VIDEO ENCODER BASED ON LIFTING TRANSFORMS ON GRAPHS

E. Martínez-Enríquez, F. Díaz-de-María[1] *and Antonio Ortega*[2]

[1] Department of Signal Theory and Communications, Universidad Carlos III
[2] Department of Electrical Engineering, University of Southern California

ABSTRACT

We propose a complete video encoder based on directional "non-separable" transforms that allow spatial and temporal correlation to be jointly exploited. These lifting-based wavelet transforms are applied on graphs that link pixels in a video sequence based on motion information. In this paper, we first consider a low complexity version of this transform, which can operate on subgraphs without significant loss in performance. We then study coefficient reordering techniques that lead to a more realistic and efficient encoder that the one we presented in our earlier work. Our proposed technique shows encouraging results as compared to a comparable scheme based on the DCT transform.

Index Terms— Wavelet transforms, Video coding, MCTF, Lifting, Graphs

1. INTRODUCTION

Wavelet based transforms have been widely used for image and video processing in the last decade. Typical coding approaches based on these transforms transmit the position and magnitude of large transform coefficients. Thus, overall bit-rates can be reduced for a given video if the selected transform compacts the signal into a smaller number of large coefficients. This has led to interest in transforms that are able to filter along directional paths, as these can avoid filtering across large discontinuities, resulting in smaller high frequency coefficients in those locations. For image coding, directional transforms have been proposed in order to filter following the directions of low variation in pixel intensity, avoiding filtering crossing edges [1], [2]. For video coding, filtering along the motion trajectories, e.g., motion-compensated temporal filtering (MCTF), has been investigated [3],[4]. Side information (e.g., motion vectors or edge locations) is typically transmitted so that the decoder can identify the directional transform that was selected. Some of these works [3],[4], [2], are based on the lifting scheme, which is an intuitive and structurally invertible approach to construct multiresolution signal representations [5]. In [3] lifting is applied in the temporal domain, using motion compensated lifting steps to implement the transform. Lifting wavelet transforms in trees have been proposed to be applied to image coding in [2]. This idea is extended in [6] to arbitrary graphs data.

In [7] we presented a new lifting-based wavelet transform for video signals that allows spatial and temporal correlation to be jointly exploited. Our transform can filter following arbitrary 3-dimensional (spatio-temporal) directions. Firstly, we constructed a graph in which any pixel could be linked to a number of spatial and temporal neighbors. Then, we extended the transform in [6] to N levels of decomposition and applied it to the constructed graph, obtaining an invertible, critically sampled and "non-separable" transform. This transform showed improvements in performance as compared to the LIMAT method [3] and to a motion compensated DCT video encoder. This paper proposes a low-complexity version of the transform introduced in our previous work [7]. We also introduce here a new approach for reordering the coefficients that leads to an efficient real encoder. We achieve average gains of around 1.5 dB as compared to a simple DCT based encoder. Note that while in [7] we reported non-linear approximation results (PSNR as a function of number of non-zero coefficients), here we provide rate-distortion results to evaluate our method. The rest of the paper is organized as follows. Section 2 summarizes our proposed graph-based transform for video. Section 3 introduces our proposed low complexity transform. Our novel coefficient reordering method is presented in Section 4. Experimental results are provided in Section 5 and conclusions in Section 6.

2. GRAPH-BASED TRANSFORM

2.1. Lifting Transforms on Graphs

In a lifting-based multi-resolution representation [5], transform invertibility can be guaranteed if the input data at each specific level of decomposition j is split into prediction (P_j) and update (U_j) disjoint sets. Then prediction ($\mathbf{p}_{m,j}(m \in P_j)$) and update ($\mathbf{u}_{n,j}(n \in U_j)$) filters can be defined so that the m-th detail d_m can be computed from k update neighbors, and the n-th smooth s_n coefficients can be computed from l prediction neighbors as:

$$d_{m,j} = s_{m,j-1} + \sum_{k \in U_j} \mathbf{p}_{m,j}(k) s_{k,j-1}$$
$$s_{n,j} = s_{n,j-1} + \sum_{l \in P_j} \mathbf{u}_{n,j}(l) d_{l,j}. \quad (1)$$

The smooth coefficients at $(j-1)$-th decomposition level (s_{j-1}) are projected onto the approximation and detail subspaces, yielding, respectively, the smooth ($s_{n,j}$) and detail ($d_{m,j}$) coefficients at the j-th decomposition level. The iterative application of this process gives rise to a multiresolution decomposition.

