EDGE-ADAPTIVE DEPTH MAP CODING WITH LIFTING TRANSFORM ON GRAPHS

Yung-Hsuan Chao, Antonio Ortega
University of Southern California
Los Angeles, CA, U.S.A
yunghsuc@usc.edu, ortega@sipi.usc.edu

Wei Hu
Hong Kong University
of Science and Technology
Hong Kong, China
huwei@ust.hk

Gene Cheung
National Institute of Informatics
Tokyo, Japan
cheung@nii.ac.jp

***Abstract*—We present a novel edge adaptive depth map coding based on lifting on graphs. The transform is localized, of low complexity, and guarantees perfect reconstruction as long as a proper predict-update split is defined. During the transform process, data in the prediction set are predicted by data in the update set; the prediction errors are then stored for encoding. In order to reduce the energy of the prediction residue, we propose to use optimized sampling on graphs to select the update set. Experiments show that the optimized sampling approach achieves better results than the conventional maximum cut based splitting in terms of transform efficiency and reconstruction quality. In addition, performance using the lifting transform is comparable to the state-of-the-art graph based depth map encoder using graph Fourier transform (GFT), which requires high complexity for signal projection.**

***Index Terms* – Lifting, Graphs, Transform Coding, Depth Map, Sampling Theory**

I. Introduction

The separable discrete cosine transform (DCT) is a popular transform used in image and video codecs such as JPEG, H.264, and HEVC. This transform has been shown to be equivalent to the optimal Karhunen-Loeve transform (KLT) for stationary Markov-1 signals for which the correlation between adjacent data points is close to 1. However, DCT does not exploit the fact that most images contain strong edge structures, which are especially significant in piece-wise smooth images such as depth maps. Clearly, the correlation between pixels across edges is lower than the correlation of pixels in the smooth areas. Thus, while on average correlation between pixels may approach 1, it can be useful to identify the location of these discontinuities and use a different model to represent them and code them. Moreover, note that separable DCT favors signals that have vertical and horizontal edges, and is not as efficient for blocks with diagonal edges or corners, and thus non-separable transforms may be advantageous in some situations.

In order to represent edges with more complicated orientations, in [9] an edge adaptive transform based on the graph Fourier transform (GFT) [4] is proposed for depth map compression, where discontinuities are signaled as low weight links in a graph representation of the image. The signals are projected onto the basis formed by the eigenvectors of the corresponding graph Laplacian. The computation of eigenvalue decomposition, while costly, can usually be performed off-line for typical graph structures. However, the signal projection needs to be done in real time. In GFT, the transform matrix is dense and usually does not have symmetries that could lead to a fast implementation. Therefore, the computation complexity is still high, which may limit the practical use of GFT.

In this work, we propose a fast depth map coding algorithm that can incorporate the edge information. We make two contributions: First, we introduce a lifting transform for the block-based encoder, in order to reduce the complexity of signal projection as compared to using the GFT. The transform can be applied for any irregular graph, and thus can achieve edge-adaptivity, similar to what is possible with the GFT. Besides, the transform is highly localized. Unlike GFT, which is a global transform, the computation of one transform coefficient requires information only from neighboring vertices on the graph. Furthermore, the lifting coefficients are rational rather than real as in the GFT. This leads to lower complexity implementations than the GFT. Experiments show that the compression using the lifting transform has similar performance to GFT in terms of rate distortion and perceptual quality.

The second contribution is a new predict-update assignment in the lifting transform based on optimized sampling on graphs. In the lifting based video encoder proposed in [8][7][6], the predict-update assignment is done using the maximum cut method (MaxCut). Since the link weights in the graph capture the similarity between nodes, maximizing the total weight means that the similarity between nodes across sets is maximized, which can reduce the prediction residual energy. The problem of selecting a good predictor has been addressed in the contexts of graph sampling and active learning. In this work, we take the optimized sampling proposed in [1] and [3] as a starting point. The greedy sampling optimization aims to minimize the number of samples that guarantee perfect recovery of signals with bandwidths lower than a given cut-off frequency. By using the same concept on the lifting transform, better PSNR performance can be achieved for depth map coding, as compared to the MaxCut approach. Moreover, the sampling process in optimized sampling is more efficient in that the prediction error saturates faster than MaxCut as the sample number increases. As a result, the computation cost

is lower in optimized sampling since the size of update set is smaller at each level, reducing the number of lifting levels needed for a target size of the lowest frequency band.

