Motion Compensation and Scalability in Lifting-Based Video Coding

Grégoire Pau [a], Christophe Tillier [a], Béatrice Pesquet-Popescu [a], Henk Heijmans [b]

[a] *ENST, Signal and Image Proc. Dept., 46, rue Barrault, 75634 Paris, France* {*gpau,tillier,pesquet*}*@tsi.enst.fr*

[b] *CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands henk.heijmans@cwi.nl*

Abstract

Motion-compensated temporal filtering subband video codecs have attracted recently a lot of attention, due to their compression performance comparable with that of state-of-the-art hybrid codecs and due to their additional scalability features. In this paper we present a scalable video codec based on a 5/3 adaptive temporal lifting decomposition. Different adaptation criteria for coping with the occluded areas are discussed and new criteria for optimizing the temporal prediction are introduced. For our simulations, we use a memory-constraint "on-the-fly" implementation. We also evaluate the temporal scalability properties of this video coding structure.

Key words: scalable video coding, temporal lifting, temporal scalability, prediction optimization, 5/3 motion compensated temporal filtering

1 Introduction

With the expansion of multimedia applications and the need for delivering compressed bitstreams over heterogeneous networks, scalability has become an important feature for video coders. Recently, a new generation of video compression techniques has attracted the attention of the research engineering community, by proposing state-of-the-art compression performance in addition to temporal/spatial/SNR scalability features. They exploit interframe redundancy through a temporal wavelet transform in the motion direction. Wavelet transforms have proved their efficiency for still image compression, and their

implementations exploiting the lifting scheme provided at the same time faster algorithms, in-place computation and parallelism. Moreover, lifting schemes enable the design of nonlinear [1], [2] or adaptive [3] wavelet transforms.

Three-dimensional subband decompositions not involving motion or with simplified motion models have been proposed quite early in the video coding literature [4,5]. Later, the spatio-temporal (or "2D+t") subband coding (or 3D-SBC) schemes [6], [7] relying on the idea that a subband decomposition applied along the temporal axis of a video sequence leads to an efficient energy concentration on low-pass temporal subbands quickly became an alternative approach to hybrid coding concepts used in today's video standards. Previous works exploiting the motion-compensated Haar transform [6,8] provided very promising results. This paper focuses on lifting schemes that can be used for the temporal coding part in a motion-compensated temporal-filtering subband coder (MCTF-SBC) for video. Lifting formulations of temporal motion-compensated wavelet decompositions have been proposed in [9] and [10]. The latter work highlights the fact that the integration of the motion estimation/compensation (ME/MC) in the temporal decomposition leads to non-linear predict and update operators. This framework, that can apply to any temporal lifting scheme with one lifting step (prediction plus update), allowed to propose several improvements by modifying the update operator. For example, it has been shown that an overlapped motion compensation corresponds to a specific non-linear update filter. Several improvements to this lifting scheme have also been proposed recently, concerning for example the spatial filtering of occluded areas [11,12] or the motion estimation accuracy [13]. Unconstrained prediction using multiple reference frames is another interesting technique that has been proposed in [14], but in this case the scheme does not involve an update step.

An important issue concerning temporal multiresolution analysis (MRA) is the choice of the temporal filter length : long filters take better advantage of the temporal correlation existing between successive frames but have an increased motion overhead, are more complex and have higher latency than MC Haar filters. The 5/3 temporal transform is a good compromise whose benefits have been presented and justified in [9,15,16]. In this paper, following the approach we presented in [16], we propose the use of motion-compensated 5/3 filters, and compare it with classical Haar MRA. Note that a mesh-based MCTF for video coding was proposed in [17]. Our main contribution in this paper is to introduce two new criteria for optimizing the predict step, and to show the coding performance improvement achieved with this strategy.

The paper is organised as follows: in the next section we review the principles of non-linear lifting in the considered spatio-temporal context. In Section 3, the 5/3 MCTF is introduced, while in Section 4 an adaptive update operator able to take into account different optimization criteria is presented. The op-

timization of the predict operator is discussed in Section 5, where the motion vector redundancy is also taken into account. Temporal scalability features are discussed in Section 6. Implementation specifics and some simulation results with the resulting 5/3 non-linear transforms are given in Section 7 and some conclusions are drawn in Section 8.

2 Temporal Filtering and the Lifting Scheme: General Framework

First, we introduce some notations: the frames in the sequence will be denoted by $(x_t(\boldsymbol{n}))$, where t is the temporal index and $\boldsymbol{n}$ is a spatial variable that takes values in $S = \{1, \ldots, M\} \times \{1, \ldots, N\}$. In the wavelet decomposition, we will denote by $h_{t,j}$ the detail (temporal "high-frequency") subband frames and by $l_{t,j}$ the approximation (temporal "low-frequency") subband frames at temporal resolution level $j \in \mathbb{N}$. Below we will only describe one transform level, but it is clear that one can obtain a multiresolution decomposition by subsequent decomposition of the approximation band. Due to this fact, we will drop the index j in what follows.

If we consider a filter bank whose lifting implementation [18] corresponds to a single lifting step, then the approximation and detail coefficients are obtained from the following equations (we make abstraction for the moment of the final multiplicative constants):

$$h_t = x_{2t+1} - \mathcal{P}\Big\{(x_{2t})_{t\in\mathbb{N}}\Big\} \tag{1}$$

$$l_t = x_{2t} + \mathcal{U}\Big\{(h_t)_{t\in\mathbb{N}}\Big\}, \tag{2}$$

where $\mathcal{P}$ and $\mathcal{U}$ are the predict and update operators respectively and where $(x_{2t})_{t\in\mathbb{N}}$ and $(h_t)_{t\in\mathbb{N}}$ are the set of all even samples and the set of all detail coefficients respectively.

The lifting formalism guarantees the invertibility of the scheme, and therefore the original samples can be simply retrieved using the same lifting steps in reverse order with negated signs:

$$x_{2t} = l_t - \mathcal{U}\Big\{(h_t)_{t\in\mathbb{N}}\Big\} \tag{3}$$

$$x_{2t+1} = h_t + \mathcal{P}\Big\{(x_{2t})_{t\in\mathbb{N}}\Big\} \tag{4}$$

This formalism can also be applied to implement a temporal MRA, by considering the frames of the video sequence as temporal samples. In this case, it is more effective to perform the temporal analysis *in the motion direction* and we have highlighted [10] that the motion-compensated lifting involves in a very natural way *non-linear* operators, related to the motion compensation.

The predict and update operators then also involve the motion vectors used to match corresponding positions. Moreover, in the 2D+t framework they actually become *spatio-temporal* operators:

$$h_t(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \mathcal{P}\Big[\{x_{2(t-k)}, \boldsymbol{v}_{2t+1}^{2(t-k)}\}_{k\in T_k^p}\Big], \quad \forall \boldsymbol{n} \in S \tag{5}$$

$$l_t(\boldsymbol{p}) = x_{2t}(\boldsymbol{p}) + \mathcal{U}\Big[\{h_{t-k}, \boldsymbol{v}_{2t}^{2(t-k)+1}\}_{k\in T_k^u}\Big], \quad \forall \boldsymbol{p} \in S, \tag{6}$$

where $\boldsymbol{v}_i^j$ is the motion vector field used to predict the current frame i from the reference frame j and T_k^p (resp. T_k^u) is the support of the temporal predict (resp. update) operator. The spatio-temporal aspect is related to the fact that all the pixels from even frames can be used to predict each detail coefficient $h_t(\boldsymbol{n})$, and similarly the detail frames, once available, can be employed entirely by the update operator. We can see the ME operation as a pre-decision, influencing both predict and update operators. The motion vectors need to be transmitted as side-information. Fig. 1 illustrates these operations.

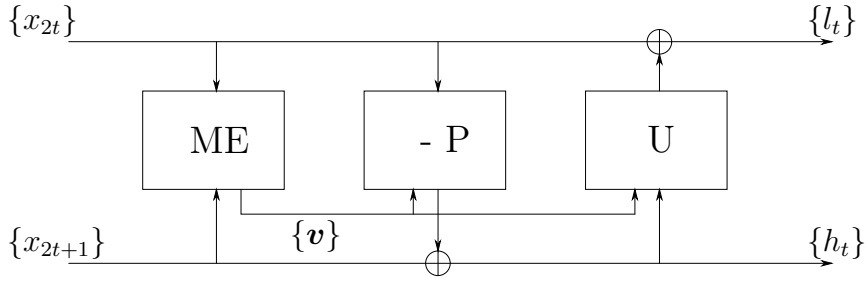


Fig. 1. Lifting representation of the motion compensated temporal filtering.

Even though $\boldsymbol{v}_i^j$ and $\boldsymbol{v}_j^i$ are not identical MV fields, in practical schemes very often only one of them will be estimated, due both to complexity and bitrate restrictions. The basic idea is to use the fact that for homogeneous motion, we should have $\boldsymbol{v}_i^j = \boldsymbol{v}_j^i$. However, when this assumption is not satisfied, this simplification will require a more intricate study, for example by looking for unconnected or multiply connected pixels [6,7], as will be described in the sequel or by exploiting the similarity of the two motion vector fields during the estimation and coding stages [17,19].

Let us give some examples for the above non-linear lifting framework. The simplest case corresponds to an integer pel ME and a linear time-invariant filter for $\mathcal{P}$, involving uniquely $x_{2(t-k)}(\boldsymbol{n}), k \in T_k^p$ and the corresponding vectors predicting the pixel $\boldsymbol{n}$, $\boldsymbol{v}_{2t+1}^{2(t-k)}(\boldsymbol{n}), k \in T_k^p$. In this case, the $\mathcal{P}$ operator is uniquely a temporal motion compensated filtering along the motion trajectory.

A different situation is the fractional pel ME, when the temporal prediction is followed by the *spatial* interpolation of the reference frames. In this case, $\mathcal{P}$ corresponds to a spatio-temporal operator.

Another example of predict operator applying in the spatio-temporal domain is

obtained by introducing an *overlapped motion compensation* within the temporal filtering algorithm, so as to reduce blocking artefacts. This operation involves in the predict step an average of pixels from adjacent windows in the reference frame. To illustrate this idea, let us consider the example of an Haar MRA and an overlap of one pixel, as in [10]. In this case, the high-pass filtering of pixels belonging to the first (respectively the last) row of a block reads:

$$h_t(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \left((1-\beta)x_{2t}[\boldsymbol{n} - \boldsymbol{v}_{2t+1}^{2t}(\boldsymbol{n})] + \beta x_{2t}[\boldsymbol{n} - \boldsymbol{e} - \boldsymbol{v}_{2t+1}^{2t}(\boldsymbol{n}-\boldsymbol{e})]\right),$$

where $\boldsymbol{e} = (0,1)$ (resp. $\boldsymbol{e} = (0,-1)$) and β is a constant, $0 < \beta < 1$. A similar processing can be applied to the first (resp. last) column of each block, by choosing $\boldsymbol{e} = (1,0)$ (resp. $\boldsymbol{e} = (-1,0)$). Obviously, larger overlaps between adjacent blocks could be realized in the same way, but in practice, our simulations show that almost no improvement is achieved with larger overlaps.

