CONTENT-ADAPTIVE FILTERING IN THE UMCTF FRAMEWORK

Deepak S. Turaga and Mihaela van der Schaar
Wireless Communications and Networking
Philips Research USA
Briarcliff Manor, NY 10510
{deepak.turaga, mihaela.vanderschaar}@philips.com

ABSTRACT

Unconstrained motion compensated temporal filtering (UMCTF) is a very general and flexible framework for temporal filtering. It allows the selection of many different filters as well as decomposition structures to allow easy adaptation to video content, bandwidth variations, complexity requirements, and in conjunction with embedded coding can provide spatio-temporal-SNR scalability. In this paper we demonstrate the content-adaptive filter selection provided within the UMCTF framework. We show improvements in coding efficiency as well as in decoded visual quality using content-adaptive filters, at different granularities.

1. INTRODUCTION

Successful transmission of video over wireless networks requires efficient coding, as well as adaptability to varying video content, network conditions, device characteristics, and user preferences, while also being resilient to losses. Scalable coding approaches have been proposed within the predictive coding framework to increase its adaptability to network and device characteristics. Among these scalable coding techniques are the MPEG-4 spatial scalability and Fine Granular Scalability (FGS). More details on these scalability techniques may be obtained from [1]. Unlike predictive coding based scalable coders, wavelet video coding schemes can provide very flexible spatial, temporal, SNR and complexity scalability with fine granularity over a large range of bit-rates, while maintaining a high coding efficiency. Early contributions to the field of wavelet and multi-resolution video coding were provided, among others, by Taubman and Zakhor [2]. Some other 3D wavelet video coding schemes were proposed by Kim *et al* [3] and by Xu et al [4].

Motion compensated temporal filtering (MCTF) is used to remove temporal redundancies in wavelet based video coding schemes. MCTF was first proposed by Ohm [5] and later improved by Choi and Woods [6]. However, we believe that MCTF may also be used successfully in DCT based coding, and is not limited to wavelet video coding schemes. In this paper we view MCTF as a method that is applicable in both scenarios. In the conventional MCTF framework, successive pairs of frames are temporally filtered, in the direction of motion, using a two-channel Haar filter-bank. This results in the creation of low-pass (L) and high-pass (H) frames, thereby removing the short-term dependencies between successive frames. The long-term temporal dependencies are removed by further decomposing the L-frames using a pyramidal or multi-resolution decomposition structure. We show the MCTF decomposition structure in Figure 1. However, conventional MCTF schemes suffer from many problems such as low-efficiency temporal filtering, low quality temporal scalability, and increased delay. The lack of adaptivity to video content in conventional MCTF leads to many of these inefficiencies.

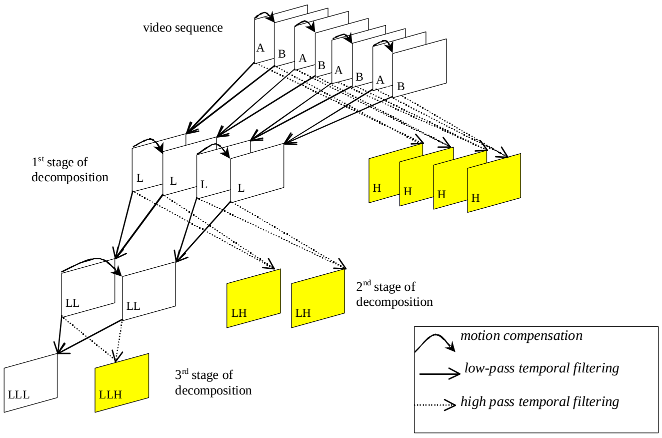
video sequence
A
B
A
B
A
B
A
B
1st stage of decomposition
L
L
L
L
H
H
H
H
LL
LL
LH
LH
2nd stage of decomposition
LLL
LLH
3rd stage of decomposition
motion compensation
low-pass temporal filtering
high pass temporal filtering

Figure 1. Motion compensated temporal filtering

In [7] we introduce a new framework for temporal filtering called Unconstrained-MCTF or (UMCTF). UMCTF involves designing temporal filters appropriately to enable greater flexibility in temporal scalability, while also improving coding efficiency by allowing greater adaptability to the video content. In this paper, we highlight the content-adaptive filtering allowed by the UMCTF framework and the gains it brings in both coding efficiency as well as visual quality.

This paper is organized as follows. We first describe briefly the general framework for UMCTF in Section 2, and summarize the flexibility it provides. We also include a discussion on the use of delta low-pass filters. We then describe the adaptive selection of low-pass filters, at different granularities, in Section 3. We show some results with coding efficiency gains and visual quality improvements in Section 4 and conclude in Section 5.

