A FULLY SCALABLE 3D SUBBAND VIDEO CODEC

Vincent Bottreau, Marion Bénetière and Boris Felts[*]

Laboratoires d'Électronique Philips, Video and Communication Group

22, av. Descartes, 94453 Limeil-Brévannes, France

bottreau@philips.com, m.benetiere@philips.com, boris.felts@polytechnique.org

Béatrice Pesquet-Popescu[*]

École Nationale Supérieure des Télécommunications,

Dept. Signal and Image Processing

46, rue Barrault, 75634 Paris, France

pesquet@tsi.enst.fr

ABSTRACT

Multimedia transmissions over heterogeneous networks require a high degree of flexibility from video compression systems. They are expected to be fully scalable, that is to say to be able to partly decode a video bitstream and to get a reconstruction quality proportional to the received amount of information.

To achieve this functionality, we propose a video codec based on 2D+t subband decomposition. Groups of frames are first temporally filtered using motion compensation and then spatially decomposed with wavelets. The spatiotemporal coefficients are further scanned and compressed using a new *SPIHT*-like strategy, namely *Fully Scalable Zerotree* coding, together with arithmetic encoding, which provides a combination of temporal, spatial and SNR scalability. In addition, scalable motion vector coding ensures a fully progressive bitstream.

1. INTRODUCTION

With the recent expansion of multimedia applications, video coding systems are expected to become more flexible: in particular they should be able to adapt a single video bitstream to variable transport conditions (bandwidth, error rate…) and to varying receiver capabilities and demands (CPU, display size, application…) as well. S*calability* is the expected functionality to address this issue. The term *scalable* refers to methods that allow the partial decoding (or transmission) of a single compressed bitstream: depending on the conditions (bit rate, errors, resources), the decoder (or server) can take portions of the stream and decode (or send) the video sequence at different quality levels, spatial resolutions or framerates.

Current standards like H.263 or MPEG-4 are based on block DCT coding of Displaced Frame Differences (DFD). In these hybrid coders, scalability is implemented through additional layers of the single-scale prediction loop that delivers one *base* and one (or more) *enhancement* video bitstream (usually two spatial or temporal resolutions are then available). The proposed solutions are therefore not very granular except for the quality (or SNR) scalability provided in MPEG-4 by the Fine-Granular-Scalability (FGS) algorithm [1], where the decoding process can be stopped at any point of the enhancement layer. Temporal scalability is obtained at a reasonable cost by sending some of the B and P frames in the enhancement layer, whereas spatial and SNR scalable schemes have a very limited efficiency. Above all, the combination of the three types of scalability is still an issue in actual standards, although some contributions recently appeared [2].

Progressive encoding techniques based on subband decomposition answer the need for increasing resolution and fidelity by allowing a fully progressive transmission. Indeed, wavelets offer a natural multiscale representation for still images that has been extended to video by a 3D (or 2D+t) wavelet analysis including the temporal dimension within the decomposition. The natural hierarchy of these subband decompositions has been efficiently exploited by still image coding techniques such as the Embedded Zerotrees Wavelet Algorithm (EZW) [3] or Set Partitioning In Hierarchical Trees (SPIHT) [4], which take advantage of the dependencies existing along hierarchical spatiotemporal trees to yield the best compression performances, together with desirable properties like providing an embedded bitstream. These algorithms were successfully extended to 3D video coding systems to give some of the most effective SNR-scalable video coders, such as the 3D Set Partitioning In Hierarchical Trees (3D SPIHT) encoder [5]. However such approaches were only tailored for SNR-scalability. We propose here to adapt these coding principles to a fully scalable video coding scheme.

The paper is organized as follows: Section 2 gives an overview of the 3D codec. Section 3 presents the new coding method named Fully Scalable Zerotree coding (FSZ) and bitstream organization for full scalability. In Section 4 simulation results are shown and Section 5 concludes the paper.

2. CODEC OVERVIEW

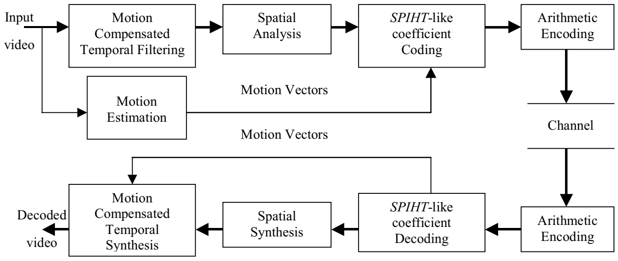


Figure 1: Overview of the codec structure.

