FAST IN-BAND MOTION ESTIMATION WITH VARIABLE SIZE BLOCK MATCHING

Davide Maestroni, Augusto Sarti, Marco Tagliasacchi, Stefano Tubaro

Dipartimento di Elettronica e Informazione - Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano – Italy

ABSTRACT

In this paper we propose a fast motion estimation technique that works in the wavelet domain. The computational cost of the algorithm turns out to be proportional to the linear size of the search window instead of its area. We complete our proposal with a variable-size block-matching scheme in the wavelet domain. We integrated the motion estimation algorithm in a fully-scalable wavelet in-band prediction coder inspired by the IB-MCTF (In-Band Motion Compensation Temporal Filtering) proposed in [1]. Comparative tests prove that our coder provides the same quality level than IB-MCTF at a very reduced computational cost. Moreover, although our method turns out to match the performance of MCTF-EZBC [2] in terms of PSNR, it clearly outperforms it in terms of perceptual quality, as it is completely free from blocking artifacts.

1. INTRODUCTION

The wavelet transform has proven to be an effective tool when applied to image and video coding. Compared to conventional DPCM/DCT coders, the main benefit of wavelet-based video coders is the natural support for rate, spatial, and temporal scalability. This feature makes the same bitstream suitable for heterogeneous devices and networks, as it allows us to avoid to encode and to store in advance several versions of the same video sequence. If we focus our attention on 3D wavelet based video coders, we can identify two distinct families:

- inter-frame wavelet coders
- in-band prediction coders

Inter-frame wavelet coders [3] perform wavelet filtering along the temporal axis first. Usually a simple Haar filter is used in this step. Each pair of frames produces a low frequency (*tL*) and a high frequency (tH) temporal subband. Temporal filtering is then iterated on the resulting *tL* frames, until only one *tL* frame per GOP is obtained. Such a process provides a rather good energy compaction, since most of the significant coefficients are located in the *tL* frame. The leaves of the temporal pyramid are then filtered with a 2D DWT, quantized and coded using one of the techniques available for still image coding (JPEG2000, EZW, SPITH, EZBC). In order to fully exploit the temporal redundancy, temporal filtering is performed along motion trajectories (MCTF – Motion Compensated Temporal Filtering), which justifies the need of a motion estimation module. On the other hand, if a conventional block-based algorithm is used in the motion estimation phase, then low-bitrate decoded sequences end up suffering from significant blocking artifacts, as *tH* frames exhibit energy peaks along block boundaries. This is due to the fact that at low bitrates most of the bit budget is spent to encode the motion vectors. Moreover, block-based motion compensated temporal filters tend to produce data with relevant high frequency components whenever the motion model is unable to accurately describe the true motion.

In-band wavelet coders [1] perform 2D spatial wavelet filtering first, while motion estimation is computed in the wavelet domain. One remarkable feature of these solutions is that, although these methods use a block-based motion estimation algorithm, no blocking artifacts are visible in the reconstructed sequence, even at very low bitrates. The main drawback of this approach, however, is that the motion estimation performance is limited by the fact that the wavelet transform is not shift-invariant. Because of this problem, an overcomplete DWT (ODWT) is performed on the reference frame, which results in a higher computational complexity and in more demanding memory requirements. This solution, also known as the IB-MCTF [1], is able to offer scalability features without generating blocking artifacts. Our work builds upon the IB-MCTF method, and its innovative features are: a fast algorithm to perform motion estimation in the wavelet domain; a variable-size block matching strategy.

2. OVERVIEW OF THE IB-MCTF CODER

In this section we briefly review the IB-MCTF codec, emphasizing the computational cost associated with the motion estimation phase. Let us consider a pair of consecutive frames. Both frames are wavelet

This work was developed within the FIRB-VICOM project (www.vicom-project.it) funded by the Italian Ministry of University and Scientific Research (MIUR); and within the VISNET project, a European Network of Excellence (www.visnet-noe.org)

transformed, up to a given number of levels L. Let us select a block in the spatial domain. The wavelet coefficients that correspond to the selected spatial location are distributed throughout the subbands and they constitute the so-called wavelet block (see Figure 2).

For each wavelet block, we search in the reference frame for the wavelet block that matches the considered one the best, using a full-search approach. Unfortunately, the wavelet transform is not shift-invariant, and the wavelet-transformed reference frame cannot be used for matching purposes. In order to overcome this issue, [4] proposes to produce an overcomplete version of the wavelet transform (ODWT), using the *low-band-shift* technique. It can be shown that the low-band-shift method is equivalent to the algorithm *à trous*, described in [5]. This algorithm, in fact, is simpler to implement and more intuitive. Its pictorial representation for a three levels wavelet decomposition of a 1D signal is shown in Figure 1. It's worth pointing out that each overcomplete subband has the same size as the input signal.

