

Keywords: H.264, AVC, video compression, inter mode decision

1. Introduction

The objective of video coding in most video applications is to reduce the amount of video data for storing or transmission purposes without affecting the visual quality. The desired video performances depend on applications requirements, in terms of quality, disks capacity and bandwidth. For portable digital video applications, highly-integrated real-time video compression and decompression solutions are more and more required. Actually, motion estimation based encoders are the most widely used in video compression. Such encoders exploits inter frame correlation to provide more efficient compression.

Recent acceptance of H.264 as a new decoding typical is expected to have far more implications than the production of presently a new documentation. The consensus among the foremost players of the communications and video industry on H.264 might afford the major thrust for this new standard. Previous MPEG video coding principles such as MPEG-1 and MPEG-2 have enabled many familiar buyer products. For instance, these standards enabled video CD's and DVD's tolerate video playback on digital VCRs/set-top-boxes and computers. The MPEG-2 video coding standard,which was urbanized about 10 years ago primarily as a conservatory of prior MPEG-1 video capability to bear of interlaced video coding, was an enabling technology for digital television systems worldwide. It is exploit for transmission of standard definition (SD) and high definition (HD) TV signals over satellite, cable and global emission and the storage of high quality SD video signals onto DVDs. MPEG-4 was launched to address a new cohort of multimedia applications and services such as interactive TV, internet video etc. The core of the MPEG-4 usual was developed during 1995-1999,however MPEG-4 is a living usual with new parts added continuously as and when technology exists to address embryonic applications. The significant advances in core video usual were achieved on the capability of coding video objects, even as at the same time, improving coding efficiency at the expense of a reserved increase in complexity.

The usual achieves clearly higher compression efficiency, often quoted as, up to a feature of two over the MPEG-2 video standard. As one would expect, the augment in compression efficiency comes at the cost of considerable increase in complexity, often quoted as a factor of four for the decoder, whereas encoding difficulty may be as high as a factor of nine over MPEG-2. Moreover, as of flexible features or subsets of the standard, the ensuing complexity depends on the profile implemented, which is submission dependent.

2. Oveview of the H.264 Standard

In order to address the need for suppleness and customizability, the H.264 standard covers a Video Coding Layer (VCL), which is designed for capable representation of the video content, and a Network Abstraction Layer (NAL), which formats the VCL illustration of the video and provides header information in a way that is suitable for conveyance by different transport layers or storage media. Figure 1 depicts the configuration of H.264/AVC video encoder.

As in all prior ITU-T and ISO-IEC JTC1 video standards, the H.264 VCL design follows the so-called block-based hybrid video coding approach. The basic coding configuration for a macroblock is depicted in Figure 2. There is no single coding constituent which provides the majority of the improvement in density efficiency. It is rather a plurality of smaller improvements that add up to the considerable gain.

A coded video sequence in H.264 consists of a succession of coded pictures. A coded picture represents either an entire frame or a lone field, as was also the case in the MPEG-2 video. H.264 uses 4:2:0 sampling design in which chroma (Cb and Cr) samples are aligned horizontally with every second luma sample and are located vertically among two luma samples. A picture is partitioned into fixed-size macroblocks that each cover a rectangular picture area of 16 x 16 samples of the Luma constituent and 8 x 8 samples of each of the Chroma components. A picture may be split into one or a number of slices. In H.264 slices consist of macroblocks processed in raster scan order when not via flexible macroblock ordering (FMO). Using FMO, a picture can be split into many macroblock scanning outline such as interleaved slices, dispersed macroblock portion, one or more “foreground” slice groups and a “leftover” slice group, or a checkerboard category of mapping.