[*] They contributed to the project when working at LEP in 2000.

0-7803-6725-1/01/$10.00 ©2001 IEEE

The 3D codec structure is presented in Figure 1.

2.1. Motion-Compensated Temporal Filtering

The proposed codec is based on a 2D+t wavelet analysis, consisting of Motion-Compensated Temporal Filtering (MCTF) and Spatial Filtering, which leads to a spatiotemporal multiresolution decomposition of the input Group Of Frames (GOF). Thus, lower spatial resolutions and/or lower framerates may naturally be obtained.

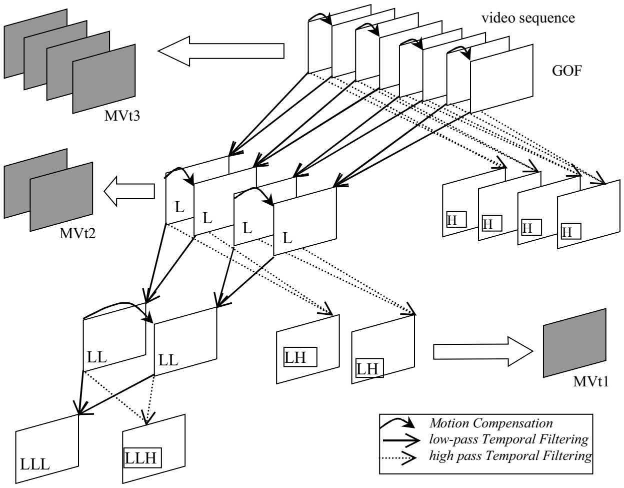


Figure 2: Temporal multiresolution analysis with motion compensation. L (resp., H) stands for the result of a low (resp., high)-pass filtering, while MVti designates the motion vector field at temporal level $i \in \{1,2,3\}$.

In this 3D subband decomposition scheme, the input video is first temporally filtered as shown in Figure 2. Each frame is considered as a temporal tap, leading to temporal subbands containing several frames. MCTF temporally filters the GOF in the motion direction [6] and results in a temporal decomposition tree in which the leaves (temporal subbands) contain several frames. These frames are further spatially decomposed and yield the spatiotemporal trees of wavelet coefficients.

In our approach Haar filters were used for TF on a GOF of 16 frames, which is a good trade-off between delay and energy compaction. With Haar filters motion estimation (ME) and motion compensation (MC) are only performed every two frames of the input sequence and the total number of ME/MC operations required for the whole temporal tree is roughly the same as in a predictive scheme (see Figure 2).

2.1.1. Motion Estimation

ME is realized by means of Block Matching Algorithms (BMA). Several implementations have been tested, namely Full-Search BMA, Hierarchical Search (performed on 2 spatial levels by means of image downsampling and motion vector refinement) and 3D-Recursive Search [7]. Fractional pixel displacements are obtained by bilinear interpolations. In our pixel-based MCTF, dense versions of the motion vector fields are used.

2.1.2. Connected pixels in MCTF

Unconnected and double connected pixels are closely related to covered and uncovered areas. For instance, two regions in the current frame may correspond, by MC, to the same (uncovered) region in the reference frame. This uncovered area appears as double-connected during TF. As in [8], these pixels are associated with the first block encountered during MC. Concerning unconnected pixels, the original value is inserted into the low-frequency temporal subband for the previous image and a DFD value is taken for the current image.

However, due to a lifting formulation of the TF [9], it is possible to choose among the pixels connected to the same pixel in the reference subband, the one that optimizes a given criterion.

2.2. Spatial Decomposition

Concerning the spatial wavelet analysis, there are fewer restrictions in the choice of the filter bank than for the temporal decomposition. However, the SPIHT algorithm relies on the hypothesis of orthonormal decomposition. Biorthogonal transforms may be employed with an appropriate renormalization of the energy in the subbands. We implemented and tested several classical filter banks. The filter bank choice is encoded and sent as side information in the bitstream. The implementation is made using the lifting technique (or *ladder* scheme).

