

# MCTF and Scalability Extension of H.264/AVC and its Application to Video Transmission, Storage, and Surveillance

Ralf Schäfer, Heiko Schwarz, Detlev Marpe, Thomas Schierl, and Thomas Wiegand\*  
Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute (HHI),  
Image Processing Department, Einsteinufer 37, 10587 Berlin, Germany

## ABSTRACT

The extension of H.264/AVC hybrid video coding towards scalable video coding (SVC) using motion-compensated temporal filtering (MCTF) is presented. Utilizing the lifting approach to implement MCTF, the motion compensation features of H.264/AVC can be re-used for the MCTF prediction step and extended in a straightforward way for the MCTF update step. The MCTF extension of H.264/AVC is also incorporated into a video codec that provides SNR, spatial, and (similar to hybrid video coding) temporal scalability. The paper provides a description of these techniques and presents experimental results that validate their efficiency. In addition applications of SVC to video transmission and video surveillance are described.

**Keywords:** Scalable video coding, video coding standards, H.264, MPEG-4 AVC

## 1. INTRODUCTION

The scalable extension of H.264/AVC as proposed in [1] has been chosen to be the starting point [2] of MPEG's Scalable Video Coding (SVC) standardization project in October 2004. In January 2005, MPEG and the Video Coding Experts Group (VCEG) of the ITU-T agreed to jointly finalize the SVC project as an Amendment of their H.264/AVC standard [3][4], and the scalable extension of H.264/AVC was selected as the first Working Draft [5]. The Working Draft provides a specification of the bit-stream syntax and the decoding process, the reference encoding process is described in the Joint Scalable Video Model (JSVM0) [6]. The corresponding JSVM 0 software [7] can be downloaded at the following web site: [http://ip.hhi.de/imagecom\\_G1/savce/index.htm](http://ip.hhi.de/imagecom_G1/savce/index.htm).

The basic design idea of the JSVM is to extend the hybrid video coding approach of H.264/AVC towards motion-compensated temporal filtering (MCTF) [8][9] by using a lifting framework [10]. Because lifting is invertible, any motion compensation (MC) technique can be incorporated into the prediction and update steps of the filter bank. By using the highly efficient motion model of H.264/AVC in conjunction with a block-adaptive switching between the Haar and the 5/3 spline wavelet, both the prediction and the update step are similar to MC techniques in the generalized B slices [11][12] of H.264/AVC.

Furthermore, the open-loop structure of a temporal subband representation offers the possibility to efficiently incorporate SNR and spatial scalability. SNR scalability is basically achieved by residual quantization with very little changes to H.264/AVC, and it is also being extended to fine granular SNR scalability. For spatial scalability, a combination of MCTF and over-sampled pyramid decomposition is proposed, which requires some additional mechanisms to convey bit rate from lower resolution to higher resolution layers. However, for each layer, the macroblock-based structure of H.264/AVC can be maintained as will be shown. Because of the similarities in MC, the approach to temporal scalability of H.264/AVC is maintained.

Motivated by these facts, we have investigated the possibility of a simple but yet efficient extension of H.264/AVC hybrid video coding towards MCTF and scalability. The next section describes the MCTF approach. Sec. 3 shows how H.264/AVC is extended towards MCTF and Sec. 4 describes the scalability extensions. Experimental results are provided in Sec. 5. Example applications to video transmission and video surveillance are described in Sec. 6.

---

\* {schaefer, hschwarz, marpe, schierl, wiegand}@hhi.fraunhofer.de

## 2. MOTION-COMPENSATED TEMPORAL FILTERING

In this section, we briefly review MCTF for the case of a 2-tap filter. The generic lifting scheme consists of three steps: polyphase operation, prediction, and update. Figure 1 shows a two-channel filter bank with "P" representing the prediction step and "U" representing the update step.

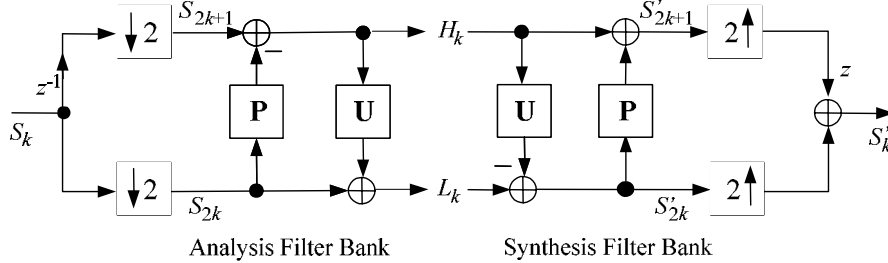


Figure 1: Lifting representation of an analysis-synthesis filter bank.

The input signal  $S_k$  to the analysis filter bank corresponds to a video picture sampled at time instant  $k$ . The polyphase decomposition splits the set of pictures into two sets of pictures with even ( $2k$ ) and odd indices ( $2k+1$ ). The picture  $S_{2k+1}$  is predicted using MC, i.e., spatial shift alignment of picture  $S_{2k}$  towards  $S_{2k+1}$  yielding  $\mathbf{P}(S_{2k})$  and the prediction residual

$$H_k = S_{2k+1} - \mathbf{P}(S_{2k}) \quad (1)$$

The difference between  $S_{2k+1}$  and  $\mathbf{P}(S_{2k})$  is then again motion-compensated, i.e., spatially shift-aligned towards  $S_{2k}$  and divided by 2 yielding  $\mathbf{U}(S_{2k+1} - \mathbf{P}(S_{2k}))$ . When the two MC operators in  $\mathbf{P}(\cdot)$  and  $\mathbf{U}(\cdot)$  are linear and invertible against each other such that  $\mathbf{U}(\mathbf{P}(s)) = s/2$ , then the following applies

$$L_k = S_{2k} + \mathbf{U}(S_{2k+1} - \mathbf{P}(S_{2k})) = \frac{1}{2} S_{2k} + \mathbf{U}(S_{2k+1}) \quad (2)$$

If the MC operators in  $\mathbf{P}(s)$  and  $\mathbf{U}(s)$  do not incur a spatial displacement of  $s$ , the signals  $H_k$  and  $L_k$  represent high-pass and low-pass bands, respectively, in a known way. Otherwise,  $H_k$  and  $L_k$  can be viewed as high-pass and low-pass bands, respectively, but with MC that spatially aligns  $S_{2k}$  and  $S_{2k+1}$  towards each other. It is easy to see that if  $H_k$  and  $L_k$  are not changed (quantized) that the operation of the synthesis filter bank inverts the update, prediction, and polyphase decomposition steps and  $S_k$  is perfectly reconstructed. It should be noted that, when the update step is removed, the presented structure is the same as an open-loop version of a hybrid video codec.