2. UNCONSTRAINED MCTF

We first briefly introduce some notation used in this paper. A more complete description may be obtained from [8].

2.1. Notation

N : Number of frames in GOF temporally filtered together;

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

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

A_i^d : Unfiltered frames at level $d \in [0, D]$, $i \le N^d - 1$

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

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

M^d : Number of successive H frames at level $d \in [0, D]$

0-7803-7663-3/03/$17.00 ©2003 IEEE

f_i^d : High-pass filter used to create H_i^d frames. $i \le N^d - 1$ (the filter taps are weights for the source and reference frames).

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

2.2. UMCTF framework

Conventional MCTF schemes use the Haar filter-bank for temporal filtering. However, this choice of filters leads to many inefficiencies. Instead, the UMCTF framework allows easy adaptability to video content, network or device characteristics by a simple choice of "controlling parameters". Some of these are summarized in Table 1.

Table 1. Adaptation parameters for UMCTF

Controlling Parameter	Adaptation Result
N	Changes Group Of Frames (GOF) size
D	Limits the number of temporal decomposition levels
M^d	Enables flexible temporal scalability; allow different decodable frame rates
R_p^d	Varies the number of reference frames used from the past; can be different at different levels
R_f^d	Varies the number of reference frames used from the future; can be different at different levels
g_i^d	Low-pass filter. Adaptively creates L frames with different characteristics; can be different at different levels
f_i^d	High-pass filter. Adaptively creates H frames. Changes the relative importance between reference and current frames, selects between available reference frames, can be different at different levels

This adaptivity is allowed at different granularities, block-by-block, or frame-by-frame etc., leading to very flexible temporal filtering. Through appropriate choice of filters and decomposition structures many different improvements to MCTF become possible. We have shown in [8] how sub-pixel accuracies, bi-directional prediction, multiple reference frames, etc., may easily be introduced into the MCTF framework. One simple choice of filters may be designed using delta low-pass filters and we discuss this briefly.

2.3. Delta low-pass filters

To provide this flexibility while achieving perfect reconstruction, a complicated design and implementation of filters is required. Alternatively, UMCTF may use a very simple set of filters, the delta low-pass filter, obtained by setting $g_i^d(j) = \delta(i - jM^{d-1})$, i.e. leaving low-pass frames unfiltered. Once this choice is made, the high-pass filters f_i^d may be designed without any constraints, to create H frames with the desired improvements, while guaranteeing perfect reconstruction. For instance, by appropriately choosing f_i^d ,we can perform sub-pixel accurate, bi-directional, multiple reference temporal filtering etc.

Note that by setting $g_i^d(j) = \delta(i - jM^{d-1})$, the effective motion estimation and compensation methods used in predictive coding can also be introduced in MCTF. Nevertheless, UMCTF with this filter choice differs significantly from predictive coding. Specifically, in UMCTF we retain the multiresolution decomposition structure in order to exploit both long term as well as short term temporal dependencies. Also, we use a non-recursive prediction structure and fully embedded coding, such that spatial and SNR scalabilities do not suffer from the drift problems occurring in predictive coding. Most importantly, the flexibility and features supported by UMCTF are unmatched in predictive coding or conventional MCTF. We can adaptively change the number of reference frames, the relative importance attached to each reference frame, the extent of bi-directional filtering etc.

3. CONTENT ADAPTIVE FILTERING

Delta low-pass filters enable many coding efficiency improvements in the MCTF framework. However, the low-pass frames created using these filters are not true temporal averages. Sometimes we can determine the true motion in the video sequence. In such cases, it is useful to create the L frames using non-delta low-pass filters g_i^d . This is important, especially while decoding at lower frame rates, as the L frames then represent the true temporal averages. We show an example in Figure 2.

