LIFTING TRANSFORMS ON GRAPHS FOR VIDEO CODING

Eduardo Martínez-Enríquez[a] *and Antonio Ortega*[b]

[a] Department of Signal Theory and Communications, Universidad Carlos III;
[b] Department of Electrical Engineering, University of Southern California

ABSTRACT

We present a new graph-based transform for video signals using wavelet lifting. Graphs are created to capture spatial and temporal correlations in video sequences. Our new transforms allow spatial and temporal correlation to be jointly exploited, in contrast to existing techniques, such as motion compensated temporal filtering, which can be seen as "separable" transforms, since spatial and temporal filtering are performed separately. We design efficient ways to form the graphs and to design the prediction and update filters for different levels of the lifting transform as a function of expected degree of correlation between pixels. Our initial results are promising, with improvements in performance as compared to existing methods in terms of PSNR as a function of the percentage of retained coefficients of the transform.

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

1. INTRODUCTION

The lifting scheme is an intuitive and structurally invertible approach to construct multiresolution signal representations [1]. Lifting based wavelet transforms have been widely used for image and video processing, mainly in the last ten years, and there are many works about these topics in the literature. In the field of image coding, lifting-based wavelets have been used to capture directional information, avoiding filtering across the object edges, giving rise to very efficient representations of the image. Examples can be found in [2], [3] or [4]. For video coding, lifting is usually applied in the temporal domain. The main multiresolution decomposition structures in wavelet-based video coding are $t + 2D$ and $2D + t$. In the former, the video sequence is first filtered in the temporal direction along the motion trajectories (motion-compensated temporal filtering-MCTF) and then a 2D wavelet transform is carried out in the spatial domain [5]. In the latter, each frame is firstly wavelet transformed in the spatial domain, followed by MCTF. Focusing in the temporal domain, representative examples of MCTF implementations are [6] and [7], which use motion-compensated lifting steps to implement the temporal wavelet transform, filtering along a set of motion trajectories described by a specific motion model. These approaches can be described as "separable" in that spatial and temporal filtering are applied in separate steps.

In all of these works, in order to perform the prediction and update steps of the lifting scheme, the input sequence is split into update (even frames) and prediction (odd frames) subsequences, and for each level of the transform, the prediction subsequence is predicted from the update subsequence giving rise to the high-pass subband sequence, and the update subsequence is updated by using a filtered version of the prediction one, thus obtaining the low-pass subband sequence. In cases in which the motion model cannot accurately capture the real motion of the scene, this kind of splitting into even and odd frames will lead to the linking of update and prediction pixels with very different luminance values. In this way, prediction frames will be poorly predicted from update

frames, leading to significant energy in the high pass subband sequence, and thus relatively low energy compaction. Moreover, when using MCTF, problems arise due to occlusions and uncovered areas (pixels that are filtered several times or are not filtered at all). Some authors handle this problem by identifying unconnected and multiple connected pixels and adapting the predict and update operators accordingly (e.g., [8]).

The key novelty in our work is describing the video sequence as a graph of connected pixels and applying the lifting transform on this graph. Given that pixels (the nodes of the graph) are linked to spatial or temporal neighbors (or both) it is easy to make use of spatio-temporal filtering operations, selected to follow spatio-temporal directions of maximum correlation between pixels. Similar to [2], [3], bandelets [9] or directionlets [10], examples of directional wavelet transforms whose basis functions are adapted to any 2-dimensional direction in the spatial domain for efficient image representation, our approach can filter following any 3-dimensional direction of the spatio-temporal domain. Moreover, our proposed scheme can avoid problems due to occlusions and uncovered areas, leading to simple critically sampled invertible transform.

Our starting point is the lifting-based wavelet transform for graph data presented in [11]. We extend the transform to N-levels of decomposition and apply it to video coding. The connections in the graph are constructed in such a way that pixels expected to have similar luminance will tend to be connected. Therefore, the prediction of a pixel from its graph neighbors can be more accurate, leading to reduced energy in the high-pass subband at any decomposition level of the transform. To get a more accurate prediction, the connection between any pair of pixels is weighted as a function of estimates of correlation between the pixels, i.e., higher expected correlation tends to lead to better prediction and larger prediction weights. This weighting will be used in the design of the prediction and update filters and in the construction of the graph in successive levels of decomposition, thus helping improve prediction at all levels.

