This copy represents the peer reviewed and accepted version of paper:
F. Verdoja and M. Grangetto, "Directional graph weight prediction for image compression," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 1517–1521.

The published version is available at:
http://ieeexplore.ieee.org/

DIRECTIONAL GRAPH WEIGHT PREDICTION FOR IMAGE COMPRESSION

Francesco Verdoja, Marco Grangetto

Università degli Studi di Torino, Dept. of Computer Science

ABSTRACT

Graph-based models have recently attracted attention for their potential to enhance transform coding image compression thanks to their capability to efficiently represent discontinuities. Graph transform gets closer to the optimal KLT by using weights that represent inter-pixel correlations but the extra cost to provide such weights can overwhelm the gain, especially in the case of natural images rich of details. In this paper we provide a novel idea to make graph transform adaptive to the actual image content, avoiding the need to encode the graph weights as side information. We show that an approach similar to spatial prediction can be used to effectively predict graph weights in place of pixels; in particular, we propose the design of directional graph weight prediction modes and show the resulting coding gain. The proposed approach can be used jointly with other graph based intra prediction methods to further enhance compression. Our comparative experimental analysis, carried out with a fully fledged still image coding prototype, shows that we are able to achieve significant coding gains.

***Index Terms*—** Image compression, graph transform, directional prediction

1. INTRODUCTION

The history of lossy still image compression has been dominated by the transform based approach, as witnessed by the long lasting success of the JPEG coding standard. The 26 years old JPEG, based on fixed block size Discrete Cosine Transform (DCT), is still by far the most widespread image format. The DCT transform is known for its inefficiency when applied to a block that contains arbitrarily shaped discontinuities. In this case, the DCT is likely to result into a non-sparse signal representation resulting in poor coding performance. Many solutions have been proposed over the years to cope with this drawback, e.g. to mention only a few we recall shape-adaptive DCT [1], adaptive block-size transform [2], directional DCT [3] and wavelets [4]. Some of these tools have found their ways into standards, e.g. wavelets in JPEG2000 and hybrid prediction and transform based approaches with adaptive block-size in most recent video coding standards such as AVC and HEVC. We refer to [5] for a comprehensive analysis of the performance of current standards for still image compression.

Recently, the growing interest in graph based discrete signal processing [6] has stimulated the study of graph-based transform approaches. In this case, the image is mapped onto a topological graph where nodes represent the pixels and edges model relations between them, e.g. in terms of a criterion based on correlation or similarity. Proper weights can be associated to edges in the graph so as to model image discontinuities precisely. The Fourier transform can be

generalized to graphs obtaining the so called *Graph Fourier Transform* [7]. Several block based compression methods using GFT have been proposed [8–10]. In [11, 12] GFT is extended to provide a multiresolution representation. These works propose to use GFT for compression of piece-wise smooth data, e.g. depth images. It turns out that one of the major issue of graph-based compression techniques is represented by the cost required to encode the graph edge weights that, for natural images rich of details, can even exceed the potential coding gain provided by GFT. In [13, 14] a preliminary analysis of the overhead required by different methods to encode the graph structure is analyzed in the case of natural images. In [15, 16] it is shown that graph based models represent a framework that also helps in the design of optimal prediction and following transformation of residuals. In particular, in [15, 16] the authors show that the graph can be adapted to different directional modes defined in AVC. In [17] GFT is generalized including edges to model the prediction relationships.

In this paper we provide a novel idea to make graph transform adaptive to the actual image content avoiding the need to encode the graph weights as side information. We show that an approach similar to spatial prediction can be used to effectively predict graph weights in place of pixels; in particular, we propose the design of directional graph weight prediction modes and show the resulting coding gain. Moreover, we show that our approach can be used jointly with other graph based intra prediction methods, such as [17], allowing us to further enhance the compaction capability of the transform. Another important contribution is that to analyze the achievable compression gain we design a fully fledged lossy image codec, taking advantage of the statistical properties of the transformed coefficients. The encoding stage is based on context-based arithmetic coding and provides bitplane progressive description of the coefficients. This choice, as opposed to simpler energy compaction measures of the sole transform stage, allows us both to compare a set of prediction and transform modes, and to provide an absolute compression performance including all required overheads. Our experimental analysis is carried out on set of heterogeneous images including both photographic and computer rendered images and shows that the proposed approach provides a significant compression gain with respect to standard DCT; moreover, it allows to improve the performance also when coding spatial prediction residuals.