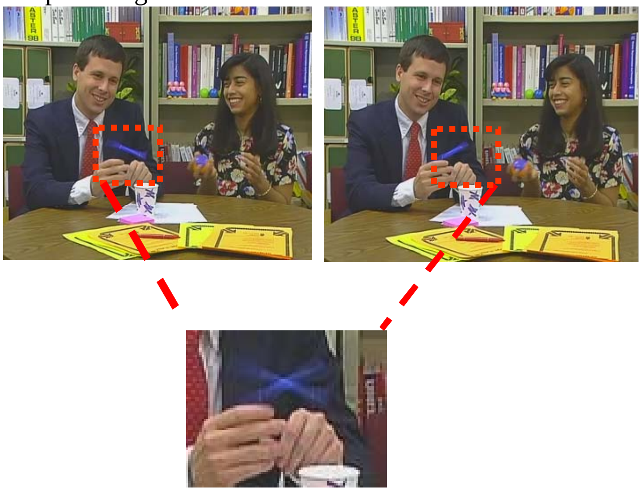

Figure 2. Example of temporal averaging

In the figure, we show two consecutive frames from the Paris sequence, on top. As may be seen, the region of interest consists of a pen being rapidly rotated. Clearly, without temporal averaging, the sequence decoded at half the frame rate cannot capture this rotation. In fact, when decoded at half the frame rate, the pen might appear to be static. At the bottom, we show the result of temporal averaging (low-pass filtering with a non-delta filter), focusing on the region of interest.

Clearly, the rotational motion of the pen is captured by the averaging. If now the video is decoded at half the frame rate, the rotation of the pen is still evident. This temporal averaging may thus lead to content that is more consistent with the filtering performed by our visual systems. Also, when true motion can be identified, averaging can smooth out some temporal artifacts, thereby increasing the coding efficiency. The above discussion holds when we can identify the true motion in the sequence.

Of course, during our pyramidal temporal decomposition, frames get farther apart at higher levels, thereby decreasing the likelihood of identifying true motion. In such cases, temporal averaging is likely to introduce artifacts, instead of removing them. In such cases we should use delta low-pass filters, both for efficiency as well as for improved visual quality. Hence, we need to adaptively choose between non-delta and delta low-pass filters, depending on the quality of the match, and the nature of motion in the sequence.

We show an example scenario, in Figure 3, of a pyramidal decomposition scheme with different low-pass filters chosen at different levels of the decomposition.

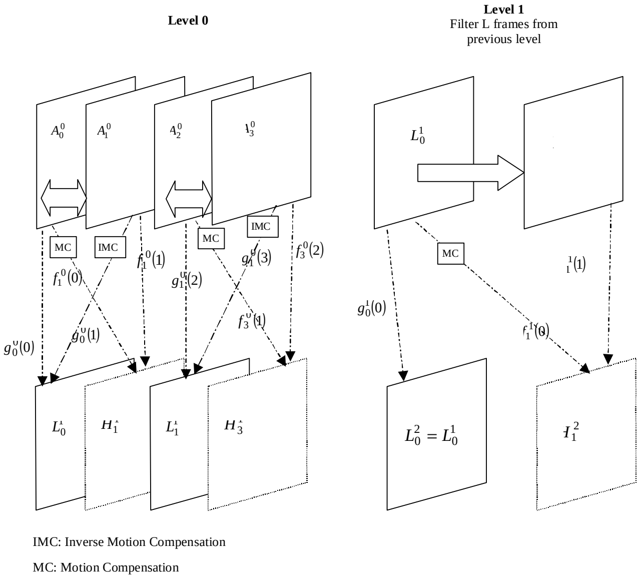

Figure 3. Different low-pass filters at multiple levels

In the figure, we have $N = 4$, $D = 2$, $M^0 = 2$, $M^1 = 2$, $R_p^0 = 1$, $R_f^0 = 0$, $R_p^1 = 1$ and choose a mixture of Haar-like low-pass filters and delta low-pass filters at different levels. Example values for the filter coefficients depicted in the figure are as follows:

$$g_0^0(0) = 1, g_0^0(1) = 1; \quad f_1^0(0) = -1, f_1^0(1) = 1 \;;$$
$$g_1^0(2) = 1, g_1^0(3) = 1; \quad f_3^0(2) = -1, f_3^0(3) = 1;$$
$$g_0^1(0) = 1; \quad f_1^1(0) = -1, f_1^1(1) = 1;$$

We can choose the high-pass filters f_i^d appropriately corresponding to the coding options that we wish to enable, such as bi-directional filtering etc. In the example scenario shown, we do not use multiple reference frames or bi-directional filtering at either of the two levels. Also, as mentioned before, we use delta low-pass filters at higher decomposition levels, due to the lower likelihood of finding true motion.

Importantly, the granularity at which we make this decision between the different low-pass filters may be varied. For instance, we may use a mixture of low-pass filters within one decomposition level, since within the same GOF, some frames may have very low and smooth motion, while others may have large and random motion. Similarly, the adaptive selection of different filters can also be performed at the block level. For instance, blocks that have very good matches may be averaged using Haar-like filters, while blocks that are poorly matched may be filtered using delta low-pass filters. We show an example of this in Figure 4.

