Turk J Elec Engin, VOL.13, NO.3 2005, © TÜBİTAK

# A Survey on the H.264/AVC Standard

**Nükhet ÖZBEK, Turhan TUNALI**
*International Computer Institute, Ege University,*
*35100, İzmir-TURKEY*
*e-mail: {ozbek, tunali}@ube.ege.edu.tr*

**Abstract**

*H.264/AVC is a recently completed video compression standard jointly developed by ITU-T VCEG and ISO/IEC MPEG standards committees. The standard is becoming more popular as it promises much higher compression than that possible with earlier standards. The standard provides flexibilities in coding and organization of data which enable efficient error resilience. The increased coding efficiency offers new application areas and business opportunities. As might be expected, the increases in compression efficiency and flexibility come at the expense of increase in complexity, which is a fact that must be overcome. This paper provides an overview of the technical features of H.264 and summarizes the emerging studies related to new coding features of the standard.*

**Key Words:** *H.264, AVC, video compression, inter mode decision, MCTF, SP/SI frames, HRD.*

## 1. Introduction

Recent acceptance of H.264 as a new decoding standard is expected to have far more implications than the production of just a new documentation. The consensus among the major players of the communications and video industry on H.264 might provide the major thrust for this new standard. Previous MPEG video coding standards such as MPEG-1 and MPEG-2 have enabled many familiar consumer products. For instance, these standards enabled video CD's and DVD's allowing video playback on digital VCRs/set-top-boxes and computers. The MPEG-2 video coding standard [1], which was developed about 10 years ago primarily as an extension of prior MPEG-1 video capability with support of interlaced video coding, was an enabling technology for digital television systems worldwide. It is utilized for transmission of standard definition (SD) and high definition (HD) TV signals over satellite, cable and terrestrial emission and the storage of high quality SD video signals onto DVDs. MPEG-4 was launched to address a new generation of multimedia applications and services such as interactive TV, internet video etc. The core of the MPEG-4 standard was developed during 1995-1999 [2], however MPEG-4 is a living standard with new parts added continuously as and when technology exists to address evolving applications [3]. The significant advances in core video standard were achieved on the capability of coding video objects, while at the same time, improving coding efficiency at the expense of a modest increase in complexity.

Meanwhile, an increasing number of services and growing popularity of HDTV are creating much more need for higher coding efficiency. Besides, other transmission media such as Cable Modem, UMTS or xDSL

offer much lower data rates than broadcast channels, and enhanced coding efficiency can enable more video channels or higher video quality within existing transmission capacities. Also, some applications such as internet multimedia, wireless video, personal video recorders, video-on-demand and videoconferencing have an inexhaustible demand for much higher compression to enable best video quality as possible. The H.264 standard [4] is a new state of the art video coding standard that addresses aforementioned applications. The core of this standard was completed in the form of final draft international standard (FDIS) in June 2003 while an extension for professional applications is currently in progress. H.264 promises significantly higher compression than earlier standards. Another name for H.264 is MPEG-4 Advanced Video Coding (AVC) standard. Since the standard is the result of collaborative effort of the VCEG and MPEG standards committees, it is informally referred to as Joint Video Team (JVT) standard as well.

The standard achieves clearly higher compression efficiency, often quoted as, up to a factor of two over the MPEG-2 video standard [5]. As one would expect, the increase in compression efficiency comes at the cost of substantial increase in complexity, often quoted as factor of four for the decoder, whereas encoding complexity may be as high as factor of nine over MPEG-2. Moreover, because of flexible features or subsets of the standard, the resulting complexity depends on the profile implemented, which is application dependent.

This paper provides an overview and summarizes emerging studies on the new coding features of the H.264 standard. The paper is organized as follows: Section 2 presents an overview of the H.264 standard. It provides details of coding structure and preferences of H.264. Following sections highlight some key technical features that enable improved operation for broad variety of applications. Section 3 examines new methods for effective macroblock inter mode decision and motion estimation processes. Section 4 emphasizes another important feature of H.264, that is flexible GOP structure. Section 5 provides information about new frame types, namely, SP/SI design which is not completed yet. Section 6 elaborates some of the other features of the standard such as the network interface and present reference code that is available to public. Section 7 presents some experimental results. Finally, in Section 8, concluding remarks are made.

## 2. Overview of the H.264 Standard

In order to address the need for flexibility and customizability, the H.264 standard covers a Video Coding Layer (VCL), which is designed for efficient representation of the video content, and a Network Abstraction Layer (NAL), which formats the VCL representation of the video and provides header information in a way that is appropriate for conveyance by different transport layers or storage media. Figure 1 depicts the structure of H.264/AVC video encoder [6].

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 [6]. The basic coding structure for a macroblock is depicted in Figure 2 [6]. There is no single coding element which provides the majority of the improvement in compression efficiency. It is rather a plurality of smaller improvements that add up to the significant gain [6].

A coded video sequence in H.264 consists of a sequence of coded pictures. A coded picture represents either an entire frame or a single field, as was also the case in MPEG-2 video. H.264 uses 4:2:0 sampling format in which chroma (Cb and Cr) samples are aligned horizontally with every second luma sample and are located vertically between two luma samples [7]. A picture is partitioned into fixed-size macroblocks that each cover a rectangular picture area of $16 \times 16$ samples of the luma component and $8 \times 8$ samples of

each of the chroma components. A picture maybe split into one or several slices. In H.264 slices consist of macroblocks processed in raster scan order when not using flexible macroblock ordering (FMO). Using FMO, a picture can be split into many macroblock scanning patterns such as interleaved slices, dispersed macroblock allocation, one or more "foreground" slice groups and a "leftover" slice group, or a checker-board type of mapping.