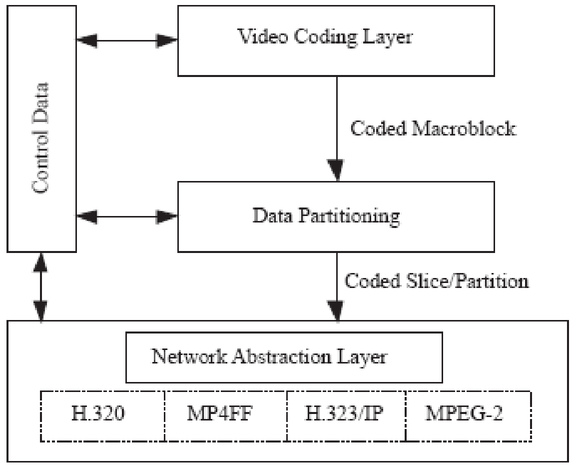


Figure 1: Structure of H.264/AVC video encoder

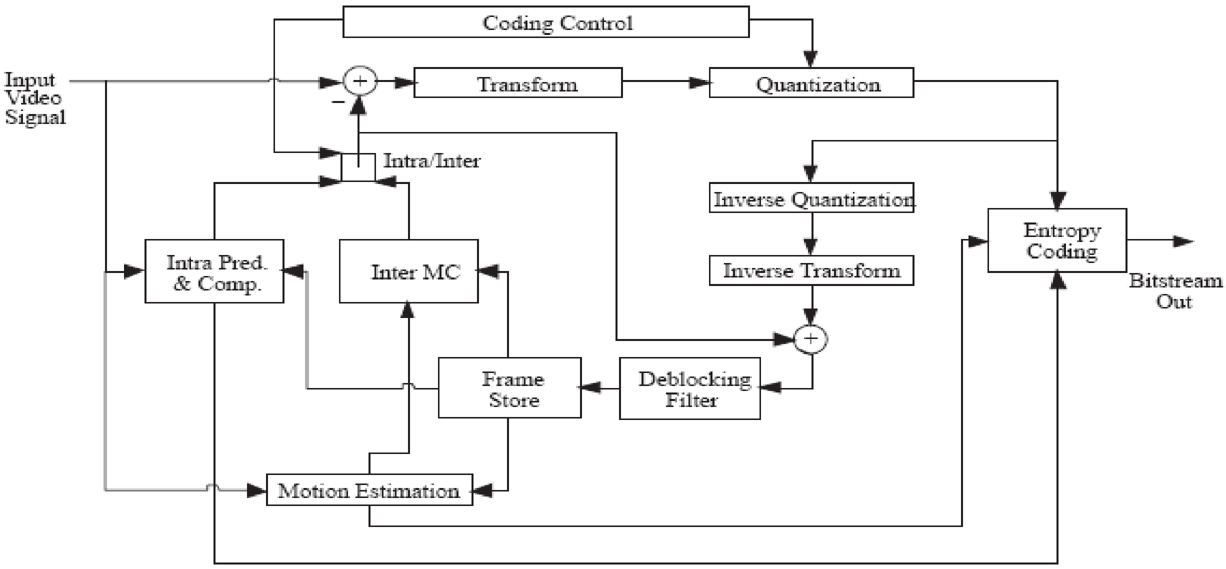


Figure 2: Basic coding structure of H.264/AVC for a MB

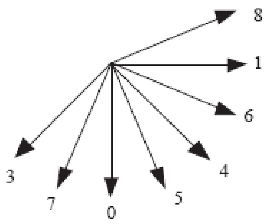
Each slice can be coded by using I, P, B, SP and SI frames. The first three are very comparable to those in previous standards with the exclusion of the use of reference pictures as described in the following. SP and SI slices, which are so-called switching P and I slices correspondingly, are the new ones. SP slices aim at efficient switching among different versions of the same video sequence whereas SI slices aim at arbitrary access and error recovery.

All Luma and Chroma samples of a macroblock are either spatially or temporally forecast, and the prediction residual is encoded by an integer transform. The transform coefficients are quantized and encoded using entropy coded methods (Figure 2). There are several new features and possibilities in H.264 for Intra/Inter-frame prediction. The types of intra coding support are denoted as Intra-4 x 4 or Intra- 16 x 16, which are lame forecast models, together with a chrome prediction (8 x 8) and I-PCM (i.e. Direct) forecast modes. The Intra-4 x 4 mode is well suited for coding of picture parts with considerable detail while the Intra-16 x 16 mode is more suitable for very smooth areas of the picture. The I-PCM mode allows the encoder to purely bypass the prediction and transform coding processes and express sending of the values of the encoded samples.

