Martínez-Enríquez, E., Cid-Sueiro, J., Díaz-de-María, F., Ortega, A. (2018) *Directional Transforms for Video Coding Based on Lifting on Graphs*. In: IEEE Transactions on Circuits and Systems for Video Technology, 28(4), pp.: 933-946.

DOI:10.1109/TCSVT.2016.2633418

Directional Transforms for Video Coding Based on Lifting on Graphs

Eduardo Martínez-Enríquez, Jesús Cid-Sueiro, Fernando Díaz-de-María, *Member, IEEE*, and Antonio Ortega, *Fellow, IEEE*

***Abstract*—In this work we describe and optimize a general scheme based on lifting transforms on graphs for video coding. A graph is constructed to represent the video signal. Each pixel becomes a node in the graph and links between nodes represent similarity between them. Therefore, spatial neighbors and temporal motion-related pixels can be linked, while non-similar pixels (e.g., pixels across an edge) may not be. Then, a lifting-based transform, in which filterin operations are performed using linked nodes, is applied to this graph, leading to a 3-dimensional (spatio-temporal) directional transform which can be viewed as an extension of wavelet transforms for video. The design of the proposed scheme requires four main steps: (i) graph construction, (ii) graph splitting, (iii) filte design, and (iv) extension of the transform to different levels of decomposition. We focus on the optimization of these steps in order to obtain an effective transform for video coding. Furthermore, based on this scheme, we propose a coefficien reordering method and an entropy coder leading to a complete video encoder that achieves better coding performance than a motion compensated temporal filterin wavelet-based encoder and a simple encoder derived from H.264/AVC that makes use of similar tools as our proposed encoder (reference software JM15.1 configu ed to use 1 reference frame, no subpixel motion estimation, 16×16 inter and 4×4 intra modes).**

***Index Terms*—Video coding, Lifting transform, Directional transforms, Signal processing on graphs.**

I. Introduction

A. Motivation

Compact representations of signals are very useful in many applications such as coding, denoising or feature extraction. Classical transforms such as Discrete Cosine Transforms (DCT) or Discrete Wavelet Transforms (DWT) provide sparse approximations of smooth signals, compacting most of the information into a small number of coefficients However, classical transforms lose efficien y when they are applied to D-dimensional signals with large discontinuities. In such cases, *directional transforms*, which are able to adapt their basis functions to the underlying signal structure, can lead to better performance.

A graph-based signal representation allows us to generalize standard signal processing operations, such as filterin or transforms, to a broad class of D-dimensional signals [1]–[12]. In this way, there are many scenarios in which one can construct a graph to represent D-dimensional signals where weights reflec specifi relationships between samples (e.g., correlation, geometric distance or connectivity). In a video signal, each graph node can represent a pixel and links between nodes may capture similarity between luminance values. The motivation of this paper is to design a video encoder based on directional transforms constructed from graph-based representations, in which filterin operations are performed following directions of high correlation.

B. Related Work

The design of directional transforms has been an active research fiel in the past two decades. Representative examples are Curvelets [13], Contourlets [14], Bandelets [15], Directionlets [16] or directional DCTs [17], [18]. The lifting scheme [19] allows us to construct critically sampled transforms whose basis functions can be adapted to the signal structure in a simple way. To perform lifting, the input signal should be split into update ($\mathcal{U}$) and prediction ($\mathcal{P}$) samples and the update ($\mathbf{u}$) and prediction ($\mathbf{p}$) filter should be defined Then, in the prediction stage of the transform, $\mathcal{P}$ samples are predicted from $\mathcal{U}$ samples using $\mathbf{p}$ filter providing subsampled high-pass (detail coefficients versions of the signal, and $\mathcal{U}$ samples are updated from $\mathcal{P}$ detail coefficient using $\mathbf{u}$ filter giving rise to subsampled low-pass (smooth coefficients versions of the signal. If detail coefficient are close to zero, the main information is kept in the smooth coefficients thus obtaining a more compact representation. Applying this process iteratively on the smooth coefficient leads to a multiresolution analysis (MRA) [20] of the original signal. Due to its simplicity, some directional transforms based on lifting have been proposed in the literature for image [21]–[23] and video [24]–[29] representation.

The main multiresolution decomposition structures in wavelet-based video coding using lifting are referred to as "$t+2D$" and "$2D+t$". In the former, motion-compensated lifting steps are applied on the video sequence to implement the temporal wavelet transform, filterin along a set of motion trajectories described by a specifi motion model (an approach known as motion compensated temporal filtering MCTF).

Eduardo Martínez-Enríquez is with the Instituto de Óptica, Consejo Superior de Investigaciones Científica , Madrid, Spain (e-mail: emenriquez@tsc.uc3m.es). Jesús Cid-Sueiro and Fernando Díaz-de-María are with the Department of Signal Theory and Communications, Carlos III University, Madrid, Spain (e-mail: jcid@tsc.uc3m.es, fdiaz@tsc.uc3m.es). Antonio Ortega is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, (antonio.ortega@sipi.usc.edu).

This work was supported in part by NSF under grant CCF-1018977 and by Spanish Ministry of Economy and Competitiveness under grants TEC2014-53390-P and TEC2014-52289-R.

Then, a 2-dimensional wavelet transform is carried out in the spatial domain [24]. In the latter, each frame is firs wavelet transformed in the spatial domain, followed by MCTF. Representative examples of MCTF implementations are [25], [26] and [27]. These approaches can be described as separable because spatial and temporal filterin are applied in separate steps. Side information (e.g., motion vectors (MV)) is typically transmitted so that the decoder can identify the directional transform that was selected. In all of these works, in order to perform the prediction and update stages of the lifting scheme, the input sequence is split into $\mathcal{U}$ (even frames) and $\mathcal{P}$ (odd frames) subsequences, and for each level of the transform, the $\mathcal{P}$ subsequence is predicted from the $\mathcal{U}$ subsequence giving rise to the high-pass subband sequence, and the $\mathcal{U}$ subsequence is updated by using a filtere version of the $\mathcal{P}$ 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 $\mathcal{U}$ and $\mathcal{P}$ pixels with very different luminance values. In this way, $\mathcal{P}$ frames will be poorly predicted from $\mathcal{U}$ frames, leading to significan 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 filtere several times or are not filtere at all). Some authors handle this problem by identifying unconnected and multiple connected pixels and adapting the predict and update operators accordingly (e.g., [28]).

C. Contributions

In this paper we describe and optimize a video encoder based on lifting transforms on graphs. By construction, lifting on graphs [30]–[32] leads to a critically sampled and invertible transform, in contrast to other graph-based transforms [1], [6]–[8]. To this end, every node is labeled as $\mathcal{U}$ or $\mathcal{P}$, and only edges between $\mathcal{U}$ and $\mathcal{P}$ sets are used for filtering which is equivalent to findin a bipartite approximation to the graph.

The proposed scheme gives rise to a 3-dimensional (spatio-temporal) non-separable directional transform that can be viewed as an extension/generalization of wavelet transform-based video encoders that operate in the spatial and in the temporal domains independently. Thanks to the versatility of the proposed scheme, $\mathcal{U}$ and $\mathcal{P}$ nodes and filter can be arbitrarily chosen, solving some problems that arise in the MCTF approaches, e.g., multiply connected or disconnected pixels. This versatility provides a great freedom to choose filterin directions, which are define by means of the links between nodes on the graph, allowing the transform to adapt to the video content, thus improving its performance. Once the transform is defined we propose a coefficien reordering approach and an entropy coder, leading to a complete video encoder. On average, our proposed system achieves improvements of 1.24 dB with respect to a MCTF encoder [25] and 0.34 dB with respect to a simplifie encoder derived from H.264/AVC (reference software JM15.1 configure to use tools similar to those in the proposed encoder, i.e., 1 reference frame, no subpixel motion estimation, 16×16 inter and 4×4 intra modes), for a variety of standard QCIF and CIF video sequences. These improvements are more significan at high qualities, where they are in the range of 1 to 3 dBs with respect to the simplifie H.264/AVC video encoder, obtaining similar coding results in six out of twelve test sequences when comparing to JM15.1 configure allowing 5 reference frames, all the inter and intra modes available, and motion estimation similar to the proposed encoder (subpixel motion estimation disabled).

Previous work was presented by the authors in [33]–[35]. In this paper we formalize and solve analytically some of the underlying problems that arise from the transform design, describing and optimizing the complete scheme. Besides, we provide experimental results that compare different transform designs in terms of energy compaction ability and that justify our design choice. In order to obtain an improved complete encoder, we propose a reordering method and a new entropy coder, significantl improving previous versions of the encoder. Finally, we extend the results in [33]–[35] by including a comparison of the proposed encoder with MCTF and H.264/AVC video encoders for QCIF and CIF sequences.

The outline for the rest of the paper is as follows. In Section II we present the notation and some definition that will be used throughout the paper and we outline lifting transforms on graphs, motivating some optimization problems that are discussed in Section III, where we formally defin these problems and propose solutions. We also compare our solutions in terms of energy compaction to choose the best transform design. In Section IV we propose a complete video encoder based on lifting transforms on graphs, discussing the quantization, reordering, and entropy coding, as well as the side information that is sent to the decoder. In Section V we provide experimental results that prove the effica y of the proposed video encoder in comparison to H.264/AVC and a MCTF video encoder. Finally, in Section VI we draw some conclusions and propose some directions for future research.

II. Preliminaries

A. Notation

A graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ is a set of nodes (or vertices) and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ a set of edges (or links) between nodes. In the present work we consider arbitrary, undirected, edge-weighted graphs, denoting with $w_{mn} \in \mathbb{R}^+$ the weight of the edge $mn \in \mathcal{E}$ ($w_{mn} = 1$ for unweighted graphs). The order of the graph $N = |\mathcal{V}|$ is the number of nodes of the graph. $\mathcal{N}_m = \{n \in \mathcal{V} : mn \in \mathcal{E}\}$ is the set of neighbors of node m, and $\mathcal{N}_{[m]}$ is its closed neighborhood set ($\mathcal{N}_{[m]} = \mathcal{N}_m \cup m$). The degree of a node m, D_m, is the sum of weights of all its incident edges (i.e., the number of neighbors if the graph is unweighted), $D_m = \sum_{n \in \mathcal{N}_m} w_{mn}$.

We use the index i for nodes in set $\mathcal{P}$, k for nodes in set $\mathcal{U}$, and j for indexing the level of the transform, while m and n are general indexes.

B. Classical Graph-Partition Problems

As previously discussed, lifting requires findin a bipartition of the graph, splitting the node set into two disjoint subsets

$\mathcal{U}$ and $\mathcal{P} := \mathcal{V} \backslash \mathcal{U}$, which is called a cut in graph theory. The weight of the cut (W) is given by the function

$$W(\mathcal{U}, \mathcal{P}) = \sum_{i \in \mathcal{P}, k \in \mathcal{U}} w_{ik}. \quad (1)$$

The *weighted maximum-cut* (WMC) *problem* can be define as, given a weighted graph $\mathcal{G}$, findin the cut of maximum weight:

$$\text{WMC}(\mathcal{G}) = \max_{\forall \mathcal{U} \subseteq \mathcal{V}} W(\mathcal{U}, \mathcal{P}). \quad (2)$$

If $\mathcal{G}$ is an unweighted graph, the WMC problem will be referred to as the maximum-cut (MC) problem.