The links between pixels can be temporal (pixels connected by means of a motion model) or spatial (one-hop neighbor pixels that do not cross edges), and the number of neighbors that one pixel can have in the graph can vary locally so that we can have flexibility in designing the corresponding filtering operations. Our work could be considered as a generalization of wavelet-based video coding that gives rise to a more versatile solution where spatial and temporal operations are no longer separable. The transform requires that some side information be sent to the decoder, so that the same graph can be constructed at both encoder and decoder. Specifically, temporal information (motion vectors) and spatial information (edges) have to be sent. In terms of non-linear approximation we achieve average gains of 2.3 dB and 1.3 dB as compared to a DCT based encoder and the LIMAT method [6], respectively.

The rest of this paper is organized as follows. In Section 2 we describe in detail our novel video coding scheme based on lifting transforms on graphs. In Section 3 we provide experimental results. Finally, conclusions and future work are given in Section 4.

2. PROPOSED SCHEME

2.1. Lifting Transforms on Graphs

The lifting approach for wavelet construction and its relation with the multiresolution analysis are presented in [1]. To perform the transform and ensure its invertibility, the input data at each specific level of decomposition j should be split into prediction (P_j) and update (U_j) disjoint sets, and the predict ($\mathbf{p}_{m,j}(m \in P_j)$) and update ($\mathbf{u}_{n,j}(n \in U_j)$) filters should be specified. Then, following the notation employed in [12], the m-th detail d_m and n-th smooth s_n coefficients can be computed as:

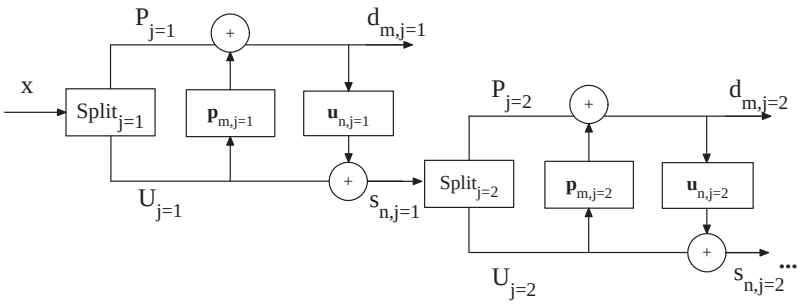

Fig. 1. Lifting scheme. Two levels of decomposition.

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

The smooth coefficients at $(j-1)$-th decomposition level ($s_{n,j-1}$) 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. Applying this process iteratively gives rise to a multiresolution decomposition. In Figure 1 the lifting structure for two levels of decomposition is shown. Note that the data at level $j = 0$ will be the original raw data (the luminance of the pixels in our case), and will be denoted as x_g, so $x_g = s_{n,j=0}$, where g is the pixel index. For $j > 0$, $s_{n,j}$ will be the low pass version (smooth projection) of the $(j-1)$-th level.

Lifting-based wavelet transforms on trees are applied to image coding in [2] and to sensor networks applications in [12]. In these works, the authors split the data into prediction and update sets at each level of decomposition according to their depth with respect to the root of the tree, so that no pair of directly connected nodes belongs to the same set (i.e., prediction pixels only connect to update pixels). [11] extends this idea to arbitrary graphs (which are in general cyclic and non-planar). Nevertheless, in this case, for an arbitrary prediction-update assignment, nodes that are neighbors in the graph are not guaranteed to have opposite parity, so that prediction pixels that are connected to other prediction pixels cannot be used to obtain the detail in (1). As a solution to this problem, the authors seek techniques that minimize the number of conflicts (i.e., the percentage of direct neighbors in the graph that have the same parity). Then, they perform the transform using the edges of the graph that do not present any conflict. In the next sections we describe how to represent video content as a graph, defining the prediction-update assignment at each transformation level and the construction of the filter operators.

2.2. Graph Construction

In this section we propose a graph representation for video content. The goal in the construction of the graph at the j-th level of decomposition will be to link pixels with similar luminance values, in such a way that detail coefficients $d_{m,j}$ in (1) will be very close to zero. In this manner, the high pass subband energy at this level j will be low, achieving an efficient representation of the data. Here we explain how to form the graph at the $j = 0$ level of decomposition from the original video sequence. In successive levels $j > 0$, we will construct the graph at level j from the graph at level $j - 1$ as explained in Section 2.4.

