# Temporal scalability through adaptive $M$-band filter banks for robust H264/MPEG-4 AVC video coding

C. Bergeron, C. Lamy-Bergot, G. Pau, B. Pesquet-Popescu

***Abstract*— This paper presents different structures that use adaptive $M$-band hierarchical filter banks for temporal scalability. Open-loop and closed-loop configurations are introduced and illustrated using existing video codecs. In particular, it is shown that the H.264/MPEG-4 AVC codec allows us to introduce scalability by frame shuffling operations, thus keeping backward compatibility with the standard. The large set of shuffling patterns introduced here can be exploited to adapt the encoding process to the video content features, as well as to the user equipment and transmission channel characteristics. Furthermore, simulation results show that this scalability is obtained with no degradation in terms of subjective and objective quality in error-free environments, while in error-prone channels the scalable versions provide increased robustness.**

***Index Terms*— scalable video coding, H.264/MPEG-4 AVC, temporal scalability, backward compatible coding, $M$-band motion-compensated temporal filter banks.**

## I. Introduction

Modern wireless communication applications relying on the use of video services and videostreaming are facing a problem that high speed wired networks seemed to have overcome: for them, the available bandwidth is still a limiting factor. Moreover, IP wireless networks have to cope with both bit errors and packet losses. This is why a new generation of standards, such as H.264/MPEG-4 AVC finalized in May 2003 [1] jointly by ISO MPEG and ITU-T, and also the new wavelet-based codecs solutions proposed within the Scalable Video Coding (SVC) group, such as [2], take into account the interaction with the network (for the former, through the Network Abstraction Layer concept). Such codecs provide significant compression efficiency improvement when compared to the other existing standards such as MPEG-2 or MPEG-4, and that is why they are so attractive for multimedia applications over wireless communication links. However, H.264/MPEG-4 AVC does not support scalability, which is a very efficient tool to adapt to the bandwidth variations and to the error-prone nature of the wireless channels[1]. Solutions are currently being proposed in the literature and within the SVC standardisation group to adress this limitation, generally by introducing modifications to the H.264/MPEG-4 AVC syntax to integrate Progressive Fine Granular Scalability coding or subband decompositions [3], [4]. In parallel, solutions relying on motion-compensated (MC) spatio-temporal subband decompositions are being proposed, first with a classical dyadic subband decomposition [5], then by exploiting a non-linear lifting implementation [6] and making use of efficient 3D entropy coding algorithms [7]. Such solutions are unfortunately not compliant with basic H.264/MPEG-4 AVC decoders and often introduce a higher level of complexity, which may not be acceptable for the use in small and cheap mobile equipments.

Following the approach initiated in [8] where the introduction of temporal scalable solutions fully compliant with H.264/MPEG-4 AVC has been proposed and interpreted in the framework of adaptive $M$-band hierarchical filter banks, in this paper we show that this framework can be further generalized to include dyadic temporal decompositions and also to introduce scalability inside both open-loop and closed-loop temporal prediction structures. In particular, we show that the resulting hierarchical representation of H.264/MPEG-4 AVC frames inside a Group of Pictures (GOP) preserves the coding performance of the original non-scalable scheme in an error-free environement, and improves the subjective and objective quality of the sequences transmitted over error-prone channels.

This paper is organized as follows. Section II introduces the proposed hierarchical filterbank structures and discusses their interest for video coding and scalability. In Section III, an application of these filter banks to the temporal prediction compliant with the H.264/MPEG-4 AVC standard is proposed and discussed. In Section III-C the adaptation of this filterbank scheme to the case of H.264/MPEG-4 AVC where the number of reference frames should be limited, namely for simple levels, is considered. Section IV describes a practical setup for easily applying filtering in a conformant way to an H.264/MPEG-4 codec, through the application of an interleaver, as well as the simulation chain model considered for testing the various shuffling configurations, both in error-free and error-prone environments. Finally, in Section V experimental results are presented and in Section VI the conclusions are drawn.

## II. $M$-band hierarchical filter banks

Temporal scalability is achieved by introducing a hierarchy among the frames encoded in a group of pictures. This is true for both classical closed-loop temporal Differential Pulse Code Modulation (DPCM) schemes, and for motion-compensated wavelet decompositions, using open-loop schemes based on motion-compensated temporal filter banks. In both cases, some constraints are introduced in the temporal prediction in order

This work was partially supported by the European Community through project IST-FP6-001812 PHOENIX and project IST-FP6-1-507113 DANAE.

Cyril Bergeron and Catherine Lamy–Bergot are with Thales Communications, EDS/SPM, Colombes, France.

Grégoire Pau and Béatrice Pesquet–Popescu are with Ecole Nationale Supérieure des Télécommunications, TSI Department, Paris, France

[1] Temporal scalability can be achieved using B frames in profiles that support B frames. The Baseline profile of H.264 does not support B frames.

to create successive layers of importance. In this section we point out the analogies between the two approaches, by describing a common framework based on temporal subband decompositions.

Let us consider the lifting form of the motion-compensated wavelet decompositions [9]. Basically, the desired temporal dyadic filter bank is represented in its lifting form with one (or several) *predict* and *update* steps involving motion compensation. In designing these structures, a particular attention should be paid to the motion prediction direction in the temporal operators, so as to facilitate the filtering along motion trajectories. In order to simplify the comparison, our model will not include the update step (which is however essential for the good performances of these schemes). For a bidirectional prediction (from past and future frames, as commonly used in the 5/3 filter banks), the basic scheme is illustrated in Fig. 1, where the input frames (at times $t \in \mathbb{N}$) are denoted by $x_t$, and the resulting temporal detail frames, corresponding to high temporal frequencies, are denoted by $h_t$. After the quantization block $Q$, the same frames are denoted by $\bar{x_t}$, respectively $\bar{h_t}$. In this one-level decomposition, the even indexed frames (following the notation in Fig. 1) will enter the approximation subband, while the error prediction frames will yield the detail subband.