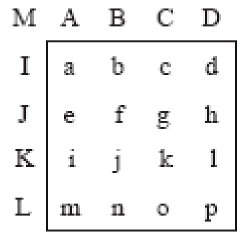
Each of the 4 *x* 4 luma blocks can be predicted using moreover the dc mode or one of the eight coding directions scheduled in Figure 3(c) and illustrated in Figure 3(a). For the use of illustration, Figure 3 (b) shows a 4 x 4 block of pixels a, b, c, .., P, belonging to a macroblock to be coded [3]. Pixels A, B, C, .., H and J, K, L, M are previously decoded neighboring pixels used in computation of forecast of pixels of current 4 *x* 4 block. Directional predictions use a linear weighted average of pixels of A through H and I through M, depending on the specific direction of the forecast. When utilizing the Intra-16 x 16 mode, four prediction modes are supported. Forecast mode 0 (vertical prediction), mode 1 (horizontal prediction), mode 2 (DC prediction), and mode 3 (plane prediction) are particularly similar to the modes in Intra-4 *x* 4 prediction except the number of neighboring pixels. The 8 *x* 8 chroma mode also uses a prediction procedure which is similar to the one for Intra-16 x 16.

H.264 standard is more flexible in the selection of motion compensation (MC) block sizes and shapes than any previous usual, with a minimum Luma MC block size as small as 4 x 4. Figure 4 illustrates the macroblock partitioning for MC prediction. The precision of MC is in units of one quarter of the distance between Luma samples. Predicted values at half-sample positions are obtained by applying a one dimensional 6-tap FIR filter horizontally and vertically. Predicted values at quarter-sample positions are produced by averaging samples at integer and half-sample positions. Since it is 4:2:0 video format, the displacements used for Chroma have one-eight sample location accuracy. The motion vector components are differentially coded using moreover median or directional prediction from neighboring blocks.

The H.264 syntax supports multi-picture motion-compensated prediction , in which extra than one previously coded picture can be used as a position for MC prediction. This new feature requires both encoder and decoder to store the position pictures used for inter prediction in a multi-picture buffer. Multiple position pictures not only contribute to the improvement of the compression efficiency, but also help error revival. In addition to the motion-compensated macroblock modes, a P macroblock can also be coded in P-Skip type. With this coding, neither quantized forecast error signal, nor a motion vector is transmitted. The useful effect of P-Skip mode is that large areas with no change or steady motion like slow panning can be represented with very few bits.



(a)



(b)

Num	Intra-4x4 Pred. Mode
0	vertical
1	horizontal
2	dc
3	diagonal-down-left
4	diagonal-down-right
5	vertical-right
6	horizontal-down
7	vertical-left
8	horizontal-up

(c)

Figure 3. a) Intra-4* 4 prediction directions, b) block prediction process, c) prediction modes

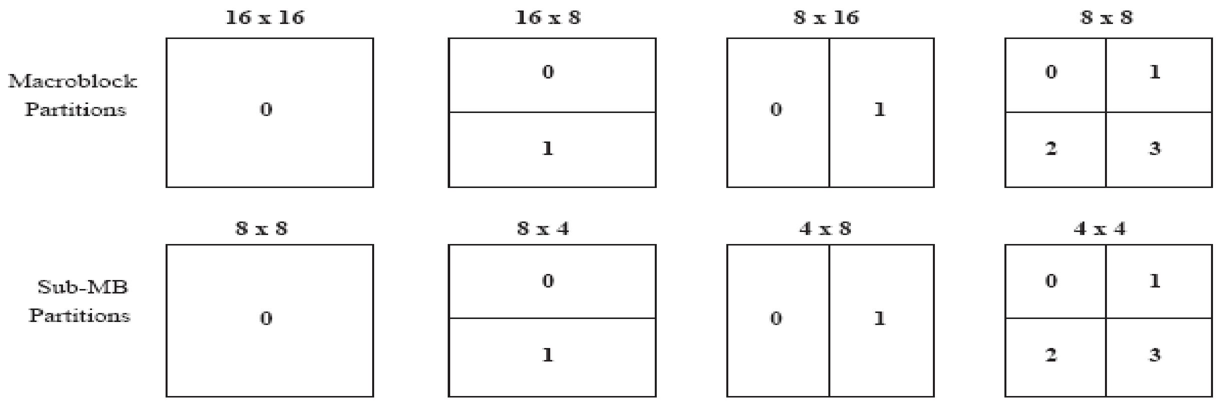


Figure 4. Macroblock partitions, sub-macroblock partitions and partition scans

The concept of B slices is sweeping in H.264 when compared with prior video coding standards . In B slices, to build the forecast signal, some macroblocks or blocks may use a weighted standard of two distinct motion-compensated prediction values. B slices employ two distinct lists of position pictures, which are referred to as the first *(list 0)* and the second *(list 1)* reference picture lists. Four dissimilar types of inter prediction are supported: list 0, list 1, bi-predictive, and straight prediction. For the bi-predictive mode, a weighted average of motion-compensated list 0 and list 1 prediction signals is used for the forecast signal. The direct prediction mode is inferred from previously transmitted syntax elements and can be any of the other types of modes. For each 16 x 16, 16 x 8, 8 x 16, and 8 x 8 panel, list 0, list 1 or bi-predictive methods can be chosen separately. A 8 x 8 partition of a B macroblock can also be coded in straight mode. Similar to P-Skip mode, if no prediction error signal is transmitted for a straight macroblock mode, it is also referred to as B-Skip mode.

