SPATIAL AND TEMPORAL FILTERING IN A LOW-COST MPEG BIT-RATE TRANSCODER

Anthony Morel and Arnaud Bourge

Laboratoires d'Electronique Philips
94453 Limeil-Brévannes Cedex, France
{anthony.morel, arnaud.bourge}@philips.com

ABSTRACT

The MPEG bit-rate transcoder with motion compensation, MC-BRT, that exactly derives from a simplified synchronized decoder/encoder cascade, is considered. Such a transcoder is cost effective but lacks features such as spatial and temporal filtering. These features are known to improve the encoder performance at low bit-rates. Such features would be highly desirable in the MC-BRT since most applications now involve critical bit-rates. This paper shows that spatial and motion compensated temporal filtering can be implemented in the MC-BRT at a negligible cost. The filters are analyzed and control variables are given. Simulations reveal the efficiency of the method in terms of noise reduction as well as its great impact on image quality.

1. INTRODUCTION

Transcoding of MPEG bitstreams has been initially studied for a cable head end application. In that application, a high quality MPEG signal is inserted into a new multiplex for broadcast. The bit-rate of the MPEG signal is reduced as much as possible to decrease the broadcasting costs. In that context, Keesman et al. have derived a cost-effective transcoder that is equivalent to a decoder/encoder cascade provided the motion compensation information is kept [1]. The transcoder will be denoted as motion compensation bit-rate transcoder (MC-BRT).

Now, with the advent of home digital recording of MPEG broadcasts, the MC-BRT could be used in consumer devices to implement long-play modes or guarantee the recording time. However, the input signal is already encoded at a variable bit-rate with a low average rate. This is due to the generalization of statistical multiplexing that allows broadcasters to save the bandwidth and put more programs in a multiplex [2]. The transcoder, as is, is not adapted for low bit-rate applications. An alternative has been proposed where the quantizer scale is unchanged and high-order DCT coefficients are damped, which can be viewed as a kind of low-pass filtering [3].

In this paper, we started from encoders designed for low bit-rates. Such encoders perform noise reduction, so that bits are only spent on the useful information, using pre-filtering. We demonstrate that those features can be implemented in the MC-BRT at a negligible cost.

The paper is organized as follows. In Section 2, we show how to integrate temporal filtering in the MC-BRT structure. Then, in Section 3, we describe how to add spatial filtering to the scheme previously found. In Section 4, we present results. Finally, Section 5 concludes the paper.

2. TEMPORAL FILTERING

Temporal filtering allows reducing signals that are uncorrelated from frame to frame. It can very effectively reduce noise when combined with motion compensation, as motion compensation correlates the image content from frame to frame. This makes this processing suitable to improve the efficiency of subsequent encoders [4]. It is implemented using a recursive filter since it provides a better selectivity at lower costs.

The transcoding chain with a motion compensated recursive temporal filter is depicted in Figure 1, where D_1 corresponds to decoded pictures, D_f to pictures after filtering, and D_2 to decoded pictures after encoding. To reduce costs, the motion compensation in the encoder is reused in the recursive temporal filter. Thus, the signal D_2 is fed back instead of D_f. This trick has been successfully employed to embed both pre-processing and encoder on a single-chip [5].

2.1. Definition of the temporal filter

In the case of a forward motion compensated block, the filtering equation is

$$\begin{aligned} D_f(n,m) &= (1-\alpha)\, D_1(n,m) \\ &+ \alpha\, \mathrm{MC}\big(D_2(p(n)), V(n,m)\big), \end{aligned} \tag{1}$$

where n is the index of the current image, m is the index of a block, $V(n,m)$ is the motion associated with block m of

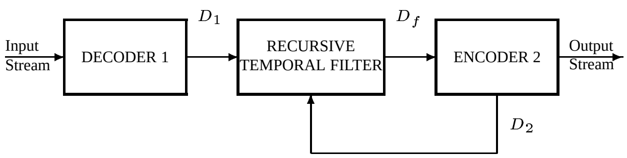

Figure 1: transcoding chain with temporal filter.