2.2. Graph Construction

In [7] we proposed a graph-based representation for video data, in which pixels are linked to spatial or temporal neighbors with similar luminance values. To exploit spatial correlation, a pixel is linked to any pixel in its 8-connected neighborhood if no edge exists between the pixels (explicit edge information is provided for some frames and extrapolated to others). This criterion is similar to that employed in [2] for image compression. The intuition behind this approach is that it will tend to link correlated pixels and avoid filtering across edges, which would lead to high energy coefficients. Regarding the temporal correlation, we linked those pixels that are connected to each other by a motion model (e.g., corresponding pixels in two

blocks in successive frames that are connected via a motion vector, which is transmitted as side information). Thus, any pixel can be linked simultaneously in the graph to multiple spatial and temporal neighbors. Finally, note that links between pixels on the graph were weighted as a function of the reliability of each connection (i.e., the expected correlation between the pixels intensity). In this work, temporal connections will be weighted with a value of $t = 10$ and spatial connections with $s = 2$, based on the assumption that temporal prediction is usually more accurate than the spatial one. This weighting will influence the update-predict assignment, the filter design and the reordering of the coefficients.

2.3. Update-Predict Assignment and Filter Design.

In order to perform the transform, we split the nodes of the graph into disjoint prediction and update sets. This is essentially a 2-color graph coloring problem, and the transform operates on a bi-partite graph, where prediction pixels are only linked to update pixels [6]. The criterion used in the proposed transform was to assign a label to each pixel so as to maximize the reliability with which update nodes could predict prediction neighbors, which is equivalent to maximizing the total weight of the edges between the P_j and the U_j sets. To do that, we used the greedy solution in [8]. Note that the update and predict filters designed in [7] take into account the weights of the graph. Once we have the graph, the update and predict filters, and the prediction and update sets, we can obtain the smooth and detail coefficients (1) using the edges of the graph that did not present any conflict (neighbor nodes in the original graph with the same label).

2.4. Extending the transform to higher levels of decomposition.

In order to carry out a multi-resolution analysis, in [7] we proposed a multilevel extension of the transform on [6]. Summarizing, to construct the graph at decomposition level j from the graph at level $j-1$, we connected those update nodes that were directly linked or at one-hop of distance in the graph at level $j-1$. In this manner, the graph at successive levels of decomposition continues to capture the correlation between pixels. Once we construct the graph at level j, nodes again have to be split into disjoint prediction and update sets in order to perform the transform. Refer to [7] for details. Figure 1 shows the encoder and the decoder data flow. Note that, since the motion vectors and the edge map are sent to the decoder, the process performed at encoder is known at decoder so that the system is invertible.

3. LOW COMPLEXITY TRANSFORM USING SUBGRAPHS

In our previous work encoder complexity increases rapidly with the number of nodes D (e.g., the worse-case time complexity for the greedy update-predict assignment algorithm used in our encoder is $O\left(D^3 \cdot logD\right)$ [8]). We now propose a transform that operates on subgraphs of the original graph in order to reduce overall complexity with negligible loss of performance.

The goal will be to divide the original graph node set V of size $M \times N \times F$ (where $M \times N$ is the frame size and F the number of frames considered in the graph construction) in I subsets A_i, so that in any of the subgraphs formed with the nodes of each subset $S_i\left(A_i, E_i\right)$ we can obtain a transform that is invertible and critically sampled, which takes into account the interactions between the nodes of different subgraphs. Critical sampling means that the number of transform coefficients generated over all sub-graphs is the same as the original number of pixels. A necessary condition to achieve these objetives will be that the A_i node subsets have to been

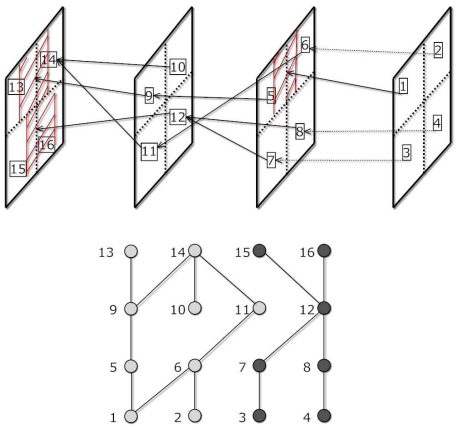


Fig. 2. Subgraph construction from 4 frames. Top: motion dependency tree. Bottom: two subgraphs are formed corresponding to dark and light grey block pixels, respectively.