The update operator $\mathcal{U}$ should also be ideally applied on the motion trajectory. So, even though the structural property of the lifting scheme is that all the information available from the predict step (high-frequency frames) can be used for the update, the most useful information will be in the neighborhood of the pixels on the same trajectory. This feature can be exploited in different ways to improve the temporal filtering in critical areas like, for example, the occlusion zones. For example, a selection criterion among the possible candidates for update lifting step among multiple connections is the one minimizing the energy of the detail coefficients [10], [20]. The intuition behind this criterion is that the correct match between moving areas leads to a minimum prediction error (detail frame) energy. The above criterion can be made more robust by considering a mean energy around the considered pixel, instead of a single pixel value.

Another example of exploiting the spatial information in the update operator is to apply a smooth transition filtering [12,11] to connected pixels within a transition range around an unconnected pixel, in order to avoid an abrupt change in processing neighboring connected and unconnected pixels.

In the next two sections we consider in more detail the design of the 5/3 temporal transform by exploiting these concepts.

3 5/3 Motion-Compensated Temporal Filtering

We consider a simple 5/3 biorthogonal transform applied along the temporal axis, where both the prediction and the update operator have only two taps. In this case, the detail and approximation are respectively (if we skip the spatial

indices):

$$h_t = x_{2t+1} - (\alpha x_{2t} + \beta x_{2t+2}) \tag{7}$$
$$l_t = x_{2t} + \gamma h_{t-1} + \delta h_t \tag{8}$$

For video processing, the temporal filtering can be performed as described above, except that motion compensated (MC) frames have to be used as samples. Special attention is required if fractional-pel ME is performed and the prediction and update operators involve not only ME/MC but also interpolation. As in our approach, filtering is performed on MC frames, by predicting x_{2t+1} from x_{2t} and x_{2t+2}, it is obvious that, for all $t \in \{0, \ldots, T-1\}$, we have to deal with two motion vector fields (MVF): a forward MVF from x_{2t} to x_{2t+1} and a backward one from x_{2t+2} to x_{2t+1}. We will also try to keep the same conventions as in the Haar filtering case i.e. the h_t frame will be computed at the same positions as x_{2t+1}, while the approximation will be computed at the corresponding positions in the previous frame, x_{2t}. We introduce the following notations for the motion vectors (MV): the forward MV, predicting the position $\boldsymbol{n}$ in frame x_{2t+1} from the frame x_{2t} will be denoted by $\boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$, while the backward MV, predicting the same position in frame x_{2t+1}, but from frame x_{2t+2} will be $\boldsymbol{v}^-_{2t+1}(\boldsymbol{n})$ (see Fig. 2). With these notations, Eqs. (7) and (8) can be rewritten as the following temporal filtering in the motion direction:

$$h_t(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \Big[\alpha\, x_{2t}\Big(\boldsymbol{n} - \boldsymbol{v}^+_{2t+1}(\boldsymbol{n})\Big) + \beta\, x_{2t+2}\Big(\boldsymbol{n} - \boldsymbol{v}^-_{2t+1}(\boldsymbol{n})\Big)\Big] \tag{9}$$
$$l_t(\boldsymbol{p}) = x_{2t}(\boldsymbol{p}) + \gamma h_{t-1}\Big(\boldsymbol{p} + \boldsymbol{v}^-_{2t-1}(\boldsymbol{m})\Big) + \delta h_t\Big(\boldsymbol{p} + \boldsymbol{v}^+_{2t+1}(\boldsymbol{n})\Big) \tag{10}$$

In the last equation, the positions $\boldsymbol{n}$ and $\boldsymbol{m}$ have to be determined and this is a tricky point which will be addressed in Subsection 4.1.

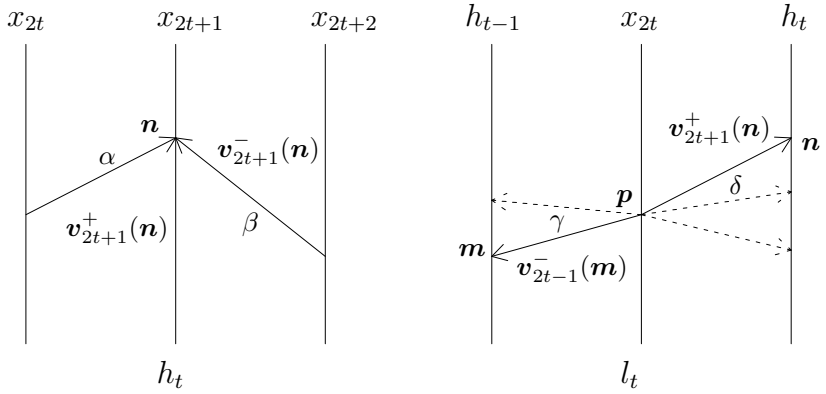


Fig. 2. Predict (left) and update (right) operators in the spatio-temporal domain. Straight arrows indicate the motion compensation directions. Dashed arrows indicate the multiple connections to the pixel $\boldsymbol{p}$.

Remark: Eq. (9) makes sense only if the frame x_{2t} can be bidirectionally predicted. When there is a scene cut, one of the predictions will be meaningless

and has not to be used. This amounts to set α or β to 0. This requires detecting the scene cuts, and possibly segmentating the video sequence into homogeneous Groups of Frames (GOF) (or spatial parts of a GOF). This means that an adaptive prediction can be realized in this manner. Note that this kind of adaptive behaviour is not yet implemented in the current version of our temporal lifting method, as it entails a more complex coding strategy. In particular, side information would probably need to be transmitted although it would be small compared to the amount of motion and texture information. In the present approach, we do not consider a special processing of the scene cuts.

4 An Adaptive Update Operator

4.1 General Update Strategy

Positions corresponding to the same motion trajectory in successive frames are illustrated in Fig. 3. At this point, we observe that the computation of l_t is symmetric with respect to the time $2t$. The problem is to find the positions $\boldsymbol{m}$ and $\boldsymbol{n}$ which are associated to $\boldsymbol{p}$. This is equivalent to solving the implicit equations $\boldsymbol{p} = \boldsymbol{m} - \boldsymbol{v}^-_{2t-1}(\boldsymbol{m}) = \boldsymbol{n} - \boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$. Actually, these equations may have several solutions or no solution at all for a given $\boldsymbol{p}$! The interpretation of this fact is strongly related to the existence of unconnected and multiple connected pixels in the case of Haar temporal lifting (see [10]). The present case is different in the sense that there exist unconnected/multiply connected pixels coming both from forward and backward predictions. Referring to Fig. 3, we denote by $\mathrm{P}^- = \{\boldsymbol{m} \mid \boldsymbol{m} - \boldsymbol{v}^-_{2t-1}(\boldsymbol{m}) = \boldsymbol{p}\}$ the set of all pixels connected to $\boldsymbol{p}$ in the previous frame and by $\mathrm{P}^+ = \{\boldsymbol{n} \mid \boldsymbol{n} - \boldsymbol{v}^+_{2t+1}(\boldsymbol{n}) = \boldsymbol{p}\}$ the set of all pixels connected to $\boldsymbol{p}$ in the next frame. A pixel $\boldsymbol{p}$ is said to be backward (resp. forward) unconnected if $\mathrm{P}^- = \emptyset$ (resp. $\mathrm{P}^+ = \emptyset$), backward (resp. forward) connected if $\mathrm{P}^- \neq \emptyset$ (resp. $\mathrm{P}^+ \neq \emptyset$) and multiple backward (resp. forward) connected if card $\mathrm{P}^- > 1$ (resp. card $\mathrm{P}^+ > 1$).

We distinguish four subclasses of pixels $\boldsymbol{p}$ in the frame at time $2t$:

(1) Forward and backward connected pixels (corresponding to $\mathrm{P}^+ \neq \emptyset$ and $\mathrm{P}^- \neq \emptyset$). Here, the existence of both $\boldsymbol{n}$ and $\boldsymbol{m}$ is guaranteed. Consequently, we have $\gamma \neq 0$ and $\delta \neq 0$ in Eq. (10). In simple cases, e.g. without motion or with uniform motion, we have $\boldsymbol{v}^-_{2t-1}(\boldsymbol{m}) = -\boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$ and the position $\boldsymbol{m}$ can be directly deduced from $\boldsymbol{n}$, namely $\boldsymbol{m} = \boldsymbol{n} - 2\boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$. This particular case can apply to large parts in a video sequence, for example, a (fixed) background or objects exhibiting uniform motion over several frames.

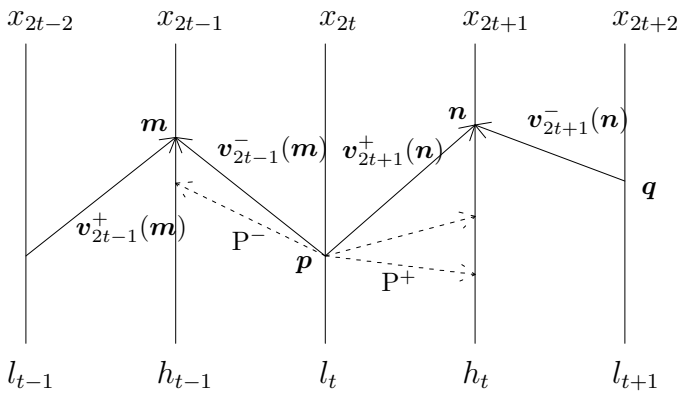


Fig. 3. The positions corresponding to the same motion trajectory in successive frames for temporal lifting scheme with 5/3 biorthogonal filters. Arrows correspond to the motion compensation direction.

(2) Forward connected and backward unconnected pixels ($\mathrm{P}^+ \neq \emptyset$ and $\mathrm{P}^- = \emptyset$). Here we choose $\gamma = 0$. This situation arises for instance if $\boldsymbol{p}$ is a pixel in a part of an object that is covered until time $2t-1$ and becomes visible from time $2t$ onwards.
(3) Forward unconnected and backward connected pixels ($\mathrm{P}^+ = \emptyset$ and $\mathrm{P}^- \neq \emptyset$). This situation is similar to the previous one by interchanging the roles of $\boldsymbol{n}$ and $\boldsymbol{m}$. Here we put $\delta = 0$.
(4) Forward and backward unconnected pixels ($\mathrm{P}^+ = \emptyset$ and $\mathrm{P}^+ = \emptyset$). In this case we take: $\gamma = \delta = 0$.