## 3. MCTF EXTENSION OF H.264/AVC

H.264/AVC is a hybrid video codec specifying for macroblocks either motion-compensated prediction [13] or intra prediction [3]. Both predictions are followed by residual coding [14][4]. When lifting is used to implement an MCTF, the prediction and update steps are separate mechanisms. Hence, we use the MC of H.264/AVC for the prediction step and a similar technique for the update step which is derived from the prediction step. Thus the update step also consists of block-based H.264/AVC MC, but with a bit-depth expansion by 1 compared to the prediction step.

### 3.1 Motion Compensation in H.264/AVC

One of the reasons for the improved coding efficiency of H.264/AVC [3] compared to previous standards [15][16][17] is because it permits variable block size MC and multiple reference pictures for MC [18][19]. H.264/AVC specifies block sizes that are signaled through macroblock types for block sizes 16x16, 16x8, 8x16, and 8x8 luma samples. When macroblock type specifies 8x8 blocks, each of these can be split again to 8x4, 4x8, or 4x4 blocks through the sub-macroblock type. For all blocks one or two motion vectors per block can be signaled for MC, corresponding to predictive or bi-predictive MC, respectively. For all blocks smaller than 8x8 samples, the reference picture applies that is chosen for the 8x8 block that contains them. All other blocks can freely choose between the reference pictures and signal the reference picture index together with each motion vector.

### 3.2 Extension Towards MCTF Using the Update Step

We now explain how the prediction in H.264/AVC is combined with the corresponding update step. For that, we apply a notation for video samples where  $s[\mathbf{l}, k]$  is a video sample at spatial location  $\mathbf{l} = (x, y)$  at time instant  $k$ . Let the locations within a block  $\mathbf{B}$  be noted as  $\mathbf{l} \in \mathbf{B}$ . The prediction and update operators for the temporal decomposition using the lifting representation of the Haar wavelet for MCTF and block  $\mathbf{B}$  are given by

$$\mathbf{P}_{Haar}(s[\mathbf{l}, 2k]) = s[\mathbf{l} + \mathbf{m}_{p_0}, 2k - 2r_{p_0}], \quad \mathbf{l} \in \mathbf{B} \quad (3)$$

$$\mathbf{U}_{Haar}(h[\mathbf{l}, k]) = \frac{1}{2}h[\mathbf{l} + \mathbf{m}_{u_0}, k + r_{u_0}], \quad \mathbf{l} \in \mathbf{B} \quad (4)$$

This Haar wavelet corresponds in the prediction step exactly to predictive coding in H.264/AVC using the motion vector  $\mathbf{m}_{p_0}$  and the reference picture index  $r_{p_0}$ . The update step also consists of block-based MC, but with a bit-depth expansion by 1 compared to the prediction step. The algorithm for the derivation of the motion vector  $\mathbf{m}_{u_0}$  and the reference picture index  $r_{u_0}$  is given below. For the 5/3 wavelet, the prediction and update operators are given by

$$\mathbf{P}_{5/3}(s[\mathbf{l}, 2k]) = \frac{1}{2}(s[\mathbf{l} + \mathbf{m}_{p_0}, 2k - 2r_{p_0}] + s[\mathbf{l} + \mathbf{m}_{p_1}, 2k + 2 + 2r_{p_1}]), \quad \mathbf{l} \in \mathbf{B} \quad (5)$$

$$\mathbf{U}_{5/3}(h[\mathbf{l}, k]) = \frac{1}{4}(h[\mathbf{l} + \mathbf{m}_{u_0}, k + r_{u_0}] + h[\mathbf{l} + \mathbf{m}_{u_1}, k - 1 - r_{u_1}]), \quad \mathbf{l} \in \mathbf{B} \quad (6)$$

Again, the prediction step is exactly the same as bi-predictive MC in H.264/AVC. Note that the prediction utilizes two lists of indices to reference pictures. These lists are named list 0 ( $\mathbf{m}_{p_0}$  and  $r_{p_0}$ ) and list 1 ( $\mathbf{m}_{p_1}$  and  $r_{p_1}$ ) and may contain the same or different reference pictures.

The derivation of motion vectors and reference picture indices in the update step works as follows. The design goals of the algorithm are to derive a set of H.264/AVC motion vectors and reference picture indices for the update step that can be the input to the H.264/AVC motion compensation process without having to change it while achieving highest coding efficiency. The algorithm determines for each  $4 \times 4$  luma block  $\mathbf{B}_{4 \times 4}$  in the picture  $\mathbf{U}(H_k)$  the motion vectors and reference picture indices. For each block  $\mathbf{B}_{4 \times 4}$ , all motion vectors  $\mathbf{m}_{p_0}$  and  $\mathbf{m}_{p_1}$  are evaluated that point into this block. Those  $\mathbf{m}_{p_0}$  and  $\mathbf{m}_{p_1}$  are selected that use the maximum number of samples as a reference out of the block  $\mathbf{B}_{4 \times 4}$  and the update motion vectors are given as  $\mathbf{m}_{u_0} = -\mathbf{m}_{p_0}$  and  $\mathbf{m}_{u_1} = -\mathbf{m}_{p_1}$ . The reference indices  $r_{u_0}$  and  $r_{u_1}$  are specifying those pictures into which MC is conducted using  $\mathbf{m}_{p_0}$  and  $\mathbf{m}_{p_1}$ , respectively. When no motion vectors  $\mathbf{m}_{p_0}$  exist that point into  $\mathbf{B}_{4 \times 4}$  or when not more than  $\frac{3}{4}$  of the samples of  $\mathbf{B}_{4 \times 4}$  are used as reference for MC using any  $\mathbf{m}_{p_0}$ , the update step using  $\mathbf{m}_{u_0}$  is omitted for  $\mathbf{B}_{4 \times 4}$ . The same conditions apply to  $\mathbf{m}_{p_1}$  and  $\mathbf{m}_{u_1}$  as well. After processing all blocks  $\mathbf{B}_{4 \times 4}$ , the determined  $\mathbf{m}_{u_0}$ ,  $\mathbf{m}_{u_1}$ ,  $r_{u_0}$ , and  $r_{u_1}$  are formatted into H.264/AVC syntax and syntax limitations are applied if necessary. Please note that the above algorithm is simultaneously applied at coder and decoder so that each step is exactly specified. No side information is transmitted for the update step.