disjoint. Our proposed solution creates subgraphs based on disjoint subsets that contain linked pixels in F temporal hops, thus keeping the more reliable links of the subgraph in any level of the transform. To do that, we divide each frame into blocks of size $P \times Q$. Then, we perform the motion estimation for the F frames. With this information, a tree is generated in which the children of a given block a are the blocks of the reference frame that are linked to a by means of the motion model. Finally, each subgraph is composed of the pixels that belong to the blocks that are linked along the F frames. This is achieved using Algorithm 1, which given an initial set of n groups G_i, each composed by a block a and its children, constructs the subgraphs by joining groups that have common elements, and deleting those groups that have already been aggregated into a subgraph. An example of the subgraph construction is shown in Fig. 2.

Algorithm 1 Subgraph Formation

Require: n Groups G_i
while $Flag \neq 0$ **do**
  Set $Flag = 0$
  for $i = 1$ **to** n **do**
    if $G_i \neq deleted$ **then**
      for $j = i + 1$ **to** n **do**
        if $(G_j \neq deleted)$ and $(G_i \cap G_j \neq \phi)$ **then**
          Set $Flag = 1$
          Set $G_i = G_i \cup G_j$
          Delete G_j
        end if
      end for
    end if
  end for
end while
return G_i and Index Vector of Non-deleted Groups

Table 1 provides experimental results for the number of subgraphs formed and the corresponding complexity reduction (CR), calculated as the ratio of encoding times when using the subgraphs and the original graph, applied to coding 20 frames of three $QCIF$ sequences. It can be seen that the complexity reduction can be significant. Nevertheless, one drawback of this approach is that the final complexity depends on the motion of the video sequence (faster motion sequences tend to lead to larger subgraphs). There are several approaches to mitigate this problem. For example, motion vectors

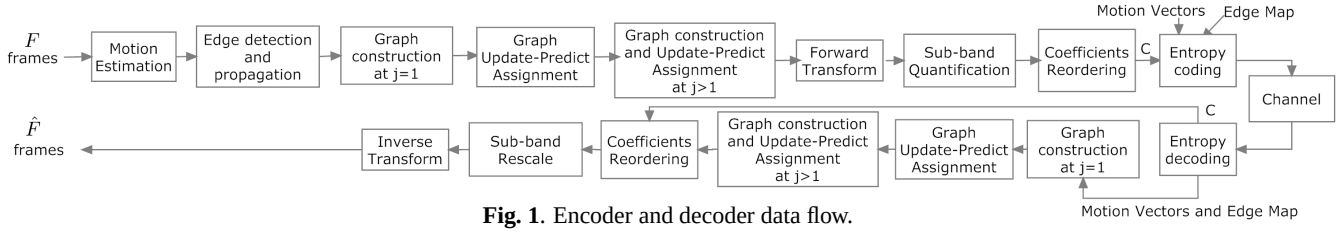


Fig. 1. Encoder and decoder data flow.

Table 1. Subgraph approach performance

	Number of Subgraphs	CR
Mobile	82	48
Foreman	14	4
Carphone	1	1

could be constrained (e.g., motion vectors would have to point to co-located slices in previous frame). Alternatively, links between nodes in the bigger subgraphs could be removed leading to new smaller disjoint subgraphs until a required complexity restriction is achieved. Or the maximum number of blocks that a subgraph can have in Algorithm 1 could be limited. Any of these approaches will lead to a simpler transform but have an impact on performance. We plan to consider these trade-offs in future work.

4. COEFFICIENT REORDERING

In [7] performance comparisons were based on a non-linear approximation as in [1], which gives an idea of how the transform compacts the energy in a few coefficients. In typical practical encoders, instead, quantized transform coefficients are scanned in a given order to apply entropy coding. For example, in DCT based encoders, the reordering is usually performed following a zigzag scanning order within each block, while in wavelet based approaches, bitplane by bitplane scanning of transform coefficients has been a popular approach [9], [10]. We propose two different approaches to re-order the coefficients generated by our graph-based transform: (i) inter-subband reordering, which implies sorting the coefficients as a function of the subband to which they belong; and (ii) intra-subband reordering, which sorts the coefficients of a subband as a function of the reliability with which they were predicted.