Consider now case 3 (but case 2 can be treated analogously) i.e. $\mathrm{P}^+ = \emptyset$ and $\mathrm{P}^- \neq \emptyset$. We have :

$$l_t(\boldsymbol{p}) = x_{2t}(\boldsymbol{p}) + \gamma h_{t-1}\Big(\boldsymbol{p} + \boldsymbol{v}^-_{2t-1}(\boldsymbol{m})\Big) \tag{11}$$

and we obtain a situation similar to the one described for Haar MCTF in [10]. Two subcases are possible: either there exists an unique $\boldsymbol{m} \in \mathrm{P}^-$ ($\boldsymbol{p}$ is simply connected with the previous frame) or P^- has more than one element ($\boldsymbol{p}$ is multiply connected with the previous frame). In the former case, by simply applying Eq. (11) we obtain the filtered value. In the latter case, the "best" choice can be found using one of the criteria introduced for one-directional prediction in [10] and described in the next Subsection 4.2.

The importance of these subclasses of pixels is illustrated in Fig. 4, where we denote by $\mathrm{N}^- = \mathrm{card}\ \mathrm{P}^-$ the cardinal of the set P^- and by $\mathrm{N}^+ = \mathrm{card}\ \mathrm{P}^+$ the cardinal of the set P^+. We can remark that the most frequent case is the forward and backward simply connected one, especially in lower temporal levels. This is mainly due to the uniformity of the motion at these levels. One can also remark that the percentage of unconnected and multiple connected pixels increases with the temporal decomposition level, requiring thus a special attention in processing. This increase of unconnected pixels is mainly due to the non-uniformity of the motion at higher temporal levels, and is

undesirable from a compression point of view. Indeed, connected and unconnected pixels are low-pass filtered using different filters and this may lead to abrupt changes between connected and unconnected zones, therefore reducing the spatial coding efficiency of approximation frames. Moreover, when further temporally decomposing these frames, this inhomogeneity gives rise to large wavelet coefficients, reducing the coding efficiency of the detail subbands.

$N^-\backslash N^+$	0	1	>1
0	0.25	2.55	0.18
1	2.36	89.36	2.35
>1	0.17	2.54	0.18

First temporal level

$N^-\backslash N^+$	0	1	>1
0	0.97	5.76	0.88
1	4.43	76.63	4.45
>1	0.82	5.47	0.55

Second temporal level

$N^-\backslash N^+$	0	1	>1
0	4.16	14.30	3.14
1	5.28	50.71	5.70
>1	3.03	11.39	2.25

Third temporal level

$N^-\backslash N^+$	0	1	>1
0	9.83	15.80	5.75
1	8.06	34.14	6.88
>1	3.89	11.15	4.46

Fourth temporal level

Fig. 4. Percentages of unconnected ($N = 0$), simply connected ($N = 1$) and multiply connected ($N > 1$) pixels in the forward (N^+) and backward direction (N^-) at several temporal decomposition levels. These results were obtained on a four level decomposition of the video sequence "Foreman" CIF 30 fps on the frames 16-32.

4.2 Multiple Connections Update Strategy

The most difficult case concerns forward and backward multiple connected pixels. We have to make a choice between several connected pixels and some interesting criteria are the following:

(1) Independent optimization of forward and backward criteria deduced from the Haar case, e.g.
$(\boldsymbol{m}, \boldsymbol{n}) = \Big(\arg\min_{\boldsymbol{m}} E_{t-1}(\boldsymbol{m}) + \lambda^- \|\boldsymbol{v}^-_{2t-1}\|, \arg\min_{\boldsymbol{n}} E_t(\boldsymbol{n}) + \lambda^+ \|\boldsymbol{v}^+_{2t+1}\|\Big)$,
where λ^-, λ^+ are non-negative constants and $E_t(\boldsymbol{n})$ (resp. $E_{t-1}(\boldsymbol{m})$) is the mean energy of the detail subband in the forward (resp. backward) direction around the considered pixel. They can be written as

$$E_t(\boldsymbol{n}) = \sum_{\boldsymbol{k} \in S(\boldsymbol{n})} \left(h_t(\boldsymbol{n} - \boldsymbol{k}) u(\boldsymbol{k})\right)^2, \quad E_{t-1}(\boldsymbol{m}) = \sum_{\boldsymbol{k} \in S(\boldsymbol{m})} \left(h_{t-1}(\boldsymbol{m} - \boldsymbol{k}) u(\boldsymbol{k})\right)^2,$$

where $S(\boldsymbol{n})$ is a neighborhood around the pixel $\boldsymbol{n}$ and $u(\boldsymbol{k})$ corresponds to a weighting factor for each pixel in this neighborhood, depending on its distance to the central point.

(2) If we aim at obtaining uniform motion, then ideally we will obtain: $\boldsymbol{v}^-_{2t-1}(\boldsymbol{m}) = -\boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$. This suggests the following criterion: $(\boldsymbol{m}, \boldsymbol{n}) =$

$\arg \min_{(\boldsymbol{m},\boldsymbol{n})} \|\boldsymbol{v}^-_{2t-1}(\boldsymbol{m}) + \boldsymbol{v}^+_{2t+1}(\boldsymbol{n})\|$.

(3) If only the continuity of the motion trajectories is imposed, then ideally the relation betwen successive motion vector fields will be: $\boldsymbol{v}^-_{2t-1}(\boldsymbol{m}) = -c\boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$, $c > 0$. This can be achieved by minimizing the following expression:
$(\boldsymbol{m},\boldsymbol{n}) = \arg \min_{(\boldsymbol{m},\boldsymbol{n})} \left| \left\langle \frac{\boldsymbol{v}^-_{2t-1}(\boldsymbol{m})}{\|\boldsymbol{v}^-_{2t-1}(\boldsymbol{m})\|}, \frac{\boldsymbol{v}^+_{2t+1}(\boldsymbol{n})}{\|\boldsymbol{v}^+_{2t+1}(\boldsymbol{n})\|} \right\rangle + 1 \right|$, with the convention that $\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} = \boldsymbol{0}$ if $\boldsymbol{v} = \boldsymbol{0}$.

4.3 Choice of the Parameters

Now we have to determine the values of the parameters α, β, γ and δ in Eqs. (9), (10) in the different cases described in Subsection 4.1. Moreover, for maximum generality, one might use $\tilde{h}_t, \tilde{l}_t$ given by $\tilde{h}_t(\boldsymbol{n}) = k_h \cdot h_t(\boldsymbol{n})$ and $\tilde{l}_t(\boldsymbol{n}) = k_l \cdot l_t(\boldsymbol{n})$, where the parameters k_h and k_l have to be determined for optimal coding efficiency. Of course, they can be set to $k_h = k_l = 1$ and then the quantizers in each subband can be adjusted accordingly. This solution however may lead to a more complex coding strategy, as different look-up tables have to be managed simultaneously.

Strictly speaking, we have prediction and update filters that vary both in temporal and in spatial directions, making a rigorous analysis quite delicate. To simplify matters, we consider a still region of the video sequence (i.e. with zero motion vectors), but where intensity changes in time may occur, due to illumination variations, noise, etc. Such areas can represent large parts of the scene, like background or still objects. In such a motion-less region, we have time-invariant filters at any given position, which allows the use of standard tools such as the z-transform.

In order to choose the α and β parameters, we have to take into account the high-pass characteristic of the prediction filter (with output h_t). Considering a (temporally) constant input signal, the resulting details have to be 0, so from Eq. (7) we obtain $\alpha + \beta = 1$. Moreover, if there is no reason to give preference to prediction based on the past (x_{2t}) over prediction from the future (x_{2t+2}), we will choose $\alpha = \beta = 1/2$. Note that this analysis is independent of k_h. This parameter may be determined from a "quasi-orthonormality condition": $\left|\tilde{H}(-1)\right| = \sqrt{2}$, where $\tilde{H}(z)$ is the transfer function associated with $\tilde{h}_t$. This condition is satisfied for $k_h = 1/\sqrt{2}$.
In order to determine γ and δ, we use the expected low-pass behaviour of the update filter, meaning that for the transfer function $L(z)$ associated with l_t, we have $L(z) = -\alpha\gamma z^{-2} + \gamma z^{-1} + (1 - \beta\gamma - \alpha\delta) + \delta z - \beta\delta z^2$, with $L(-1) = 0$. This yields $\gamma + \delta = 1/2$ and if we assume, as before, that there is no reason to

distinguish between forward and backward filtering, then $\gamma = \delta = 1/4$. The constant k_l can be obtained by setting, as in the case of orthonormal filters, $\tilde{L}(1) = \sqrt{2}$, which leads to $k_l = \sqrt{2}$. Note that the set of values for $\alpha, \beta, \gamma, \delta$ obtained this way corresponds to the classical biorthogonal c_{22} transform [21].

What is interesting in this approach for the determination of the parameters is that it also applies when one of cases 2, 3 or 4 in Subsection 4.1. arises ($\mathrm{P}^- = \emptyset$ or $\mathrm{P}^+ = \emptyset$). For example, if $\mathrm{P}^- \neq \emptyset$ and $\mathrm{P}^+ = \emptyset$, hence $\delta = 0$, we have $\tilde{l}_t = k_l(x_{2t} + \gamma h_{t-1})$, and by a similar reasoning, we get $\gamma = 1/2$ and $k_l = \sqrt{2}$.
Finally, if both $\gamma = 0$ and $\delta = 0$, we obtain again $k_l = \sqrt{2}$.

Note that a more elaborate strategy for computing the multiplicative constants has been recently proposed in [11].

5 Temporal Prediction and Motion Estimation

Compared to a Haar MRA (equivalent, from this point of view, with a hybrid video coding scheme), the number of motion vectors in the 5/3 scheme is multiplied by two. This means that on the one hand the motion estimation problem is more involved, and on the other hand more attention should be paid to the encoding of these motion vectors. Especially at low and medium bitrates, a large part of the bit budget would be otherwise used for motion vector coding, instead of texture coding. Different trade-offs in bit allocation are possible between motion vector accuracy and the quality of the temporal prediction. Moreover, the hierachical dependencies between motion vector fields at different temporal levels can be taken into account in top-down or bottom-up strategies of motion estimation and motion coding [22], [23].