The H.264/AVC de-blocking filter [21] is applied to the low-pass pictures that are reconstructed in the prediction steps.

### 3.3 Intra Coding in MCTF

When MC does not work, e.g. for scene cuts or uncovered background, the incorporation of intra coding modes increases coding efficiency. For the intra macroblock, the corresponding prediction or update step is skipped and the original macroblock samples are placed into the high-pass pictures  $H_k$  and coded using the intra coding tools of H.264/AVC. Note, that these intra samples are set to zero before they are used for MC in the update steps.

### 3.4 Temporal Coding Structure

The temporal coding structure of MCTF is changed relative to hybrid video coding in that not only high-pass pictures  $H_k$  (prediction residuals) are resulting from the prediction step but also low-pass pictures  $L_k$  are resulting from the update step. Typically, a group of  $N_0$  input pictures is partitioned into two sets of pictures with one set containing  $N_A$  ( $0 < N_A < N_0$ ) input pictures and the other set containing  $N_B = N_0 - N_A$  input pictures. The pictures of the first set are labeled as pictures  $A_k$  and the pictures of the second set are labeled as pictures  $B_k$ . The decomposition is performed in a way that the high-pass pictures  $H_k$  are spatially shift-aligned with pictures  $B_k$  and the low-pass pictures  $L_k$  are spatially shift-aligned with pictures  $A_k$ . Note, that for generating the high-pass pictures in the prediction step, only the input pictures  $A_k$  can be used as reference pictures for predicting an input picture  $B_k$ . Figure 2 provides examples for temporal decompositions.

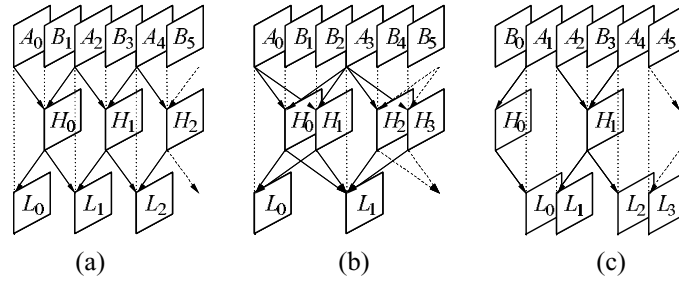


Figure 2: Temporal decomposition of input pictures into low and high-pass pictures: a)  $N_A=N_B$ , b)  $N_A=2N_B$ , c)  $2N_A=N_B$ .

For groups of  $N_0 > 2$  pictures it is in general advantageous to apply a multi-channel decomposition instead of a two-channel decomposition. Therefore, the presented two-channel decomposition is iteratively applied to the set the low-pass pictures until a single low-pass picture is obtained or a given number of decomposition stages is performed. In Figure 3(a), a general dyadic temporal decomposition for a group of 16 pictures (GOP) is illustrated. The picture 0 refers to the last (low-pass) picture of the previous GOP, which is used as additional reference picture in the prediction steps, but not altered in the update steps. For clarity, only the prediction and update steps using directly neighboring reference pictures are illustrated.

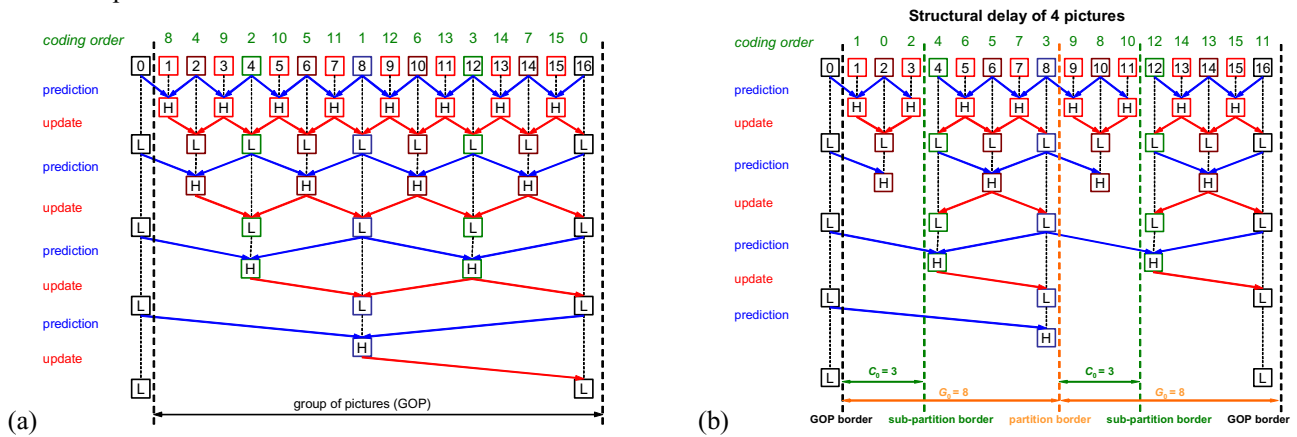


Figure 3: Dyadic temporal decomposition of a group of 16 pictures: a) without delay constraints, b) with a maximum structural encoding-decoding delay of 4 pictures at the highest temporal resolution.

The delay introduced by each decomposition stage is coupled with the use of reference pictures that are displayed later than the predicted or updated picture. Therefore, if the reference pictures for the prediction step are in the past relative to the predicted picture and the update step is omitted, no additional delay is introduced. This allows controlling the delay for large groups of pictures or in other words, the low-pass pictures of groups of pictures which correspond to the maximum tolerated delay can be transmitted using motion-compensated prediction as in hybrid video coding. For that, in the prediction steps, the low-pass picture of the previous GOP that is obtained after performing all  $n$  decomposition stages is used as additional reference picture for motion-compensated prediction of the current group of pictures. The motion-compensated update is only performed inside the GOP. Figure 3(b) shows an example for the dyadic decomposition of a group of 16 pictures with a maximum structural encoding-decoding delay of 4 pictures at the highest temporal resolution. In order to meet the delay constraints, the GOP is partitioned into subgroups, and neither backward prediction steps nor update steps are allowed across the partition boundaries. For more details, the reader is referred to [6].