The paper is organized as follows: Section 2 presents the proposed graph prediction method and Section 3 describes how the transform coefficients are coded into a compressed bitstream. In Section 4 the results of our experimental analysis are discussed, whereas in Section 5 conclusions are drawn.

2. GRAPH WEIGHT PREDICTION

Let us denote with $\mathbf{X} = \{x_{i,j}\}$, $j = 1, \ldots, W$, $i = 1, \ldots, H$ a grayscale image of resolution $W \times H$. In the following, pixel intensity $x_{i,j}$ will also be referenced using a single subscript as x_k
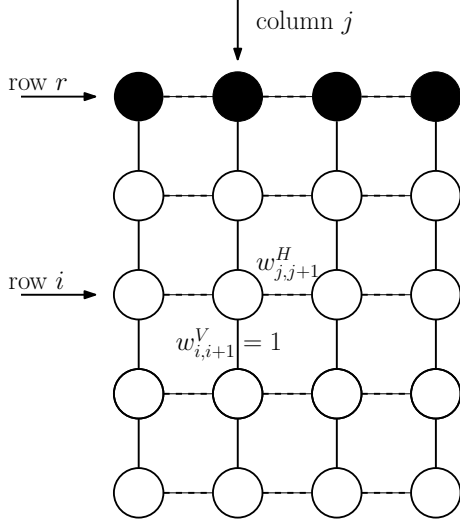
Fig. 1. Vertical weight prediction

with $k = (j-1)H + i$, i.e. we assume a standard column based raster scan matrix storage. Given any image region identified by a set of pixel indexes $\mathcal{B}$, one can define an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, with vertex set $\mathcal{V} = x_k, k \in \mathcal{B}$, edge set $\mathcal{E} = e_{i,j}$ if pixels x_i and x_j are connected, and weighted adjacency matrix $\mathbf{W}$. This latter is a symmetric $|\mathcal{B}| \times |\mathcal{B}|$ matrix, where $w_{i,j}$ is the weight associated to $e_{i,j}$.

The graph model $\mathcal{G}$ captures the inter-pixel correlation and can be used to derive the optimal decorrelating transform leveraging on spectral graph theory [18]. From the adjacency matrix, the combinatorial graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ can be computed, where $\mathbf{D}$ is the degree matrix: diagonal matrix whose i-th diagonal element d_i is equal to the sum of the weights of all edges incident to node i. The Laplacian matrix $\mathbf{L}$ is a symmetric positive semi-definitive matrix and it has eigen decomposition:

$$\mathbf{L} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} \qquad (1)$$

where $\mathbf{U}$ is the matrix whose rows are the eigenvectors of $\mathbf{L}$ and $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues. The matrix $\mathbf{U}$ is used to compute the graph Fourier transform (GFT) of the block $\mathcal{B}$: as:

$$\mathbf{y} = \mathbf{U}\mathbf{b}, \qquad (2)$$

with $\mathbf{b} = x_k, k \in \mathcal{B}$. The inverse graph transform is then given by

$$\mathbf{b} = \mathbf{U}^T \mathbf{y} \qquad (3)$$

For simplicity, in the following we will consider the most common block coding approach where the transformation is applied to non overlapping $s \times s$ square blocks $\mathcal{B}$. It is worth noticing that, as opposed to common 2D separable transformations, the GFT approach can be extended to arbitrarily shaped regions $\mathcal{B}$ without difficulties, except for the cost of signaling an irregular image segmentation. It is well known that the use of a 4-connected graph with uniform weights $w_{i,j} = 1$ on square block of pixels corresponds to the standard separable DCT [15].

Recent advances in image and video coding have shown the effectiveness of directional intra prediction modes, where different predictors are tested on the encoder side for each block; the best mode is signaled and the corresponding prediction residuals are transformed and coded. In this paper we propose to use a similar approach to predict graph weights in place of pixels. In this case one wishes to predict the weights in the adjacency matrix $\mathbf{W}$.