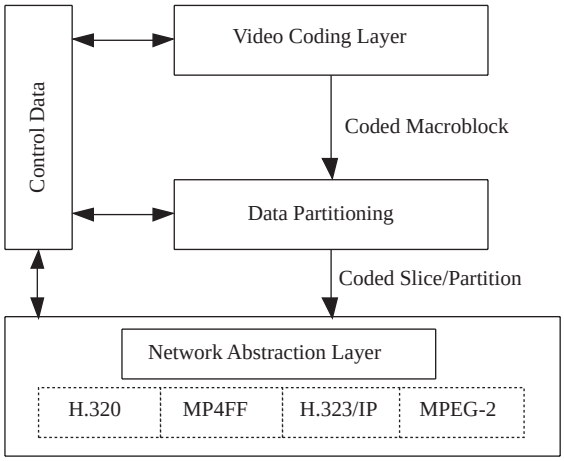



**Figure 1.** Structure of H.264/AVC video encoder [6].

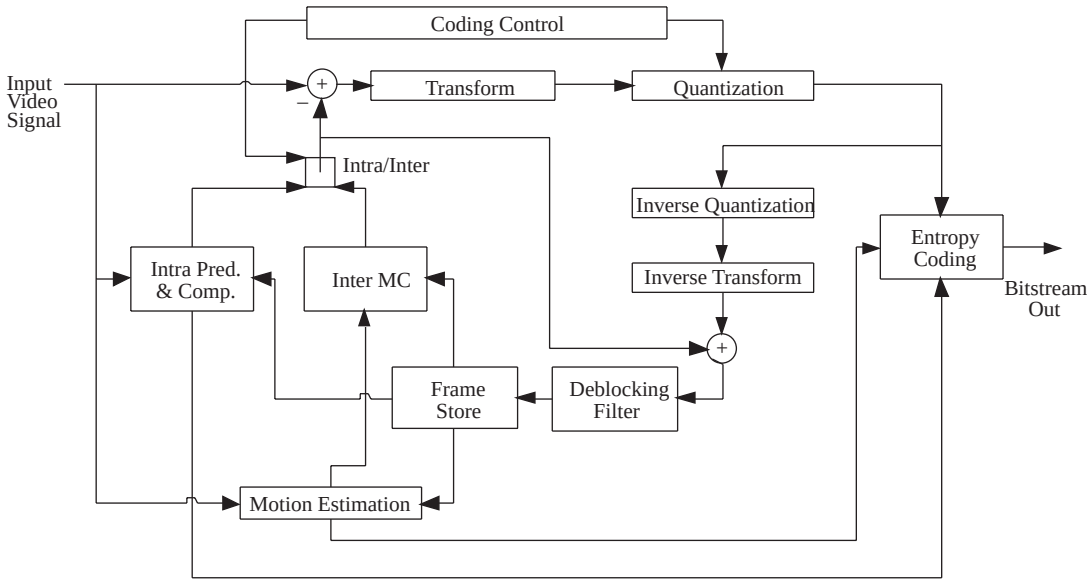



**Figure 2.** Basic coding structure of H.264/AVC for a MB [6].

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

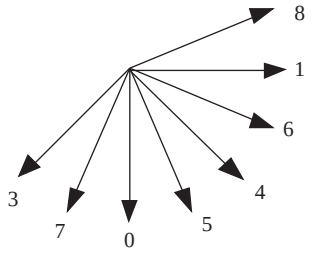
All luma and chroma samples of a macroblock are either spatially or temporally predicted, and the prediction residual is encoded using an integer transform. The transform coefficients are quantized and encoded using entropy coded methods (Figure 2). There are many new features and possibilities in H.264

for Intra/Inter-frame prediction. The types of intra coding supported are denoted as Intra-$4\times4$ or Intra-$16\times16$, which are luma prediction modes, together with chroma prediction ($8\times8$) and I-PCM (i.e. Direct) prediction modes. The Intra-$4\times4$ mode is well suited for coding of picture parts with significant detail while the Intra-16 $\times$ 16 mode is more suitable for very smooth areas of the picture. The I-PCM mode allows the encoder to simply bypass the prediction and transform coding processes and direct sending of the values of the encoded samples.

Each of the $4\times4$ luma blocks can be predicted using either the dc mode or one of the eight coding directions listed in Figure 3(c) and illustrated in Figure 3(a). For the purpose of illustration, Figure 3(b) shows a $4\times4$ block of pixels a, b, c, .., p, belonging to a macroblock to be coded [3]. Pixels A, B, C, .., H and I, J, K, L, M are already decoded neighboring pixels used in computation of prediction of pixels of current $4\times4$ 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 prediction. When utilizing the Intra-$16\times16$ mode, four prediction modes are supported. Prediction mode 0 (vertical prediction), mode 1 (horizontal prediction), mode 2 (DC prediction), and mode 3 (plane prediction) are specified similar to the modes in Intra-$4\times4$ prediction except the number of neighboring pixels. The $8\times8$ chroma mode also uses a prediction technique which is similar to the one for Intra-$16\times16$.

H.264 standard is more flexible in the selection of motion compensation (MC) block sizes and shapes than any previous standard, with a minimum luma MC block size as small as $4\times4$. Figure 4 illustrates the macroblock partitioning for MC prediction [4]. The accuracy of MC is in units of one quarter of the distance between luma samples. Prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap FIR filter horizontally and vertically. Prediction values at quarter-sample positions are generated 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 position accuracy. The motion vector components are differentially coded using either median or directional prediction from neighboring blocks.

The H.264 syntax supports multi-picture motion-compensated prediction [8], in which more than one previously coded picture can be used as reference for MC prediction. This new feature requires both encoder and decoder to store the reference pictures used for inter prediction in a multi-picture buffer. Multiple reference pictures not only contribute to the improvement of the compression efficiency, but also help error recovery. In addition to the motion-compensated macroblock modes, a P macroblock can also be coded in P-Skip type. With this coding, neither quantized prediction error signal, nor a motion vector is transmitted. The useful effect of P-Skip mode is that large areas with no change or constant motion like slow panning can be represented with very few bits.