### 3.5 Impact on H.264/AVC Syntax, Decoding, and Encoding

With the exception of some high-level syntax elements for arbitrarily controlling the reference list for the update steps, the syntax of H.264/AVC is not affected by the MCTF extension since all data for the update step are derived from data that are already present in the bit-stream. Moreover, the concept of generalized B pictures in H.264/AVC allows the

hierarchical temporal decomposition but without the update step. Later, we will refer to this concept as *hierarchical B pictures*. The decoding process of H.264/AVC needs to be extended for the update step by the following:

- The derivation process for the motion vectors in the update step must be specified.
- The motion compensation for the update step requires a bit-depth expansion by one bit. In theory this bit-depth expansion happens with every level of the decomposition hierarchy. However, we have found that disallowing any further bit-depth expansion by clipping the low-pass pictures after the update steps does not affect performance.

We have employed Lagrangian methods for the encoding process similar to those in H.264/AVC [20]. However, the open-loop characteristic of the analysis filter bank and the hierarchical temporal decomposition make a straightforward re-use of these techniques difficult. Due to the update steps, the encoding process needs to be operated in reverse order of the decoding process. The Lagrangian costs used for determining the coding modes are based on original reference pictures and do thus only present an even less accurate estimate of the actual costs compared to the approach in [20].

These normalization factors of the analysis-synthesis filter bank are taken into account during quantization. In general, the low-pass pictures are coded with the highest fidelity, since they are employed for motion-compensated prediction of all other pictures. The quantization parameter differences from one decomposition level to the next are determined based on the number of samples for which prediction, bi-prediction, or intra coding is employed.

## 4. SCALABILITY EXTENSION OF H.264/AVC

### 4.1 Temporal Scalability

The temporal decomposition as described in Sec. 3.4 permits temporal scalability in a similar way as in hybrid video coding. The scalability is achieved by removing those bit-stream parts that correspond to pictures that are not reference pictures for the remaining pictures.

### 4.2 SNR Scalability

For the SNR base layer, H.264/AVC-compatible transform coding is used. The high-pass pictures contain intra or residual macroblocks as in hybrid video coding. For the residual macroblocks, the coding as in H.264/AVC including transformation and quantization is employed. The intra macroblocks are coded using the intra coding modes of H.264/AVC. For each macroblock, the coded block pattern (CBP), and conditioned on CBP the corresponding residual blocks are transmitted together with the macroblocks modes, intra prediction modes, reference picture indices and motion vectors using the B or P slice syntax of H.264/AVC. Low-pass pictures are either coded independently of each other as H.264/AVC intra pictures or are inter coded as H.264/AVC inter pictures.

On top of the SNR base layer, SNR enhancement layers are coded. For that, the quantization error between the SNR base layer and the original subband pictures is re-quantized exactly using the same methods as for the base layer but with a finer quantization step size, i.e., a lower value of the quantization parameter. This enhancement layer together with the base layer can be considered to be the base layer for another enhancement layer and the same methods can then be applied again. The reasons why this simple approach to SNR scalability shows good coding efficiency in case enhancement layers are removed is the open-loop encoding method and the temporal decomposition as presented above.

In a simple version, the transform coefficient levels of the SNR enhancement layers are transmitted using the residual syntax of H.264/AVC. With this approach only coarse grains of scalable SNR layers can be efficiently represented as factors of 2 in bit-rate. In order to support fine granular SNR scalability, we have introduced so-called progressive refinement slices (for details, see [1][5][6]). Each NAL unit for a progressive refinement slice represents a refinement signals that corresponds to a bisection of the quantization step size. These signals are represented in a way that only a single inverse transform has to be performed for each transform block at the decoder side. The progressive refinement NAL units can be truncated at any arbitrary point, so that the quality of the SNR base layer can be improved in a fine granular way. Therefore, the coding order of transform coefficient levels has been modified. Instead of scanning the transform coefficients macroblock by macroblock as it is done in “normal” slices, the transform coefficient blocks are scanned in several paths, and in each path only a few coding symbols for a transform coefficient block are coded. With exception of the modified coding order, the CABAC entropy coding [22] as specified in H.264/AVC is re-used.

### 4.3 Spatial Scalability

We consider spatial scalable coding of video at multiple resolutions (e.g. QCIF, CIF, and 4CIF) with a factor of 2 in horizontal and vertical resolution. We have represented the video signal using an oversampled pyramid and code the various spatial resolutions independently of each other. From this experiment we have found that it clearly depends on the chosen bit rates in conjunction with the sequence characteristics to what extent the coding efficiency of a spatial layer (e.g. the 4CIF layer) is affected by the presence of additional lower spatial resolution layers (e.g. QCIF and CIF layers). We have also found that it would be efficient to allow the encoder to freely choose which dependencies between the spatial resolution layers need to be exploited through switchable prediction mechanisms. For that, the following techniques turned out to provide gains and are described below:

- Prediction of a macroblock using the up-sampled lower resolution signal
- Prediction of motion vectors using the up-sampled lower resolution motion vectors
- Prediction of the residual signal using the up-sampled residual signal of the lower resolution layer

For prediction of motion vectors using the up-sampled lower resolution signal, we introduced two additional macroblock modes that utilize motion information of the lower resolution layer. The macroblock partitioning is obtained by up-sampling the partitioning of the corresponding 8x8 block of the lower resolution layer. For the obtained macroblock partitions, the same reference picture indices as for the corresponding sub-macroblock partition of the base layer block are used; and the associated motion vectors are scaled by a factor of 2. While for the first of these macroblock modes no additional motion information is coded, for the second one, a quarter-sample motion vector refinement is transmitted for each motion vector. Additionally, our approach includes the possibility to use a scaled motion vector of the lower resolution as motion vector predictor for the conventional motion-compensated macroblock modes. A flag that is transmitted with each motion vector difference indicates whether the motion vector predictor is build by conventional spatial prediction or by the corresponding scaled base layer motion vector.

In order to also incorporate the possibility of exploiting the residual information coded in the lower resolution layer, an additional flag is transmitted for each macroblock, which signals the application of residual signal prediction from the lower resolution layer. If the flag is true, the base layer residual signals is block-wise up-sampled using a bi-linear filter with constant border extension and used as prediction for the residual signal of the current layer, so that only the corresponding difference signal is coded.