To this end, we now consider an image block $\mathcal{B}$ and the corresponding graph $\mathcal{G}$, as depicted in Figure 1, where empty circles represent pixels to be coded and the solid ones are already coded pixels of the top row r. Without loss of generality, let us assume strong vertical correlation among the pixels to be coded. In this case, a graph model where vertical edges connecting rows i and $i+1$ represent maximum correlation can be used. As shown in Figure 1, we can define vertical weights $w_{i,i+1}^V = 1$ between the i-th and the $(i+1)$-th row. In this paper we set such weights to 1 but, in general, any estimated correlation value ρ_V can be used and signaled to the decoder. It also follows from the above correlation assumption that the horizontal weights in the graph depend only on the considered column j and can be estimated form the top row. In particular, we can define the horizontal weights $w_{j,j+1}^H = f(|\hat{x}_{r,j} - \hat{x}_{r,j+1}|)$ as a function of the absolute difference between the pixel values $\hat{x}_{r,j}$; here we use $\hat{x}_{r,j}$ to denote a reconstructed pixel intensity after encoding and decoding. Motivated by the experimental analysis in [13] we use the Cauchy function to compute such weights:

$$f(d) = \frac{1}{1 + \left(\frac{d}{\alpha}\right)^2}, \qquad (4)$$

where α is a parameter. The GFT computed using the predicted graph is expected to match the block correlation structure and therefore to be closer to the optimal decorrelation transform for the block. As opposed to other approaches that require to encode the weights of the adjacency matrix, our method, similarly to intra prediction approaches, requires minimum coding overhead, i.e. simple signaling of the coding mode selected by the encoder. We term this coding mode as *Graph Weight Prediction* (GWP). Clearly, many similar directional prediction strategies can be devised, based on the structure of the already coded surrounding blocks. In this paper, to show the effectiveness of the proposed approach, we limit our analysis to the cases of vertical and horizontal GWP. To summarize when using GWP graph weights are estimated as follows:

Vertical mode	Horizontal mode
$w_{j,j+1}^H = f(|\hat{x}_{r,j} - \hat{x}_{r,j+1}|)$	$w_{j,j+1}^H = 1$
$w_{i,i+1}^V = 1$	$w_{i,i+1}^V = f(|\hat{x}_{i,r} - \hat{x}_{i+1,r}|)$

$$(5)$$

The GFT computed using the obtained $\mathbf{W}$ is then applied to the image block using (2). In the following we will identify this transformation as GWP-GFT.

GWP can be also used to encode intra prediction residuals by leveraging on previous results in the literature. In [19] it has been shown that, under the assumption of a separable first-order Gauss-Markov model for image signal, the optimal transformation for intra prediction residuals is the ADST (an asymmetric variant of the discrete sine transform). In [17] the approach is extended using graphs with the definition of the generalize GFT (GGFT); this latter is based on a generalized Laplacian $\mathbf{L}' = \mathbf{L} + \mathbf{D}'$, where $\mathbf{D}'$ is degree diagonal matrix whose i-th diagonal element d_i' is not zero if the corresponding node in the graph is on the prediction boundary; in particular, an extra weight is added as a function of the expected inaccuracy of intra-prediction. Looking at the vertical prediction example in Figure 1, $\mathbf{D}'$ is used to take into account the vertical edges

connecting the first row of pixels (empty dots) to the prediction samples (solid dots). Using non zeros weights $d'_i = 1$ and a 4-connected graph with uniform weights, ADST is obtained. GWP can be used along with GGFT by using (5) to set the graph weights and the extra term $\mathbf{D}'$ to take prediction edges into account. In the following we will refer to this approach using the acronym GWP-GGFT.

3. CODING OF TRANSFORM COEFFICIENTS

The proposed GWP approach has been used to develop a simple block based image codec. Every image block $\mathcal{B}$ is processed in raster scan order, transformed with GFT according to (2) and the obtained coefficients are quantized to integer values by using uniform scalar quantizer with quantization step q. The DC coefficient of each block (corresponding to the null eigenvalue of $\mathbf{L}$) is first predicted using the DC value of the previously coded neighbor block (if available). After subtraction of the DC prediction, $s \times s$ signed integers coefficients are obtained and arranged in a vector $\mathbf{y}_q$ for increasing values of the corresponding eigenvalue.

