Unconstrained Motion Compensated Temporal Filtering (UMCTF) for Efficient and Flexible Interframe Wavelet Video Coding[♣]

D. S. Turaga[1+], M. van der Schaar[1*], Y. Andreopoulos[2], A. Munteanu[2] and P. Schelkens[2]
[1]Philips Research USA [2]Vrije Universiteit Brussel

Abstract

We introduce an efficient and flexible framework for temporal filtering in wavelet based scalable video codecs called unconstrained motion compensated temporal filtering (UMCTF). UMCTF allows for the use of different filters and temporal decomposition structures through a set of controlling parameters that may be easily modified during the coding process, at different granularities and levels. The proposed framework enables the adaptation of the coding process to the video content, network and end-device characteristics, allows for enhanced scalability, content-adaptivity and reduced delay, while improving the coding efficiency as compared to state-of-the-art motion-compensated wavelet video coders. Additionally, a mechanism for the control of the distortion variation in video coding based on UMCTF employing only the predict step is proposed. The control mechanism is formulated by expressing the distortion in an arbitrary decoded frame, at any temporal level in the pyramid, as a function of the distortions in the reference frames at the same temporal level. All the different scenarios proposed in the paper are experimentally validated through a coding scheme that incorporates advanced features (such as rate-distortion optimized variable block-size multihypothesis prediction and overlapped block motion compensation). Experiments are carried out to determine the relative efficiency of different UMCTF instantiations, as well as to compare against the current state-of-the-art in video coding.

1. Introduction

Reliable transmission of video over wireless networks poses many challenges. This is not only due to the inherently lower bandwidth provided by wireless networks, but also due to the bandwidth variations and losses caused by fading, interference, mobility and competing traffic. Furthermore, many applications require video content adaptability to different end-devices, characterized by widely varying features and capabilities. Thus, successful wireless transmission of video requires efficient coding, adaptability to varying network conditions, device characteristics, and user preferences, and error resilience.

[♣] to appear in *Signal Processing: Image Communication*
[+] Currently with IBM Research, Hawthorne.
[*] Contact author, currently with University of California, Davis. Email: mvanderschaar@ece.ucdavis.edu

H.264 is an emerging predictive video coding standard that provides very high coding efficiency. It uses efficient motion estimation and compensation techniques, combined with integer transforms, R-D optimization, and context based arithmetic coding to achieve significant improvements over previous standards such as MPEG-1, 2 and 4 and the older H.261 and H.263. More details on the algorithms used in H.264 may be obtained from the work by Wiegand et al [1], and Wiegand and Girod [2]. However, the H.264 bitstreams are optimized for a specific target bit rate, and cannot be easily adapted to varying network conditions without suffering a considerable loss in coding efficiency[1]. Moreover, these bitstreams are not easily adaptable to device capabilities and user preferences. Hence, H.264 does not provide a complete solution to wireless video transmission.

Extensions to the predictive coding framework [3] have been proposed to increase its adaptability to network and device characteristics. Among these extensions are the MPEG-4 spatial scalability and Fine Granular Scalability (FGS) [4]. However, these scalability approaches provide significantly lower coding efficiency than non-scalable video coders, thereby making them unsuitable for low-bandwidth wireless video transmission.

Unlike predictive coding based scalable coders, wavelet video coding schemes can provide very flexible spatial, temporal, SNR and complexity scalability with fine granularity over a large range of bit-rates, while maintaining a high coding efficiency. Early contributions to the field of wavelet and multi-resolution video coding were provided, among others, by Gharavi [5], Zhang and Zafar [6], and by Taubman and Zakhor [7]. Furthermore, recent advances in wavelet-based image compression have motivated and significantly influenced wavelet video coding algorithms. For instance, the Set Partitioning into Hierarchical Trees (SPIHT) algorithm [8], was later extended to 3D wavelet video coding by Kim et al [9]. Various approaches have been proposed in the area of wavelet-based video coding, and these have been classified into the following categories by Ohm and Ebrahimi [10]:

- **wavelet in loop** - that preserve the conventional predictive coder structure, but replace the DCT, for the residual error in the motion-compensation prediction loop, with the wavelet transform;
- **in-band prediction** - where the spatial wavelet transform for each frame is performed first, followed by exploitation of interframe redundancy by predicting the wavelet coefficient values, or by defining temporal contexts in entropy coding;

[1] In H.264, switching frames (S-frames) were introduced to provide easy adaptability to bandwidth variations by switching between two non-scalable coded bitstreams. However, this assumes storage of the same video content at a multitude of bit rates and resolutions that is often not practical for wireless transmission systems or for applications expanded over large and heterogeneous networks.

- **interframe wavelet -** that perform wavelet filtering along the temporal axis followed by 2D spatial wavelet transform. Alternatively, it was recently proposed that the order of the transforms can be switched (2D+t) [33], leading to the so-called in-band class of interframe wavelet video coding algorithms.

From the first class of codecs, Blasiak and Chan [11] and Asbun et al [12] proposed closed-loop compression schemes (the decoded signal is used as reference during motion estimation) with 'wavelet in loop'. The disadvantage of these techniques is that whenever the entire residual signal is included into the motion-compensation prediction loop, drift occurs if decoding is performed at lower bit-rates. Moreover, if the residual signal is *not* entirely included into the motion-compensation prediction loop, then there is a considerable coding penalty associated with SNR scalability. Spatial scalability obtained with this scheme also suffers from drift effects and/or significant compression inefficiencies. Furthermore, as mentioned in [10], motion compensation is more complex than for DCT-based predictive coders. This is because the mismatch between motion boundaries and wavelet basis functions needs to be overcome by applying, for instance, a smoothing scheme like overlapped block motion compensation (OBMC).

The second category, 'in-band prediction', can achieve spatial scalability without experiencing drift, as the motion-compensation prediction is applied separately in each spatial resolution level. However, if a closed-loop prediction structure is used *within* these spatial levels, drift still occurs as soon as cropping of bits for quality (SNR) scalability is applied. Typical coding results obtained with such schemes can be found in [13].

Alternatively, 'interframe wavelet coding' does not employ a closed-loop structure for removing the temporal redundancies. Instead, it performs motion compensated temporal filtering (MCTF), as proposed first by Ohm [14] and later improved by Choi and Woods [15]. MCTF is a *superset* of the MC-prediction paradigm. The motion-compensated temporal lowpass filter separates noise and sampling artifacts from the content relevant over time, while the prediction establishes similarity to MC frame-rate conversion. In the temporal pyramid resulting from MCTF, the high-pass frames can be encoded more coarsely than the lowpass frames. Even more, if the highpass frames are discarded, synthesis is performed purely based on the low-pass temporal frames and motion information. A very efficient codec using MCTF and an intra-band coding technique, called embedded zero block coding (EZBC), was proposed by Hsiang and Woods in [16]. This codec is labeled Motion Compensated EZBC or MC-EZBC. Xu et al [17] introduced embedded subband coding with optimized truncation (ESCOT) in which they further extended the MCTF to filter across longer motion threads. A summary of wavelet video compression techniques is provided by Woods et al in [18]. Advances in the area include the implementations of MCTF

using lifting, proposed recently in [22] and [23]. Lifting decomposes the temporal filtering into a *predict* step and an *update* step, and can significantly improve the flexibility and efficiency of MCTF as it allows for synthesizing various temporal filters using classical lifting structures.

In this paper we introduce a framework for temporal filtering in wavelet interframe codecs called the unconstrained motion compensated temporal filtering (UMCTF) [19], [20], [21]. The UMCTF framework uses lifting and appropriate temporal filters, adapted to the video content, to enable flexibility in temporal scalability and reduce delay, while also improving coding efficiency. Furthermore, we address the issue of the lack of orthonormality in the case of UMCTF employing only the predict step, and we describe a mechanism for the control of the distortion variation in the decoded video sequence.

This paper is organized as follows. In Section 2 we first introduce MCTF and describe Haar MCTF, as proposed in [15], and highlight its problems and inefficiencies. Section 3 introduces UMCTF and describes the different choices for the filters and the decomposition structures, which enable various enhancements as compared to Haar MCTF. In section 4 we present the mechanism for the control of the distortion variation in the decoded sequence in the case of low-delay MCTF involving only the predict step. Moreover, a delay analysis for this particular instantiation of UMCTF is presented in Section 5. We summarize our results, including content-adaptability, improved temporal scalability and distortion control, in Section 6. Finally, conclusions and directions for future work are presented in Section 7.