(a)

| M | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| I | a | b | c | d | | | | |
| J | e | f | g | h | | | | |
| K | i | j | k | l | | | | |
| L | m | n | o | p | | | | |

(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-$4\times4$ prediction directions, b) block prediction process, c) prediction modes [3].

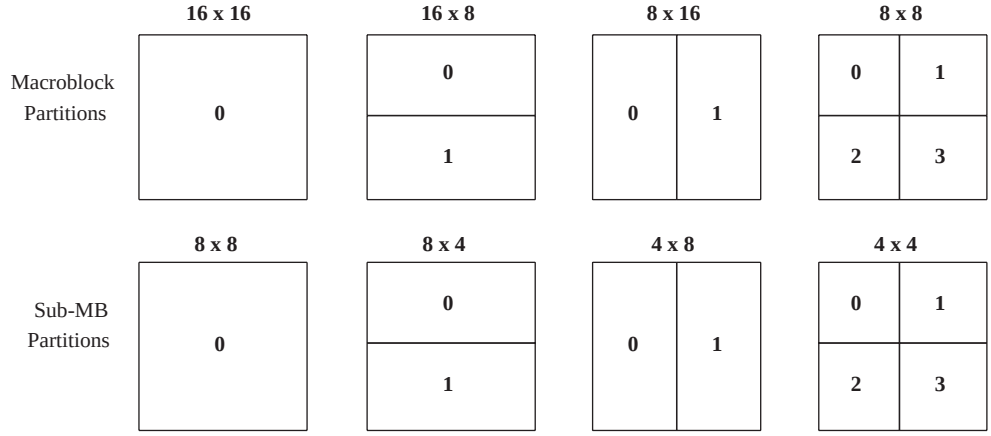



**Figure 4.** Macroblock partitions, sub-macroblock partitions and partition scans [4].

The concept of B slices is generalized in H.264 when compared with prior video coding standards [9]. In B slices, to build the prediction signal, some macroblocks or blocks may use a weighted average of two distinct motion-compensated prediction values. B slices employ two distinct lists of reference pictures, which are referred to as the first (***list 0***) and the second (***list 1***) reference picture lists. Four different types of inter prediction are supported: list 0, list 1, bi-predictive, and direct prediction. For the bi-predictive mode, a weighted average of motion-compensated list 0 and list 1 prediction signals is used for the prediction signal. The direct prediction mode is inferred from previously transmitted syntax elements and can be any of the other types of modes. For each $16\times16$, $16\times8$, $8\times16$, and $8\times8$ partition, list 0, list 1 or bi-predictive methods can be chosen separately. An $8\times8$ partition of a B macroblock can also be coded in direct mode. Similar to P-Skip mode, if no prediction error signal is transmitted for a direct macroblock mode, it is also referred to as B-Skip mode.

H.264 uses three transforms depending on the type of residual data that is to be coded: A Hadamard transform for the $4\times4$ array of luma DC coefficients in Intra-$16\times16$ mode, a Hadamard transform for the $2\times2$ array of chroma DC coefficients and a DCT-based integer transform for all other $4\times4$ blocks in the residual data. Thanks to the integer transformation, inverse-transform mismatches are avoided. All inverse transform operations in H.264 can be implemented using only additions and bit-shifting operations of 16-bit integer values. A quantization parameter (QP) is used in quantization process which can take 52 different values on a macroblock basis. These values are arranged so that an increase of one in QP means an increase of quantization step size by approximately 12%. Rather than constant increment, the step sizes increase at a compounding rate. This feature is not present 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 syntax elements are switched depending on already transmitted syntax elements. As VLC tables are designed to match the corresponding conditioned statistics, the entropy coding performance is superior to the schemes using a single VLC table. CABAC [10] improves the coding efficiency further (approximately 5-15% bit saving) by means of context modeling which is a process that adapts the probability model of arithmetic coding to the changing statistics within a video frame. In this process, conditional probabilities of the coding symbols and inter-symbol redundancy can be exploited as well.

H.264 specifies the use of an adaptive de-blocking filter [11] that operates on the horizontal and vertical block edges within the prediction loop in order to remove artifacts caused by block prediction errors.

With the filter, the blockiness is reduced, while the sharpness of the content is basically unchanged and the subjective quality is significantly improved. The filter reduces bit rate typically by 5-10% compared to the non-filtered video.

There are three Profiles in the standard. The Baseline Profile supports I and P slices, and entropy coding with CAVLC. Also, it utilizes redundant slices and arbitrary slice ordering (ASO) for error resilient coding. Potential applications are videotelephony, videoconferencing and wireless communications. The Main Profile includes support for interlaced video, B slices, inter coding using weighted prediction and entropy coding with CABAC. Well suited application areas are television broadcasting and video storage. The Extended Profile does not support interlaced video or CABAC but includes SP/SI slices to enable efficient 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 support layered scalable coding. Furthermore, unlike MPEG-4 part 2, it does not support object-based video or object based scalable coding [3]. The focus of the standard is achieving higher coding efficiency. Thus, it consists of a large number of tools designed to address efficient coding over a wide variety of video material.

## 3. MB Inter Mode Decision and ME Algorithms

In the emerging H.264/AVC standard, for Inter coded Macroblocks (MBs), tree structured variable block-sizes can be utilized in motion estimation (ME). There are totally seven different block-sizes as given in Figure 4. In the reference software, a Fast Full Search (FFS) algorithm is employed for the ME. The required SAD (Sum of Absolute Differences) computation is larger than that for a 16 $\times$ 16 MB with 961 search points, which means vast amount of computation [12].