Consider a video sequence of V frames of size $M \times N$ and a subsequence of F frames ($F \leq V$). We will employ a new graph for every subset of F frames, until all the V frames in the sequence are coded. Let x_g be the luminance value of pixel $g \in G = \{1, 2, ..., M \times N \times F\}$. Any pixel $g \in G$ can be linked to any subset of pixels $H \subset G$, with $h \in H, g \neq h$, following criteria to be described next. Since we exploit the spatial and temporal correlation jointly, a pixel g can be linked to spatial and temporal neighbors at the same time.

With respect to the spatial correlation, the criterion for graph construction will be very similar to that employed in [2] for image compression. Pixels that are close to each other, and in general pixels that belong to the same object, will tend to have correlated luminance values. In contrast, when filtering across edges, there can be a significant amount of energy in the high pass subbands, because the value of neighboring pixels will be very different. Thus, if we avoid filtering across the edges, we will obtain a more compact representation of the data. Following this reasoning, we will link those pixels that are one-hop neighbors in any direction and do not cross any edge. To do that, we will need to estimate the edges and send this information to the decoder. To reduce the resulting overhead, we note that if there are no occlusions and the motion model captures object motion accurately, it will be possible to obtain edge information in the current frame using edge data obtained from the reference frame, along with motion information. Thus, in practice we only need to explicitly send edge information to the decoder once every F frames.

Regarding the temporal correlation, we will link those pixels that are related by means of a motion model. In our example, block motion search is used, and every pixel belonging to a block is linked to the corresponding pixel belonging to the best block match in the reference frame. Motion vectors (MV) need to be sent to the decoder in order to describe this movement. Finally, note that motion mappings are made using the original video frames, that is, the reference frame is not a reconstruction from a previously encoded frame. An example of graph construction and edge information transmission is shown in Figure 2 for two frames, where it can be seen that links between pixels follow the motion direction and avoid crossing edges within a frame.

Temporal links are identified using an explicit search that minimizes a distortion measure between pixels (i.e., the standard motion estimation). Therefore, in general, temporal links in the graph will be more reliable than spatial links, that is, the expected correlation between temporal linked pixels will be higher than the one between spatial linked pixels. In order to take these features into account, we will weight the edges of the graph as a function of the reliability of each connection. This will influence the update-predict assignment and the filter design, to be discussed in Section 2.3. As a starting point, temporal connections will be weighted with a value of t and spatial connections with s. Nevertheless, more specific metrics that depend on the features of the sequence can be investigated.

2.3. Update-Predict Assignment and Filter Design

Once a graph is constructed, the next step is to split the nodes into prediction (P_j) and update (U_j) disjoint sets in order to perform the transform. The criterion to assign a label to each pixel will be to maximize the reliability with which update nodes can predict prediction neighbors, which is equivalent to maximizing the total weight of the edges between the P_j and the U_j sets. This problem is generally known as the *weighted maximum cut problem*. In this paper we use the approach of [13] for simplicity, leaving for future work a study of alternative methods (e.g., the one proposed in [14]). The greedy solution in [13] is described in Algorithm 1, where U_j and P_j form a bipartition of the node set U_{j-1}, and we consider *gain* of a node to be the sum of weights of all its incident edges.

An example of the update-predict assignment is shown in Figure 3. Note that the update nodes are usually connected by means of reliable links to prediction nodes, so we can obtain an accurate prediction of these prediction nodes from the update nodes. Conflicts are indicated as broken links. As in [11], we will only use no-conflict links to perform the transform. In order to obtain the detail