2. MCTF

2.1. Haar MCTF

MCTF was first proposed by Ohm [14] and later improved by Choi and Woods [15]. Unlike closed-loop (predictive) coding, where decoded frames are used as references for the motion compensation of future frames, MCTF does not employ a temporal recursive structure. Instead, the original frames are filtered temporally in the direction of motion (see Figure 1), with the resulting filtered frames being transformed and coded using 2D spatial wavelet transforms and embedded coding. At transmission time, only the desired portion of the embedded interframe bitstream is sent, but no drift is incurred, due to the open-loop structure of both the encoder and decoder.

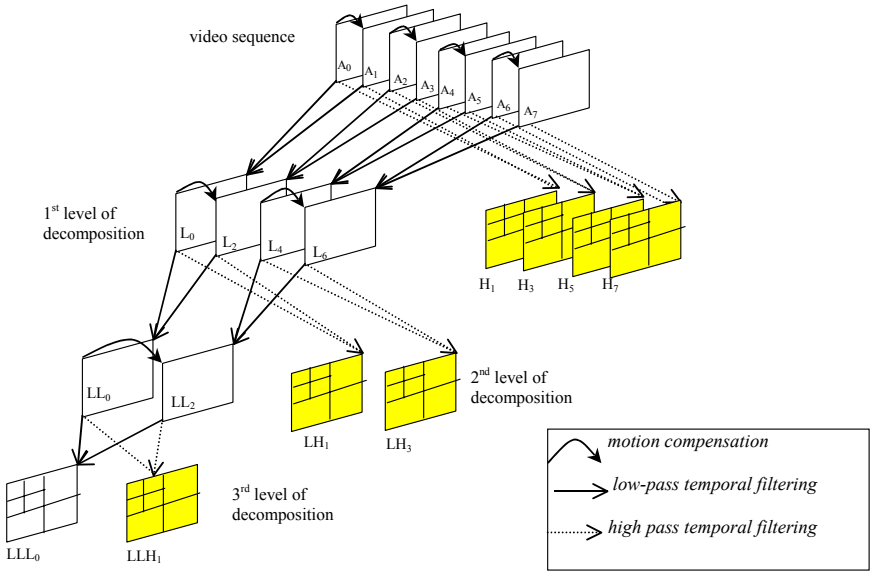


Figure 1. Motion Compensated Temporal Filtering (MCTF).

In Haar MCTF, successive pairs of frames are temporally filtered using a two-channel Haar filter-bank to create low-pass (L) and high-pass (H) frames. This filtering operation may be written as

$$L_0\left(y+v_y, x+v_x\right)=\frac{1}{\sqrt{2}}\left[A_0\left(y+v_y, x+v_x\right)+A_1(y,x)\right]$$
$$H_1(y,x)=\frac{1}{\sqrt{2}}\left[A_1(y,x)-A_0\left(y+v_y, x+v_x\right)\right] \quad ,$$

where $\left(v_y, v_x\right)$ is the motion vector[2] connecting pixels from the two frames. Un-referenced pixels, in the reference frame, are copied into the L frames, while pixels referenced multiple times are stored only once. This is done to avoid the appearance of "holes" in the filtered L frames; more details on handling such pixels may be obtained from [15].

In order to remove long-term temporal dependencies in the sequence, L frames are further decomposed using a pyramidal or multi-resolution decomposition structure using the same Haar filter-bank.

2.2. Constraints and inefficiencies in Haar MCTF

The Haar MCTF framework suffers from several constraints and inefficiencies.

- **Low efficiency temporal filtering.** Uni-directional[3] motion-estimation, used in Haar MCTF, does not perform efficiently in the presence of irregular motion, or scene changes. As a result,

[2] MCTF is performed after motion estimation, and in general does not include the motion model used, or the accuracy of the motion vectors etc.

[3] Note that in [16], the reference for the B-frames pixels can be found from either the previous or the next A-frames. However, the prediction is still unidirectional (i.e. the temporal filter has a fixed length of 2), and a flag indicates to the decoder which reference frames were used in the temporal filtering process.

compensation and filtering are performed across poorly matched regions, leading to the creation of visual artifacts in L frames (see Figure 2) and also to reduced coding efficiency.

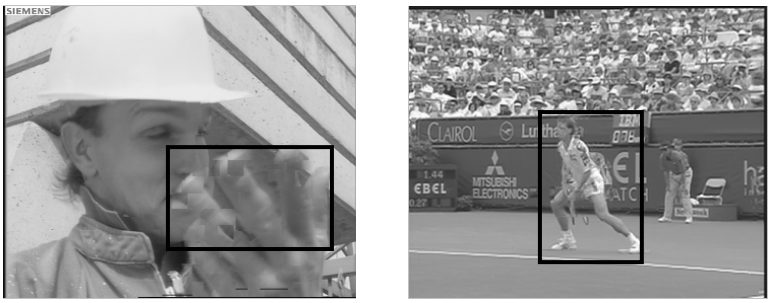


Figure 2. Visual artifacts in L frames from Foreman (left) and Stefan (right)

Figure 2 shows sample artifacts (enclosed in black rectangles) occurring in L-frames from the sequences Foreman and Stefan. Typical artifacts include blurring and stretching and sometimes blockiness (due to block-based motion estimation) and shadowing.

- **Low quality temporal scalability.** Since temporal scalability is achieved in MCTF-based interframe wavelet coding by transmitting only the L-frames associated with a specific frame rate, poor quality L-frames translate directly into low visual quality when the video is decoded at lower temporal rates. This also directly affects the visual quality for spatial scalability, especially when decoding at lower temporal frame rates.
- **Constrained temporal scalability.** Although with the current MCTF structure, dyadic (powers-of-two e.g. half, quarter, one-eighth) frame-rate scalability is easy to achieve, non-dyadic frame-rates cannot be provided, thereby limiting the flexibility of temporal scalability.
- **Increased delay.** Due to the low-pass filtering of the frames at the various resolutions, all the H-frames within the Group-Of-Frames (GOF) need to be received before the frames can be decoded at the full frame rate.

These inefficiencies are a direct consequence of the rigid temporal filtering methods in the Haar MCTF, i.e. fixed filter choice, fixed number of levels etc.

2.3. Lifting Implementations of MCTF

Lifting implementations of MCTF [22] and [23] increase its flexibility significantly and as a result, improved motion-compensation features like bi-directional filtering, multiple reference frames etc. can all be introduced into MCTF. Lifting consists of a cascade of two steps: a *predict* and an *update* step. In the predict step, H frames are created by high-pass temporal filtering. In the update step, the H frames, resulting from the predict step, are then used to create the L frames. As an example, Haar MCTF may be implemented using lifting as follows:

Predict: $H_1(y,x) = \frac{1}{\sqrt{2}}\left[A_1(y,x) - A_0\left(y + v_y, x + v_x\right)\right]$

Update: $L_0\left(y + v_y, x + v_x\right) = \sqrt{2}A_0\left(y + v_y, x + v_x\right) + H_1(y,x)$, or equivalently

$$L_0(y,x) = \sqrt{2}A_0(y,x) + H_1\left(y - v_y, x - v_x\right)$$

The inverse temporal filtering involves *inverse update* followed by *inverse predict* steps.

MCTF is a superset of the motion compensated prediction paradigm and the presence of a low-pass motion compensated temporal filter goes beyond traditional compression concepts. MCTF provides many advantages over motion compensated prediction:

- **A non-recursive predictive loop** is provided. As a result, no drift occurs if decoding is performed at various bit-rates and improved error resilience is provided (no successive accumulation of errors).
- **Improved scalability**. Temporal scalability is provided by transmitting only a subset of the temporal decomposition levels. The motion information, in conjunction with the decoded L frames, may be used to interpolate discarded frames[4]. Moreover, due to the non-recursive coding structure, the encoding/decoding can cease at any point, thereby allowing for a high degree of freedom in the design of complexity-scalable codecs. When MCTF is used with the 2D wavelet transform and embedded coding, spatial and SNR scalability are inherently provided. Since no drift is incurred when decoding at various bit-rates, SNR scalability is provided without a coding penalty, unlike in predictive coding.
- Due to **clear prioritization of the coded video coefficients**, the MCTF can be easily combined with unequal error protection schemes for improved error resilience.
- **Noise and sampling artifacts are separated** from the content relevant over time and MCTF effectively removes long range as well as short range temporal redundancies.

The proposed UMCTF framework generalizes MCTF as it allows for the adaptation of the decomposition structure, the number of decomposition levels, the temporal-filter choice, etc. This framework retains thus the advantages of MCTF over motion-compensated prediction, and in the same time, it enhances MCTF as it allows for further reducing the uncertainty about the true motion by an adaptive choice of the employed temporal transforms and decomposition structures. The proposed UMCTF framework is described next.

