MCTF AND SCALABILITY EXTENSION OF H.264/AVC

Heiko Schwarz, Detlev Marpe, and Thomas Wiegand

Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute, Image Processing Department
Einsteinufer 37, D-10587 Berlin, Germany, [hschwarz,marpe,wiegand]@hhi.fhg.de

ABSTRACT

The extension of H.264/AVC hybrid video coding towards motion-compensated temporal filtering (MCTF) and scalability 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.

1. INTRODUCTION

In recent years, various approaches to MCTF-based video coding have been presented (e.g. see [1][2]). The main reason for the advances in coding efficiency of these codecs is the possibility to use techniques for motion compensation (MC) known from hybrid video coding through the lifting representation of filter banks [3]. These filter banks are typically cascaded sequences of (motion-compensated) prediction and (motion-compensated) update steps.

Because lifting is invertible, any 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 [4][5] in connection 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 generalized B slices 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 achieved by residual quantization with very little changes to H.264/AVC. 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. Section 3 shows how H.264/AVC is extended towards MCTF. Section 4 describes the scalability extensions. Section 5 provides experimental results.

2. MCTF

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. Fig. 1 shows a two-channel filter bank with "P" representing the prediction step and "U" representing the update step.

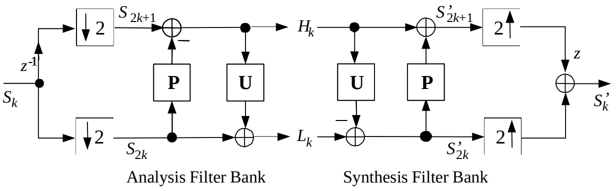


Fig. 1: Lifting representation of an analysis-synthesis filter bank.

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

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

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 **P**( ) and **U**( ) are linear and invertible against each other such that $\mathbf{U}(\mathbf{P}(s)) = s/2$, then the following applies

$$L_k = S_{2k} + \mathbf{U}(S_{2k+1} - \mathbf{P}(S_{2k})) = ½\, S_{2k} + \mathbf{U}(S_{2k+1}) \tag{2}$$

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. The following two cases should be noted and will be referred to later:

1. When the update step is removed, the presented structure is the same as an open-loop version of a hybrid video codec.
2. When the MC in **P**( ) and **U**( ) are not invertible against each other, the low-pass band may contain unwanted signal components.

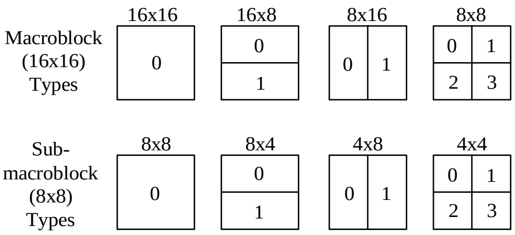


Fig. 2: Segmentations of the macroblock for MC. Top: segmentation of macroblocks, bottom: segmentation of 8x8 partitions.

3. MCTF EXTENSION OF H.264/AVC

3.1 Motion Compensation in H.264/AVC

One of the reasons for the improved coding efficiency of H.264/AVC [6] compared to previous standards is because it permits variable block size MC and multiple reference pictures for MC. Fig. 2 shows the specified 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

$$\mathbf{P}_{Haar}\left(s[\mathbf{l}, 2k]\right) = s[\mathbf{l} + \mathbf{m}_{\mathrm{P0}}, 2k - 2r_{\mathrm{P0}}], \quad \mathbf{l} \in \mathbf{B} \qquad (3)$$

$$\mathbf{U}_{Haar}\left(h[\mathbf{l}, k]\right) = \tfrac{1}{2} h[\mathbf{l} + \mathbf{m}_{\mathrm{U0}}, k + r_{\mathrm{U0}}], \quad \mathbf{l} \in \mathbf{B} \qquad (4)$$

This Haar wavelet corresponds in the prediction step exactly to predictive coding in H.264/AVC using the motion vector $\mathbf{m}_{\mathrm{P0}}$ and the reference picture index r_{P0}. 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 $\mathbf{m}_{\mathrm{U0}}$ and the reference picture index r_{U0} is given below.

