Available online at www.sciencedirect.com
SCIENCE DIRECT®

Signal Processing: *Image Communication* 19 (2004) 653–673

SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

In-band motion compensated temporal filtering

Yiannis Andreopoulos[a,*], Adrian Munteanu[a], Joeri Barbarien[a],
Mihaela Van der Schaar[b], Jan Cornelis[a], Peter Schelkens[a]

[a] *Department of Electronics and Information Processing (ETRO), Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium*
[b] *Department of Electrical and Computer Engineering, University of California Davis, One Shields Avenue, Davis, CA 95616-5294, USA*

Abstract

A novel framework for fully scalable video coding that performs open-loop motion-compensated temporal filtering (MCTF) in the wavelet domain (in-band) is presented in this paper. Unlike the conventional spatial-domain MCTF (SDMCTF) schemes, which apply MCTF on the original image data and then encode the residuals using the critically sampled discrete wavelet transform (DWT), the proposed framework applies the in-band MCTF (IBMCTF) after the DWT is performed in the spatial dimensions. To overcome the inefficiency of MCTF in the critically-sampled DWT, a complete-to-overcomplete DWT (CODWT) is performed. Recent theoretical findings on the CODWT are reviewed from the application perspective of fully-scalable IBMCTF, and constraints on the transform calculation that allow for fast and seamless resolution-scalable coding are established. Furthermore, inspired by recent work on advanced prediction techniques, an algorithm for optimized multihypothesis temporal filtering is proposed in this paper. The application of the proposed algorithm in MCTF-based video coding is demonstrated, and similar improvements as for the multihypothesis prediction algorithms employed in closed-loop video coding are experimentally observed. Experimental instantiations of the proposed IBMCTF and SDMCTF coders with multihypothesis prediction produce single embedded bitstreams, from which subsets are extracted to be compared against the current state-of-the-art in video coding.

Keywords: Motion-compensated temporal filtering; Wavelet-domain motion estimation and compensation; Multihypothesis prediction; Overcomplete discrete wavelet transforms

1. Introduction

Apart from improved source-coding efficiency, recent efforts on coding standards for still images and video have focused on a number of different issues. These mainly involve multilayer content description and efficient manipulation of the compressed information during decoding. For example, the JPEG-2000 image coding standard [15] enables these features at the source-coding level based on a multilayer decomposition offered by the *discrete wavelet transform* (DWT), and a very flexible coding engine, the EBCOT algorithm [32]. In fact, a number of different modes are

*Corresponding author. Tel.: +32-2-6293951; fax: +32-2-6292883.
E-mail address: yandreop@etro.vub.ac.be (Y. Andreopoulos).

0923-5965/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2004.05.007

supported for decoding the compressed images following successive refinement in resolution and decoded-image quality. In general, the successive refinement functionality in image coding has been termed *scalability*. In the case of video, this additionally involves the capability to progressively increase the decoding frame-rate, i.e. achieve spatio-temporal refinement of each *group of pictures* (GOP).

A number of techniques have been proposed that achieve fully-scalable (i.e. quality/resolution/temporal—scalable) video coding. These are mainly based on the application of the multilevel DWT both in the spatial and temporal direction. Starting with early work on three-dimensional subband coding [17,20], the compression of a video GOP can be perceived as three-dimensional compression of a volumetric image, hence, techniques for successive refinement of this information can be borrowed from conventional image coding, using, for example, three-dimensional zero-tree [19] or cube-splitting algorithms [29].

Nevertheless, it was soon identified that for natural video sequences, *motion compensation* (MC) is an essential step for the efficient decorrelation of the video information along the temporal axis [10]. As a result, a number of pioneering works effectively incorporated MC steps in the temporal transform [24,33], leading to a class of algorithms that perform *motion-compensated temporal filtering* (MCTF). MCTF that uses short-kernel filter-banks like the Haar or the 5/3 kernel achieves efficient coding performance and allows for the complete decoupling of the transform and coding steps, since the MCTF is performed in the temporal domain *prior to* the spatial DWT, quantization and coding [5,6,24,33]. As a result, full-scalability can be achieved by using embedded coding algorithms for the compression of the residual information in the spatio-temporally decomposed GOP. Furthermore, recent work has focused on permitting arbitrary sub-pixel accuracy in MCTF while allowing for perfect reconstruction [26,30]. This has been achieved by using the lifting framework for the performance of predict and update steps [7] in the temporal decomposition. Additionally, a generic framework of *unconstrained motion compensated temporal filtering* (UMCTF) has been proposed [34], which effectively supports the selective application of the update step in the temporal decomposition, leading, in the extreme case, to a purely predictive framework for MCTF. The absence of temporal update may be a desirable feature for MCTF [5,34], since this case provides *guaranteed* artifact-free low frame-rate video even with simple motion-estimation models, such as the block-matching techniques.

In this paper, we propose a novel framework for scalable video coding that applies the MCTF *after* the spatial DWT decomposition. The proposed framework is detailed in Section 2. In order to overcome the shift-variance problem of the critically sampled DWT and to allow for an efficient application of the MCTF in the wavelet domain (in-band), we construct a shift-invariant wavelet representation by using a *complete-to-overcomplete discrete wavelet transform* (CODWT). Recent theoretical findings on the CODWT are reviewed from the application perspective of fully scalable IBMCTF in Section 3. In a strive to improve the prediction efficiency in MCTF video coding, we focus in Section 4 on the prediction part of MCTF and explore the use of multihypothesis prediction optimized in a rate-distortion sense. The efficiency of the proposed prediction scheme is studied in Section 5 using experimental instantiations for both the SDMCTF and IBMCTF video coding architectures. In addition, a performance comparison against state-of-the-art scalable and non-scalable algorithms is carried out. Our conclusions are presented in Section 6.

2. From conventional hybrid video coding to in-band motion-compensated temporal filtering

We begin by reviewing the conventional hybrid video coding structure as well as the new open-loop video coding schemes that perform a temporal decomposition using temporal filtering. Then, the proposed framework is introduced as a modification of temporal filtering that allows the independent operation across different video resolutions by operating in the transform domain.

2.1. Hybrid video coding structure and motion compensated temporal filtering

All the currently standardized video coding schemes are based on a structure in which the 2-D spatial transform and quantization is applied to the error frame coming from closed-loop temporal prediction. A simple structure describing such architectures is shown in Fig. 1(a). The operation of temporal prediction $\mathcal{P}$ typically involves block-based motion estimation and compensation (ME/MC). The decoder receives the motion vector information and the compressed error-frame C_t and performs the identical loop using this information for MC within the $\mathcal{P}$ operator. Hence, in the decoding process (seen in the dashed area in Fig. 1(a)), the reconstructed frame at time instant t can be written as

$$\widetilde{A}_t = \mathcal{P}\widetilde{A}_{t-1} + \mathcal{T}_S^{-1}\mathcal{Q}_S^{-1}C_t, \quad \widetilde{A}_0 = \mathcal{T}_S^{-1}\mathcal{Q}_S^{-1}C_0. \quad (1)$$

The recursive operation seen in (1) creates the well-known drift effect between the encoder and decoder if different information is used between the two sides, i.e. if $C_t \neq \mathcal{Q}_S\mathcal{T}_SH_t$ at any time instant t in the decoder. This is not uncommon in practical systems, since transmission errors or loss of compressed data due to limited channel capacity can be a dominant scenario in wireless or IP-based networks, where a number of clients compete for the available network resources. In general, the capability to seamlessly adapt the compression bit-rate without transcoding, i.e. SNR (quality) scalability, is a very useful feature for such network environments. Solutions for fine grain scalable (FGS) video coding based on the coding structure of Fig. 1(a) basically try to remove the prediction drift by artificially reducing at the encoder side the bit-rate of the compressed information C_t to a base layer for which the network can guarantee the correct transmission [27]. This however reduces the prediction efficiency [27], thereby leading to degraded coding efficiency for SNR scalability. To overcome this drawback, techniques that include a certain amount of enhancement layer information into the prediction loop have been proposed. For example, leaky prediction [12] gracefully decays the enhancement information introduced in the prediction loop in order to limit the error propagation and accumulation. Scalable coding schemes employing this technique achieve notable coding gains over the standard MPEG-4 FGS [21] and a good trade-off between low drift errors and high coding efficiency [12,14]. Progressive fine granularity scalable (PFGS) coding [38] yields also significant improvements over MPEG-4 FGS by introducing two prediction loops with different quality references. A generic PFGS coding framework employing multiple prediction loops with different quality references and careful drift control lead to considerable coding gains over MPEG-4 FGS, as reported in [13,37].

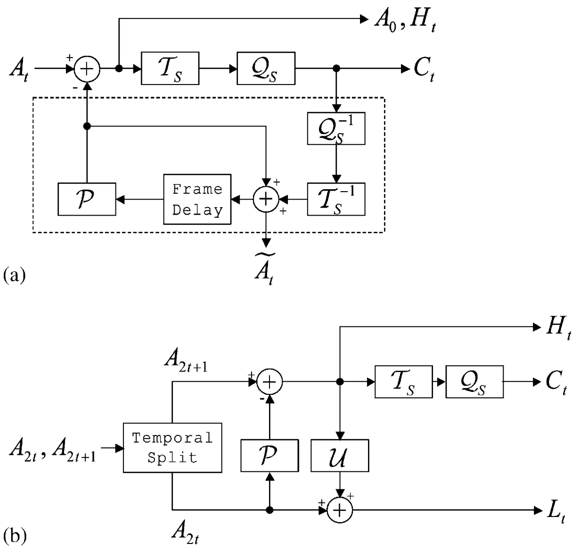


Fig. 1. (a) The hybrid video compression scheme. (b) Motion-compensated temporal filtering. Notations: A_τ consists the input video frame at time instant $\tau = 0, t, 2t, 2t+1$; $\widetilde{A}_t$ is the reconstructed frame; H_t consists the error frame and L_t is the updated frame; C_t denotes the transformed and quantized error frame using the spatial operators $\mathcal{T}_S$ and $\mathcal{Q}_S$, respectively; $\mathcal{P}$ denotes temporal prediction, while $\mathcal{U}$ denotes the temporal update.

Alternative proposals for efficient scalable video coding focus on open-loop systems, depicted in Fig. 1(b), which incorporate recursive temporal filtering. This can be perceived as a temporal wavelet transform with motion compensation [24], i.e. MCTF. Similar to the polyphase separation of the conventional lifting-based transform [7], this scheme begins with a separation of the input into even and odd temporal frames (temporal split).