The rest of the paper is organized as follows. In Section II, we briefly review lifting transforms and the graph construction. In Section III we describe the optimized sampling approach for predict-update assignment. The encoding system is described in Section IV. In Section V, we compare the complexity of our algorithm with that of a GFT based encoder. Experiments on depth images and conclusions are presented in sections VI and VII, respectively.

II. Edge-Adaptive Transform

A weighted graph $G = (V, E)$ consists of a set of nodes V with $|V| = N$, and a set of links $e(m, n, w_{mn}) \in E$ connecting nodes m and n. $w_{ij} \in [0, 1]$ is the link weight modeling the similarity between two nodes. The $N \times N$ adjacency matrix W has its element $W(m, n) = w_{mn}$. The degree deg_m of node m is equal to $\sum_k w_{mk}$. A degree matrix D is a diagonal matrix with $D(m, m) = deg_m$. The combinatorial Laplacian $L = D - W$. To prevent confusion, we use *links* to denote the connections in the graph, and *edges* to denote the intensity discontinuities within an image.

A. Lifting transform on graphs

The lifting transform is a multi-level filtering process, where at each level i, nodes are divided into a prediction set (P_i) and an update set (U_i). The transform coefficients in the update set at level i are taken as the input nodes for level $i + 1$. For filtering on pixels in P_i, only the information of pixels in U_i is used, and vice versa. This is equivalent to approximating the original graph by a bipartite graph, since the connections within P_i or within U_i are not used. The resulting detail coefficients, $d^i \in P_i$, and smooth coefficients, $s^i \in U_i$, are computed as follows:

$$\begin{aligned} d_m^i &= s_m^{i-1} + \sum_{k \in U_i} p^i(m, k) s_k^{i-1} \\ s_n^i &= s_n^{i-1} + \sum_{r \in P_i} u^i(n, r) d_r^i \end{aligned} \tag{1}$$

where $p^i(m, k)$ is the filtering weight on node $m \in P_i$ using the information from node k, and $u^i(n, r)$ is the filtering weight on node $n \in U_i$ with the information from node r.

For the filter design, *i.e.*, selecting $p^i(m, k)$ and $u^i(n, r)$, we adopt the method proposed in [10], which is an extension of CDF 5/3 to arbitrary graphs with orthogonalization. For a 1-D line graph, CDF 5/3 filterbanks can be implemented via lifting by choosing the filters $p(m, k)$ and $u(n, r)$ as:

$$\begin{aligned} p(m, k) &= -\frac{W(m, k)}{\sum_j W(m, j)}, \\ u(n, r) &= \frac{W(n, r)}{2 \sum_j W(n, j)}. \end{aligned} \tag{2}$$

These filterbanks are nearly orthogonal. However, once these operators are applied on irregular graphs, where path merges exist, the inner product between two different filters is increased [10]. Therefore, we apply the technique of [10] to make the update filters orthogonal to the prediction filters.

B. Multi-level graph construction

In the first level of lifting (i.e., the original image), a graph is constructed for each block that links pixels to their 4-connected neighbors, with weights decided based on the signal geometry. In the experiments, we consider two different graph design: the first one, proposed in [9], uses only weights 0 and 1. The links across image discontinuities are disconnected (weight 0). The other design is based on [5]. Besides fully disconnected links corresponding to sharp edges, weak links with nonzero weights less than 1 are also included. The weight values are optimized based on a Gauss-Markov model. The detailed derivation is provided in [5].

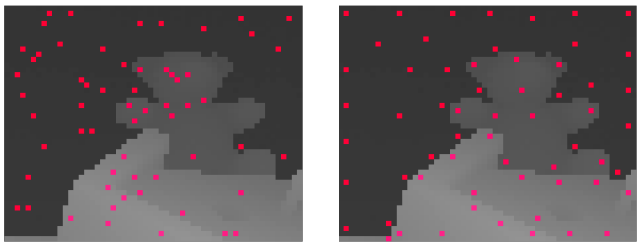
For higher levels of decomposition $i > 1$, we keep the 1-hop links that are not utilized in level $i-1$ and combine them with 2-hop links from level $i - 1$ in order to create the direct links at level i. The link weights in level i corresponding to the 2-hop links in the previous level are computed as $W^i(m, n) = W^{i-1}(m, k) \times W^{i-1}(k, n)$. If two nodes have multiple links connecting them, the average is used as the weight for the resulting combined link.