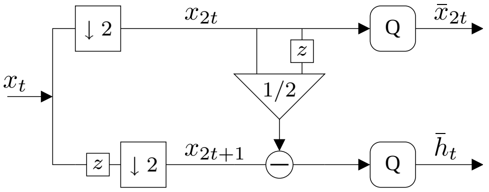

Fig. 1. Open-loop prediction scheme – one level of decomposition.

By just changing the place of one of the quantizers in Fig. 1, we get a prediction based on the previously reconstructed frames, as illustrated in Fig. 2 (here, for the sake of simplicity, the inverse quantization and the spatial direct and inverse transforms have been omittted).

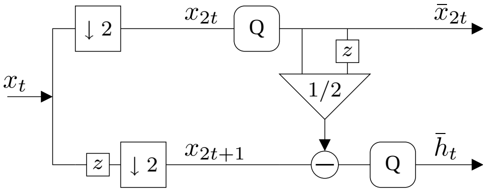

Fig. 2. Basic closed-loop prediction scheme.

By iterating the splitting into odd and even frames, we obtain a four-band polyphase decomposition, on which the successive application of the previous prediction scheme leads to an approximation subband (containing the equivalent to the Intra frames), a detail subband at the coarse resolution level similar to a B-frame in the base layer (denoted in the Fig. 3 by $h_t^2$), and two detail frames at the finest resolution level, $h_{t,1}^1$ and $h_{t,2}^1$, similar to B-frames in the enhancement layer. Note that this hierarchical structure can be seen as consisting of two levels of a wavelet decomposition without the update lifting step.

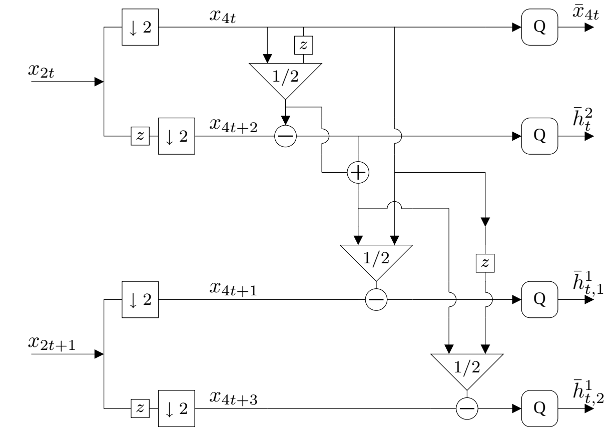

Fig. 3. Open-loop scheme with 4 temporal subbands (2 temporal decomposition levels).

The two-level structure in Fig. 3 can be transposed into a closed-loop structure, equivalent to a four-band decomposition, as illustrated in Fig. 4.

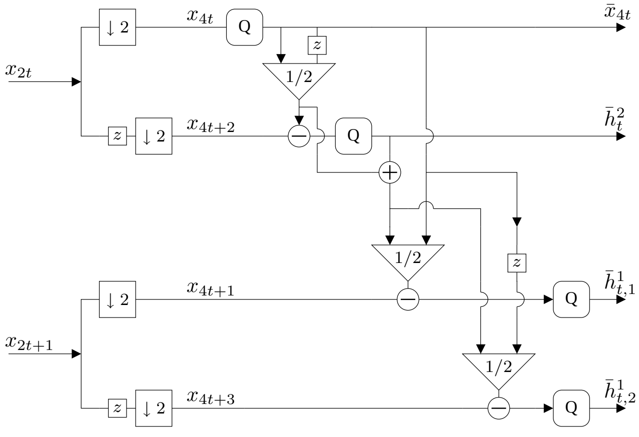

Fig. 4. Closed-loop scheme with 4 temporal subbands (2 temporal decomposition levels).

The previous open-loop and closed-loop subband decompositions can be extended to an arbitrary number of decomposition levels, involving groups of frames with power of two number of frames[2].

A common property of these structures is that each GOP is independently decodable, which is a very useful feature in error-prone environments, in order to avoid error propagation.

## III. Application to the H.264/MPEG-4 AVC video standard

Relying on the motion-compensated temporal subband decompositions presented in Section II, in this section we show that the existing properties of the H.264/MPEG-4 AVC standard allow us to build a hierarchical representation inside

[2] Note also that in [8] we have introduced temporal subband decompositions with an odd number of subbands, allowing pyramidal or tree-like hierarchical structures.

the GOPs, without requiring any modification or addition to the standardized codec, thus remaining fully compliant with each of the standard profiles [3].

Moreover, contrarily to previous video coding standards that were using simple reference modes, for which the prediction of inter frames could only be done with respect to a given preceding picture, it is important to point out that H.264/MPEG-4 AVC allows the usage of up to 16 different frames as reference in some levels, for each P-slice. In practice, this capability means that several previous frames (in encoding order) can be used as references for the current frame.

Considering groups of pictures of $N$ frames, denoted by their original time reference $\{0,\ 1,\ 2,\ \ldots,\ N-1\}$, the aim of our approach is to intelligently distribute the frames so that the encoding process that will follow is done efficiently. As a matter of fact, the classical prediction order in the GOP may not be the most efficient one, from a temporal scalability standpoint because (a) one wants to obtain a regular frame rate when using the temporal scaling, and (b) a better compression efficiency can be obtained when placing the reference frames closer to the predicted ones in display order [8]. The classical decomposition, which we call "normal" configuration is presented in Fig. 5 in the case of a GOP size $N = 7$. The dependencies between frames are illustrated by the arrows in the figure, which shows that frame 1 depends on frame $0$, frame 2 on frame 1 and consequentely also on frame $0$ etc. This can be obtained from the closed-loop scheme in Fig. 4, by considering only unidirectional predictions and as many levels as the number of frames in the GOP.