We adopt here a new approach in optimizing the predict operator: since the aim of this step is to achieve the best temporal prediction, two possibilities can be envisaged. Indeed, we can either optimize the structure of the filter, for example by considering long-term prediction or an adaptive technique for intra-inter prediction [14], or consider the motion estimation problem and optimize the choice of the vectors involved in this prediction. In this latter case, optimizing could mean minimizing the distortion of the temporal detail frames h_t, in order to facilitate their spatial coding. This leads to a two-parameter minimization problem, where the following criterion should be optimized:

$$(\widehat{\boldsymbol{v}}^+_{opt}, \widehat{\boldsymbol{v}}^-_{opt}) = \arg \min_{\substack{\boldsymbol{v}^+ \in W^+ \\ \boldsymbol{v}^- \in W^-}} \sum_{\boldsymbol{n} \in \mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n} - \boldsymbol{v}^+) + x_{2t+2}(\boldsymbol{n} - \boldsymbol{v}^-)}{2}\Big], \tag{12}$$

where d is any usual distortion measure (quadratic, absolute error, ...), W^+

(resp. W^-) is the forward (resp. backward) search window in x_{2t} (resp. x_{2t+2}) and we simplified the notation for forward and backward motion vectors: $\boldsymbol{v}^+ = \boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$, $\boldsymbol{v}^- = \boldsymbol{v}^-_{2t+1}(\boldsymbol{n})$. $\mathcal{B}$ is the block of pixels in the current frame x_{2t+1} which has to be predicted. Ideally, a full search in the two searching windows should be performed in order to find the optimal solution. The complexity of such an operation is much too high, which led us to look for sub-optimal solutions.

Note that till now only a separate search for the two motion vectors has been used. We propose an iterative joint search algorithm of the two motion vectors, approximating the optimal solution. The initialization can be done with one of the motion vectors found by a classical estimation (minimizing a two-frame criterion).

- **Initialization:** $\widehat{\boldsymbol{v}}^+_0 = \arg \min_{\boldsymbol{v}^+ \in W^+} \sum_{\boldsymbol{n} \in \mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - x_{2t}(\boldsymbol{n} - \boldsymbol{v}^+)\Big]$
- **Iteration** i**, for** $i \geq 1$**:**

$$\widehat{\boldsymbol{v}}^-_i = \arg \min_{\boldsymbol{v}^- \in W^-} \sum_{\boldsymbol{n} \in \mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n} - \widehat{\boldsymbol{v}}^+_{i-1}) + x_{2t+2}(\boldsymbol{n} - \boldsymbol{v}^-)}{2}\Big]$$

$$\widehat{\boldsymbol{v}}^+_i = \arg \min_{\boldsymbol{v}^+ \in W^+} \sum_{\boldsymbol{n} \in \mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n} - \boldsymbol{v}^+) + x_{2t+2}(\boldsymbol{n} - \widehat{\boldsymbol{v}}^-_i)}{2}\Big]$$

At each iteration (in fact, at each motion estimation corresponding to half of one iteration), the algorithm allows the proposed criterion to be decreased, and ensures a better solution (in the sense of Eq. (12)) than carrying out two separate backward/forward motion estimations.

It is worth nothing that the method does not rely on a fixed size of the block $\mathcal{B}$ and can be combined with an adaptive block size approach for motion estimation [6]. Moreover, the block partitioning of the MVFs for successive estimation steps does not need to be the same.

The maximum number of iterations has to be limited due to complexity reasons. In this rough form, each iteration adds the complexity of two searching operations. However, it is possible to reduce the searching window at each iteration, based on the fact that the motion vector refinement does not need the same search domain as the previous estimation. We show in Section 7 that even only one iteration leads to substantial improvement in coding performances.

A simplified version of this criterion consists in estimating not two, but a *unique* motion vector. This idea could be supported by a uniform motion assumption between three consecutive frames, in the same way we have proceeded for optimizing the update operator in Section 3. This amounts to considering forward and backward motion vectors of identical amplitudes and

directions, but with opposite signs, and minimizing the criterion:

$$J(\boldsymbol{v}) = \sum_{\boldsymbol{n}\in\mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n}-\boldsymbol{v}) + x_{2t+2}(\boldsymbol{n}+\boldsymbol{v})}{2}\Big] \tag{13}$$

The optimization of this criterion can be realized similarly to a one-variable motion estimation problem. This simplification not only reduces the complexity of the criterion defined in Eq. (12), but seems very attractive since we will have to encode only one motion vector field instead of two. One can thus expect to use the spared bitrate to better transmit wavelet coefficients, thus increasing the coding performance. However, the motion compensation error is expected to be larger than in the previous case, especially when the motion is not uniform, as in higher temporal levels, for example. Moreover, the motion field may be less homogeneous than those obtained by the joint criterion (12). Note that the idea of constraining the forward and backward MV fields to be opposite was also proposed in [24], but the considered criterion was the minimization of the sum of forward and backward centered and normalized quadratic errors.

6 Temporal Scalability

Scalable video coding technology is important for applications such as Internet video, wireless LAN video, mobile wireless video for conversational, video on demand, live broadcasting, surveillance and storage etc. [25]. The scalable bitstream can in this case be adapted to variable conditions of the network (bandwidth, transmission errors) as well as to various capabilities (display size, CPU, memory, battery life) of the receivers. The scalability may concern the framerate (temporal scalability), the spatial resolution (spatial scalability) or the quality (SNR scalability). By extension, one can also talk about object-based scalability (when the bitstream is layered according to the importance of the objects in a scene). Complexity scalability is another interesting feature, allowing to adapt the processing resources involved in the decoding to the quality/resolution of the targeted reconstruction. In the sequel of this section, we will focus our analysis on the temporal scalability feature, even though the codec we use for simulations integrates combined temporal/spatial/fine granular scalabilities. Therefore, even though other strategies are possible [26], the bitstream is organized at the encoder side hierarchically by decreasing order of temporal decomposition levels, the temporal approximation subband being encoded at the beginning.

If the network conditions are the main limitation, then the bitstream can be parsed and cut before decoding. In this case, the decoder will only dispose of the temporal approximation subband at a certain level (so, a reduced framer-

ate) and if the user wishes to display the sequence at full framerate, then the synthesis scheme can be used to obtain the temporal upconversion, by introducing temporal detail subbands with null coefficients. Note that all spatial detail subbands of the decoded temporal levels are always preserved for reconstruction in our analysis. The comparison of temporal scalability features of two codecs can be performed in this case by taking as reference the original sequence at full framerate.

If on the contrary, the receiver capabilities are limiting the decoder framerate, then the displayed sequence will correspond to the approximation subband. The reference sequence in this case is not as easy to decide. One can compare the approximation subband with the original subsampled sequence, which has the drawback of leading to temporal aliasing and a jerkyness effect. Another possibility is to compute the PSNR of each decoded sequence at reduced framerate with the corresponding original approximation subband. However, the dynamic range of the coefficients in the approximation subband frames is larger than that of the original frames (and varies depending on the number of temporal decomposition levels and on the temporal filters used for the analysis), which would require the use of a different definition for the PSNR measure. Moreover, the reference in this case is different from one encoder to the other, and in such a comparison one can have a very good PSNR but a reference frame of very bad subjective quality. An acceptable comparison criterion could be the visual quality of the reconstructed approximation subband, which still remains very subjective, especially since the artefacts can be of different origins depending on the encoder characteristics. Visual comparison of the quality of the approximation subband frames in different MC MRAs is provided in Section 7.4. In the sequel of this section, we give a theoretical analysis of the upsampling process for the MC Haar and 5/3 filters.

If we consider a MC temporal analysis with J temporal resolution levels, temporal scalability of a factor $2^j,\ 0 < j \leq J$, is straightforwardly achieved by applying the decoding algorithm up to the reconstruction of the approximation subband at temporal level j. This corresponds to the second scenario presented above. If the reconstruction at the original framerate is targeted, as in the first scenario, then the synthesis algorithm needs to be applied also for the first j decomposition levels on the reconstructed approximation subband at level j and the remaining detail subbands set to zero. Two options are possible here: either the MV fields at the first j levels are used in this reconstruction, or the synthesis is performed without these vectors. The former case requires the entire transmission of the MV fields, at all resolution levels. Very little gain can be expected therefore at low bitrates from this scalable scheme. Moreover, the complexity related to the motion compensation is not reduced compared to the full framerate decoding (the only complexity reduction is the inverse spatial transform of the detail frames). The second alternative seems therefore more realistic for low bitrates and limited resources applications. This kind of

comparison of temporal scalability features is based on the reconstruction of the sequence at the original framerate. We believe that it provides an objective means to compare the quality of the approximation subband of two different 2D+t schemes.

In order to simplify our analysis, we analyse the case of connected pixels only. Subsequently, $l_{t,j}$ denotes the temporal approximation frame t at level j and $\widehat{x}_t$ the reconstructed frame t at the original framerate. At first, let us consider the case $j = 1$. Then, the equations allowing to upsample to the original framerate when using a 5/3 MRA are:

$$\widehat{x}_{2t}(\boldsymbol{n}) = l_{t,1}^{5/3}(\boldsymbol{n}) \tag{14}$$

$$\widehat{x}_{2t+1}(\boldsymbol{n}) = \frac{\widehat{x}_{2t}(\boldsymbol{n} - \boldsymbol{v}_{2t+1}^{+}(\boldsymbol{n})) + \widehat{x}_{2t+2}(\boldsymbol{n} - \boldsymbol{v}_{2t+1}^{-}(\boldsymbol{n}))}{2}. \tag{15}$$

We remark that (up to quantization effects) the reconstructed even frames correspond to the approximation frames (denoted here with an upper index in reference to the temporal structure), while the odd frames are obtained as an average of compensated neighboring frames. Not involving the motion vectors in the synthesis leads to the following reconstruction equations:

$$\widehat{x}_{2t}(\boldsymbol{n}) = l_{t,1}^{5/3}(\boldsymbol{n}) \tag{16}$$

$$\widehat{x}_{2t+1}(\boldsymbol{n}) = \frac{\widehat{x}_{2t}(\boldsymbol{n}) + \widehat{x}_{2t+2}(\boldsymbol{n})}{2}. \tag{17}$$

The situation is different for a Haar MRA. Indeed, when using motion vectors, the reconstruction with details set to zero leads to:

$$\widehat{x}_{2t}(\boldsymbol{n}) = l_{t,1}^{H}(\boldsymbol{n}) \tag{18}$$

$$\widehat{x}_{2t+1}(\boldsymbol{n}) = \widehat{x}_{2t}(\boldsymbol{n} - \boldsymbol{v}_{2t+1}^{+}), \tag{19}$$

and when the MVs are not used in the synthesis, to:

$$\widehat{x}_{2t}(\boldsymbol{n}) = l_{t,1}^{H}(\boldsymbol{n}) \tag{20}$$

$$\widehat{x}_{2t+1}(\boldsymbol{n}) = \widehat{x}_{2t}(\boldsymbol{n}). \tag{21}$$

Since odd frames are a repetition of even frames, we can expect a much lower average quality of the reconstructed sequence at the full framerate, compared to the original sequence.