[4]When the motion matches are imperfect, the L frames contain visual artifacts and may be unsuitable for display.

3. Unconstrained Motion Compensated Temporal Filtering (UMCTF)

For ease of presentation, we first introduce the notation used in the remainder of the paper.

3.1. Notation

N : Number of frames in the GOF that are temporally filtered together

D : Number of levels in temporal decomposition pyramid; the frames at level $d = 0$ are the original frames

N_d : Number of frames at level $d \in [0, D]$

L_i^d : Low-pass filtered frames at level $d \in [0, D]$, $i \le N^d - 1$

A_i^d : Low-pass filtered frames using a delta filter (all-pass filter) at level $d \in [0, D]$, $i \le N^d - 1$; A_i^0 are the original frames.

H_i^d : High-pass filtered frames at level $d \in [0, D]$, $i \le N^d - 1$

M_d : Temporal sub-sampling factor at level $d \in [0, D]$.

f_i^d : High-pass filter used to create H_i^d frames. $i \le N^d - 1$.

g_i^d : Low pass filter used to create L_i^d frames, $i \le N^d - 1$.

$\left(v_{y,k\to i}^d, v_{x,k\to i}^d\right)$: Motion vector connecting frames *k* and *i* at level $d \in [0, D-1]$

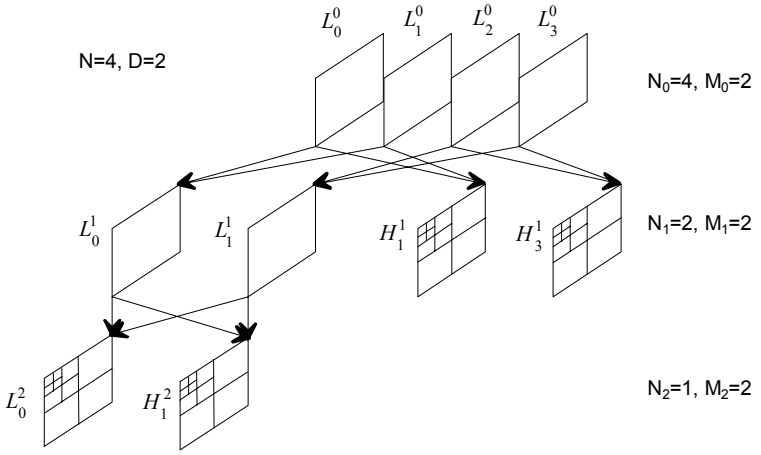


Figure 3. Illustration of UMCTF notation.

3.2. UMCTF framework

UMCTF provides adaptive temporal filtering through:

- variable number of temporal decomposition levels based on the video content or desired complexity level;
- adaptive selection of filters enabling different temporal filtering enhancements;

- adaptive selection of filters, within and between temporal and spatial decomposition levels;
- variable number of successive H frames within and between levels, for flexible (non-dyadic) temporal scalability and different temporal filtering enhancements;
- different temporal decomposition structures.

These filters may be adapted not just across the different frames and between different levels, but also within a frame. In a majority of cases, we choose from a small set of filters that may be indicated to the decoder with minimal overhead

Different multi-tap filters may be designed in order to introduce multiple reference frames, bi-directional filtering etc. in MCTF, and can be implemented using lifting. We discuss different filter choices and decomposition structures in greater detail, along with their corresponding impact on coding efficiency and scalability performance, in the following sections. A thorough analysis of the UMCTF complexity as well as several strategies for reducing this complexity are presented in [25].

We conclude this section by summarizing the flexibilities provided within the UMCTF framework; allowing for easy adaptability to video content, network or device characteristics by a simple choice of a set of "control parameters", as indicated in Table 1.

Table 1. Adaptation parameters for UMCTF

Control Parameter	**Adaptation Result**
N	Changes GOF size
D	Limits the number of temporal decomposition levels
M_d	Enables flexible temporal scalability; allow different decodable frame rates
R_p^d	Varies the number of reference frames used from the past; can be different at different levels
R_f^d	Varies the number of reference frames used from the future; can be different at different levels
g_i^d	Adaptively creates L frames with different characteristics (e.g. leaves A frames unfiltered); can be different at different levels
f_i^d	Changes the relative importance between reference and current frames, selects between available reference frames, can be different at different levels

3.3. UMCTF with update step disabled

In this section we describe one embodiment of the UMCTF obtained by disabling the update step in the lifting process. As mentioned previously, the use of long temporal filters associated with multiple reference frames, bi-directional filtering, etc. leads to added decoding delay, and also may create visual artifacts in L frames. In order to overcome these problems, we may disable the update step, thereby leave the A frames unfiltered. This corresponds to using a delta low-pass filter, i.e. by setting $g_{i,j}^d = \delta(i - jM_{d-1})$. In principle, this is equivalent to temporal sub-sampling of the sequence with different sub-sampling factors at different temporal levels. The

sub-sampling factor corresponds to M_d (the decimation coefficient at level $d \in [0, D]$). For d>0, we have: $A_i^d = A_{M_d i}^{d-1},\ i \in \{0, ..., N_d - 1\}$.

The temporally high-pass filtered frames H_i^d at level $d \in [0, D]$ are obtained by motion-compensated filtering as follows:

$$H_i^d(n,m) = A_i^{d-1}(n,m) - \sum_{j \in S_i^d} f_{i,j}^d(n,m) A_{i-j}^{d-1}\left(n - v_{y,i-j\to i}^d, m - v_{x,i-j\to i}^d\right)$$

where:

- $i \in \left\{[1, N_{d-1} - 1] : \frac{i}{N_{d-1}} \notin \mathbb{Z}\right\}$, i.e. we skip frames with indices multiple of M_d, which are the A frames;
- j corresponds to the index of the temporal filter tap, $S_i^d = \{[i - N_{d-1} + 1, i] \cap j \neq 0\}$ is the support of the temporal filter kernel;
- $f_{i,j}^d$ is the j^{th} coefficient[5] of the temporal high-pass filter to create H_i^d frames;
- $A_{i-j}^{d-1}(n - v_{y,i-j\to i}^d, m - v_{x,i-j\to i}^d)$ represents the motion-compensated $(i\text{-}j)^{\text{th}}$ frame; this may possibly include a spatial interpolation, in case of a fractional-pel motion estimation.

We denote by R_p^d (respectively R_f^d) the maximum number of reference frames allowed from the past (respectively, from the future).

3.3.1 Multiple reference frames and bi-directional filtering

By changing S_i^d and the values of the filter taps $f_{i,j}^d$ we may perform filtering across multiple reference frames, bi-directional filtering etc. Note that depending on the number of used reference frames $R^d = R_p^d + R_f^d$, we need to send a maximum of $R^d - 1$ additional sets of motion vectors as compared with the conventional MCTF case. However, if certain filter coefficients $f_{i,j}^d$ are equal to 0, the corresponding motion vectors do not need to be sent. The tradeoff between the improved prediction and the bits required for sending additional motion vectors may be exploited depending on the sequence characteristics, optimal bit rate versus quality etc. More details on some of these tradeoffs may be obtained from [25].

In order to avoid increasing the complexity and the delay, R_f^d should be kept small. Also, when multiple reference frames from the future are used, we have to select frames with indices

multiples of M_d, (the L frames at the next level) in order to maintain causality. Since these frames are decoded before the current decomposition level, using them as references does not increase the decoding delay.

3.3.2 Variable Decomposition Structures: Number of successive H frames

While with Haar MCTF, only dyadic (powers-of-two e.g. half, quarter) frame-rate scalability can be achieved, with UMCTF any fraction of the full frame-rate can be simply obtained, by varying M_d, the number of H-frames between successive A/L-frames (temporal subsampling factor). For instance, to achieve a sixth of the full-frame rate, we can use $N = 6$, $D = 2$ with $M_0 = 2$, and $M_1 = 3$. We show an example decomposition with these different options and $R_p^0 = 3$, $R_f^0 = 1$, $R_p^1 = 2$, in Figure 4.

[5]As the coefficients are defined here, they are constrained to sum to 1. However, this does not preclude scaling all the filter coefficients (and correspondingly the factor 1 applied to the current frame) to account for adaptive quantization, motion boundary filtering, impact factors [32] etc.