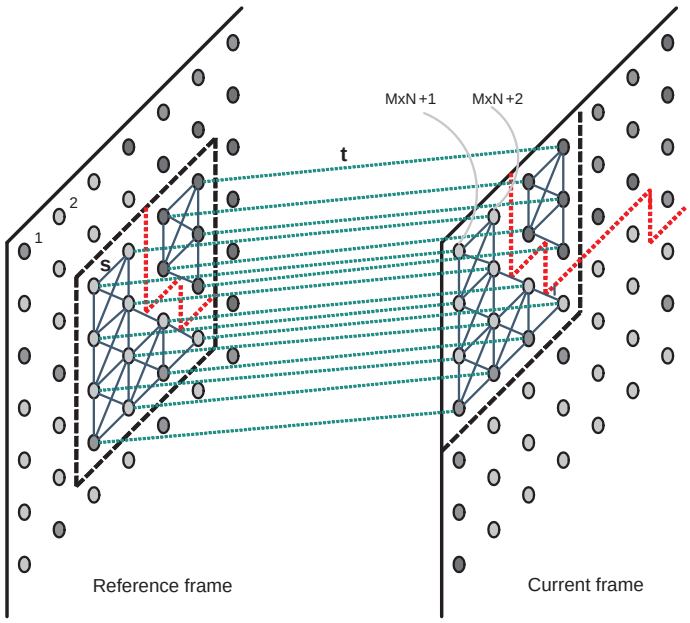

Fig. 2. Spatio-temporal graph construction. The grey level represents the luminance value of each pixel; the red-thick dashed lines are the object edges; the green-fine dashed lines represent temporal connections, and the blue solid lines spatial connections. Finally, the black dashed lines represent the block size.

coefficient (1) in a prediction pixel m, we define robust predict filters that weight the neighbor update pixels taking into account the reliability of each of their connections to m. Thus, we define the following vector of prediction weights:

$$\mathbf{p}_{m,j} = -\frac{[w_{1_u,j}\, w_{2_u,j}\, \ldots,\, w_{V_u,j}]}{\sum_{i_u=1}^{V_u} w_{i_u,j}} \tag{2}$$

where $[w_{1_u,j}\,, w_{2_u,j}, \ldots,\, w_{V_u,j}]$ is the vector of weights in the graph between the update neighbors $\{1_u, 2_u, ...V_u\} \in U_j$ and the prediction pixel $m \in P_j$.

Algorithm 1 Weighted Maximum Cut Algorithm

Require: $U_j = \{\varnothing\}$, $P_j = \{U_{j-1}\}$
1: Calculate the $Gain$ of the U_{j-1} node set
2: Select the node a with largest $Gain$, $a = max(Gain)$
3: **while** $Gain > 0$ **do**
4: Let $U_j \leftarrow U_j \cup \{a\}$
5: Let $P_j \leftarrow P_j \backslash \{a\}$
6: Change the sign of the incident edge weights
7: Update $Gains$ of adjacent nodes
8: Select the node a with largest $Gain$, $a = max(Gain)$
9: **end while**
10: **return** U_j and P_j

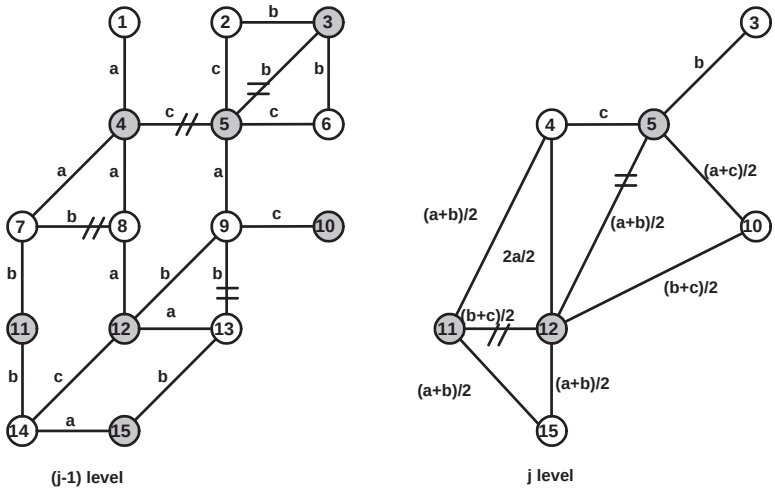

Fig. 3. Graph construction for consecutive levels of decomposition. $a = 10$, $b = 5$ and $c = 3$ are the different weight values. Grey nodes are update nodes, and white ones are prediction nodes.

The update filter is designed to smooth the next level approximation coefficients as:

$$\mathbf{u}_{n,j} = \frac{\left[w_{1_p,j}\, w_{2_p,j}\, \ldots,\, w_{W_p,j}\right]}{2(\sum_{i_p=1}^{W_p} w_{i_p,j})} \tag{3}$$

where $\left[w_{1_p,j}\,, w_{2_p,j}, \ldots,\, w_{W_p,j}\right]$ are the weights to be applied to the neighbors $\{1_p, 2_p, ...W_p\} \in P_j$ of an update pixel $n \in U_j$.