Now, if $j > 1$ i.e. the temporal scalability factor is 2^j, it is easy to show by induction from Eqs. (20),(21) and (16),(17) that in the Haar case we reconstruct:

$$\widehat{x}_{2^j t+k}(\boldsymbol{n}) = l_{t,j}^{H}(\boldsymbol{n})\,, \qquad \forall k \in \{0, \ldots, 2^j - 1\}, \tag{22}$$

whereas for the 5/3 filters we have:

$$\widehat{x}_{2^j t+k}(\boldsymbol{n}) = \Big(1 - \frac{k}{2^j}\Big) l_{t,j}^{5/3}(\boldsymbol{n}) + \frac{k}{2^j}\, l_{t+1,j}^{5/3}(\boldsymbol{n})\,, \qquad \forall k \in \{0, \ldots, 2^j - 1\}. \quad (23)$$

This simply means that the reconstruction at the original framerate in the Haar case corresponds to the repetition 2^j times of the approximation frames at level j. In the 5/3 case, the reconstruction of each frame from the original sequence is obtained by a linear interpolation from the two adjacent approximation frames taking into account the temporal "distance" to these frames.

One could think of using the same kind of linear interpolation in the Haar case, but this would not necessarily be a good idea. Indeed, it is well known from the subband literature [27] that the interpolation filters used in the synthesis scheme should correspond to the ones used in the analysis scheme. To illustrate this for our temporal transforms, for simplicity we come back to the case $j = 1$, neglecting the quantization error on the approximation frame. If we were to apply the 5/3 interpolation formulas to the Haar approximation, we would obtain:

$$\widehat{x}_{2t}(\boldsymbol{n}) = l_{t,1}^{H}(\boldsymbol{n}) = \frac{x_{2t}(\boldsymbol{n}) + x_{2t+1}(\boldsymbol{n})}{2}$$
$$\widehat{x}_{2t+1}(\boldsymbol{n}) = \frac{l_{t,1}^{H}(\boldsymbol{n}) + l_{t+1,1}^{H}(\boldsymbol{n})}{2} = \frac{x_{2t}(\boldsymbol{n}) + x_{2t+1}(\boldsymbol{n}) + x_{2t+2}(\boldsymbol{n}) + x_{2t+3}(\boldsymbol{n})}{4}.$$

In particular, unlike the 5/3 case, the interpolation of $\widehat{x}_{2t+1}(\boldsymbol{n})$ is obtained by an asymmetric expression, giving the same weight to the frames x_{2t}, x_{2t+2} and x_{2t+3}, although the last frame should be less correlated with x_{2t+1} than the first two ones.

7 Simulation Results

7.1 Implementation Details

During the temporal filtering and in order to avoid boundary effects in the current GOF, our encoder processes frames from the next GOF and also keeps in memory frames and motion vectors from the previous GOF (see Fig. 5). This actually corresponds to a "sliding window" (or "on-the-fly") technique for temporal filtering and therefore the notion of group of frames previously introduced for the Haar multiresolution decomposition is no longer relevant. The boundaries of the sequence are processed using two-tap (Haar-like) update operators. A memory constraint buffer was used for the implementation, as described in [16]. Frames are buffered only when needed and the lifting implementation is exploited to perform "in place" calculations, thus reducing the

memory requirements, the memory access frequency and also the processing delay, which is an important aspect for streaming applications. Note that a

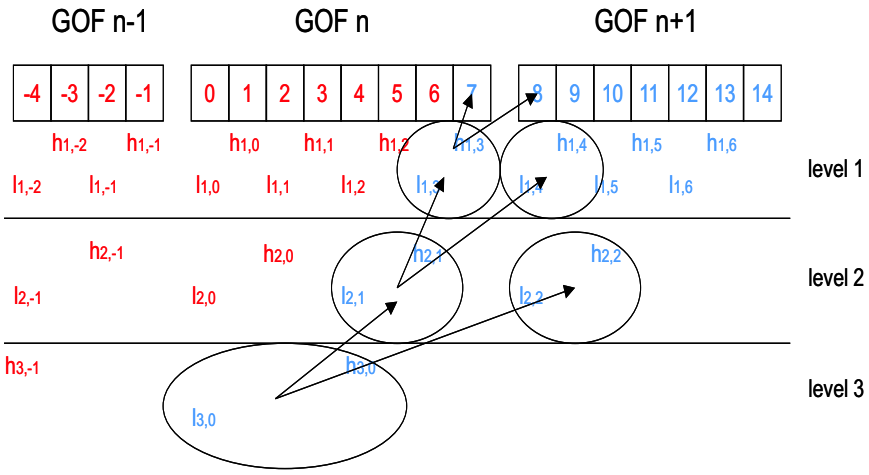


Fig. 5. Illustration of the temporal filtering over three decomposition levels: the computation of the third level approximation frame in n-th GOF requires prior processing of eight frames from the $n+1$-th GOF.

similar technique has been used in [28] for a 3D scheme but without ME/MC.

Motion estimation is done within the MC-EZBC framework [6,29] through the Hierarchical Variable Size Block Matching (HVBSM) algorithm with block sizes varying from 64x64 to 4x4. Search range is first initialized at [-2;2], is increased if no good match can be found and is doubled at each temporal level.

Temporal subbands are spatially decomposed over five resolution levels using biorthogonal 9/7 wavelets and the resulting spatio-temporal wavelet coefficients are encoded using the EZBC [29] algorithm. Motion vector fields encoding and bitrate allocation are done within the MC-EZBC framework ; MVF are encoded as quad-tree maps and motion vector values are encoded with a 0-order arithmetic coder, in raster-scan order.

7.2 Optimized ME Algorithm

First, we study the behaviour of the iterative algorithm proposed in Section 5. For complexity reasons, we have chosen the sum of absolute differences (SAD) as a distortion measure d for the ME algorithm. In Table 1 we compare the mean SAD of the temporal detail frames at different temporal resolution levels between the separate MEs, as it is done in the non modified 5/3 temporal decomposition, and the iterative algorithm presented in Section 5. As stated before, this algorithm requires one ME for initialization and each half iteration

requires an extra ME. As expected, we observe the global decrease of the criterion at each half iteration, reaching up to 15 % for two iterations, compared to the separate MEs approach. We also remark that only a half iteration leads to smaller of SAD values by up to 10 % compared to the separate MEs, for the same complexity (as the two approaches requires two MEs).

In order to better compare potential gains in terms of PSNR, the mean energy of detail frames is also given in Table 2. Even though this is not the criterion optimized in the iterative algorithm, it also significantly decreases with the number of estimations performed, reaching up to 30 % of decrease for two iterations when compared to the separate MEs approach.

Mean SAD	Separate MEs	0.5 it	1 it	1.5 it	2 it
Level 1	4.70	4.21	4.04	4.03	4.01
Level 2	7.78	6.99	6.72	6.71	6.67
Level 3	11.94	10.65	10.22	10.24	10.16
Level 4	17.46	15.60	15.01	15.03	14.92

Table 1
Mean SAD of temporal detail frames on Stefan CIF 30 fps, using a MVF separate search and the iterative MVF joint search, with a full pixel accurate motion estimation (the 0.5 it case has the same complexity of the separate MEs search and each new half iteration requires an extra ME).

Mean energy	Separate MEs	0.5 it	1 it	1.5 it	2 it
Level 1	79.33	62.67	57.04	56.95	56.22
Level 2	194.54	154.21	141.57	141.59	139.76
Level 3	433.38	339.09	310.11	312.98	308.29
Level 4	893.81	705.19	649.74	653.49	642.21

Table 2
Mean energy of temporal detail frames on Stefan CIF 30 fps, using a MVF separate search and the iterative MVF joint search, with a full pixel accurate motion estimation (the 0.5 it case has the same complexity of the separate MEs search and each new half iteration requires an extra ME).

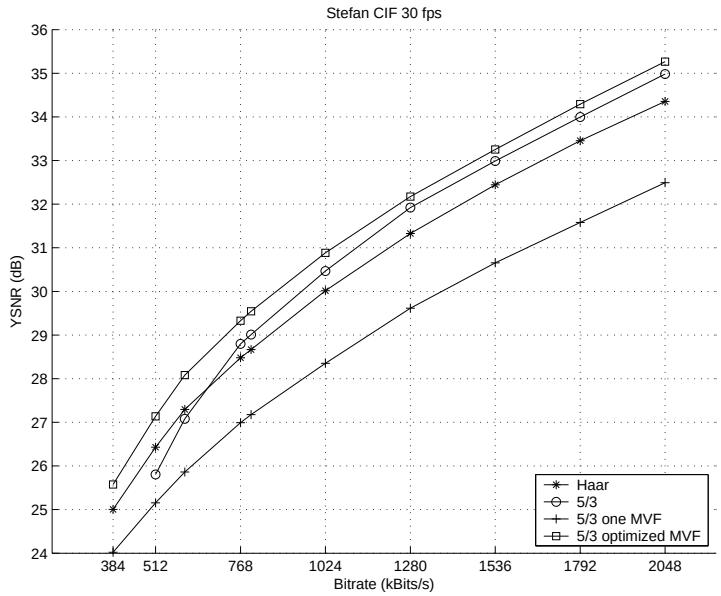
7.3 Global PSNR Results

For simulations we consider four representative test YUV420 video sequences in CIF format at 30 fps : “Stefan”, “Foreman”, “Mobile” and “Tempete” which have been selected for their very different motion and texture characteristics. These video sequences have been decomposed over four temporal levels and motion vectors have been estimated to $1/8^{th}$ pixel accuracy. All video sequences have been encoded in the YUV420 color mode meaning that the bit budget is shared by the luminance and chrominance components, the bitstream headers and the motion vector fields. Coding performance is expressed in terms of Y component PSNR (YSNR), calculated by averaging the Y component PSNR over all decoded frames.

We compare in Figs. 6, 7, 8 and 9 the compression performances of the following temporal filterbanks :

- the classical Haar MRA
- the plain 5/3 MCTF
- the 5/3 temporal analysis using only one MV field, resulting from the optimization of the criterion in Eq. (13)
- the 5/3 iterative joint optimized MVF search algorithm described in Section 5.