Each block is transformed using 3 different coding modes, namely uniform graph, horizontal and vertical GWP defined in the Section 2. The coding mode that, after quantization, produces the largest number of zeros is selected as the best one and is sent to the following entropy coding stage.

Entropy coding is founded on context adaptive binary arithmetic coding (CABAC) of the coefficient bitplanes in a progressive fashion. Since this approach is quite standard and does not represents the major contribution of this paper in the following we will provide a concise description omitting some implementation details. Four context classes are defined for sign, most and least significant bits, and for ancillary information, respectively. The bit position of the most significant bit of the magnitude of DC n_{DC} and the largest non DC coefficient n_{AC} are computed. These integer values are represented as unary codes and coded with CABAC in the *BlockHeader* context. Then, the coefficients are scanned from the highest to the lowest bitplane $\max(n_{DC}, n_{AC}) \leq n \leq 0$. A coefficient is significant at bitplane n if its magnitude is larger than 2^n. The n-th bit of a non yet significant coefficient is coded with CABAC using the *MSB* contexts that can take 8 different values depending on the significance of the 3 previous coefficients in $\mathbf{y}_q$; the rationale between these 8 contexts is that the probability of finding a new significant bit increases when previous coefficients with lower eigenvalues are already significant. If a coefficient turns significant, the corresponding sign is coded in the *Sign* context. Every bit of an already significant coefficient is coded using the *LSB* context. It is worth pointing that the used progressive bitplane scan creates a scalable bitstream for each block, that is therefore amenable to scalable coding. In this work we do not exploit this feature since we are primarily interested in the analysis of the compression gain obtained by GWP.

Unary coding and CABAC are also used to encode the best selected coding mode for each block using an additional *ModeHeader* context. Finally, to get a complete bit budget we also include a small header with global picture informations such as resolutions and transform block size s.

4. EXPERIMENTAL RESULTS

In this section the proposed GWP approach is compared with closely related contributions in the field in order to asses its potential for image compression. All the experiments are worked out on the set of standard images described in Table 1, that includes both photographic and computer rendered images with pixel resolution ranging

Table 1. Test images

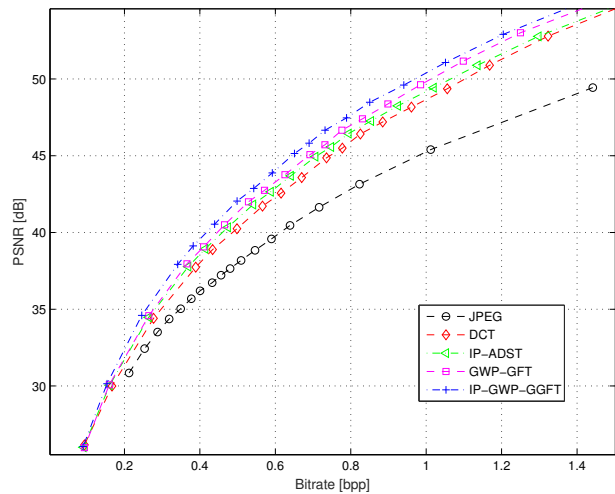| image	$W \times H$	source
bike, cafe	1152×1440	ITU [20]
p26	4288×2848	Microsoft [20]
kodim07	768×512	Kodak [21]
airplane	256×256	SIPI Image database [22]
bunnies, teapot	835×512	MIT [23]

Fig. 2. PSNR as a function of bitrate: Teapot image.

from 256×256 up to 4288×2848. All color images have been converted to grayscale. The coding gain achievable with GWP has been estimated using the full image codec described in Section 3, whose prototype has been implemented in C++ language leveraging on popular linear algebra libraries for GFT computation. The codec will be soon made available to the research community for reproducibility of our results and future works in the area.

The coding performance has been measured in terms of PSNR versus coding rate in bit per pixels (bpp) by varying the quantization step q. The block size has been fixed to $s = 8$ and graph weights are computed according to (5) with *Cauchy* function parameter $\alpha = 6.0$. Comparison with other methods and codecs will be presented using the standard Bjøntegaard Delta (BD) Rate, (ΔR in percentage) and Delta PSNR (ΔP).