image n, $p(n)$ is the index of the anchor picture associated with image n, MC is the motion compensation operator, and α is a positive scalar smaller than one that tunes the filter response. Under the linear assumption, i.e., no motion and $D_2 = D_f$, the filter z-transfer function is

$$H(z = e^{jw}) = \frac{1-\alpha}{1-\alpha z^{-1}}. \quad (2)$$

An expression similar to Equation (1) can be drawn for bi-directional motion compensation. However, without loss of generality, we shall restrict the demonstration to the uni-directional case. Note that intra-coded blocks cannot be filtered since no prediction is formed for them.

The transcoding system depicted in Figure 1 can be simplified down to the complexity of an MC-BRT. The demonstration is threefold. First, we define notations and give the system equations according to the hypothesis of [1], i.e., the motion compensation information is unchanged. Then, we show that D_f does not need to be computed explicitly, which allows merging the filter and the encoder. Finally, we derive the merged decoder/encoder structure.

2.2. Decoding and encoding equations

In Decoder 1, for a motion compensated macroblock,

$$\begin{aligned} D_1(n,m) &= M^t R_1(n,m) M \\ &+ \mathrm{MC}\big(D_1(p(n)), V(n,m)\big), \end{aligned} \quad (3)$$

where M is the 8×8 discrete cosine transform matrix and $R_1(n,m)$ is the DCT domain residue retrieved from the input bit-stream after variable length decoding and dequantization. M is defined by Equation (4) and is such that $MM^t = I$:

$$M_{i,j} = \begin{cases} \sqrt{2}/4 & \text{if } i = 0, \\ \cos\big(\pi i(2j+1)/16\big)/2 & \text{otherwise.} \end{cases} \quad (4)$$

In Encoder 2, the filtered block is encoded using the same motion compensation information. Let $R_f(n,m)$ be the corresponding residue,

$$\begin{aligned} R_f(n,m) &= M D_f(n,m) M^t \\ &- M\,\mathrm{MC}\big(D_2(p(n)), V(n,m)\big) M^t. \end{aligned} \quad (5)$$

The residue is then quantized and dequantized again to compute the locally decoded pictures, D_2. Let $R_2(n,m)$ be the quantized and dequantized residue,

$$\begin{aligned} R_2(n,m) &= M D_2(n,m) M^t \\ &- M\,\mathrm{MC}\big(D_2(p(n)), V(n,m)\big) M^t. \end{aligned} \quad (6)$$

2.3. Temporal filter into Encoder 2

The filtering equation, Equation (1), and the encoding equation, Equation (5), can be combined so that R_f is derived directly from D_1 and D_2:

$$\begin{aligned} R_f(n,m) &= (1-\alpha)\Big[M D_1(n,m) M^t \\ &- M\,\mathrm{MC}\big(D_2(p(n)), V(n,m)\big) M^t\Big]. \end{aligned} \quad (7)$$

Thus, the filtered signal, D_f, does not need to be explicitly computed and it is possible to embed the recursive filter in the encoder.

2.4. Temporal filter into a MC-BRT

Combining the decoding equation, Equation (3), with Equation (7) gives:

$$\begin{aligned} R_f(n,m) &= (1-\alpha)\Big[R_1(n,m) \\ &+ M\,\mathrm{MC}\big(D_1(p(n)), V(n,m)\big) M^t \\ &- M\,\mathrm{MC}\big(D_2(p(n)), V(n,m)\big) M^t\Big]. \end{aligned} \quad (8)$$

Since motion compensation is performed identically from D_1 and from D_2, the MC operator can operate on the picture difference, i.e., on the error signal due to the transcoding operation. Defining $\delta D = D_1 - D_2$, Equation (8) can be rewritten as:

$$\begin{aligned} R_f(n,m) &= (1-\alpha)\Big[R_1(n,m) \\ &+ M\,\mathrm{MC}\big(\delta D(p(n)); V(n,m)\big) M^t\Big]. \end{aligned} \quad (9)$$

The error signal, δD, can be deduced from the prediction errors, combining Equations (6) and (7):

$$\delta D(n,m) = M^t \left[\frac{R_f(n,m)}{1-\alpha} - R_2(n,m)\right] M. \quad (10)$$

Equations (9) and (10) define the MC-BRT like transcoder structure depicted in Figure 2.

