

# Multiresolution Graph Fourier Transform for Compression of Piecewise Smooth Images

Wei Hu, *Student Member, IEEE*, Gene Cheung, *Senior Member, IEEE*,  
Antonio Ortega, *Fellow, IEEE*, and Oscar C. Au, *Fellow, IEEE*

**Abstract**—Piecewise smooth (PWS) images (e.g., depth maps or animation images) contain unique signal characteristics such as sharp object boundaries and slowly varying interior surfaces. Leveraging on recent advances in graph signal processing, in this paper, we propose to compress the PWS images using suitable graph Fourier transforms (GFTs) to minimize the total signal representation cost of each pixel block, considering both the sparsity of the signal’s transform coefficients and the compactness of transform description. Unlike fixed transforms, such as the discrete cosine transform, we can adapt GFT to a particular class of pixel blocks. In particular, we select one among a defined search space of GFTs to minimize total representation cost via our proposed algorithms, leveraging on graph optimization techniques, such as spectral clustering and minimum graph cuts. Furthermore, for practical implementation of GFT, we introduce two techniques to reduce computation complexity. First, at the encoder, we low-pass filter and downsample a high-resolution (HR) pixel block to obtain a low-resolution (LR) one, so that a LR-GFT can be employed. At the decoder, upsampling and interpolation are performed adaptively along HR boundaries coded using arithmetic edge coding, so that sharp object boundaries can be well preserved. Second, instead of computing GFT from a graph in real-time via eigen-decomposition, the most popular LR-GFTs are pre-computed and stored in a table for lookup during encoding and decoding. Using depth maps and computer-graphics images as examples of the PWS images, experimental results show that our proposed multiresolution-GFT scheme outperforms H.264 intra by 6.8 dB on average in peak signal-to-noise ratio at the same bit rate.

**Index Terms**—Image compression, graph Fourier transform, piecewise smooth images.

## I. INTRODUCTION

A POPULAR approach to image compression is *transform coding* [1]: an image is first divided into non-overlapping blocks of pixels, with each block projected to a chosen transform domain, and the resulting transform coefficients are quantized, entropy-coded and transmitted to the decoder. While one can apply a transform to an entire image,

Manuscript received April 14, 2014; revised August 26, 2014; accepted November 4, 2014. Date of publication December 4, 2014; date of current version December 23, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Christine Guillemot.

W. Hu and O. C. Au are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: huwei@ust.hk; eeau@ust.hk).

G. Cheung is with the National Institute of Informatics, Graduate University for Advanced Studies, Tokyo 113-0033, Japan (e-mail: cheung@nii.ac.jp).

A. Ortega is with the 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/TIP.2014.2378055

most compression codecs (e.g., JPEG,<sup>1</sup> H.264 [2]) employ transform coding block-by-block. This is because block-based transforms can adapt to the non-stationary statistics of natural images and are more amenable to hardware implementation. We focus on block-based transform coding of images in this paper.

A key to good compression performance is to select a transform that promotes *sparsity in representation*; i.e., upon signal transformation there remain few non-zero transform coefficients, resulting in coding gain. For a vector following a known correlation model (statistically described by a covariance matrix), the *Karhunen-Loève Transform* (KLT) is the optimal transform in terms of decorrelating components of the vector<sup>2</sup> [7]. In order to use the KLT for coding of non-overlapping pixel blocks in an image, one would first adaptively classify each block into one of several statistical classes, then apply the KLT corresponding to the identified statistical class for decorrelation. However, description of the identified statistical class for the coded block (and hence the transformation employed) must be conveyed to the decoder for correct transform decoding. Given this transform description overhead, how to suitably select a good set of statistical classes for KLT transform coding to achieve good overall performance—in terms of *both* sparsity of signal representation *and* compactness in transform description—is the challenge.

The *Discrete Cosine Transform* (DCT), on the other hand, is a fixed transform with no signal adaptation, thus it requires no description overhead. It is equivalent to the KLT for the single statistical class where the correlation between each pair of adjacent pixels is assumed to be 1. The DCT is widely used in image compression systems, such as the JPEG and H.26x codecs [8]. However, the assumption that the pixel correlation tends to 1 is not always true. In particular, adjacent pixels across sharp boundaries clearly have correlation much smaller than one. This means that for blocks that straddle sharp boundaries, the DCT will lead to non-sparse signal representation, resulting in poor coding performance.