H.264 uses three transforms depending on the type of outstanding data that is to be coded: A Hadamard alter for the 4 x 4 array of luma DC coefficients in Intra-16 x 16 mode, a Hadamard alter for the 2 x 2 array of chroma DC coefficients and a DCT-based integer transform for all other 4 x 4 blocks in the residual data. Thanks to the integrated change, inverse-transform mismatches are avoided. All inverse transform operations in H.264 can be realized using only additions and bit-shifting operations of 16-bit integer values. A quantization parameter (QP) is used in quantization procedure which can take 52 different values on a macroblock basis. These values are arranged so that a raise of one in QP means an increase of quantization step size by approximately 12%. Rather than even increment, the step sizes increase at a compounding rate. This feature is not there in prior standards and it is of great importance for compression efficiency.

In H.264, two methods of entropy coding are supported. The first one is Context-Adaptive Variable Length Coding (CAVLC) and the other one is Context-Adaptive Binary Arithmetic Coding (CABAC). In CAVLC, VLC tables for various sentence structure elements was switched depending on already transmitted syntax elements. As VLC tables are designed for competition the corresponding conditioned statistics, the entropy coding presentation is superior to the schemes using a single VLC table. CABAC improves the coding efficiency further (approximately 5-15% bit saving) by means of context modeling which is a procedure that adapts the probability model of arithmetic coding to the changing statistics within a video frame. In this procedure, conditional probabilities of the coding symbols and inter-symbol redundancy can be subjugated as well.

There are three Profiles in the standard. The Baseline outline supports I and P slices, and entropy coding with CAVLC. Also, it exploits redundant slices and arbitrary slice ordering (ASO) for error resilient coding. Potential submissions are video telephony, videoconferencing and wireless communications. The Main Profile includes hold for interlaced video, B slices, inter coding using weighted forecast and entropy coding with CABAC. Well suited application areas are television broadcasting and video storage. The Extended Profile does not hold interlaced video or CABAC but includes SP/SI slices to enable competent switching and data partitioning for improved error resilience. This profile may be particularly useful for streaming media applications.

Unlike MPEG-2, MPEG-4 part 2 or H.263, H.264 currently does not hold layered scalable coding. Furthermore, unlike MPEG-4 part 2, it does not hold an object-based video or object based scalable coding . The focus of the typical is achieving higher coding efficiency. Thus, it consists of a large number of tools designed to address competent coding over a wide variety of video material.

3. Details of the Compression Scheme

To adapt to the H.264 intraframe coding, the two proposed methods are developed as two intracoding modes: RSQ and BCIM. They will be discussed in detail in this section.

3.1 RSQ Mode

For text and graphics blocks containing edges of many directions as shown in Fig. 1, intraprediction along a single direction cannot completely remove the directional correlation among samples. After intraprediction, residues still preserve strong anisotropic correlation. In this case, it is not efficient to perform a transform on them. One method is to skip the transform and directly code prediction residues, which is similar to traditional pulse-code modulation (PCM). However, the question is whether the performance of PCM is better than that of a transform for text and graphics residual blocks. To answer it, we introduce the method proposed to analyze the coding gain of PCM over a transform. Given the same rate , the coding gain is defined as the ratio of distortions on transform coefficients and residual samples,respectively

$$G_{PCM/TC} = \frac{D_{TC}}{D_{PCM}}$$

D_{PCM} is the distortion of PCM and D_{TC} is the distortion on transform coefficients. We assume the distortions result from the optimal quantization. Considering a stationary source, each sample will have the same variance σ_S , D_{CPM} is equal to

$$D_{CPM} = \in_s^2 \sigma_s^2 2^{-2R}$$

$\in_s$ is a factor that depends on the probability distribution function of signal.