For the 5/3 spline wavelet, the prediction and update operators for block $\mathbf{B}$ are given by

$$\begin{aligned}\mathbf{P}_{5/3}\left(s[\mathbf{l}, 2k]\right) = \tfrac{1}{2}(&s[\mathbf{l} + \mathbf{m}_{\mathrm{P0}}, 2k - 2r_{\mathrm{P0}}] + \\ &s[\mathbf{l} + \mathbf{m}_{\mathrm{P1}}, 2k + 2 + 2r_{\mathrm{P1}}]), \quad \mathbf{l} \in \mathbf{B}\end{aligned} \qquad (5)$$

$$\begin{aligned}\mathbf{U}_{5/3}\left(h[\mathbf{l}, k]\right) = \tfrac{1}{4}(&h[\mathbf{l} + \mathbf{m}_{\mathrm{U0}}, k + r_{\mathrm{U0}}] + \\ &h[\mathbf{l} + \mathbf{m}_{\mathrm{U1}}, k - 1 - r_{\mathrm{U1}}]), \quad \mathbf{l} \in \mathbf{B}\end{aligned} \qquad (6)$$

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 ($\mathbf{m}_{\mathrm{P0}}$ and r_{P0}) and list 1 ($\mathbf{m}_{\mathrm{P1}}$ and r_{P1}) 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 4x4 luma block $\mathbf{B}_{4\mathrm{x}4}$ in the picture $\mathbf{U}(H_k)$ the motion vectors and reference picture indices. For each block $\mathbf{B}_{4\mathrm{x}4}$, all motion vectors $\mathbf{m}_{\mathrm{P0}}$ and $\mathbf{m}_{\mathrm{P1}}$ are evaluated that point into this block. Those $\mathbf{m}_{\mathrm{P0}}$ and $\mathbf{m}_{\mathrm{P1}}$ are selected that use the maximum number of samples as a reference out of the block $\mathbf{B}_{4\mathrm{x}4}$ and the update motion vectors are given as $\mathbf{m}_{\mathrm{U0}} = -\mathbf{m}_{\mathrm{P0}}$ and $\mathbf{m}_{\mathrm{U1}} = -\mathbf{m}_{\mathrm{P1}}$. The reference indices r_{U0} and r_{U1} are specifying those pictures into which MC is conducted using $\mathbf{m}_{\mathrm{P0}}$ and $\mathbf{m}_{\mathrm{P1}}$, respectively.

When no motion vectors $\mathbf{m}_{\mathrm{P0}}$ exist that point into $\mathbf{B}_{4\mathrm{x}4}$ or when not more than ¾ of the samples of $\mathbf{B}_{4\mathrm{x}4}$ are used as reference for MC using any $\mathbf{m}_{\mathrm{P0}}$, the update step using $\mathbf{m}_{\mathrm{U0}}$ is omitted for $\mathbf{B}_{4\mathrm{x}4}$. The same conditions apply to $\mathbf{m}_{\mathrm{P1}}$ and $\mathbf{m}_{\mathrm{U1}}$ as well. The condition involving the ¾ of the samples of $\mathbf{B}_{4\mathrm{x}4}$ is based on the empirical observation that if the motion between the prediction and update step is not invertible, unwanted artifacts are introduced and the update step should be avoided.

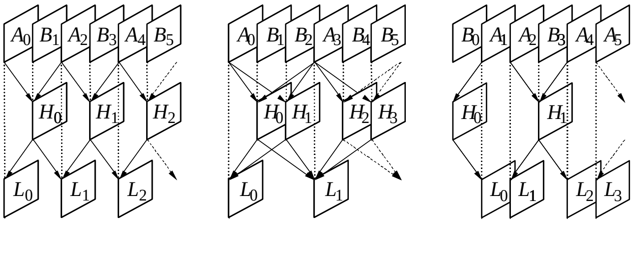
After processing all blocks $\mathbf{B}_{4\mathrm{x}4}$, the determined $\mathbf{m}_{\mathrm{U0}}$, $\mathbf{m}_{\mathrm{U1}}$, r_{U0}, and r_{U1} are formatted into H.264/AVC syntax and syntax limitations are applied if necessary. Please note that the above algorithm is simultaneously applied at coder and decoder so that each step is exactly specified. No side information is transmitted for the update step.