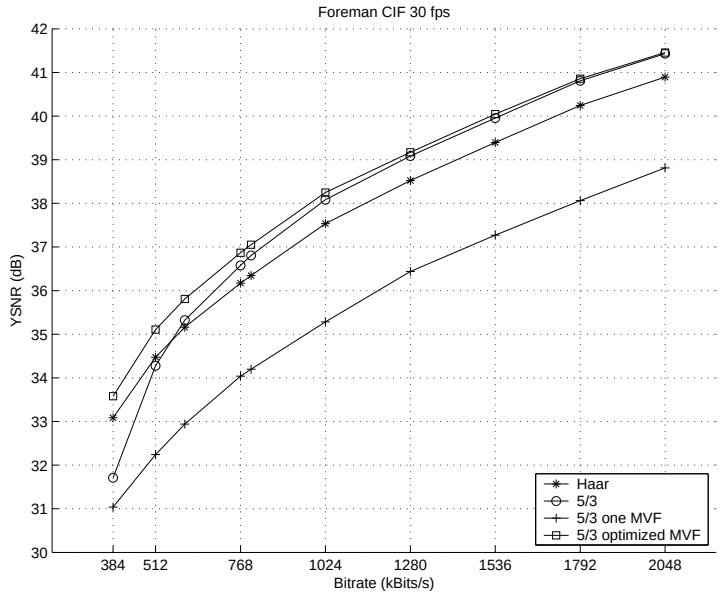
The simulations using the 5/3 iterative joint MVF search algorithm have been performed with one full iteration (equivalent of 3 MEs). For complexity reasons, we have chosen the sum of absolute differences (SAD) as a distortion measure d for the iterative joint optimized MVF search algorithm.



YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	26.42	28.47	30.01	32.44	34.35
5/3	25.80	28.79	30.46	32.98	34.98
5/3 one MVF	25.15	26.99	28.35	30.65	32.49
5/3 optimized MVF	27.13	29.32	30.88	33.25	35.26

Fig. 6. Rate-distortion curves and figures for different temporal filters on "Stefan" CIF video sequence.

We first compare the plain 5/3 and the Haar filterbanks. We remark that the 5/3 filterbank is much more effective than Haar at medium and high bitrates, where it surpasses the latter by almost 1 dB. This can be explained by a better temporal prediction due to the bidirectional ME/MC, as well as a bidirectional update in the computation of the approximation subband. However, this difference is slightly smaller on sequences like "Foreman", where the complex rotational motions reduce the benefits of the bidirectional operators. At

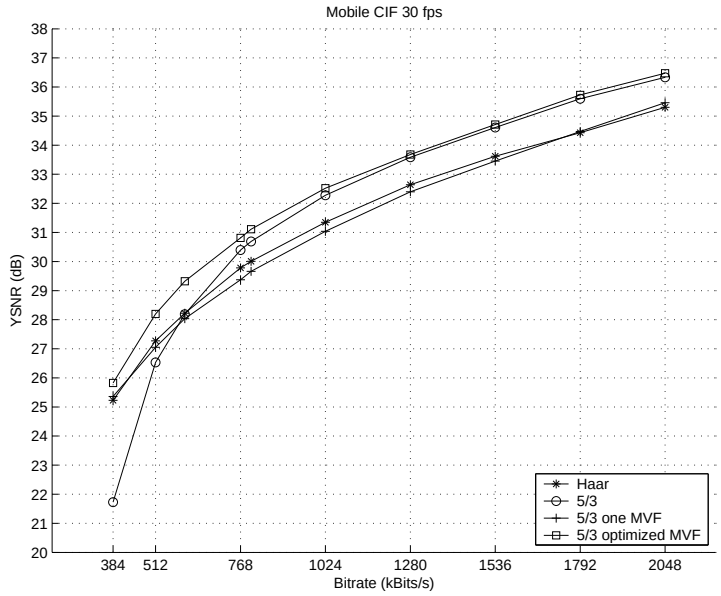


YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	34.47	36.17	37.53	39.39	40.89
5/3	34.27	36.57	38.08	39.95	41.43
5/3 one MVF	32.24	34.03	35.28	37.27	38.81
5/3 optimized MVF	35.10	36.86	38.24	40.04	41.45

Fig. 7. Rate-distortion curves and figures for different temporal filters on "Foreman" CIF video sequence.

low bitrates the Haar MRA is more efficient, essentially due to the fact that in the 5/3 MRA the bit budget is mainly spent to encode two MV fields. Due to the lossless compression of these two motion vector fields, the 5/3 scheme can start decoding only at a higher bitrate than Haar.

We expect that the scheme using the constrained MV field defined by Eq. (13) (and denoted by "5/3 one MVF" in Figs. 6-9) will compensate this drawback. This criterion could be effective at low bitrates, where the reduction of the number of MV fields results in significant bit savings and compensates for the constraint we introduced. However, for sequences having high motion activity like "Stefan" and "Foreman", the quality of the reconstructed sequence is reduced compared to the case of two independently estimated motion vectors, because of the constraint we impose. Indeed, especially in the last levels of the temporal analysis, the "temporal distance" between approximation frames on which we perform motion estimation is quite large, and the assumption of uniform motion is not verified. This results in a greater temporal prediction error that needs to be encoded. In the meantime, sequences with uniform motion, like "Mobile" and "Tempete", take better advantage of the contrained ME strategy and the coding performance at low bitrates is closer to that of the plain 5/3 filterbank.

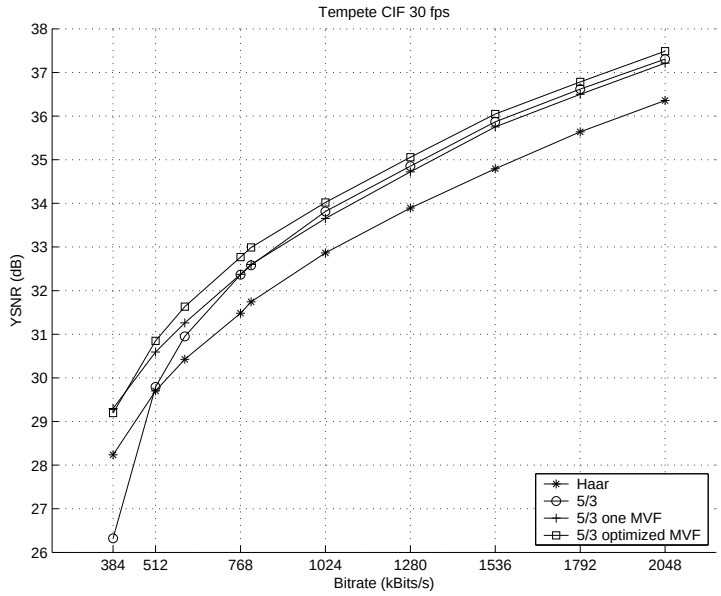


YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	27.27	29.78	31.35	33.61	35.30
5/3	26.53	30.39	32.27	34.60	36.33
5/3 one MVF	27.05	29.37	31.03	33.45	35.45
5/3 optimized MVF	28.19	30.81	32.52	34.70	36.47

Fig. 8. Rate-distortion curves and figures for different temporal filters on "Mobile" CIF video sequence.

We now compare the 5/3 optimized MVF filterbank with the previously introduced schemes. This MRA gives consistently overall better results than the one based on the plain 5/3 temporal filters and much better than Haar at all bitrates. This shows that the *joint* estimation of the forward and backward MVs can lead to significant improvements not only in the temporal prediction, visible especially at medium and high bitrates, but also in the coherence of the motion vectors. This latter aspect explains the gain at low bitrates, where the main part of the bit budget is allocated to MV coding, and makes competitive the 5/3 optimized structure with the Haar MRA also at low bitrates. Indeed, an interesting aspect of the actual implementation is that the measure d considered in the iterative algorithm is a distortion metric, but every minimization step is followed by a *rate-distortion* pruning of the MV field [6]. The two final MV fields obtained this way are much smoother than those obtained by separate estimations and part of the bit savings is related to this phenomenon. It also explains why it is possible to get better results than Haar even at very low bitrates as we have an almost equivalent number of bits for MV coding and a better temporal prediction.

As an efficient MVF coding in these schemes, fully exploiting all spatio-temporal redundancies between MVF, is still an issue that needs to be explored, we compare now the optimized 5/3 scheme with the plain 5/3 filter-



YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	29.70	31.48	32.86	34.79	36.35
5/3	29.79	32.36	33.81	35.86	37.30
5/3 one MVF	29.29	32.37	33.65	35.75	37.21
5/3 optimized MVF	30.84	32.76	34.01	36.04	37.48

Fig. 9. Rate-distortion curves and figures for different temporal filters on "Tempete" CIF video sequence.

bank by only assessing the energy compaction feature of this MRA and by excluding the motion coding cost from the bit budget. We consider therefore an ideal situation, where the best available MVF are used for MC and their cost is identical (or negligible) for the two situations. In order to satisfy the first hypothesis (best MVF), we considered the dense MVF obtained before pruning. Under these hypotheses and by neglecting the cost of the motion, Tab. 3 compares the plain 5/3 filterbank with the optimized 5/3 structure. The difference in PSNR is about 0.3 - 0.4 dB for all bitrates, which is comparable with the difference in PSNR between these two schemes at high bitrate when motion is part of the bitstream.

YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
5/3	32.76	34.12	35.13	36.89	38.28
5/3 optimized MVF	33.00	34.38	35.43	37.22	38.67

Table 3
PSNR values for different temporal filters on "Mobile" CIF video sequence without considering the cost of the MVF in the bit budget. MC was performed with dense MVF.

As the difference in coding performance at low bitrates between these two schemes (when motion cost is included) is much larger than 0.3 dB (see Figs. 6-9), these results highlight once again the fact that ME/MC is an essential part of the non-linear temporal transform. Our optimization of the MVF involved

in this scheme influences the coding performance both by providing a better motion prediction, and thus compacting the information carried by the spatio-temporal coefficients (as shown by Tab.3), and also by reducing the bitrate necessary for motion encoding. This was achieved by the joint estimation of the two MVFs and by including the pruning step in the iterative algorithm.

7.4 Temporal Scalability Results

In the simulations conducted in this subsection we have encoded only the Y component of the test sequences using a full-pel motion estimation accuracy.

In order to compare the temporal scalability properties of the new temporal analysis with the Haar one, we have as before decomposed over four temporal levels. In Fig. 10 the approximation frames at the fourth level obtained with the Haar scheme and with the optimized 5/3 filterbank are compared. The blurring introduced by the former temporal MRA is much more important than the artefacts resulting from the processing with the latter one.

Fig. 10. Part of the approximation frames at the fourth temporal decomposition level obtained with a Haar MRA (left) and with the optimized 5/3 filterbank (right) for the sequence "Tempete" in CIF format.