In order to enable the inter-layer prediction of low-pass signals, we introduced an additional intra macroblock mode. In that coding mode, the prediction signal is generated by up-sampling the reconstruction signal of the lower resolution layer using the 6-tap filter which is defined in H.264/AVC for the purpose of half-sample interpolation. The prediction residual is transmitted using the H.264/AVC residual coding. For this prediction process it is required in general that the base layer signal is decoded by the computationally complex operations of inverse MCTF and de-blocking. We have found that this problem can be circumvented by restricting the prediction from up-sampled decoded pictures to those parts of the lower layer picture that are coded with intra macroblocks. For that, the intra prediction signal is directly obtained by de-blocking and up-sampling the corresponding 8x8 luma block inside the corresponding lower layer high-pass picture. With the proposed changes, the decoding complexity is significantly reduced, since the inverse MCTF is only required for the spatial layer that is actually decoded.

Note, that all described inter-layer prediction techniques can also be applied when the base layer has the same spatial resolution as the current layer, i.e., for coarse grain SNR scalability with optimized motion parameters for each SNR layer. In this case, the up-sampling operations are simply discarded.

### 4.4 Combined Scalability

The basic coding scheme for achieving a wide range of combined spatio-temporal and quality scalability can be classified as layered video codec. The coding structure depends on the scalability space that is required by the application. In Figure 4, a block diagram for a typical scenario with 2 spatial layers is depicted.

In each layer, an independent MCTF with layer-specific motion parameters is employed. This hierarchical structure provides a temporal scalable representation of a sequence of input pictures that is also suitable for efficiently

incorporating spatial and quality scalability. The redundancy between different layers is exploited by different inter-layer prediction concepts that include prediction mechanisms for motion parameters as well as texture data. A base representation of the input pictures of each layer is obtained by transform coding similar to that of H.264/AVC, the corresponding NAL units (NAL – Network Abstraction Layer) contain motion information and texture data. These base representations determine the minimum bit-rate at which a spatio-temporal resolution can be decoded. The NAL units of the base representation of the lowest layer are compatible with standard H.264/AVC. The reconstruction quality of a layer can be improved by an additional coding of so-called progressive refinement slices. These NAL units represent refinements of the texture data (intra and residual data). Since the coding symbols of progressive refinement slices are ordered similar to a coarse-to-fine description, these NAL units can be arbitrarily truncated in order to support fine granular quality scalability (FGS) or flexible bit-rate adaptation.

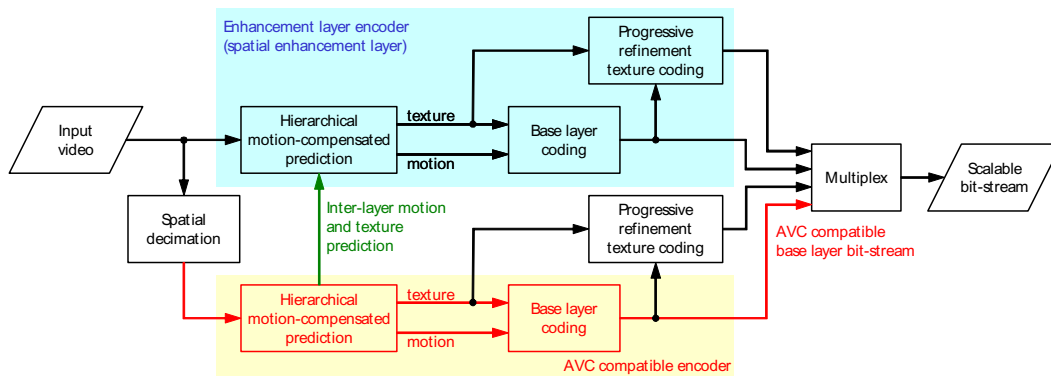


Figure 4: Codec structure example for the scalable extension of H.264/AVC.

Bit-streams for a reduced spatial and/or temporal resolution can be simply obtained by discarding NAL units (or network packets) from a global bit-stream that are not needed for decoding the spatio-temporal target resolution. NAL units that correspond to progressive refinement slices can also be arbitrarily truncated in order to further reduce the bit-rate and the associated reconstruction quality

#### 4.5 Impact on H.264/AVC Syntax, Decoding, Encoding

Scalability requires some high-level syntax support to allow the efficient removal of parts of the bit-stream. The packet-based access unit and NAL unit concept of H.264/AVC is well suited to provide such support. We have introduced additional NAL unit types to indicate the presence of enhancement layers such as an SNR or spatial enhancement layer. Moreover, enhancement slice types need to be specified to facilitate the inter-layer prediction allowing the inclusion of data from one or more lower layers into the current layer such as reconstructed samples, motion vectors, or decoded prediction residual samples.

For the decoding of SNR enhancement layers, an additional process for importing motion data as well as intra and residual signals of the subordinate layer is required. The decoding process for spatial enhancement layers needs to be extended by

- a process for importing and up-sampling of the reconstructed lower resolution signal,
- a process for importing and up-sampling of residual signals of the lower resolution layer,
- a process for importing and scaling motion information of the lower resolution layer.

In addition, the parsing process needs to be adapted to the modified and newly introduced syntax elements, and the motion vector decoding needs to be extended by the motion vector prediction from the base layer.

The encoding process must consider the entire range of rate-distortion points that the decoder may choose to decode. The encoder optimization therefore needs to carefully trade-off mainly motion data and prediction residuals. Since several SNR layers of a spatio-temporal resolution employ a single motion vector field, the trade-off between motion and residual data needs to be adjusted for the entire supported bit-rate range.

## 5. EXPERIMENTAL RESULTS

### 5.1 Results for the MCTF Extension

For evaluating the coding efficiency of the both single-layer MCTF extension and hierarchical B pictures, we compared it to a closed-loop H.264/AVC coder using a similar degree of encoder optimizations [12]. In Figure 5, diagrams with rate distortion curves for the sequences “Mobile” and “Football” are depicted. For the H.264/AVC reference, only the first picture is encoded as IDR picture, all following pictures are coded as P and B pictures. Five reference pictures are used, and the rate-distortion curves have been obtained by varying the quantization parameter (QP), where the QP for B pictures was increased by 2 in comparison to the QP for I and P pictures. The GOP size of the MCTF extension was set to 32 pictures and two reference pictures have been used. CABAC was used as entropy coding method for all encoders.

