Intra-Prediction and Generalized Graph Fourier Transform for Image Coding

Wei Hu, *Student Member, IEEE*, Gene Cheung, *Senior Member, IEEE*, and Antonio Ortega, *Fellow, IEEE*

***Abstract*—Intra-prediction is employed in block-based image coding to reduce energy in the prediction residual before transform coding. Conventional intra-prediction schemes copy directly from known pixels across block boundaries as prediction. In this letter, we first cluster differences between neighboring pixel pairs. Then, for each pixel pair, we add the cluster mean to the known pixel for prediction of the neighboring unknown pixel. The cluster indices are transmitted per block, allowing the decoder to mimic the same intra-prediction. We then propose an optimized transform for the prediction residual, based on a generalized version of previously developed Graph Fourier Transform (GFT). Experimental results show that our generalized intra-prediction plus transform coding outperforms combinations of previous intra-prediction and ADST coding by 2.5 dB in PSNR on average.**

***Index Terms*—Graph signal processing, image compression.**

I. Introduction

INTRA-PREDICTION is a popular tool for video coding since its adoption in the H.264/AVC standard [1]. In conventional intra-prediction one or more decoded pixels from a neighboring block ("boundary" pixels) are used to predict pixels in the current block ("predicted" pixels). Assume for now that only one boundary pixel is used, and denote x_0 and x_i the intensities of the boundary and predicted pixels, respectively. When intra-prediction is used, the prediction residuals $x_i - x_0$ will be transformed and encoded.

Intra-prediction is used in combination with techniques to optimize the choice of prediction direction, essentially assigning different boundary pixels to different sets of predicted pixels. Thus each specific intra-prediction mode corresponds to choosing a *subset of boundary pixels* to be used for prediction. For example, selecting a horizontal direction would lead to predicting pixel rows using as boundary pixels only the ones located immediately to the left of the block. Denoting S a set of pixels in the block that are predicted using x_0, the corresponding residual energy would be $\epsilon(x_0, S) = \sum_{x_i \in S} (x_i - x_0)^2$. Prediction directions, and thus the (x_0, S) pairs, will be chosen to minimize the total residual energy (the sum of all the $\epsilon(x_0, S)$ terms for the block[1]). The statistics of the resulting residual block can be significantly different from those of the original block (*e.g.*, the mean of the residues for a block will tend to be close to zero), leading to the use of the *asymmetric discrete sine transform* (ADST) [2], which outperforms the discrete cosine transform (DCT) for these residual blocks.

Existing intra-prediction approaches can select one of several possible prediction directions, but, for each boundary pixel, x_0, only two choices are possible: either x_0 is used as a predictor, *i.e.*, $x_i - x_0$ is computed for some x_i, or x_0 is not used for prediction. This simple prediction is useful when x_0 and the corresponding set S are part of a smooth region, but will tend to fail in other cases. For example, poor intra prediction performance may result *if there exist discontinuities* either at the block boundaries or within the block (especially when these discontinuities have orientations orthogonal to the chosen prediction direction). In such situations either intra-prediction will not be chosen or it will perform worse than might have been possible with a better predictor.

In this letter, we extend our approach in [3] to improve overall intra-prediction performance in the presence of discontinuities. In [3] we started by using simple thresholds to detect the presence of strong and weak discontinuities (edges) in an image. The location and type of these discontinuities were then transmitted as overhead and used to create a graph with edge weights corresponding to the correlation of pixel pairs across these discontinuities. The graph Fourier transform (GFT) [4] corresponding to the graph in each block was then used to transform the residue data after intra-prediction.

In this letter we improve performance by: i) **modifying intra-prediction** to take into account discontinuities occurring at block boundaries, ii) **optimizing thresholds** to classify the different types of discontinuities, iii) proposing a **novel generalized GFT (GGFT)** optimized for intra-prediction residues such as those generated with our approach. Our modified intra-prediction essentially allows us to introduce a small number of additional predictors, which can be viewed as shifted versions of the original predictor, *i.e.*, $\mu + x_0$ is used instead of x_0 for prediction, where μ will be conveyed as side information (as in [3]). A clustering method is used to classify different types of discontinuities, resulting in a different predictor for each cluster. This is essentially an optimization of the threshold selection in [3]. Our proposed GGFT is based on a generalized Laplacian, where the diagonal terms corresponding to vertices

Manuscript received May 06, 2015; accepted June 11, 2015. Date of publication June 17, 2015; date of current version June 24, 2015. This work was supported in part by JSPS under a Long Term Invitation Fellowship and by JSPS KAKENHI under Grant 23700136. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sanghoon Lee.