A fast variable block-size motion estimation algorithm, based on merge and split procedures is proposed in [12]. In this algorithm, the search points (SP) can be reduced to 4% of that using the FFS ME for a $16\times16$ MB. The idea behind the search procedure is to take advantage of the correlation of different block-sizes. The initial block-size and the accuracy of the MV prediction are important for the efficiency of the algorithm which starts from $8\times8$ block-size, and merges to larger block-sizes or splits to smaller block-sizes. It runs so called ADSS (Adaptive Diversity Search Strategy) ME for the 8 $\times$ 8 blocks. In the bottom-up Merge process, for matched MVs, it returns the same MV for the larger block-sizes. For other MVs, it takes the average MV as the prediction MV and performs MV search for 8 $\times$ 16, $16\times8$ and $16\times16$ blocks. In the top-down split process, $8\times8$ blocks are split to $8\times4$, 4 $\times$ 8 and $4\times4$ blocks. It takes the MV of the corresponding larger-block as the prediction MV and performs so called SDSP (Small Diamond Search Pattern) search. The performance result taken under simulation conditions with 30 fps frame rate and [-16, 15] search range shows that by achieving almost the same PSNR and bit rate values, the FFS method uses 961 SP per MB whereas the Bottom-up Merge uses 32, the Top-down Split 35, and the Merge & Split 33 SP per MB [12].

In H.264/AVC, Macroblock mode decision and motion estimation are the most computationally expensive processes. Mode decision is a process such that for each block-size, bit-rate and distortion are calculated by actually encoding and decoding the video. Therefore, the encoder can achieve the best Rate Distortion (RD) performance, at the expense of calculation complexity. In the JM reference code, the ME and the mode decision are executed together. For each mode, ME is done first and the resulting cost is used for the mode decision.

In [13], a fast inter-mode decision and ME algorithm is proposed by excluding the low-possibility modes in the mode decision process. It is reported that the proposed algorithm could achieve similar RD performance with about $^1/_2$ computation saving when compared to the JM low-complexity mode with a FFS ME algorithm.

There are two important observations underlying the main idea of the algorithm of [13]. The first is that large areas of background in a picture may be still or under global motion and they can be predicted well by the MVs of neighboring blocks. $16\times16$ generally is the best block-size in these areas. The second is that, if the cost of a larger block-size mode is larger than that of the current block-size mode, then further larger block-size modes can be excluded. Similarly, if the cost of a smaller block-size mode is larger than that of the current block-size mode, then further smaller block-size modes can be excluded.

Firstly, the algorithm estimates the motion for the $16\times16$ block only at (0,0) or PMV (Prediction Motion Vector). If either of the motion costs is smaller than a certain threshold, the mode decision process stops and returns the $16\times16$ mode. Then, it estimates the motion for the four $8\times8$ blocks by ADSS. If all of the four $8\times8$ motion costs are smaller than another threshold, the best mode is $8\times8$; else for the two $16\times8$ and $8\times16$ blocks, the motion is estimated by MV merging, which is proposed in [12]. For further details of the algorithm, the reader should refer to [13].

The number of excluded modes in the algorithm can be determined for different chosen (best) modes. For instance, if the $16\times16$ mode is selected as the best, all other modes are excluded. If the $8\times8$ mode is chosen, either all other modes except $8\times8$ or $16\times16$ and $4\times4$ modes are excluded. If the $16\times8$ or $8\times16$ mode is selected, $4\times8$, $8\times4$ and $4\times4$ modes are excluded. In the case of selecting $4\times8$, $8\times4$ or $4\times4$ mode, only mode $16\times16$ is excluded. Thanks to the exclusion, the calculation is reduced by half with a slight bit rate increase when compared with the JM low-complexity mode (LCM) [13]. The bit-rate increase arises from the mode misjudge.

Performance results show that, compared to LCM with Merge-Split Search (MSS) ME algorithm, the computation can be reduced by about $^1/_4$ with negligible quality degradation. The saving is about $^1/_2$, compared to LCM with FFS [13]. In particular, when Foreman QCIF video with GOP structure of IPPPPIPPPP and 25 as Qp value is used, in terms of run time, the LCM-MSS achieves 29% reduction whereas the algorithm of [13] achieves 48% reduction over the LCM-FS while achieving the same 38 dB PSNR. However, the trade-off turns out to be the increase in the bit rate. Compared to the 328 kbps of LCM-FS, LCM-MSS has 348 kbps and the algorithm of [13] has 352 kbps.

More calculation reduction may be expected by pre-defined bit rate applications. For high bit rate applications, sub-MB modes can be checked first to obtain the best mode faster, which can save some computation for larger blocks. For low bit rates, larger modes can be checked first.

# 4. Flexible GOP Structure

In prior standards, there was a strict dependency between the ordering of pictures for motion compensation referencing purposes and the ordering of pictures for display purposes [6]. In H.264/AVC, these restrictions are largely removed and the new features are added such as multiple reference picture motion compensation, stored-B pictures, decoupling of referencing order from display, decoupling of picture representation methods from picture referencing capability and weighted prediction.

By means of the rich prediction feature set, an MCTF (Motion Compensated Temporal Filtering) approach is implemented in [14]. This study shows that it is possible to produce an effective layering

scheme without modifying the standard. MCTF has traditionally been studied with fully scalable wavelet video coders. It performs temporal bi-orthogonal wavelet transform on frames using lifting that involves prediction and update steps. However, it suffers degradations at scene changes and occlusion regions.

The H.264 syntax has a feature called "stored B pictures" which allows the use of B frames as reference frames for other pictures. The configuration of lifting scheme of [14] with the GOP structure IBBBBBBBI is given in Figure 5 [14]. This configuration provides a three layer bit stream that includes the base layer of which the members are F1, F5, F9 and two enhancement layers of which the members are F3, F7 and the rest of the frames respectively. In this scheme, each temporal layer is encoded and decoded independently from higher temporal layers, hence, no drift occurs in lower bands when higher temporal frames are unavailable. Flexibility in choosing the number of layers in every GOP provides better adaptation to varying bit-rates.