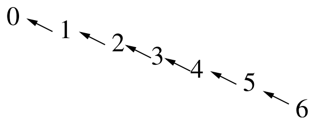

Fig. 5. Normal configuration, GOP size = 7. The arrows are directed towards the reference frame.

Three different approaches for temporal scalability are considered in the following:

- symmetric filtering schemes, meaning that a unique Intra frame, which is taken for these configurations to be actually an Instantaneous Decoding Refresh (IDR) frame, is considered as main reference for the whole GOP; in our approach by frame shuffling, it is placed in the middle of the GOP (in output display order);
- asymmetric filtering schemes, where each Intra frame is used as reference by two consecutive GOPs,
- a combination of the above two approaches, taking into account possible limits in terms of frame reference buffer sizes, to meet the eventually more restrictive requirements of certain levels of the standard and practical implementations.

### A. Symmetric filtering schemes

A first decomposition configuration ensuring the temporal scalability features, that we shall call "Zigzag" configuration, is illustrated in Fig. 6 for $N = 15$. Firstly introduced in [8], this regular pattern corresponds to the subband decomposition for GOP of size $N = 2^L - 1$, $L \in \mathbb{N}$. It is obtained as follows:

- select a reference frame (the Intra frame) for the first level having the temporal index at the median value of the GOP, where the median index is $median = (GOP_{size} + 1)/2$; define each part separated by the median as sub-GOP;
- repeat for each sub-GOP: take as reference frames the median ones and define accordingly remaining sub-GOPs for the next level.

In practice, one sees that the first resolution level consists only of the Intra frame which is placed at the median value of the GOP (and not in the beginning of the GOP as usual). In this configuration, dependencies between frames are very important, as each frame $i$ can depend (based on the efficiency of the compression mechanism and of the considered sequence) of the $i-1$ previous ones. In practice, one observes that the coding efficiency is smaller for the first levels, since the temporal distance between the predicted and the reference frames can be greater than one. Still, simulation results show that this is in practice compensated by the fact that the latest frames offer better compression rate, as they are closer to their main reference frames.

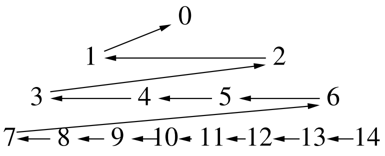

Fig. 6. Zigzag configuration, GOP size = 15.

This "Zigzag" decomposition is obviously very efficient for $N = 2^L - 1$, which corresponds to a fully regular repartition pattern of the subband decomposition, but can easily be used in other cases, at the cost of some loss in compression efficiency. In particular, one can think to generalize the decomposition to other sub-sampling factors R (greater than 2),as well as for other values of $N$. This hierarchical structure can be achieved as follows [8]:

- select $R-1$ reference frames at each level (for instance, with the Intra frame being the first of them) at equal temporal distance in the GOP, i.e. having temporal indices $m_i = \lfloor i(GOP_{size}+1)/R \rfloor$ for $i = 1, ..., R-1$; each part of the GOP between these frames is defined as a sub-GOP;
- repeat for each sub-GOP: take $R-1$ reference frames uniformly distributed in the sub-GOP and define accordingly $R$ remaining sub-GOPs.

Fig. 6 then corresponds to $N = 15$ and $R = 2$ for each level. Another illustration is given in Fig. 7 for a GOP of 19 frames, with sub-frame rate $R = 3$ and three temporal levels.

In such generalizations, note that for values of $N$ different of $2^L - 1$, that we call "irregular" $N$ values, many different

[3] In the baseline profile, this approach corresponds to using predictive (P) frames. But this method could also be easily generalized with B frames for other profiles. As a consequence, the proposed temporal scalability feature is the only one compatible with all the profiles.

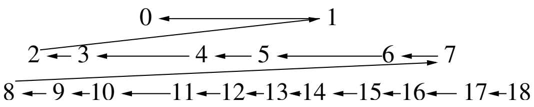

Fig. 7. Generalized Zigzag configuration with $R = 3$, GOP size = 19.

decompositions can be proposed that will have similar performances. As a consequence, when considering such irregular values of $N$, it is recommended to consider adaptation of the generalized pattern based on regular repartition of reference frames at each level.

To illustrate the advantage of this ”Zigzag” scalable structure, we introduce other GOP reorganizations that correspond to structures with smaller gaps between the frames at the first level of importance. Two such decompositions are considered that can be seen as variations of the ”Zigzag” shuffling. The first, called the ”Christmas tree” decomposition, is obtained as follows:

- select a first reference frame (the Intra frame) placed at the median position in the GOP, where the median temporal index is $median = \lfloor (GOP_{size} + 1)/2 \rfloor$, and define the two parts separated by the median as sub-GOPs;
- repeat alternatively for each sub-GOP (for instance, by beginning with the left sub-GOP): use the frame closest to the median one as reference and remove it from the sub-GOP frame set.

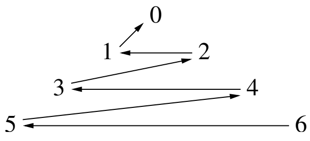

Fig. 8. Christmas Tree configuration, GOP size = 7.

The “Mirror” decomposition is obtained as follows:

- select a first reference frame (the Intra frame) and place it at the median position in the GOP, where the median index is $median = \lfloor (GOP_{size} + 1)/2 \rfloor$, and define the parts separated by the median frame as sub-GOPs
- repeat for each sub-GOP: use the frame closest to the median one as reference and define the set of remaining frames as a new sub-GOP.

Illustrated respectively in Fig. 8 and Fig. 9 for $N = 7$, these “Christmas tree” and “Mirror” configurations will provide better results in terms of compression as with them, each frame is at closer distance of its main reference than in “Zigzag”. However, this is obtained at the cost of a less efficient temporal scalability: indeed, if the last refinement levels are lost, the reconstructed sequence presents long frozen sub-sequences.