Another interesting approach is to fin a bipartition of the graph so that: (i) every node of one of the subsets has at least one neighbor in the other subset and (ii) one of the subsets has the minimum possible number of nodes. This can be achieved by applying the classical *set-covering* (SC) *problem* to the collection of sets of closed neighbors of a graph. More formally, given a collection $\mathcal{M}$ of all sets $\mathcal{N}_{[n]}$, $n \in \mathcal{V}$, a set-cover $\mathcal{C} \subseteq \mathcal{M}$ is a subcollection of the sets whose union is $\mathcal{V}$, and the goal of the SC problem is findin a minimum-cardinality set-cover $m\mathcal{C}$ such that $m\mathcal{C} = \left\{\mathcal{N}_{[n_m]}\right\}_{m \in 1,2,\ldots,l}$. The corresponding cut arises naturally: we can denote set $\{n_m\}_{m \in 1,2,\ldots,l}$ as $\mathcal{U}$ nodes and the remaining as $\mathcal{P}$ nodes ($\text{SC}_{\mathcal{U}}$) or vice-versa ($\text{SC}_{\mathcal{P}}$).

C. Graph-Based Representation of a Generic Signal

Definitio II.1. Graph-based signal representation

Let $\mathbf{x} = [x_1, x_2, \ldots, x_m, \ldots, x_N]$ be a signal define in a D-dimensional space (where $x_m \in \mathbb{R}$ is define at position $\mathbf{c}_m \in \mathbb{R}^D$) whose samples have been placed in a vector in arbitrary order. Assume that data are organized in an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ so that x_m is the value on node $m \in \mathcal{V}$, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of edges between nodes. Note that node $a \in \mathcal{V}$ can be linked to any subset of nodes $\mathcal{F} \subset \{\mathcal{V} \backslash \{a\}\}$ without restrictions. This leads to a graph-based signal representation, $\mathcal{G}_{\mathbf{x}}$, of $\mathbf{x}$.

Throughout this paper, edges between nodes capture similarity between them (i.e., nodes a and b are linked if x_a and x_b are similar enough). Similarity is derived from sample position in D-dimensional space (e.g., pixels that are neighbors in an image) and other information, such as motion and presence of image contours[1].

D. Lifting Transforms on Arbitrary Graphs

Lifting transforms on arbitrary graphs were initially proposed by [30], [31] and [32]. Given a signal $\mathbf{x}$ define on an arbitrary undirected graph, $\mathcal{G}_{\mathbf{x}} = (\mathcal{V}, \mathcal{E})$, lifting is specifie by three main stages (see Figure 1): (i) a *split stage*[2], which find a bipartition of the graph so that the input node set at each specifi level of decomposition j (s_{j-1}) is split into *prediction* ($\mathcal{P}_j$) and *update* ($\mathcal{U}_j$) sets; (ii) a *prediction stage*, where every sample $s_{i,j-1} \in \mathcal{P}_j$ is predicted from an arbitrary number of $\mathcal{U}_j$ neighbors using the $\mathbf{p}_{i,j}$ filte , yielding the detail coefficien $d_{i,j}$; and (iii) an *update stage*, where every sample $s_{k,j-1} \in \mathcal{U}_j$ is filtere with the $\mathbf{u}_{k,j}$ filte using $s_{k,j-1}$ and an arbitrary number of $\mathcal{P}_j$ neighbor detail coefficients giving rise to the smooth coefficien $s_{k,j}$. Mathematically, lifting on graphs can be written as:

$$\begin{aligned} d_{i,j} &= s_{i,j-1} - \sum_{k \in \mathcal{N}_{i,j} \cap \mathcal{U}_j} p_{i,k,j}\ s_{k,j-1} = s_{i,j-1} - \hat{s}_{i,j-1}, \\ s_{k,j} &= s_{k,j-1} + \sum_{i \in \mathcal{N}_{k,j} \cap \mathcal{P}_j} u_{k,i,j}\ d_{i,j}, \end{aligned} \quad (3)$$

where $p_{i,k,j}$ (resp. $u_{k,i,j}$) is the value of the $k-th$ (resp. $i-th$) position in the $\mathbf{p}_{i,j}$ (resp. $\mathbf{u}_{k,j}$) filte . Note that inverting the operations of the forward transform to obtain the inverse transform is straightforward from (3) as long as only connections between $\mathcal{U}$ and $\mathcal{P}$ nodes are used for filtering

Lifting transforms on graphs can operate with arbitrary graphs, $\mathcal{P}_j/\mathcal{U}_j$ disjoint splittings and $\mathbf{p}_j$ and $\mathbf{u}_j$ filte designs without compromising the perfect reconstruction and critically sampled properties of the transform [36]. This fl xibility in the design makes the choice of the transform parameters a crucial task in order to achieve an efficien transformation. In Section III we focus on addressing the following questions:

- How should the graphs be constructed to capture the correlation of the signal? (Section III-A).
- How should the $\mathbf{p}$ and $\mathbf{u}$ filter be defined (Section III-B).
- How should the $\mathcal{U}/\mathcal{P}$ splitting be performed? (Section III-C).
- How should the graphs be constructed at decomposition level $j > 1$? (Section III-D).

Finally, note that throughout the next sections we focus the explanation on the firs level of the transform, so that $\mathbf{x} = \mathbf{s}_{j=0}$ is the raw data, $\mathcal{G}_{\mathbf{x}}$ its graph representation, and detail coefficient (3) are written as $d_i = x_i - \sum_{k \in \mathcal{N}_i \cap \mathcal{U}} p_{i,k}\ x_k = x_i - \hat{x}_i$. Nevertheless, all the described processes can be easily extended to any level j.

III. Lifting Transforms on graphs for Video Coding

A. Graph Construction

The graph construction includes the graph-based signal representation and the graph weighting. Observe that, by Definitio II.1, there exist several $\mathcal{G}_{\mathbf{x}}$ for the same $\mathbf{x}$ depending on the way in which the similarity between nodes is defined Furthermore, given that the filterin operations are performed using neighboring (linked) nodes, $\mathcal{G}_{\mathbf{x}}$ define the filterin directions.

1) Graph-Based Representation of a Video Signal: Let $\mathbf{x}$ be a given video sequence where x_m refers to the luminance value of pixel m, belonging to a specifi frame and spatial position. Let $\mathcal{G}_{\mathbf{x}} = (\mathcal{V}, \mathcal{E})$ be its graph representation, where links between nodes can be spatial ($\mathcal{S}$) or temporal ($\mathcal{T}$) so that $\mathcal{S} \cup \mathcal{T} = \mathcal{E}$.

[1]To avoid confusion we call image "contours" edges that appear in the image between sets of pixels of different intensities, while we reserve the term "edge" for the links between nodes in a graph.

[2]The split stage of the transform will be referred to as $\mathcal{U}/\mathcal{P}$ assignment or graph bipartition problem throughout this paper.

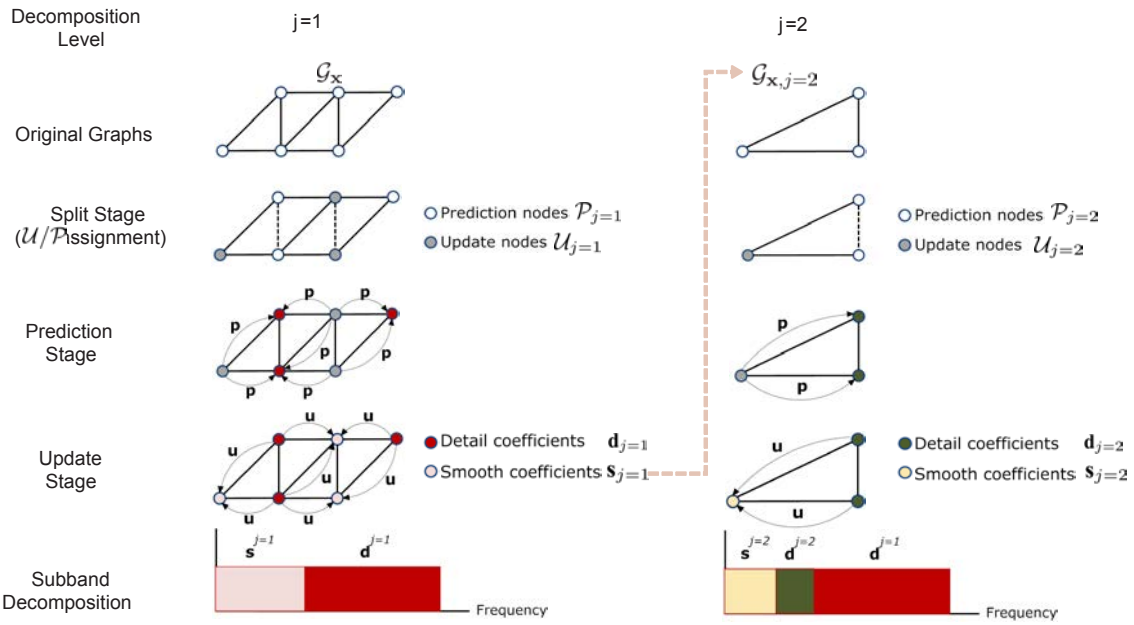


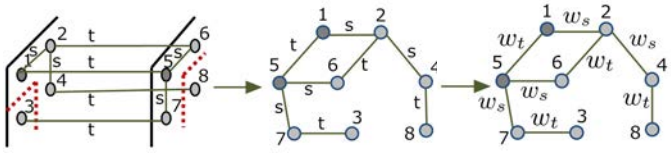
Fig. 1: Example of the lifting scheme applied to a graph. Two levels of decomposition of the forward transform. Discarded links (links between same-label neighbors, not used to perform the filtering are indicated with dashed lines in the split process. Left column shows the corresponding stages of the transform at decomposition level $j = 1$, and right column at $j = 2$. Last row represents the schematic subband decomposition obtained for each j.

In a firs example of the graph representation of a video signal, every pixel is linked (i) to temporal neighbors following a motion estimation (ME) process and (ii) to 8 one-hop spatial neighbors (i.e., pixels of the same frame), assuming that spatial neighboring pixels will have similar luminance values. In this example ME is performed findin the best-matching block on the previous (reference) frame, so that a pixel in frame t is linked to the pixel that it points to in frame $t-1$ and, possibly, to one or more pixels in frame $t+1$ that use it as a reference (i.e., if one or more blocks in frame $t+1$ points to this pixel in frame t).

A reasonable approach to improve the graph representation could be to remove links between spatial neighboring pixels that cross contours of an image (frame) assuming that they will have very different luminance values. This gives rise to the graph representation illustrated in Figure 2(a), where red dashed line represents a contour within a frame. Finally it should be noticed that the encoder should send some side information to allow the decoder to correctly construct the same graph. Therefore, a trade-off exists between accuracy in the graph description and side information to be sent (e.g., using smaller block sizes in ME leads to more accurate graphs, but more side information has to be sent).

2) Graph Weighting: Similarities between nodes depend on the nature of links between them (i.e., spatial or temporal) and on the specifi $\mathcal{G}_{\mathbf{x}}$ used for the signal at hand. In the particular case of a video signal, it is natural to assign a specifi weight to every spatial link and another weight to every temporal link. We fin the optimal graph weights that minimize the quadratic prediction error (assuming one-hop prediction filter define below) for a given $\mathcal{G}_{\mathbf{x}}$.