With an appropriate design of the *predict* and *update* operators, it is possible to construct transforms that map integers to integers, for which there are no round-off errors and perfect reconstruction is possible when no additional quantization is performed [10].

3. CODING METHOD

3.1. SPIHT, principle and limitations

Our approach adopts the SPIHT compression principle by looking for zero-trees in the wavelet subbands in order to reduce redundancies between them. The wavelet coefficients are encoded according to their nature: *root* of a possible zero-tree or *insignificant set*, *insignificant* pixel and *significant* pixel.

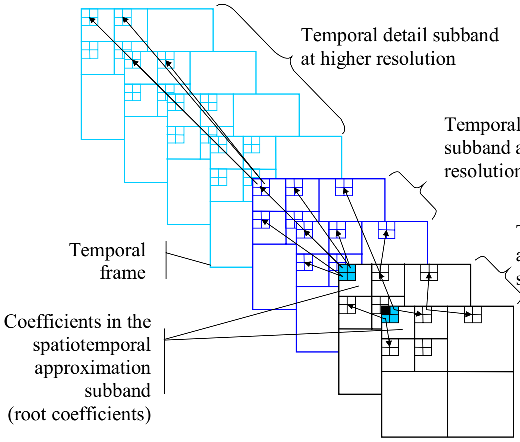


Figure 3: Examples of parent-offspring dependencies in the spatial temporal orientation tree (3D case).

In this approach, significance map and bitplane coding are combined. The significance map is efficiently encoded by exploiting the inter-subband correlations and the bitplane approach is retained to encode the refinement bits.

The SPIHT algorithm is mainly based on the management of three lists (List of Insignificant Sets, List of Insignificant Pixels and List of Significant Pixels). An iterative process of in-depth search successively scans and encodes the coefficients of each spatiotemporal tree (see Figure 3). The pixels belonging to the same 3D offspring tree but coming from different spatiotemporal subbands are therefore encoded and put one after the other in the lists, which has for effect to mix coefficients of different subbands. Thus, the neighborhood dependencies between pixels of the same subband are lost.

This technique is very efficient for SNR-scalability, but it is not suited for a fully scalable scheme. In order to preserve the potential spatial and temporal scalability allowed by the 3D wavelet analysis, the produced bitstream must preserve the former structure of the decomposition. It means that the algorithm must scan the subbands one by one by decreasing order of importance. Therefore, the coding strategy was modified so as to fulfill these new requirements.

3.2. Fully Scalable Zerotree coding

Though directly inspired from the SPIHT, FSZ preserves the initial subband structure of the 3D wavelet transform. The set-significance coding takes up the principles of the original SPIHT algorithm, but a subband scanning and a flag interpretation replace the list scanning. A flag is added to each coefficient in order to indicate which type of encoding will be performed on this coefficient: set-significance and/or pixel-significance. Moving a coefficient from a list to another becomes a simple change of flag. The interest of this “virtual moving” is to make the reading order independent from the changes performed by the logic of the SPIHT algorithm and consequently to deliver a fully scalable bitstream.

From the maximum to the last significance level, a full exploration of the spatiotemporal subbands is performed in an order that respects the parent-offspring relationships (see Section 3.3). Flags are updated for each coefficient according to original SPIHT *pixel and set significance* rules.

3.3. Scanning order

FSZ is very flexible and provides several progressive scanning modes. Indeed, the spatiotemporal volume of coefficients can be explored either by following its temporal, spatial or “diagonal” orientation. Thus three types of “multi-scalable” bitstreams may be obtained, one leaded by the spatial, the second by the temporal and the third being a hybrid version of the former two, increasing jointly the spatial and the temporal resolution.

We have favored here the temporal scalability. For each bitplane, the tree scanning is temporally oriented, since in this scheme the temporal resolutions are fully explored one after the other as shown in Figure 4. Inside each temporal scale, all the spatial resolutions are successively scanned and therefore all the spatial frequencies are available.

Inside each spatiotemporal subband, coefficients are scanned horizontally or vertically according to the direction of the details in the subband. This particular scanning order also facilitates the context determination used to encode the current pixel (see Section 3.6).