III. Predict/Update assignment based on optimized sampling

In image coding, a transform resulting in sparse representation in the transform domain is always desired. Therefore, in the lifting transform, we look for a predict-update assignment that can provide lower residual energy in the prediction set with fewer update samples. In [6] and [7], the assignment is made so that the sum of total link weights between U_i and P_i is maximized. Therefore, high correlation between the two sets is expected, which intuitively leads to lower residual energy after prediction. However, for each lifting level, around half of the nodes are needed to ensure low prediction error. Therefore, in order to reduce the number of smooth coefficients, which usually have large magnitude, a higher level lifting transform has to be used, making the transform computation costly.

In order to achieve sampling efficiency, we apply the optimized graph sampling developed in [1] for lifting bipartition. This optimized sampling defines a problem where the goal is to optimize the set of points to be sampled in an arbitrary graph in order to maximize the cut-off frequency, i.e., maximize the bandwidth of a signal that can be reconstructed exactly from those samples. More specifically, if we have a graph with N vertices, and given $k < N$, we look for the best subset of size k to maximize the bandwidth of a signal that can be perfectly reconstructed from those k samples. Equivalently, given a cut-off frequency, the objective is to find the subset of minimum size needed. Our main observation is that such a subset of k vertices would also be efficient in terms of minimizing the prediction error, when those k samples are used as the predictors for the remaining nodes.

Given that the problem of finding the optimal set of size k is in general combinatorial, we make use of a greedy heuristic that was initially developed in [1] and has been applied to active learning [3]. For predict-update assignment, the algorithm starts from an empty update set, and a prediction set containing all the nodes. The sample to be picked for the update set is decided as follows. First, a submatrix $\tilde{Ls}$ of the normalized Laplacian $L_n = D^{-1/2}LD^{-1/2}$ is extracted by selecting the rows and columns that correspond to nodes in the prediction set. Then, the eigenvector of the smallest eigenvalue of matrix $\tilde{Ls}$ to a specified power r is computed. The sample with the largest magnitude in the eigenvector is added to the update set. The process is repeated until a specified criterion is reached. In the toy example in Fig. 1, we see that as compared to the MaxCut approach, the samples are almost evenly distributed within each object. The mean squared error of the level 1 prediction for the toy example using CDF 5/3 as a function of the size of the update set, is shown in Fig. 2. As an example, for graph size $|V| = 2576$, the MaxCut based algorithm requires more than 1200 nodes selected to achieve low prediction error, while for optimized sampling (OptSamp), only around 900 nodes are needed. Since the size of the update set at each level is reduced, the number of lifting levels required to achieve a target size for the lowest frequency band is smaller than that of the MaxCut.



(b) MaxCut (a) Optimized Sampling

Fig. 1: First 60 samples selected from MaxCut, and the optimized sampling

IV. Depth Map Encoder

The encoding system used in our experiment is shown in Fig. 3. The input depth maps are first divided into non-overlapped 8×8 blocks, and intra-prediction is applied on each block using its causal neighbors. For the design of intra-predictor, we adopt the 9 mode 4×4 predictor used in $H.264/AVC$ [11]. Therefore, each 8×8 block is decomposed into 4 small blocks before prediction. For the upper left small block, reconstructed pixels from the causal neighbors are used as the predictor; for other 4×4 blocks, both the reconstructed and predicted pixels are used. The optimal prediction mode is decided based on the mean squared error. Such quad-tree division is solely for prediction purpose. For the transform and transform mode selection, 8×8 block is applied.

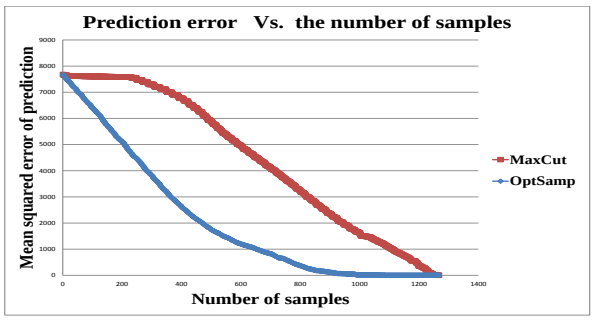


Fig. 2: Mean squared error of the prediction for different update set sizes

Once the intra-prediction residual blocks are computed, a graph structure is formed in each block based on the edge characteristics. Graph bipartition is done using the optimized sampling described in Sec. III, which determines the update and prediction transforms. The RD costs for DCT and lifting are compared by computing the sum of squared errors (SSE) and bit rate (R). For graph based lifting, both the bits for coefficient encoding and the edge information overhead are considered. DCT is chosen if there is no edge component in the current block or if its RD cost $RD_{dct} = SSE_{dct} + \lambda R_{dct}$ is smaller than the RD cost from the lifting transform, which is computed as $RD_{lifting} = SSE_{lifting} + \lambda(R^{coeff}_{lifting} + R^{edge}_{lifting})$.