As we explained in Section 6, excepting this visual comparison of the approximation subband, an objective way to assess the performance in temporal scalability of two subband schemes is the PSNR of the reconstructed sequence at the original framerate, taking as reference the original sequence. If for example the first two temporal levels are missing, the sequence at the original frame rate is reconstructed using temporal detail frames set to zero instead of the missing temporal subbands, and compared with the outcome of the original sequence. Fig. 11 illustrates this comparison between Haar and optimized 5/3 filterbanks with the first two temporal levels missing at the decoder. The original sequence was encoded at 2048 kbs and then only the last two tem-

poral levels have been decoded. One can remark the difference of 3 to 4 dB between the two schemes, even at this very high bitrate. Moreover, the PSNR peaks for the Haar structure are clearly synchronized with the position of the approximation frames, and in between the PSNR decreases continuously. For the 5/3 scheme this latter phenomenon is attenuated, the reconstructed frames being obtained by averaging the approximations at different levels, according to formula (23).

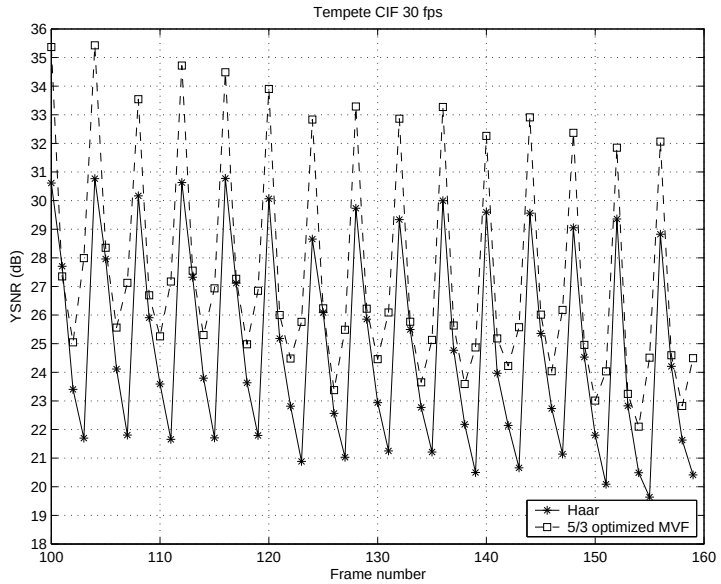


Fig. 11. PSNR of the reconstructed sequence at the original framerate from two over four temporal levels by using a Haar and an optimized 5/3 temporal MRA (frames 100 - 160 from the "Tempete" sequence, 30 fps, CIF format).

We now compare the average PSNR of the reconstructed sequences with the same MRAs when decoding three over four temporal levels. Figs. 12, 13, 14 and 15 present the rate-distortion curves when the MV fields at the first temporal decomposition level are *not* used in the reconstruction. We remark gaps of more than 3 dBs between Haar and the 5/3 MRA. These good results can be explained by the fact that the 5/3 scheme better concentrates energy on the temporal approximation subband, due to the bidirectional motion compensated prediction and update operators. Moreover, the reconstruction of the original framerate by interpolation of the odd frames based on the even ones consequently increases the average PSNR at a given bitrate compared to the simple repetition in the Haar reconstruction.

The difference between the 5/3 structure and its optimized counterpart is smaller at medium and high bitrates than for full framerate decoding. This is related to the fact that reconstructing 3 over 4 temporal levels means involving two times less motion vectors than for a full reconstruction at the same bitrate and therefore the percentage of MV in the total bitrates is smaller. The observed improvement mainly comes from the better temporal predic-

tion, which additionally improves the quality of the approximation subband. The difference is however noticeable at low bitrates, both because of the homogeneity of the MV fields, which are thus easier to encode, and due to the better temporal prediction obtained with the joint criterion.

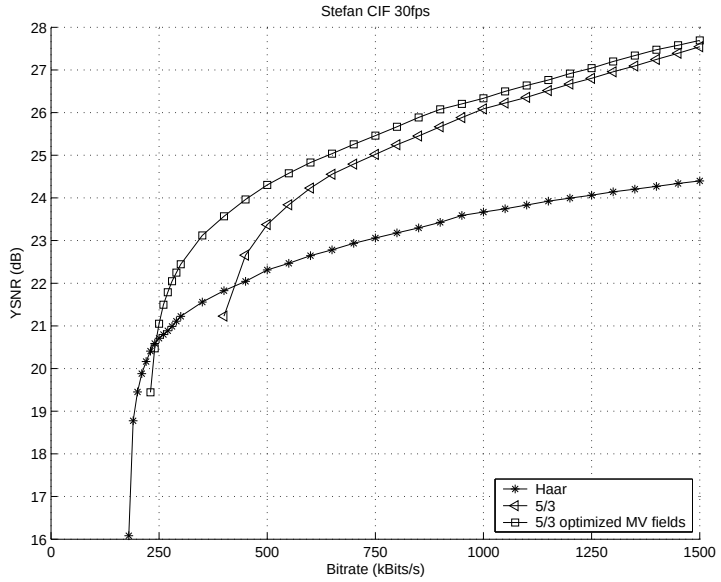


Fig. 12. Reconstruction without the first level motion vectors of three temporal decomposition levels out of four for the sequence "Stefan" in CIF format.

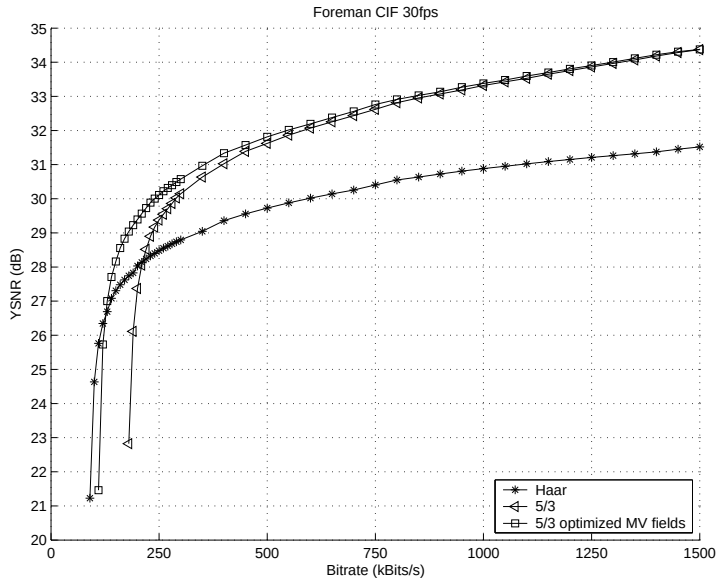


Fig. 13. Reconstruction without the first level motion vectors of three temporal decomposition levels out of four for the sequence "Foreman" in CIF format.

Figs. 16, 17, 18, and 19 show the rate-distortion curves when the MV fields at the first temporal decomposition level are also used in the reconstruction of the sequence at the full framerate. The use of motion compensation in the synthesis algorithm improves a lot the quality of the reconstruction at

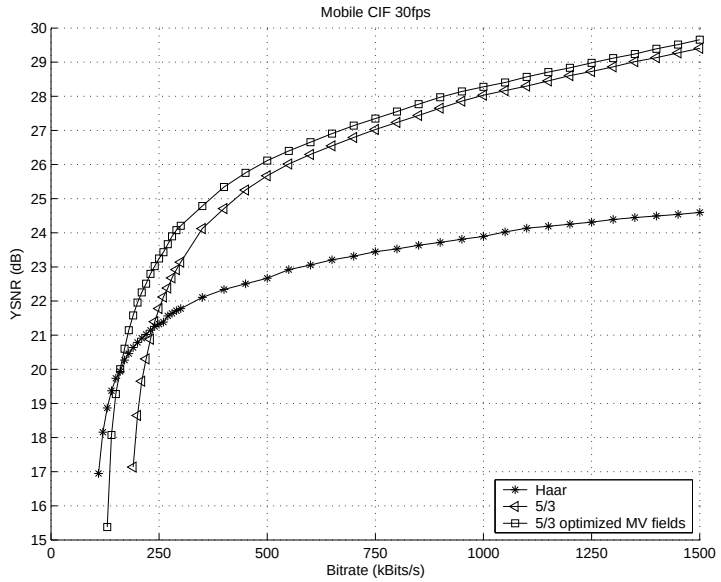


Fig. 14. Reconstruction without the first level motion vectors of three temporal decomposition levels out of four for the sequence "Mobile" in CIF format.

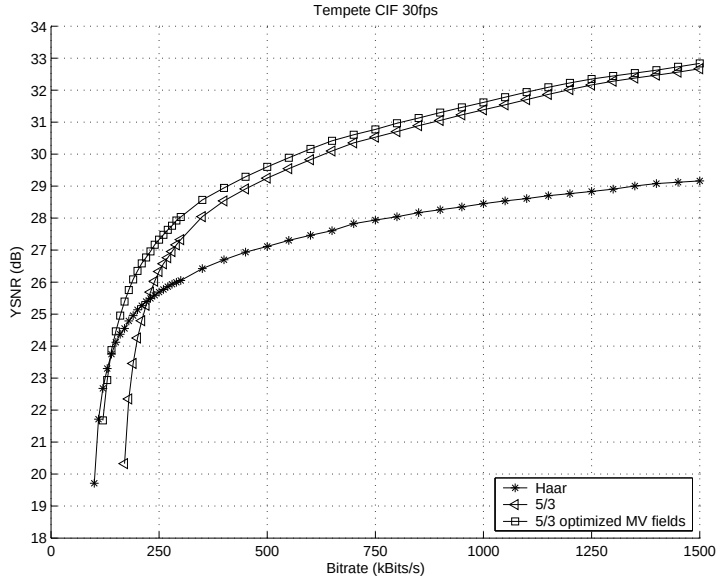


Fig. 15. Reconstruction without the first level motion vectors of three temporal decomposition levels out of four for the sequence "Tempete" in CIF format.

medium and high bitrates, as expected. It even becomes comparable with the reconstruction using all the high frequency subbands. One can remark that these high frequency details improve the quality of the reconstructed sequence mainly at high bitrates. Before, their lack is compensated by a finer encoding of the detail frames at intermediate temporal decomposition levels.

At low bitrates however, not using the motion vectors leads to better performances. Associated with reduced complexity, this method is interesting in

terms of temporal scalability for devices with limited resources.