The current frame is DWT filtered, while an ODWT is performed on the reference frame. For each wavelet block k, a motion vector $(dx,dy)_k$ is computed as the one which yields the lowest MAD (Mean Absolute Difference), adopting a full search approach. The computational complexity (no. of operations required) turns out to be:

$$T = 2W^2N^2$$

where W is the size of the search window and N is the block size. The residual image, which is coded and sent in the wavelet domain, still exhibits singularities along block boundaries. Nevertheless, when it is transformed back to the spatial domain, the inverse wavelet transform filters out such singularities so that a human observer will not be able to perceive any blocking artifacts, even at very low bitrates.

3. FAST IN-BAND MOTION ESTIMATION

The proposed fast motion estimation algorithm lays its foundations on optical flow estimation techniques. The optical flow equation, based upon the brightness constraint along the motion trajectory states that:

$$I_x v_x + I_y v_y + I_t = 0$$

where I_x, I_y and I_t are the horizontal, vertical and temporal gradients, respectively. Notice that when $I_x >> I_y$, i.e. when the local texturing is almost vertically oriented, only the dx ($dx = v_x dt$) component can be accurately estimated because

$$\frac{I_x v_x}{I_x} + \frac{I_y v_y}{I_x} + \frac{I_t}{I_x} \approx v_x + \frac{I_t}{I_x} = 0$$

This is the result of the so-called "aperture problem", which consists of the fact that when the observation window is too small, we can only estimate the optical flow component that is parallel to the local gradient. That of the aperture is indeed a problem for traditional motion estimation methods, but what we do in our motion estimation algorithm is to try to turn it to our advantage.

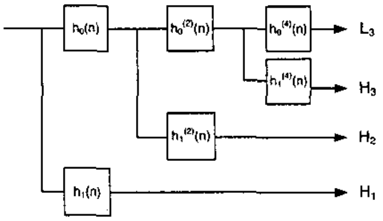

Figure 1 - Overcomplete DWT (ODWT) computed through the algorithm à trous. $h^{(k)}$(n) is the dilated version of h(n) obtained inserting k-1 zeros between two consecutive samples.

For the sake of clarity, let us consider a pair of images that exhibit a limited displacement between corresponding elements, and let us focus on the HL subband only (before subsampling) This subband is low-pass filtered along the vertical axis and high-pass filtered along the horizontal axis. The output of this separable filter looks like the spatial horizontal gradient I_x. In fact the HL subbands tend to preserve only those details that are oriented along the vertical direction. This suggests us that the family of HL subbands, all sharing the same orientation, could be used to accurately estimate the dx motion vector component. Similarly, LH subbands have details oriented along the horizontal axis, therefore they are suitable for computing the dy component. We are now ready to summarize the details of our algorithm. For each wavelet block, a coarse full search is applied to subband LL$^{(L)}$ only, where L is the number of the considered DWT decomposition levels. This initial computation allows us to determine a good starting point (dx^{FS}, dy^{FS}) for the fast search algorithm, which reduces the risk of getting trapped into a local minimum. As the LL$^{(L)}$ subband has 2^L fewer samples than the whole wavelet block, block matching is not computationally expensive. In fact, the computational complexity of this initial step expressed in terms of the number of additions and multiplications turns out to be

$$T = 2 \cdot \left(\frac{W}{2^L}\right)^2 \left(\frac{N}{2^L}\right)^2 = \frac{W^2N^2}{2^{4L-1}}$$

At this point we can focus on the HL subbands. In fact, we use a block matching process on these subbands in order to compute the horizontal displacements and estimate the dx component. The search window is reduced to *W/4*, as we only need to refine the coarse estimate provided by the full search. In formulas:

$$dx_k = \arg\min \sum_{i=1}^{L} \sum_{x_i = x_{i,k}}^{x_{i,k}+N/2^i} \sum_{y_i = y_{i,k}}^{y_{i,k}+N/2^i} \left| \begin{matrix} HL_{cur}^{(i)}(x_i, y_i) - \\ HL_{ref^O}^{(i)}(2^i x_i + dx^{FS} + dx_k, 2^i y_i + dy^{FS}) \end{matrix} \right| +$$

$$\sum_{x_l = x_{L,k}}^{x_{L,k}+N/2^L} \sum_{y_i = y_{L,k}}^{y_{L,k}+N/2^L} \left| LL_{cur}^{(L)}(x_L, y_L) - LL_{ref^O}^{(L)}(2^L x_L + dx^{FS} + dx_k, 2^L y_L + dy^{FS}) \right|$$