3. SPATIAL FILTERING

We have demonstrated that temporal filtering can be implemented in the MC-BRT; yet, intra-coded areas will not be processed. On the contrary, we show here that spatial filtering can be performed for the whole picture, as another mean to perform noise reduction.

The transcoding chain with spatial filtering is depicted in Figure 3. As the decoded pictures, D_1, are not available in the transcoder shown in Figure 2, we are going to define a pixel domain filtering that can be related to a process on residues, R_1.

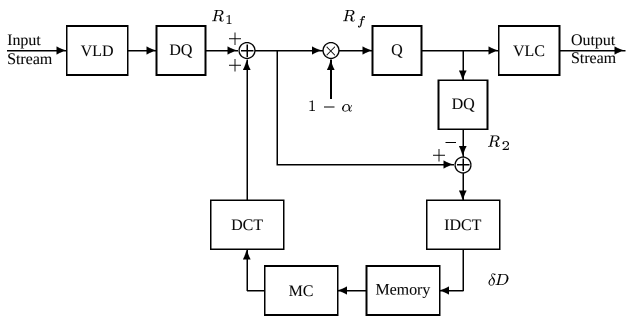

Figure 2: transcoder (MC-BRT) with temporal filter.

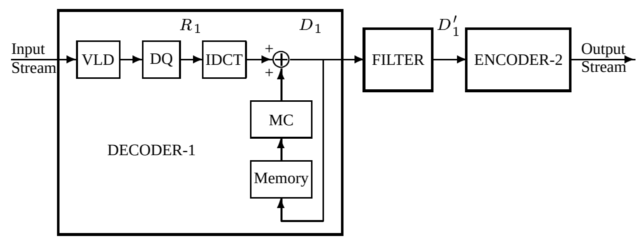

Figure 3: transcoding chain with spatial filter.

3.1. Definition of the spatial filter

The pixel domain filter shall have the same granularity as the decoder's granularity; thus we consider a block-wise filter. Let $D_1(n,m)$ be block m of picture n. The filtered block $D'_1(n,m)$ is computed using

$$D'_1(n,m) \quad = \quad F_v(n) D_1(n,m) F_h^t(n), \qquad (11)$$

where $F_v(n)$ and $F_h(n)$ are matrices that define respectively the vertical and horizontal filtering within the block.

3.2. Spatial filter into Decoder 1

Combining the filtering equation, Equation (11), with the decoding equation, Equation (3), we find:

$$\begin{aligned} D'_1(n,m) = & \; F_v(n) M^t R_1(n,m) M F_h^t(n) \\ & + F_v(n) \mathrm{MC}\big(D_1(p(n)), V(n,m)\big) F_h^t(n). \end{aligned} \qquad (12)$$

If the filter is constant for a group of picture, then $F_v(n) = F_v(p(n))$ and $F_h(n) = F_h(p(n))$. Thus, we can give the following approximation for Equation (12) based on the assumption that block-wise filtering commutes with motion compensation:

$$\begin{aligned} D'_1(n,m) \quad = & \quad F_v(n) M^t R_1(n,m) M F_h^t(n) \\ + & \quad \mathrm{MC}\big(D'_1(p(n)), V(n,m)\big). \end{aligned} \qquad (13)$$

It follows that the block-wise filter can be applied to residue $R_1(n,m)$ after IDCT.