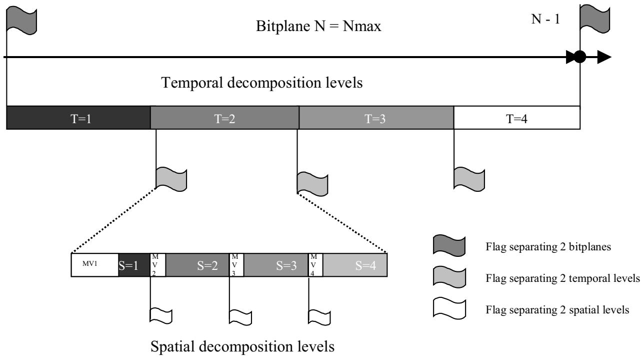

Bitplane N = Nmax
N - 1
Temporal decomposition levels
T=1
T=2
T=3
T=4
MV1
S=1
MV2
S=2
MV3
S=3
MV4
S=4
Spatial decomposition levels
Flag separating 2 bitplanes
Flag separating 2 temporal levels
Flag separating 2 spatial levels

Figure 4: bitstream organization when temporal ordered is privileged.

3.4. Scalable motion vectors

Encoding and sending all the motion vector fields for all spatiotemporal levels at the beginning of each GOF is not satisfactory with respect to scalability. Indeed, when a low bit rate is targeted and the receiver only wants a reduced framerate or spatial resolution, it is desirable to send or decode only the corresponding motion vectors. To this end, the hierarchy of temporal and spatial levels has been transposed to the motion vector coding as well.

Concerning temporal scalability, motion vector fields and associated high-frequency temporal subband frames are sequentially encoded and inserted into the bitstream for all temporal levels except the “base” one (see Figure 2).

Concerning spatial scalability, motion information is further adapted to each spatial level inside a particular temporal level. A low-resolution motion vector field corresponding to the lowest spatial resolution is first obtained by a simple integer divide-by-2 operation. It is then progressively refined and only the differences between motion resolutions need to be encoded and transmitted. The lowest spatial resolution motion vectors are encoded with a DPCM technique followed by entropy coding using VLC tables. For upper levels, refinement bits are encoded by contextual arithmetic encoding. As shown in Figure 4, motion vector fields are embedded inside spatiotemporal resolutions.

3.5. Scalability marker coding

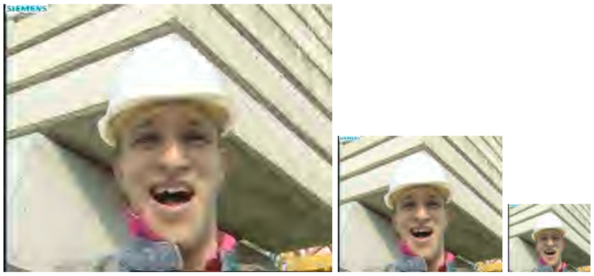
If the desired spatial resolution and/or framerate and/or quality are inferior to the ones provided by the encoder, the decoder (or server) must skip some parts of the bitstream. This is achieved by the introduction of special markers that indicate the end of a spatial level, a temporal level, a bit plane level or the end of a GOF. Due to the above-mentioned strategy, scalability is inherently associated to the bitstream organization and directly available at the decoder (or server) side, without any further processing.

3.6. Arithmetic encoding

In our approach, the probabilities of the previously sent binary symbols are re-estimated at each step of the FSZ coding loop and used to encode the current wavelet coefficients. This probability estimation step, which is crucial for the efficiency of the arithmetic coding, is performed using the very efficient Context-Tree Weighting method. It estimates the probability of the current bit according to the context given by neighboring coefficients. Several models have been specifically designed for each kind of information [11]. The evaluated contexts are all the more pertinent given that the proposed FSZ coding strategy preserves the pixel neighborhoods within subbands and therefore their coherence.

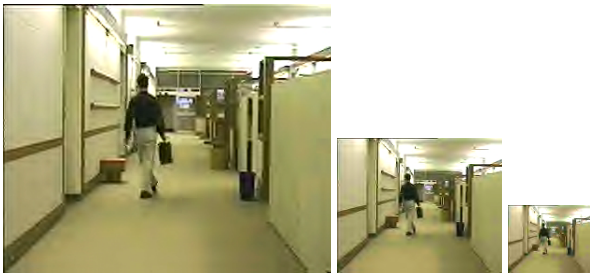
4. EXPERIMENTAL RESULTS

The tests presented below are performed on the CIF "*Foreman*" and "*Hall Monitor*" video sequences. A single bitstream has been encoded at 4 Mbit/s (bs) and 30 frame/s (fps). Five spatial and four temporal decomposition levels were used. To demonstrate how the three types of scalability can be combined, this bitstream has been decoded at various framerates, bitrates and display sizes (see Figure 5 and Figure 6).