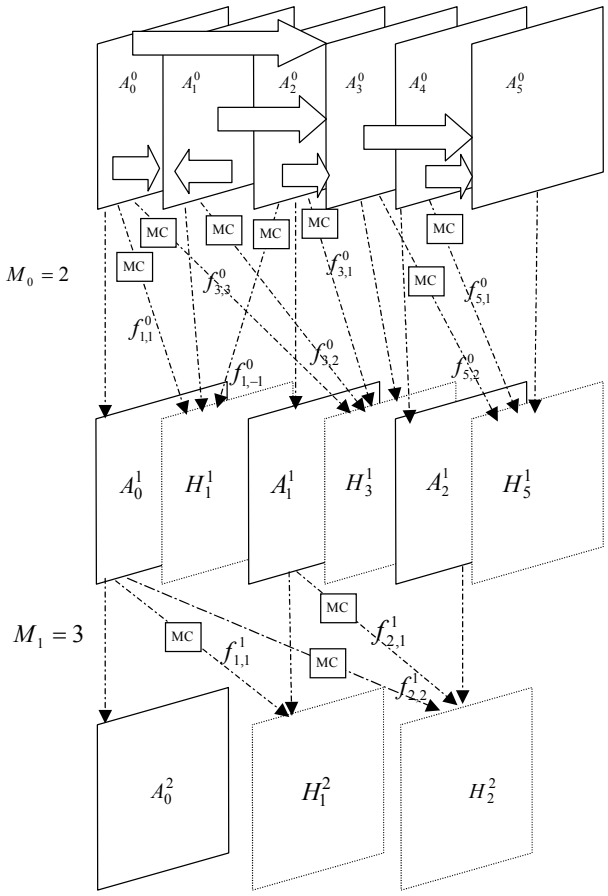


Figure 4. Pyramidal Temporal Decomposition Scheme

As may be seen from the figure, some frames use bi-directional prediction, while others use different numbers of reference frames. This variation can occur *within* and *between* the multiple temporal decomposition levels.

Although UMCTF without the update step may be predictive in its nature, significant differences exist from predictive (closed-loop) coding. Specifically, in UMCTF we retain the multiresolution decomposition structure in order to exploit both long term as well as short-term temporal dependencies. Also, we use an open-loop prediction structure such that SNR scalability does not suffer from drift. Finally, we can adaptively change the number of reference frames, the relative importance attached to each reference frame, the bi-directional filtering etc.

3.4. Mixture of Low-pass filters

When motion matches are poor, temporal low-pass filtering creates visual artifacts; however when motion matches are good it is useful to actually capture the temporal average, instead of merely using sub-sampled versions of the sequence. In such cases, L frames should be created using non-delta low-pass filters g_i^d. During a multi-resolution temporal decomposition, frames

get farther apart at higher levels, thereby increasing the likelihood of finding poor motion matches. Hence temporal low-pass filtering should be disabled (delta low pass filters should be used) at higher temporal levels. UMCTF allows for the adaptive selection of a mixture of low-pass filters (non-delta and delta) corresponding to the quality of the match. Due to the lifting implementation, the low pass filters may be chosen without being constrained by the high-pass filter choice $f_{i,j}^{d}$. Different low-pass filters can be used for different frames within a temporal level, or even for different blocks within a frame; notice though that the choice of filters needs to be transmitted to the decoder, and this overhead should be minimized.

4. Distortion Fluctuation Control

A major problem in MCTF-based video coding is the control of the actual distortion resulting from decoding to different temporal and/or SNR (quality) settings. The cause of such fluctuations in the decoded frames is the fact that inverse motion compensation during the update step (or complete lack of update) changes the decomposition basis (locally or globally) into a non-orthonormal basis. As a result, the large distortion variations, particularly visible in low-rate coding, limit the applicability of such systems in multimedia applications targeting agreed/contracted video quality needs. A mechanism to control such distortion fluctuations has been recently proposed in [24]. We present the description of the control mechanism for UMCTF with update step disabled, using bi-directional filtering ($R_p^d = 1,\ R_f^d = 1$) with the immediately preceding and future frames used as reference frames. For each block in the frame, we allow for three choices for the temporal filter coefficients $f_{i,j}^{d}$:

Backward filtering: $f_{i,1}^{d} = 1$ and $f_{i,-1}^{d} = 0$

Forward filtering: $f_{i,1}^{d} = 0$ and $f_{i,-1}^{d} = 1$

Bi-directional filtering: $f_{i,1}^{d} = f_{i,-1}^{d} = 0.5$

At each temporal level, the estimated distortion (mean square error) for each block in the reconstructed frame A_i^{d-1} is given by:

$$E\left[e_{A_i^{d-1}}^2\right] = E\left[\left(e_{H_i^d} + f_{i,1}^{d} e_{A_{i-1}^{d-1}} + f_{i,-1}^{d} e_{A_{i+1}^{d-1}}\right)^2\right] \tag{1}$$

where $e_{A_i^{d-1}}$ is the random variable defining the reconstruction error within each block for the frame A_i^{d-1}. By assuming no correlation between the reconstruction errors for the three frames H_i^d, A_{i-1}^{d-1} and A_{i+1}^{d-1} we may rewrite (1) as

$$E\left[e^2_{A_i^{d-1}}\right] = E\left[e^2_{H_i^d}\right] + \left(f^d_{i,1}\right)^2 E\left[e^2_{A_{i-1}^{d-1}}\right] + \left(f^d_{i,-1}\right)^2 E\left[e^2_{A_{i+1}^{d-1}}\right] \quad (2)$$

Denote by p_p and p_f the percentages of pixels in the frame A_i^{d-1} that are linked during the ME process only with the past (A_{i-1}^{d-1}) and only with the future (A_{i+1}^{d-1}) frames respectively. Assuming that no intra-coding is used, the percentage of pixels that are bi-directionally predicted from the previous and next frames is $\left(1-p_p-p_f\right)$. Writing equation (2) for each block in A_i^{d-1}, accounting for the type of temporal prediction (uni-directional or bi-directional) employed per block, and averaging over the entire frame yields the estimated distortion in frame A_i^{d-1}:

$$E\left[e^2_{A_i^{d-1}}\right] = E\left[e^2_{H_i^d}\right] + \frac{\left(1+3p_p-p_f\right)}{4} E\left[e^2_{A_{i-1}^{d-1}}\right] + \frac{\left(1+3p_f-p_p\right)}{4} E\left[e^2_{A_{i+1}^{d-1}}\right] \quad (3)$$

In the proposed distortion-fluctuation control mechanism, the adaptive selection of temporal filters performed for each block at each temporal level is captured by the p_p and p_f parameters. These parameters are determined and stored at the coding stage and do not need to be signaled to the decoder; they are needed only in the parsing stage (at the transmission time) in order to determine for each temporal frame the optimum truncation points corresponding to the user requirements in terms of temporal and/or SNR (quality) settings.

To limit distortion variation in the decoded sequence, we need to ensure that the error in the current frame A_i^{d-1} is not significantly larger than the error in its reference frames, A_{i-1}^{d-1} and A_{i+1}^{d-1}. We may constrain the error using a control parameter *a*, by requiring:

$$E\left[e^2_{A_i^{d-1}}\right] = (1+a)\max\left(E\left[e^2_{A_{i-1}^{d-1}}\right], E\left[e^2_{A_{i+1}^{d-1}}\right]\right) \quad (4)$$

Large values of *a* indicate a high increase of the distortion in the frame A_i^{d-1}, and hence a larger distortion fluctuation in the decoded output is expected. For small values of *a*, the distortion behavior is expected to be quasi-constant. Replacing equation (4) in equation (3) we obtain:

Case 1: $E\left[e^2_{A_i^{d-1}}\right] = (1+a)E\left[e^2_{A_{i-1}^{d-1}}\right]$:

$$E\left[e^2_{H_i^d}\right] = \left(0.25p_f - 0.75p_p + a + 0.75\right)E\left[e^2_{A_{i-1}^{d-1}}\right] + \left(0.25p_p - 0.75p_f - 0.25\right)E\left[e^2_{A_{i+1}^{d-1}}\right] \quad (5)$$

Case 2: $E\left[e^2_{A_i^{d-1}}\right] = (1+a)E\left[e^2_{A_{i+1}^{d-1}}\right]$:

$$E\left[e^2_{H_i^d}\right] = \left(0.25p_p - 0.75p_f + a + 0.75\right)E\left[e^2_{A_{i+1}^{d-1}}\right] + \left(0.25p_f - 0.75p_p - 0.25\right)E\left[e^2_{A_{i-1}^{d-1}}\right] \quad (6)$$

Using the above expressions we can establish a control-mechanism for the distortion variation, as outlined in the algorithm in Figure 5.

During encoding:

1. Establish q rate-distortion points for each frame in the MCTF of the GOF. For each frame, keep also the percentage of the frame that was predicted from the previous and the next reference frame in the current temporal level (p_p , p_f) respectively.