2.4. The transform in higher levels of decomposition

In order to carry out a multiresolution analysis, the low pass coefficients are successively projected in different transformation levels onto smooth and detail subspaces. To obtain the graph at transformation level j from the graph at level $j-1$, we connect those update nodes that are directly connected or at one-hop of distance in the graph at level $j-1$, so that the simplified graph continues to capture the correlation between pixels. In the new graph in j, the weight between nodes will be the average of the weights in the path between connected nodes at level $j-1$, so that high-correlation paths at level $j-1$ imply high weight links at level j. Once we construct the graph at level j, we should split the nodes again into prediction (P_j) and update (U_j) disjoint sets in order to perform the transform. Figure 3 shows an example of a graph construction at level j from a graph at level $j-1$, and the update-predict assignment at both transformation levels.

3. EXPERIMENTAL RESULTS

In order to evaluate the performance of our approach, we will employ the K term non-linear approximation (outlined in [10]). It consists of keeping the K largest coefficients of the transform and setting the rest to zero. This is a good indicator of energy compaction and thus of coding performance. We compute the average PSNR of each sequence of $V = 100$ frames as a function of the percentage of retained coefficients.

In our experiments, five levels of decomposition of the proposed transform are performed on the constructed graphs. Our method is compared to the Haar version of the MCTF approach described in [6] (the LIMAT method), and to a motion-compensated discrete cosine transform (DCT) video coder

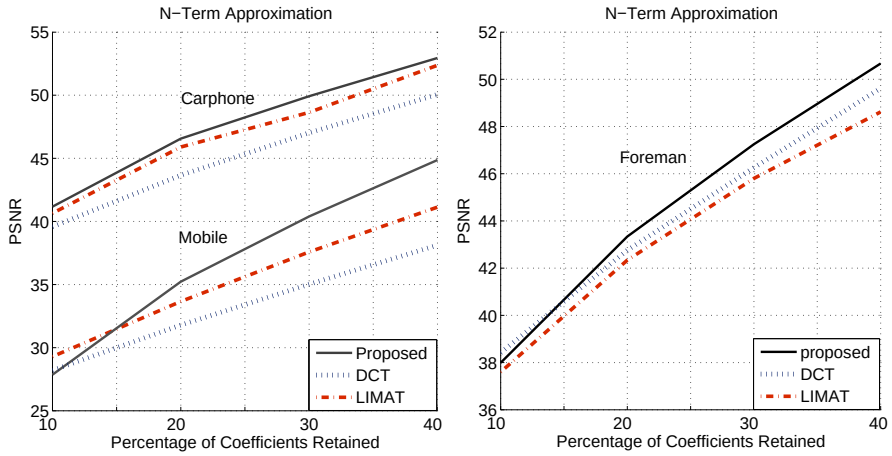

Fig. 4. PSNR versus percentage of retained coefficients.

in which a residual image, obtained after block motion estimation and compensation processes, is transformed by a 8×8 DCT; this scheme is the basis of the latest video coding standards as $H.264/AVC$.

Side information is not taken into account in the results. In the proposed method, we will have an overhead associated with the temporal and spatial information needed to construct the graph at the decoder. Regarding the temporal overhead, the same motion model is employed in *all* compared methods, i.e., a standard motion vector on 8×8 pixel blocks is assumed (only one reference frame), and thus this overhead does not need to be considered in the comparison. As for the spatial information overhead, we choose $F = 20$ and assume a binary edge map (obtained using Roberts' gradient operators) is sent to the decoder once every F frames, so that the spatial side information will be very low [1]. Since this overhead is very small it is not considered in our current version of the work. Note that there exists a trade-off between graph accuracy and the side information needed construct the graph. Higher rate to describe the spatial and temporal information (e.g., very small block sizes for motion), means that the correlation between linked pixels is also better captured by the graph, leading to potential compression gains. The weights used in the experiments are $t = 10$ and $s = 2$, following the reasoning that temporal prediction is more accurate (and costly in bits) than the spatial one. Finally, note that, for simplicity, in the current version only the two more reliables neighbors of each pixel are used for filtering and to construct the graphs at the different levels of decomposition. We plan to further investigate alternative filter and graph design approaches in follow up work.

