DELAY–PERFORMANCE TRADE-OFFS IN MOTION-COMPENSATED SCALABLE SUBBAND VIDEO COMPRESSION

[1]*Grégoire Pau,* [1]*Béatrice Pesquet-Popescu,* [2]*Mihaela van der Schaar and* [3]*Jérôme Viéron*

gpau@tsi.enst.fr
[1]GET-ENST, Signal and Image Proc. Dept., 46 rue Barrault, 75634 Paris, France
[2]Univ. of California Davis, Dept. of ECE, 3129 Kemper Hall, Davis, CA 95616-5294
[3]Thomson Multimédia, 35576 Cesson-Sévigné cedex, France

ABSTRACT

Scalable video coding based on motion-compensated spatio-temporal ($t + 2D$) wavelet decomposition is becoming increasingly popular, as it provides coding performance competitive with state-of-the-art codecs while accommodating varying network bandwidth and different receiver capabilities (e.g., frame-rate, display size, CPU, memory size). However, these temporal multiresolution schemes may introduce a non-negligible delay preventing their use by applications which require low latency or lightweight memory usage. In this paper, we provide a flexible approach to reduce the delay in motion-compensated temporal filtering schemes and illustrate the trade-offs between compression performance and low coding delay in this framework.

1. INTRODUCTION

Scalable video coding based on motion-compensated spatio-temporal ($t + 2D$) wavelet decomposition is becoming increasingly popular, as it provides coding performance competitive with state-of-the-art codecs while accommodating varying network bandwidth and different receiver capabilities (e.g., frame-rate, display size, CPU, memory size).

Since pioneering works exploiting the motion-compensated Haar transform [1, 2], several improvements have been brought to this scheme, improving the compression performance either by a finer analysis of the Haar structure [3, 4] or by introducing longer temporal filters inspired from the 5/3 biorthogonal ones [5, 6, 7]. Recently, non-dyadic multiresolution analyses have been introduced [8, 9], allowing even more flexible temporal scalability (by factors multiples of 3) and producing a succession of temporal frames comparable with the classical IBBP... hybrid coding. However, all these new schemes coming with additional functionalities and higher compression efficiency suffer from a common drawback: a more complex structure and a higher encoding-decoding delay.

On the other hand, there is a real need from applications requiring low latency or lightweight memory usage to design algorithms with low decoding delay like conversational applications or video surveillance ones. For example, in video surveillance applications, if an event is detected, it is mandatory to be able to transmit the video in the shortest time, so the encoding does not have to depend on future frames. Low encoding-decoding delay is also important in conversational applications, where very often the memory size of the device is limited.

In this paper, we provide a flexible approach to reduce the delay in motion-compensated temporal filtering (MCTF) schemes. If in terms of compression performance, bidirectional prediction and update operators involved in the scheme bring the main improvement, they are also responsible for the higher encoding-decoding delay. Indeed, the delay characteristics of the transform are related to the maximum number of *future* frames that need to be known and processed in order to be able to encode/decode the current frame. We show that the forward part of the update step of the underlying temporal filter is mostly concerned by this phenomenon, while stripping it in gradually way, from the coarsest temporal level to the finest, leads to a small and graceful degradation of the coding performance. The other factors involved in the delay analysis are the number of temporal resolution levels and the forward part of the predict operator. However, we show by simulation results that the penalty introduced by stripping the predict step or reducing the number of resolution levels is more dramatic.

This paper is organised as follows: in the next Section we review the existing MCTF schemes and the delay they introduce. In Section 3 we introduce a mechanism for reducing the decoding delay. Section 4 illustrates by simulation results the delay-performance trade-offs and we conclude in Section 5.

2. EXISTING SCHEMES AND DELAY ANALYSIS

The existing MCTF structures can be divided in two classes: dyadic multiresolution schemes, which will be analysed in subsection 2.1 and non-dyadic structures, which are mainly 3-band schemes [9].

2.1. Dyadic structures

Let us denote by x_t the original sequence frame at time t and by h_t, l_t the high-pass and low-pass subband frames. The first scheme that has been introduced in the lifting MCTF framework was the Haar one [4]. Even though we are dealing with motion-compensated lifting schemes, for the sake of simplicity we shall write in the sequel of this paper the corresponding filtering equations without taking into account the motion aspect. Then, up to the normalization constants, Haar lifting reads:

$$h_t = x_{2t+1} - x_{2t}$$
$$l_t = x_{2t} + \frac{1}{2}h_t.$$

If we denote by L the number of temporal decomposition levels, one can see that the encoding delay time is equal to $2^L - 1$ frames.
The 5/3 MCTF is the next most popular dyadic structure. In lifting form, it can be described by the following equations:

$$h_t = x_{2t+1} - \frac{1}{2}\big(x_{2t} + x_{2t+2}\big)$$
$$l_t = x_{2t} + \frac{1}{4}\big(h_{t-1} + h_t\big).$$

The "sliding window" filtering implementation of the temporal 5/3 decomposition (see Fig. 1) requires frames from adjacent Group of Frames (GOF) (previous and future) to perform the temporal filtering. For this dyadic temporal transform with bidirectional prediction and update operators, the encoding delay corresponds to $2^{L+1} - 2$ frames.