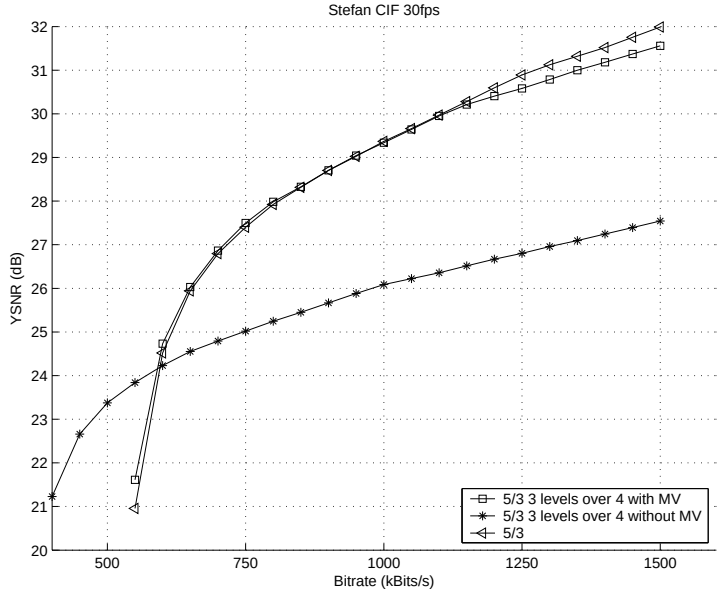


Fig. 16. Reconstruction using the first level motion vectors of three temporal decomposition levels out of four for the sequence "Stefan" in CIF format.

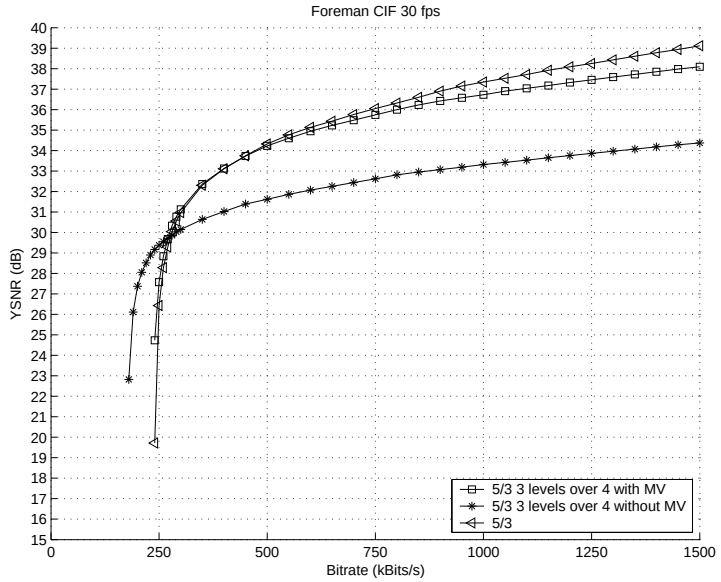


Fig. 17. Reconstruction using the first level motion vectors of three temporal decomposition levels out of four for the sequence "Foreman" in CIF format.

8 Conclusion

We have presented a general lifting formulation for motion compensated temporal multiresolution analysis, which provides a comprehensive framework for

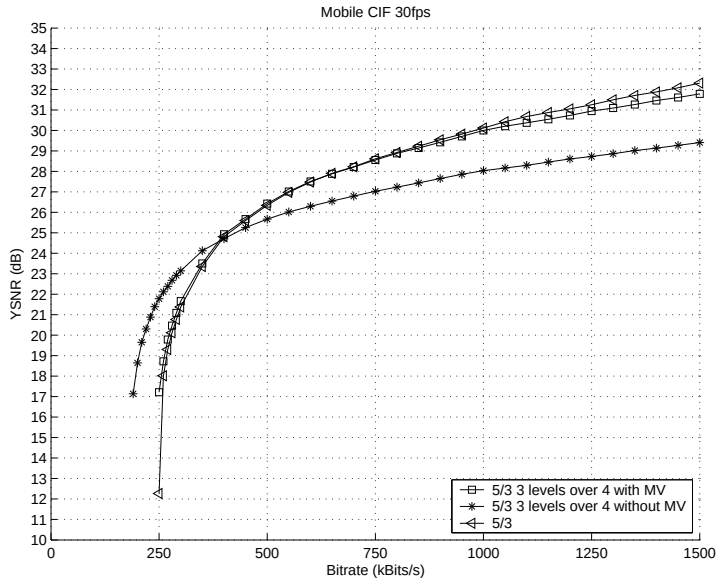


Fig. 18. Reconstruction using the first level motion vectors of three temporal decomposition levels out of four for the sequence "Mobile" in CIF format.

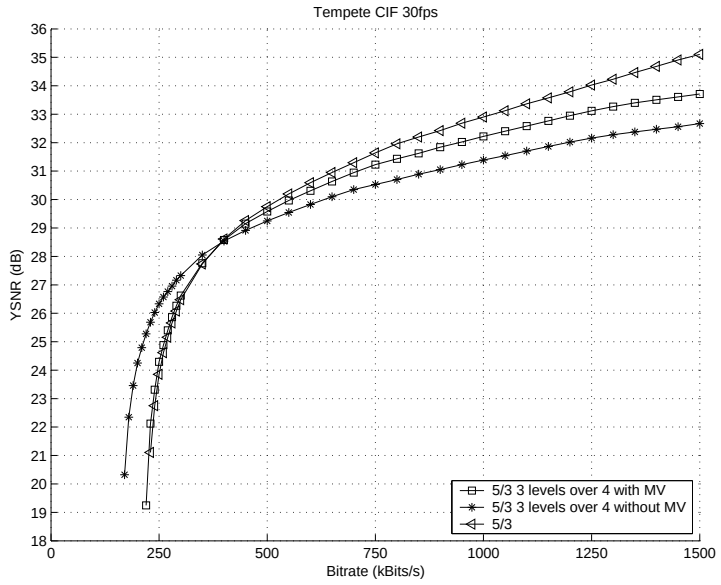


Fig. 19. Reconstruction using the first level motion vectors of three temporal decomposition levels out of four for the sequence "Tempete" in CIF format.

the design of optimized 5/3 MCTF. We have shown the importance of the bidirectional prediction and update on the coding efficiency as well as on the temporal scalability features of these schemes. A new optimization criterion for joint estimation of the forward and backward motion vectors has been proposed and shown to bring substantial improvement in the rate-distortion performance of the proposed video codec. This shows that motion-compensated wavelet temporal filtering offers powerful solutions for future scalable video coding standards. However, a lot of work remains to be done to take full

advantage of all the potential of this approach.

References

[1] H. J. A. M. Heijmans and J. Goutsias, "Nonlinear multiresolution signal decomposition schemes: Part II: morphological wavelets," *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1897–1913, 2000.

[2] F. J. Hampson and J.-C. Pesquet, "M-band nonlinear subband decompositions with perfect reconstruction," *IEEE Transactions on Image Processing*, vol. 7, pp. 1547–1560, 1998.

[3] G. Piella, B. Pesquet-Popescu, and H. Heijmans, "Adaptive update lifting with a decision rule based on derivative filters," *IEEE Signal Processing Letters*, pp. 329– 332, oct 2002.

[4] G. Karlsson and M. Vetterli, "Three-dimensional subband coding of video," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, NY, Apr. 1988, pp. 1100–1103.

[5] D. Taubman and A. Zakhor, "Multi-rate 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 3, pp. 572–588, 1994.

[6] S.J. Choi and J.W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 8, pp. 155–167, 1999.

[7] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, pp. 559–589, 1994.

[8] B.-J. Kim, Z. Xiong, and W.A. Pearlman, "Very low bit-rate embedded video coding with 3-D set partitioning in hierarchical trees (3D-SPIHT)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 1365–1374, 2000.

[9] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proceedings of the IEEE International Conference on Image Processing*, Thessaloniki, Greece, Oct. 2001.

[10] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Salt Lake City, UT, May 2001.

[11] K. Hanke, J.-R. Ohm, and T. Rusert, "Adaptation of filters and quantization in spatio-temporal wavelet coding with motion compensation," in *Proc. of the Picture Coding Symposium*, St. Malo, France, April 2003, pp. 49–54.

[12] K. Hanke, "Interframe wavelet video coding with lowpass transition," doc. m8997, Shanghai MPEG meeting, Oct. 2002.

[13] J.W. Woods, P. Chen, and S.-T. Hsiang, "Exploration experimental results and software," doc. m8524, Klagenfurt MPEG meeting, July 2002.

[14] D. Turaga and M. van der Schaar, "Unconstrained temporal scalability with multiple reference and bi-directional motion compensated temporal filtering," doc. m8388, Fairfax MPEG meeting, 2002.

[15] J.-R. Ohm, "Complexity and delay analysis of MCTF interframe wavelet structures," doc. m8520, Klagenfurt MPEG meeting, July 2002.

[16] Y. Zhan, M. Picard, B. Pesquet-Popescu, and H. Heijmans, "Long temporal filters in lifting schemes for scalable video coding," doc. m8680, Klagenfurt MPEG meeting, July 2002.

[17] A. Secker and D. Taubman, "Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation," in *Proceedings of the IEEE International Conference on Image Processing*, Oct. 2002.

[18] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, pp. 186–200, 1996.

[19] D. Turaga, M. van der Schaar, and B. Pesquet-Popescu, "Temporal prediction and differential coding of motion vectors in the MCTF framework," in *Proceedings of the IEEE International Conference on Image Processing*, Barcelona, Spain, Oct. 2003.

[20] C. Tillier, B. Pesquet-Popescu, Y. Zhan, and H. Heijmans, "Scalable video compression with temporal lifting using 5/3 filters," in *Picture Coding Symposium, PCS-2003*, St. Malo, France, Apr. 2003.

[21] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, pp. 332–369, 1998.

[22] D.S. Turaga, M. van der Schaar, and B. Pesquet-Popescu, "Differential motion vector coding for scalable coding," in *Proc. of Image and Video Communications and Processing*, Santa Clara, CA, Jan. 2003, vol. SPIE 5022, pp. 87–97.

[23] D. S. Turaga, M. van der Schaar, and B. Pesquet-Popescu, "Complexity scalable motion compensated wavelet video encoding," submitted to IEEE Trans. on Circ. and Syst. for Video Tech., 2003.

[24] V. Valentin, M. Cagnazzo, M. Antonini, and M. Barlaud, "Scalable context-based motion vector coding for video compression," in *Proc. of the Picture Coding Symposium*, St. Malo, France, April 2003, pp. 63–70.

[25] ISO/IEC JTC1/SC29/WG11, "Requirements and applications for scalable video coding," doc. n6025, Brisbane MPEG meeting, Oct. 2003.

[26] V. Bottreau, M. Bénetière, B. Felts, and B. Pesquet-Popescu, "A fully scalable 3D subband video codec,," in *Proc. of IEEE Int. Conf. on Image Proc. (ICIP)*, Thessaloniki, Greece, oct 2001.

[27] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[28] C. Parisot, M. Antonini, and M. Barlaud, "3D scan based wavelet transform for video coding," in *IEEE Fourth Workshop on Multimedia Signal Proc.*, 2001, pp. 403–408.

[29] "3D MC-EZBC software package," available on the MPEG CVS repository.