Let $\mathcal{G}_{\mathbf{x}} = (\mathcal{V}, \mathcal{E})$ be the graph-based representation of a video signal, with $\mathcal{S}, \mathcal{T}$ the set of spatial and temporal



(a) *Graph representation* (b) *Graph weighting*

Fig. 2: Graph construction of a video signal: (a) Graph representation removing those spatial links that cross the contours of a frame; (b) graph weighting.

edges, respectively. Let $\mathcal{N}_m^s = \{n : mn \in \mathcal{S}\}$ (resp. $\mathcal{N}_m^t = \{n : mn \in \mathcal{T}\}$) denote one-hop spatial (resp. temporal) neighborhood of m, for all nodes $m \in \mathcal{V}$.

Thus, the mean values of the spatial and the temporal neighbors of node m are define as

$$\bar{x}_m^s = \frac{1}{|\mathcal{N}_m^s|} \sum_{n \in \mathcal{N}_m^s} x_n, \qquad (4)$$

$$\bar{x}_m^t = \frac{1}{|\mathcal{N}_m^t|} \sum_{n \in \mathcal{N}_m^t} x_n,$$

where $|\mathcal{N}_m^s|$ (resp. $|\mathcal{N}_m^t|$) is the number of spatial (resp. temporal) neighbors of m. Let us assume that every node $m \in \mathcal{V}$ is linearly predicted from its spatial and temporal neighbors as:

$$\hat{x}_m = w_s \bar{x}_m^s + w_t \bar{x}_m^t. \qquad (5)$$

Then, our problem becomes:

Problem III.1. ***Optimal Weighting Problem Formulation.***

Find the weights w_s and w_t that minimize the quadratic prediction error over all the nodes $m \in \mathcal{V}$:

$$\min_{w_s,w_t} \sum_{m\in\mathcal{V}} \left(x_m - w_s \bar{x}_m^s - w_t \bar{x}_m^t\right)^2. \tag{6}$$

Differentiating with respect to w_s and w_t we obtain the classical least-squares solution:

$$\mathbf{w}^* = [w_s^*, w_t^*] = \mathbf{R}^{-1}\mathbf{r}, \tag{7}$$

where

$$\mathbf{R} = \begin{bmatrix} \sum_{m\in\mathcal{V}} \bar{x}_m^s \bar{x}_m^s & \sum_{m\in\mathcal{V}} \bar{x}_m^s \bar{x}_m^t \\ \sum_{m\in\mathcal{V}} \bar{x}_m^t \bar{x}_m^s & \sum_{m\in\mathcal{V}} \bar{x}_m^t \bar{x}_m^t \end{bmatrix} \tag{8}$$

and

$$\mathbf{r} = \sum_{m\in\mathcal{V}} x_m \begin{bmatrix} \bar{x}_m^s \\ \bar{x}_m^t \end{bmatrix} \tag{9}$$

are the correlation matrices. It should be noted that $\mathbf{R}$ is a positive semidefinit matrix and $\mathbf{R^{-1}}$ is define for any $\mathcal{G}_\mathbf{x}$ constructed from a non-constant $\mathbf{x}$.

The graph topology can be described by its adjacency matrix, so that we can express the optimal weights as a function of spatial and temporal adjacency matrices of the graph. Let $\overline{\mathbf{A}}_\mathbf{s} = [\overline{a}_{s_{mn}}]$ and $\overline{\mathbf{A}}_\mathbf{t} = [\overline{a}_{t_{mn}}]$ be the adjacency matrices of the subgraphs containing only the spatial and temporal edges, respectively, where each column is normalized (i.e., $\overline{a}_{s_{mn}} = 1/|\mathcal{N}_n^s|$ if $mn \in \mathcal{S}$; $\overline{a}_{s_{mn}} = 0$ if $mn \notin \mathcal{S}$). Vectorizing the sequence into a $1 \times (L \times H \times K)$ row vector $\mathbf{x}$, where $L \times H$ is the frame size and K the number of frames considered, we can write:

$$\mathbf{w}^* = \begin{bmatrix} \mathbf{x}\overline{\mathbf{A}}_\mathbf{s}\overline{\mathbf{A}}_\mathbf{s}^\mathbf{T}\mathbf{x}^\mathbf{T} & \mathbf{x}\overline{\mathbf{A}}_\mathbf{s}\overline{\mathbf{A}}_\mathbf{t}^\mathbf{T}\mathbf{x}^\mathbf{T} \\ \mathbf{x}\overline{\mathbf{A}}_\mathbf{t}\overline{\mathbf{A}}_\mathbf{s}^\mathbf{T}\mathbf{x}^\mathbf{T} & \mathbf{x}\overline{\mathbf{A}}_\mathbf{t}\overline{\mathbf{A}}_\mathbf{t}^\mathbf{T}\mathbf{x}^\mathbf{T} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{x}\overline{\mathbf{A}}_\mathbf{s}\mathbf{x}^\mathbf{T} \\ \mathbf{x}\overline{\mathbf{A}}_\mathbf{t}\mathbf{x}^\mathbf{T} \end{bmatrix}, \tag{10}$$

where the symbol T denotes transposition.

Once $\mathbf{w}^*$ has been calculated, we assign w_s^* (resp. w_t^*) to every spatial (resp. temporal) link, leading to a weighted $\mathcal{G}_\mathbf{x}$ that accurately captures the correlation between nodes. Figure 2 illustrates the graph construction process, including the graph representation and graph weighting.

Note that the correlation between temporal and spatial neighbors changes with the video content, and thus the value of the optimal graph weights would change as well. The optimal weights can be computed for any subgraph $\mathcal{H} \subseteq \mathcal{G}_\mathbf{x}$ (i.e., their value can change for every subgraph and thus with the video content) with the formulation given above. For a video signal, optimal weights can be computed, for example, between two consecutive frames (i.e., weights between pixels in frames t and $t-1$ are computed and used, then weights between $t+1$ and t, and so on). Given that the weights should be sent to the decoder as side information, a trade-off exists between accuracy in the weights selection (lower $\mathcal{H}$ sizes) and side information to be sent.

B. Graph-Based Filter Design

In this section we firs focus on the prediction filte design assuming a given weighted graph $\mathcal{G}_\mathbf{x}$ for which a bipartition (i.e., $\mathcal{U}/\mathcal{P}$ assignment) has been chosen. Then, we describe the update filte design, which is based on the methods proposed by [37] and [38].

1) Prediction Filter: The problem of optimizing prediction filter in lifting transforms has been considered by several authors, typically based on optimization criteria that seek to minimize the expected energy of the detail coefficient [39]–[43]. Given that the graph topology can be locally different, we design prediction filter in a natural way from the graph weights, which were optimized in order to minimize the one-hop prediction error (i.e., the energy of the detail coefficients)

The general expression for the prediction filte at node $i \in \mathcal{P}$ is:

$$\mathbf{p}_i = \frac{[p_{i,1}\ , p_{i,2}\ , \ldots , p_{i,k}\ , \ldots , p_{i,m_i}]}{\sum_{k=1}^{m_i} p_{i,k}}, \tag{11}$$

where $p_{i,k}$ is the prediction coefficien associated with neighbor node $k \in \mathcal{N}_i \cap \mathcal{U}$ and m_i is the number of $\mathcal{U}$ neighbors of i. The normalization factor is important when $\sum_{k=1}^{m_i} p_{i,k} \neq 1$ (e.g., to defin prediction filter in higher levels of the transform, where $p_{i,k}$ at j is calculated as the product of the weights in the path between connected nodes at $j-1$, as will be explained in Section III-D). Prediction coefficient $p_{i,k}$ are define from weights of the graph as follows:

Definitio III.1. Prediction filte for unweighted $\mathcal{G}_\mathbf{x}$

Consider a given unweighted $\mathcal{G}_\mathbf{x}$ and $\mathcal{U}/\mathcal{P}$ assignment. In this case, we defin $p_{i,k} = 1/m_i$, leading to unweighted predictors:

$$\hat{x}_i = \sum_{k\in\mathcal{N}_i\cap\mathcal{U}_j} p_{i,k}\ x_k = \frac{1}{m_i} \sum_{k\in\mathcal{N}_i\cap\mathcal{U}} x_k. \tag{12}$$

Definitio III.2. Prediction filte for weighted $\mathcal{G}_\mathbf{x}$

Consider a weighted $\mathcal{G}_\mathbf{x}$ as define in Section III-A, where every link can be spatial ($\mathcal{S}$) or temporal ($\mathcal{T}$), with weights w_s and w_t[3], respectively. Let us defin m_i^s (resp. m_i^t) as the number of $\mathcal{U}$ spatial (resp. temporal) neighbors of $i \in \mathcal{P}$ (i.e., $m_i^s = |\mathcal{N}_i^s \cap \mathcal{U}|$ and $m_i^t = |\mathcal{N}_i^t \cap \mathcal{U}|$). Normalizing the weights by m_i^s and m_i^t, respectively, $p_{i,k}$ is obtained as:

$$p_{i,k} = \begin{cases} w_s/m_i^s, & \text{if } ik \in \mathcal{S}, \\ w_t/m_i^t, & \text{if } ik \in \mathcal{T}. \end{cases} \tag{13}$$

This leads to spatio-temporal weighted predictors define as

$$\hat{x}_i = \sum_{k\in\mathcal{N}_i\cap\mathcal{U}_j} p_{i,k}\ x_k = \frac{w_s}{m_i^s} \sum_{k\in\mathcal{N}_i^s\cap\mathcal{U}} x_k + \frac{w_t}{m_i^t} \sum_{k\in\mathcal{N}_i^t\cap\mathcal{U}} x_k. \tag{14}$$

[3]Without loss of generality, throughout the rest of the paper we assume that $w_s + w_t = 1$, which is the case in most of the examples. If this assumption is not considered, the weight values in the Formulas of the paper should be divided by $(w_s + w_t)$.

2) Update Filter: For each update node we design an update filte that is orthogonal to the prediction filter of its prediction neighbors. The general expression for the update filte at node $k \in \mathcal{U}$ is:

$$\mathbf{u}_k = [u_{k,1}\ , u_{k,2}\ , \ldots , u_{k,i}\ , \ldots , u_{k,m_k}]\,, \tag{15}$$

where $u_{k,i}$ is the update coefficien associated with neighbor node $i \in \mathcal{N}_k \cap \mathcal{P}$ and m_k is the number of $\mathcal{P}$ neighbors of k. Let $\mathbf{P}_k$ be an $N \times m_k$ matrix having the prediction vectors of nodes $i \in \mathcal{N}_k \cap \mathcal{P}$ as its columns. Let $\mathbf{p}_k^*$ the vector containing the elements of row k in matrix $\mathbf{P}_k$. Orthogonal $\mathbf{u}_k$ filte is obtained as:

$$\mathbf{u}_k^T = -\left(\mathbf{P}_k^T \mathbf{P}_k\right)^{-1} \mathbf{p}_k^{*T}. \tag{16}$$

It can be shown [38] that $\mathbf{P}_k^T \mathbf{P}_k$ is invertible and that filter define as (16) always exist. While the resulting update filter are not orthogonal to all the prediction filters this solution reduces the impact of the "worst-case" coherence, because the prediction filter centered in prediction nodes that are not neighbors have little or no common support with the given update filte . Other approaches for update filte design can be found in the literature [44].