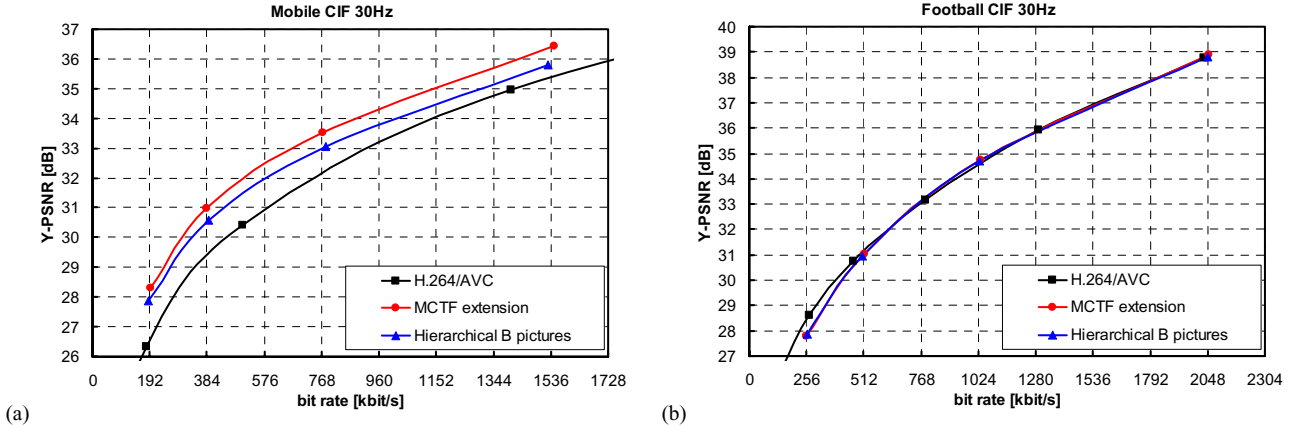


Figure 5: Coding efficiency of the MCTF extension in comparison to a closed-loop H.264/AVC coder for the sequences “Mobile” (a) and “Football” (b).

For the “Mobile” sequence, which shows smooth motion, the coding efficiency in comparison H.264/AVC with IBBPBBP... coding is improved by both the MCTF extension and hierarchical B pictures. For the latter, the coding gains are the result of the changed temporal decomposition structure together with the cascading of quantization parameters. The usage of the update step further increases the coding efficiency and reduces the PSNR fluctuations inside a group of pictures as illustrated in Figure 6. The “Football” sequence is characterized by strong local motion. For this sequence, the coding efficiency of both the MCTF extension and the hierarchical B picture approach is similar to that of the H.264/AVC reference with the IPPPP... temporal coding structure.

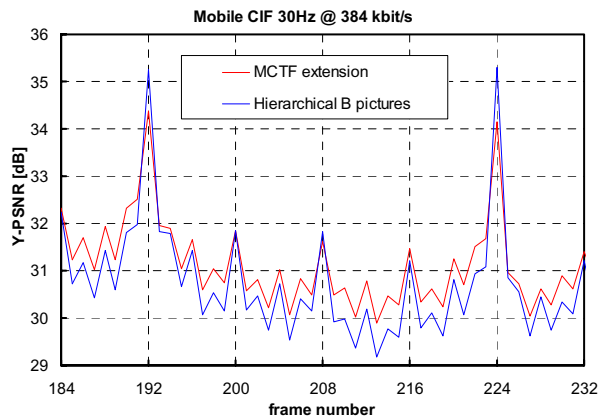
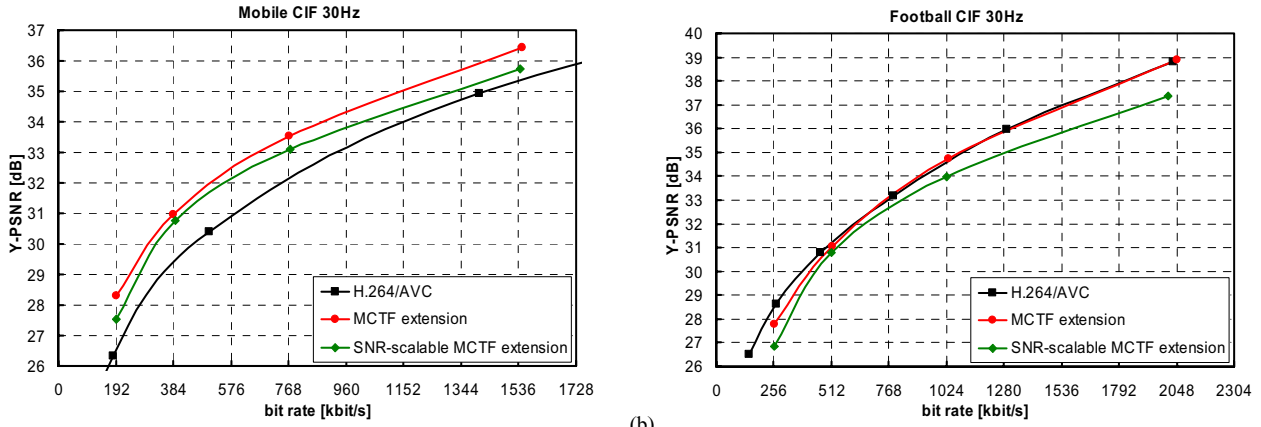


Figure 6: PSNR fluctuations of both the MCTF extension and hierarchical B pictures for “Mobile” coded at 384 kbit/s.