Note also that the “Mirror” configuration is somehow different from the “Zigzag” and “Christmas tree” ones in the sense that the two sub-GOPs on each side of the Intra frame are in fact independent from each other. Therefore, the “Mirror” configuration can be considered a first type of limited reference configurations, close to those that will be presented in Section III-C.

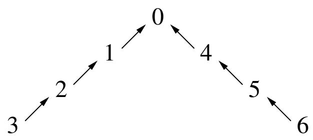

Fig. 9. Mirror configuration, GOP size = 7.

### B. Asymmetric filtering schemes

Let us now consider the case when two Intra frames are used for the prediction of the frames in a given GOP. This configuration ensuring the temporal scalability features, that we shall call “Dyad” configuration, is a regular repartition pattern that corresponds to a closed-loop $2^L$-band filterbank, as described in Section II, Fig. 4. It can be obtained as follows:

- select a reference frame (the Intra frame) placed at the extremity of the GOP (for instance, at the right extremity, when the other considered Intra frame is the one of the previous GOP), and define the set of remaining frames as sub-GOP;
- apply the “Zigzag” decomposition to the sub-GOP.

This is illustrated in Fig. 10 for $N = 16$. A generalization can here be done, following the generalization of the “Zigzag” decomposition principle. As an example, we give in Fig. 11 a decomposition pattern for $N = 15$.

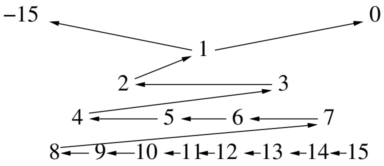

Fig. 10. “Dyad” configuration, GOP size = 16.

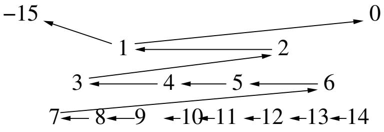

Fig. 11. Generalized “Dyad” configuration, GOP size = 15.

In this “Dyad” configuration, the dependency on the Intra frames is even more important, as any error in a frame at the first level leads to errors in two consecutive GOPs. In turn, the compression efficiency is better than that of the “Zigzag” decomposition, as the number of high quality references is higher.

### C. Limited references filtering schemes

Due to some practical limitations, coming either from the use of given levels [10] in the standard profiles or from practical implementation limitations, the configurations presented in the previous sections may not be realistic, as the codec may not be allowed to use up to 16 references in its prediction algorithm. As such, it becomes important to propose decompositions with a limited number of reference frames, this number being intimately linked to the total memory necessary to implement the encoding and decoding process. Naturally, such a limitation leads to some degradation in terms of compression efficiency, but it first meets the requirements of any level for any profile in H.264/MPEG-4 AVC (hence also any practical implementation), and second it ensures that the error propagation can be further reduced in erroneous environements.

Considering first the unidirectional schemes presented in Section III-A, the reduction of the number of reference frames can be done by imposing that a frame can only use reference frames from the upper levels. This "Tree" configuration (illustrated in Fig. 12 for $N = 15$ and in Fig. 13 for $N = 19$ and $R = 3$) is obtained as follows:

- apply the "Zigzag" decomposition to assign to each frame its corresponding level of refinement;
- repeat for each frame: choose as reference frame (or father) the closest one between those in the refinement level immediately above. When two frames can be equivalently chosen as reference, select the one that is the closest to its own father and so on. If no discrimination can be done, choose, for instance, the one closest to the Intra frame.

In this "Tree" configuration, the dependencies between frames are clearly reduced, which will be a major advantage in a noisy environment, as errors occuring at lower refinement levels will be less likely to propagate.

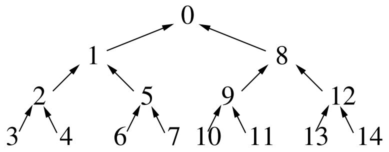

Fig. 12. Tree configuration, GOP size = 15.

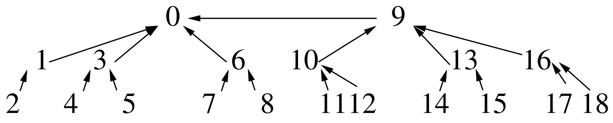

Fig. 13. Tree configuration, GOP size = 19.

Considering now the "Dyad" scheme and its generalization, as presented in Section III-B, the reduction of the number of references can be done similarly to the symmetric case by imposing again that a frame can only use reference frames from the upper levels. This "Limited Dyad" configuration is obtained as follows:

- apply the "Dyad" decomposition to assign to each frame its corresponding level of refinement
- repeat for each frame: choose as reference frame (or father) the closest one from those in the refinement level immediately above. When two frames can be equivalently chosen as reference, select the one that is the closest to its own father and so on. If no discrimination can be done, choose, for instance, the one closest to the Intra frame.

"Limited Dyad" configuration is illustrated in Fig. 14 and Fig. 15 for $N = 16$ and $N = 15$ respectively. The limitation on the number of reference frames clearly reduces the dependencies between frames, which will ensure that error propagation is limited in an error-prone environment. However, as for the other configurations, the loss of Intra frames will affect all the frames that reference them.

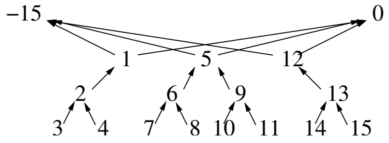

Fig. 14. "Limited Dyad" configuration, GOP size = 16.

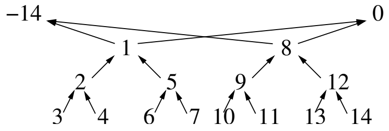

Fig. 15. "Limited Dyad" configuration, GOP size = 15.