C. $\mathcal{U}/\mathcal{P}$ Assignment

Assuming we have a weighted $\mathcal{G}_{\mathbf{x}}$ and $\mathbf{p}$ and $\mathbf{u}$ filter definition this section describes how to split graph nodes into two disjoint sets $\mathcal{U}$ and $\mathcal{P}$. We discuss two different approaches: coloring-based $\mathcal{U}/\mathcal{P}$ assignment, which fin a bipartition by solving "classical" graph partition problems described in Section II-B, and model-based $\mathcal{U}/\mathcal{P}$ assignment, which minimizes the expected value of the quadratic prediction error (i.e., the detail coefficien energy) assuming a signal model and a predictor.

1) Coloring-based $\mathcal{U}/\mathcal{P}$ Assignment: As we proposed in [33], [34], one good solution to address the $\mathcal{U}/\mathcal{P}$ assignment is by solving the WMC problem. The underlying idea is to maximize the reliability with which update nodes can predict prediction neighbors, which intuitively seems equivalent to maximize the total weight of the links between the $\mathcal{P}$ and $\mathcal{U}$ sets. An alternative approach for $\mathcal{U}/\mathcal{P}$ assignment is by solving *set-covering* (SC) problems. In particular, the $\mathrm{SC}_{\mathcal{U}}$ solution involves obtaining the minimum number of $\mathcal{U}$ nodes that guarantees that every $\mathcal{P}$ node has at least one $\mathcal{U}$ neighbor and thus can be predicted. This leads to a large number of $\mathcal{P}$ nodes in which the signal is decorrelated which would have, in general, a low number of $\mathcal{U}$ neighbors. On the other hand, the $\mathrm{SC}_{\mathcal{P}}$ solution involves having a low number of $\mathcal{P}$ nodes with many $\mathcal{U}$ neighbors.

2) Signal model-based $\mathcal{U}/\mathcal{P}$ Assignment: Assume that video signals are modeled considering (i) smooth noise variations between neighbors on the graph and (ii) that spatial and temporal neighbor pixels may have different correlations. Under these assumptions, we propose the Spatio-Temporal Model (STM), where every pixel x_m is generated as:

Definitio III.3. Spatio-Temporal Model

$$x_m = \left(\frac{w_s}{|\mathcal{N}^s_{[m]}|} \sum_{n \in \mathcal{N}^s_{[m]}} \epsilon_n + \frac{w_t}{|\mathcal{N}^t_{[m]}|} \sum_{n \in \mathcal{N}^t_{[m]}} \epsilon_n \right) + \eta_m, \tag{17}$$

where $\mathcal{N}^s_{[m]}$ and $\mathcal{N}^t_{[m]}$ are the closed sets of spatial and temporal neighbors, respectively, of node m; w_s and w_t are the graph weights; and ϵ_n and η_m are zero-mean independent random variables with variances v_{ϵ_n}, and v_{η_m}, respectively.

Given the predictor $\hat{x}_i$ define in Definitio III.2, our goal is to solve the next problem:

Problem III.2. $\mathcal{U}/\mathcal{P}$ ***Assignment Problem:***

Find the $\mathcal{U}/\mathcal{P}$ assignment that minimizes the total prediction error given by

$$E_{tot} = \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\} \tag{18}$$

for a given number of $\mathcal{P}$ nodes, $|\mathcal{P}|$:

$$\min_{\mathcal{U}/\mathcal{P}} E_{tot},\ \textit{subject to}\ |\mathcal{P}| = T. \tag{19}$$

Fixing $|\mathcal{P}|$ in the problem formulation is important because E_{tot} is minimized by minimizing the size of $\mathcal{P}$. Thus, solving (19) is practical only if some constraint on the size of $\mathcal{P}$ is introduced. In practice, one can fi $|\mathcal{P}| = |\mathcal{V}|/2$ in order to obtain a dyadic decomposition of the graph similar to the one obtained in classical wavelets.

It can be proven [45] that, considering that $v_{\eta_m} = v_\eta$ and that $v_{\epsilon_m} = v_\epsilon$ for any $m \in \mathcal{V}$, the prediction error of i is given by

$$\begin{aligned} E_{\mathrm{ST}_i} &= \mathbb{E}\{(x_i - \hat{x}_i)^2\} = \mathbb{E}\{(x_i)^2\} + \mathbb{E}\{(\hat{x}_i)^2\} - 2\mathbb{E}\{x_i \hat{x}_i\} \\ &= v_\eta + v_\epsilon \left(\frac{w_s^2}{|\mathcal{N}^s_{[i]}|} + \frac{w_t^2}{|\mathcal{N}^t_{[i]}|} + \frac{2 w_s w_t}{|\mathcal{N}^s_{[i]}||\mathcal{N}^t_{[i]}|} \right) \\ &+ v_\eta \left(\frac{w_s^2}{m_i^s} + \frac{w_t^2}{m_i^t} \right) \\ &+ v_\epsilon \left(\frac{w_s^2}{(m_i^s)^2} G_i + \frac{w_t^2}{(m_i^t)^2} H_i + \frac{2 w_s w_t}{m_i^t m_i^s} I_i \right) \\ &- 2 v_\epsilon \left(\frac{w_s}{m_i^s} J_i + \frac{w_t}{m_i^t} K_i \right), \end{aligned} \tag{20}$$

where terms G_i, H_i and I_i are closely related with the correlation between $\mathcal{U}$ neighbors of node i, and J_i and K_i with the correlation between node i and its $\mathcal{U}$ neighbors. The main idea behind (20) is that, for node i, the expected value of the prediction error decreases when i has a large number of correlated $\mathcal{U}$ neighbors, which is quite reasonable from the prediction theory point of view. Furthermore, correlation between nodes increases with the value of the weight between them and with the proportion of shared neighbors on the graph.

Summarizing, the optimal $\mathcal{U}/\mathcal{P}$ assignment (i.e., the one that minimizes $\sum_{i \in \mathcal{P}} E_{\mathrm{ST}_i}$) under the assumed model and predictor, depends on the weight values and the graph topology. In

Section III-E we obtain prediction error results for the STM by using a greedy algorithm that locally minimizes (20) in each iteration.

D. Extending the Transform to Multiple Levels of Decomposition

In order to carry out a multiresolution analysis, the low pass coefficient 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 $\mathcal{U}$ nodes that are either (i) directly connected or (ii) at two-hop of distance in the graph at level $j-1$, so that the simplifie graph continues to capture the correlation between pixels. If the link exists at level $j-1$ then the corresponding link at level j inherits the same weight. Alternatively, if two nodes are linked that were two hops away at level $j-1$ then the corresponding link weight is the product of the weights in the path between connected nodes at level $j-1$. Note that if there exist multiple two-hop paths between the two nodes, the resulting weight is the highest among available paths (maximum similarity). Once we have constructed the graph at level j, we should split the nodes again into prediction ($\mathcal{P}_j$) and update ($\mathcal{U}_j$) disjoint sets in order to perform the transform. Algorithm 1 shows the implementation of the graph construction at level j from the graph at level $j-1$ and Figure 3 shows an illustrative example with the $\mathcal{U}/\mathcal{P}$ assignment at both transformation levels.

Algorithm 1 $\mathcal{G}_{\mathbf{x},j}$ construction from $\mathcal{G}_{\mathbf{x},j-1}$.

Require: $\mathcal{U}_{j-1}/\mathcal{P}_{j-1}$, $\mathcal{G}_{\mathbf{x},j-1} = (\mathcal{V}_{j-1}, \mathcal{E}_{j-1})$,$\mathbf{W}_{j-1} = [w_{mn,j-1}]$
$\mathcal{E}_j = \{\varnothing\}$, $\mathcal{V}_j = \mathcal{U}_{j-1}$
for $\forall m \in \mathcal{U}_{j-1}$ **do**
  for $\forall n \in \mathcal{N}_{m,j-1}$ **do**
    if $n \in \mathcal{U}_{j-1}$ **then**
      $\mathcal{E}_j \leftarrow mn$
      $w_{mn,j} \leftarrow w_{mn,j-1}$
    else if $n \in \mathcal{P}_{j-1}$ **then**
      for $\forall l \in \mathcal{N}_{n,j-1} \backslash m$ **do**
        if $l \in \mathcal{U}_{j-1}$ **then**
          $\mathcal{E}_j \leftarrow ml$
          $w_{ml,j} \leftarrow w_{mn,j-1} w_{nl,j-1}$
        end if
      end for
    end if
  end for
end for
return $\mathcal{G}_{\mathbf{x},j} = (\mathcal{V}_j, \mathcal{E}_j)$, $\mathbf{W}_j = [w_{mn,j}]$

E. Evaluation of Different Transform Designs

Different transform designs discussed previously are compared in terms of compaction ability (energy of detail coeffi cients in the firs level of the transform ($j = 1$)). This will give some insight about the importance of the different processes and optimizations and will allow us to select the best design to our purposes. Specificall , we evaluate: (i) two different graph

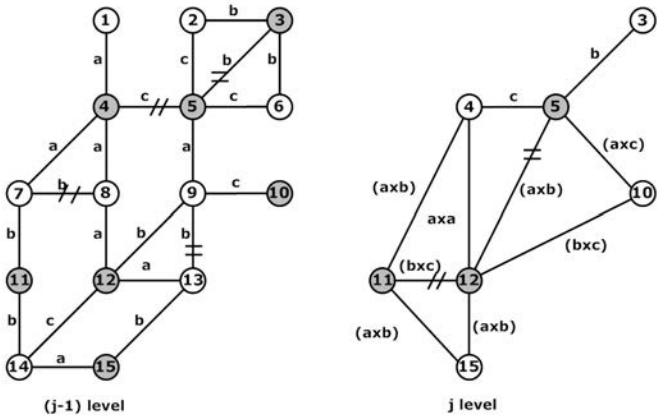


Fig. 3: Graph construction and $\mathcal{U}/\mathcal{P}$ assignment for consecutive levels of decomposition. $a = 10$, $b = 5$ and $c = 3$ are the different weight values. Grey nodes are $\mathcal{U}$ nodes, and white ones are $\mathcal{P}$ nodes. Discarded links (links between same-label neighbors) are indicated as broken links.

representations, contours connected $\mathcal{G}_{\mathbf{x}}$ (CCG) and contours disconnected $\mathcal{G}_{\mathbf{x}}$ (CDG), define below; (ii) two different $\mathbf{p}$ filters obtained from unweighted $\mathcal{G}_{\mathbf{x}}$ (unw) as in Definitio III.1 and from optimal weighted $\mathcal{G}_{\mathbf{x}}$ ($\mathbf{w}^*$) as in Definitio III.2; and (iii) different $\mathcal{U}/\mathcal{P}$ assignment strategies investigated in Section III-C, namely: $\text{SC}_{\mathcal{U}}$, $\text{SC}_{\mathcal{P}}$ and MC for unweighted $\mathcal{G}_{\mathbf{x}}$, and STM and WMC for weighted $\mathcal{G}_{\mathbf{x}}$.

In both, CCG and CDG, every node is firstl connected to its 8 one-hop spatial neighbors and to an arbitrary number of temporal neighbors following a ME model. The difference is that in CDG links between neighbor nodes across contours are disconnected. To obtain the $\text{SC}_{\mathcal{U}}$ and $\text{SC}_{\mathcal{P}}$ we use the greedy approach described in [45]; WMC algorithm is described in Algorithm 2, where $\mathcal{U}_j$ and $\mathcal{P}_j$ form a bipartition of the node set $\mathcal{U}_{j-1}$. Note that, if the given $\mathcal{G}_{\mathbf{x}}$ is unweighted, the algorithm provides the MC solution; and STM algorithm is described in [45]. The experiments have been carried out using subgraphs of real video data extracted from different standard test sequences.