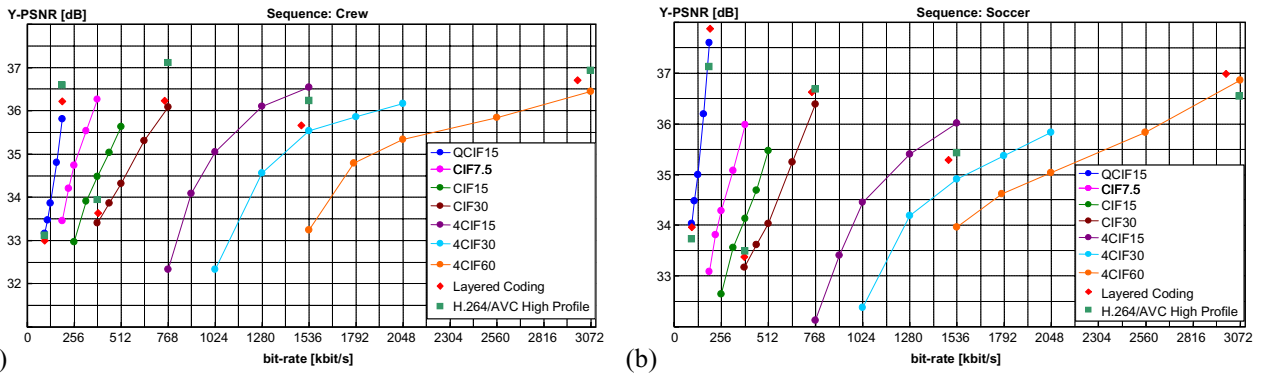
## 5.2 Results for the Scalability Extension

In Figure 7, the coding efficiency of the coarse grain SNR-scalable MCTF extension (without the usage of progressive refinement slices) is compared to the coding efficiency of the single-layer MCTF extensions and the H.264/AVC references. While all rate-distortion points for the H.264/AVC reference and the single layer MCTF coder represent different bit-streams, all rate-distortion points for the scalable codec have been obtained by selecting and decoding NAL units of a single scalable bit-stream.



(a) Figure 7: Coding efficiency of the SNR-scalable (coarse grain) MCTF extension for the sequences “Mobile” (a) and “Football” (b).

As it can be seen in the rate-distortion plots, the coding efficiency of the SNR-scalable MCTF extension is 0.2 to 1.6 dB worse than that of the single-layer version. This coding efficiency loss is related to the fact that for the SNR-scalable codec, a single motion field is used for all rate points, while for the single layer version, the trade-off between motion and residual data is optimized for each bit-rate point. This also explains why the PSNR losses for the “Football” sequence are larger, since this sequence is characterized by strong local motion.



(a) Figure 8: Coding efficiency of the spatio-temporal-SNR scalable MCTF extension for the sequences “Crew” (a) and “Soccer” (b).

The coding efficiency of the MCTF extension for the support of a large degree of combined spatio-temporal-SNR scalability was evaluated for the sequences “Crew” and “Soccer” and the results are depicted in Figure 8. For this experiment, the progressive refinement slices have been used for providing an entire range of extractable bit-rates for each supported spatio-temporal resolution. The diagrams contain also the case of layered coding, in which only 6 spatio-temporal-SNR points are provided, and no fine grain SNR scalability is supported. For further information, results obtained by H.264/AVC High Profile have been additionally plotted in the diagrams (green squares). These results are provided for the spatial and temporal resolutions and bit rates that are supported by the layered coding. However, in contrast to the combined scalability and layered coding cases, the classical IBBPBBP... coding temporal structure is

used and some of the differences can be accounted to that. Additionally, the green squares case is also non-scalable. Note that the scalability extension of H.264/AVC also builds on top of High Profile and the optimization approach and the other parameters including motion search range are similar for all three cases following [20]. While for the sequence “Soccer” the results of layered coding and H.264/AVC are similar, for the Crew sequence, H.264/AVC High Profile provides additional improvements indicating that further work is needed towards an efficient scalable representation.

## 6. APPLICATIONS OF SVC

### 6.1 Video Transmission

The hierarchical scalable video coding approach in general allows the split of the bit-stream into a base layer and various enhancement layers. That allows the transmission of these partial bit-streams via different channels or in different network streams. In IP networks the different quality classes can be assigned to one or even more consumed network streams, thus a possible billing depending on network streams is guaranteed. Such network types could be DVB-H or MBMS of 3GPP, with different terminal capability classes.

The layered coding approach has further benefits. If the network supports transmission of certain network streams to certain devices only, not even all streams have to be received by all terminals. In MBMS or DVB-H networks it should be possible to send out, e.g., the base layer of a scalable H.264/AVC stream to the low performance device only and guarantee that all layers of the stream reach the high performance terminal only. Such an example is shown in Figure 9. By using the scalable H.264/AVC stream instead of simulcasting, the backbone could be drastically eased.

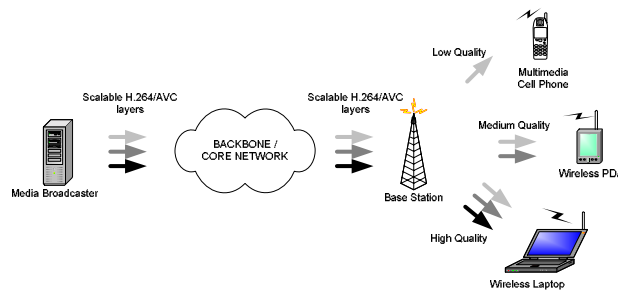


Figure 9: Scalable H.264/AVC video transmission in wireless broadcast networks.

Another benefit can be achieved, if the protection of the scalable H.264/AVC layers is treated in different ways. Forward error correction (FEC) is often used to protect data sent out via wireless broadcast channels. Unequal error or erasure protection (UEP/UXP) schemes [23] can be used to ensure an error free transmission of important layers like, e.g., the base layer of a scalable H.264/AVC stream. UEP/UXP can be used on top of the already existing channel FEC. Such a scheme can guarantee a basic video quality over a large range of channel error rates. Such a scenario is shown in Figure 10.

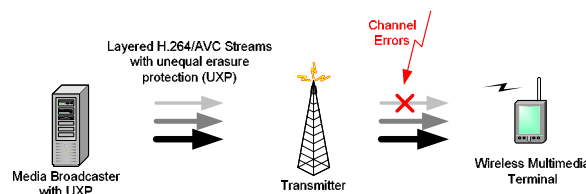


Figure 10: Unequal erasure protection with scalable H.264/AVC quality layers.

### 6.2 Video Surveillance

A very promising application of scalable coding is video surveillance. Typically videos from many cameras have to be stored and viewed on diverse displays, which may have different spatial and temporal resolutions (Figure 11). Examples are split screen display of many scenes on one monitor or viewing of scenes from dedicated cameras on mobile devices

such as video phones or PDAs. For such applications scalable coding is most attractive, because no transcoding or format conversion is required. Decoding of the lower resolutions videos for split-screen display also saves computing power, therefore many videos can be decoded and displayed with the computing power required for one full resolution video.



Figure 11: General networked surveillance scenario.