## IV. Implementation details

The purpose of the schemes presented in the previous sections is to introduce temporal scalability within an a priori non-scalable configuration, such as the one provided by the H.264/MPEG-4 AVC codecs. The scheme shuffles the frames in a GOP to distribute them as regularly as possible.

The practical implementation of the different schemes presented in Section III is easily done in a standard compatible codec based on the consideration that two different frame numbering solutions do exist in the H264/MPEG-4 AVC standard. The first, *frame_num*, corresponds to the decoding order of access units, but does not necessarily indicate the final display order that the decoder will use. The second, POC or *Picture Order Count*, corresponds to the display order of the decoded frames (or fields) that will be used by the decoder for the display order. Considering now the number of reference frames to be used, here again the practical implementation is quite easily managed thanks, in the case of non-limited models, to the existence of a reference buffer of up to 16 different frames for any P-slice, and in the case of limited models, to the existence of memory management standardised functions that can be used to remove given frames from the reference buffer or mark given frames not to be used as reference. The only drawback of this scheme is that the shuffling operation introduces a delay and the necessity of frame buffering, both at the encoder and the decoder sides.

As presented in Section III, the most important frames, corresponding to those decoded from the lowest frame rates, can be regularly distributed along the timeframe. The intervals between those most important frames are then filled with less important ones, that are decoded only at higher frame rates. A temporal scalability enhanced H.264/MPEG-4 encoder can thus be implemented by first performing a re-arrangement of the frames according to their encoding order before the source encoder, and then by classical H.264/MPEG-4 AVC encoding.

The advantage of being able to define different scalable configurations at the encoder side, while not needing to transmit any supplementary information to the decoder or predefining said configuration during an initialisation phase, is that the chosen configuration can adapt either to the sequence actually being transmitted or to the channel conditions. As an example, transmitting over an erroneous channel may favor limited reference schemes, in order to avoid error propagation. Also, the choice of the frame shuffling pattern can be made based on GOP particularities. For instance, to better take into account some scene cuts, the frames corresponding to such changes will be coded with higher quality (the choice of the pattern will then be made such that they are placed at a low level of temporal resolution) and consequently ensure high rendering quality thanks to a better adaptivity of the codec. This GOP analysis and shuffling may lead however to a delay in sequence transmission, which needs to be compatible with the time constraints of the application. Finally, in a less adaptive mode, the choice of the configuration can be made based on the capabilities of the encoder and the decoder, in particular when they are implemented on low memory/CPU platforms. The simulation chain used to obtain and test the scalability features is presented in Fig. 16. The shuffling operation is applied directly on the video sequence to be encoded by means of an interleaver (denoted by Π in the figure), before the standard H.264/MPEG-4 AVC encoding process which is only modified to the extent of inserting knowledge of the used shuffling table, corresponding to the different scalability configurations presented, to permit the insertion of the correct display order values in the POC fields. The fully compliant H.264/MPEG-4 AVC codestream can then be sent over the transmission channel, which can be error-prone as in case of transmission over wireless links, or error-free, as in case of transmission with an efficient Forward Error Correction/Automatic Repeat-reQuest mechanism.

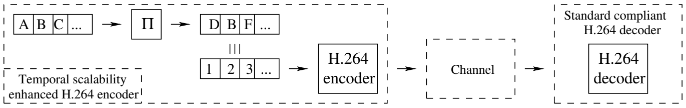
A
B
C
...
Π
D
B
F
...
|||
1
2
3
...
Temporal scalability enhanced H.264 encoder
H.264 encoder
Channel
Standard compliant H.264 decoder
H.264 decoder

Fig. 16. Simulation chain. The block denoted by Π corresponds to the interleaver.

## V. Simulation results

The simulations used the Joint verification Model (JM) version 8.4 [11], with some modifications to ensure that the number of frames that can be used as reference corresponds to the actual number of decomposition patterns. Indeed, some of the proposed patterns need more reference frames than the maximum number implemented by JM8.4.

The average PSNR values, derived as the average of MSE values over the whole sequence, for ’Foreman’, ’Mobile’ and ’Akiyo’ reference sequences (QCIF, 15 Hz, M=7) are given in Table I for different unidirectional decompositions and regular GOP sizes equal to 7 or $15$. In each case, the quantization parameters have been adjusted to yield a target bitrate of 64 kbs or 128 kbs.

It can be observed that the scalability feature is obtained in each case with small (less than 1% in the worst case, compared with the “mirror” configuration - at the tested bitrates, this is independent of the bitrate, but it may slightly depend on the sequence characteristics) or no quality degradation. This confirms the advantage of placing the Intra frame at the median position of the GOP (in the display order), which reduces the maximum distance between a predicted frame and its reference. By comparing the differences between different configurations (which are quite small), the advantage of choosing the configuration according to the actual transmission conditions, that is to say to adapt the configuration choice either to the transmission channel, to the sequence actually been transmitted or to the encoder or decoder capacities, as mentioned in Section IV, become obvious. Still, it can be observed that the “Tree” and “Mirror” configurations obtain here the best performance among all configurations. This can be partly explained by a particularity of the H.264/MPEG-4 AVC codec syntax, which relies on variable-length codes for indicating the considered reference frames. The “Tree” and “Mirror” patterns, ensuring that frames mostly use as reference the closest one in decoding order have then an advantage when compared to the others.