Algorithm 2 *Weighted Maximum-Cut* Algorithm

Require: $\mathcal{G}_{\mathbf{x}}$, $\mathcal{U}_j = \{\varnothing\}$, $\mathcal{P}_j = \{\mathcal{U}_{j-1}\}$
Calculate the Degree of the $\mathcal{U}_{j-1}$ node set: $\mathbf{D} = \left[D_1, \ldots, D_k, \ldots, D_{|\mathcal{U}_{j-1}|}\right]$, $k \in \mathcal{U}_{j-1}$
Select the node a with largest Degree: $D_a = \max(\mathbf{D})$
while $D_a > 0$ **do**
  Let $\mathcal{U}_j \leftarrow \mathcal{U}_j \cup \{a\}$
  Let $\mathcal{P}_j \leftarrow \mathcal{P}_j \backslash \{a\}$
  Change the sign of the weights of the incident edges to a
  Update Degrees of adjacent nodes to a
  Select the node a with largest Degree, $D_a = \max(\mathbf{D})$
end while
return $\mathcal{U}_j$ and $\mathcal{P}_j$

To evaluate the performance of each approach, we measure

the average prediction error over all nodes in $\mathcal{P}$ as:

$$E_{\text{ap}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(x_i - \hat{x}_i)^2 = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(d_i)^2, \tag{21}$$

where x_i is the actual luminance value of the pixels, and $\hat{x}_i$ the prediction.

Figure 4 shows experimental results where E_{ap} is plotted as a function of the value $|\mathcal{U}|/N$ corresponding to the different configurations/transfor designs. Note that the number of $\mathcal{U}/\mathcal{P}$ nodes is fi ed for the WMC, MC, SC$_{\mathcal{U}}$ and SC$_{\mathcal{P}}$ solutions, and varies in the STM approach (by letting the given $|\mathcal{P}|$ in Problem III.2 vary). This is because WMC, MC, SC$_{\mathcal{U}}$ and SC$_{\mathcal{P}}$ have unique solutions that give rise to a specifi $\mathcal{U}/\mathcal{P}$ bipartition (and thus a specifi number and location of $\mathcal{U}$ and $\mathcal{P}$ nodes).

As expected, E_{ap} generally decreases as the number of $\mathcal{U}$ nodes increases, because better predictions are obtained. Furthermore, as the SC$_{\mathcal{U}}$ solution involves having a large number of $\mathcal{P}$ nodes with a low number of $\mathcal{U}$ neighbors, the prediction will not usually be so accurate, and the mean energy of detail coefficient E_{ap} will be large as is shown in Figure 4. On the other hand, the SC$_{\mathcal{P}}$ solution implies having accurate predictions (e.g., the E_{ap} will be low) but a small number of detail coefficient in which data is decorrelated.

E_{ap} is consistently much lower when using CDG instead of CCG for the same predictors and $\mathcal{U}/\mathcal{P}$ strategy, which means that including the directional information in the spatial domain removing links between nodes across contours helps to improve the prediction and thus to decrease the detail coefficien energy. For that purpose, we need to estimate the contours and send this information to the decoder.

E_{ap} obtained with optimal weighted CDG is lower than with unweighted CDG, so it can be concluded that it is important to take into account that temporal and spatial linked neighbors usually have different correlations and thus graph weight should be different. Therefore, we weight CDG following the process explained in Section III-A2, calculating $\mathbf{w}^*$ every two consecutive frames, and use the $\mathbf{p}$ filter define from the graph weights (Definitio III.2) and $\mathbf{u}$ filter described in Section III-B2.

Regarding the $\mathcal{U}/\mathcal{P}$ assignment process, we observe that using STM leads to lower detail coefficien energy for a given number of $|\mathcal{P}|$ nodes than WMC. Nevertheless, WMC obtains reasonably good results that are close to the STM solution with simpler greedy algorithms and thus with lower computational cost. Summarizing, given the lower computational cost and the near-optimal performance of the WMC, we use it as criterion to perform the $\mathcal{U}/\mathcal{P}$ assignment in every level of the transform j. An example of the WMC $\mathcal{U}/\mathcal{P}$ assignment for two levels of decomposition is shown in Figure 3.

IV. Complete Video Encoder

Once the transform has been discussed, in this section we describe in detail the proposed quantization, reordering and entropy coding of the coefficient to obtain the fina bitstream, as well as the side information to be sent to the decoder. Then, we describe step by step the implementation of the complete encoder.

A. Quantization, Reordering, Entropy Coder and Side Information

The transform coefficient are quantized using a subband dependent quantization (i.e., the quantization step is smaller in low frequency subbands). Specifi quantization step values are given in Section V. These quantized coefficient are scanned following two different approaches based on the methods proposed in [34]: (i) inter-subband reordering, and (ii) intra-subband reordering. The energy of middle-high frequency coefficient will tend to be low, and thus these subbands will be likely to have a large number of zero coefficient after quantization. Inter-subband reordering groups coefficient that belong to the same subband, increasing the probability of having long strings of zero coefficients Specificall , the coefficient are sorted as:

$$\mathbf{coeffs}_{inter} = \left[\mathbf{s}^{j=J}, \mathbf{d}^{j=J}, \mathbf{d}^{j=J-1}, \ldots, \mathbf{d}^{j=1}\right], \tag{22}$$

where $\mathbf{s}^{j=J}$ are the smooth coefficient at level of decomposition $j = J$ (the lower frequency subband), and $\mathbf{d}^j$ are the detail coefficient at a generic level of decomposition j. Intra-subband reordering is based on the fact that edge weights provide an estimate of the reliability with which one $\mathcal{P}$ node is predicted from $\mathcal{U}$ neighbors, and that it is reasonable to assume that the magnitude of detail coefficient will tend to be smaller if they have been predicted from more "reliable" $\mathcal{U}$ neighbors. Intra-subband reordering aims to group together the most reliably predicted nodes, reordering the coefficient in each subband as a function of the average of the link weights between every $\mathcal{P}$ node and its $\mathcal{U}$ neighbors. Let us defin the average degree of node m with its $\mathcal{U}$ neighbors as: $\overline{D}_{m\mathcal{U}} = \frac{\sum_{n\in\mathcal{N}_m\cap\mathcal{U}} w_{mn}}{|\mathcal{N}_m\cap\mathcal{U}|}$. For a generic level of decomposition j, detail coefficient are sorted in increasing order of $\overline{D}_{m\mathcal{U}}$:

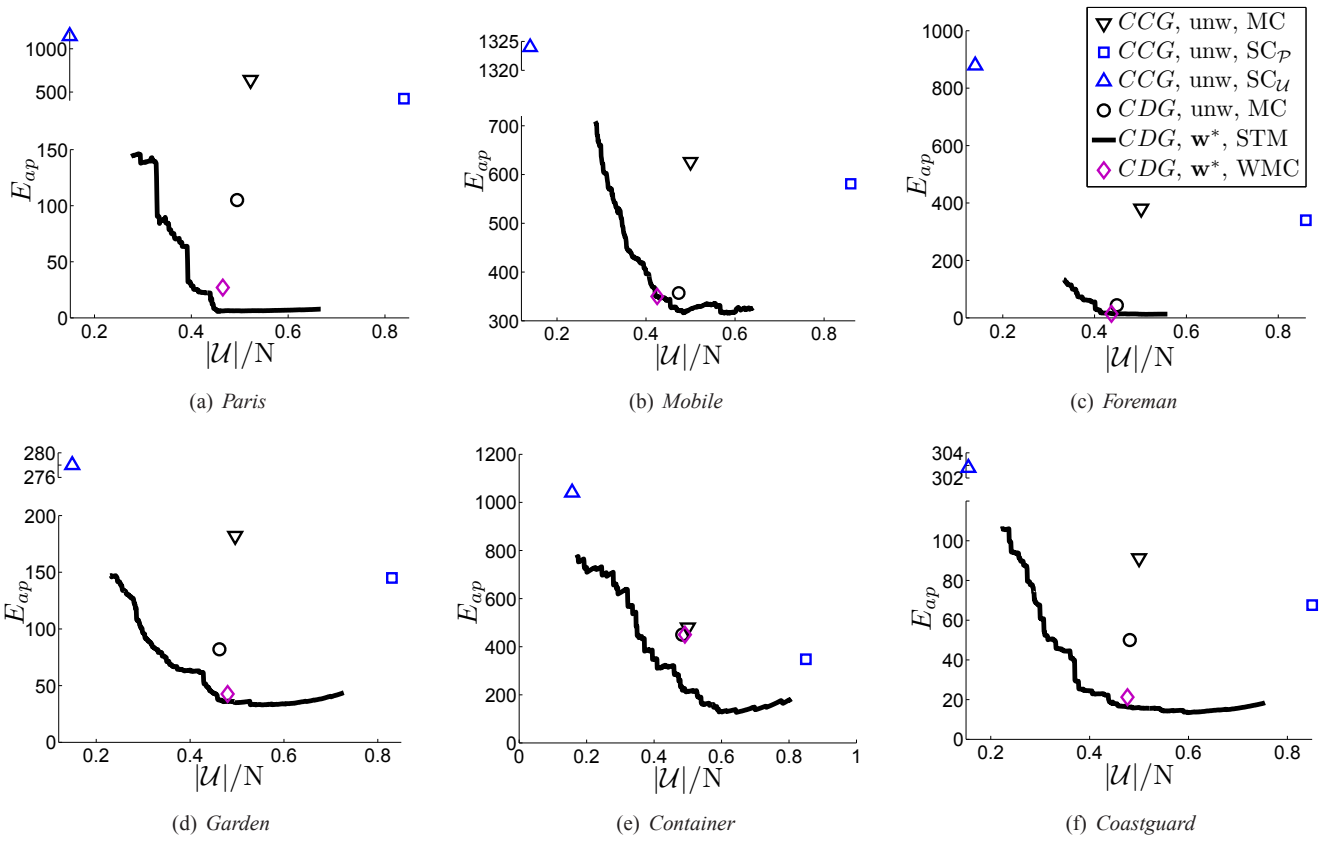
$$\mathbf{d}^j_{intra} = [d_a, d_b, \ldots, d_n], \tag{23}$$

where $\overline{D}_{a\mathcal{U}} < \overline{D}_{b\mathcal{U}} < \ldots < \overline{D}_{n\mathcal{U}}$.

This process is invertible because the weighted graph is known at both encoder and decoder. Figure 5 shows an example of the effect of reordering on quantized coefficient from 20 frames of the sequence *Carphone*.

After quantization and reordering, the coefficien vector is typically sparse, with larger coefficient at the beginning, and a large number of zero and ± 1 middle-high frequency coefficient (trailing ones). The entropy coder is designed to take advantage of these characteristics, working in scanning units of size S (i.e., the input of the entropy coder is a group of S coefficients in reverse order (from higher to lower frequencies) as described in Algorithm 3.