In order to achieve a balance between signal representation and transform description, we propose to use *graph Fourier transforms* (GFTs) [9] for transform coding of images. GFT is

<sup>1</sup>Available at <http://www.jpeg.org/jpeg/>

<sup>2</sup>Optimality in decorrelating a vector statistically, however, does not necessarily imply optimality in transform coding. It is shown that the KLT is optimal if the input vector is jointly Gaussian or a mixture of Gaussians, for variable-rate high-resolution coding [3]–[5]. For other sources, the KLT may yield sub-optimal performance [6].

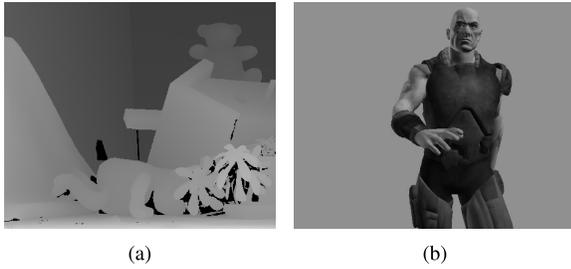


Fig. 1. Examples of PWS images: (a) a depth map of an image sequence Teddy; (b) a computer-graphics image Duke.

a matrix of eigenvectors of the graph Laplacian matrix of a constructed graph. For transform coding, the graph nodes represent pixels in a coding block, and each edge weight represents correlation between two connected (adjacent) pixels. Given a defined search space of graphs that connect pixels in a given block, an “optimal” graph (and hence corresponding “optimal” GFT) minimizes the total signal representation cost, evaluating *both* the sparsity of the signal’s transform coefficients *and* the compactness of transform description.

In particular, we propose to use GFTs for transform coding of a class of images—*piecewise smooth* (PWS) images. PWS images consist of sharp boundaries and smooth interior surfaces. They have recently received renewed attention in the image processing community, due to advances in depth sensing technologies, such as Microsoft Kinect, that capture fairly accurate *depth maps* in real-time. A depth image is a per-pixel map that measures the physical distance between objects in the 3D scene and the capturing camera. Because textural contents of objects are not captured, depth images are smooth within an object’s surface, and hence PWS. Depth maps can be used for advanced image processing tasks, such as foreground/background segmentation, 3D motion estimation [10], depth-image-based rendering (DIBR) [11], etc. Other examples of PWS images include animation images, *magnetic resonance imaging* (MRI) images, dense motion fields [12], computer-graphics images, etc. See Fig. 1 for examples of PWS images. Unique signal characteristics of sharp boundaries and smooth interior surfaces in PWS images mean that conventional coding schemes designed for natural images such as JPEG do not offer good compression performance. We focus on the compression of PWS images in this paper.

Our proposed GFT-based coding scheme for PWS images, called *multi-resolution GFT* (MR-GFT), has three main contributions. First, we consider a large space of possible graphs, including unweighted graphs *and* weighted graphs, to exploit the generality afforded by GFT. Further, we define an optimality criterion of GFT for compression that aims to minimize the total representation cost, including transform domain representation *and* transform description. To enable low GFT description cost, we restrict our feasible space of GFTs to graphs with simple connectivity and a small discrete set of edge weights. Discrete edge weights are derived from a statistical analysis under a model specifically designed to characterize PWS signals, for which we demonstrate that the derived GFT approximates the KLT.

Second, given a defined search space of graphs (or equivalently GFTs), we search for the optimal GFT for each pixel block with our proposed efficient algorithms. In particular, for ease of optimization we divide the search space into two subspaces—GFTs on weighted graphs and GFTs on unweighted graphs. For GFTs on weighted graphs, we formulate a *separation-deviation* (SD) problem and solve it using a known efficient graph cut algorithm [13]. Meanwhile, we search for good GFTs on unweighted graphs via a greedy algorithm, leveraging on a graph partitioning technique based on spectral clustering [14].