| Sequence | bit rate | configuration | Av. PSNR (dB) | |
|---|---|---|---|---|
| | | | GOP size 7 | GOP size 15 |
| Akiyo | 64 kbit/s | normal | 40.19 | 43.09 |
| Akiyo | 64 kbit/s | mirror | 40.41 | 43.36 |
| Akiyo | 64 kbit/s | christ. tree | 40.33 | 43.15 |
| Akiyo | 64 kbit/s | tree | 40.32 | 43.34 |
| Akiyo | 64 kbit/s | zigzag | 40.32 | 43.25 |
| Foreman | 64 kbit/s | normal | 32.19 | 33.35 |
| Foreman | 64 kbit/s | mirror | 32.54 | 33.71 |
| Foreman | 64 kbit/s | christ. tree | 32.36 | 33.38 |
| Foreman | 64 kbit/s | tree | 32.48 | 33.58 |
| Foreman | 64 kbit/s | zigzag | 32.28 | 33.28 |
| Mobile | 128 kbit/s | normal | 27.94 | 29.66 |
| Mobile | 128 kbit/s | mirror | 28.32 | 30.30 |
| Mobile | 128 kbit/s | christ. tree | 28.26 | 29.96 |
| Mobile | 128 kbit/s | tree | 28.30 | 30.12 |
| Mobile | 128 kbit/s | zigzag | 28.27 | 30.03 |

TABLE I

Average PSNR (over the entire sequence) at 64 Kbps and 128 Kbps for different configurations of the symmetric hierarchical subband tree H.264/MPEG-4 AVC codec for QCIF 15 Hz video sequences and GOP size $2^L - 1$.

Results obtained for the same three sequences and same target bitrates for different asymmetric decompositions and regular GOP size equal to $2^L$ are presented in Table II, where the average PSNR is computed over the entire sequence. Comparing the two asymmetric configurations, it can be observed that like for the symmetric ones, the limited version performs better than the “Dyad” one, based on “Zigzag” decomposition, for the same syntactical reasons. Based on this observation, we can now compare the results for a GOP size of $16$ with those obtained for symmetric decompositions and GOPs of size $15$. One can observe a quality gain of $0.1$ to $0.5$ dB. Yet, this is obtained at the cost of a higher dependency on the Intra frames, which again highlights the importance of choosing the hierarchical configuration in error prone environements according to the actual transmission conditions, and not only based on pure average PSNR considerations.

| Sequence | bit rate | configuration | Av. PSNR (dB) |
|---|---|---|---|
| | | | GOP size 16 |
| Akiyo | 64 kbit/s | Dyad | 43.59 |
| Akiyo | 64 kbit/s | Limit. dyad | 43.85 |
| Foreman | 64 kbit/s | Dyad | 33.58 |
| Foreman | 64 kbit/s | Limit. dyad | 33.79 |
| Mobile | 128 kbit/s | Dyad | 30.21 |
| Mobile | 128 kbit/s | Limit. dyad | 30.52 |

TABLE II
AVERAGE PSNR AT 64 KBPS AND 128 KBPS FOR THE DIFFERENT CONFIGURATIONS OF THE "DYAD" HIERARCHICAL SUBBAND TREE H.264/MPEG-4 AVC CODEC FOR QCIF 15 HZ VIDEO SEQUENCES AND GOP SIZE $2^L$.

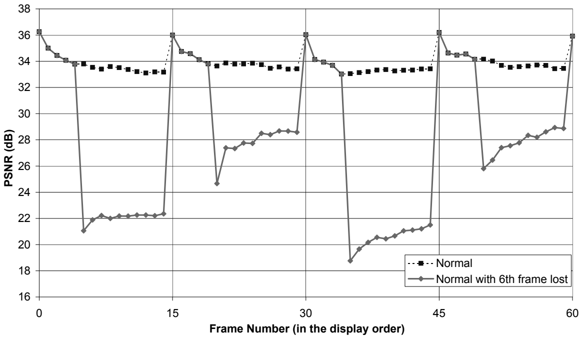

Fig. 17. 'Foreman' PSNR evolution for the "normal" configuration in an error-free and an erroneous environment (every 6th coded frame impaired). The $x$ axis indicates the frame number in the output (display) order.

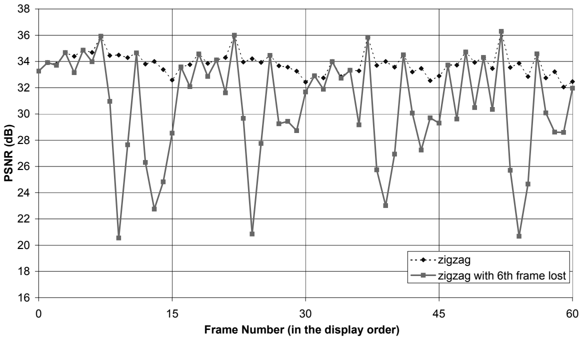

Fig. 18. 'Foreman' PSNR evolution for the "Zigzag" configuration in an error-free and an erroneous environment (every 6th coded frame impaired). The $x$ axis indicates the frame number in the output (display) order.

A second set of simulations have been conducted in an erroneous context, to observe the impact of transmission errors in various configurations. In our experiments, we selected a scenario where one frame in the GOP (the sixth frame when the first frame of the GOP is frame 0) is impaired (completely black) at the decoder.[4]

We observed the corresponding PSNR evolution of the whole GOP. Figs. 17, 18 and 19 present the PSNR evolution for 'Foreman' QCIF, 15Hz, 64 kbps, GOP size = 15 for "Normal", "Zigzag" and "Tree" configurations in the case of loss in information in the $6^{th}$ frame in the GOP (i.e., frame number $5$ in the encoding order) which appears in different scalable modes in the enhancement level. In these figures, the $x$ axis indicates the frame number in the output (display) order. As foreseen from the results presented in Table I, the three configurations have similar results in error-free environements. However, this changes greatly when errors occur. As a matter of fact, the degradation is quite noticeable for the "Normal" configuration, due to the error propagation from the erroneous frame to the end of the GOP. The "Zigzag" configuration presents the same number of frames affected by the error propagation, but the frame shuffling reduces the impact of errors, due to the fact that most of those frames are partly predicted from correct ones. Finally, the "Tree" configuration limits the error propagation to a small set of frames, which in counterpart are more deeply degraded, due to the fact that when compared to "Normal" case they rely on only a small set of reference frames, with one of their main reference being impaired.