2.2. 3-band structures

The first 3-band MCTF scheme used for scalable encoding of video is the 3-band equivalent of the dyadic Haar MCTF. It has two detail subbands and a bidirectional update step. In lifting form, it reads [8]:

$$h_t^+ = x_{3t+1} - x_{3t}$$
$$h_t^- = x_{3t-1} - x_{3t}$$
$$l_t = x_{3t} + \frac{1}{4}\big(h_t^- + h_t^+\big).$$

Here, the encoding delay is directly related to the forward part of the update step. For L temporal resolution levels, it is of $(3^L - 1)/2$ frames.

In order to introduce bidirectional prediction, we have recently proposed [9] other MCTF schemes. The following equivalent of the 5/3 scheme has perfect reconstruction but does not however enter directly the lifting framework:

$$h_t^+ = x_{3t+1} - \frac{1}{2}\big(x_{3t} + x_{3t+2}\big)$$
$$h_t^- = x_{3t-1} - \frac{1}{2}\big(x_{3t} + x_{3t-2}\big)$$
$$l_t = x_{3t} + \frac{1}{4}\big(h_t^- + h_t^+\big).$$

The delay associated with this scheme increases due to the bidirectional predict operators and it becomes of $(3^{L+1} - 3)/2$ frames.
Other 3-band lifting schemes with bidirectional prediction are easy to construct, by predicting only based on already available polyphase components or computed subbands. For example, we can have:

$$h_t^+ = x_{3t+1} - \frac{1}{2}\big(x_{3t} + x_{3t+3}\big)$$
$$h_t^- = x_{3t-1} - \frac{1}{2}\big(x_{3t} + x_{3t-3}\big)$$
$$l_t = x_{3t} + \frac{1}{4}\big(h_t^- + h_t^+\big).$$

However, this scheme would increase even more the encoding delay. It will therefore not be considered for the further discussion.

3. REDUCING THE CODING DELAY

Among the simpliest techniques to reduce the delay of filters longer than Haar, entirely skipping the update step of the lifting was already investigated in another context [10]. Not doing the update step in the dyadic 5/3 MCTF (i.e. $l_t = x_{2t}$) leads to a reduced delay of 2^{L-1} frames, which is only related to the forward part of the prediction. In the 5/3-like 3-band MCTF, not doing the update step (i.e. $l_t = x_{3t}$) implies a delay of $2 \times 3^{L-1}$ frames.
However, as we have seen in the previous section, the delay characteristics of the transform are uniquely related to the maximum number of *future* frames that need to be known and processed in order to be able to encode/decode the current frame (see Fig. 1). In order to reduce this delay, it is therefore enough to strip in the coarsest levels the part of the update and predict operators corresponding to *future* frames, which have not yet been processed.
We can then build a new temporal transform based on a existing one where the K_p coarsest prediction steps are made causal and where the K_u coarsest update steps are also made causal, by stripping the part of these operators which corresponds to frames in the future.
In the case of the dyadic 5/3 MCTF temporal scheme described in the previous section, this leads to the following

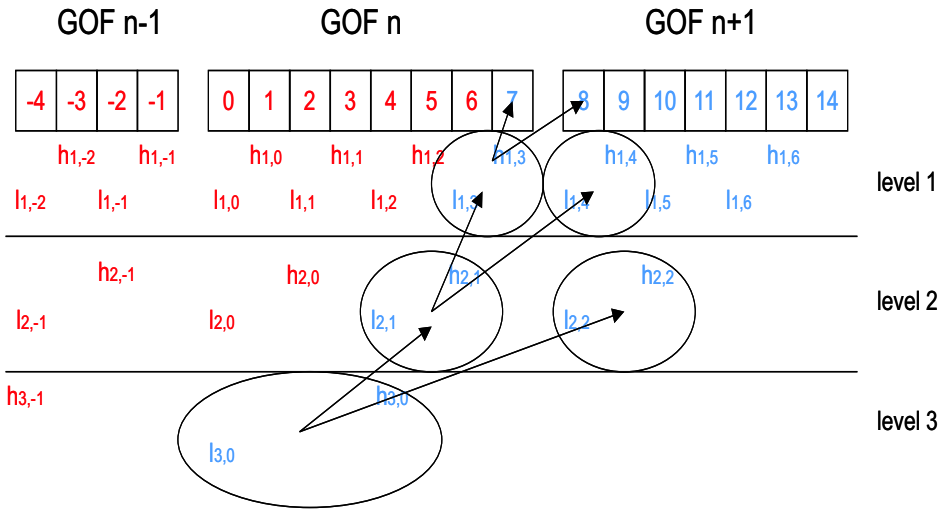

Figure 1: Illustration of the temporal filtering in the 5/3 dyadic structure over three decomposition levels: the computation of the third level approximation frame in n-th GOF requires prior processing of eight frames from the $n+1$-th GOF.