Third, for practical implementation, we design two techniques to reduce computation complexity. In the first technique, we propose a multi-resolution (MR) approach, where detected object boundaries are encoded in the original high resolution (HR), and smooth surfaces are low-pass-filtered and down-sampled to a low-resolution (LR) one, before performing LR GFT for a sparse transform domain representation. At the decoder, after recovering the LR block via inverse GFT, we perform up-sampling and interpolation adaptively along the encoded HR boundaries, so that sharp object boundaries are well preserved. The key insight is that *on average PWS signals suffer very little energy loss during edge-adaptive low-pass filtering*, which enables the low-pass filtering and down-sampling of PWS images. This MR technique also enables us to perform GFT on large blocks, resulting in large coding gain. In the second technique, instead of computing GFT from a graph in real-time via eigen-decomposition of the graph Laplacian matrix, we pre-compute and store the most popular LR-GFTs in a table for simple lookup during actual encoding and decoding. Further, we exploit graph isomorphism to reduce the number of GFTs required for storage to a manageable size. Using depth maps and graphics images as examples of PWS images, experimental results show that our proposed MR-GFT scheme outperforms H.264 intra (with intra prediction used) by 6.8 dB on average in peak signal-to-noise ratio (PSNR). Further, when applying compressed depth maps to DIBR, we achieve 2.2 dB gain on average in synthesized view PSNR compared to H.264 intra.

The outline of the paper is as follows. We first discuss related work in Section II. We then overview our proposed MR-GFT coding system in Section III and review the definition of GFT in Section IV. Next, we present a mathematical analysis of optimal GFTs in Section V. The problem formulations and algorithms for optimal GFTs on two types of graph supports are discussed in Section VI. We then outline the detailed implementation in Section VII. Finally, experimental results and conclusions are presented in Section VIII and Section IX, respectively.

## II. RELATED WORK

The field of PWS image compression encompasses diverse source coding schemes, ranging from segmentation-based techniques to popular transform approaches. In general, they feature efficient representations of the geometric structures in PWS images. We discuss them in order as follows.

Segmentation-based compression schemes segment the image into homogeneous partitions followed by coding of each partition. One of the most popular is quadtree-based compression [15]–[17], which recursively divides the image into simple geometric regions. [15] designs the quadtree segmentation in an optimal Rate-Distortion (RD) framework and then approximates each segment using a polynomial model separated by a linear boundary. The direction of the boundary, however, is chosen from a given limited discrete set, which may not be sufficiently descriptive to describe arbitrarily shaped boundaries. In contrast, our proposed MR-GFT can represent any boundary accurately and efficiently via *arithmetic edge coding* (AEC) of HR edges [18], [19]. In [16] depth maps are modeled by piecewise-linear functions (called *platelets*) separated by straight lines, which are adaptive to each subdivision of the quadtree with variable sizes in a global RD tradeoff. However, this representation has non-vanishing approximation error, since depth maps are not exactly piecewise linear. In contrast, our proposed MR-GFT can accurately represent PWS images in GFT domain.

Transform approaches are also designed for PWS image compression. An ensemble of transforms exploit the geometry of PWS images, such as the wavelet-domain compression [20], [21], curvelet [22], and contourlet [23]. However, they are all deployed over the entire image, and so are not easily amenable to block-based processing for hardware-friendly implementation. In contrast, our proposed MR-GFT is a block-based transform coding approach.

GFT is first used for depth map coding in [24], which empirically demonstrated that using GFT for transform coding of a depth pixel block with sharp boundaries, one can achieve a sparser transform domain representation than with DCT. (Sparsity in representation of depth block in GFT domain is also used as a signal prior in depth image denoising in [25].) However, [24] has the following shortcomings. First, given an input pixel block, [24] considers only a single variant of GFT based on an unweighted graph, while we consider a much larger space of possible graphs, including *both* unweighted *and* weighted graphs. Second, graphs are deduced directly from detected boundaries in [24], thus it is not clear if the construction of graphs is optimal in any sense. In contrast, we formally define an optimality criterion that reflects representation cost and propose efficient algorithms to search for graphs that lead to optimal GFTs. Finally, for each  $\sqrt{N} \times \sqrt{N}$  pixel block, [24] requires real-time eigen-decomposition of an  $N \times N$  graph Laplacian matrix to derive basis vectors at *both* the encoder and decoder. This large computation burden also means that GFT cannot be practically performed for larger blocks in PWS images, which will otherwise result in larger coding gain. To address this issue, we design two techniques to reduce computation complexity, namely, the MR scheme and table lookup. Experimental results show that we achieve 5.9 dB gain on average in PSNR over [24].