W. Hu is with Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China (e-mail: huwei@ust.hk).

G. Cheung is with National Institute of Informatics, Chiyoda-ku, Tokyo, Japan 101-8430 (e-mail: cheung@nii.ac.jp).

A. Ortega is with Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: antonio.ortega@sipi.usc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2015.2446683

[1]In practice this decision is often made based on a criterion that incorporates both residual energy and the rate required to encode the predictor assignment, but this is not important for the purpose of this discussion.

at block boundaries are modified: an extra weight is added as a function of the expected inaccuracy of intra-prediction. We show that, for the 1D case, the DCT and ADST [2] can be viewed as special cases of our proposed GGFT design. Using piecewise smooth and natural images as examples, experimental results show that we outperform combinations of previous intra-prediction and ADST coding by 2.5 dB in PSNR on average.

Related work includes approaches that also consider intra-prediction from a graph transform perspective [5], [6], as well as mode-dependent directional transforms (MDDT) [7]. Graph-based transforms for intra-prediction were introduced in [8] and formalized in [5] under the assumption of a Gauss-Markov random field (GMRF) model. In contrast to [5], here we use a model that incorporates explicit signal discontinuities. Further, the problem of identifying suitable GMRFs from actual data in order to design these transforms was not considered in [5]. Empirical models were derived in [6] but, unlike those we propose here, these were unstructured and the problem of representing them efficiently was not considered. In contrast, our starting point is a simple model of discontinuities between pixels, which does not require a significant amount of side information to be conveyed to the decoder. Finally, while MDDT [7] optimizes transforms for different directional predictions in a data-driven fashion, we allow different predictions and transforms as a function of both directions and the presence of discontinuities at the block boundaries. A significant advantage of our proposed techniques is that by explicitly defining a sparse graph, they are amenable to lower complexity graph-based transforms (*e.g.*, based on lifting [9], [10]). To the best of our knowledge, using the GGFT as an image transform, and its interpretation as a generalization of the DCT/ADST, has not been proposed before.

II. Signal Modeling and Intra Prediction

A. Signal Modeling

We first model a 1D signal (one row of pixels) $\mathbf{x}$ as:

$$x_n = x_{n-1} + \mu_n, \tag{1}$$

where μ_n is an identical and independently distributed (iid) random variable, and x_n and x_{n-1} are two neighboring pixels. Fig. 1 shows two examples of log distributions of inter-pixel differences—empirical estimates of the pdf for μ_n.

As observed in Fig. 1, inter-pixel difference distributions are concentrated around 0 (*i.e.*, images are mostly *smooth*), but occasionally have large inter-pixel differences in both color and depth images (*e.g.*, around 10% of pixel pairs have inter-pixel differences exceeding twice the standard deviation.)

If the encoder explicitly signals large inter-pixel differences μ_n and their locations to the decoder, then μ_n can be subtracted from signal $\mathbf{x}$ at those locations, resulting in a smoother residual signal for coding. However, signaling many different μ_n values is expensive. Thus, we propose to *quantize* μ_n using K quantization bins. Each bin k is defined by its boundaries $[T_k, T_{k+1})$ and average $\hat{\mu}_k$. We compute the K quantization bins via *clustering*, employing the well-known Lloyd algorithm [11]. The computed mapping $\hat{\mu}_k = f(\mu_n)$, where $k = i(\mu_n)$ is the index of the bin assigned to μ_n, maps the observed inter-pixel difference μ_n to the bin average $\hat{\mu}_k$ of cluster k. The only change

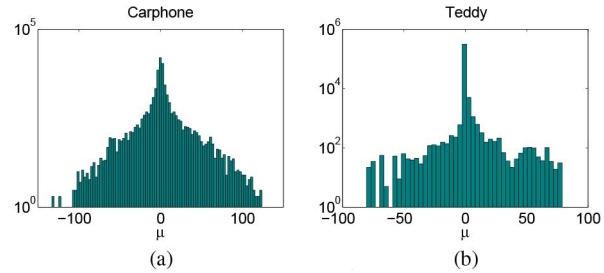

Fig. 1. (a) The log histogram of μ_n for the natural image `Carphone`. (b) The log histogram of μ_n for the depth map `Teddy`.

with respect to the standard Lloyd algorithm, is that for the *zero bin*—the one containing the inter-pixel difference equal to 0—we assign $\hat{\mu}_0 = 0$ as the centroid. By doing so, as will be shown shortly, our prediction strategy will correspond to standard intra prediction when μ_n is mapped to the zero bin.