Then the temporal predictor performs ME and MC to match the information of frame A_{2t+1} with the information present in frame A_{2t}. Subsequently, the update step inverts the information of the prediction error back to frame A_{2t}, thereby producing, for each pair of input frames, an error frame H_t and an updated frame L_t. The update operator performs either MC using the inverse vector set produced by the predictor [26], or generates a new vector set by backward ME [30]. The process iterates on the L_t frames, which are now at half temporal-sampling rate (following the multilevel operation of the conventional lifting), thereby forming a hierarchy of *temporal levels* for the input video. The decoder performs the mirror operation: the scheme of Fig. 1(b) operates from right to left, the signs of the $\mathcal{P}$, $\mathcal{U}$, operators are inverted and a temporal merging occurs at the end to join the reconstructed frames. As a result, having performed the reconstruction of the L_t, denoted by $\widetilde{L}_t$, at the decoder we have

$$\begin{aligned} \tilde{A}_{2t} &= \tilde{L}_t - \mathcal{U}\mathcal{T}_S^{-1}\mathcal{Q}_S^{-1}C_t, \\ A_{2t+1} &= \mathcal{P}\widetilde{A}_{2t} + \mathcal{T}_S^{-1}\mathcal{Q}_S^{-1}C_t, \end{aligned} \tag{2}$$

where $\widetilde{A}_{2t}$, $\widetilde{A}_{2t+1}$ denote the reconstructed frames at time instants $2t$, $2t+1$. As seen from (2), even if $C_t \neq \mathcal{Q}_S\mathcal{T}_S H_t$ in the decoder, the error affects locally the reconstructed frames $\widetilde{A}_{2t}$, $\widetilde{A}_{2t+1}$ and does not propagate linearly in time over the reconstructed video. Error-propagation may occur only across the temporal levels through the reconstructed $\widetilde{L}_t$ frames. However, after the generation of the temporal decomposition, embedded coding may be applied in each GOP by prioritizing the information of the higher temporal levels based on a dyadic-scaling framework, i.e. following the same principle of prioritization of information used in wavelet-based SNR-scalable image coding [32]. Hence, the effect of error propagation in the temporal pyramid is limited and seamless video-quality adaptation occurs during the process of bit-rate adaptation for SNR scalability [5,6]. In fact, the experimental results obtained with the proposed fully scalable MCTF video coder will demonstrate that this coding architecture can be comparable in rate-distortion sense to an optimized non-scalable coder that uses the closed-loop structure.

2.2. Extensions and capabilities of the MCTF structure

Similar to the extensions that have been proposed for the hybrid video coding structure of Fig. 1(a) that allow for improved functionality and higher coding efficiency, relevant work was performed recently on MCTF-based video coding. For instance, newly-proposed MCTF structures [34] allow for adaptive temporal splitting operators that can process the input in sets of frames that are larger than two in order to allow for non-dyadic temporal decompositions. Similar to the conventional lifting [7], more complex series of predict-and-update steps may be envisaged, thereby leading to longer temporal filters for MCTF [39]; on the other hand, temporal filtering may be performed even without an update operator [34]. This may be necessary in order to reduce visual artifacts that occur in the L-frames due to the poor prediction performance of the commonly employed block-based ME methods. To this end, several proposals attempt to improve the prediction performance in MCTF, based on bidirectional ME [5], or variable block sizes [6], i.e. by incorporating in the MCTF some of the advanced prediction tools proposed for the hybrid video coders. In this context, we investigate the use of multihypothesis prediction optimized in rate-distortion sense; this topic is elaborated in Section 4.

2.3. Proposed in-band motion compensated temporal filtering

In this section, we present a modification of the conventional MCTF video coding architecture that allows for temporal filtering to be performed across different resolutions of the video content. This may be a desirable functionality for MCTF since, in this way, all the advanced features discussed previously may be applicable with different configurations for each resolution of the input video. For example, different predict and update operators may be applied for each resolution, thereby allowing for additional levels of optimization or complexity reduction. In addition, since the multiresolution MCTF permits the complete decoupling of the various decodable resolutions, the use of different temporal decompositions and a variable number of temporal levels for each resolution becomes possible. This creates an additional degree of freedom for compact scalable video representations across spatial resolution.

In general, a multiresolution MCTF is achievable if the $\mathcal{T}_S$ operator is a multiresolution discrete wavelet transform and the process of temporal filtering occurs in-band, i.e. after the spatial analysis of the input video frames by the DWT. Such a scheme is shown in Fig. 2. In the proposed architecture, first a spatial transform $\mathcal{T}^l_S$ splits the input video into a discrete set of resolutions $l, 1 \leqslant l \leqslant k$, and subbands S, with $S = \{LL, LH, HL, HH\}$ (L: lowpass, H: highpass filter, in rows and columns). For each resolution, the process of temporal splitting separates the input frames in groups, and the prediction and update operations are performed in the wavelet domain. Since the critically sampled (complete) DWT is a shift-variant transform, the operator $\mathcal{S}^l_S$ is used in the subbands S of each resolution l of the reference frame $\mathcal{T}^l_S A_{2t}$ in order to construct the overcomplete DWT (ODWT). This complete-to-overcomplete DWT is necessary in order to obtain a shift-invariant representation, which is suitable for the efficient performance of in-band prediction. As it is explained in the following subsection, the predicted subbands of resolution l remain critically sampled. As a result, the subsequently produced error-frame subbands are critically sampled as well. The process then continues with the performance of the update step. Ideally, the update step would invert the motion information produced during the prediction to the overcomplete representation of the reference frame. However, it is imperative to retain critical sampling in the updated frames. To this end, the following approach is proposed: the motion vectors that stem from the critically sampled (zero-phase) positions in the reference frame are inverted to the same positions; additionally, the motion vectors that stem from a non-zero phase, or an interpolated (fractional-phase) position in the overcomplete representation, are inverted to the nearest zero-phase position of the inverted motion vector and a non-zero phase of the error frame is used instead (integer or fractional). This is similar to the technique used to obtain arbitrary sub-pixel accuracy in conventional spatial-domain MCTF [5]. The difference is that the additional ODWT phases required from the error frame are created by the $\mathcal{S}^l_S$ operator followed by interpolation directly in the subbands of each resolution l of the error frame. This process can be replicated in the decoder and perfect reconstruction is guaranteed. More details and a lifting-based formulation of this process are given later in this section.

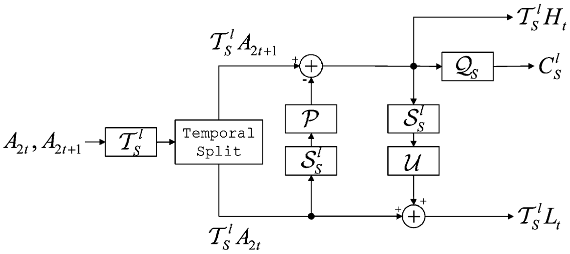


Fig. 2. In-band motion-compensated temporal filtering. A_τ represents the input video frame at time instant $\tau = 2t, 2t+1$, H_t is the error frame, while L_t consists the updated frame; C_t denotes the transformed and quantized error frame; $\mathcal{T}^l_S W$denotes the resolution level l of the DWT of frame W, $W = \{A, L, H\}$; $\mathcal{S}^l_S$ is the CODWT of resolution level l; $\mathcal{P}$ denotes temporal prediction, while $\mathcal{U}$ denotes the temporal update.

Decoding occurs following the principle of inverse MCTF, i.e. for each resolution level $l, 1 \leqslant l \leqslant k$, the structure of Fig. 2 is inverted by operating from right to left, inverting the sign of the predict and update operators, and performing a temporal merging. When the necessary number of resolutions is collected at the decoder, the inverse transform inverts the accumulated set of subbands to the spatial-domain representation. In the general case of a long temporal filter, a series of predict and update step can be used, following the principles explained before.

2.4. Practical instantiation of the proposed in-band prediction and update

The proposed IBMCTF framework employs classical biorthogonal filter-pairs (like the 9/7 filter-pair) for the spatial transform with a total of k decomposition levels. We denote the two-dimensional critically sampled subbands of the decomposition level $l, 1 \leqslant l \leqslant k$, as $S^l_{(0,0)}$ where the superscript indicates the decomposition (resolution level) and the subscript indicates the polyphase components retained after the downsampling in the rows and columns. Note that the subband $LL^l_{(0,0)}$ belongs to the critically sampled DWT only if $l = k$.

For progressive decoding in resolution, when the decoder is processing the decomposition level $l, k \geqslant l \geqslant 1$, we have $LH^q_{(0,0)} = HL^q_{(0,0)} = HH^q_{(0,0)} = \varnothing$ for all $l > q \geqslant 1$, i.e. the finer-resolution levels are set zero since they have not been received. As a result, under progressive decoding in resolution, the structure of Fig. 2 is invertible if the $\mathcal{S}^l_S$ operator of the encoder constructs the ODWT at each resolution level l accounting for the fact that, for the low-resolution decoders, the DWT subbands of the finer resolution-levels q are not considered. Under this constraint, for the CODWT of the reference frame(s) one can use either classical techniques such as the low-band shift (LBS) algorithm [25], or more advanced techniques [1] that use a single-rate calculation scheme with reduced computational and memory requirements. Since this process may well be one of the most computationally intensive tasks at the decoder side and it occurs separately for each resolution level and for each reference frame, fast and single-rate calculation schemes are imperative for an optimized, low-complexity implementation. The reader is referred to Section 3 for details on this topic.

After the construction of the ODWT, a set of critically sampled subbands $S^l_{(i,j)}$ is produced for each level l, where (i,j) denotes the phase in the ODWT domain [22,25] with $0 \leqslant i < 2^l$, $0 \leqslant j < 2^l$. An ODWT example is shown in Fig. 3. Interpolation to sub-pixel accuracy that is typically used in spatial-domain ME/MC can be performed directly in the ODWT of the reference frame(s), if their ODWT-domain phase components are interleaved to create the undecimated DWT (UDWT) [30]. This essentially stems from the fact that linear interpolation and DWT filtering without downsampling are both linear shift-invariant operators and their order of application to the input signal can be interchanged [22].