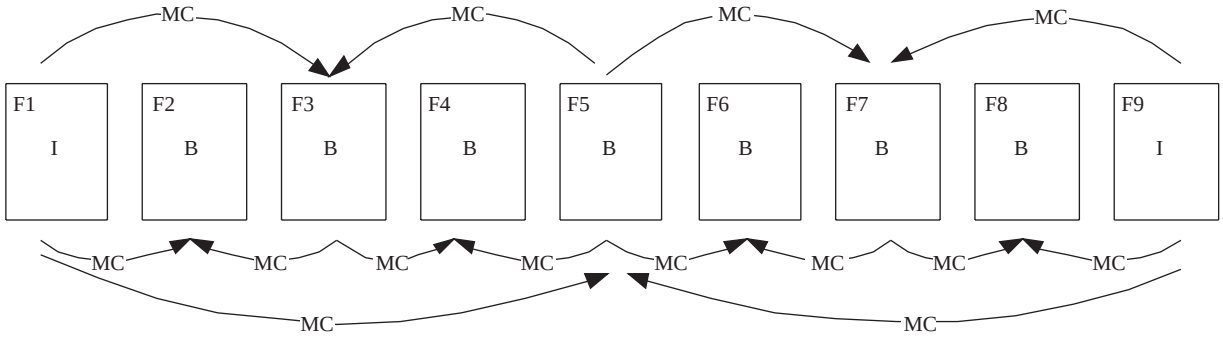



**Figure 5.** H.264/AVC configuration of lifting scheme with GOP=8 [14].

When compared with other MCTF based scalable video coders with mesh based [15] and block based [16, 17, 18] motion estimation, the approach of [14] outperforms them in terms of bit-rate and PSNR relation. Other scalable coders may support SNR and spatial scalability but H.264 is already known as a non-scalable video coder. The difference is mainly because of the advanced motion compensated prediction features of the H.264 standard. Comparative results also show that the PSNR difference gets larger as the bit-rate increases due to the amount of information sent by encoder.

The approach of [14] is also compared with the latest MPEG Scalable Video Coding (SVC) group's reference codec which implements H.264 like features and also employs Barbell lifting for temporal decomposition [19]. For a fair comparison, the Barbell codec is run at only temporal scalability mode with three layers. CABAC is set in both codecs. GOP size is set to 16 in the MCTF in H.264 scheme to avoid GOP boundary effects as much as possible. According to test results given in [14], Barbell lifting based MCTF performs 37.52 dB PSNR at 665.2 kbps coding bit rate whereas MCTF in H.264 achieves 38.09 dB PSNR at 652.0 kbps coding bit rate. It is also reported that the performance of [14] is superior even for corresponding temporal sub layers.

# 5. SP/SI Frame Design

The H.264 standard includes new frame types that allow exact synchronization through replacing I frames. This property enables switching a decoder between representations of the video content that used different data rates (bit stream switching-splicing), recovery from data losses or errors, and support for trick modes (fast forward, fast reverse) and random access as well. SP frames make use of motion compensated predictive

coding to exploit temporal redundancy in a sequence similar to P frames, but it differs from P frames by allowing identical frames to be reconstructed even when they are predicted by using different reference frames [20]. Since SP frames utilize temporally predictive coding, they require significantly fewer bits than I frames to achieve similar quality. In some applications, SI frames are used in conjunction with SP frames. An SI frame uses only spatial prediction as an I frame and still reconstructs identically the corresponding SP frame.

The H.264 standard adopts an SP coding method that allows seamless switching between bit streams with different bit rates at predictive frames. The general scenario is given in Figure 6 [22]. At time t, S1 and S2 are at a switching point provided for switching from Bitstream 1 to Bitstream 2 and vice versa. S1, S2 and S12 are compressed as SP frame with a slight difference. Due to this difference, S1 and S2 are referred to as primary SP frames whereas S12 as secondary SP frame [20]. Assume that Bitstream 1 is being transmitted to the user. When there is a switch to Bitstream 2, instead of S1, S12 is transmitted at time t. By decoding S12, the decoder can obtain exactly the same reference as the one obtained by decoding S2 at time t.

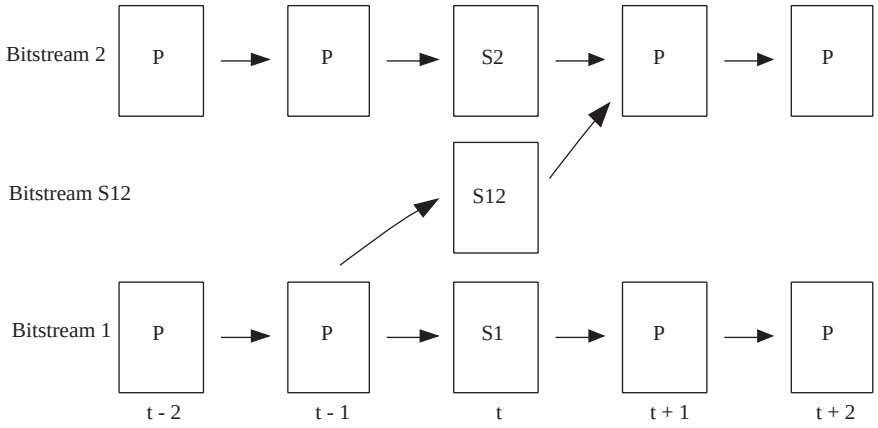



**Figure 6.** Switching from Bitstream 1 to Bitstream 2 through SP frames [22].

Therefore, it can continue with decoding Bitstream 2 at time t+1 seamlessly. This property of switching frames allows convenience while switching between streams. However, there are some potential problems with SP scheme as reported in [21]. The performance is limited due to coding efficiency and streaming quality. Firstly, there are three quantization operations in the SP scheme, particularly the quantization Qs (i.e. quantization parameter for switching purpose) module with the same parameter is used two times. Each quantization would inevitably decrease the coding efficiency of the SP coding. Secondly, the quantized, with Qs, and then dequantized ($L_{pred}$) prediction DCT coefficients ($K_{pred}$) are subtracted from original DCT coefficients to form the prediction at the encoder, but the reconstruction uses $K_{pred}$, without quantization (with Qs) and dequantization, to form the prediction. Though this mismatch would not cause error propagation, it would decrease the coding efficiency.