Assuming that the encoder will signal a bin index $i(\mu_n)$ for each pixel pair, the decoder can use the corresponding bin average $\hat{\mu}_{i(\mu_n)}$ to approximate μ_n and we can model the approximation error as a random variable $g_{i(\mu_n)}$, for each quantization bin $i(\mu_n)$, so that the model of (1) can be modified as follows:

$$x_n = x_{n-1} + \hat{\mu}_{i(\mu_n)} + g_{i(\mu_n)}, \quad n = 1, \ldots, N, \tag{2}$$

where a larger quantization bin for a given $k = i(\mu_n)$ will lead to increased variance $\sigma^2_{g_k}$ of the error g_k.

B. 1D Block-based Intra-Prediction

We now show how to use the model of (2) to perform intra-prediction on a length-N block $\mathbf{x} = [x_1, \ldots, x_N]^T$ given a boundary pixel x_0. At the encoder, we first map inter-pixel differences $\mu_n = x_n - x_{n-1}$ to clusters according to the computed bin boundaries T_k. Without loss of generality, we assume that $\hat{\mu}_a = f(\mu_1)$, $\hat{\mu}_b = f(\mu_m)$, and $0 = f(\mu_n)$ for the remaining μ_n, as shown in Fig. 2(a), that is, there are two discontinuities, one at the boundary and one between pixels m and $m-1$. We can now expand (2) as:

$$\begin{aligned} x_1 - x_0 &= \hat{\mu}_a + g_a \\ x_2 - x_1 &= g_0 \\ &\vdots \\ x_m - x_{m-1} &= \hat{\mu}_b + g_b \\ &\vdots \\ x_N - x_{N-1} &= g_0, \end{aligned} \tag{3}$$

which can be written in matrix form as:

$$\mathbf{F}\mathbf{x} = \mathbf{d} + \mathbf{e}, \tag{4}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ -1 & 1 & \cdots & 0 & 0 \\ & \ddots & \ddots & & \\ 0 & 0 & \cdots & -1 & 1 \end{bmatrix}, \mathbf{d} = \begin{bmatrix} x_0 + \hat{\mu}_a \\ 0 \\ \vdots \\ \hat{\mu}_b \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{e} = \begin{bmatrix} g_a \\ g_0 \\ \vdots \\ g_b \\ g_0 \\ \vdots \\ g_0 \end{bmatrix}. \tag{5}$$

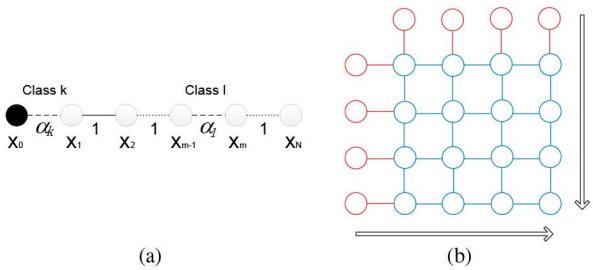

Fig. 2. Intra-prediction in 1D (a) and 2D (b). Red circles denote boundary vertices, and the blue ones represent vertices for the target block.

$\mathbf{F}$ is invertible, thus

$$\mathbf{x} = \mathbf{F}^{-1}\mathbf{d} + \mathbf{F}^{-1}\mathbf{e}. \tag{6}$$

Note that

$$\mathbf{F}^{-1}\mathbf{d} = \begin{bmatrix} x_0 + \hat{\mu}_a \\ \vdots \\ x_0 + \hat{\mu}_a \\ x_0 + \hat{\mu}_a + \hat{\mu}_b \\ \vdots \\ x_0 + \hat{\mu}_a + \hat{\mu}_b \end{bmatrix} \tag{7}$$

is the predictor for $\mathbf{x}$ and that if we set $\hat{\mu}_a = \hat{\mu}_b = 0$, then $\mathbf{F}^{-1}\mathbf{d} = [x_0, \ldots, x_0]^T$, *i.e.*, the conventional intra-prediction. Thus we can interpret $\mathbf{F}^{-1}\mathbf{e}$ as the intra-prediction residual $\mathbf{r}$.