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} ($0 < N_{\mathrm{A}} < N_0$) input pictures and the other set containing $N_{\mathrm{B}} = N_0 - N_{\mathrm{A}}$ 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. Fig. 3 provides examples for temporal decompositions.

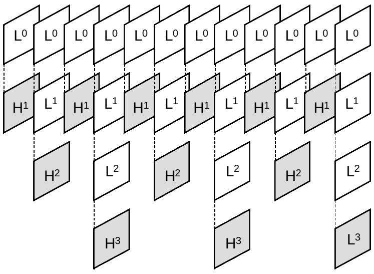


A_0 B_1 A_2 B_3 A_4 B_5
H_0 H_1 H_2
L_0 L_1 L_2
A_0 B_1 B_2 A_3 B_4 B_5
H_0 H_1 H_2 H_3
L_0 L_1
B_0 A_1 A_2 B_3 A_4 A_5
H_0 H_1
L_0 L_1 L_2 L_3

a) b) c)

Fig. 3: 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 Fig. 4, the decomposition of a group of 12 pictures is illustrated.



L⁰ L⁰ L⁰ L⁰ L⁰ L⁰ L⁰ L⁰ L⁰ L⁰ L⁰ L⁰
H¹ L¹ H¹ L¹ H¹ L¹ H¹ L¹ H¹ L¹ H¹ L¹
H² L² H² L² H² L²
H³ H³ L³

Fig. 4: Temporal decomposition of a group of 12 pictures providing 3 levels of temporal resolution with ratios of 1/2, 1/4, and 1/12.

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.

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

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:

1. The derivation process for the motion vectors in the update step must be specified.
2. 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 beyond one bit 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 [6]. 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 Lagrangain 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 compare to the approach in [6].

Note that the 2-tap filter bank as briefly discussed in Sec. 2 requires multiplication of the L_k by $\sqrt{2}$ and the H_k by $1/\sqrt{2}$ to become orthonormal. These normalization factors 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.

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

4. SCALABILITY EXTENSION OF H.264/AVC

In this work, we refer to scalability as a functionality that allows the removal of parts of the bit-stream while achieving a reasonable coding efficiency of the decoded video at reduced temporal, SNR, or spatial resolution.

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, the SNR enhancement layer is coded. For that, the quantization error between the SNR base layer and the original subband pictures is transformed and 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. Fig. 5 illustrates the idea.

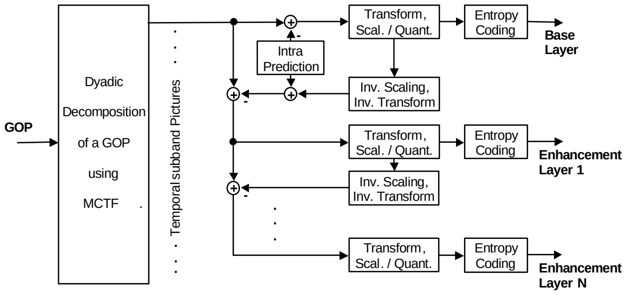


Fig. 5: General concept of the SNR scalable coding scheme.

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 [1] and the temporal decomposition as presented above.

Please note that currently no dependencies for coding are exploited between the SNR layers. Therefore, only coarse grains of scalable SNR layers can be efficiently represented such as factors of 2 in bit rate. However, this factor of 2 in bit rate also typically corresponds to the subjective difference in reconstruction quality between two videos that is noticeable by most non-expert viewers.

4.3 Spatial Scalability

While for efficient temporal and SNR scalability only small extensions to H.264/AVC are necessary, efficient spatial scalability requires a few more additions. The design goals are again straightforward extensions not affecting the H.264/AVC core algorithm.

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:

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

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 [4][5]. The prediction residual is transmitted using the H.264/AVC residual coding.