In [22], a new SP coding scheme is proposed aiming at overcoming the aforementioned drawbacks and improving coding efficiency. The new approaches that [22] employs are two coefficient predictive coding modes and the rate-distortion optimization, separating the quantization parameters for up-switching and down-switching, decoupling the up-switching and down-switching points, and removing unnecessary quantization processes.

Primary SP coding block diagrams for the original scheme and the one proposed in [22] are given

in Figure 7 and 8, respectively [22]. The main improvement is that two DCT coefficient coding modes are proposed for prediction to efficiently limit the mismatch between the references used in prediction and reconstruction. According to the current coefficient predictive coding mode determined by a rate-distortion optimized mode decision, either $K_{pred1}$ or $K_{spred1}$ is subtracted from $K_{orig1}$. In the scheme proposed in [22], the mismatch between the references of prediction and reconstruction still exists. However, as the rate-distortion optimization is involved in the mode decision criterion, the side effect of such mismatch is effectively reduced.

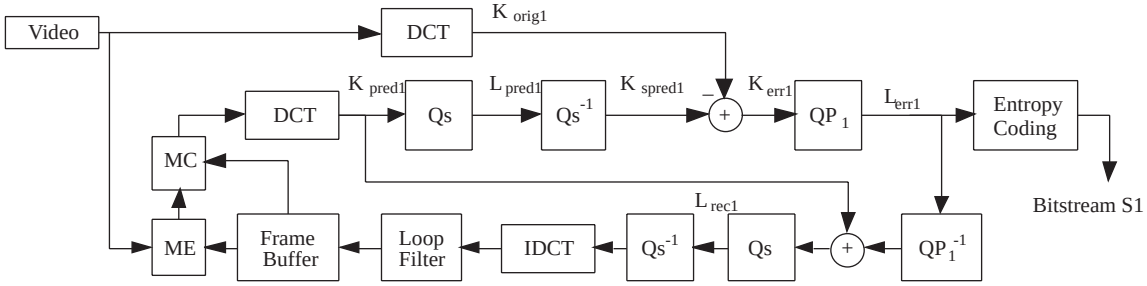



**Figure 7.** Primary SP frame coding in H.264/AVC [22].

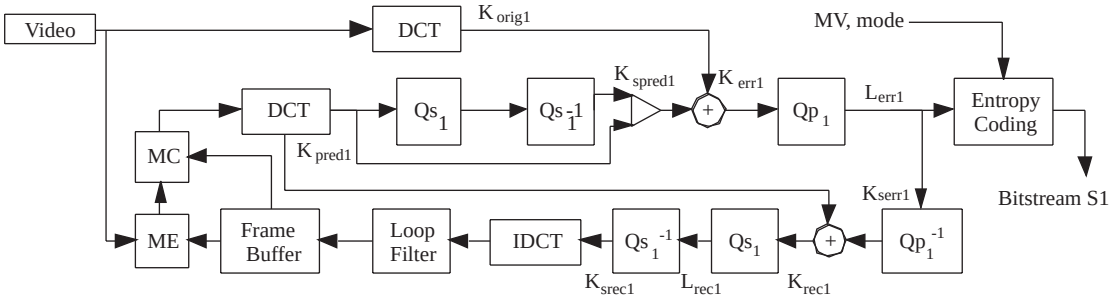



**Figure 8.** Primary SP frame coding in the proposed encoder [22].

Figure 9(a) and 9(b) illustrate the most important part of secondary SP frame coding for the original scheme and the one proposed in [22], respectively [22]. Figure 9(a) shows the encoding of switching bitstream S12, which first subtracts $L_{pred1}$ in S1 from the reconstruct level $L_{rec2}$ in S2, and then performs entropy coding of the obtained difference. Different from the original SP method, the quantization of the prediction coefficients is moved out from the primary SP encoding loop to the secondary SP encoding loop. The prediction coefficients $K_{pred1}$ of S1 is quantized using $Qs_2$ to obtain coefficients $L_{pred12}$. The difference between $L_{pred12}$ and the reconstructed level $L_{rec2}$ in S2 is entropy coded to generate the bitstream S12. Since the quantization with the Qs on the prediction is released from the encoding loop, bitstream S12 is only related to $Qs_2$ , while bit stream S21 for switching from Bitstream S2 to Bitstream S1 only depends on $Qs_1$. Therefore the switching points for up-switching and down-switching can be decoupled according to real streaming requirements. It can also provide more switching-down points than switching-up points.

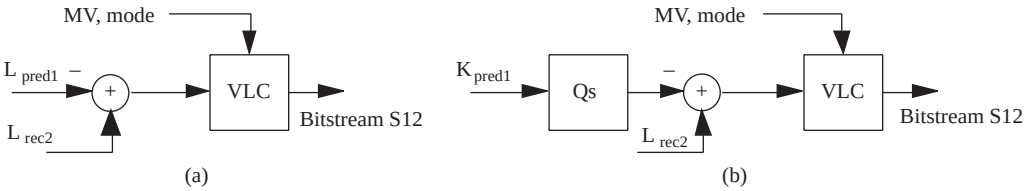



**Figure 9.** Secondary SP frame coding in a) H.264/AVC, b) the proposed one [22].

Another important feature is the size of the down-switching bitstream which can be much smaller than that of the up-switching one. Such features are of great importance for TCP-friendly streaming systems.

## 6. Other Features/Advances

H.264/AVC is an attractive candidate for many applications including fixed and wireless video transmission over the Internet Protocol (IP) because of its significantly increased compression efficiency, seamless and easy integration into all current and possible future protocol architectures and enhanced error resilience features [23, 24].