$\mathbf{F}^{-1}\mathbf{d}$ is in fact the *optimal* predictor of $\mathbf{x}$ for our given model—any other predictor will yield non-zero expected prediction error. To see this, we compute the expectation of x_n given x_0 for $1 \le n \le m-1$:

$$\begin{aligned} E[x_n|x_0] &= E[x_0 + \hat{\mu}_a + g_a + \ldots + g_0|x_0] \\ &= E[x_0|x_0] + E[\hat{\mu}_a|x_0] + E[g_a|x_0] + \ldots + E[g_0|x_0] \\ &= x_0 + \hat{\mu}_a. \end{aligned} \tag{8}$$

It can be derived similarly that $E[x_n|x_0] = x_0 + \hat{\mu}_a + \hat{\mu}_b$ for x_n, $m \le n \le N$. Hence, $\mathbf{F}^{-1}\mathbf{d}$ is the optimal predictor in the sense that it results in a zero-mean prediction residual.

III. Generalized Graph Fourier Transform

We first derive the inverse covariance matrix for $\mathbf{r} = \mathbf{F}^{-1}\mathbf{e}$ in (6), which leads to the KLT. Since $E[\mathbf{r}] = \mathbf{F}^{-1}E[\mathbf{e}] = \mathbf{0}$, the covariance matrix $\mathbf{C}$ of the prediction residual $\mathbf{r}$ is simply:

$$\mathbf{C} = E[\mathbf{r}\mathbf{r}^T] = E[\mathbf{F}^{-1}\mathbf{e}\mathbf{e}^T(\mathbf{F}^T)^{-1}] = \mathbf{F}^{-1}E[\mathbf{e}\mathbf{e}^T](\mathbf{F}^T)^{-1}. \tag{9}$$

Further computation gives

$$E[\mathbf{e}\mathbf{e}^T] = \mathrm{diag}[\sigma_{g_a}^2, \sigma_{g_0}^2, \ldots, \sigma_{g_0}^2, \ldots, \sigma_{g_0}^2]. \tag{10}$$

The precision matrix $\mathbf{Q}$, *i.e.*, the inverse of $\mathbf{C}$, is a tridiagonal matrix:

$$\mathbf{Q} = \mathbf{C}^{-1} = \mathbf{F}^T E[\mathbf{e}\mathbf{e}^T]^{-1}\mathbf{F}$$

$$= \frac{1}{\sigma_{g_0}^2} \begin{bmatrix} \alpha_a+1 & -1 & & & & & & & \\ -1 & 2 & -1 & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & & -1 & 1+\alpha_b & -\alpha_b & & & & \\ & & & -\alpha_b & \alpha_b+1 & -1 & & & \\ & & & & \ddots & \ddots & \ddots & & \\ & & & & & -1 & 2 & -1 & \\ & & & & & & -1 & 1 & \end{bmatrix}, \tag{11}$$

where $\alpha_a = \sigma_{g_0}^2/\sigma_{g_a}^2$ and $\alpha_b = \sigma_{g_0}^2/\sigma_{g_b}^2$. Since $\mathbf{Q}$ shares the same eigenvectors with $\mathbf{C}$ [5], the KLT for residual signals, $\mathbf{r}$, is thus the eigenvector set of $\mathbf{Q}$ in (11). Note that, since the multiplier $1/\sigma_{g_0}^2$ has no influence on the eigenvectors, we consider $\mathbf{Q}$ without this factor in the sequel. Next we show that this KLT is equivalent to the GGFT of a particular graph.