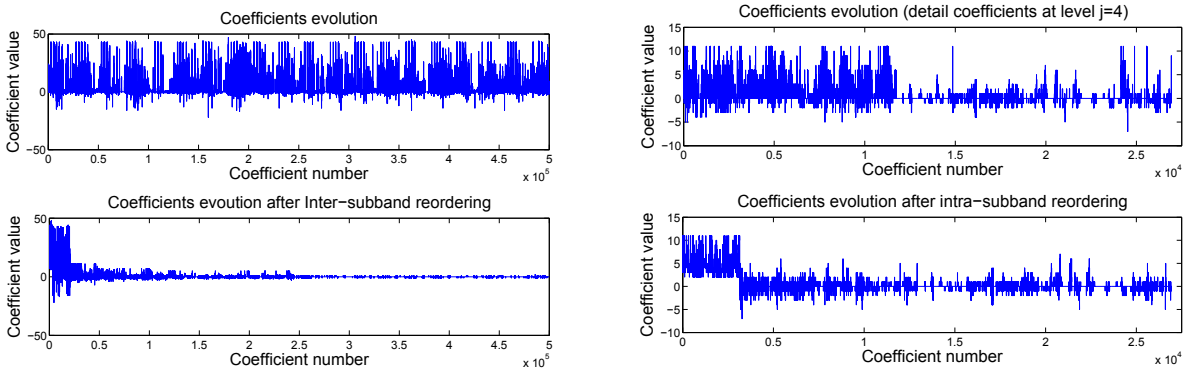
Note that, for every scanning unit of size S, a fla is also sent to the decoder to indicate whether the scanning unit is all zeros. The contour map, the MVs, and the weights have to be sent to the decoder as side information so that the process performed at the encoder is known at the decoder and thus



(a) *Paris* (b) *Mobile* (c) *Foreman*

(d) *Garden* (e) *Container* (f) *Coastguard*

Fig. 4: E_{ap} for different sequences. A comparison of different transform designs.



(a) Inter-subband reordering (b) Intra-subband reordering

Fig. 5: Inter and intra-subband reorderings. Top: original coefficients Bottom: reordered coefficients Data extracted from 20 frames of the sequence *Carphone*.

the system is invertible. The contour map is estimated using a standard Sobel edge detector and thresholding. To reduce the resulting overhead, we note that if there are no occlusions and the motion model captures object motion accurately, it is possible to estimate the contours of the current frame using contour data obtained from the reference frame along with motion information. Thus, in practice we only need to explicitly send contour information to the decoder once every K frames. Contour maps are encoded using JBIG. Regarding the temporal correlation, MVs are obtained from a standard integer full search within a specifi search range. Note that motion mappings are estimated using the original video frames, that is, the reference frame is not a reconstruction from a previously encoded frame as in the latest video coding standards such as H.264/AVC and H.265/HEVC (High Efficien y Video Coding). MVs are differentially encoded with respect to a predicted MV obtained from adjacent blocks. Then, a variable length code (VLC) is used to code the difference MV. Finally, weights are encoded using 9 bits per weight.

B. Complete Encoder Implementation

Figure 6 shows the encoder and the decoder data fl w. First, ME and contour detection processes are performed, obtaining the MVs and the contour map that are needed to obtain the

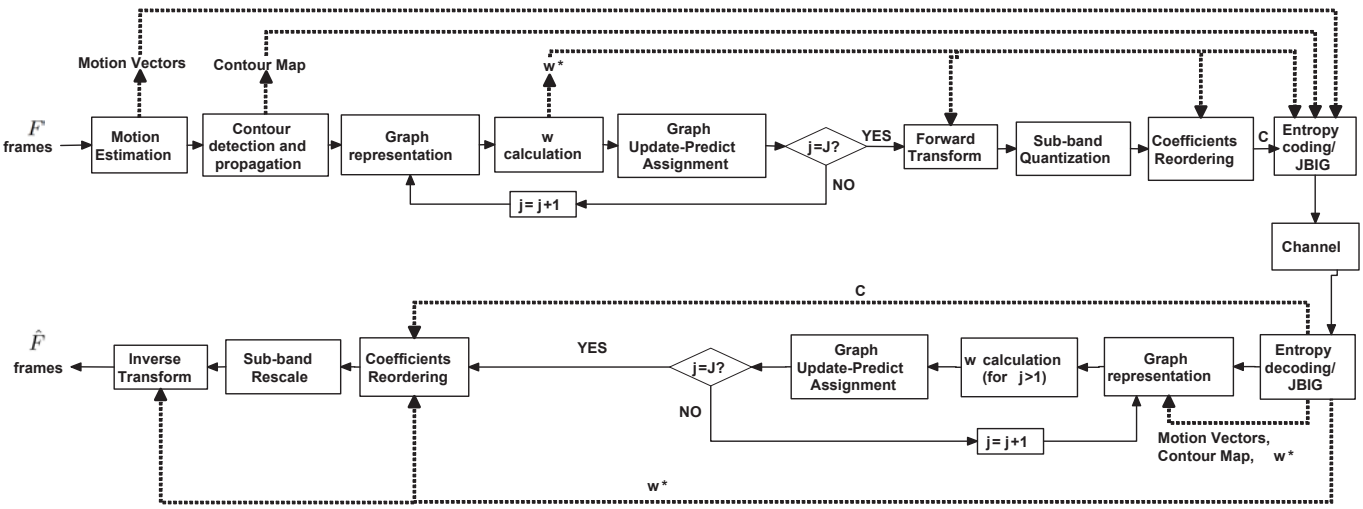


Fig. 6: Encoder and decoder data fl w.

Algorithm 3 *Entropy Coding of the Quantized Coefficient*

Require: S coefficient to be encoded

1: Encode the total number of non-zero coefficient and number of trailing ones using a fi ed number of bits $n = log_2(S)$
2: Encode the sign of the trailing ones
3: Encode the level of the remaining non-zero coefficient using an adaptive arithmetic coder
4: Encode the total number of zeros before the last non-zero coefficien using an Exp-Golomb code
5: Encode the number of zeros preceding each non-zero coefficien using an Exp-Golomb code
6: **return** Bitstream

graph representation of the video signal CDG (Fig. 2(a)). Then, the encoder calculates the optimal weights, $\mathbf{w}^*$ (10), and assigns these weights to the links (Fig. 2(b)). At this point, the encoder performs the $\mathcal{U}_{j=1}/\mathcal{P}_{j=1}$ assignment process solving the WMC problem using Algorithm 2. Next, for every $j > 1$, the weighted graphs are obtained following Algorithm 1, and the $\mathcal{U}_{j>1}/\mathcal{P}_{j>1}$ assignments are made (Algorithm 2). Once we have the graphs and $\mathcal{U}/\mathcal{P}$ assignments for all levels of decomposition, the encoder performs the transform (3) using the $\mathbf{p}$ filte of Definitio III.2 and the $\mathbf{u}$ filte of (16), quantizes the coefficients and reorders them ((22) and (23)). Finally, the contour map is encoded using JBIG, MVs are differentially encoded using a VLC, and coefficient ($\mathbf{C}$) are entropy coded using Algorithm 3 to generate the definit ve bitstream. Note that, as can be seen in Figure 6, the weight values are needed to perform the $\mathcal{U}/\mathcal{P}$ assignment, the filterin operations of the transform, and the reordering of the coefficients

V. Experimental Results

A. Video Coding Results

To evaluate the coding performance of the proposed encoder, we compare it with a MCTF approach [25] and with two different configuration of the H.264/AVC reference software JM15.1 [46]. In the firs configuratio of JM15.1 (H.264$_{simp}$), the test conditions are set so that only similar tools to the ones implemented in our encoder are enabled. In this way, subpixel ME is disabled, and ME is performed in blocks of size 16x16 (only 16x16 is available among all the inter modes), using 1 reference frame and ± 32 search range. To exploit spatial redundancy, mode intra 4x4 is allowed (in our system, spatial redundancy is exploited by means of spatial links between nodes). Note that allowing more than one intra mode would be equivalent to allowing our encoder to test different thresholds in the contour map detection and different k-hop spatial neighborhoods in the graph construction. Finally, RDO mode is set to low complexity (note that our encoder does not use rate-distortion optimization) and entropy coder used is CAVLC, which is the most similar to the one used in our system. The second configuratio of JM15.1 (H.264$_{full}$) allows different tools implemented in the standard (all the modes available, 5 reference frames,...) but uses the same ME as in the proposed encoder (subpixel ME disabled). The experiments were conducted using an IPPP GOP pattern, QP values ranging from 24 to 40, and 20 frames per sequence. Table I summarizes these conditions.

In MCTF and in our approach, the coefficient are quantized using a subband dependent quantization with the values specifie in Table II, where every column corresponds to a specifi subband, and every row shows different quality points ordered from higher (**Q1**) to lower (**Q4**) qualities. These values result in rate-distortion points comparable to the ones obtained using QP values ranging from 24 to 40 in the H.264/AVC encoder. Quantized coefficient are scanned following our proposed inter and intra-subband reorderings in our encoder and the inter-subband reordering in the MCTF encoder (intra-subband reordering is not possible in this encoder). Then, entropy coding is performed in both encoders using Algorithm 3 in scanning units of size $S = 4096$.

Regarding the ME process, block sizes of 16×16 pixels, search range of ± 32, and one reference frame are used. MVs are differentially encoded with respect to a predicted MV with a VLC. Although the MVs are different in the proposed and MCTF encoders (where the ME is carried out in original frames) and the H.264/AVC encoder (where reconstructed

TABLE I: H.264/AVC test conditions.

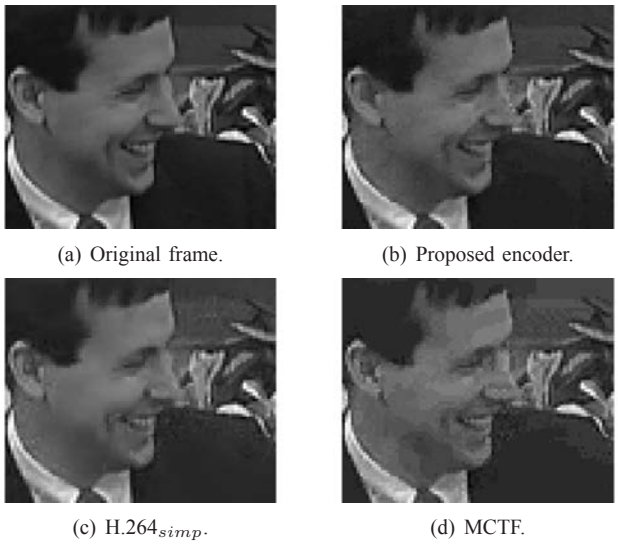
	Enabled Modes	Search Range	Number of References	RDO mode	Subpixel ME	Entropy Coder	QP
H.264$_{simp}$	Inter 16x16 and INTRA 4x4	± 32	1	Low complexity	Disabled	CAVLC	24,28,32,36,40
H.264$_{full}$	All modes enabled	± 32	5	Low complexity	Disabled	CAVLC	24,28,32,36,40

TABLE II: Subband quantization matrix for the proposed and the MCTF approaches

	$\mathbf{s}^{j=5}$	$\mathbf{d}^{j=5}$	$\mathbf{d}^{j=4}$	$\mathbf{d}^{j=3}$	$\mathbf{d}^{j=2}$	$\mathbf{d}^{j=1}$
Q1	5	5	5	10	20	30
Q2	5	5	10	20	30	40
Q3	10	10	20	30	40	50
Q4	20	20	60	70	70	70

reference frames are used), the rate turns out to be similar in both cases. The proposed encoder has an extra overhead because it has to send the contour information to the decoder once every K frames ($K = 20$ in our experiments) and the optimal weights every frame. Contour maps are encoded using JBIG, obtaining low rates of around 10 Kbps in average for QCIF sequences, and weights are coded using 9 bits per weight, giving rise to insignifican rates. In the experiments, f ve levels of decomposition are performed in the proposed and MCTF transforms.