We conducted informal subjective evaluation of the decoded sequences affected by frame loss. The corresponding visual results obtained for one entire GOP are presented in Fig. 20 for the "Normal" configuration, in Fig. 21 for the "Zigzag" one and in Fig. 22 for the "Tree" one. The degradation due to the impaired frame is clearly more annoying in the "Normal" case, as it leads to the degradation of the entire second part of the GOP, whereas it is quite acceptable in the "Zigzag" case, where the degradations are less distinguishable. They are even less important in the "Tree" case, where only three frames are degraded. The concealment in this later case is very easy, the impaired frame (6th in the display order) being restored by frame copy from its main reference (and, due to this, looking visually "good", even though it is not correct, i.e., it is not equivalent to the original frame) and the two frames depending on it (5th and 7th in the display order) being the only ones predicted from an erroneous frame (more sophisticated concealment techniques can also be applied). The PSNR results for these three frames are quite low, but visually (see Fig. 22) even the simple concealment technique we used (frame copy from the main reference) provides very satisfactory results.

Finally, let us illustrate the advantage of adapting the scalable pattern based on transmission conditions, as mentioned in Section IV for the case when a back channel is available. Considering the case of a wireless channel such as the GSM or UMTS ones, where errors often appear in bursts, the video transmission is confronted to time intervals when the channel is error free, and others where the channel is erroneous. In practice, based on simulation results presented in Tables I and II, the pattern recommended for error free channels can be

[4] A black frame at the decoder can be obtained if the NAL header is received, but it is incorrect. Such can happen when bit errors are present.

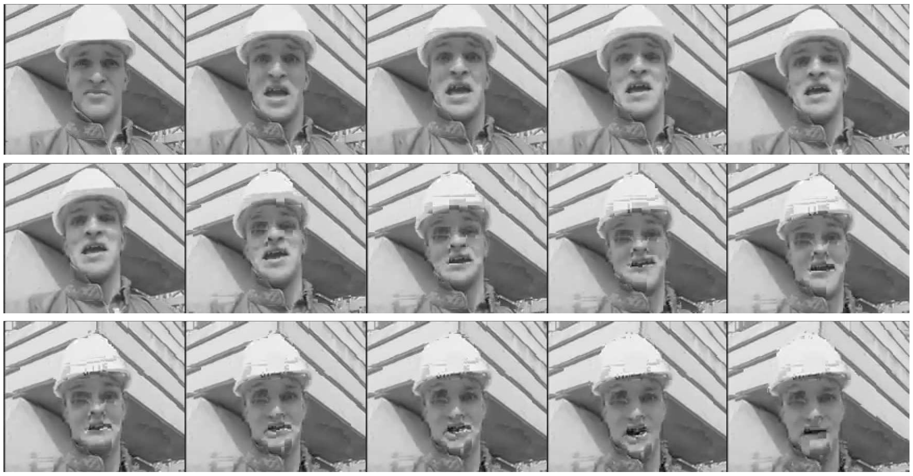

Fig. 20. Visual results over a GOP ($N$=15) in an erroneous environment for the “Normal” configuration, Foreman sequence (6th frame impaired).

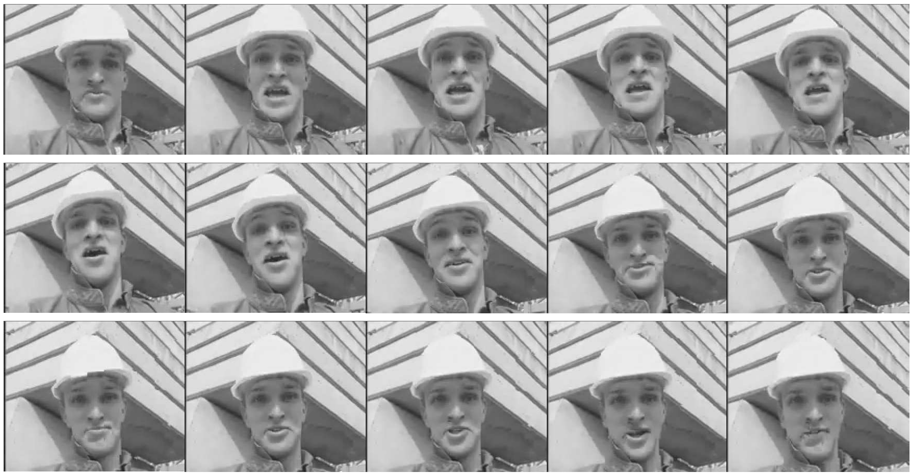

Fig. 21. Visual results over a GOP ($N = 15$) in an erroneous environment for the “Zigzag” configuration (6th frame impaired).

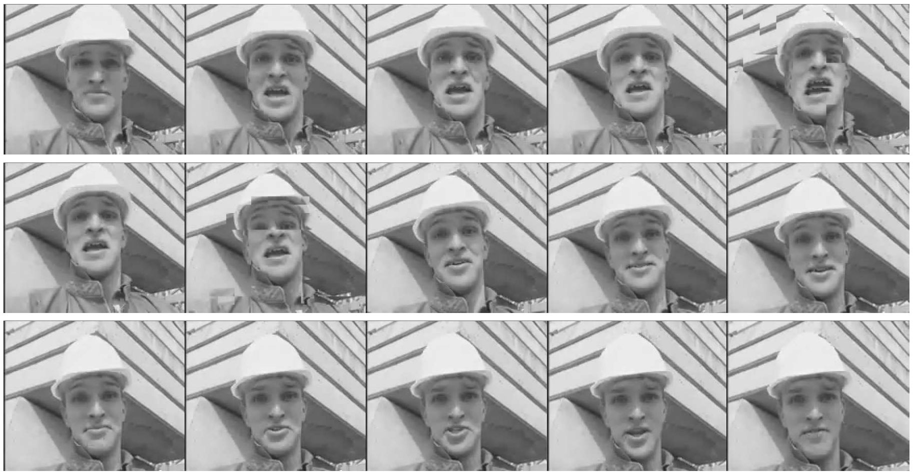

Fig. 22. Visual results over a GOP ($N = 15$) for the “Tree” configuration in an erroneous environment (6th frame impaired).