We construct an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ composed of a vertex set $\mathcal{V}$ of cardinality $|\mathcal{V}| = N$, an edge set $\mathcal{E}$ connecting vertices (each edge $e_{i,j}$ connects vertices i and j), and a weighted *adjacency matrix* $\mathbf{W}$. $\mathbf{W}$ is a real symmetric $N \times N$ matrix, where $w_{i,j}$ is the weight assigned to $e_{i,j}$. For the 1D example of (11), as illustrated in Fig. 2(a), each vertex $v \in \mathcal{V}$ corresponds to a pixel in the target code block, *e.g.*, $\{x_1, \ldots, x_N\}$ in Fig. 2(a); two vertices are connected with an edge in $\mathcal{E}$ if the corresponding pixels are neighbors in the target block. We assign a weight $w_{n,n-1} = \alpha_k$ if $\hat{\mu}_k = f(\mu_n)$. The *combinatorial graph Laplacian* [3], [8], [12] is defined as $\mathcal{L} := \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the *degree matrix*—a diagonal matrix where $d_{i,i} = \sum_{j=1}^{N} w_{i,j}$. A *generalized Laplacian* $\mathcal{L}^g$ is a symmetric matrix with non-positive off-diagonal entries [13], which can be constructed, for example, as $\mathcal{L}^g = \mathcal{L} + \mathbf{D}'$, where $\mathcal{D}''$ is a diagonal matrix. The GGFT is then the orthogonal matrix of eigenvectors of $\mathcal{L}^g$.

Using $\mathcal{L}$ from the graph we just defined, we can write $\mathbf{Q} = \mathcal{L} + \mathbf{D}'$, with $\mathbf{D}' = \mathrm{diag}[\alpha_a, 0, \ldots]$. $\mathbf{D}'$ is defined by first identifying the set of *boundary vertices* $\partial\mathcal{V}$ (in the 1D case of (11) x_0 is the only boundary pixel) and the set of *boundary edges* $\partial\mathcal{E}$ (in this case a single connection between x_0 to x_1). $\mathbf{D}'$ will be non-zero *only* for the diagonal positions corresponding to vertices in $\mathcal{V}$ connected to $\partial\mathcal{V}$. Then, for a vertex i, the corresponding entry in $\mathbf{D}'d'_{i,i}$ will be equal to the additional degree introduced by all boundary edges connecting to i, i.e., $d'_{i,i} = \sum_{i \in \mathcal{V}, v \in \partial\mathcal{V}} w'_{i,v}$, where $w'_{i,v}$ is the weight assigned to the edge connecting vertices i and v. Since $\mathbf{Q}$ is equal to a generalized Laplacian, the corresponding GGFT is the optimal decorrelating transform for the 1D case. Note that in the 1D case the DCT and ADST can be viewed as special cases of this GGFT. The GGFT defaults to the DCT [14] if $\alpha_a = 0$ and $\alpha_b = 1$ in (11). $\alpha_b = 1$ means that the target pixel block is smooth. $\alpha_a = 0$ means $\sigma_{g_a}^2 \gg \sigma_{g_0}^2$ and x_0 and x_1 are disconnected; in this case intra-prediction is not performed. Practically this occurs when the inter-pixel difference across the block boundary is large. If $\alpha_a = \alpha_b = 1$ in (11), the GGFT is the ADST [2]. $\alpha_a = 1$ means the signal is smooth across the block boundary, so that predicting by subtracting x_0 is effective [2].

Next, we propose the 2D GGFT design[2]. First, we build a 4-connected graph connecting adjacent pixels in the target block, as illustrated in Fig. 2(a) where pixels and links in blue are defined as $\mathcal{V}$ and $\mathcal{E}$ respectively. Second, we connect vertices in the first row and first column to boundary vertices

[2]Note that we are not claiming optimality of the 2D GGFT, and instead we use the 1D analysis to provide insights to design the 2D approach, similar to the approach followed in [2] for the ADST.

1-hop from the target block, *i.e.*, pixels and links in red in Fig. 2(b) are defined as $\partial\mathcal{V}$ and $\partial\mathcal{E}$. Then, for any two vertices i and j that are neighbors in the 4-connected graph, ordered so that i is reached first in a standard raster scan of the block, we define the corresponding weight $w_{i,j} = \alpha_k$, if $\hat{\mu}_k = f(x_i - x_j)$. We similarly define the weights of edges connecting nodes in $\partial\mathcal{V}$ to nodes in $\mathcal{V}$ as shown in Fig. 2(b), i.e., $w'_{i,j} = \alpha_k$, if $\hat{\mu}_k = f(x_i - x_j)$, where $i \in \partial\mathcal{V}$ and $j \in \mathcal{V}$, and we define $\mathbf{D}'$ as in the 1D case, by computing the additional degree of any nodes connected to boundary nodes.