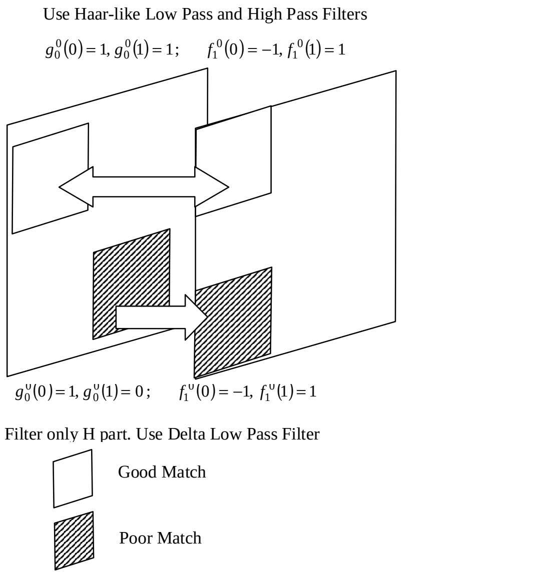
Use Haar-like Low Pass and High Pass Filters
$g_0^0(0) = 1, g_0^0(1) = 1;$ $f_1^0(0) = -1, f_1^0(1) = 1$
$g_0^0(0) = 1, g_0^0(1) = 0;$ $f_1^0(0) = -1, f_1^0(1) = 1$
Filter only H part. Use Delta Low Pass Filter
□ Good Match
▨ Poor Match

Figure 4. A and L regions in same frame

Whenever we have such adaptivity, we need to indicate the choice of filters to the decoder, so there is an overhead that increases with the increasing granularity at which this decision is made.

4. RESULTS

We present results for the adaptive filter choice at two different granularities. We first show results when this choice is made once for each temporal decomposition level, and then we show preliminary results when this choice is made on a block-by-block basis.

4.1. Filters chosen adaptively at different temporal decomposition levels

We use CIF sequences at 30Hz, with $N = 16$, $D = 4$, $M^d = 2$ for all levels *d*. We use full search with fixed block sizes of 16×16 and full pixel resolution. We use a mixture of the delta filters and Haar-like filters at different temporal decomposition levels. Hence we have two types of decomposition levels:

- Haar Level $R_p^d = 1$, $R_f^d = 0$ and $g_j^d\left(jM^{d-1}\right) = 1, g_j^d\left(jM^{d-1}+1\right) = 1$ with $f_k^d(k) = 1, f_k^d(k-1) = -1$
- Delta Level $R_p^d = N$, $R_f^d = 1$ and $g_i^d(j) = \delta\left(i - jM^{d-1}\right)$. Filter f_k^d is chosen for bi-directional and multiple reference filtering. $f_k^d(k) = 1$ and $f_k^d(j), f_k^d(k+1) \in \{-0.5, 0, -1\}$ with $f_k^d(j) + f_k^d(k+1) = -1$, where frame *j* provides the best match from the past for the current block of source frame *k*. All other coefficients are set to zero.

We use multiple reference and bi-directional filtering at Delta levels, but do not use them at Haar levels. D_{haar} and D_{delta} represent the number of Haar and delta levels, respectively, with $D_{haar} + D_{delta} = D$. As mentioned earlier, Haar-like filters are preferable whenever motion estimation provides good matches, and hence should be used at earlier decomposition levels than the delta filters. Thus, we create five different combinations of filter choices: $D_{haar} = 4$ (Levels 0, 1, 2 and 3), $D_{haar} = 3$ (Levels 0, 1, 2), $D_{haar} = 2$

(Levels 0, 1), $D_{haar} = 1$ (Level 0) and $D_{haar} = 0$. We use these different choices across two sequences, Coastguard and Stefan and plot the PSNR curves at different bit-rates. These results are generated using the EZBC [9] wavelet video codec, but as mentioned earlier the UMCTF method may also be used successfully with DCT based coding schemes. Also, on the same plots we combine the best filter choice results for each GOF, and plot them as one curve. We label this curve Best.

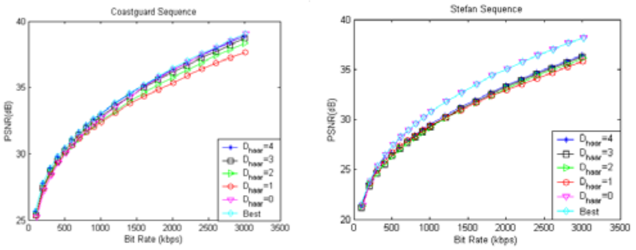

Figure 5. PSNR results for Coastguard and Stefan: Using five filter combination choices, and the best results

As expected, the Best curve forms the envelope of all the other rate-distortion curves. In order to highlight the content-dependent nature of the filter choices, we also show the histogram of the best filter choice, i.e. the percentage of times each of the five filter choices provided the best PSNR for a GOF.