Because of the shift-varying behavior of the wavelet transform, block matching is performed considering the

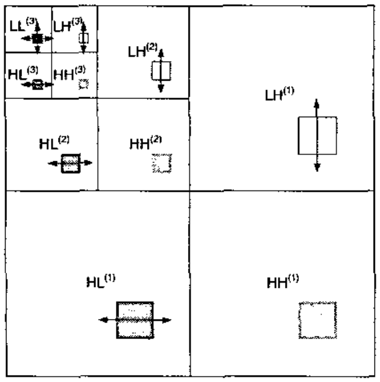

Figure 2 - The subbands belonging to the same wavelet block are used to estimate the different components of the motion vector according to their orientation

overcomplete DWT of the reference frame.

Similarly we can work on the LH subbands in order to estimate the *dy* component. In order to improve the accuracy of the estimate, this second stage takes $(x_{i,k}+ dx^{FS} + dx_{k}, y_{i,k} + dx^{FS})$ as a starting point, where $(x_{i,k},y_{i,k})$ are the coordinates of the top-left corner of the block. We refer to this algorithm for motion estimation as Fast In-Band Motion Estimation (FIBME). Figure 2 gives a pictorial representation of the algorithm.

The algorithm achieves a good solution that compares favourably with respect to a full search approach with a modest computational effort. The computational complexity of this method turns out to be

$$T = 2 \cdot \frac{2}{3} \cdot \frac{W}{4} N^2 + \frac{W^2N^2}{2^{4L-1}} = \frac{1}{3} WN^2 + \frac{W^2N^2}{2^{4L-1}}$$

where W/4 comparisons are required to compute the horizontal component and other W/4 for the vertical component. Each comparison involves either HL or LH subbands, whose size is approximately one third of the whole wavelet block (if we neglect the LL subband). If we keep the block size *N* fixed, the proposed algorithm runs in linear time with the search window size, while the complexity of a full search grows with the square power. The speedup with respect to the exhaustive full search turns out to be, for *L* greater than 2:

$$Speedup = \frac{2W^2N^2}{\frac{1}{3}WN^2 + \frac{W^2N^2}{2^{4L-1}}} \approx 6W$$

We have investigated the accuracy of our search algorithm in case of large displacement. If we do not use a full search for the LL$^{(L)}$ subband, our approach tends to give a bad estimate of the horizontal component when the vertical displacement is too large. In this scenario, when the search window scrolls horizontally it cannot match the reference displaced block. We observed that the maximum allowed vertical displacement is approximately as large as the low-pass filter impulse response used by the critically sampled DWT. This is due to the fact that such filter operates along the vertical direction by stretching the details proportionally to its impulse response extension.

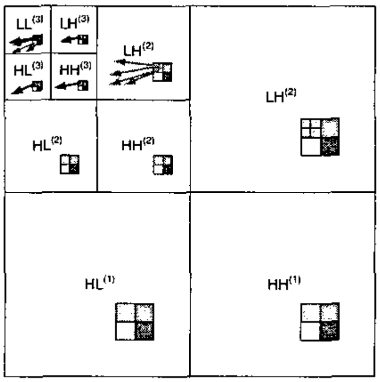

Figure 3 - Variable size block matching. Three splitting iteration are performed (light black, bold and gray vectors respectively).

4. VARIABLE SIZE BLOCK MATCHING

In [1] block matching is performed on blocks of fixed size. Conversely, MCTF-EZBC, that is one of the state-of-the-art scalable video coders, takes advantage of a Hierarchical Variable-Size Block-Matching algorithm (HVSBM) [6] in the spatial domain. Our variable-size block-matching algorithm is built upon HVSBM but it works directly in the wavelet domain. Let us consider a three-level wavelet filtering and a wavelet block of size 16x16 (refer to Figure 3). If we focus on the lowest frequency subband, the wavelet block covers a 2x2 pixel area, when $L = 3$. Splitting this area into four and taking the descendants of each element, we generate four 8x8 wavelet blocks, which are the offspring of the 16x16 parent block. This step is closely related to splitting in the spatial domain. Block matching is performed on those smaller wavelet blocks to estimate four distinct motion vectors. This simple splitting strategy can be iterated further until the child wavelet block has only one pixel in the LL$^{(L)}$ subband, i.e. only once in our example. In order to provide a motion field of finer granularity, we can still assign a new motion vector to each subband LH$^{(L)}$, HL$^{(L)}$, HH$^{(L)}$, plus the refined version of LL$^{(L)}$ alone. This way we produce four motion vector children, shown in Figure 3 as bold arrows. In this case the motion vector shown in subband HL$^{(3)}$ is the same one used for compensating all of the coefficients in subbands HL$^{(3)}$, HL$^{(2)}$ and HL$^{(1)}$ that appear in light grey. The same figure shows a further splitting step performed on the light grey wavelet block of the LH$^{(3)}$ subband. In fact, the splitting can be iterated at lower scales, by assigning one motion vector to each one-pixel sub-block at level L-1 (in subband LH$^{(2)}$ in this example). The wavelet block with roots on the light grey pixel in subband $LH^{(3)}$, is split into four sub-blocks in the $LH^{(2)}$ subband. These refinement steps allow us to compensate elements in different subbands that correspond to the same spatial location with different motion vectors. Once the smallest block size is reached, the motion vector tree is processed by a pruning algorithm inspired to [6], which merges those blocks that are not worth being split. Following this procedure, we end up with sub-blocks of arbitrary size in the wavelet domain.