Dead-zone uniform quantization is applied on the transform coefficients before scanning. The coefficients of lifting transform are ordered according to the frequency of the corresponding subband as in [7]. For piece-wise smooth images such as depth maps, pixels are highly correlated with neighboring pixels except for the pixels separated by edges. Such characteristics are embedded in the graph structure. Therefore, one pixel only considers the neighbors with similar intensities during the filtering, which makes most of the coefficients zero within the high frequency subbands. The coefficients of i level lifting are then arranged in the order: $[s_i, d_i, d_{i-1}, \ldots, d_2, d_1]$. Coefficients in the smooth subband, which contains most of the signal energy, are scanned first, followed by the detail coefficients with less energy in lower level. Such ordering increases the probability that zeros are scanned together, making the entropy coding more efficient.

V. Complexity Analysis

In general, there are two main sources of complexities in the encoder. One is due to determining the transform for the given graph, $i.e.$, the transform matrix in GFT and the filter coefficients in the lifting transform. The other is due to applying the transform on the input signal.

Computing the transform operation in GFT and the lifting transform based on optimized sampling requires the eigenvalue decomposition. In the former the eigenvectors of the combinatorial Laplacian form the transform basis. In the latter one, the smallest eigenvector for the normalized Laplacian corresponding to the prediction set is needed. The complexities are both significant, making an online computation impractical. Therefore, in the real coding system, the transforms are usually computed and stored in advance by limiting the number of graph structures used. In [5], a pre-determined number of

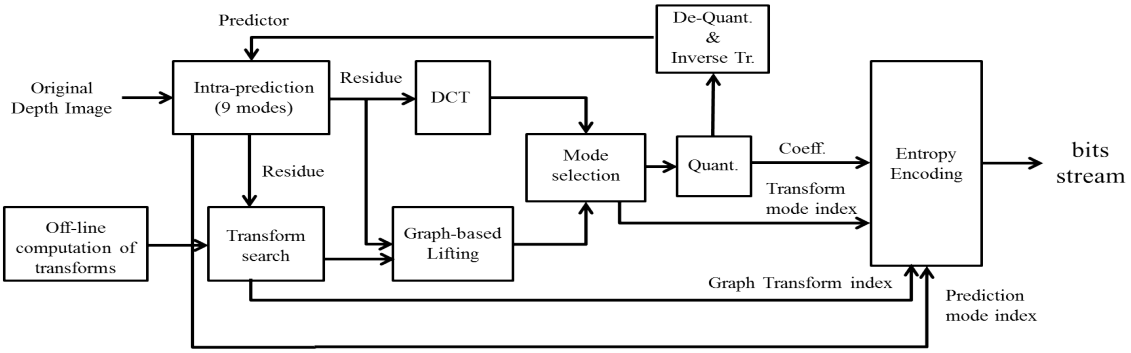


Fig. 3: Depth map encoder

transforms most frequently used in the training images are precomputed off-line for simple lookup. In this paper, we make the assumption that transforms are computed off-line, and focus on the complexity of online application of the resulting transforms

In GFT, the transform requires multiplying the input signal (denoted as a vector) with a dense matrix, which leads to $O(N^2)$ complexity. For lifiting, on the other hand, the transform is highly localized. If for each lifting level, half of the nodes are sampled into the update set, which is usually the upper bound of sampling, at most $\log N$ transform levels will be required. For the computation of each coefficient in level i, the number of operations equals to the degree deg_m of node m. deg_m is usually a constant number, and will not be scaled as the graph size increases. As a result, only $O(N \log N)$ complexity is needed for lifting application. Besides, the GFT coefficients are real valued, while in lifting the coefficients are rational, which leads to additional reduction in implementation cost. In our experiments, we next show that with the lifting transform, the quality of depth map reconstruction after compression is comparable to GFT, so that we pay no penalty for these significant reductions in complexity.