Fig. 2 shows that the prediction and update steps are performed in a level-by-level fashion. For each level, full search can be performed in order to jointly minimize the distortion measure for each triplet of blocks from the $LH^l_{(0,0)}, HL^l_{(0,0)}, HH^l_{(0,0)}$ subbands of the current frame that correspond to the same spatial-domain location with a triplet of blocks from each of the $LH^l_{(i,j)}, HL^l_{(i,j)}, HH^l_{(i,j)}$, of the reference(s). For the LL subband (coarsest resolution level), one can potentially perform the ME process separately, or jointly with the triplet of high-frequency subbands. In total, the prediction step can be expressed as

$$\mathcal{T}^l_S H_t[m,n] = \mathcal{T}^l_S A_{2t+1}[m,n] - \mathcal{I}_{(i_m,i_n)} \mathcal{S}^l_{(p_m,p_n)} \mathcal{T}^l_S A_{2t}[m - d_m, n - d_n], \tag{3}$$

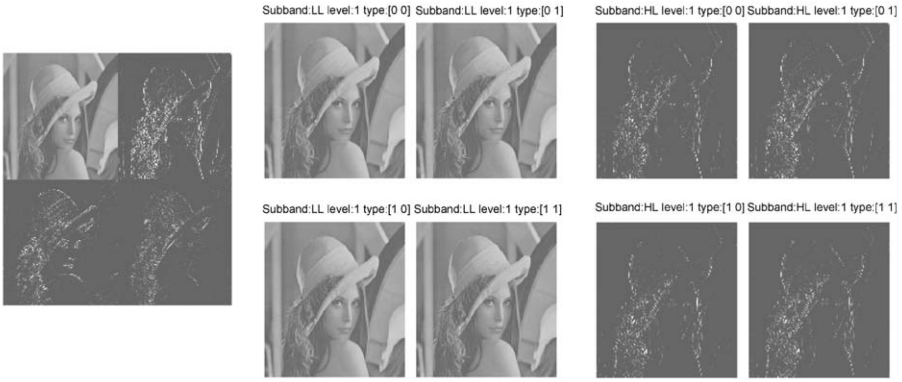

Subband:LL level:1 type:[0 0]
Subband:LL level:1 type:[0 1]
Subband:HL level:1 type:[0 0]
Subband:HL level:1 type:[0 1]
Subband:LL level:1 type:[1 0]
Subband:LL level:1 type:[1 1]
Subband:HL level:1 type:[1 0]
Subband:HL level:1 type:[1 1]

Fig. 3. A one-level DWT and the ODWT-domain phase components (subbands LL and HL are shown as examples). The four phase-components of each subband can be interleaved to create the UDWT subband representation.

where $\mathcal{T}^l_S A[m,n]$ is the wavelet coefficient at position (m,n) in subband S (of level l) of the critically sampled DWT of frame A; $\mathcal{S}^l_{(i,j)}\mathcal{T}^l_S A[m,n]$ is the ODWT coefficient with phase (i,j) that corresponds to $\mathcal{T}^l_S A[m,n]$; and $\mathcal{I}_{(k,l)}\mathcal{S}^l_{(i,j)}\mathcal{T}^l_S A[m,n]$ is the interpolated coefficient at fractional distance (k,l) from coefficient $\mathcal{S}^l_{(i,j)}\mathcal{T}^l_S A[m,n]$ in the interpolated UDWT, with $k,l = \{0, 1/R, .., (R-1)/R\}$ and R representing the maximum interpolation precision.

Eq. (3) shows that each motion vector consists of three different components: the phase component (p_m, p_n), the in-band translation (d_m, d_n) and the fractional-phase component (i_m, i_n). In addition, although the ODWT is used for the reference frame(s) during the prediction process, the produced H-frames remain *critically sampled*. The in-band prediction approach of (3) corresponds to a prediction with a single motion vector from a single reference frame per subband and resolution level; however, similar to the spatial-domain methods proposed in the literature, multiple references, and multihypothesis prediction can be used per resolution level or wavelet subband. In general, it was recently proposed [36] that different motion estimation approaches producing independent motion vectors per subband or per resolution level [18] can be viewed as a form of spatial multihypothesis prediction, which complements the temporal multihypothesis capabilities.

Concerning the in-band application of an update step, Eq. (3) indicates that two aspects of inverting the motion vectors have to be treated: inversion of an in-band motion vector (d_m, d_n) with non-zero phase component (p_m, p_n), and, additionally, inversion of motion vectors pointing to the interpolated wavelet coefficients at fractional position (i_m, i_n) in the interpolated UDWT of the reference frames. For both cases, a strategy similar to the technique used to obtain arbitrary sub-pixel accuracy in conventional spatial-domain MCTF [5] is proposed. Specifically, it is first defined that

$$i^{\mathrm{res}}_m = \begin{cases} 0, & \forall i_m = 0, \\ 1 - i_m, & \forall i_m \neq 0, \end{cases}$$

$$p'_m = \begin{cases} p_m, & \forall i_m = 0, \\ p_m + 1, & \forall i_m \neq 0, \end{cases}$$

$$p^{\mathrm{res}}_m = \begin{cases} 0, & \forall p'_m = 0, \\ |2^l - p'_m|, & \forall p'_m \neq 0, \end{cases}$$

$$d^{\mathrm{res}}_m = \begin{cases} 0, & \forall p'_m = 0, \\ 1, & \forall p'_m \neq 0. \end{cases} \tag{4}$$

In addition, we define i^{res}_n, p^{res}_n and d^{res}_n in the same successive manner. Then, the update step that corresponds to the prediction step of (3) can be performed as follows:

$$\begin{aligned} \mathcal{T}^l_S L[m - d_m, n - d_n] = {} & \mathcal{I}_{(i^{\mathrm{res}}_m,\, i^{\mathrm{res}}_n)}\mathcal{S}^l_{(p^{\mathrm{res}}_m,\, p^{\mathrm{res}}_n)} \\ & \times \mathcal{T}^l_S H_t\left[m + d^{\mathrm{res}}_m,\, n + d^{\mathrm{res}}_n\right] \\ & + \mathcal{T}^l_S A_{2t}[m - d_m,\, n - d_n]. \end{aligned} \tag{5}$$

The last equation demonstrates that, similar to spatial domain MCTF, the successive definitions of (4) perform phase inversion of the in-band motion vector of the interpolated ODWT of the error frame: first the fractional (interpolated) phase component $(i_m,\ i_n)$ is inverted to $\left(i^{\mathrm{res}}_m, i^{\mathrm{res}}_n\right)$ and then the integer (ODWT) phase $(p_m, p_n)$is inverted to $\left(p'_m,\ p'_n\right)$. Finally, the in-band position in the critically sampled wavelet decomposition of the error frame is modified by $\left(d^{\mathrm{res}}_m,\ d^{\mathrm{res}}_n\right)$.

Generalizing the previous example, k predict and update procedures can be performed for k decomposition levels of the DWT. The motion vectors produced for the luminance channel are subsequently subsampled and used for the chrominance channels as well. The analysis of the motion-vector overhead resulting from the use of separate block-based prediction loops per resolution level [3] reveals that, for $k > 1$, the uncoded rate necessary for the motion vectors does not exceed 1.5 times the motion-vector coding rate of the equivalent spatial-domain ME that derives one vector per block. In addition, the prediction accuracy in the majority of cases is increased [18]. We conclude that the trade-off between the number of wavelet-domain vectors and the prediction accuracy can be optimally investigated within a rate-distortion framework, similar to what is done in the rate-constrained temporal multihypothesis prediction [9]. This consists one of our current research topics in this area.

3. Complete to overcomplete discrete wavelet transform

We elaborate on two methods for the CODWT used under the IBMCTF-coding scenario, namely the low-band shift (LBS) method [25,28] that is a specific implementation for the "à-trous" algorithm [23], and the prediction-filters method [1,35].

3.1. Notations

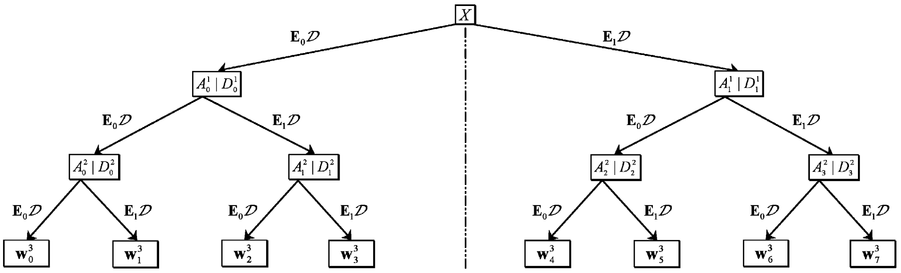
We briefly define the notations used in this section. Bold-faced capital and lower letters indicate matrices and vectors, respectively, while **I** denotes the identity matrix. All the used indices are integers and the superscripts denote the decomposition level, except for superscript T that denotes transposition. The polyphase separation (lazy wavelet) of a given signal or filter $X(z)$ is denoted as $\mathcal{D}X(z) = [X_0(z)\, X_1(z)]^{\mathrm{T}}$, with the inverse operation given by: $\mathcal{D}^{-1}\mathcal{D}X(z) = X_0(z^2) + zX_1(z^2) = X(z)$. The analysis polyphase matrices that produce the even or odd polyphase components of the non-decimated transform (0 = even, 1 = odd) are denoted as $\mathbf{E}_0(z)$, $\mathbf{E}_1(z)$, respectively, and their definition is

$$\mathbf{E}_0(z) = \begin{bmatrix} H_0(z) & H_1(z) \\ G_0(z) & G_1(z) \end{bmatrix}, \quad \mathbf{E}_0(z) = \mathbf{E}_1(z)\begin{bmatrix} 0 & 1 \\ z^{-1} & 0 \end{bmatrix},$$

where H, G are the low- and high-pass analysis filters, respectively. The corresponding synthesis polyphase matrices are denoted as $\mathbf{R}_i(z) = [\mathbf{E}_i(z)]^{-1}, i = \{0, 1\}$. For all the signals and filters in this paper, we use the typical Type-I and Type-II polyphase definitions [7], and in order to simplify the expressions we always assume that the filters H and G are properly shifted so that perfect reconstruction is achieved with zero delay [7], i.e. $\det \mathbf{E}_0(z) = -1$. For a one-dimensional signal $X(z)$, the p-phase wavelet decomposition of level l of the ODWT [23] is denoted as $\mathbf{w}_p^l(z) = [A_p^l(z)\; D_p^l(z)]^{\mathrm{T}}$, $0 \leqslant p < 2^l$, with $A_p^l(z)$, $D_p^l(z)$ the p-phase low- and high-frequency subbands of level l, respectively

3.2. Resolution-scalable CODWT with the level-by-level low band shift

Fig. 4 shows an example of the one-dimensional ODWT for three decomposition levels starting from an input signal X. This figure facilitates the description of the LBS method. Initially, the input signal X is decomposed in two subband sets A_0^1, D_0^1 and A_1^1, D_1^1 by retaining separately the even and odd polyphase components of the non-decimated decomposition, respectively. Equivalently, two critically sampled wavelet decompositions can be performed: one to the zero-shifted and one to the unit-shifted input signal, respectively [25,31]. Each of the low-frequency subbands A_0^1 and A_1^1 is further analyzed in the same manner, while the high-frequency subbands D_0^1 and D_1^1 represent the output of the first decomposition level. The process is repeated recursively as shown in Fig. 4, yielding the fast approach for the ODWT calculation from the input signal X. The subbands A_0^3 andD_0^l, $l = 1, 2, 3$ consist the critically sampled DWT of three levels, while the subbands A_i^3, D_i^l, $1 \leqslant l \leqslant 3$, $0 \leqslant i < 2^l$ represent the calculated ODWT for three decomposition levels.