3.2 BCIM Mode

Having limited colors but complicated shapes is another property of the text and graphics parts on compound images. Such text/graphics blocks can be expressed concisely by several base colors together with an index map. It is somewhat like color quantization that is a process of choosing a representative set of colors to approximate all the colors of an image . In the BCIM mode, we first get the base colors of a block by using a clustering algorithm. All the base colors constitute a base color table. Then, each sample in the block will be quantized to its nearest base color. The index map indicates which base color is used by each sample.

Different from color quantization, each text/graphics block, but not an entire image, has its own base colors and an index map for representation in our scheme. Thus, it is content adaptive for each block. In addition, since the base color number of a block is small, fewer bits are required to represent each mapped index. Let us take the luminance plane of a 16 16 text/graphics block as an example, with each sample expressed by 8 bits. If four base colors are selected to approximate colors of that block, only two bits are required to represent each sample's index without compression.

Thus, the total bits required to represent the block can be reduced from to, saving almost 3/4 of the bits. Taking chrominance components of a block into account, the base colors and an index map representation can also be applied on a vector block consisting of three color components such as that in the color space. In this case, each base color is a vector but only one index map is required for that vector block. For convenience, we will refer to the representation on a vector block as Dim3 and that only on luminance with chrominance components coded by conventional method as Dim1. The Dim3 case is efficient for blocks having high correlation among three color components and the Dim1 case is efficient for those with low correlation among them.

3.3 Mode Selection and Mode Structure

Each mode has its advantages at dealing with blocks of different features. One question that arises here is how to fully take advantage of each mode in the proposed scheme. It can be solved by the RDO algorithm that has been adopted by H.264.The best mode with the best block partition having the minimum rate-distortion cost will be selected to compress the currentblock. All modes in the proposed scheme can be categorized into two types: spatial domain (SD) and DCT frequency domain (FD).There is a flag in the bit stream to distinguish them. The organized structure of all the modes. FD indicates the original intramodes in H.264, where the compression is performed in the DCT domain. SD indicates our proposed RSQ and BCIM modes. To adapt to the local nonstationary property ofcompound images, the spatial domain (SD) modes are applied to 16 X 16, 8 X 8, and 4 X 4 block sizes as those DCT frequency domain (FD) modes. The best mode in the spatial domain is compared with the best mode in the DCT frequency domain for the same size block in the rate-distortion sense. The better one is selected.

4. Experimental Results

The H.264/AVC standard is quite different and more flexible when compared to older video coding standards in terms of picture coding order, group of picture organization and assignment of reference pictures. In contrast to older standards, the coding and display order of the pictures is completely decoupled. Thanks to these features, H.264/AVC not only gives a better compression efficiency but also enables temporal scalability. In this section, performance resultsare reported for certain test sequences under several picture coding order and organization patterns.

a) A frame from BCIM MODE sequence

b) A frame from RSQ MODE sequence

c) A frame of OUTPUT MODE sequence

The hierarchical B picture concept is only examined in terms of compression efficiency in here. As far as temporal scalability is concerned, other issues such as reference selection should be considered. The reference picture lists have to be selected in a way that only pictures that belong to a coarser or same temporal level as the current picture are included in the reference picture lists.

5. Conclusion

The video signal has high temporal redundancies between a number of frames and this redundancy has not been exploited enough by current video compression technics. In this research, Video Compression in H.264/MPEG-4 Advanced Video Coding (AVC) compressed video streams for Spatial Correlation we suggest a new video compression method .With the apparent gains in compression efficiency we foresee that the proposed method could open new horizons in video compression domain; it strongly exploits temporal redundancy with the minimum of processing complexity which facilitates its implementation in video embedded systems. It presents some useful functions and features which can be exploited in some domains as video surveillance. In high bit rate, it gives the best compromise between quality and complexity. It provides better performance than MJPEG and MJPEG2000 almost in different bit rate values. Over .2000kb/s. bit rate values our compression method performance becomes comparable to the MPEG 4 especially for low motion sequences. There are various directions for future investigations. First of all, we would like to explore others possibilities of video representation. Another direction could be to combine. Accordiong representation. with other transformations such as wavelet transformation.

References

[1] Cuiling Lan, Guangming Shi, and Feng Wu "Compression Compound images in H.264/MPEG-4 AVC By Exploiting Spatial Correlation", IEEE Transactions On Image Processing, VOL. 19, NO. 4, APRIL 2010.