4.1. Inter-subband reordering

Because our transform achieves significant energy compaction, the energy in the middle-high frequency subbands will tend to be very low, so that these sub-bands will be likely to have a large number of zero coefficients after quantization. Based on this, we will group coefficients that belong to the same subband, increasing the probability of having long strings of zero coefficients. Specifically, the coefficients will be sorted as $\mathbf{coeffs}_{inter} = \left[\mathbf{s}^{j=N}, \mathbf{d}^{j=N}, \mathbf{d}^{j=N-1}, \ldots, \mathbf{d}^{j=1}\right]$, where $\mathbf{s}^{j=N}$ are the smooth coefficients at level of decomposition $j = N$ (the lower frequency subband), and $\mathbf{d}^j$ are the detail coefficients at generic level of decomposition j. Refer to Fig. 3 for an example of the effect of ordering on quantized coefficients from 20 frames of the sequence $Carphone$.

4.2. Intra-subband reordering

The graph is known at both encoder and decoder. Its edge weights provide an estimate of the reliability with which one predict node is predicted from update neighbors. We make the assumption that the magnitude of detail coefficients in predict nodes will tend to be smaller if they have been predicted from more "reliable" update neighbors (i.e., prediction is better). Thus, we propose to re-

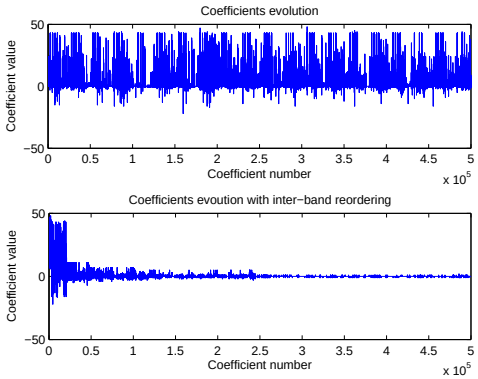


Fig. 3. Inter-subband reordering. Top: original coefficients. Bottom: reordered coefficients

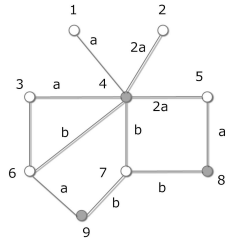


Fig. 4. Intra-subband reordering example.

order the coefficients in each subband as a function of the reliability of their links, grouping together the more reliable predicted nodes, which will likely lead to smaller magnitude detail coefficients. An example of this reordering is shown in Fig. 4. In the example, the detail coefficients (the white nodes) of a generic subband j, $\mathbf{d}^j = [1, 2, 3, 5, 6, 7]$, have the following reliability values, $\mathbf{r}^j = [a, 2a, a, 3a/2, (a+b)/2, 3b/3]$, calculated as the average of the weights of all graph edges used to compute that coefficient. Assuming that $a > b$, this gives rise to intra-subband reordered coefficients $\mathbf{d}^j_{intra} = [7, 6, 1, 3, 5, 2]$.

Table 2 shows some results of the bit rate in Kbps after coding 20 frames of the sequences Foreman and Carphone at different qualities (32.9 dB and 36 dB respectively) without reordering the coefficients, employing inter-reordering, and inter and intra reordering. The rate is obtained with an arithmetic coder as is explained in Section 5.

5. EXPERIMENTAL RESULTS

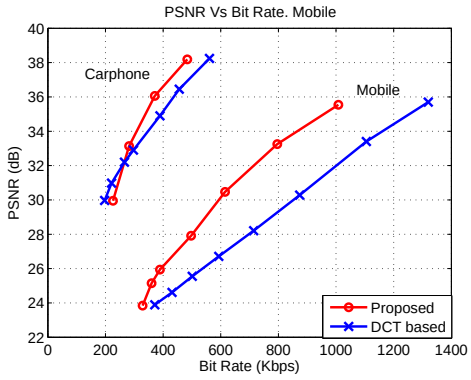
To evaluate the coding performance of the proposed encoder, we compare it against a motion-compensated DCT video encoder in terms of Rate-Distortion for different test sequences. The coefficients are quantized using a uniform dead-zone quantizer in the DCT, and a subband dependent quantization in our encoder (i.e., the quantization step is lower in low frequency subbands and vice versa). These quantized coefficients are scanned as is explained in Section 4 in our proposed method, and in the tradicional zigzag scanning order in the DCT. Note that this process is performed in scanning units of size S. Then, a run-length encoding is performed in both encoders,