During the parsing stage:

1. For each GOF, establish the maximum number of temporal levels *D*. Establish the value of *a*.
2. For frame A_0^{D-1} (the remaining A-frame of the MCTF of the GOF), read the set of q distortion points $\left\{E\left\lfloor e^2_{A_0^{D-1}}\right\rfloor\right\}_{z=1,2...q}$ and associated rates $\left\{R\left(A_0^{D-1}\right)\right\}_{z=1,2...q}$, which were produced during the encoding of the GOF.
3. For each of the *q* distortion values $\left\{E\left\lfloor e^2_{A_0^{D-1}}\right\rfloor\right\}_{z=1,2...q}$, and for all levels *d*, use equations (5) and (6) to establish the corresponding set of theoretical distortion points $E\left\lfloor e^2_{H_i^d}\right\rfloor_z$ for the H_i^d frames in the GOF.
4. For each of the *q* distortion values $E\left\lfloor e^2_{H_i^d}\right\rfloor_z$, identify for every frame H_i^d the truncation point $\bar{z}$ for which the distortion $E\left\lfloor e^2_{H_i^d}\right\rfloor_{\bar{z}}$ is the closest to theoretical distortion $E\left\lfloor e^2_{H_i^d}\right\rfloor_z$ calculated at the previous step. Keep this truncation point and the associated rate $\left\{R\left(H_i^D\right)\right\}_{\bar{z}}$.
5. If rate-control is desired, scan the produced set of *q* distortion-rate points of each GOF, and match the average rate constraint (Kbps).

Figure 5. Distortion control algorithm

We may also extend this distortion control algorithm to obtain a scheme for rate control, as described in Step 5 of Figure 5. The proposed control mechanism is applicable to the *parser* of the embedded bit-stream, as long as an embedded scheme is used to encode the MCTF frames, and a set of potential truncation points (R-D points) is generated for each frame during encoding.

We present indicative results on distortion control in Section 6.5. For illustration, we use the EBCOT [26] entropy-coding scheme to encode the data, as it allows for the easy establishment of R-D points during encoding.

The above described distortion-fluctuation control mechanism is presented for the case of UMCTF with update-step disabled employing an adaptive selection of temporal filters for each block and for each temporal level. Moreover, although the proposed control mechanism works in a frame-by-frame basis, it can also be applied locally to certain frame areas, if the expected distortions between dependent frames are linked in a block-by-block manner. This however comes with the expense of increased complexity at the parsing stage. Finally, the generic formulation for the UMCTF case with enabled update-step follows also similar derivations (the distortion estimates $E\left[e^2_{A_i^d}\right]$ are derived from the update step, similar to the derivation of $E\left[e^2_{H_i^d}\right]$ in the predict step). This generic formulation is left as a topic of further investigation.

5. Analysis of Delay

The motivation behind the use of UMCTF without an update step comes from the fact that this mode corresponds to the lowest delay scenario. This is because the L frames at a certain temporal level may be decoded independently of the H frames at that level. In this section, we quantify the delay for this case. In our analysis, we consider an *N* frame GOF with *D* decomposition levels, and for the sake of simplicity, we set $M_d = M$. We use multiple reference frames from the past, and one reference frame from the future, i.e. $R_p^d = N_{d-1} - 1$ and $R_f^d = 1$ for all levels *d*. We assume $N \bmod (M)^D = 0$.

The decoding delay for any frame corresponds to the number of frames that need to be decoded before the frame can be displayed. Since UMCTF involves a multi-resolution decomposition, we need to examine the delay for frames filtered at different temporal levels. Within temporal decomposition level d , H frames are located at positions $k_{d,i}$ such that $(k_{d,i} \bmod M) \neq 0$ and $i = 1, 2, \ldots, N\left(\frac{M-1}{M^{d+1}}\right)$. This corresponds to a position $k_{d,i} M^{d-1}$ in the GOF. As an example, with $N = 27$, $M = 3$ and $D = 3$, H frames at level $d = 2$ are at positions 1, 2, 4, 5, 7, 8 within the level, and at positions 3, 6, 12, 15, 21 and 24 in the GOF.

Since we allow multiple reference frames from the past, each of these frames can use a maximum of $k_{d,i}$ references from the past. In addition, since we allow bi-directional filtering with $R_f^d = 1$, we also need to consider one reference frame from the future[6]. Hence, the frame $k_{d,i}(M)^{d-1}$ depends directly on a maximum of $(k_{d,i} + 1)$ frames at level *d*.

However, we also need to consider the indirect reference frames, i.e. frames at higher levels that these $(k_{d,i} + 1)$ frames are dependent on. All indirect reference frames from the past are already included in the count $(k_{d,i} + 1)$, and there is only one additional indirect reference frame from the future at each higher decomposition level (*D*–1–*d* such levels). Hence, the maximum number of frames that need to be decoded for frame $k_{d,i}(M)^d$ to be displayed are $(k_{d,i} + 1 + D - 1 - d)$. Hence, the maximum delay incurred by a frame is $(k_{d,i} + D - d) - k_{d,i}(M)^d$, i.e. its position subtracted from the number of frames that need to be decoded before it can be displayed.

[6] We can ignore the case when we cannot use bi-directional filtering due to absence of future frames. This is because the delay for these frames is always smaller than for frames that are filtered bi-directionally.

In order to quantify this delay, we consider UMCTF with $N=16$, $D=4$, $M_d=2$. For this example, except for Frame 1 and Frame 3, all other frames have negative delay. This means that they can always be decoded before they need to be displayed. Frame 1 and Frame 3 experience a delay of 1 frame each. Alternatively, Haar MCTF schemes with the same structure experience a much larger maximum delay of $\frac{N}{2}+1$, since the temporal filtering is performed in pairs.

6. Results and Discussion

We present results for UMCTF with different filter choices and decomposition structures, to highlight its advantages in terms of efficiency and scalability features. We first present results on coding efficiency, followed by results on content adaptive update, variable decomposition structures, and the temporal scalability features of UMCTF. Additionally, the experimental results obtained with the distortion control mechanism are presented in Section 6.5. Finally, we also include a comparison against AVC.

6.1. Results on Coding Efficiency

In this section, we present improvements in coding efficiency for UMCTF with the addition of different coding optimization tools. We start with the Haar MCTF and add the following tools:

- Variable Size Block Motion (VSBM). Given a macroblock size of $B\times B$, we support 7 different partitioning schemes where each sub-partition is of size $\frac{B}{2^i}\times\frac{B}{2^j}$ with $(i,j)\in\{(0,0),(1,0),(0,1),(1,1),(2,1),(1,2),(2,2)\}$.
- Overlapped Block Motion Compensation (OBMC). We use a weighted average of pixels, obtained using neighboring motion vectors, as predictors. This is similar to what is proposed in [31] to improve the coding efficiency and smooth out block boundaries.
- Bi-directional prediction. We disable the update step (delta low-pass filter) and use one reference frame from the past and one from the future during the predict step. We label this *No Update Bidirectional UMCTF*.
- Multi-hypothesis prediction. We further allow two reference frames from the past. We set $\sum_j f_{i,j}^d(n,m)=1$, where either two coefficients are non-zero in the summation and they are equal to ½ (bi-directional filtering, or multi-hypothesis prediction from the past), or only one is non-zero and it is equal to 1 (forward/backward filtering). We label this *No Update Multi Hypothesis UMCTF*.

The best motion vector, prediction direction and partition size are determined using an exhaustive Lagrangian R-D optimized mode decision strategy, i.e. by minimizing $SAD + \lambda(ModeBits + MVBits)$ across all modes and partition sizes[7]. The parameter λ cannot be optimally determined since we use the same mode decision for all target decoding rates. However, we have determined heuristically that a value of 25 performs reasonably well across the range of desired bit-rates.

We present results with UMCTF settings: $N = 16$, $D = 4$, $M_d = 2$, using a dyadic decomposition structure. We present results for three CIF sequences, Foreman, Mobile and Football with a search range of ±8, half-pixel motion accuracy and a macroblock size *B*=32. We use the spatial 9/7 wavelet decomposition followed by EZBC for entropy coding [16]. Unless otherwise stated, these settings were used for all results in this section.