5. EXPERIMENTAL RESULTS

We implemented a fully-scalable video coder that integrates the new features described in this paper, namely the fast search in the wavelet domain and the variable size block matching. Wavelet coefficients are coded using EZBC, thus reflecting the same choices as MCTF-EZBC. Figure 4 shows the average PSNR for the luminance component, comparing three codec implementations: MCTF-EZBC, IB-MCTF – full search, fixed size block matching, (our implementation of the coder presented in [r]) and IB-MCTF – FIBME – variable size block matching. Similar results are obtained for all the tested sequences. Notice that our coder produces approximately the same results as MCTF-EZBC in terms of objective quality, while it pays a very little penalty when compared with the original IB-MCTF, which adopts a more demanding full-search approach. Subjective inspection demonstrates that our coder produces a reconstructed sequence that is free from block-artifacts even at very low bitrates. If we turn our attention to computational complexity, our coder runs much faster than IB-MCTF, which adopts a full-search approach. According to our implementation of IB-MCTF, the overall speedup of the coder integrating FIBME with respect to the one that uses the full search is between 13 to 15 on all test sequences. Our algorithm is even faster than MCTF-EZBC of approximately 30% during the encoding phase while MCTF-EZBC runs faster during the decoding, as it does not have the burden of the IDWT-ODWT computation and our fast matching algorithm affects the encoding phase only.

Variable size block matching allows to gain up to +0.6dB with respect to fixed size block matching. It should be emphasized that the effectiveness of this feature depends on the complexity of the scene to be encoded. The more complex the motion, the highest the benefit of using blocks of variable size.

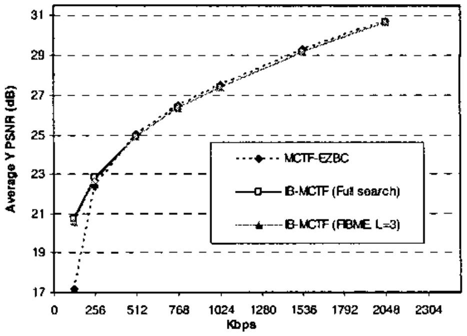

Figure 4 – Mobile&Calendar, CIF at 30fps

6. CONCLUSIONS

In this paper we introduced a new fast motion estimation method that works in the wavelet domain. The algorithm has a complexity that grows linearly with the size of the search window. The motion estimation module has been inserted in a fully scalable coder, which is inspired to IB-MCTF. In addition, variable-size block matching and our implementation of the update step have been integrated in the coder architecture. Experimental results proved that our coder yields a coding efficiency that is comparable with the state-of-the-art MCTF-EZBC coder, outperforming it from a subjective standpoint.

7. REFERENCES

[1] J.C. Ye, M. van der Schaar, "Fully Scalable Overcomplete Wavelet Video Coding using Adaptive Motion Compensated Temporal Filtering". *In Proceedings of VCIP2003*, July 2003, Lugano, Switzerland

[2] P. Chen, "Fully Scalable Subband/Wavelet Coding". *Ph.D. thesis* – Rensselaer Polytechnic Institute – Troy, NY, May 2003

[3] J. Ohm, "Motion-compensated Wavelet Lifting Filters with Flexible Adaptation". *In Proceedings of Tyrrenian International Workshop on Digital Communications*, September 2002, Capri, Italy

[4] H. Park, H. Kim, "Motion Estimation Using Low-Band-Shift Method for Wavelet-Based Moving-Picture Coding". In IEEE Trans. on Image Processing, vol. 9, No. 4, April 2000

[5] S. Mallat, "A Wavelet Tour of Digital Signal Processing", Academic Press, 1998

[6] S. Choi, J.W. Woods, "Motion-Compensated 3-D Subband Coding of Video". In IEEE Trans. on Image Processing, vol. 8, No. 2, February 1999