X
$\mathbf{E}_0\mathcal{D}$
$\mathbf{E}_1\mathcal{D}$
$A_0^1 \mid D_0^1$
$A_1^1 \mid D_1^1$
$\mathbf{E}_0\mathcal{D}$
$\mathbf{E}_1\mathcal{D}$
$\mathbf{E}_0\mathcal{D}$
$\mathbf{E}_1\mathcal{D}$
$A_0^2 \mid D_0^2$
$A_1^2 \mid D_1^2$
$A_2^2 \mid D_2^2$
$A_3^2 \mid D_3^2$
$\mathbf{E}_0\mathcal{D}$
$\mathbf{E}_1\mathcal{D}$
$\mathbf{E}_0\mathcal{D}$
$\mathbf{E}_1\mathcal{D}$
$\mathbf{E}_0\mathcal{D}$
$\mathbf{E}_1\mathcal{D}$
$\mathbf{E}_0\mathcal{D}$
$\mathbf{E}_1\mathcal{D}$
$\mathbf{w}_0^3$
$\mathbf{w}_1^3$
$\mathbf{w}_2^3$
$\mathbf{w}_3^3$
$\mathbf{w}_4^3$
$\mathbf{w}_5^3$
$\mathbf{w}_6^3$
$\mathbf{w}_7^3$

Fig. 4. The construction of the ODWT of three levels starting from the input signal X. A number of one-level DWTs are performed that retain the even or odd samples of the non-decimated transform ($\mathbf{E}_i\mathcal{D}$ with $i = 0, 1$, respectively).

Notice that the ODWT subbands A_i^3, D_i^l shown in Fig. 4 stem from the classical ODWT decomposition scheme [28], which is equivalent to the "à trous" algorithm [23,28]. The difference is that, for every level, the subbands of Fig. 4(a) must be interleaved in order to produce the UDWT obtained with the algorithm "à trous". As a result, any subband D_i^l in the ODWT of Fig. 4 is the ith polyphase component of the UDWT of level l. In the two-dimensional case, the two-dimensional ODWT can be constructed in the same manner as in Fig. 4, by applying the LBS method on the input-subband rows and on the columns of the result. Hence, in the remaining part of this section we focus on the one-dimensional case in order to facilitate the description, with the extension in two dimensions following the row–column approach.

In the resolution-scalable coding framework of Fig. 2, the key-difference is that the coarsest-resolution subbands of the critically sampled decomposition (subbands $\mathbf{w}_0^3$) may be processed without the subbands of the higher-resolution levels $l = 2, 1$; this means that, when decoding the quarter resolution input sequence, the subbands D_0^l will not be present at the decoder. Under such a resolution-progressive decoding environment, the LBS method is readily adaptable to perform a *level-by-level* construction of the ODWT representation (denoted by LL-LBS), starting from the subbands of the critically sampled transform of each decoded level. Specifically, starting from subbands $\mathbf{w}_0^3$ (coarsest resolution level) of Fig. 4, three inverse wavelet transforms are performed. Subsequently, from the reconstructed signal, all the subbands $\mathbf{w}_i^3$, $1 \leqslant i < 8$ are constructed by performing the forward transforms shown in Fig. 4. Since in this case the subbands D_0^2, D_0^1 are not available, and due to the fact that lossy decoding may have occurred for the subbands $\mathbf{w}_0^3$, the reconstructed signal and the subbands $\mathbf{w}_i^3$ are an approximation of X and of the original ODWT of level 3, respectively, shown in Fig. 4. However, given the information available at the decoder side, this ODWT representation is the best possible approximation for the current resolution.

The construction of the ODWT by the LL-LBS is performed in the same manner for the finer resolution levels (2 and 1).

3.3. Resolution-scalable CODWT with the prediction-filters

In this section, we present an alternative approach to the LL-LBS method for the level-by-level CODWT of levels 3, 2 and 1. In this approach, the CODWT uses a set of prediction filters [1,35], denoted as F_j^i, $1 \leqslant i \leqslant 3$, $0 \leqslant i < 2^{l+1}$, which are convolved with the subbands $\mathbf{w}_0^i$ to calculate the ODWT representation of each level. We demonstrate that, by using the prediction-filters, the overcomplete representation is "predicted" in a level-by-level manner. As a result, no upsampling or downsampling is performed with this algorithm and no reconstruction of the spatial-domain signal X is performed.

The form of these filters has been derived [1,35] as a separate set of propositions for each decomposition level E. The proposition $\mathcal{Q}(1)$ corresponding to $E = 1$ is

$$\mathcal{Q}(1) : \mathbf{w}_1^1(z) = \mathbf{F}_0^1(z)\mathbf{w}_0^1(z) \tag{6}$$

with $\mathbf{F}_0^1(z)$ defining the prediction-filters matrix of the proposition $\mathcal{Q}(1)$, which is defined as

$$\mathbf{F}_0^1(z) = \begin{bmatrix} F_0^1(z) & F_1^1(z) \\ F_2^1(z) & F_3^1(z) \end{bmatrix} = \begin{bmatrix} zH_0(z)G_0(z) - H_1(z)G_1(z) & H_1(z)H_1(z) - zH_0(z)H_0(z) \\ zG_0(z)G_0(z) - G_1(z)G_1(z) & H_1(z)G_1(z) - zH_0(z)G_0(z) \end{bmatrix}. \tag{7}$$

The proof of (6) can be derived by performing an inverse transform to subbands $\mathbf{w}_0^1(z)$ followed by a forward wavelet transform that retains the odd polyphase components of the non-decimated decomposition [1,35], as shown in Fig. 4. In summary

$$\mathbf{w}_1^1(z) = \mathbf{E}_1(z)\mathcal{D}\mathcal{D}^{-1}\mathbf{R}_0(z)\mathbf{w}_0^1(z) = \mathbf{F}_0^1(z)\mathbf{w}_0^1(z).$$

Based on the proof of [1], the construction of the level-by-level CODWT is generalized to the following proposition:

$$\mathcal{Q}(k) : \mathbf{w}_x^k(z) = \mathbf{F}_p^{l+1}(z)\mathbf{w}_0^k(z). \tag{8}$$

In (8), $1 \leqslant x < 2^k$ denotes the ODWT subband index at level k (phase x), and it is written as $x = 2^l + p$ where l is given by $l = \lfloor \log_2 x \rfloor$ ($\lfloor a \rfloor$ denotes the integer part of a) and p defined as $p = \sum_{j=0}^{l-1} b_j 2^j$, $b_j = \{0, 1\}$. In the particular case of $l = 0$ corresponding to $k = 1$ and $x = 1$, we set $p = 0$ to ensure that Eq. (8) is identical with $\mathcal{Q}(1)$ given by (6). Proposition $\mathcal{Q}(k)$ consists in fact the modification of the result given in [1] for the case where $D_0^l = 0$ for every l, $1 \leqslant l < k$.

The prediction filters needed to calculate the ODWT subbands of level k, $k > 1$, are the filters of the $\mathbf{F}_0^1(z)$ matrix given by (7), and the matrices $\mathbf{F}_p^{l+1}(z)$, $1 \leqslant l < k$, given by [1]:

$$\begin{aligned}\mathbf{F}_p^{l+1}(z) &= \begin{bmatrix} F_{4p}^{l+1}(z) & F_{4p+1}^{l+1}(z) \\ F_{4p+2}^{l+1}(z) & F_{4p+3}^{l+1}(z) \end{bmatrix} \\ &= F^l_{4\lfloor p/2 \rfloor, b_0}(z)\mathbf{I} + z^{-(1-b_0)} F^l_{4\lfloor p/2 \rfloor, 1-b_0}(z)\mathbf{F}_0^1(z).\end{aligned} \tag{9}$$

3.4. Generalization of the derived formulation and efficient implementation

For each decomposition level k of the DWT, the proposition $\mathcal{Q}(k)$ consists of the convolution of the critically-sampled subbands of level k with the prediction filters of the matrix $\mathbf{F}_p^{l+1}(z)$, $0 \leqslant l < k$. In general, for the case of the level-by-level calculation, the CODWT of each level k can be written as

$$\mathbf{w}^k_{\mathrm{ODWT}}(z) = \begin{bmatrix} \mathbf{P}^1(z) \\ \mathbf{P}^2(z) \\ \vdots \\ \mathbf{P}^k(z) \end{bmatrix} \mathbf{w}_0^k(z), \tag{10}$$

where

$$\mathbf{w}^k_{\mathrm{ODWT}}(z) = \begin{bmatrix} \mathbf{w}_1^k(z) \\ \vdots \\ \mathbf{w}^k_{2^k-1}(z) \end{bmatrix}$$

and

$$\mathbf{P}^{l+1}(z) = \begin{bmatrix} \mathbf{F}_0^{l+1}(z) \\ \mathbf{F}_1^{l+1}(z) \\ \vdots \\ \mathbf{F}^{l+1}_{2^l-1}(z) \end{bmatrix}$$

for every l, $0 \leqslant l < k$.

The prediction filters of level $l+1$ are defined recursively in Eq. (9) based on the filters $F^l_{4\lfloor p/2 \rfloor}(z)$. Based on this observation, in [2] the following symmetry property for the prediction filters of an arbitrary decomposition level l was demonstrated:

$$\begin{aligned} F^l_{4\lfloor p/2 \rfloor, 0}(z) &= zF^l_{2^{l+1}-4-4\lfloor p/2 \rfloor, 1}(z^{-1}), \\ F^l_{4\lfloor p/2 \rfloor, 1}(z) &= zF^l_{2^{l+1}-4-4\lfloor p/2 \rfloor, 0}(z^{-1}). \end{aligned} \tag{11}$$

This symmetry can be used to derive computationally efficient algorithms for the CODWT since the multiplications for half of the filters of each level can be reused for the production of the results of the other half [2].

3.4.1. Practical benefits from the codwt based on the prediction filters

In a resolution-scalable scenario, the prediction-filters (PF) approach exhibits complexity gains in comparison to the classical LBS algorithm. A comparison between the two CODWT approaches for the 9/7 filter-pair under the level-by-level construction [1] reveals that the PF approach decreases the execution time by an average of 83% and 78% as compared to the convolution and lifting-based LBS implementation, respectively. All the experiments were carried out using "C" implementations in an Intel Pentium IV for the case of a coding system for HDTV resolution sequences with a maximum of $k = 4$ decomposition (resolution) levels.

Another important advantage of the CODWT based on the prediction filters is in the required memory for the ODWT calculation for each block during the temporal filtering process. In our experiments, after the performance of the ME process in the ODWT domain during the predict step, we have utilized a memory-efficient implementation for the temporal filtering exhibited in Eqs. (3) and (5). Specifically, we exploit the knowledge of the in-band motion vector to create a coefficient-by-coefficient calculation of the temporal filtering in a two-step process: for the example of Eq. (3) and the generic case of a fractional-phase in-band motion vector, first the necessary (integer-phase) ODWT coefficients required for the convolution with the interpolation filter-kernel are calculated. This is performed using a localized direct filtering with the required prediction filters of (9). Subsequently, the fractional-phase coefficient required in (3) is created by applying the interpolation-filter kernel on the currently produced coefficients. The same strategy is followed for the update step. In this way, no memory for the storage of the interpolated UDWT is necessary during in-band temporal filtering since every coefficient participating in IBMCTF is calculated on the fly. This consists the extreme example for the minimization of the memory requirements for the IBMCTF and it becomes a realizable solution only with the use of the proposed single-rate CODWT.