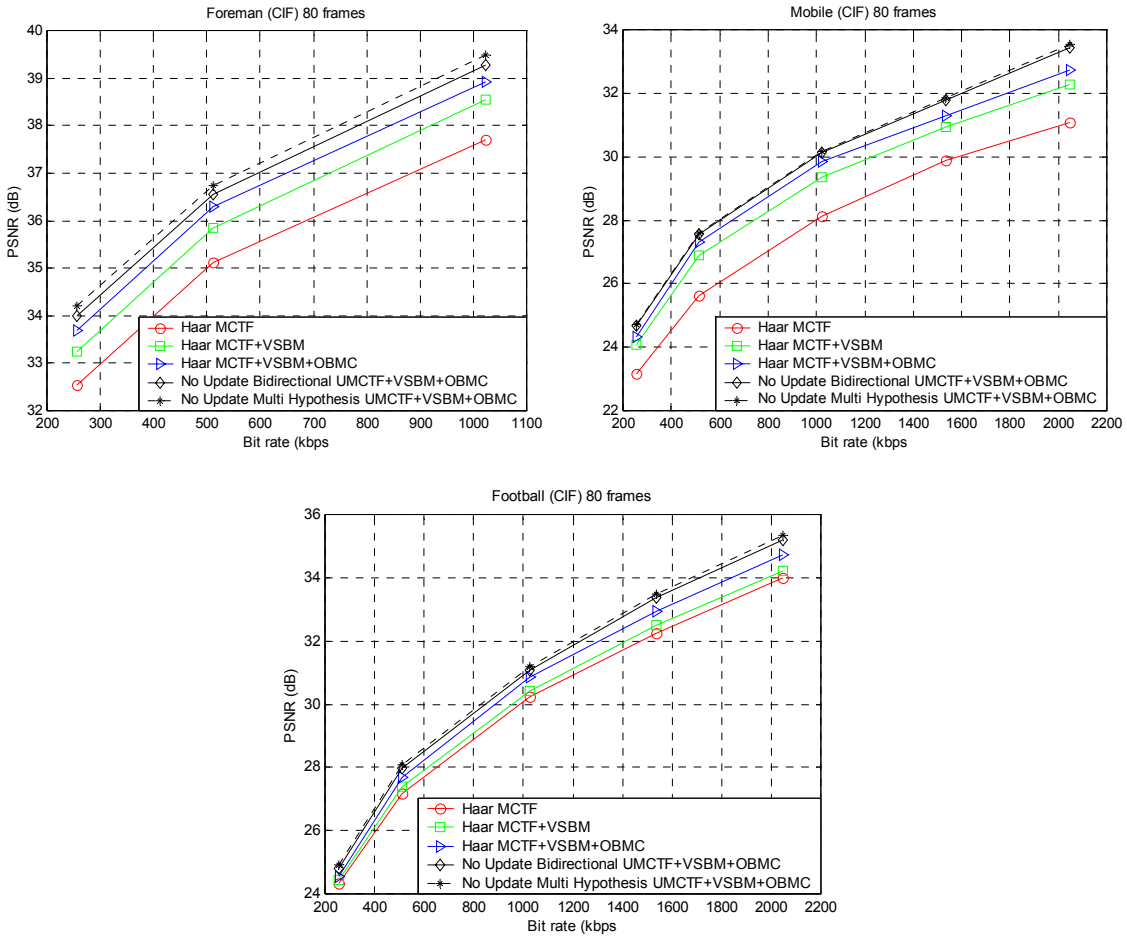


Figure 6. Incremental improvements in coding efficiency

[7] We also allow for intra blocks that are spatially predicted as in [31]. Also, if the number of intra blocks exceeds a threshold (25% of all blocks), we designate the frame as an Intra frame and disable all prediction.

We observe from Figure 6 that each of these tools leads to a systematically improved R-D performance, although the improvements vary depending on the sequence content.

6.2. Adaptive Update

While disabling the update step allows us great flexibility in terms of prediction options, it leads to significant distortion variations in the decoded sequence. This is because of the use of a non-orthonormal temporal transform, which requires successive decoding, thereby leading to quantization noise propagation from the reference frame to the reconstructed frame. Hence, we also need to use an appropriate low-pass filter, using the update step of the lifting process. When the motion matches are good, then such an update can lead to an improvement in the coding performance, however when the motion matches are not good, this can lead to the creation of artifacts in the low-pass frame leading to worse coding performance. More details on this can be obtained from [28]. Hence, the update should be performed adaptively, depending on the quality of the motion match. To illustrate this we implement an adaptive update strategy as follows:

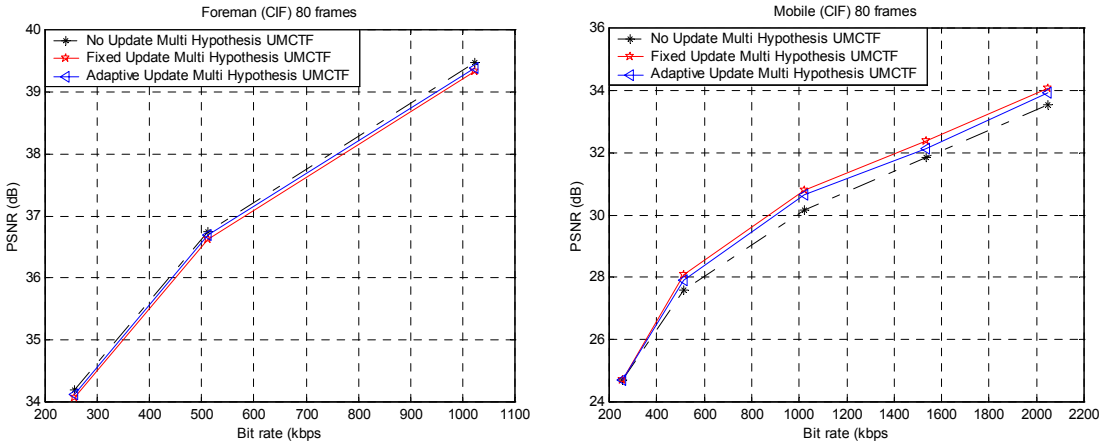
If $\left|H_i^d(m,n)\right| \le T_{update}$

Perform update using $H_i^d(m,n)$

Else

Perform update using $sign\left(H_i^d(m,n)\right)\times T_{update}$

This strategy reduces the artifacts in the L frame due to poor motion matches (by limiting the magnitude of the updating pixel), while allowing update for good matches. In addition, as this can be deduced at the decoder, there is no overhead involved. We present results for UMCTF with multi-hypothesis prediction: with no update, with update always enabled, and with adaptive update in Figure 7.



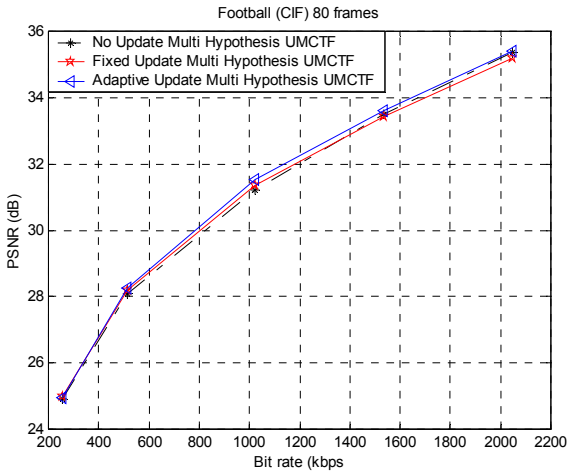


Figure 7. Effect of adaptive update strategy

Since the Mobile sequence has correlated motion and smaller prediction errors, performing the update is beneficial, and leads to improved coding performance. On the other hand, the performance for the Foreman sequence degrades when the update step is enabled, while it remains almost unaffected for the Football sequence. With the simple adaptive update strategy we can achieve performance close to that of the best scheme (update always enabled or disabled) for both the Foreman and Mobile sequences, while for the Football sequence, we can actually outperform both strategies. This adaptive strategy can further be extended to account for the connectivity of pixels.

6.3. Variable temporal decomposition structures

In this section we present results to illustrate the performance of a non-dyadic temporal decomposition structure. We compare the dyadic *No Update Multi Hypothesis UMCTF* results (as in the previous section) with a non-dyadic decomposition with the following structure:

- $N = 27$, $D = 3$, $M_d = 3$ (two H frames between L frames). Again, we allow multi-hypothesis, forward, backward or bi-directional filtering. We label this the *No Update Multi Hypothesis Non-Dyadic* UMCTF.

Using this decomposition structure we encode 81 frames, and compare the average PSNR for the first 80 frames with previously presented results. We plot this comparison in the following figures.