Our comparative study is carried out by using the proposed codec with different prediction modes and transformation variants. In particular, we use standard DCT without prediction on all blocks (that coincides with GFT on uniform 8×8 graph) as a benchmark. This choice is clearly motivated by the long lasting success of the JPEG codec. Then, we add the two proposed vertical and horizontal GWP coding modes (GWP-GFT): as described in Section 3 the coding modes yielding the largest number of transform coefficients quantized to zeros is selected and signaled in the bitstream block by block. Moreover, we compare with an alternative solution based on 3 coding modes: classic DCT, vertical and horizontal intra prediction with ADST as proposed in [19] (this method will be referred to as IP-ADST). Finally, we investigate if ADST and GWP can be used jointly by applying the GWP-GGFT on intra prediction residuals: we use GGFT with unitary $\mathbf{D}'$ weights as recalled in Section 2

Table 2. Comparison of coding efficiency of proposed codec using DCT, IP-ADST, GWP-GFT, IP-GWP-GGFT and baseline JPEG.

image	IP-ADST vs. DCT		PWG-GFT vs. DCT		IP-PWG-GGFT vs. DCT		IP-PWG-GGFT vs. JPEG	
	ΔR	ΔP	ΔR	ΔP	ΔR	ΔP	ΔR	ΔP
airplane	0.13	-0.02	-5.83	0.47	-6.86	0.60	-36.77	2.57
bike	-1.00	0.14	-2.65	0.33	-3.99	0.48	-28.11	2.99
bunnies	-2.20	0.25	-4.19	0.45	-7.59	0.83	-30.89	3.51
cafe	-0.80	0.11	-2.58	0.32	-4.00	0.49	-26.93	3.25
kodim07	-3.09	0.28	-1.26	0.11	-4.77	0.42	-23.18	2.13
p26	-6.23	0.53	-4.18	0.30	-9.97	0.83	-36.60	2.70
teapot	-3.43	0.40	-5.90	0.69	-10.87	1.30	-30.91	4.13
Average	-2.37	0.24	-3.80	0.38	-6.86	0.71	-30.48	3.04
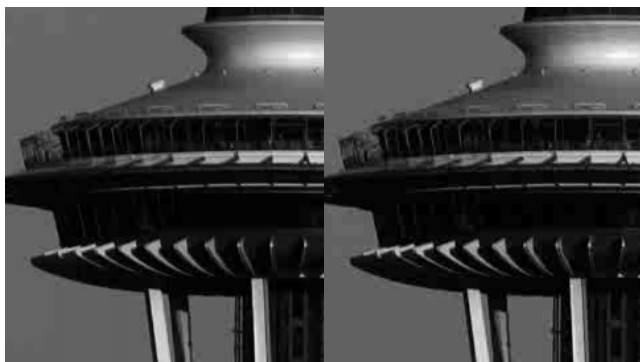
Fig. 3. Visual comparison of IP-GWP-GGFT (left) vs. DCT (right) over a cropped detail of image p26.

that makes it equivalent to ADST. We will refer to this variant as IP-GWP-GGFT.

In Figure 2 the rate/distortion curves obtained with the experimented methods on the Teapot image are shown. The performance yielded by the standard Baseline JPEG codec is reported as a benchmark as well. We can observe that the proposed bitplane codec, although quite simple, achieves a competitive performance with respect to JPEG: when encoding the same DCT coefficients as JPEG our codec (red curve) yields a PSNR gain of about 2.5 dB in the bitrate range between 0.5 and 1 bpp. The most interesting observations can be made when comparing the GWP-GFT (magenta), IP-ADST (green) and IP-GWP-GGFT (blue) curves. It can be noted that GWP-GFT significantly improve the compression performance even without resorting to spatial intra prediction. Indeed the GWP-GWT PSNR is almost equal or slightly better than IP-ADST that employs intra prediction and ADST transform. Finally, it is worth pointing out that graph weight prediction and pixel prediction can be exploited jointly to enhance the performance further: in fact, the IP-GWP-GGFT curve that jointly uses intra prediction, GWP and ADST achieves the best results with a gain larger than 1 dB with respect to DCT in the range between 0.5 and 1 bpp.