4. Optimized multihypothesis temporal filtering

In this section, we focus on the temporal filtering aspects of MCTF and present a new algorithm for optimized multihypothesis prediction and update using block-based ME/MC. The prediction step of the algorithm operates following a macroblock concept, i.e. the current frame (or wavelet subbands of each resolution level for IBMCTF) is partitioned into non-overlapping blocks of $B \times B$ pixels (or $(B/2^l) \times (B/2^l)$ wavelet coefficients for each subband of resolution l). Then the algorithm performs a prediction for the macroblocks and produces a set of motion-vectors and the predicted error frame (or error subbands of each resolution). After the performance of the prediction step for a sufficient number of frames, the update step inverts the error-frame information to the reference frames using the inverse motion vector set and creates the lowpass-filtered frames to be used for the subsequent temporal levels.

A pruning algorithm for variable block-size ME prediction has already been proposed in the context of MCTF [6]. Our approach differs from [6] in the use of multihypothesis, the use of multiple reference frames (i.e. longer temporal filters) and the more exhaustive approach for the macroblock prediction-mode selection.

Multihypothesis prediction has been originally proposed in the context of closed-loop video coding in order to improve prediction efficiency [8,9]. Fig. 5(a) illustrates that the basic concept can be seen as a generalization of bi-directional prediction: each block may be predicted using a unidirectional or bidirectional prediction with one or two reference blocks. To utilize such a technique in MCTF-based coding, we couple the prediction step with the corresponding update step as shown in Fig. 5(b): the current error frame is used to update the reference frames and create a set of L-frames. The example of Fig. 5 corresponds in general to temporal filtering using the

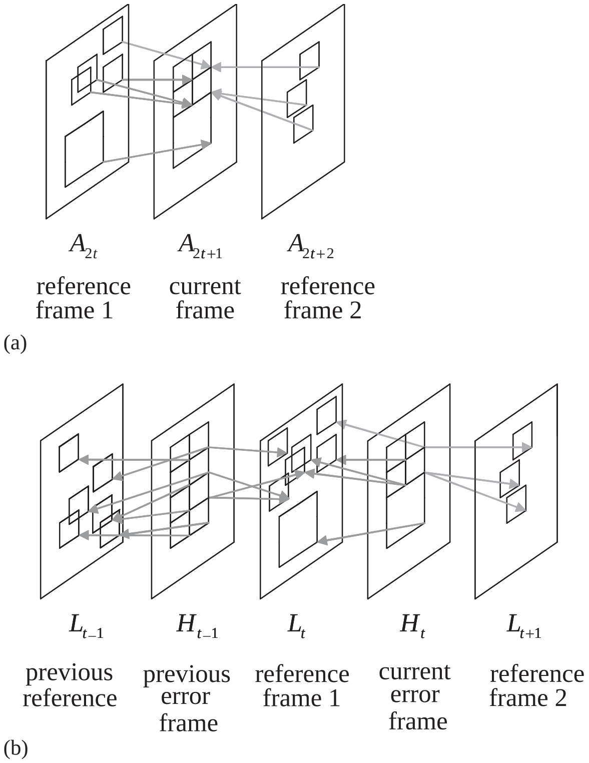


Fig. 5. (a) Examples of variable block size multihypothesis prediction of frame A_{2t+1} using frames A_{2t}, A_{2t+2} as references. The arrows indicate the prediction direction. (b) The corresponding example of the update step: the information of the produced error frame from the multihypothesis prediction is used to update the two reference frames. To complete the creation of the L_t frame, the previous update step with error-frame H_{t-1} is necessary, leading to a temporal dependency of five frames.

motion-compensated 5/3 filter-pair. However, due to the adaptive properties of the optimized multihypothesis prediction step, the temporal filtering is locally adapted to motion-compensated 1/3 filter-pair (5/3 with no update step), motion-compensated bi-directional 2/2 (Haar) filter-pair and the motion-compensated bi-directional 1/2 filter-pair (Haar with no update step). In general, the optimized multihypothesis MCTF proposed in this paper can be seen as a rate/distortion optimized adaptive motion-compensated temporal lifting decomposition with bookkeeping: the algorithm performs a best-basis selection process in the direction of motion and indicates the decisions to the decoder using the motion-vector information. Although our experiments are restricted to the biorthogonal MC filter-pairs with maximally one predict-and-update step, generalizations to smoother wavelet families that capture better the long-term temporal correlations can be envisaged.

4.1. Prediction step

Although the proposed ME is based on the algorithm of [9], its novelty lays in the joint optimization process for the multihypothesis prediction with variable block sizes and in its ability to generate a rate-constrained motion-vector data set for a *multiple* set of rates, *without* multiple application of the complex multihypothesis estimation step. The latter is possible by performing the operation of optimized prediction for each macroblock in three different phases, as shown in Fig. 6.

- *Macroblock split:* Starting from a macroblock of $B \times B$ pixels (or $(B/2^l) \times (B/2^l)$ coefficients in the case of IBMCTF), a splitting process generates a number of non-overlapping subblocks. In the presented experiments, we follow a quadtree splitting approach to P partition levels, where each level p_s contains $2^{p_s} \times 2^{p_s}$ subblocks, $0 \leqslant p_s < P$. For each level p_s, a number of subblocks have been pruned out due to the selection of their parent block during pruning. As a result, for each level, the following steps are only performed to the subblocks that have been selected during the pruning step of the previous level.
- *Multihypothesis ME:* For the current subblock, a local splitting to its four quadrants is performed. A number of hypotheses M is established for the subblock and its four quadrants, with $M = 2$ in our experiments. The case of $M = 0$ can correspond to the use of intra-prediction modes based on the causal neighborhood around the current subblock (or subblock quadrant). For the current subblock (or subblock quadrant) and each hypothesis m=1,…,M, we apply the multihypothesis estimation algorithm of [8] without a rate constraint. This means that, for $m > 1$, the algorithm initiates all motion vectors and then iterates by estimating one vector at a time so that the prediction error of the current subblock is minimized. When no vector was modified during the last iteration, the algorithm terminates. As a result, for each hypothesis m, the combination of motion vectors that minimizes the produced error-frame distortion in the certain subblock (or subblock quadrant) of the macroblock is chosen. In our experiments, we use the sum of absolute differences as the distortion measure. The motion vector data for each hypothesis and the corresponding distortion are kept in a structure for use in the following steps.
- *Pruning process*: The pruning process performed for the current subblock is given in Fig. 7. Three distinct passes occur: The first pass, **Estimation(i)**, identifies the rate and distortion of each possible combination n_i

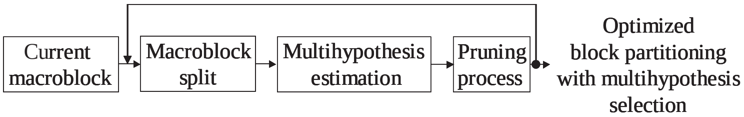

Current macroblock
Macroblock split
Multihypothesis estimation
Pruning process
Optimized block partitioning with multihypothesis selection

Fig. 6. Proposed motion estimation for the prediction step of MCTF.

```
For the current subblock at $(r_s, c_s)$ of partition level $p_s$, $0 \le p_s < P$ in the macroblock:
  If $p_s = 0$ then Set $\mathcal{N} = \varnothing, \mathcal{N}_{deleted} = \varnothing$     // Initialize to zero: no MB partition exists yet.
  else $\mathcal{N} = \mathcal{N}_{deleted}$                                // Restore partition points excluded before.
  For every $p$: $p \leftarrow [0,1], p \in \mathbb{N}$                    // For all the different combinations of hypotheses in
    For every $i$: $i \in \left[0, (M+1)^{2^{2p}} - 1\right], i \in \mathbb{N}$ // the subblock ( $p=0$ ) and its quadrants ( $p=1$ ).
      Begin_Estimation(i):
      Set $R \leftarrow 0$, Set $D \leftarrow 0$
      For every quadrant of the subblock at $(b_r, b_c)$: $b_r, b_c \in \left[0, 2^p - 1\right], b_r \in \mathbb{N}$
        Set $h(b_r, b_c) \leftarrow \left(\left\lfloor i/(M+1)^{b_r 2^p + b_c} \right\rfloor \bmod (M+1)\right)$     // mod: modulo operator
        Set $R \leftarrow R + \text{Estimate}_R(b_r, b_c, h(b_r, b_c))$
        Set $D \leftarrow D + \text{Estimate}_D(b_r, b_c, h(b_r, b_c))$     // precalculated from the ME step
      Set $R \leftarrow R + R_{rem}$, $D \leftarrow D + D_{rem}$             // rate,distortion of remaining parts of MB
      Create $n_i = \left\{R, D, \bigcup_{b_r, b_c} h(b_r, b_c), \left[(r_s, c_s), p_s\right]\right\}$
      For every $n_k, n_k \in \mathcal{N}$
        If $n_k(R) < n_i(R)$ and $n_k(D) \le n_i(D)$ then Goto End_Estimation(i)
      Add $n_i$ to $\mathcal{N}$
      End_Estimation(i)
  // prune the list of valid R-D points to make a monotonically-decreasing slope
  Begin_RD_Prune:
  Set $l = 0$, Set $n_l(R) = 0$, Set $n_l(D) = \infty$                 // Set the initial point
  For every $n_k, n_k \in \mathcal{N}$
    Set $\Delta R_k \leftarrow n_k(R) - n_l(R)$, Set $\Delta D_k \leftarrow n_l(D) - n_k(D)$, Set $S_k \leftarrow \frac{\Delta D_k}{\Delta R_k}$
    If $l \ne 0$ and $S_k > S_l$ then Move $n_l$ from $\mathcal{N}$ to $\mathcal{N}_{deleted}$, Goto Begin_RD_Prune
    else Set $l = k$
  End_RD_Prune
  Begin_Estimation_Truncation:
  Input $\lambda$                                                  // input of the control parameter
  For every $n_k, n_k \in \mathcal{N}$
    If $S_k - \lambda^{-1} < S_{k+1} - \lambda^{-1}$ then Set $n_k$ as the stop point
  Apply_partition $(n_k)$                                          // apply the vectors and partition data of $n_k$
  End_Estimation_Truncation
```

Fig. 7. Pseudocode of the pruning algorithm for multihypothesis variable block-size prediction for each macroblock. We use the following notations: $\mathcal{N}$ is a list of partitioning points in the macroblock, it contains items in the form $n_i = \{R, D, \cup_{b_r, b_c} h(b_r, b_c), [(r_s, c_s), p_s]\}$ where R, D are the rate, distortion, respectively, $\cup_{b_r, b_c} h(b_r, b_c)$ is the union of the quadrants of the subblock (b_r, b_c) each having a hypothesis $h(b_r, b_c)$ and $(r_s, c_s), p_s$ indicate the subblock coordinates in the macroblock; $\mathcal{N}_{\text{deleted}}$ contains points that have been removed from $\mathcal{N}$; Estimate$_R(\bullet)$ estimates the rate for coding the motion vector data of the subblock (b_r, b_c) using the first-order entropy, Estimate$_D(\bullet)$ estimates the prediction error of the subblock (b_r, b_c); R_{rem}, D_{rem} contain the rate, distortion of the remaining are of the macroblock (besides area covered by the current subblock); λ is the Lagrangian control parameter.