Foreman	(a)	(b)	(c)
Average PSNR-Y (dB)	26.90	27.44	27.13
Average PSNR-U (dB)	35.78	36.16	33.71
Average PSNR-V (dB)	35.66	35.49	33.69

Figure 5: Foreman sequence decoded at (a) CIF-128 kbs-15 fps (b) QCIF-64 kbs-7.5 fps (c) $\frac{1}{4}$ QCIF-40 kbs-7.5fps.



Hall-Monitor	(a)	(b)	(c)
Average PSNR-Y (dB)	29.59	29.55	29.93
Average PSNR-U (dB)	37.38	36.86	35.54
Average PSNR-V (dB)	39.76	39.18	36.45

Figure 6: Hall-Monitor sequence decoded at (a) CIF-128 kbs-15 fps (b) QCIF-64 kbs-7.5 fps (c) $\frac{1}{4}$ QCIF-32 kbs-7.5 fps.

We must keep in mind that in the context of temporal scalability, PSNR calculation is an elusive issue: the original sequence to be compared to the reconstructed video is not directly available at lower framerate due to TF. The coding efficiency is therefore evaluated using as reference at each temporal resolution the approximation subband frames from the temporal analysis of the original sequence.

5. CONCLUSION

We have presented a fully scalable 3D subband video codec that outperforms all the standard solutions in terms of scalability functionality. Motion-Compensated Temporal Filtering followed by spatial wavelet decomposition lead to a spatiotemporal representation of the video signal. The 3D wavelet coefficients are encoded using a new *Fully Scalable Zerotree* compression strategy, where the list scanning of the original SPIHT was replaced by flag management. The original subband decomposition structure is thus preserved, which leads to an increased flexibility in the bitstream organization. This method, combined to the progressive transmission of motion vectors delivers a fully embedded bitstream providing temporal, spatial and SNR scalabilities at the same time.

Future work will focus on compression efficiency with the aim to make the single-scale version competitive with MPEG-4, and on demonstrating the capabilities of such an approach for a particular application exploiting scalability, such as video streaming or elastic storage.

6. REFERENCES

[1] H. Radha et al., "Scalable Internet Video Using MPEG-4", *Signal Processing: Image Communication*, vol. 15, no. 1-2, pp. 95-126, Sept. 99.

[2] Mihaela van der Schaar, "All FGS temporal-SNR-spatial scalability", Contribution to 54th MPEG Meeting, m6490, La Baule France, October 2000.

[3] J. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[4] A. Said and W.A. Pearlman, "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.

[5] Kim, Z. Xiong, W.A. Pearlman, "Low Bit-Rate Scalable Video Coding with 3D Set Partitioning in Hierarchical Trees (3D SPIHT)", *IEEE Trans. on Circuits and Systems for Video Technolog*., vol. 10, no. 8, pp. 1374-1387, Dec. 2000.

[6] J.-R. Ohm, "Three Dimensional Subband Coding with Motion Compensation", *IEEE Trans. on Image Processing*, vol. 3, no. 5, pp. 559-571, Sept. 1994.

[7] G. de Haan, P. W. A. C. Biezen, H. Huijgen and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 5, pp. 368-379, Oct. 1993.

[8] S.-J. Choi and J.W. Woods, "Motion-Compensated 3D Subband Coding of Video", *IEEE Trans. on Image Processing*, vol. 8, no. 2, pp. 155-167, Feb. 1999.

[9] B. Pesquet-Popescu and V. Bottreau, "Three-Dimensional Lifting Schemes for Motion Compensated Video Compression", *to be published in the Proc. of ICASSP 2001*.

[10] A.R. Calderbank, I. Daubechies, W. Sweldens and B.L. Yeo, "Wavelet Transforms that Map Integers to Integers", *Applied Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332-369, 1998.

[11] B. Felts and B. Pesquet-Popescu, "Efficient Context Modeling in scalable 3D wavelet-based video compression", *Proc. of IEEE Int. Conf. on Image Processing*, pp. 1004-1007, Vancouver, Canada, Sept. 10-13, 2000.