VI. Experimental Results

As a first comparison, we use the graph-based lifting and the baseline methods on several depth images. The quantized transformed coefficients and the index of optimal intra-prediction mode are coded with an arithmetic encoder. Besides the coefficients, for graph based method, 2 overheads are included in the entropy coder: The overhead of edge information, and the index of the optimal transform. The first one is encoded using the method proposed in [2], and the second one is computed using arithmetic coding.

The graphs are built under two settings: In the first case, graph is constructed based on the edge geometry of the intra-prediction residual using the same method as in [9]. The results in Fig. 4 show the PSNR comparison between DCT, GFT, and the lifting transform with optimized sampling (OptSamp) on image Teddy. In table I, we show the RD curve difference of lifting from GFT on different depth maps using Bjontegaard metric. The square of the normalized Laplacian is used for sampling set selection [1]. For optimized sampling, samples are selected greedily in each lifting level until every node in the prediction set has at least one neighbor in the update set. As a result, only 3 lifting levels are needed on average. It can be seen that the proposed method outperforms DCT and has similar performance to GFT. In Fig. 5, we compare the results of different level's lifting based on optimized sampling and MaxCut [7]. In order to get comparable performance as optimized sampling, MaxCut requires 5 lifting levels on average, making the computation costly. Table II shows the average RD curve difference of optimized sampling from MaxCut. In most of the cases, optimized sampling outperforms MaxCut in both PSNR gain and bitrate savings. Fig. 6 shows the reconstructed images for the various methods. It can be seen that the lifting method also achieves edge adaptivity similar to GFT, while in DCT method the edge structure contains lots of artifacts.

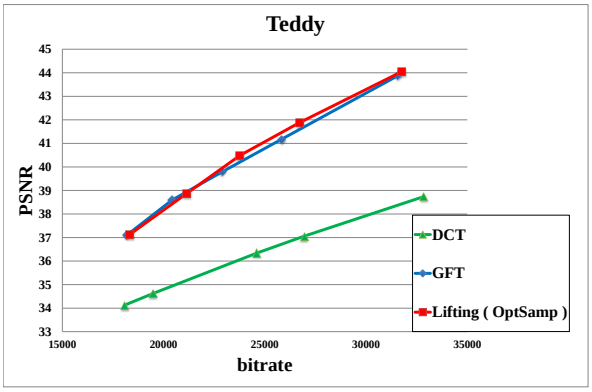


Fig. 4: RD performance comparison between DCT, GFT, and 3 level lifting based on optimized sampling (OptSamp)

	Bjontegaard metric	
	$\Delta PSNR(dB)$	$\Delta rate(\%)$
Teddy	0.10	-0.55
Cones	-0.04	0.35
Champagne Tower	-0.01	0.22
Tsukuba	-0.01	0.11
Ballet	0.02	-0.39

TABLE I: Average RD curve difference between 3 level lifting with optimized sampling (OptSamp) and GFT

As a second comparison we use the graph training proposed in [5]. The training set consisits of 10 depth images. 124 trained graph structures mostly used by the training blocks

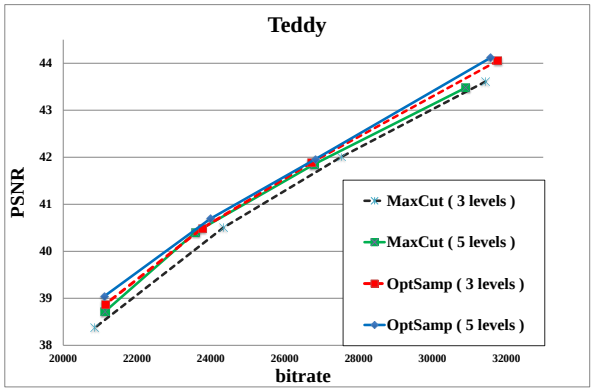


Fig. 5: RD performance comparison between different level's lifting based on MaxCut and optimized sampling (OptSamp)

Bjontegaard metric		
	$\Delta PSNR(dB)$	$\Delta rate(\%)$
Teddy	**0.11**	**-1.03**
Cones	-0.12	0.66
Champagne Tower	**0.06**	**-0.10**
Tsukuba	**0.03**	**-0.50**
Ballet	**0.11**	**-1.23**

TABLE II: Average RD curve difference between optimized sampling (OptSamp) and MaxCut

are stored. Fig. 7 shows the PSNR comparison, where the performance of the lifting transform is very close to GFT (less than 0.5 dB difference), and both have above 4 dB gain over DCT based method.