Figure 7 shows the rate-distortion curves for different QCIF (*Container*, *Carphone*, *Flower*, *Mobile*, *Football* and *Husky*) and CIF sequences (*Silent*, *Paris*, *Deadline*, *Galleon*, *Tennis* and *Coastguard*). Table III shows the average PSNR differences of the proposed method with respect to the MCTF (ΔPSNR$_{\text{MCTF}}$) and the simplifie (ΔPSNR$_{\text{H.264simp}}$) and full (ΔPSNR$_{\text{H.264full}}$) configuration of the H.264/AVC encoder, calculated as described in [47].



(a) Original frame. (b) Proposed encoder. (c) H.264$_{simp}$. (d) MCTF.

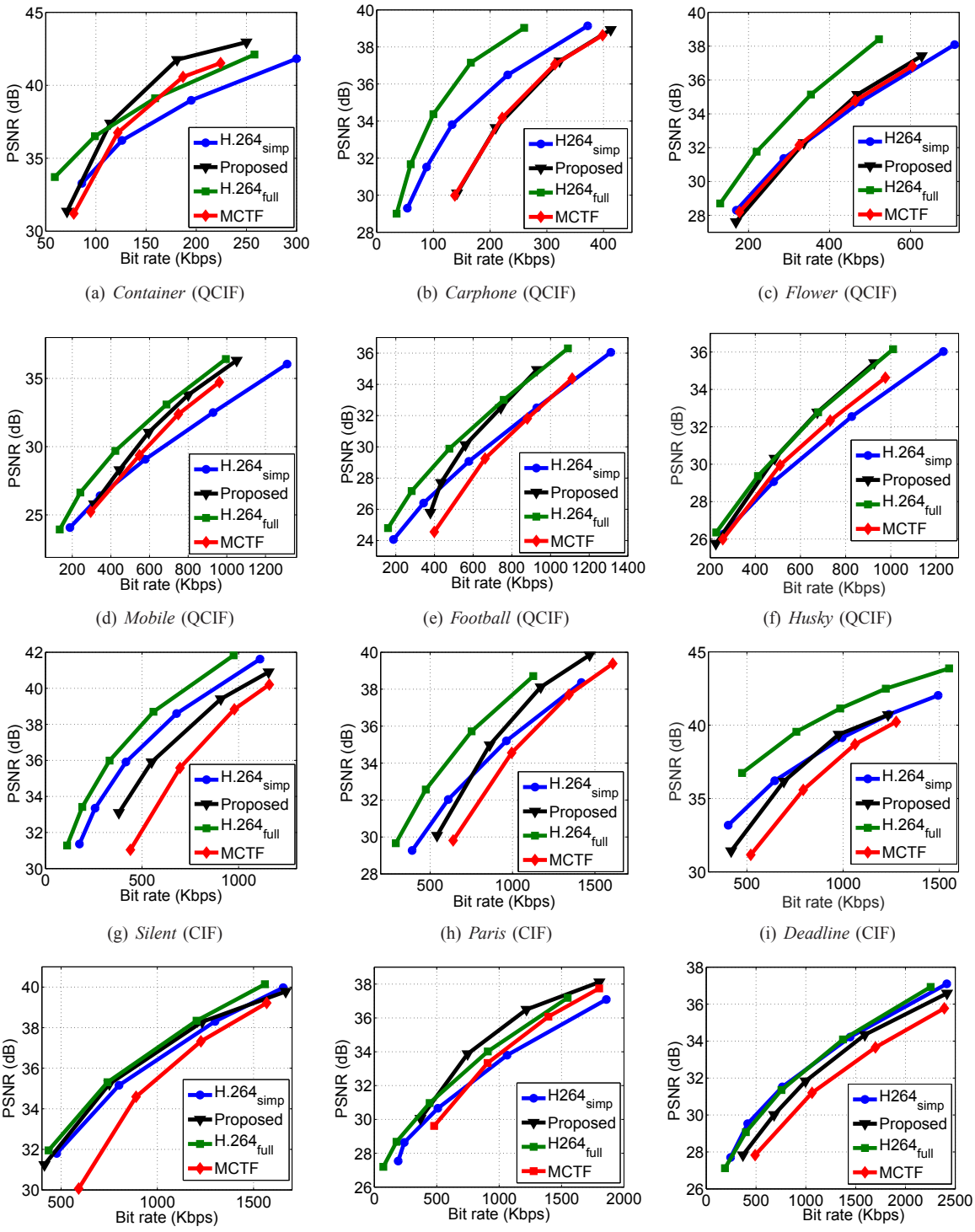
Fig. 8: Original and reconstructed frames with different encoders at 1100 Kbps.

The proposed method consistently outperforms the MCTF approach leading to an average PSNR improvement of 1.24 dB. In comparison to H.264$_{simp}$, the proposed method achieves an average PSNR improvement of 0.34 dB, and is better in eight out of twelve sequences. In medium to high qualities, our method is around 3 dB better than H.264$_{simp}$ in some sequences (*Mobile*, *Football*, *Container*, *Husky* or *Tennis*), outperforming H.264$_{full}$ in *Tennis* and *Container*, and obtaining similar results in *Husky*, *Galleon*, *Mobile* and *Football*. However, the efficien y of the proposed and MCTF encoders at low qualities or in simple sequences gets worse. This is due in part to the fact that the overhead is fi ed and does not adapt to the rate.

Figure 8 shows the raw frame $\#15$ of *Paris* and the reconstructed frame when it is coded at around 1100 Kbps with the proposed approach, the H.264$_{simp}$ configuratio and MCTF. The subjective quality obtained with the proposed encoder clearly outperforms that of MCTF and is slightly better than H.264$_{simp}$.

B. Discussion

Recent video coding standards include many different tools to improve the compression efficien y, such as multiple reference frames; different prediction modes and partition sizes; subpixel ME; or context adaptive entropy coders (CAVLC or CABAC), which significantl improve the efficien y of the encoder. Furthermore, a rate-distortion optimization process is performed that allows the encoder to choose the best coding option among different combinations of the previously described tools [48], [49]. In this paper the number of tools designed and implemented in our encoder is limited. Nevertheless, thanks to the versatility of the proposed encoder, it would be possible to combine our approach with many of the previously mentioned tools. As define in Definitio II.1, every node can be linked to any subset of nodes without restrictions maintaining the perfect reconstruction and critically sampled properties of the transform. Therefore, using biprediction, increasing the number of references, or incorporating variable partition sizes could be done in a straightforward way just by creating the corresponding links between nodes and sending the needed side information to the decoder. Weighted prediction can also be incorporated by managing the weights of the links between nodes. Moreover, as shown in [28], [50], arbitrary subpixel ME can be applied in MCTF implementations without loosing the transform invertibility. This can be extended to our system so that subpixel motion estimation is invertible using samples interpolated (in the spatial or temporal domains) from $\mathcal{U}$ nodes. While the proposed entropy coder is similar to CAVLC, it could be improved exploiting inter-subband correlations [51], [52] or making it context-adaptive as in H.264/AVC, where different look-up tables are chosen as a function of the context (e.g., number of nonzero coefficient



(a) *Container* (QCIF) (b) *Carphone* (QCIF) (c) *Flower* (QCIF)

(d) *Mobile* (QCIF) (e) *Football* (QCIF) (f) *Husky* (QCIF)

(g) *Silent* (CIF) (h) *Paris* (CIF) (i) *Deadline* (CIF)

(j) *Galleon* (CIF) (k) *Tennis* (CIF) (l) *Coastguard* (CIF)

Fig. 7: Coding performance comparison: PSNR versus bit rate.

in neighboring blocks or recently coded level magnitudes). As described in Section VI, we are currently working on the rate-distortion optimization of the graph that would allow to select the best coding option among all these tools.

As for complexity, our Matlab implementation is approximately twice the complexity of a comparable MCTF system also implemented in Matlab. Note that with the used $\mathcal{G}_{\mathbf{x}}$ (CDG) the mean number of temporal neighbors per node is 2 for every sequence, but the standard deviation is usually higher for sequences with large motion (ranging from 0.3 in *Container* to 0.55 in *Football* in our data set). The mean number of spatial neighbors per node greatly varies among sequences, being lower in sequences with high texture or big number of contours (1.42 in *Mobile*) and vice-versa (4.36 in *Carphone*). Although we employ the distributed approach of [35] and the subgraph approach of [34] for performing the $\mathcal{U}/\mathcal{P}$ assignment, which greatly reduce the encoder complexity, this complexity can be further reduced by limiting the size of the subgraphs, or using parallel processing (e.g., the $\mathcal{U}/\mathcal{P}$ assignment for different blocks can be parallelized).

VI. Conclusions and Future Work

A broad class of graph-based lifting transforms and their optimization for video coding have been proposed. These trans-

TABLE III: ΔPSNR (dB) of the proposed encoder with respect to 3 reference encoders

	Container	*Carphone*	*Flower*	*Mobile*	*Football*	*Husky*	*Silent*	*Paris*	*Deadline*	*Galleon*	*Tennis*	*Coastguard*	**Average**
$\Delta\mathbf{PSNR}_{\mathbf{MCTF}}$	1.37	−0.01	0.01	0.81	2.29	0.65	1.96	1.90	1.75	1.82	1.43	0.94	**1.24**
$\Delta\mathbf{PSNR}_{\mathbf{H.264simp}}$	2.39	−2.18	0.03	1.51	1.42	1.33	−1.36	0.46	−0.57	0.36	1.50	−0.80	**0.34**
$\Delta\mathbf{PSNR}_{\mathbf{H.264full}}$	0.28	−4.9	−2.44	−1.63	−1.29	−0.22	−2.6	−2.11	−2.62	−0.25	0.55	−0.77	**-1.50**

forms follow 3-dimensional (spatio-temporal) high-correlation filterin paths through the video signal, and can be considered a generalization of classical separable wavelet-based encoders.

To obtain an efficien graph-based lifting transform for video coding, some optimization problems have been discussed, namely: (i) the construction of suitable graph representations of the original video signal, including the graph weighting, that aims to capture the correlation between samples; (ii) the design of prediction filter for a given arbitrary weighted graph and update filter that are orthogonal to the prediction filter of its neighbors; (iii) $\mathcal{U}/\mathcal{P}$ assignment techniques that aim to fin a bipartition of the graph that leads to an efficien transform, and (iv) the extension of the transform to multiple levels of decomposition. Besides, we have proposed new reordering and entropy coding methods to obtain a complete coding scheme. The proposed method shows improved performance over a MCTF-based encoder and a simple encoder derived from H.264/AVC (JM15.1 configure to use similar tools as our proposed encoder), outperforming the JM15.1 configure to use 5 reference frames and all modes enabled, but subpixel ME disabled, at medium-high qualities for some sequences.

There are some interesting directions for future work. The fl xibility of the transform and the good results obtained in a video coding application provides confidenc that it may be successfully applied in a broad kind of signals and applications. For example, it may be used for multichannel-audio coding, image and video denoising, or biomedical signals compact representation, where one usually has multiple signals that present correlation in different domains (e.g., data extracted from the temporal evolution of different brain sensors present spatial and temporal correlation). Our current work is focused in the rate-distortion optimization of the encoder and the incorporation of subpixel ME.