Table 2. Performance comparison using inter-subband and intra-subband reordering

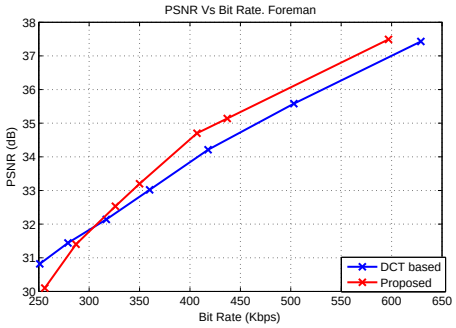
	Without reordering	Inter reordering	Inter and Intra reordering
Foreman	503 Kbps	404 Kbps	350 Kbps
Carphone	502 Kbps	425 Kbps	371 Kbps

obtaining the symbols to be entropy coded. An end-of-block special symbol is used to indicate that all remaining coefficients in the scanning unit are quantized to zero. Finally, the bitstream is obtained coding the symbols using an adaptive arithmetic coder.

Regarding the side information, motion vectors are differentially encoded with respect to a predicted motion vector obtained from adjacent blocks. Then, a variable length code (VLC) is used to code this predicted motion vector. Note that the motion vectors to transmit will be different in the proposed and in the DCT based encoders, since the matching is carried out in original frames in the former and in reconstructed reference frames in the latter. Nevertheless, the rate will be similar in both cases. Our encoder will have an extra overhead because it should send the edge information to the decoder once every F frames. These edge maps are encoded using JBIG, obtaining negligible rates of around 5 Kbps. In the experiments, $F = 20$, $S = 256$, and five levels of decomposition of the proposed transform are performed. Block sizes of 16×16 and one reference frame are assumed in the motion estimation process. In the DCT encoder, we use 8×8 DCT.



(a) Mobile (QCIF) and Carphone (QCIF)



(b) Foreman (QCIF)

Fig. 5. PSNR versus Bit Rate.

Fig. 5 shows the Rate-Distortion curves for three different $QCIF$ sequences, $Mobile$, $Carphone$ and $Foreman$. In general, our proposed method outperforms the DCT based approach. In $Mobile$ sequence, our method is 4 dB better than the DCT in medium to high qualities. The gain is significant in $Carphone$ (about 1-1.5 dB) and moderate in $Foreman$ (around 0.5 dB). However, the efficiency of the encoder at low qualities gets worse, losing against the DCT based encoder in $Foreman$ and $Carphone$ for qualities lower than 32 dB. The results are in agreement with those of [7].

6. CONCLUSIONS

We have developed a video encoder based on lifting transforms on graphs that can perform filtering following spatio-temporal directions of high correlation between pixels. A low complexity approach of this transform has been proposed. A coefficient reordering method is also given that compacts the zero coefficients together. This provides improved performance over a DCT based encoder.

7. REFERENCES

[1] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: anisotropic multidirectional representation with separable filtering," *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 1916 –1933, July 2006.

[2] G. Shen and A. Ortega, "Compact image representation using wavelet lifting along arbitrary trees," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, October 2008, pp. 2808 –2811.

[3] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (limat) framework for highly scalable video compression," *Image Processing, IEEE Transactions on*, vol. 12, no. 12, pp. 1530 – 1542, December 2003.

[4] G. Pau, C. Tillier, B. Pesquet-Popescu, and H. Heijmans, "Motion compensation and scalability in lifting-based video coding," *Signal Processing: Image Communication*, vol. 19, no. 7, pp. 577 – 600, 2004, Special Issue on Subband/Wavelet Interframe Video Coding.

[5] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," Tech. report 1995:6, Industrial Math. Initiative, Dept. of Math., University of South Carolina, 1995.

[6] Sunil K. Narang and A. Ortega, "Lifting based wavelet transforms on graphs," in *APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, October 2009.

[7] E. Martínez Enríquez and A. Ortega, "Lifting transforms on graphs for video coding," in *Data Compression Conference (DCC), 2011*, March 2011, pp. 73 –82.

[8] C.-P. Hsu, "Minimum-via topological routing," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 2, no. 4, pp. 235 – 246, 1983.

[9] J. M. Shapiro, "An embedded wavelet hierarchical image coder," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, March 1992, vol. 4, pp. 657 –660 vol.4.

[10] B.-J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," in *Data Compression Conference, 1997. DCC '97. Proceedings*, March 1997, pp. 251 –260.