(partitioning point) of hypotheses of the current subblock or its quadrants. The second pass, **RD_Prune**, scans the list of acceptable points $\mathcal{N}$ to establish a monotonically decreasing slope value S_k for each point n_k, $n_k \in \mathcal{N}$, similar to the rate-distortion optimization procedure of JPEG2000 [34]. Finally, the third pass minimizes the Lagrangian cost function $R(n_k)+\lambda \cdot D(n_k)$ by establishing the partitioning point $n_{k_{\min}} = \arg\min_{n_k \in N}\left(\left|S_k - \lambda^{-1}\right|\right)$, i.e. the partitioning point with a slope value closest to λ^{-1}. The splitting and hypothesis number for each subblock (or for the subblock's quadrants) is used for motion compensation, if no additional partition levels are to be performed. Otherwise, an additional level of macroblock split occurs and the multihypothesis ME and pruning occur for the subblocks that have been selected in the previous levels, and their quadrants.

4.2. Update step

The application of the update step for the creation of the L_t frame (Fig. 5) occurs in two consecutive phases, schematically shown in Fig. 8; first the update information is created by inverting the error frame samples of the H_{t-1}, H_t frames using the inverted motion-vector fields. To avoid strong motion-related artifacts in the output L_t frame and the irregular increase of the image-sample magnitudes in multiconnected areas, a normalization process divides the magnitude of the update samples for each pixel with the number of connections. Finally, before the update coefficients are added to the reference frame, they are scaled according to the lifting equation for the update step, taking into account the type of the specific connection (Haar or 5/3 filter-pair [7]). Additional scaling can be incorporated for the areas where the update samples have large magnitudes, or, alternatively, the update step can be adaptively disabled in areas where bad connections due to motion-prediction failure is seen [5]. These approaches typically require additional signaling information to be transmitted to the decoder. We do not investigate these options for the update step in this paper.

5. Experimental evaluation

In this section, we evaluate the coding performance obtained with experimental instantiations of the proposed video coding framework following the two major aspects treated in this paper. Our comparisons are grouped on three topics: First, we evaluate the different modes available for the proposed multihypothesis ME algorithm of Section 4 in the SDMCTF and IBMCTF frameworks using the equivalent setting for both frameworks. In order to assess better the increase in coding efficiency offered by the proposed multihypothesis framework, a comparison is carried out against the MC-EZBC fully scalable video coder [5] as well as the non-scalable advanced video coder (AVC), jointly standardized by MPEG and ITU. Lastly, we attempt a comparison of SDMCTF and IBMCTF in terms of behavior under resolution and temporal scalability.

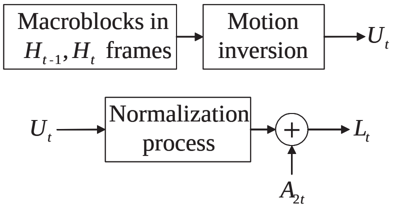


Fig. 8. The application of the update step in the MCTF process.

5.1. Instantiation of the experimental test suites

For all our experiments, we use four temporal decomposition levels. The default scaling factors of the lifting implementations of the temporal transforms (Haar, 5/3) is used for the temporal filtering. For the case of the 1/2 and 1/3 filters (i.e. prediction-only without update), the frames of each temporal level t remain the original input frames, scaled by $(\sqrt{2})^{t-1}$.

The variable block-size motion estimation algorithm of Section 4 is used. For the case of IBMCTF, block-based motion estimation with a block size of $B \times B$ pixels in the spatial domain corresponds to k separate motion estimations (for a total of k spatial-resolution levels of in-band motion estimation) with triplets of blocks of $(B/2^l) \times (B/2^l)$ wavelet coefficients in the high-frequency subbands of each resolution l, $1 \leqslant l < k$, as described in Section 2.4. The number of hypotheses used was set to 1 or 2, and this always corresponds to temporal multihypothesis prediction. If multiple in-band motion estimations are performed ($k > 1$), the number of vectors corresponding to each spatial location in the IBMCTF codec varies according to the temporal filtering of each resolution.

Concerning the compression aspects, the motion vectors of each frame (and resolution in the case of IBMCTF) are compressed using adaptive arithmetic coding; no spatial or temporal motion-vector correlations were exploited. The quantization and entropy coding is performed with the QuadTree-Limited (QT-L) coder of [29], which is an intra-band embedded wavelet coding algorithm combining quadtree coding and block-based coding of the significance maps [29]. Scaling in resolution is performed by skipping resolution levels of the compressed bitstream (and their corresponding motion-vectors in the case of IBMCTF). Dyadically reduced frame-rates are straightforwardly produced by skipping frames that correspond to the first temporal levels. Finally, rate scaling is performed by a bitstream extractor that follows the principles of the bitstream scaling method used in [5].

5.2. *Efficiency of multihypothesis MCTF in SDMCTF and IBMCTF frameworks*

We applied the proposed MCTF technique with a different number of features enabled for each experiment in order to assess the impact on the coding efficiency of MCTF. Typical results are given in Fig. 9 for the IBMCTF and SDMCTF for 128 frames of the CIF sequences “Stefan” and “Coastguard”, which include a variety of object motions. Since no attempt was made to jointly optimize the prediction performance across resolutions, in the experiments of this subsection we set $k = 1$ (one level of in-band prediction and update); two additional levels of spatial transform are performed in the MCTF residuals of the LL subbands in order to match the SDMCTF codec in the number of spatial decomposition levels (three). The objective comparison of Fig. 9 shows that, in both the SDMCTF and IBMCTF architectures, a large PSNR gain comes from the use of

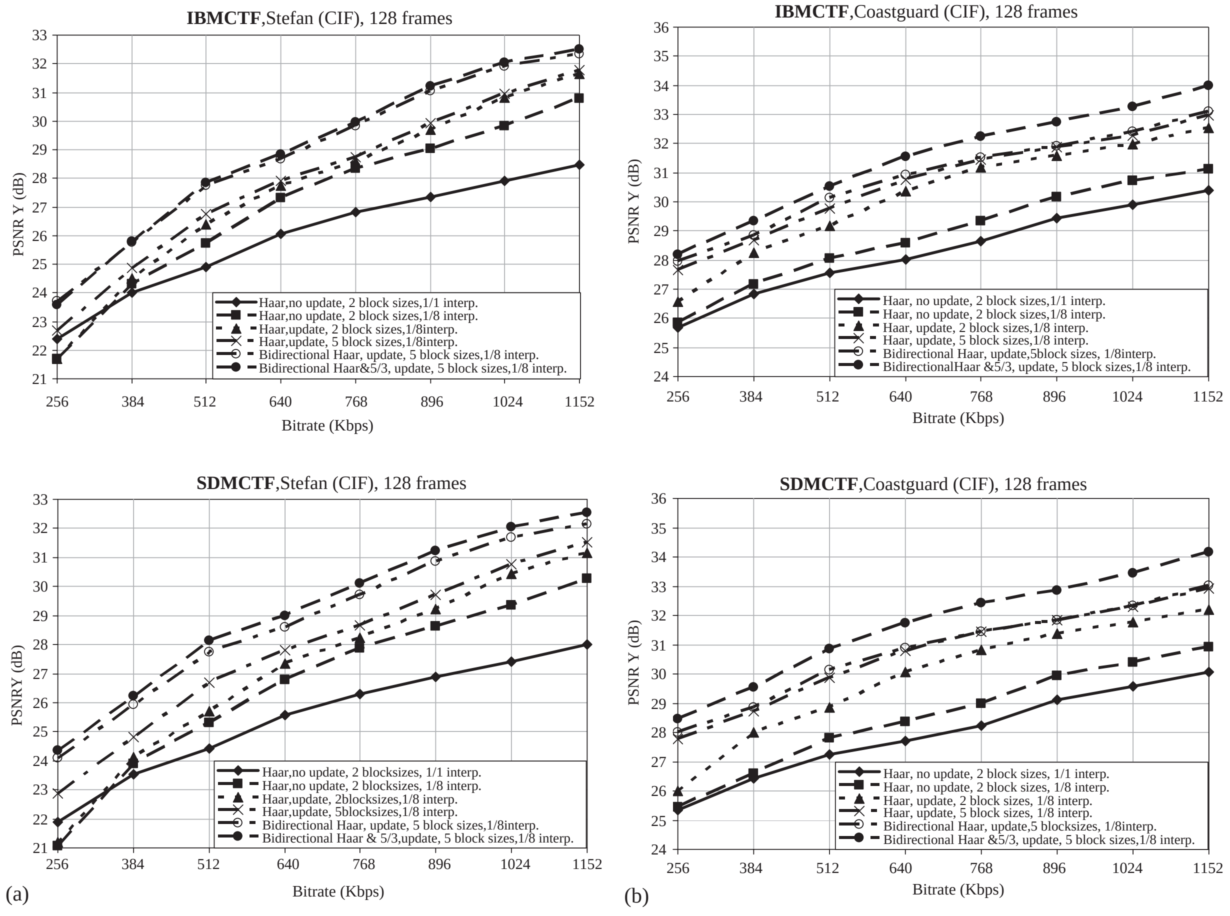


Fig. 9. PSNR results for SDMCTF and IBMCTF coders with different features: (a) Stefan sequence, (b) Coastguard sequence.

multihypothesis and longer temporal filters (5/3). Additionally, the use of the update step improves the objective performance, especially at high bit-rates. Furthermore, both architectures appear to produce comparable results for full-resolution decoding across a large range of bit-rates.

Concerning implementation complexity of the IBMCTF and SDMCTF encoders, we found that the use of all the advanced tools for the motion estimation (macroblocks pruned using five partition levels and two hypotheses) incurs a penalty of a factor of 4–6 times increase in execution time as compared to conventional block-based full search motion estimation. This result corresponds to execution-time comparisons of our platform-independent "C" implementation running on an Intel Pentium IV processor. However, preliminary testing indicates that several optimizations for motion-vector search using spiral or diamond search patterns can decrease this complexity factor significantly. In addition, the effect of the advanced prediction tools is much less severe for decoding. Our experiments indicate that only an increase by a factor of 2 is observed. Finally, the IBMCTF encoder using the proposed prediction-filters approach for the CODWT runs (on average) for approximately 150% of the SDMCTF encoding time with the same settings; IBMCTF decoding runs, on average, about 3 times slower than SDMCTF decoding.