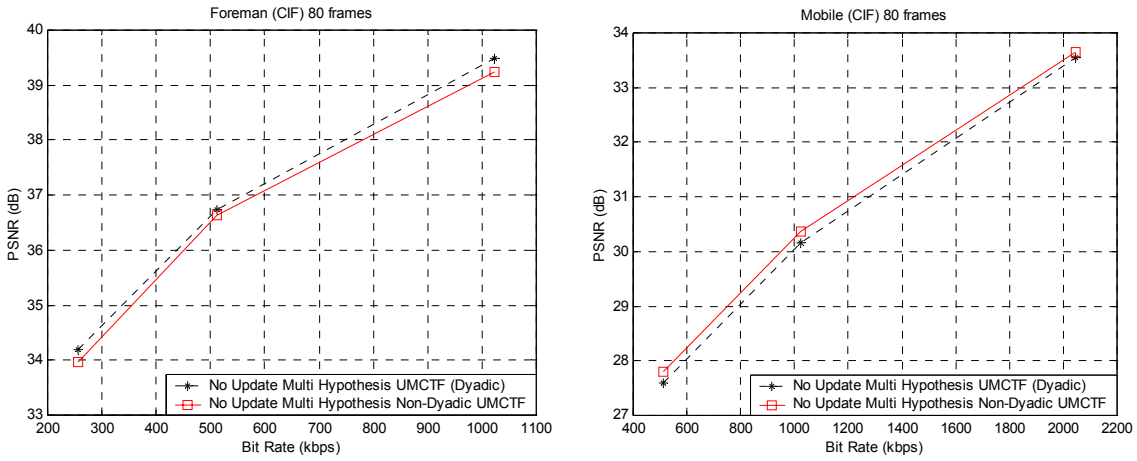


Figure 8. Comparison of dyadic and non-dyadic decomposition structures

For the Foreman sequence, the non-dyadic decomposition performs worse than the dyadic decomposition, while this trend is reversed for the Mobile sequence. With the non-dyadic decomposition the distance between source and reference frames is larger than with the dyadic decomposition, and this can lead to worse temporal prediction. However, at the same time, as the GOF size (N) is larger, a greater amount of temporal correlation can be exploited. It is the tradeoff between these two conflicting reasons that determines the performance for a particular sequence. Since the motion can be easily captured for the Mobile sequence, even using reference frames farther apart, the non-dyadic decomposition outperforms the dyadic decomposition.

6.4. Results on Temporal Scalability

In order to evaluate the temporal scalability performance, we disable the update step, so that we can use the sub-sampled original sequence as reference and evaluate objective metrics like PSNR. We present results for the sequences Foreman and Mobile with *No Update Multi Hypothesis UMCTF* settings with dyadic decomposition, as in Section 6.1. The sequences are decoded at three different frame rates, 30 Hz, 15 Hz and 7.5 Hz, corresponding to decoding all, half or quarter the number of frames.

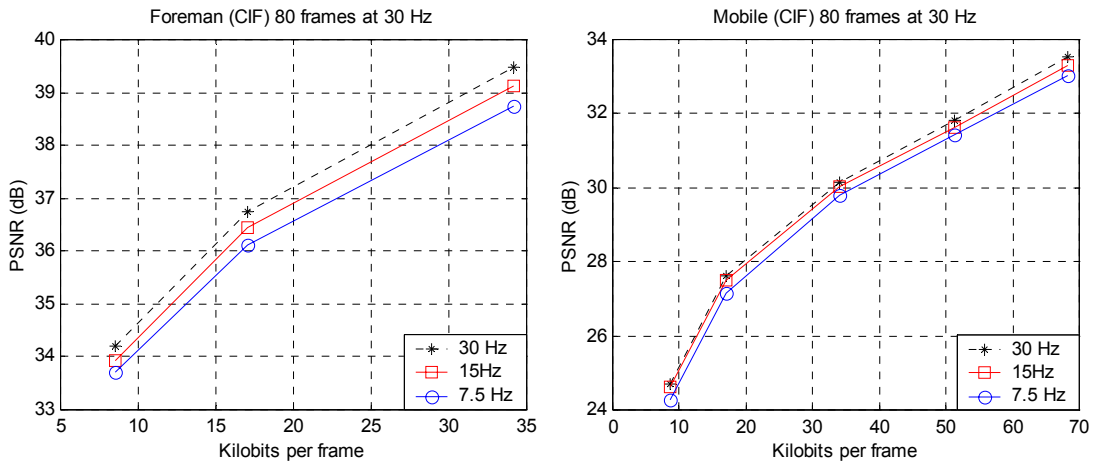


Figure 9. PSNR for Foreman and Stefan decoded at multiple frame rates (30Hz, 15Hz and 7.5 Hz)

We compare results at different frame rates against each other, at bit rates corresponding to an equivalent average number of bits per frame. Hence, PSNR at 300 kbps for 30Hz video may be compared against PSNR at 150 kbps for 15Hz video, corresponding to 10 Kilobits per frame. Comparing these results we can conclude that the efficiency of UMCTF is not significantly degraded while providing temporal scalability. The results are expectedly worse at lower frame rates as a greater amount of temporal correlation can be exploited at higher frame rates.

It is important to notice that, as described in Section 2, the visual quality of the L frames can be poor and they may contain artifacts, when motion matches are poor. Consequently, we believe that for a true evaluation of the temporal scalability performance, the update step should be disabled, and the temporally subsampled video should be used as reference. Independent experiments [30] support this conclusion.

6.5. Distortion control results

Disabling the update step leads to improved temporal scalability performance, however, as mentioned before, the use of a non-orthonormal temporal decomposition leads to distortion variation in the decoded frames. In this section we present results for UMCTF with distortion control. We use the JPEG-2000 entropy coder in conjunction with UMCTF to generate these results. For UMCTF we use the settings corresponding to our derivation for the distortion control scheme outlined in Section 4. These correspond to the *No Update Bidirectional UMCTF* described in Section 6.1. For motion estimation, we use full search with 8×8 blocks, with quarter-pixel accuracy and a search range of $[-64,64]$.

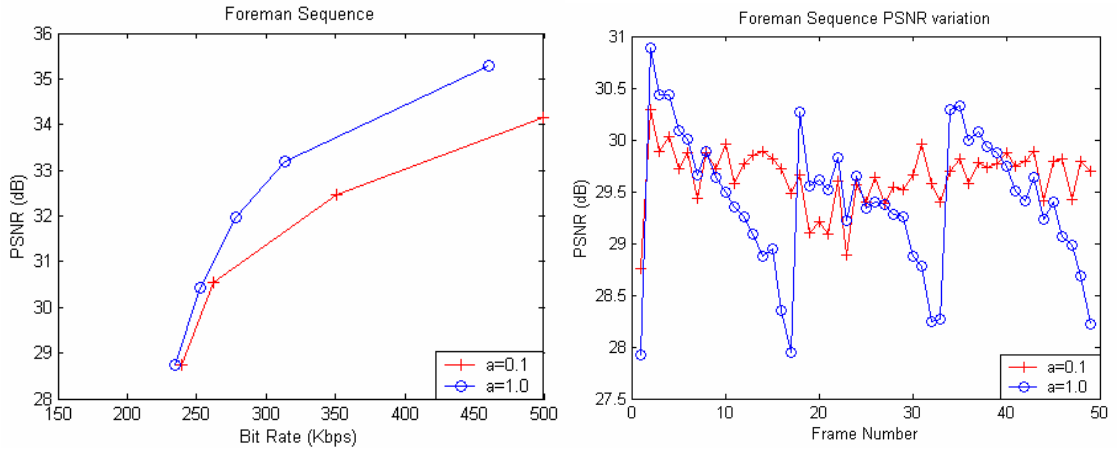


Figure 10. Distortion control results for Foreman

Although the control parameter a can be specified separately per GOF, we chose two cases, $a = 0.1$ and $a = 1.0$. The resulting average PSNR is shown on the left in Figure 10, while the PSNR variation is shown on the right. The results indicate that PSNR fluctuations in the sequence can be reduced by appropriately selecting the parameter a, although this comes at the expense of an average PSNR loss. However, it is observed that at low rates, the loss in the average PSNR is limited. Since PSNR variations are mainly visible in low-rate decoding, we conclude that the proposed control-mechanism is particularly beneficial at low bit-rates.

6.6. Comparison against AVC

We compare our codec against AVC under the following test conditions. We use a dyadic decomposition structure with multi-hypothesis prediction, adaptive update, VSBM and OBMC for UMCTF. We use a motion search range of ±32 with quarter pixel accuracy, and a macroblock size B=32. For the Mobile sequence we use $N = 32$, $D = 5$, $M_d = 2$, while for the Football sequence we use $N = 8$, $D = 3$, $M_d = 2$. We use the same GOP size (with IBBP structure) and motion search range for AVC. We use JM 6.3 with R-D optimization enabled and CABAC for all our experiments. These results are presented in Figure 11.