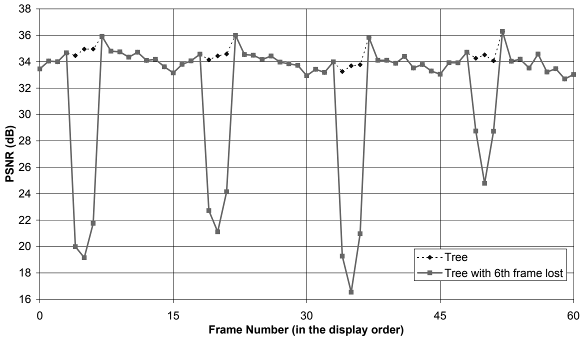

Fig. 19. ’Foreman’ PSNR evolution for the “Tree” configuration in an error-free and an erroneous environment (every 6th coded frame impaired). The $x$ axis indicates the frame number in the output (display) order.

“Limited Dyad” configuration, which offers the best PSNR of all configurations. Now, when considering noisy transmissions, the impact of loosing an Intra frame is more dramatic on bidirectional configurations as the Intra is used for prediction over two GOPs. As such, one can recommend the following efficient adaptation pattern:

- use “Limited Dyad” configuration by default
- when detecting at the receiver side that an Intra frame has been impaired, inform the encoding side by the back channel and suggest to select a symmetric configuration, for instance, “Tree”, and use it up until a sufficient number of frames has been received without errors.

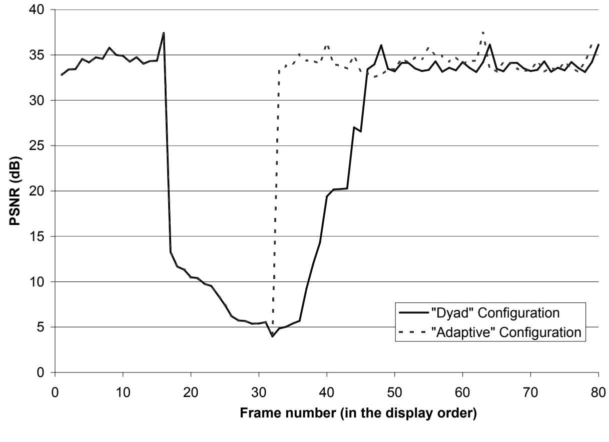

Fig. 23. ’Foreman’ PSNR evolution comparison in noisy environement: “Dyad” configuration vs. “Adaptive” mode.

Fig. 23 illustrates the results obtained when comparing the use of a non-adaptive “Limited Dyad” configuration over several GOPs, and the use of the adaptive method proposed above, where the Intra frame of the second GOP has been impaired (*i.e.* the $17^{th}$ frame in encoding order, and the $32^{th}$ one in decoding order). The advantage of going back for one GOP to “Tree” configuration is obvious, while “Limited Dyad” remains the best choice when the channel is error-free.

## VI. Conclusions

In this paper, we have introduced a general $M$-band filter-bank framework for adaptive motion-compensated temporal filtering and have shown how different temporal scalable solutions can be derived from it in a H.264/MPEG-4 AVC compliant manner. The proposed configurations have been compared in error-free and error-prone environments and the advantage provided by scalability in terms of robustness has been shown. By analysing the dependencies between frames in these configurations, one can predict not only the error propagation, but also the impact of the sequence features on the ability to perform error concealment.

## References

[1] *Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264, ISOIEC 14496-10 AVC), Doc JVT-G050r1*, Geneva, Switzerland, May 2003.

[2] *Registered Responses to the Call for Proposals on Scalable Video Coding, doc. m10569*, Munich, March 2004.

[3] L. Blaszak, M. Domanski, A. Luczak, and S. Mackowiak, “AVC video coders with spatial and temporal scalability,” in *Proc. of PCS’03*, Saint-Malo, France, April 2003, pp. 41–47.

[4] H. Schwarz, D. Marpe, and T. Wiegand, *Subband extension for H.264/AVC, Doc JVT-K023*, Munich, Germany, March 2004.

[5] J.-R. Ohm, *Multimedia Communication Technology*. Springer, 2004.

[6] G. Pau, C. Tillier, B. Pesquet-Popescu, and H. Heijmans, “Motion compensation and scalability in lifting-based video coding,” *EURASIP Signal Processing: Image Communication, special issue on Wavelet Video Coding*, pp. 577–600, August 2004.

[7] J. Xu, Z. Xiong, S. Li, and Y. Zhang, “Three-dimensional embedded subband coding with optimized truncation (3D-ESCOT),” *Applied Comp. Harmonic Analysis*, vol. 10, pp. 290–315, 2001.

[8] C. Bergeron, C. Lamy-Bergot, and B. Pesquet-Popescu, “Adaptive $M$-band hierarchical filter bank for compliant temporal scalability in H.264 standard,” in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP ’05)*, 2005.

[9] B. Pesquet-Popescu and V. Bottreau, “Three-dimensional lifting schemes for motion compensated video compression,” in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP ’01)*, vol. 3, Salt Lake City, UT, May 2001, pp. 1793–1796.

[10] A. Luthra and P. Topiwala, “Overview of the H.264/AVC video coding standard,” in *Applications of Digital Image Processing XXVI, Proceedings of the SPIE.*, A. Tescher, Ed., vol. 5203, 2003, pp. 417–431.

[11] *Joint verification model for H.264 (JM 8.4)*, http://iphome.hhi.de/suehring/tml, July 2004.