Figure 4 shows the PSNR versus percentage of retained coefficients of three different $QCIF$ sequences, $Mobile$, $Carphone$ and $Foreman$. Our proposed method outperforms the DCT based and the LIMAT transforms in terms of PSNR. In the $Mobile$ sequence, when 40 percent of coefficients are retained, our method is 7 dB and 4 dB better than the DCT and the LIMAT respectively. However, the LIMAT method is better than the proposed when a very small percentage of coefficients are retained in $Mobile$. One posible reason could be to have chosen spatio-temporal filtering directions worse than the temporal ones chosen by the LIMAT. This problem could be fixed by using more accurate metrics to weight the graph.

For subjective evaluation, Figure 5 shows the original of the frame number 12 of the sequence $Mobile$ (upper-left part) and the reconstruction from the DCT transform on the residual (upper-right part), the LIMAT approach (lower-left part) and the proposed method (lower-right part). The

[1]For example, with JBIG compression of edge maps as used in [2] rates of the order of 0.02 bits per pixel are required every 20 frames, so that the overall overhead is negligible, 0.001 bits per pixel overall

reconstruction is carried out from the 20 % of retained coefficients. It can be shown that our transform achieves significant better perceptual quality than the DCT, and slight improvements over the LIMAT method (see for example the three animals of the upper-left part of the frames).

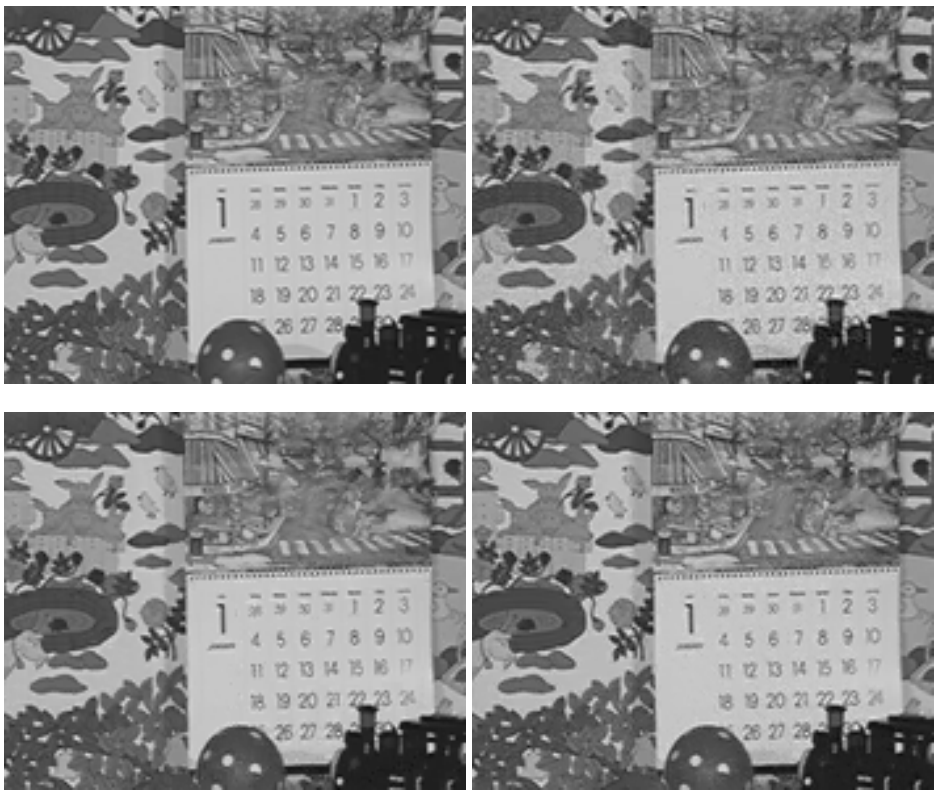

Fig. 5. Original (upper-left) and reconstruction with 20 % of the transform coefficients from the DCT on the residual image (upper-right), LIMAT (lower-left) and our proposed method (lower-right).

3.1. Performance in uncovered areas

To further explain the advantages of our proposed scheme we now consider in more detail situations involving uncovered areas. Refer to Figure 6, where we show the motion mappings used by the Haar version of the LIMAT approach with two levels of decomposition. Prediction frames (P) will be filtering following the directions indicated by the MV, and update frames (U) will be updated using inverse mappings MV^{-1}. Grey pixels represent non-updated pixels in the $j-1$ level of decomposition, that is, pixels that have not been low-pass filtered and thus contain high frequency energy. This high frequency will not be removed using the smooth coefficients at j level, giving rise to inefficiency. The black pixel represents a pixel that has not been decorrelated at any level, so that the coefficient after both levels of decomposition will be the raw pixel. The proposed method can solve this problem by representing video information as a graph (Figure 2) leading to a versatile prediction-update assignment, in which prediction and update nodes can belong to the same frame. To show this statement we have encoded two different 32×32 pixel areas of the sequence Foreman. Area 1 starts at pixel (1,1), so that could be considered a fairly static area. Area 2 starts at pixel (80,80), corresponding to a very dynamic area (the face of the man). The results in terms of PSNR when saving the 20 % of the coefficients are preserved are given in Table 1. Our proposed method