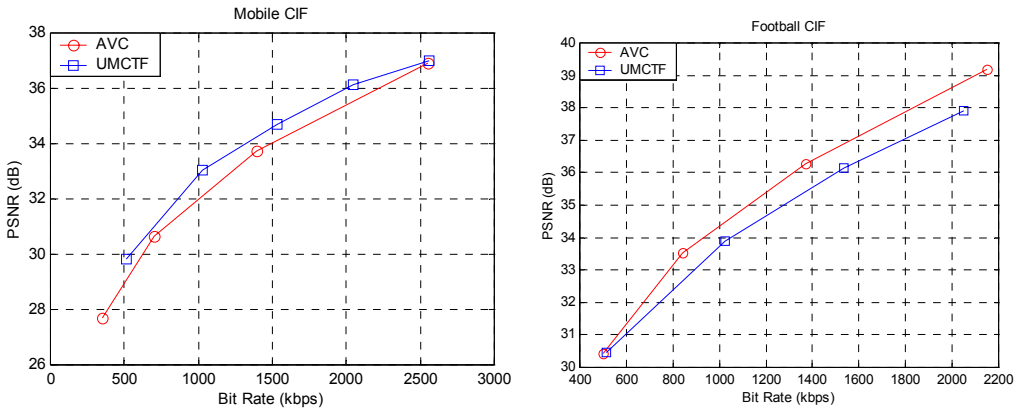


Figure 11. Comparison against AVC

While UMCTF outperforms AVC for the Mobile sequence, it underperforms AVC for the Football sequence. This is because the motion in the Football sequence is very large and random, leading to large residues, and many intra blocks that AVC can code very efficiently due to its use of a closed loop and extremely efficient intra prediction.

7. Conclusions

In this paper, we present UMCTF, a framework for efficient and flexible temporal filtering for interframe wavelet video coding. UMCTF relies on the lifting implementation of temporal filtering and uses a set of control parameters to select decomposition structures and temporal filters, thereby enabling the adaptation of the video codec to the video content, bandwidth constraints and end-device capabilities. This leads to improved coding efficiency and higher decoded video quality over current interframe wavelet video codecs. Furthermore, the temporal scalability provided by UMCTF is also significantly improved, by eliminating the constraint of pair-wise filtering, as in Haar MCTF. UMCTF supports decoding of the video at arbitrary desired fractions of the full frame rate, unlike Haar MCTF schemes, where only dyadic (powers-of-two) temporal scalability is supported. Also, the frames decoded at lower frame-rates are free of visual artifacts, unlike with Haar MCTF schemes. In addition, we propose a mechanism for the control of the distortion variation in UMCTF employing only the predict step, which is particularly beneficial in low-rate coding. Our experimental results show that the coding efficiency of UMCTF (in conjunction with EZBC entropy coding) is comparable to the state-of-the-art non-scalable codec, AVC/H.264.

8. Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments that significantly improved the quality of the paper. The work of Y. Andreopoulos, A. Munteanu and P. Schelkens

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 by the European Community under the IST Program (Mascot, IST-2000-26467). P. Schelkens also has a post-doctoral fellowship with the Fund for Scientific Research -Flanders (FWO), Egmontstraat 5, B-1000 Brussels, Belgium.

9. References

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra , "Overview of the H.264/AVC video coding standard", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560-576, July 2003.

[2] T. Wiegand and B. Girod, "Multi-Frame Motion-Compensated Prediction for Video Transmission", Kluwer Academic Publishers, Sep. 2001.

[3] J. Ohm, W. Li, *et al*, "Summary of Discussions on Advanced Scalable Video Coding", Contribution to MPEG - M7016, March 2001.

[4] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming over IP", *IEEE Trans. Multimedia*, vol. 3, no. 1, March 2001.

[5] H. Gharavi, "Subband Coding of Video Signals" in Subband Image Coding, Chap.6, Kluwer Academic Publishers, 1991-Edited by J.W. Woods.

[6] Y.-Q. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color video compression", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no.3, pp. 285-96, Sept. 1992.

[7] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video", *IEEE Trans. Image Processing,* vol. 3, pp. 572–588, Sept. 1994.

[8] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", *IEEE Trans. Circuits Syst. Video Technol.,* vol. 6, pp. 243–250, June 1996.

[9] B.-J Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D Set partitioning in Hierarchical Trees (3-D SPIHT)", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1374-1387, December 2000.

[10] J. R. Ohm and T. Ebrahimi, "Report of Ad hoc Group on Exploration of Interframe Wavelet Technology in Video", End Contribution of MPEG meeting, Jeju, Korea, March 2002.

[11] D. Blasiak and W.-Y. Chan, "Efficient wavelet coding of motion compensated prediction residuals", Proc. ICIP 1998, vol. 2, pp. 287 –291, 1998.

[12] E. Asbun, P. Salama and E.J. Delp, "A rate-distortion approach to wavelet-based encoding of predictive error frames", Proc. ICIP 2000, Vol.3, pp. 154-157, Vancouver, Canada, Sept. 10-13, 2000.

[13] Y. Andreopoulos, A. Munteanu, *et al*, "Complete-to-overcomplete discrete wavelet transforms: theory and applications," IEEE Transactions on Signal Processing, to be published.

[14] J. R. Ohm, "Three-dimensional subband coding with motion compensation", *IEEE Trans. Image Proc.*, vol. 3, no. 5, pp. 559-571, September 1994.

[15] S.-J. Choi and J. W. Woods, "Motion compensated 3-D subband coding of video", *IEEE Trans. Image Proc.*, vol. 8, no. 2, pp. 155-167, February 1999.

[16] S.-T. Hsiang and J. W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank", *Signal Processing: Image Communication* vol. 16, pp. 705-724, May 2001.

[17] J. Xu, Z. Xiong, *et al*, "Three-Dimensional Embedded Subband Coding with Optimized Truncation (3-D ESCOT)", *Applied and Computational Harmonic Analysis*, vol. 10, pp. 290–315, May 2001.

[18] J. W. Woods, S.-C. Han, *et al*, "Spatiotemporal subband/wavelet video compression", in Handbook of Image and Video Processing (A. Bovik, ed.), Academic Press, May 2000.

[19] D. Turaga and M. van der Schaar, "Unconstrained motion compensated temporal filtering", Contribution to MPEG - M8388, May 2002.

[20] D. Turaga and M. van der Schaar, "Wavelet coding for video streaming using new unconstrained motion compensated temporal filtering," *Proc. International Workshop on Digital Communications: Advanced Methods for Multimedia Signal Processing*, IWDC 2002, Capri, Italy, pp. 41-48, Sept. 2002.

[21] M. van der Schaar and D. S. Turaga, "Unconstrained motion compensated temporal filtering framework for wavelet video coding," Proc. ICASSP 2003, May 2003.

[22] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression", Proc. IEEE ICASSP 2001, Salt Lake City, UT, 7-11 May 2001.

[23] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting", Proc. ICIP 2001, Thessaloniki, Oct. 2001.

[24] A. Munteanu, Y. Andreopoulos, *et al*, "Control of the distortion variation in video coding systems based on motion compensated temporal filtering," Proc. ICIP 2003, Barcelona, September 2003.

[25] D. Turaga, M. van der Schaar and B. Pesquet-Popescu, "Reduced complexity spatio-temporal scalable motion compensated wavelet video encoding," Proc. ICME 2003, Baltimore, July 2003.

[26] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Proc*., vol. 9, no. 7, July 2000.

[27] K. Hanke, "RD performance of fully scalable MC-EZBC," Contribution to MPEG, M9000, October 2002.

[28] D. S. Turaga and M. van der Schaar, "Content-adaptive filtering in the UMCTF framework," Proc. ICASSP 2003, May 2003.

[29] N. Mehrseresht and D. Taubman, "Adaptively Weighted Update Steps in Motion Compensated Lifting Based Scalable Video Compression", Proc. ICIP 2003, Barcelona, September 2003.

[30] S. Tsai, H.-M. Hang and T Chiang, "Exploration experiments on the temporal scalability of interframe wavelet coding", Contribution to MPEG, M 8959, October 2002.

[31] J. Woods, Y. Wu and R. Cohen, "An overlapped block motion estimation for EZBC", Contribution to MPEG, M10158, October 2003.

[32] T. Rusert and K. Hanke, "Optimized quantization in interframe wavelet coding", Contribution to MPEG, M9003, October 2002.

[33] Y. Andreopoulos, A. Munteanu, *et al*, "In-band motion compensated temporal filtering," to be published in Signal Processing: Image Communication, Special Issue on Interframe Wavelet Coding, 2003.