Drawing a connection between GFT and a graphical statistical model called *Gaussian Markov Random Field* (GMRF), [26] provides a theoretical analysis on the optimality of GFT in terms of decorrelating the input vector under the GMRF model. Unlike [26], we select a

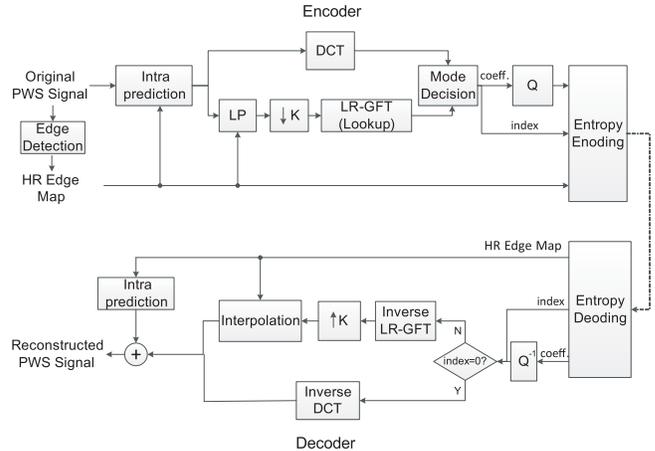


Fig. 2. MR-GFT coding system for PWS images.

GFT for each coding block by choosing from a set of candidate GFTs using an optimality criterion that considers *both* the sparsity of the GFT domain signal representation *and* the compactness of the transform description.

Image compression specifically tailored for depth maps, a representative class of PWS images, has been studied extensively in recent years. Beyond the class of techniques that exploit the piecewise smooth characteristics of depth signals [16], [17], [21] as previously discussed, another class of depth coding techniques are tailored specifically for virtual view rendering via DIBR [27]–[32]. Specifically, they optimize depth coding by considering the impact of depth errors due to lossy image compression on the DIBR-synthesized views. [27] and [28] utilize synthesized view distortion as metric for depth coding optimization, such as mode selection. [29] and [30] use synthesized view distortion as objective for *transform domain sparsification* (TDS). In [31], view synthesis is incorporated for improved depth prediction in a RD optimized framework. [32] proposes down-sampling/up-sampling techniques in an optimized fashion by incorporating the effect of depth re-sampling on view synthesis quality. In contrast, since we focus on compression of general PWS images, we do not consider synthesized view distortion as an optimization metric. Nevertheless, our proposed method implicitly leads to good synthesized views via edge-preserving of depth maps, as shown in Section VIII.

### III. MULTI-RESOLUTION GRAPH FOURIER TRANSFORM CODING SYSTEM

We first provide an overview of our proposed MR-GFT coding system for compression of PWS images, shown in Fig. 2. Given a PWS image, we discuss the encoding and decoding procedures as follows.

#### A. Encoder

At the encoder, we first detect prominent boundaries (large inter-pixel intensity difference) in the HR image via hard thresholding of image gradients. The threshold is set based on the mean and variance of the image, so that the boundary detection is adaptive to the image statistics.

We encode HR boundaries losslessly for the adaptive intra prediction and interpolation at the decoder (discussed later), using AEC [18], [19]. We apply AEC for the entire image, which avoids initialization for each block and efficiently encodes long continuous boundaries in the image. Then for each  $K\sqrt{N} \times K\sqrt{N}$  target pixel block considering a down-sampling factor  $K = 2$ , we execute the following three steps.

First, we perform edge-aware intra prediction as proposed in [33]. Different from the intra prediction in H.264 [2], edge-aware intra prediction efficiently reduces the energy of the prediction error by predicting within the confine of detected HR boundaries, thus reducing bits required for coding of the residual signal.

Second, we try two types of transforms for transform coding of the residual block: i) fixed DCT on the original HR residual block (HR-DCT), ii) a pre-computed set of LR GFT (LR-GFT) on the down-sampled LR residual block (including LR weighted GFT and LR unweighted GFT, as discussed in Section V). We then choose the one transform with the best RD performance. Before transform coding using LR-GFT, however, we first adaptively low-pass-filter and down-sample the  $K\sqrt{N} \times K\sqrt{N}$  block uniformly to a  $\sqrt{N} \times \sqrt{N}$  block. Low-pass filtering is first used to avoid aliasing caused by down-sampling. We propose an edge-adaptive low-pass filter in the pixel domain for the preservation of sharp boundaries. Specifically, a pixel is low-pass-filtered by taking average of its neighbors on the same side of HR boundaries within a  $(2K - 1) \times (2K - 1)$  window centering at the to-be-filtered pixel. The advantage of this edge-adaptive low-pass filtering is that filtering across arbitrary-shape boundaries will not occur, so pixels across boundaries will not contaminate each other through filtering.