REFERENCES

[1] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129-150, Mar. 2011.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging fiel of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83-98, 2013.

[3] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filte banks for graph structured data," *Signal Processing, IEEE Transactions on*, vol. 60, no. 6, pp. 2786-2799, 2012.

[4] D. Liu and M. Flierl, "Motion-adaptive transforms based on the laplacian of vertex-weighted graphs," in *Data Compression Conference (DCC), 2014*, Mar. 2014, pp. 53-62.

[5] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multiresolution graph fourier transform for compression of piecewise smooth images," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 419-433, Jan. 2015.

[6] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffi analysis," in IN IEEE INFOCOM, 2002.

[7] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53 - 94, 2006, special Issue: Diffusion Maps and Wavelets.

[8] W. Wang and K. Ramchandran, "Random multiresolution representations for arbitrary sensor network graphs," in *Acoustics, Speech and Signal Processing (ICASSP), 2006 IEEE International Conference on*, vol. 4, May 2006, pp. IV-IV.

[9] D. Liu and M. Flierl, "Video coding using multi-reference motion-adaptive transforms based on graphs," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, Jul. 2016, pp. 1-5.

[10] H. E. Egilmez, A. Said, Y. H. Chao, and A. Ortega, "Graph-based transforms for inter predicted video coding," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept. 2015, pp. 3992-3996.

[11] T. Maugey, A. Ortega, and P. Frossard, "Graph-based representation for multiview image geometry," *Image Processing, IEEE Transactions on*, vol. 24, no. 5, pp. 1573-1586, May 2015.

[12] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *Image Processing, IEEE Transactions on*, vol. 25, no. 4, pp. 1765-1778, Apr. 2016.

[13] E. J. Candès and D. L. Donoho, "Curvelets – a surprisingly effective nonadaptive representation for objects with edges," in *Curve and surface fittin* , A. Cohen, C. Rabut, and L.L. Shumaker, Eds. Saint-Malo: Vanderbilt University Press, 1999.

[14] M. Do and M. Vetterli, "The contourlet transform: an efficien directional multiresolution image representation," *Image Processing, IEEE Transactions on*, vol. 14, no. 12, pp. 2091 -2106, Dec. 2005.

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

[16] 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, Jul. 2006.

[17] B. Zeng and J. Fu, "Directional discrete cosine transforms-a new framework for image coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 305-313, 2008.

[18] G. Fracastoro, S. M. Fosson, and E. Magli, "Steerable discrete cosine transform," *Image Processing, IEEE Transactions on*, vol. PP, no. 99, pp. 1-1, 2016.

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

[20] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 674-693, 1989.

[21] R. L. Claypoole, G. M. Davis, W. Sweldens, and R. G. Baraniuk, "Nonlinear wavelet transforms for image coding via lifting," *Image Processing, IEEE Transactions on*, vol. 12, no. 12, pp. 1449-1459, 2003.

[22] D. Taubman, "Adaptive, non-separable lifting transforms for image compression," in *Image Processing (ICIP), 1999 IEEE International Conference on*, vol. 3, 1999, pp. 772-776 vol.3.

[23] G. Shen and A. Ortega, "Compact image representation using wavelet lifting along arbitrary trees," in *Image Processing (ICIP), 2008 IEEE International Conference on*, Oct. 2008, pp. 2808 -2811.

[24] 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, Sept. 2007.

[25] 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, Dec. 2003.

[26] M. Flierl and B. Girod, "Video coding with motion-compensated lifted wavelet transforms," *Signal Processing: Image Communication*, vol. 19, no. 7, pp. 561-575, 2004.

[27] 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.

[28] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *Acoustics, Speech and*

Signal Processing (ICASSP), 2001 IEEE International Conference on, Washington, DC, USA, 2001, pp. 1793-1796.
[29] Y. H. Chao, A. Ortega, and S. Yea, "Graph-based lifting transform for intra-predicted video coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, Mar. 2016, pp. 1140-1144.
[30] R. Wagner, H. Choi, and R. Baraniuk, "Distributed wavelet transform for irregular sensor network grids," in *IEEE Statistical Signal Processing (SSP) Workshop*, 2005.
[31] S. K. Narang and A. Ortega, "Lifting based wavelet transforms on graphs," in *APSIPA ASC 2009: Asia-Pacifi Signal and Information Processing Association, 2009 Annual Summit and Conference*, Oct. 2009.
[32] G. Shen and A. Ortega, "Optimized distributed 2D transforms for irregularly sampled sensor network grids using wavelet lifting," in *Acoustics, Speech and Signal Processing (ICASSP), 2008 IEEE International Conference on*, Mar. 2008, pp. 2513 -2516.
[33] E. Martínez-Enríquez and A. Ortega, "Lifting transforms on graphs for video coding," in *Data Compression Conference (DCC), 2011*, Mar. 2011, pp. 73 -82.
[34] E. Martínez-Enríquez, F. Díaz-de-María, and A. Ortega, "Video encoder based on lifting transforms on graphs," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept. 2011, pp. 3509 -3512.
[35] E. Martínez-Enríquez, F. Díaz-de-María, J. Cid-Sueiro, and A. Ortega, "Filter optimization and complexity reduction for video coding using graph-based transforms," in *Image Processing (ICIP), 2013 IEEE International Conference on*, Sept. 2013, pp. 1948-1952.
[36] G. Shen and A. Ortega, "Transform-based distributed data gathering," *Signal Processing, IEEE Transactions on*, vol. 58, no. 7, pp. 3802-3815, 2010.
[37] B. Girod and S. Han, "Optimum update for motion-compensated lifting," *Signal Processing Letters, IEEE*, vol. 12, no. 2, pp. 150 - 153, Feb. 2005.
[38] 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.
[39] N. Boulgouris, D. Tzovaras, and M. Strintzis, "Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding," *Image Processing, IEEE Transactions on*, vol. 10, no. 1, pp. 1 -14, Jan. 2001.
[40] A. Deever and S. Hemami, "Lossless image compression with projection-based and adaptive reversible integer wavelet transforms," *Image Processing, IEEE Transactions on*, vol. 12, no. 5, pp. 489 - 499, May 2003.
[41] J. Solé and P. Salembier, "Generalized lifting prediction optimization applied to lossless image compression," *Signal Processing Letters, IEEE*, vol. 14, no. 10, pp. 695 -698, Oct. 2007.
[42] G. P. Christophe, C. Tillier, and B. Pesquet-popescu, "Optimization of the predict operator in lifting-based motion compensated temporal filtering" in *Proc. of Visual Communications and Image Processing*, 2004.
[43] G. Shen, S. K. Narang, and A. Ortega, "Adaptive distributed transforms for irregularly sampled wireless sensor networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2009 IEEE International Conference on*, Apr. 2009, pp. 2225-2228.
[44] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar, "Improved update operators for lifting-based motion-compensated temporal filtering" *Signal Processing Letters, IEEE*, vol. 12, no. 2, pp. 146 - 149, Feb. 2005.
[45] E. Martínez-Enríquez, "Lifting transforms on graphs and their application to video coding," Ph.D. dissertation, Universidad Carlos III de Madrid, 2013.
[46] JVT H.264/AVC reference software v.15.1 [online], "http://iphome.hhi.de/suehring/tml/download/".
[47] G.Bjontegaard, "Calculation of average PSNR differences between RD-curves," ITU-T, VCEG-M33, Apr. 2001.
[48] G. J. Sullivan, T. Wiegand, and P. Corporation, "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE*, vol. 15, pp. 74-90, 1998.
[49] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 23 -50, 1998.
[50] S. Yea and W. A. Pearlman, "On scalable lossless video coding based on sub-pixel accurate MCTF," in *Proc. SPIE*, vol. 6077, 2006, pp. 60 771D-60 771D-10.
[51] 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 (DCC), 1997*, 1997, pp. 251 -260.
[52] J. M. Shapiro, "An embedded wavelet hierarchical image coder," in *Acoustics, Speech and Signal Processing (ICASSP), 1992 IEEE International Conference on,* vol. 4, 1992, pp. 657 -660 vol.4.

Eduardo Martínez-Enríquez (SM'07-M'15) received the Telecommunications Engineering degree from Universidad Politécnica de Madrid, Madrid, Spain, in 2006, and Ph.D. degree from the Universidad Carlos III de Madrid, Spain, in 2013. In 2014 he joined the Instituto de Óptica at the Consejo Superior de Investigaciones Científica , Madrid, Spain, where he is currently a postdoc researcher. His research interests include lifting transforms on graphs, wavelet-based video coding and video coding optimization. He received the Best Paper Award of ICIP 2011 for his paper on video coding based on lifting transform on graphs, co-authored with Fernando Díaz and Antonio Ortega.

Jesús Cid-Sueiro (M'95-SM'08) received the degree in Telecommunications Engineering from University of Vigo, Spain, and the Ph.D. degree from Polytechnic University of Madrid, Madrid, Spain, in 1990 and 1994, respectively.

He is currently a Professor with the Department of Signal Theory and Communications at Carlos III University of Madrid. His current research interests include machine learning, Bayesian methods, computational intelligence and their applications in sensor networks, big data and signal processing.

Fernando Díaz-de-María (M'97) received the Telecommunication Engineering degree and the Ph.D. degree from the Universidad Politécnica de Madrid, Spain, in 1991 and 1996, respectively. Since October 1996, he has been an Associate Professor in the Department of Signal Processing and Communications, Universidad Carlos III de Madrid, Madrid, Spain. His primary research interests include video coding, image and video analysis, and computer vision. He has led numerous projects and contracts in the field mentioned. He is co-author of numerous papers in peer-reviewed international journals, two book chapters, and has presented a number of papers in national and international conferences.

Antonio Ortega (F'07) received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid, Madrid, Spain in 1989 and the Ph.D. in Electrical Engineering from Columbia University, New York, NY in 1994. At Columbia he was supported by a Fulbright scholarship.

In 1994 he joined the Electrical Engineering department at the University of Southern California (USC), where he is currently a Professor. He has served as Associate Chair of EE-Systems and director of the Signal and Image Processing Institute at USC. He is a Fellow of the IEEE, and a member of ACM and APSIPA. He has been Chair of the Image and Multidimensional Signal Processing (IMDSP) technical committee, a member of the Board of Governors of the IEEE Signal Processing Society (SPS), and chair of the SPS Big Data Special Interest Group. He has been technical program co-chair of MMSP 1998, ICME 2002, ICIP 2008 and PCS 2013. He has been Associate Editor for the IEEE Transactions on Image Processing (IEEE TIP) and the IEEE Signal Processing Magazine, among others. He is the inaugural Editor-in-Chief of the APSIPA Transactions on Signal and Information Processing, an Associate Editor of IEEE T-SIPN and Senior Area Editor of IEEE TIP. He received the NSF CAREER award, the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award, the IEEE Signal Processing Society 1999 Magazine Award, the 2006 EURASIP Journal of Advances in Signal Processing Best Paper Award, the ICIP 2011 best paper award, and a best paper award at Globecom 2012. He was a plenary speaker at ICIP 2013 and APSIPA ASC 2015.

His research interests are in the areas of signal compression, representation, communication and analysis. His recent work is focusing on distributed compression, multiview coding, error tolerant compression, information representation in wireless sensor networks and graph signal processing. Almost 40 PhD students have completed their PhD thesis under his supervision at USC and his work has led to over 300 publications in international conferences and journals, as well as several patents.