scheme, depending of the current temporal level j:

$$h_t = \begin{cases} x_{2t+1} - \frac{1}{2}\left(x_{2t} + x_{2t+2}\right) & \text{if } j < L - K_p \\ x_{2t+1} - x_{2t} & \text{otherwise} \end{cases}$$

$$l_t = \begin{cases} x_{2t} + \frac{1}{4}\left(h_{t-1} + h_t\right) & \text{if } j < L - K_u \\ x_{2t} + \frac{1}{2}h_{t-1} & \text{otherwise} \end{cases}$$

For this dyadic temporal transform with bidirectional prediction and update operators, the encoding delay corresponds now to $\max(2^{L-K_p-1}, 2^{L-K_u+1} - 2)$ frames. By playing on K_p and K_u, we can decrease the number of future frames and therefore lowering the delay required at the encoding side. Assuming that predict steps are considered to be more important in a compression point of view than update ones, we impose $K_p < K_u$. Tables 1 and 2 give the optimal number of levels K_p and K_u which have to be stripped off in order to fulfil a given delay constraint, expressed in number of frames N and in seconds at 30fps. Following the same idea, we can build an Haar MCTF which has a maximum coding delay of $2^{L-K_u} - 1$ frames. Concerning the 3-band structures, the mono-directional 3-band scheme will have accordingly a maximum delay of $\max\left(3^{L-K_p-1}, (3^{L-K_u} - 1)/2\right)$ frames. Moreover, the bidirectional 5/3-like 3-band MCTF will have a maximum delay of $\max\left(2 \times 3^{L-K_p-1}, (3^{L-K_u+1} - 3)/2\right)$ frames.

4. EXPERIMENTAL RESULTS

The implementation of the temporal filtering corresponds to a "sliding window" (or "on-the-fly") technique [7]. Motion estimation is done with the Hierarchical Variable Size Block Matching (HVSBM) algorithm with block sizes varying from 64×64 to 4×4 and with $1/8^{th}$ pixel accuracy.

N	Delay (s)	K_u	K_p
62	2.07	0	0
30	1	1	0
16	0.53	2	0
14	0.47	2	1
8	0.27	3	1
6	0.2	3	2
4	0.13	4	2
2	0.07	4	3
1	0.03	5	4

Table 1: Optimal K_u and K_p for a given constrained encoder delay, expressed in number of frames N and in seconds at 30fps. Number of temporal resolution levels: $L = 5$.

N	Delay (s)	K_u	K_p
30	1	0	0
14	0.47	1	0
8	0.27	2	0
6	0.2	2	1
4	0.13	3	1
2	0.07	3	2
1	0.03	4	3

Table 2: Optimal K_u and K_p for a given constrained encoder delay, expressed in number of frames N and in seconds at 30fps. Number of temporal resolution levels: $L = 4$.

Window search range is first initialized at $[-2; 2]$, is increased if no good match can be found and is doubled

at each temporal level. Motion vector fields are encoded as quad-tree maps and motion vector values are encoded with a 0-order arithmetic coder, in raster-scan order. Temporal subbands are then spatially decomposed over five resolution levels using biorthogonal 9/7 wavelets and the resulting spatio-temporal wavelet coefficients are encoded using the MC-EZBC [11] algorithm.

In our simulations, we consider two representative test color video sequences in CIF format at 30 fps: "Mobile" and "Foreman", which have been selected for their very different motion and texture characteristics.

Both 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 efficiency is expressed in terms of YSNR, calculated by averaging the Y component PSNR of all decoded frames.

We compare in Tables 3, 4, 5 and 6 the coding performance at several bitrates of the dyadic 5/3 MC filter bank with optimized prediction [12] with a constrained encoding delay. We have also included in these tables the coding performance achieved with the simple strategy of skipping all update steps (so-called NU).