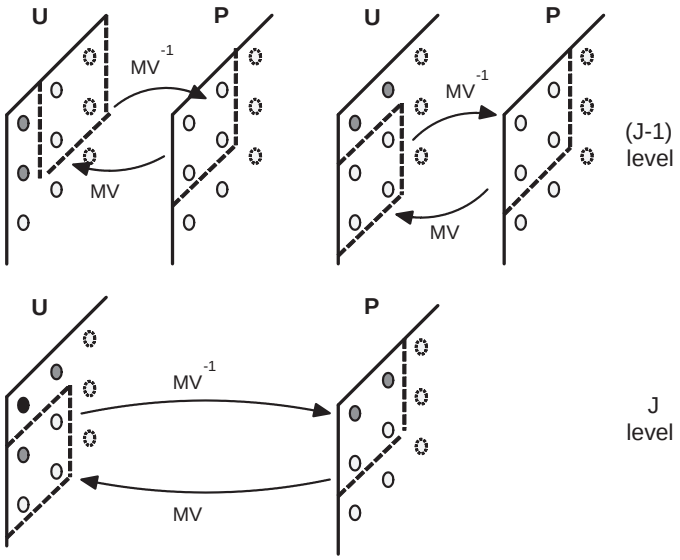

Fig. 6. Uncovered areas in LIMAT.

Table 1. Comparison of LIMAT and the proposed transform coding different areas

	PSNR(dB) in Area 1	**PSNR(dB) in Area 2**
Proposed	43.1	36
LIMAT	42.4	33.3
Δ	0.7	2.7

obtains slightly better results than the LIMAT in Area 1, while significantly outperforming LIMAT in Area 2, where there is a lot of motion and the uncovered background problem manifests itself.

4. CONCLUSIONS AND FUTURE WORK

We have proposed a directional lifting wavelet transform that is able to filter along 3 dimensional spatio-temporal directions of high correlation between pixels, leading to a compact representation of the data with low energy in the high frequencies. The results in terms of K-term non linear approximations are very promising. However, quantized transform coefficients should be encoded as compactly as possible to obtain an efficient real encoder. We are currently investigating how to use the graph information in the decoder to group together nonzero coefficients. Another interesting future line of research would be to design a low-complexity version of the transform that works with sub-graphs formed from the original graph without loss of performance.

Acknowledgments

The authors thank Sunil K Narang and Godwin Shen for the code to implement the prediction-update assignment and the lifting in graphs.

5. REFERENCES

[1] Wim 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.

[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] G. Shen and A. Ortega, "Tree-based wavelets for image coding: Orthogonalization and tree selection," in *Picture Coding Symposium, 2009. PCS 2009*, May 2009, pp. 1 –4.

[4] Raanan Fattal, "Edge-avoiding wavelets and their applications," in *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, New York, NY, USA, 2009, pp. 1–10, ACM.

[5] N. Adami, A. Signoroni, and R. Leonardi, "State-of-the-art and trends in scalable video compression with wavelet-based approaches," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1238 –1255, September 2007.

[6] 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.

[7] Gregoire Pau, Christophe Tillier, Batrice Pesquet-Popescu, and Henk 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.

[8] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *ICASSP '01: Proceedings of the Acoustics, Speech, and Signal Processing, 2001. on IEEE International Conference*, Washington, DC, USA, 2001, pp. 1793–1796.

[9] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *Image Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 423 –438, April 2005.

[10] 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.

[11] 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.

[12] Godwin Shen and A. Ortega, "Optimized distributed 2d transforms for irregularly sampled sensor network grids using wavelet lifting," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, March 2008, pp. 2513 –2516.

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

[14] S.K. Narang, G. Shen, and A. Ortega, "Unidirectional graph-based wavelet transforms for efficient data gathering in sensor networks," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 2902 –2905.