VII. Conclusion

In this paper, we present a fast graph-based lifting transform approach for depth map coding. Experiments show that lifting transform achieves similar reconstruction quality as graph Fourier transform (GFT), with the reduction in computation complexity. Further, a new predict-update assignment is proposed based on optimized sampling on graphs, which has lower transform complexity and better reconstruction quality than the conventional maximum cut approach. A design of an approximation to optimized sampling on graphs, which can be applied in real time, and a more rigorous way to decide the number of samples in the update set are left as the future works.

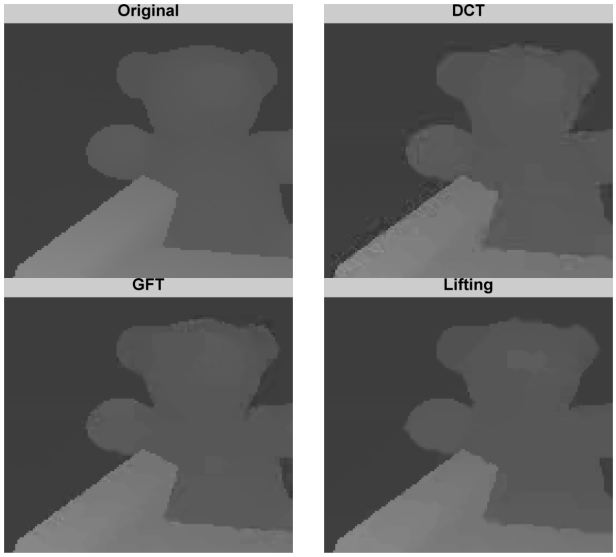


Fig. 6: Original image, and the reconstructed images from DCT, GFT, and the graph based lifting using the optimized sampling

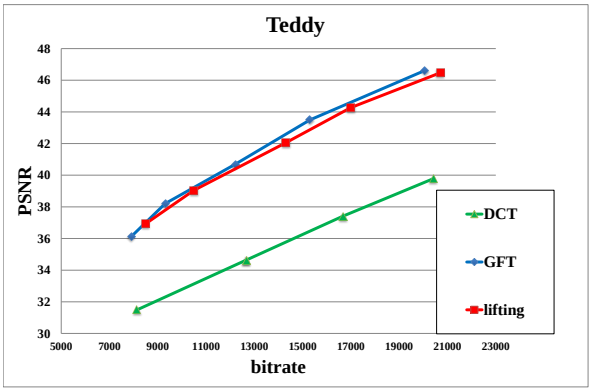


Fig. 7: RD performance comparison for Teddy using the multi-resolution GFT proposed in [5]

Acknowledgements

The work is supported in part by LG Electronics Inc.

References

[1] Aamir Anis, Akshay Gadde, and Antonio Ortega. Towards a sampling theorem for signals on arbitrary graphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[2] Ismael Daribo, Gene Cheung, and Dinei Florencio. Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1541–1544. IEEE, 2012.

[3] Akshay Gadde, Aamir Anis, and Antonio Ortega. Active semi-supervised learning using sampling theory for graph signals. *KDD'14,2014*.

[4] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[5] Wei Hu, Gene Cheung, Antonio Ortega, and Oscar C Au. Multi-resolution graph fourier transform for compression of piecewise smooth images. *accepted to IEEE Transactions on Image Processing*, 2014.

[6] E. Martínez-Enríquez, F. Diaz-de-Maria, and Antonio Ortega. Video encoder based on lifting transforms on graphs. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3509–3512. IEEE, 2011.

[7] Eduardo Martínez-Enríquez and Antonio Ortega. Lifting transforms on graphs for video coding. In *Data Compression Conference (DCC), 2011*, pages 73–82. IEEE, 2011.

[8] Sunil K Narang and Antonio Ortega. Lifting based wavelet transforms on graphs. In *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference*, pages 441–444. Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, 2009.

[9] Godwin Shen, W-S Kim, Sunil K Narang, Antonio Ortega, Jaejoon Lee, and Hocheon Wey. Edge-adaptive transforms for efficient depth map coding. In *Picture Coding Symposium (PCS), 2010*, pages 566–569. IEEE, 2010.

[10] Godwin Shen and Antonio Ortega. Tree-based wavelets for image coding: Orthogonalization and tree selection. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1–4. IEEE, 2009.

[11] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560–576, 2003.