N	Delay (s)	384	512	1024	2048
62	2.06	28.24	29.92	33.34	37.12
30	1.0	28.13	29.80	33.25	37.04
16	0.53	28.01	29.66	33.11	36.92
8	0.27	27.12	28.68	32.39	36.44
NU 16	0.53	27.45	29.01	32.40	36.34

Table 3: Rate-distortion YSNR (in dB) comparison of the 5/3 MC filter bank for several maximum coding delays (in number of frames N and in seconds at 30fps) and for several bitrates (in kbs) on "Mobile" sequence. Number of temporal resolution levels: $L = 5$. NU means no update at all.

N	Delay (s)	384	512	1024	2048
30	1.0	26.89	28.93	32.81	36.86
14	0.47	26.81	28.82	32.69	36.76
8	0.27	26.69	28.64	32.49	36.57
4	0.13	25.81	27.40	31.32	35.52
NU 8	0.27	26.33	28.15	31.95	36.04

Table 4: Rate-distortion YSNR (in dB) comparison of the 5/3 MC filter bank for several maximum coding delays (in number of frames N and in seconds at 30fps) and for several bitrates (in kbs) on "Mobile" sequence. Number of temporal resolution levels: $L = 4$. NU means no update at all.

As expected, we observe a graceful degradation of the coding performance depending of the maximum coding delay allowed at the encoder side. We notice that we can almost divide by 4 the encoding delay with a penalty of only 0.3 dB.

We also observe in Tables 3 and 4 that for a given delay constraint, significant better results (up to 1 dB) are obtained with a $L = 5$ levels temporal decomposition with a constrained delay approach than a regular $L = 4$ levels temporal decomposition. This is mainly due to the uniformity of the motion of the sequence "Mobile". In Tables 5 and 6 we remark on the contrary, that the results on "Foreman" are almost the same with a $L = 5$ or a $L = 4$ levels temporal decomposition. This is probably due to the complex rotational motions which prevent a smooth prediction in the highest temporal levels.

We also notice that for a given delay constraint, the update and predict strip strategy gives in every cases better results than skipping all update steps ("NU" cases) or than using a lower L levels temporal decomposition.

N	Delay (s)	384	512	1024
62	2.06	34.08	35.32	38.15
30	1.0	33.98	35.23	38.09
16	0.53	33.85	35.12	37.99
8	0.27	33.20	34.51	37.51
NU 16	0.53	33.59	34.80	37.65

Table 5: Rate-distortion YSNR (in dB) comparison of the 5/3 MC filter bank for several maximum coding delays (in number of frames N and in seconds at 30fps) and for several bitrates (in kbs) on "Foreman" sequence. Number of temporal resolution levels: $L = 5$. NU means no update at all.

N	Delay (s)	384	512	1024
30	1.0	33.80	35.17	38.15
14	0.47	33.70	35.07	38.06
8	0.27	33.55	34.93	37.93
4	0.13	32.56	33.85	37.00
NU 8	0.27	33.32	34.65	37.66

Table 6: Rate-distortion YSNR (in dB) comparison of the 5/3 MC filter bank for several maximum coding delays (in number of frames N and in seconds at 30fps) and for several bitrates (in kbs) on "Foreman" sequence. Number of temporal resolution levels: $L = 4$. NU means no update at all.

5. CONCLUSION AND PERSPECTIVES

In this paper, we have discussed a flexible approach for reducing the decoding delay in MCTF schemes. The forward part in the bidirectional update step was identified as responsible for the highest delay and furthermore the forward part of the prediction as introducing moderate to low delays. We have exhibited by simulation results a graceful degradation of performances from the most complex to the simplest scheme, with coding penalty by 0.1 - 1.2 dB when dividing the coding delay by factors of 2 up to nearly 8. In between these extreme cases, a large range of trade–offs exist and the temporal structure can be chosen to take into account the application requirements.
A possible extension of this work consists in performing an analysis step before encoding, allowing us to dynamically adapt the temporal decomposition depending on the sequence content.

6. REFERENCES

[1] 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.

[2] 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.

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

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

[5] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proc. of the IEEE Int. Conf. on Image Processing*, Thessaloniki, Greece, Oct. 2001, pp. 1029–1032.

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

[7] 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.

[8] C. Tillier and B. Pesquet-Popescu, "3D, 3-band, 3-tap temporal lifting for scalable video coding," in *Proc. of the IEEE Int. Conf. on Image Processing*, Barcelona, Spain, September 2003.

[9] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar, "3-band temporal lifting structures for scalable video coding," *submitted to IEEE Trans. on Image Proc.*, 2004.

[10] 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.

[11] S. Hsiang and J. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," *IEEE International Symposium on Circuits and Systems*, p. 589, 2000.

[12] G. Pau, C. Tillier, and B. Pesquet-Popescu, "Optimization of the predict operator in lifting-based motion compensated temporal filtering," in *Proc. of Visual Communications and Image Processing*, San Jose, CA, January 2004.