The computational complexity for the GGFT includes that of: 1) eigen-decomposition of the $N \times N$ graph Laplacian, which is $O(N^3)$; and 2) signal projection onto the GGFT basis, which is $O(N^2)$. While not implemented here, the complexity of the eigen-decomposition can be reduced by pre-computing and storing the commonly used GGFT basis in a table for lookup, as done in [3]. Further, the direct signal projection can be approximated by a lifting transform, as done in [10].

IV. GGFT Coding System

We use four clusters to capture correlations of pixel pairs: *strong correlation* cluster ($\hat{\mu}_0 = 0$), two *weak correlation* clusters ($\hat{\mu}_{-1} < 0 < \hat{\mu}_1$, where $\hat{\mu}_{-1} = -\hat{\mu}_1$) and *zero correlation* cluster (bins at two ends of the distribution). We assign graph edge weights $\alpha_0 = 1$, α_{-1}, α_1 and $\alpha_{-2} = \alpha_2 = 0$, respectively for each of the clusters. Note that the weights are forced to be symmetric for bit rate reduction. For the zero correlation cluster, we do not perform intra-prediction since $\sigma^2_{g_a} \gg \sigma^2_{g_0}$ in this case.

We encode the clusters as follows. The zero and weak correlation clusters typically occur across object contours, which we losslessly encode on an image basis using *arithmetic edge coding* (AEC) [15], [16] and send as side information. Further, we encode extra bits to differentiate among zero correlation cluster and positive / negative weak correlation clusters using arithmetic coding.

We extend the 1D prediction in (7) to 2D as follows. We first employ (7) to predict a 4×4 block in horizontal and vertical directions separately. We do not perform prediction across zero correlation pixel pairs. If one pixel in the target block contains both horizontal and vertical predictors, then the average of the two predictors is used as the final prediction. Our intra-prediction is analogous to planar prediction in H.264 in the absence of discontinuities, and is used here as a proof-of-concept. Note that in theory we can design a GGFT for any prediction mode, including directional prediction, planar prediction in H.264/HEVC, etc, where the correlations in the prediction residual across pixel boundaries are incorporated as edge weights in a graph to define appropriate GGFT. Finally, we perform transform coding of the 4×4 prediction residual block using the GGFT (a 16×16 matrix) as described above.

At the decoder, we first perform inverse quantization and transform to reconstruct the residual block. Encoded side information is used to identify the transform and intra-prediction chosen at the encoder. The block is then reconstructed by adding the predictor to the residual block.

V. Experimental Results

We use both PWS images (depth maps `Teddy`, `Cones` [17] and `Ballet`, and graphics images `Dude` and `Tsukuba` [18]) and natural images (`Carphone`, `Girl`, `BasketballDrill`, `Peppers` and `Cameraman` [19]) for system evaluation. In the training stage, we use the clustering method to compute $\hat{\mu}_1$ and $\hat{\mu}_{-1}$ for each test image, and estimate the corresponding weights α_1 and α_{-1}.

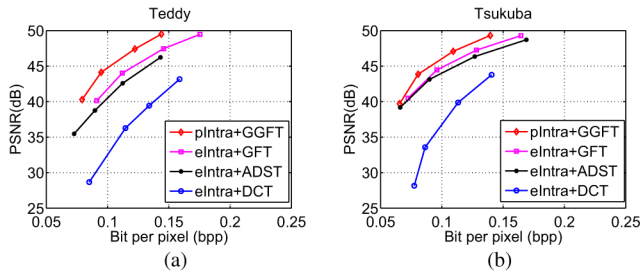

Fig. 3. RD performance comparison among different compression schemes. (a) Teddy (b) Tsukuba.

TABLE I
Average Gain in PSNR Measured with the Bjontegaard Metric

Image	eIntra+GFT	eIntra+ADST	eIntra+DCT
Teddy	1.40	3.48	10.76
Cones	0.63	7.25	12.88
Tsukuba	1.97	2.36	13.28
Dude	3.46	4.59	5.26
Ballet	0.79	3.94	9.16
Carphone	0.59	1.13	1.96
Girl	0.42	0.31	1.74
Peppers	0.22	0.19	1.24
Cameraman	0.16	0.75	1.35
BaskeballDrill	0.39	1.02	1.80

Fig. 4. The subjective quality comparison among different compression schemes. (a)–(c) `Teddy` at 0.10 bpp. (a) eIntra+DCT (b) eIntra+GFT (c) pIntra+GGFT.