3.3. Spatial filter into a MC-BRT

To implement the spatial filter in the MC-BRT, we need to substitute $R_1(n,m)$ by

$$R'_1(n,m) = M F_v(n) M^t \, R_1(n,m) \, M F_h^t(n) M^t. \qquad (14)$$

Even if the matrices $MF_v(n)M^t$ and $MF_h^t(n)M^t$ can be pre-computed, the expression seems to involve many operations. Actually, the expression can be simplified for a class of block-wise filters for which the two matrices are diagonal. Such filters are symmetric filters with an even number of taps [6]. In our implementation, we consider normalized 3-tap symmetric filters since they are more suitable for small blocks. Such filters have a single parameter, denoted b. The corresponding pixel domain filtering matrix, $(F_{i,j})_{0 \le i,j < 8}$, is defined by

$$F_{i,j} \quad = \quad \frac{1}{2+b} \begin{cases} b & \text{for } i = j = 1 \ldots 6, \\ 1 & \text{for } i = j \pm 1, \\ 1+b & \text{for } i = j = 0 \text{ and } 7, \\ 0 & \text{otherwise.} \end{cases} \qquad (15)$$

Then,

$$M F_{i,j} M^t = \frac{1}{2+b} \begin{cases} 2\cos(i\pi/8) + b & \text{for } i = j, \\ 0 & \text{otherwise.} \end{cases} \qquad (16)$$

Thus, to implement filtering with horizontal parameter b_h and vertical parameter b_v, $R_1(n,m)$ in Figure 2 needs to be weighted (component-wise) by $(W_{i,j})_{0 \le i,j < 8}$ defined as follows:

$$W_{i,j} \quad = \quad \frac{2\cos(i\pi/8) + b_v}{2 + b_v} \, \frac{2\cos(j\pi/8) + b_h}{2 + b_h}. \qquad (17)$$

This leads to the transcoder depicted in Figure 4.

4. SIMULATION RESULTS

The transcoder with spatial and temporal filters was tested on a set of broadcast sequences. We expected temporal filtering to be more efficient than spatial filtering. Yet, noise is especially visible in areas where the useful signal is weak. Such areas are often intra-coded, as they cannot be efficiently predicted. Therefore, in this case, spatial filtering is our only tool to reduce noise.

Setting the strength of the filters through α and b, a compromise had to be found between noise reduction and undesirable signal degradation. It appeared that $1 - \alpha = 0.8$ and $b = 8$ lead to an optimum average PSNR gain for most noisy sequences. On the other hand we could notice a loss, both in PSNR and picture quality, for noiseless sequences using these settings. Thus, an adaptive filtering is required

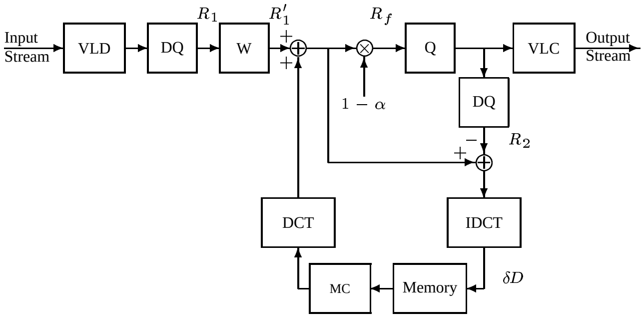

Figure 4: transcoder with spatial and temporal filters.

to adjust the noise reduction as a function of the incoming stream.

In Figure 5, we show the performance of a constant filtering (with the above settings) and of an adaptive scheme that simply switches the filters on and off based on noise detection. Our reference is the MC-BRT without filters. In term of PSNR, both perform better on noisy sequences (Movie1 and Movie2). The adaptive scheme prevents possible PSNR degradation on the other sequences. Nine video experts of LEP were asked to quantify the improvement due to filtering in the range $[-2, \ldots, +2]$ at viewing distance $4H$. The subjective results, shown in Figure 6, confirm the objective measurements.