[2] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information - Part 2: Video", ISO/IEC 13818-2 (MPEG-2), 1994.

[3] A. Puri, X. Chen, A. Luthra, "Video Coding Using the H.264/MPEG-4 AVC Compression Standard", to be published in Elsevier Science, Signal Processing: image Communication, September 2004 issue.

[4] ISO/IEC JTC 1, "Advanced video coding", ISO/IEC FDIS 14496-10, International Standard, 2003.

[5] M. Horowitz, A. Joch, F. Kossentini, A. Hallapuro, "H.264/AVC Baseline Profile Decoder Complexity Analysis", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 704-716, 2003.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 560-576, 2003.

[7] I. E. G. Richardson, H.264 and MPEG-4 Video Compression, UK Wiley, 2003.

[8] M. Flierl, B. Girod, "Generalized B Pictures and the Draft H.264/AVC Video-Compression Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 587-597, 2003.

[9] D. Marpe, H. Schwarz, T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 620-636, 2003.

[10] P. List, A. Joch, J. Lainema, G. Bjontegaard, M. Karczewicz, "Adaptive Deblocking Filter", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 614-619, 2003.

[11] Z. Zhou, M.T. Sun, S. Hsu, "Fast Variable Block-size Motion Estimation Algorithms Based on Merge and Split Procedure for H.264/MPEG-4 AVC", IEEE ISCAS 2004 Conference.

[12] Z. Zhou, M.T. Sun, "Fast Macroblock Inter Mode Decision and Motion Estimation for H.264/MPEG-4 AVC", IEEE ICIP 2004 Conference.

[13] E. Akyol, A. M. Tekalp, M. R. Civanlar, "Motion-Compensated Temporal Filtering within the H.264/AVC Standard", IEEE ICIP 2004 Conference.

[14] A. Secker, D. Taubman, "Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation", IEEE ICIP 2002, pp. 749-752, 2002.

[15] M. Flierl, B. Girod, "Investigation of motion compensated lifted wavelet transforms", IEEE ICIP 2001, pp. 1029-1032, 2001.

[16] L. Luo, J. Li, S. Li, Z. Zhuang, Y. Zhang, "Motion compensated lifting wavelet and its application in video coding", ICME 2001, pp. 481-484, 2001.

[17] P. Chen, J. W. Woods, "Improved MC-EZBC with quarter-pixel motion vectors", ISO/IEC JTC1/SC29/WG11, MPEG2002/m8366, 2002.

[18] J. Xu, R. Xiong, B. Feng, G. Sullivan, M. Lee, F. Wu, S. Li, "3D Sub-band Video Coding using Barbell lifting", ISO/IEC JTC/WG11 M10569.

[19] M. Karczewicz, R. Kurceren, "The SP- and SI-Frames Design for H.264/AVC", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 637-644, 2003.

[20] X. Sun, F. Wu, S. Li, W. Gao, Y.Q. Zhang, "Improved SP coding techique", JVT-B097, Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, Geneva, 2002.

[21] S. Wenger, "H.264/AVC Over IP", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 645-656, 2003.

[22] T. Stockhammer, M. M. Hannuksela, T. Wiegand, "H.264/AVC in Wireless Environments", iEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 657-673, 2003.

[23] T. Stockhammer, T. Wiegand, T. Oelbaum, F. Obermeier, "Video Coding and Transport Layer Techniques for H.264/AVC-Based Transmission over Packet-Lossy Networks", IEEE ICIP 2003, Spain, 2003.

[24] S. Wenger, T. Stockhammer, M. Hannuksela, "RTP payload Format for JVT Video", draft-ietf-avt-rtp-h264-00.txt, internet Draft, 2002..

[25] E. Masala, C. F. Chiasserini, M. Meo, J. C. De Martin, "Real-Time Transmission of H.264 Video over 802.11-Based Wireless Ad Hoc Networks", Proceedings of Workshop on DSP in Mobile and Vehicular Systems, Japan, 2003.

[26] J. R. Corbera, P. A. Chou, S. L. Regunathan, "A Generalized Hypothetical Reference Decoder for H.264/AVC", iEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, no. 7, pp. 674-687, 2003.

[27] S. Ma, W. Gao, F. Wu, Y. Lu, "Rate Control for JVT Video Coding Scheme with HRD Considerations", iEEE international Conference on image Processing (iCiP), Vol. 3, pp. 793-796, Spain, 2003.