In [25], different transmission schemes and error resilience features for packet lossy environments are compared and experimental results based on common test conditions are discussed. The selection of suitable features is also addressed in [25]. In the investigated transmission system, the generated slices are mapped to Network Adaptation Layer (NAL) units. The RTP payload specification [26] covers simple encapsulation of NAL units and includes the concept of aggregation packets which means that several NAL units can be transported within one IP packet (Slice Interleaving). For instance, slices containing odd MB rows are transmitted within the first IP packet and even MB rows are transmitted in the second IP packet. This concept does not reduce the coding overhead due to limited spatial prediction, but the costly IP overhead (40 bytes header per packet) can be avoided. A more advanced concept is flexible MB ordering (FMO) [27] which provides the possibility to transmit MBs in non-scan order. FMO is especially powerful with appropriate error concealment. Another error resilience concept is data partitioning which can also reduce visual artifacts resulting from packet losses, especially if prioritization or unequal error protection is provided by the network.

According to simulation results in [25], unless slice structured coding is necessary to limit the transport packet layer size, it may be preferable to transport one frame in one transport layer packet. However, if slice structured coding is applied, advanced error concealment (AEC) provides significant gains. An efficient way to limit error propagation seems to be multiple reference frames combined with low-delay feedback information. If no feedback channel is available, then Channel Adaptive Rate Distortion Optimization (CA-RDO) method, which contains channel adaptive mode selection for MB Intra updates, shows very good performance. Only for packet loss rates of about 10% or higher the application of FEC schemes, that is based on the Reed-Solomon codes and RFC2733, with CA-RDO seems to be interesting.

Real-time transmission of H.264 video over a wireless 802.11 ad hoc network is performed and analyzed in [28]. It is noted that there is a trade-off in choosing the packet size. If it is too short, so is the slice size, compression ratio decreases since slice headers would reduce the available bandwidth and context-based entropy coding would become less efficient. On the other hand, in wireless channels, longer packets are more likely to contain transmission errors. It is observed from the simulations that a video packet size as small as 300 bytes should be used when channel conditions are not favorable. It is reported that, to obtain high PSNR values for most channel conditions, maximum number of transmission attempts at the MAC layer should be three [28].

Hypothetical Reference Decoder (HRD) is a very important part in H.264, which represents a set of normative requirements on bit stream for the purpose of avoiding buffer (coded picture buffer - CPB) overflow and underflow [29]. HRD is conceptually connected to the output of an encoder and consists of a decoder buffer, a decoder and a display unit. HRD and CPB employ a leaky bucket algorithm. In H.264 a

constrained arrival time leaky bucket (CAT-LB) model has been defined. The problem of HRD requirements can be solved by rate control implemented in the encoder. An improved rate control scheme for H.264 video coding with HRD and CAT-LB constraints is proposed in [30]. The algorithm mainly contains two processes, namely bit allocation and quantization parameter (Qp) adjustment. First, the target bit for each picture is clipped with an upper and lower bound required by HRD constraints. Second, rate control must maintain the target bit by adjusting the Qp at frame and/or MB level adaptively. Sometimes even a frame may be skipped if the buffer occupancy is too high. The experimental results show that the algorithm in [30] can generate the HRD compliant bit stream and the generated bit rate is very close to the target bit rate. The encoding efficiency is similar to or even better than that of fixed Qp video coding.

In [31], a content adaptive optimal rate control algorithm is proposed. The algorithm is formulated as a constrained optimization problem which is solved by using dynamic programming. The technique minimizes the distortion and the initial waiting time for continuous playback under acceptable distortion constraints. Spatial resolution and frame rate of input video and Qp values are used as optimization variables. It is applied to encoding of soccer videos using an H.264 encoder. The input video is divided into temporal segments according to "relevancy" of the content and user context. Performance results show that the algorithm in [31] can achieve higher PSNR values comparing to RDO H.264 encoder, in which Qp may change in an MB-based manner, especially in highly relevant parts of the input video. The perceptual quality of the video encoded with the new method is also superior to that of the RDO method in terms of blockiness and flatness factors.

## 7. 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 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 results are reported for certain test sequences under several picture coding order and organization patterns.

Recently, dyadic and non-dyadic hierarchical coding structure support is implemented in the JM reference software and refered to as "Pyramid Coding". With hierarchical B pictures, instead of common plain B coding order, one could consider to code the sequence in a coding pattern I0-P4-RB2-B1-B3-P8- ... or even I0-P8-RB4-RB2-RB6-B1-B3-B5-B7-P16- ... , where RB refers to a B picture which can be used as reference. We will now call the former as 3level-3B and the latter as 4level-7B coding order. The latter coding order can be visualized as the one depicted in Figure 5.

In our experiments, the JM reference software version 9.6 is used for evaluation. The sequences used are the following: Bus CIF, Foreman CIF, and Coastguard CIF. The first 150 frames of the test sequences are coded for evaluation. The QP values are chosen as 28 for I and P slices, 29 for RB and 30 for B slices. The search range is selected as 32 for all tests. Although 5 references are stored in the reference buffer, not all of them are used. A triple is constructed such that the first element means the number of references in P list0, the second and third elements mean B list0 and B list1, respectively. Single slice per picture and the CABAC entropy method are used. For B slices, Spatial Direct mode and bi-predictive based motion estimation are activated. Rate Distortion Optimization (RDO) is on and Fast Motion Estimation (FME) is off during simulations.

Coding structures that are evaluated can be summarized as follows:

1) P only coding (IPPPP ... ) with buffer size (2,-,-),

2) 1 non-reference B slices (IBPBP ... ) with buffer size (2,1,1),

3) 2 non-reference B slices (IBBPBBP ... ) with buffer size (2,1,1),

4) 3 level pyramid using 3 B pictures (I0-P4-RB2-B1-B3-P8-) with buffer size (2,1,1),

5) 3 level pyramid using 5 B pictures (I0-P6-RB2-RB4-B1-B3-B5-P12-) with buffer size (2,3,3),

6) 3 level pyramid using 7 B pictures (I0-P8-RB2-RB4-RB6-B1-B3-B5-B7-P16-) with buffer size (2,3,3),

7) 4 level pyramid using 7 B pictures – dyadic hierarchy (I0-P8-RB4-RB2-RB6-B1-B3-B5-B7-P16-) with buffer size (2,3,3),

8) 4 level pyramid using 11 B pictures – nondyadic hierarchy (I0-P12-RB6-RB3-B1-B2-B4-B5-RB9-B7-B8-B10-B11-P24-) with buffer size (2,3,3).