In Figure 3, we show visual comparison between IP-GWP-GGFT (left) and DCT (right) on the p26 image. This images have been coded at 0.2 bpp where the former yields PSNR of 38.93 dB and the latter of 37.35 dB. From the selected cropped area one can notice that IP-GWP-GGFT improves visual quality mostly by reducing blocking artifacts; this is particular evident over the edge and the white area of the dome and along the vertical pillars.

To better support the observations made on single images in Ta-

ble 2 we show the BD rates and PSNR obtained on all the images in our heterogeneous dataset. The first 3 sections of the table show ΔR and ΔP of IP-ADST, GWP-GFT and IP-GWP-GGFT with respect to the benchmark obtained by our codec with standard DCT. These results confirm that GWP-GFT is capable to significantly improve the compression performance. On some images, the GWP-GFT offers larger bitrate reduction and PSNR gain with respect to intra prediction (IP-ADST), whereas on average the two approaches yield very similar results. Most importantly, the joint usage of GWP and intra prediction (IP-GWP-GGFT) significantly improve the performance with average $\Delta R = -6.86\%$ and $\Delta P = 0.71$. Finally, the last two columns of the table show the BD gains of IP-GWP-GGFT versus JPEG and provide an absolute reference with respect to a standard performance: in this case we report average $\Delta R = -30.48\%$ and $\Delta P = 3.04$.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a method to make graph transform adaptive to the actual image content, avoiding the need to encode the graph weights as side information. Our approach uses directional prediction to estimate the graph weights; in particular, we have proposed and analyzed vertical and horizontal graph weight prediction modes that can be exploited to improve the compaction capacity of the GFT. Moreover, we showed that the proposed technique works also in conjunction with common intra prediction modes and other adaptive transforms such as ADST. As an added value, the experimental analysis has been carried out developing a GFT-based image codec, that exploits context adaptive arithmetic coding to encode the transform samples bitplanes. The proposed image codec has been used to compare several transform and prediction approaches with 8×8 blocks. The experimental results showed that the proposed technique is able to improve the compression efficiency; as an example we reported a BD rate reduction of about 30% over JPEG. Future works will investigate the integration of the proposed method in more advanced image and video coding tools comprising adaptive block sizes and richer set of intra prediction modes.

6. REFERENCES

[1] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 5, no. 1, pp. 59–62, 1995.

[2] M. Wien, "Variable block-size transforms for H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 604–613, 2003.

[3] 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.

[4] Zixiang Xiong, K. Ramchandran, M. T. Orchard, and Ya-Qin Zhang, "A comparative study of dct- and wavelet-based image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 692–695, 1999.

[5] T. Nguyen and D. Marpe, "Performance analysis of hevc-based intra coding for still image compression," in *Picture Coding Symposium (PCS), 2012*, 2012.

[6] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[7] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph fourier transform," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6167–6170.

[8] G. Shen, W.S. Kim, S.K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *Picture Coding Symposium (PCS), 2010*. IEEE, 2010, pp. 2808–2811.

[9] G. Cheung, W. S. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsification," in *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*, Oct 2011, pp. 1–6.

[10] W.S. Kim, S.K. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 813–816.

[11] S.K. Narang, Y.H. Chao, and A. Ortega, "Critically sampled graph-based wavelet transforms for image coding," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. IEEE, 2013, pp. 1–4.

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

[13] G. Fracastoro and E. Magli, "Predictive graph construction for image compression," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept 2015.

[14] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven graph transform for image compression," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept 2015.

[15] C. Zhang and D. Florencio, "Analyzing the optimality of predictive transform coding using graph-based models," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 106–109, Jan 2013.

[16] Y. Wang, A. Ortega, and G. Cheung, "Intra predictive transform coding based on predictive graph transform," in *2013 IEEE International Conference on Image Processing*, Sept 2013, pp. 1655–1659.

[17] W. Hu, G. Cheung, and A. Ortega, "Intra-prediction and generalized graph fourier transform for image coding," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1913–1917, Nov 2015.

[18] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field 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.

[19] J. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1874–1884, April 2012.

[20] "Image compression grand challenge at ICIP 2016," `http://jpeg.org/static/icip_challenge.zip`.

[21] "True color kodak images," `http://r0k.us/graphics/kodak/`.

[22] "SIPI image database," `sipi.usc.edu/database/`.

[23] "Synthetic light field archive," `http://web.media.mit.edu/~gordonw`.