5.3. *Efficiency of proposed temporal filtering in comparison to state-of-the-art*

Table 1 illustrates the coding results obtained with the spatial-domain MCTF using the proposed multihypothesis prediction and update step. We used the MC-EZBC results reported in [11]; the version used therein includes a number of new developments in the codec like joint luminance and chrominance-channel motion estimation, intra-prediction, adaptive update modes and bi-directional Haar filtering. The MPEG-4 AVC results were produced with the settings described in [16]. These settings correspond to the profile of the codec that includes all the advanced features like rate-distortion optimization, the CABAC scheme, full-search multiframe variable block-size motion estimation with a search range equal to ± 64 pixels and an intra-refresh period smaller than 1 s that enables random access.

Concerning the proposed codec, we used 1/8-pixel accurate motion estimation with full-search. The search range in our experiments was ± 22, ± 38, ± 70, ± 110 pixels for temporal levels

Table 1
Comparison of proposed multihypothesis MCTF in the SDMCTF framework with MC-EZBC and MPEG-4 AVC in terms of average PSNR (dB) over all the frames of four test sequences

Test	Codec	1.5 Mbps				3 Mbps				6 Mbps			
		Y	*U*	*V*	Mean	*Y*	*U*	*V*	Mean	*Y*	*U*	*V*	Mean
Night	AVC	34.51	38.73	41.34	**36.35**	37.22	40.64	43.03	**38.75**	39.98	42.57	44.7	**41.2**
	MCEZBC	31.97	34.75	38.28	**33.49**	35	37.54	40.46	**36.33**	38.16	40.56	42.8	**39.33**
	Proposed	31.41	37.75	40.36	**33.96**	34.13	40.17	42.36	**36.51**	37.12	42.75	44.32	**39.26**
Crew	AVC	36.29	40.81	40.97	**37.83**	38.52	42.02	42.8	**39.82**	40.89	43.21	44.51	**41.88**
	MCEZBC	34.99	39.17	38.65	**36.3**	37.49	41.27	41.67	**38.82**	40.04	43.06	44.3	**41.25**
	Proposed	34.39	40.51	40.77	**36.48**	36.53	42.12	43.08	**38.55**	38.42	43.63	44.67	**40.33**
Harbor	AVC	31.92	41.71	44.63	**35.67**	34.63	42.85	45.76	**37.85**	37.7	44.0	46.89	**40.28**
	MCEZBC	30.92	41.35	44.41	**34.91**	33.76	42.84	46.00	**37.31**	37.13	44.39	47.79	**40.12**
	Proposed	31.71	42.87	45.69	**35.90**	34.06	44.20	46.75	**37.87**	36.82	45.58	47.81	**40.11**
Sailor men	AVC	34.77	39.04	40.12	**36.38**	36.83	40.39	41.43	**38.19**	39.09	41.93	42.90	**40.20**
	MCEZBC	34.58	40.26	41.25	**36.64**	36.52	42.15	43.28	**38.59**	38.78	43.39	44.82	**40.56**
	Proposed	34.42	41.11	41.94	**36.79**	36.50	42.75	43.69	**38.74**	38.02	43.76	44.66	**40.09**

The values for the luminance (*Y*) and chrominance channels (*U*, *V*) are reported. The mean PSNR [11], is defined as $\text{Mean\_PSNR} = (4 \cdot \text{PSNR\_}Y + \text{PSNR\_}U + \text{PSNR\_}V)/6$.

1–4, respectively, and we set $\lambda = 62$ (Lagrangian multiplier) for variable block-size ME pruning—see Fig. 7. A maximum of $M = 2$ hypotheses were used and each macroblock was pruned with dyadically reduced block sizes ranging from 64×64 down to 4×4 pixels. The 9/7 filter-pair was used for the spatial decomposition (four resolution levels). In the temporal direction, according to the pruning of the motion information during the predict step, the bi-directional Haar and the 5/3 filter-pairs were used. An implementation with a sliding window was chosen to avoid large PSNR fluctuations in the GOP borders.

The results show that, for half of the sequences of the test, multihypothesis SDMCTF-based video coding is comparable or superior in terms of mean PSNR to the highly optimized AVC over a large range of bit-rates. At the same time, the proposed scheme retains the advantages offered by fully embedded coding. Additionally, we find the proposed coder to be comparable or superior to MC-EZBC over a large range of bit-rates. Notice that in comparison to AVC's temporal prediction algorithm, the proposed algorithm does not yet include intra-prediction modes, while the motion-vector coding can be significantly improved with the use of prediction-based techniques. In addition, in comparison to MC-EZBC, we do not use techniques to adaptively disable the update step in the cases where the prediction fails to provide a good match; as a result, coding efficiency is decreased in when the motion model fails to predict efficiently (e.g. Night and Crew sequences).

5.4. Performance of spatial-domain and in-band MCTF for full-scalability

It is generally difficult to establish an objective metric for evaluation under temporal and resolution scalability since, under the MCTF framework, every coder creates its own unique reference sequence for low-resolution/low frame-rate video. Nevertheless, the ability of that reference sequence to serve as a unique original is questionable. Here we opt for the use of predict-step only MCTF in order to circumvent the problem of frame-rate scaling. In addition, we use both IBMCTF and SDMCTF with the same settings, i.e. half-pixel accurate ME with maximum two hypotheses and two reference frames, five block-sizes and a fixed search range per temporal level. In this scenario, however, the IBMCTF codec provides scalability for the motion vector bit-rate by using two levels of in-band prediction.

The results for the full-resolution/full frame-rate decoding are given in Fig. 10(a). We find that the use of multiple motion vectors for each resolution level may increase the coding performance of IBMCTF in the case of complex motion, as seen in the Football sequence. Nevertheless, without careful optimization, the increased motion-vector bit-rate may overturn this advantage for sequences with medium motion activity, like the Foreman sequence.

Fig. 10(b) depicts the performance for decoding at half resolution/half frame-rate and different quality levels. The reference sequences used in the comparison of Fig. 10(b) are obtained by one-level spatial DWT performed on a frame-by-frame basis, followed by retaining the *LL* subbands and frame skipping. We found that the reference sequences produced in this manner are always artifact-free and independent of the utilized motion model.

Both IBMCTF and SDMCTF can achieve resolution-scalable decoding. However, it is important to notice that in a scalable scenario corresponding to coarse-to-fine progressive transmission of different resolutions, only the IBMCTF guarantees lossy-to-lossless decoding at all resolutions if the *LL* subbands are used as an original [3]. This is observed experimentally in Fig. 10(b), where a large PSNR difference exists between the two alternatives. A typical visual comparison is given in Fig. 11.

6. Conclusions

A novel framework for fully scalable video coding that performs open-loop motion compensated temporal filtering in the wavelet domain (in-band) was presented in this paper. To overcome the shift variance of the DWT, a

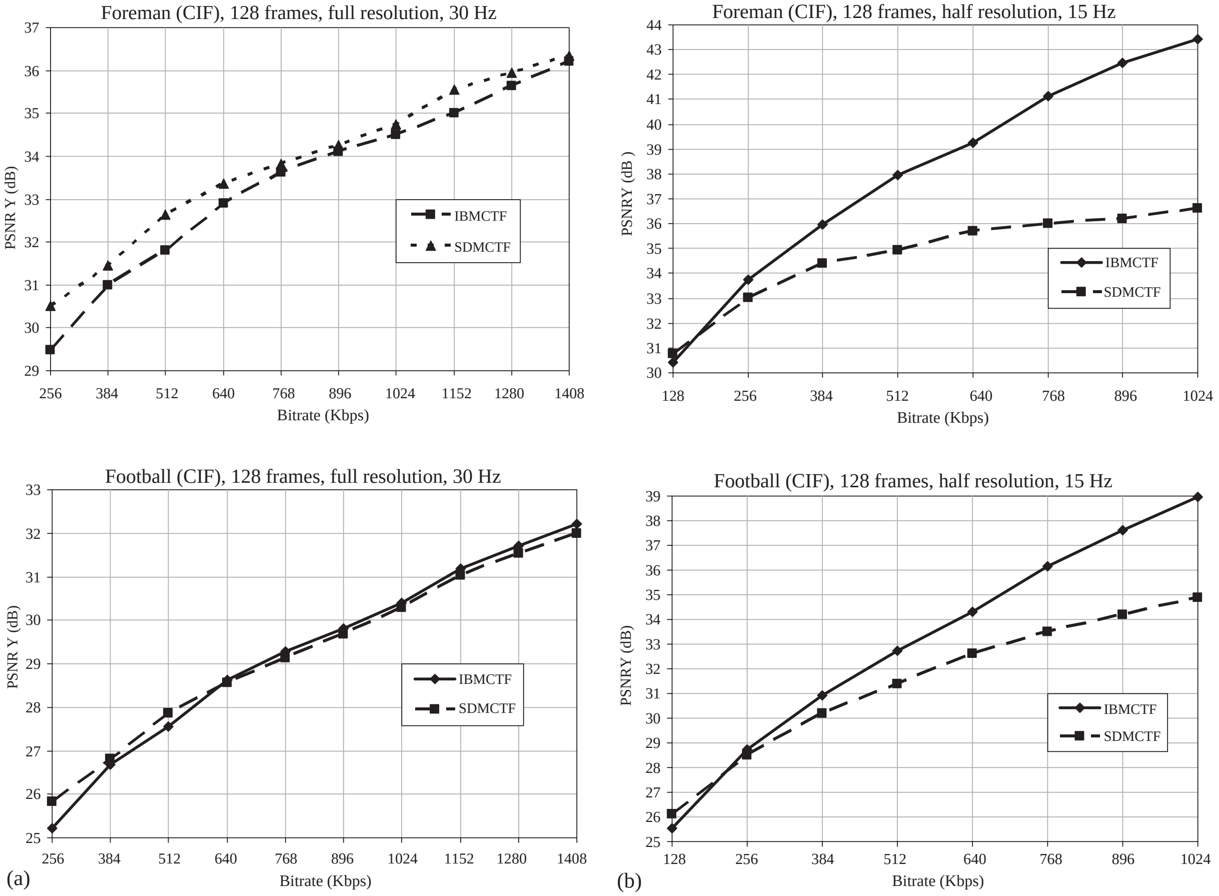


Fig. 10. Comparison between in-band MCTF and spatial-domain MCTF: (a) full resolution/full frame-rate, (b) half resolution/half frame-rate. For each sequence, all points were made by decoding a different number of resolution, quality or temporal levels. For the PSNR comparison at the low spatial resolution, the uncoded *LL* subband of the original sequence is used.

complete-to-overcomplete DWT is performed for each reference frame. Driven by the resolution-scalable operation of the in-band MCTF, the CODWT is performed via a single-rate calculation scheme that uses the DWT subbands of the current resolution level. The problem of improving the coding efficiency of both spatial-domain and in-band MCTF is addressed by a new algorithm for optimized multihypothesis motion estimation. The experimental evaluation of its effects in both architectures gives positive results. More information (sample bitstreams with executables for the IBMCTF and SDMCTF bitstream-extraction and decoding, as well as additional documentation) can be downloaded from our ftp site [4].