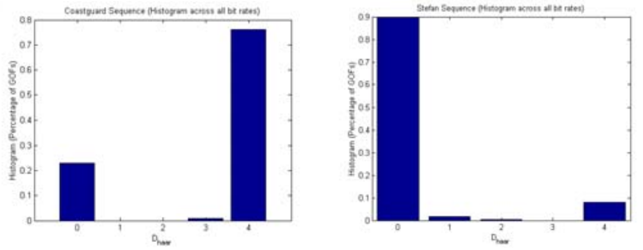

Figure 6. Histograms showing the percentage of GOFs that use one of the five filter combination choices.

These results clearly show the need for adaptivity in making the decision between different filter combinations. When the motion is low and correlated, as in Coastguard, we can find good matches, even at higher decomposition levels. Hence $D_{haar} = 4$ provides the highest PSNR for a majority of GOFs. However, when the motion is large, as in Stefan, the lack of good matches means that $D_{haar} = 0$ provides the highest PSNR for a majority of GOFs. Overall, this adaptive filter choice based on content characteristics led to up to 1.5dB coding efficiency improvement.

4.2. Adaptive filter choice within a frame

We also include some preliminary results on using a mixture of Delta and Haar-like filters adaptively, within a frame. This is done to avoid filtering across poorly matched blocks/regions. In order to make this decision adaptively, we use a similar mode decision proposed in MPEG-4 and H.263 to decide between intra and inter coding macroblocks. Since UMCTF can be used with DCT based schemes, we may reuse some of these previously proposed mode decisions. Of course, for optimality, the mode decision strategy needs to be modified specific to the transform used.

Visual artifacts in L frames are especially visible if we decode at lower temporal frame rates than the full rate. Hence to highlight the performance improvements due to the adaptive filter choice within a frame, we code the Foreman sequence at 30 Hz and decode it at 15 Hz. In terms of PSNR, we observe that the adaptive filter choice leads to an improvement of around 0.3-0.4 dB across different bit-rates. More importantly this adaptive filter choice can lead to significant reduction in temporal artifacts. We show some examples of large artifacts that can be corrected by filtering adaptively.

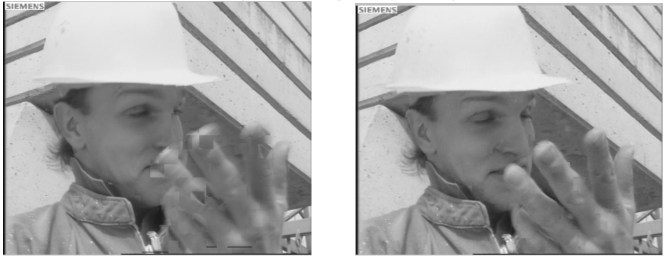

Figure 7. Sample frames with Haar-like filters (left) and with adaptive filter choice (right)

As can be seen from the figures, in frames with the adaptive filter choice, the visual artifacts are removed.

5. CONCLUSION

UMCTF provides a general and flexible framework that allows many enhancements to temporal filtering. Importantly, by appropriately choosing the UMCTF "controlling parameters" easy adaptation can be obtained to the desired video/network/device characteristics. In this paper we describe the content-adaptive filter selection possible in the UMCTF framework. We show how this adaptivity can increase both the coding efficiency, as well as the decoded visual quality. We show results with this adaptive filter selection made at different granularities; once at a temporal decomposition level or once at a block level. We show improvements in PSNR of up to 1.5 dB and also show some examples of improved visual quality.

REFERENCES

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

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

[3] B.-J Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D Set partitioning in Hierarchical Trees", *IEEE Trans. CSVT*, vol. 10, pp. 1374-1387, Dec. 2000.

[4] J. Xu, Z. Xiong, S. Li and Y.-Q Zhang, "Three-Dimensional Embedded Subband Coding with Optimized Truncation (3-D ESCOT)", *Applied and Computational Harmonic Analysis*, vol. 10, pp. 290–315, 2001.

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

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

[7] D. S. Turaga and M. van der Schaar, "Unconstrained motion compensated temporal filtering", MPEG Contrib.M8388, May 2002.

[8] M. van der Schaar and D. S. Turaga, "Unconstrained Motion Compensated Temporal Filtering (UMCTF) Framework for Wavelet Video Coding," submitted to *ICASSP 2003*.

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