In general, the requirements for surveillance applications can be summarized as follows:

- Simultaneous transmission and storage of different spatial and temporal resolutions in one bit stream, in order to feed different displays
- Fine granularity scalability for feeding different transmission links of varying capacity
- Decoding of lower resolutions should be less complex for split screen display
- Multiple adaptation of the bit stream should be possible for erosion storage

The latter requirement arises from the necessity to store a huge amount of data delivered by the surveillance cameras. By using scalable coding it becomes possible to delete higher resolution layers of stored scenes after certain expiration times and to keep just a lower resolution copy for the archive. This allows a much more flexible usage of storage capacity without the necessity for re-encoding and copying. Typically the full resolution video is kept for 1-3 days, a medium quality (reduced temporal or spatial resolution) video is kept for one week and a low quality (reduced temporal and spatial resolution) video is kept for long time archiving. This functionality results in the following requirements for a file format, which are mostly fulfilled by the MPEG-4 file format [24]:

- random access to the different layers must be possible without parsing the bit stream
- an appropriate hierarchical storage structure
- a hinting track (e.g. RTP) to support streaming applications

The prototype of a bit-stream splitter in connection with a real time decoder based on the current reference model for SVC has been developed at the HHI. This prototype allows real time extraction and decoding of partial bit streams from a CIF video. A live demonstration of the system will be given during the presentation of the paper.

## 7. CONCLUSIONS

The single-layer MCTF extension of H.264/AVC does provide for some sequences advantages in coding efficiency up to 0.5 dB in terms of objective PSNR measures. Subjectively, a reduction of quality fluctuation is achieved by the update step. The extension for coarse grain SNR scalability in conjunction with MCTF is quite efficient in a straightforward way. The prediction methods used in spatial scalability work sequence dependent and are subject to future refinements. Furthermore, fine granular SNR scalability is provided by introducing enhancement NAL units that contain a refinement signal for a subband picture in a coarse-to-fine representation and can be truncated at any arbitrary point. The simulation results show that the coding efficiency of this approach is only slightly worse than that of the layered representation, while it provides a much larger set of decodable spatio-temporal-rate points.

## ACKNOWLEDGEMENT

The authors thank Tobias Hinz and Heiner Kirchhoffer for their help in implementing some features of the coder.

## REFERENCES

1. H. Schwarz, T. Hinz, H. Kirchhoffer, D. Marpe, and T. Wiegand, "Technical Description of the HHI proposal for SVC CE1," ISO/IEC JTC1/SC29/WG11, Document M11244, Palma de Mallorca, Spain, Oct. 2004.
2. ISO/IEC JTC1/SC29/WG11, "Scalable Video Model 3.0," ISO/IEC JTC1/SC29/WG11, Document N6716, Palma de Mallorca, Spain, Oct. 2004.
3. ITU-T Recommendation H.264 & ISO/IEC 14496-10 AVC, "Advanced Video Coding for Generic Audiovisual Services," (version 1: 2003, versions 2: 2004) version 3: 2005.
4. T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. CSVT, vol. 13, no. 7, pp. 560-576, July 2003.
5. Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, "Scalable Video Coding – Working Draft 1," Joint Video Team, Document JVT-N020, Jan. 2005.
6. Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, "Joint Scalable Video Model JSVM0," Joint Video Team, Document JVT-N021, Jan. 2005.
7. Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, "JSVM 0 software," Joint Video Team, Doc. JVT-N022r1, Jan. 2005.
8. J.-R. Ohm, "Complexity and delay analysis of MCTF interframe wavelet structures," ISO/IEC JTC1/SC29/WG11, Document M8520, Jul. 2002.
9. M. Flierl, "Video Coding with Lifted Wavelet Transforms and Frame-Adaptive Motion Compensation," Proc. of VLBV, Sep. 2003.
10. W. Sweldens, "A custom-design construction of bi-orthogonal wavelets," J. Appl. Comp. Harm. Anal., vol. 3, pp. 186-200, 1996.
11. M. Flierl, T. Wiegand, and B. Girod, "A Locally Optimal Design Algorithm for Block-Based Multi-Hypothesis Motion-Compensated Prediction", in Data Compression Conference, Snowbird, USA, Mar. 1998.
12. M. Flierl and B. Girod. "Generalized B Pictures and the Draft JVT/H.264 Video Compression Standard", IEEE Trans. CSVT, vol. 13, pp. 587-597, Jul. 2003.
13. T. Wedi, "Motion Compensation in H.264/AVC," IEEE Trans. CSVT, vol. 13, pp. 577-586, Jul. 2003.
14. H. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky: "Low-Complexity Transform and Quantization in H.264/AVC," IEEE Trans. CSVT, vol. 13, pp. 598-603, Jul. 2003.
15. ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video," ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2), Nov. 1994.
16. ITU-T, "Video coding for low bit rate communication," ITU-T Recommendation H.263; version 1, Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.
17. ISO/IEC JTC1, "Coding of audio-visual objects – Part 2: Visual," ISO/IEC 14496-2 (MPEG-4 visual version 1), April 1999; Amendment 1 (version 2), February, 2000; Amendment 4 (streaming profile), January, 2001.
18. T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, pp. 70-84, Feb. 1999.
19. T. Wiegand and B. Girod, "Multi-frame Motion-Compensated Prediction for Video Transmission," Kluwer Academic Publishers, Sep. 2001.
20. T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan: "Rate-Constrained Coder Control and Comparison of Video Coding Standards," IEEE Trans. CSVT, vol. 13, pp. 688-703, Jul. 2003.
21. P. List, A. Joch, J. Lainema, G. Bjøntegaard, M. Karczewicz: "Adaptive Deblocking Filter," IEEE Trans. CSVT, vol. 13, pp. 614-619, Jul. 2003.
22. D. Marpe, H. Schwarz, and T. Wiegand: "Context-Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," IEEE Trans. CSVT, vol. 13, pp. 620-636, Jul. 2003.
23. G. Liebl, M. Wagner, J. Pandel, W. Weng, "An RTP Payload Format for Erasure-Resilient Transmission of Progressive Multimedia Streams," Document draft-ietf-avt-uxp-07.txt, IETF, Oct. 2004.
24. ISO/IEC JTC1/SC 29/WG11 N2501 + COR 1 + AMD 1, ISO/IEC 14496-1:2000(E), "Information technology – Coding of audio-visual objects – Part 1: Systems".