5. CONCLUSION

We have proposed a low-cost MPEG transcoder structure that features spatial and temporal filtering. This structure is equivalent to an MPEG transcoding chain, including a decoder, a filter, and an encoder, with the following assumptions: the block-based spatial filter commutes with motion compensation and the recursive temporal filter uses motion compensated pictures from the encoder.

Filtering allows performing noise reduction. This can improve the subjective quality of incoming pictures. It also reduces the amount of information to be encoded. As a consequence, the transcoder behavior is improved at low bit-rates.

Great picture quality enhancement was obtained with a different blend of parameters for different sequences and bit-rates. Thus, adaptivity is required to adjust the filter strength to the incoming noise level.

The authors would like to acknowledge F. Brüls and R. Klein Gunnewiek for their work, which contributed to this paper.

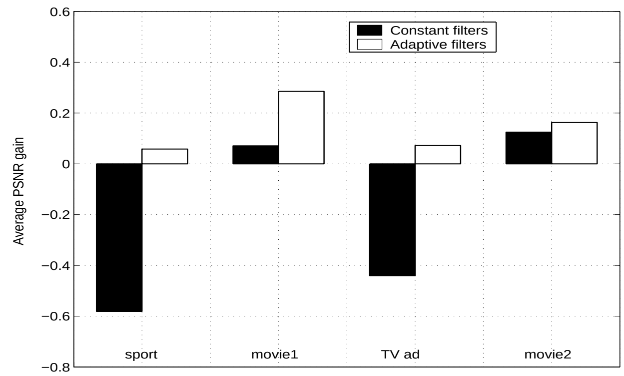

Figure 5: average PSNR gain with filtering.

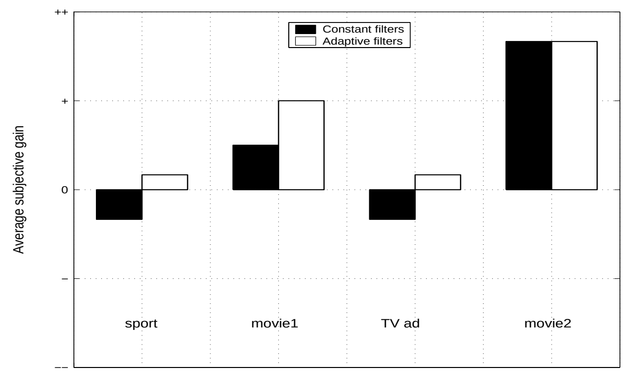

Figure 6: subjective evaluation.

6. REFERENCES

[1] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG-2 bitstreams," *Signal Processing: Image Communication*, vol. 8, pp. 481–500, September 1996.

[2] G. Keesman and D. Elias, "Analysis of joint bit-rate control in multi-program image coding," *Visual Communications and Image Processing 1994*, vol. SPIE-2308, pp. 1906–1917, 1994.

[3] R. Klein Gunnewiek, F. Brüls, B. Hunneman, and M. Holl, "A low-complexity MPEG-2 bit-rate transcoding algorithm," *IEEE Int. Conf. on Consumer Electronics*, pp. 316–317, June 2000.

[4] E. Dubois and S. Sabri, "Noise reduction in images sequences using motion-compensated temporal filtering," *IEEE Trans. on Communications*, vol. COM-32, no. 7, pp. 826–831, July 1984.

[5] A. van der Werf, F. Brüls, R. P. Kheihorst, E. Waterlander, M. J. W. Verstraelen, and T. Friedrich, "I.McIC: A single-chip MPEG-2 video encoder for storage," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, pp. 1817–1823, November 1997.

[6] H.-Y. Kim and R. Meyer, "DCT domain filter for ATV down conversion," *IEEE Trans. on Consumer Electronics*, vol. 4, pp. 1074–1078, November 1997.