Furthermore, 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, using the H.264/AVC half-sample interpolation filter, the up-sampled reconstructed residual (high-pass) signal of the base layer is used as prediction for the residual signal of the current layer and thus only the corresponding difference signal is coded.

4.4 Impact on H.264/AVC Syntax, Decoding, and 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

1. a process for importing and up-sampling of the reconstructed lower resolution signal,
2. a process for importing and up-sampling of residual signals of the lower resolution layer,
3. 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 all 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

We have chosen a set of popular CIF and 4CIF sequences with widely varying content to illustrate the impact of the proposed extensions.

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 [6]. In Fig. 6, 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.

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 Fig. 7. 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.

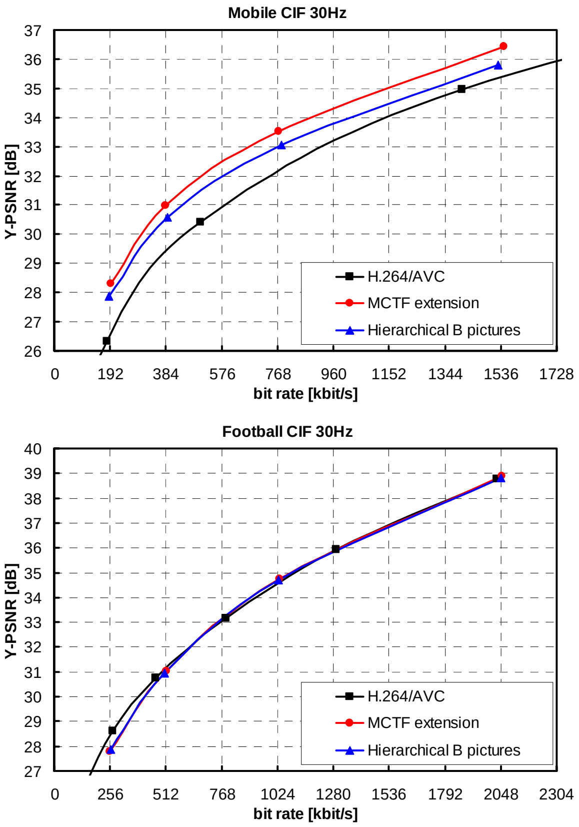


Fig. 6: Coding efficiency of the MCTF extension in comparison to a closed-loop H.264/AVC coder (“..PPPPPP..”) for the sequences “Mobile” (top) and “Football” (bottom).

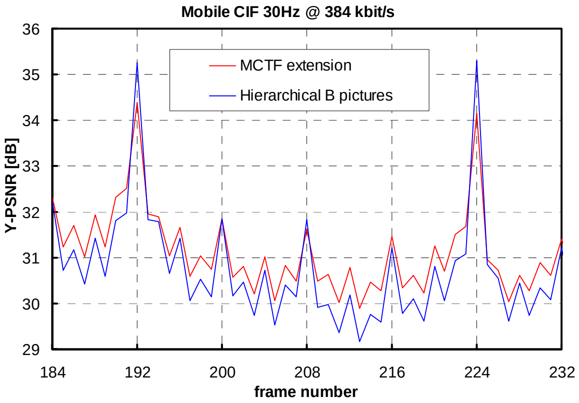


Fig. 7: PSNR fluctuations of both the MCTF extension and hierarchical B pictures for the “Mobile” sequence coded at 384 kbit/s.

5.2 Results for the Scalability Extensions

In Fig. 8, the coding efficiency of the SNR-scalable MCTF extension 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.