For different coding structures tables 1, 2 and 3 give performance results of "bus_cif", "foreman_cif" and "coastguard_cif" sequences, respectively. Total encoding time, bitrate and luminance PSNR values are obtained from the JM encoder report. The experiments are performed on a P4 3GHz PC with 1 GB RAM.

The simulation results show that the coding efficiency can be improved significantly with the hierarchical B picture coding in comparison to the classical coding structures. In general, the 3level-3B (3L3B) structure performs better than the common "IBBPBBP" structure. In "foreman" and "coastguard" sequences, increase in hierarchy level and number of B pictures (hence increase in GoP size) yields additional improvement in terms of bitrate. However, for "bus" sequence the most appealing structure seems to be 3L3B which gives considerably better performance compared to other structures. This difference in performance of "bus" sequence may be related to high motion characteristics of the video content.

**Table 1.** Performance results for "bus_cif" sequence.

| Coding Structure | Time (sec.) | Bitrate (kbps) | PSNR Y (dB) |
|---|---|---|---|
| **1 - IPPP** | 578 | 1089.85 | 34.79 |
| **2 - IBPBP** | 627 | 881.10 | 34.43 |
| **3 - IBBPBBP** | 647 | 827.50 | 34.14 |
| **4 – 3L3B** | 658 | 802.65 | 34.11 |
| **5 - 3L5B** | 1457 | 809.48 | 33.92 |
| **6 - 3L7B** | 1504 | 825.46 | 33.81 |
| **7 - 4L7B** | 1502 | 816.79 | 33.86 |
| **8 - 4L11B** | 1563 | 820.44 | 33.58 |

**Table 2.** Performance results for "foreman_cif" sequence.

| Coding Structure | Time (sec.) | Bitrate (kbps) | PSNR Y (dB) |
|---|---|---|---|
| **1 - IPPP** | 564 | 352.69 | 36.78 |
| **2 - IBPBP** | 607 | 299.73 | 36.69 |
| **3 - IBBPBBP** | 622 | 278.60 | 36.54 |
| **4 - 3L3B** | 635 | 269.55 | 36.47 |
| **5 - 3L5B** | 1417 | 257.19 | 36.39 |
| **6 - 3L7B** | 1463 | 252.73 | 36.31 |
| **7 - 4L7B** | 1462 | 254.46 | 36.35 |
| **8 - 4L11B** | 1515 | 245.91 | 36.21 |

**Table 3.** Performance results for "coastguard_cif" sequence.

| Coding Structure | Time (sec.) | Bitrate (kbps) | PSNR Y (dB) |
|---|---|---|---|
| **1 - IPPP** | 574 | 1113.35 | 34.55 |
| **2 - IBPBP** | 621 | 864.36 | 34.13 |
| **3 - IBBPBBP** | 644 | 768.90 | 33.62 |
| **4 - 3L3B** | 652 | 709.25 | 33.65 |
| **5 - 3L5B** | 1452 | 686.74 | 33.48 |
| **6 - 3L7B** | 1501 | 677.43 | 33.40 |
| **7 - 4L7B** | 1499 | 664.36 | 33.42 |
| **8 - 4L11B** | 1562 | 650.54 | 33.06 |

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.

# 8. Concluding Remarks

H.264/AVC represents a major step in the development of video coding standards, in terms of both coding efficiency enhancement and flexibility for effective use over a broad variety of network types and application domains. It is based on conventional block-based motion-compensated hybrid video coding concepts, but with some important differences. Among them are enhanced motion prediction capability, use of small block-size exact-match transform, adaptive in-loop de-blocking filter, and enhanced entropy coding methods.

In terms of compression rate, the standard typically outperforms all existing standards by a factor of two especially in comparison to MPEG-2, which is currently the basis for digital TV systems worldwide. Although it is more complex than earlier standards, encoder and decoder optimization studies are promising in both academia and industry. Being the latest, when compared with the other currently available video coding standards, H264 provides good quality with better compression rates. With these features, presently, it is the reference for comparing the performance of any new study in this field.

Another important fact is that H.264/AVC is a public and open standard. However, it should be noticed that, as has been the case in the ITU-T and ISO/IEC video coding standards, only the central decoder is standardized by imposing restrictions on the bit stream, syntax and specification of the decoding process of the syntax elements. Development of the encoder conforming to the standard is still considered to be a challenging issue, particularly for real-time applications such as video-conferencing.

# References

[1] 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.

[2] ISO/IEC JTC1/SC29, "Coding of Audio-Visual Objects", ISO/IEC 14496-2, International Standard:1999/Amd1:2000, 2000.

[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, T. Wiegand, B. Girod, "Multihypothesis Pictures for H.26L", IEEE ICIP 2001, Greece, 2001.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

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

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] X. Sun, S. Li, F. Wu, J. Shen, W. Gao, "The Improved SP Frame Coding Technique for the JVT Standard", IEEE ICIP 2003.

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

[24] 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.

[25] 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.

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

[27] S. Wenger, M. Horowitz, "Flexible MB Ordering – A new Error Resilience Tool for IP-Based Video", IWDC 2002 Conference, Italy, 2002.

[28] 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.

[29] 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.

[30] 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.

[31] T. Özçelebi, A. M. Tekalp, M. R. Civanlar, "Optimal Rate and Input Format Control for Content and Context Adaptive Video Streaming", IEEE ICIP 2004 Conference.