Although spatial-domain and in-band MCTF equipped with multihypothesis prediction and update appear to be comparable in coding efficiency over a large range of bit-rates under the same experimental conditions, the in-band architecture additionally permits the independent temporal filtering of each resolution of the input content. As a result, the proposed in-band MCTF framework enables many potential developments for multiresolution decoding and can be a viable approach for fully scalable video compression.

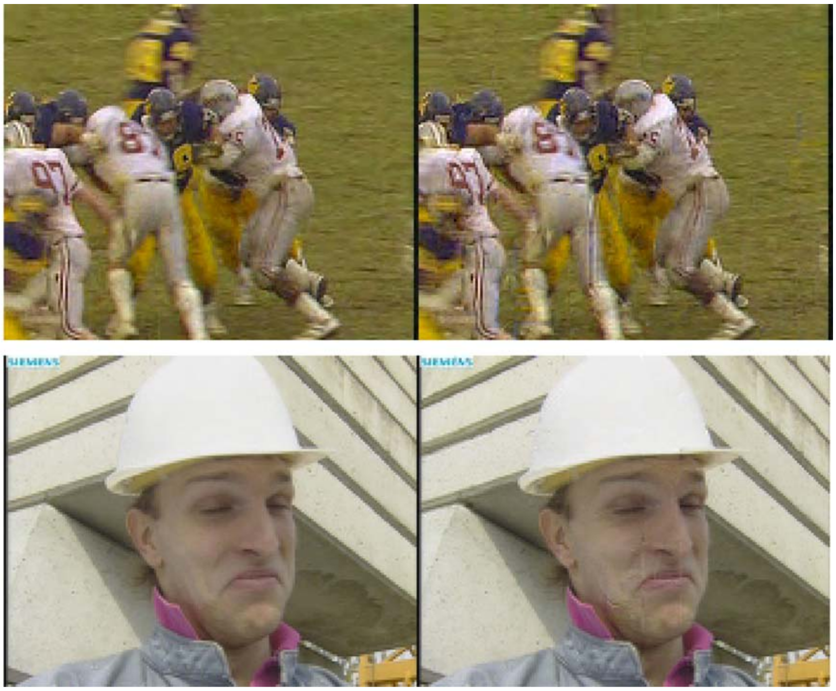


Fig 11. Visual comparison for the half-resolution/half frame-rate decoding at 768 kbps with IBMCTF (left) and SDMCTF (right). Even at high bit-rate, due to the fact that spatial filtering and motion compensation are not commutative in SDMCTF [3], motion artifacts are occasionally observed in the areas with irregular motion (player leg in the middle area of football, mouth of foreman).

Acknowledgements

This work was supported in part by the Federal Office for Scientific, Technical and Cultural Affairs (IAP Phase V—Mobile Multimedia, the Flemish Institute for the Promotion of Innovation by Science and Technology (GBOU RESUME and PhD-bursary J. Barbarien) and by the European Community under the IST Program (Mascot, IST-2000-26467). P. Schelkens has a post-doctoral fellowship with the Fund for Scientific Research—Flanders (FWO), Egmontstraat 5, B-1000 Brussels, Belgium.

References

[1] Y. Andreopoulos, A. Munteanu, G. Van der Auwera, J. Cornelis, P. Schelkens, Complete-to-overcomplete discrete wavelet transforms: theory and applications, IEEE Trans. Signal Process., to appear.

[2] Y. Andreopoulos, A. Munteanu, G. Van der Auwera, P. Schelkens, J. Cornelis, Scalable wavelet video-coding with in-band prediction—implementation and experimental results, in: Proceedings of the IEEE International Conference on Image Processing, Vol. 3, Rochester, USA, September 2002, pp. 729–732.

[3] Y. Andreopoulos, M. Van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, J. Cornelis, Fully scalable wavelet video coding using in-band motion compensated temporal filtering, in: Proceedings of the International Conf. on Acoustics Speech and Signal Processing, ICASSP 2003, Vol. 3, Hong-Kong, CN, March 2003, pp. 417–420.

[4] Anonymous FTP: ftp://ftp.etro.vub.ac.be Login: etroguest, Password: anonymous1! Directory: ./ETRO/Yiannis_Andreopoulos/IBMCTF.

[5] P. Chen, J.W. Woods, Bidirectional MC-EZBC with lifting implementation, IEEE Trans. Circuits and Systems for Video Technol., preprint submitted for publication.

[6] S.-J. Choi, J.W. Woods, Motion-compensated 3-D subband coding of video, IEEE Trans. Image Process. 3 (2) (February 1999) 155–167.

[7] I. Daubechies, W. Sweldens, Factoring wavelet transforms into lifting steps, J. Fourier Anal. Appl. 4 (3) (March 1998) 247–269.

[8] M. Flierl, B. Girod, Multihypothesis motion estimation for video coding, in: Proceedings of the Data Compr. Conference, DCC 2001, Snowbird, UT, April 2001, pp. 341–350.

[9] M. Flierl, T. Wiegand, B. Girod, Rate-constrained multihypothesis prediction for motion-compensated video compression, IEEE Trans. Circuits Systems Video Technol. 12 (11) (November 2002) 957–969.

[10] B. Girod, The efficiency of motion-compensated prediction for hybrid video coding of video sequences, IEEE J. Sel. Areas Commun. SAC-5 (August 1987) 1140–1154.

[11] A. Goldwelkar, I. Bajic, J.W. Woods, Response to call for evidence on scalable video coding, ISO/IEC JTC1/SC29/WG11, m9723 (MPEG), Trondheim, Norway, July 2003.

[12] S. Han, B. Girod, SNR scalable coding with leaky prediction, ITU-T Q.6/SG16, VCEG-N53, Santa Barbara, USA, 2001.

[13] Y. He, R. Yan, F. Wu, S. Li, H.26L-based fine granularity scalable video coding, ISO/IEC JTC1/SC29/WG11, Document M7788, Pattaya, December 2001.

[14] H.C. Huang, C-N. Wang, T. Chiang, Arobust fine granularity scalability using trellis based predictive leak, IEEE Trans. CSVT 12 (6) (June 2002) 372–385.

[15] ISO/IEC JTC 1/SC29/WG1, FCD 15444-1, JPEG 2000 Image Coding System, Dec. 2000, www.jpeg.org

[16] ISO/IEC JTC1/SC29/WG11, w5559, Call for Evidence on Scalable Video Coding Advances (MPEG), Pattaya, Thailand, March 2003.

[17] G. Karlsson, M. Vetterli, Three-dimensional sub-band coding of video, paper M7.9, in: Proceedings of the International Conference on Acoustics Speech and Signal Processing, ICASSP 1988, New York, NY, US, May 1988, p. 1100.

[18] H.S. Kim, H.W. Park, Wavelet-based moving-picture coding using shift-invariant motion estimation in wavelet domain, Signal Process. Image Commun. 16 (March 2001) 669–679.

[19] B.-J. Kim, Z. Xiong, W. Pearlman, Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT), IEEE Trans. Circuits System Video Technol. 10 (8) (December 2000) 1374–1387.

[20] A.S. Lewis, G. Knowles, Video compression using 3D wavelet transforms, Electron. Lett. 26 (6) (March 1990) 396–398.

[21] W. Li, Streaming video profile in MPEG-4, IEEE Trans. Circuits Systems Video Technol. 11 (3) (2001) 301–317.

[22] X. Li, L. Kerofski, S. Lei, All-phase motion compensated prediction in the wavelet domain for high performance video coding, in: Proceedings of the IEEE International Conference on Image Processing, Vol. 3, Thessaloniki, GR, October 2001, pp. 538–541.

[23] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, San Diego, CA, 1998.

[24] J.R. Ohm, Three-dimensional subband coding with motion compensation, IEEE Trans. Image Process. 3 (5) (September 1994) 559–571.

[25] H.W. Park, H.S. Kim, Motion estimation using low-band-shift method for wavelet-based moving-picture coding, IEEE Trans. Image Process. 9 (4) (April 2000) 577–587.

[26] B. Pesquet-Popescu, V. Bottreau, Three-dimensional lifting schemes for motion compensated video compression, in: Proceedings of the International Conference on Acoustics Speech and Signal Processing, ICASSP 2001, Vol. 3, Salt Lake City, Utah, US, May 2001, pp. 1793–1796.

[27] H.M. Radha, M. Van der Schaar, Y. Chen, The MPEG-4 fine-grained scalable video coding for multimedia streaming over IP, IEEE Trans. Multimedia 3 (1) (March 2001) 53–68.

[28] H. Sari-Sarraf, D. Brzakovic, A shift-invariant discrete wavelet transform, IEEE Trans. Signal Process. 45 (10) (October 1997) 2621–2626.

[29] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro I Nieto, J. Cornelis, Wavelet coding of volumetric medical datasets, IEEE Trans. Med. Imaging 22 (3) (March 2003) 441–458, [special issue on "wavelets in medical imaging"].

[30] A. Secker, D. Taubman, Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting, in: Proceedings of the International Conference Image Processing, ICIP 2001, Vol. 2, Thessaloniki, GR, October 2001, pp. 1029–1032.

[31] J. Skowronski, Pel recursive motion estimation and compensation in subbands, Signal Process. Image Commun. 14 (5) (April 1999) 389–396.

[32] D. Taubman, M. Marcellin, JPEG2000: Image Compression Fundamentals, Practice and Standards, Kluwer Academic Publishers, Dordrecht, 2001.

[33] D. Taubman, A. Zakhor, Multirate 3-D subband coding of video, IEEE Trans. Image Process. 3 (September 1994) 572–588.

[34] D. Turaga, M. van der Schaar, Wavelet coding for video streaming using new unconstrained motion compensated temporal filtering, in: Proceedings of the Internat. Workshop on Digital Communications: Advanced Methods for Multimedia Signal Processing, Capri, IT, September 2002, pp. 41–48.

[35] G. Van der Auwera, A. Munteanu, P. Schelkens, J. Cornelis, Bottom-up motion compensated prediction in the wavelet domain for spatially scalable video, Electron. Lett. 38 (21) (October 2002) 1251–1253.

[36] Y. Wang, S. Cui, J. Fowler, 3D video coding using redundant-wavelet multihypothesis and motion-compensated temporal filtering, in: Proceedings of the IEEE. International Conference on Image Processing, ICIP 2003, Vol. 2, Barcelona, ES, September 2003, pp. 755–758.

[37] F. Wu, S. Li, R. Yan, X. Sun, Y-Q Zhang, Efficient and universal scalable video coding, in: Proceedings of the IEEE International Conference on Image Processing, Vol. 2, Rochester, USA, (September 2002) pp. 37–40.

[38] F. Wu, S. Li, Y.-Q. Zhang, A framework for efficient progressive fine granularity scalable video coding, IEEE Trans. Circuits Systems Video Technol. 11 (3) (2001) 332–344.

[39] Y. Zhan, M. Picard, B. Popescu, H. Heijmans, Long temporal filters in lifting schemes for scalable video coding, ISO/IEC JTC1/SC29/WG11 (MPEG), m8680, Klagenfurt, July 2002.