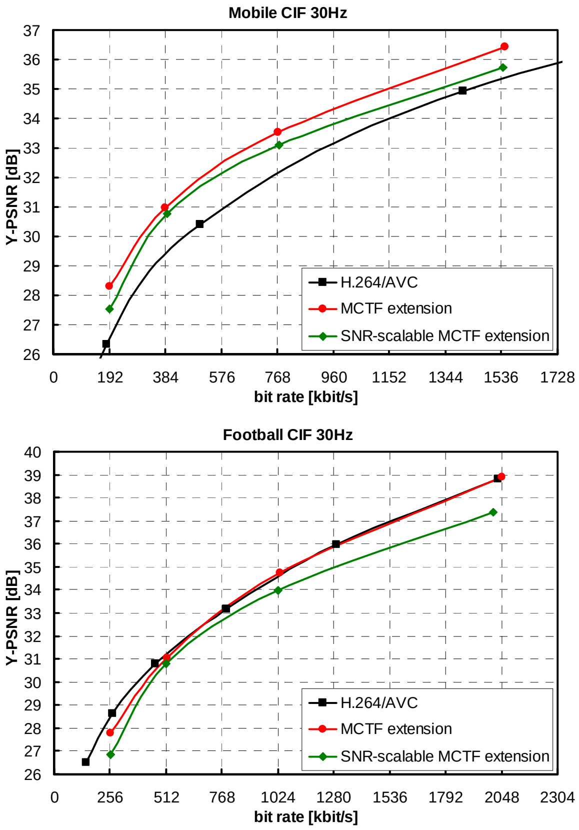


Fig. 8: Coding efficiency of the SNR-scalable MCTF extension for the sequences "Mobile" (top) and "Football" (bottom).

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.

The coding efficiency of the MCTF extension for combined spatio-temporal-SNR scalability in comparison to a single-layer H.264/AVC coder is illustrated in Fig. 9. For the "City" sequence, which is characterized by smooth global motion and high spatial detail, the coding efficiency of the scalable extension is – with the exception of one rate point – similar to or higher than that of the H.264/AVC reference. Whereas, for the complex sequence "Crew", the coding efficiency of the scalable codec is up to 1 dB worse than that of the H.264/AVC reference.

6. CONCLUSIONS

The 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.

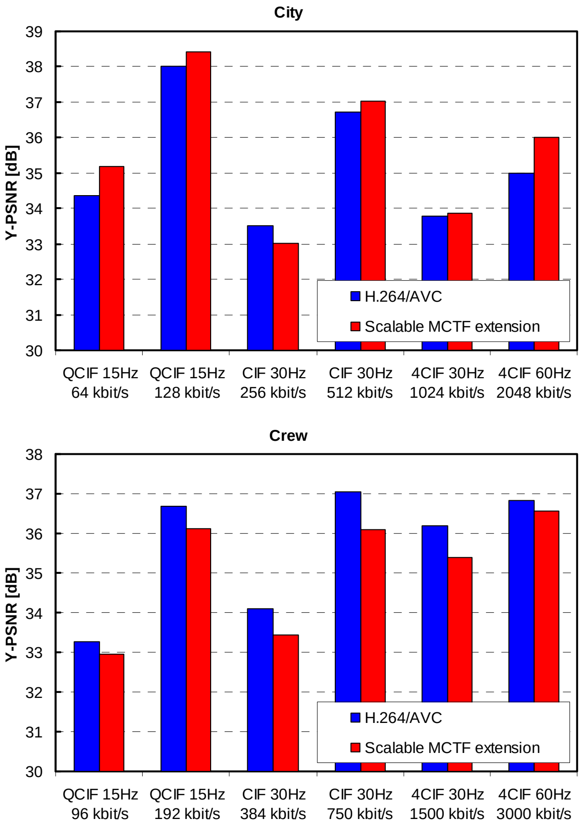


Fig. 9: Coding efficiency of the scalable MCTF extension for combined spatio-temporal-SNR scalability for the sequences "City" (top) and "Crew" bottom.

The extension for 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.

REFERENCES

[1] J.-R. Ohm, "Complexity and delay analysis of MCTF interframe wavelet structures," ISO/IEC JTC1/SC29/WG11 Doc. M8520, July 2002.

[2] M. Flierl, "Video Coding with Lifted Wavelet Transforms and Frame-Adaptive Motion Compensation," Proc. of VLBV, pp. 243-251, Sep. 2003.

[3] W. Sweldens, "A custom-design construction of biorthogonal wavelets," J. Appl. Comp. Harm. Anal., vol. 3 (no. 2), pp. 186-200, 1996.

[4] 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.

[5] 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.

[6] 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, July 2003.