We compare coding performance of our proposed intra-prediction and GGFT (`pIntra` + `GGFT`) against three compression schemes: edge-aware intra-prediction [20] with the DCT (`eIntra` + `DCT`), with the ADST (`eIntra` + `ADST`), and with the GFT (`eIntra` + `GFT`) constructed from the same graphs as those used in the GGFT but without the extra boundary edge weights added to the graph Laplacian. Fig. 3 presents the RD performance of these schemes for `Teddy` and `Tsukuba` with a typical PSNR range. More results are shown in Table I with the average gain in PSNR measured by the Bjontegaard metric. On average `pIntra` + `GGFT` achieves 5.9 dB gain over `eIntra` + `DCT`, 2.5 dB gain over `eIntra` + `ADST`, and 1.0 dB gain over `eIntra` + `GFT`.

Fig. 4 shows images reconstructed from different schemes for `Teddy`. `pIntra` + `GGFT` produces images with the sharpest boundaries and cleanest surfaces among all methods, with much fewer blocking artifacts. This is due to the cluster-based intra-prediction and GGFT.

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

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

[3] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multi-resolution graph fourier transform for compression of piecewise smooth images," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 419–433, Jan. 2015.

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs," *IEEE Signal Process. Mag.*, pp. 83–98, May 2013.

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

[6] Y. Wang, A. Ortega, and G. Cheung, "Intra predictive transform coding based on predictive graph transform," in *IEEE Int. Conf. Image Processing*, Melbourne, Australia, Sep. 2013.

[7] Y. Ye and M. Karczewicz, "Improved h.264 intra coding based on bidirectional intra prediction, directional transform, and adaptive coefficient scanning," in *15th IEEE Int. Conf. Image Processing*, San Diego, CA, USA, Oct. 2008, pp. 2116–2119.

[8] G. Shen, W.-S. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *IEEE Picture Coding Symp.*, Nagoya, Japan, Dec. 2010.

[9] S. Narang and A. Ortega, "Lifting based wavelet transforms on graphs," in *Proc. APSIPA ASC 2009*, Sapporo, Japan, 2009, pp. 441–444.

[10] Y. Chao, A. Ortega, W. Hu, and G. Cheung, "Edge-adaptive depth map coding with lifting transform on graphs," in *Picture Coding Symp.*, Cairns, Australia, May 2015.

[11] A. Gersho and R. M. Gray*, Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer, 1992.

[12] W. Hu, G. Cheung, X. Li, and O. Au, "Depth map compression using multi-resolution graph-based transform for depth-image-based rendering," in *IEEE Int. Conf. Image Processing*, Orlando, FL, USA, Sep. 2012.

[13] T. Biyikoglu, J. Leydold, and P. F. Stadler, "Nodal domain theorems and bipartite subgraphs," Dept. Appl. Statist. Data Process., Vienna Univ. Economics and Business, Vienna, Austria.

[14] G. Strang, "The discrete cosine transform," *SIAM Review*, vol. 41, pp. 135–147, 1999.

[15] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *IEEE Int. Conf. Image Processing*, Orlando, FL, Sep. 2012.

[16] I. Daribo, D. Florencio, and G. Cheung, "Arbitrarily shaped motion prediction for depth video compression using arithmetic edge coding," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4696–4708, Nov. 2014.

[17] "The middlebury stereo data," [Online]. Available: http://vision.middlebury.edu/stereo/data/scenes2003/

[18] "The Tsukuba dataset," [Online]. Available: http://www.cvlab.cs.tsukuba.ac.jp/dataset/tsukubastereo.php

[19] "The database for general images," [Online]. Available: http://www.csee.wvu.edu/xinl/database.html

[20] G. Shen, W.-S. Kim, A. Ortega, J. Lee, and H. Wey, "Edge-aware intra prediction for depth-map coding," in *IEEE Int. Conf. Image Processing*, Hong Kong, Sep. 2010.

This paper previously published in IEEE Signal Processing Letters