For the implementation of the HR-DCT and LR-GFT, we pre-compute the optimal transforms (discussed in Section VI) and store them in a lookup table (discussed in Section VII) *a priori*. During coding, we try each one and choose the one with the best RD performance. The two types of transforms, HR-DCT and LR-GFT, are employed to adapt to different block characteristics. HR-DCT is suitable for blocks where edge-adaptive low-pass filtering would result in non-negligible energy loss. If very little energy is lost during low-pass filtering, LR-GFT would result in a larger coding gain. Note that if a given block is smooth, the LR-GFT will default to the DCT in LR, and would generally result in a larger gain than HR-DCT due to down-sampling (the rates of transform indices for both, *i.e.*, the transform description overhead, are the same in this case).

Third, after the RD-optimal transform is chosen from the two transform candidates, we quantize and entropy-encode the resulting transform coefficients for transmission to the decoder. The transform index identifying the chosen transform is also encoded, so that proper inverse transform can be performed at the decoder.

### B. Decoder

At the decoder, we first perform inverse quantization and inverse transform for the reconstruction of the residual block.

The transform index is used to identify the transform chosen at the encoder for transform coding.

Secondly, if LR-GFT is employed, we up-sample the reconstructed  $\sqrt{N} \times \sqrt{N}$  LR residual block to the original resolution  $K\sqrt{N} \times K\sqrt{N}$ , and then fill in missing pixels via our proposed image-based edge-adaptive interpolation [34], where a pixel  $x$  is interpolated by taking average of its neighboring pixels on the same side of boundaries within a  $(2K - 1) \times (2K - 1)$  window centering at pixel  $x$ .

Finally, the  $K\sqrt{N} \times K\sqrt{N}$  block is reconstructed by adding the intra predictor to the residual block.

## IV. GRAPH FOURIER TRANSFORM FOR IMAGES

Before we proceed to problem formulation and algorithms for optimal GFTs, we first review the basics of GFT.

A graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$  consists of a finite set of vertices  $\mathcal{V}$  with cardinality  $|\mathcal{V}| = N$ , a set of edges  $\mathcal{E}$  connecting vertices, and a weighted adjacency matrix  $\mathbf{W}$ .  $\mathbf{W}$  is a real  $N \times N$  matrix, where  $W_{i,j}$  is the weight assigned to the edge  $(i, j)$  connecting vertices  $i$  and  $j$ . We consider here only undirected graphs, which correspond to symmetric weighted adjacency matrices, *i.e.*,  $W_{i,j} = W_{j,i}$ . We also assume weights are non-negative, *i.e.*,  $W_{i,j} \geq 0$ .

While there exist different variants of Laplacian matrices, we are interested in the unnormalized combinatorial graph Laplacian in this work, which is defined as  $\mathcal{L} := \mathbf{D} - \mathbf{W}$ , where  $\mathbf{D}$  is the *degree matrix*—a diagonal matrix whose  $i$ th diagonal element is the sum of all elements in the  $i$ th row of  $\mathbf{W}$ , *i.e.*,  $D_{i,i} = \sum_{j=1}^N W_{i,j}$ . Since the Laplacian matrix is a real symmetric matrix, it admits a set of real eigenvalues  $\{\lambda_l\}_{l=0,1,\dots,N-1}$  with a complete set of orthonormal eigenvectors  $\{\psi_l\}_{l=0,1,\dots,N-1}$ , *i.e.*,  $\mathcal{L}\psi_l = \lambda_l\psi_l$ , for  $l = 0, 1, \dots, N-1$ . We employ this Laplacian matrix for two reasons.

First, because elements in each row of  $\mathcal{L}$  sum to zero by construction, 0 is guaranteed to be an eigenvalue with  $[1 \dots 1]^T$  as the corresponding eigenvector. This means a frequency interpretation of GFT, where the eigenvalues  $\lambda_l$ 's are the graph frequencies, will always have a DC component, which is beneficial for the compression of PWS images where most regions are smooth.

Second, GFT defaults to the well known DCT when defined for a line graph (corresponding to the 1D DCT) or a 4-connectivity graph (2D DCT) with all edge weights equal to 1 [26]. That means GFT is at least as good as the DCT in sparse signal representation if the weights are chosen in this way. Due to the above two desirable properties, we use the unnormalized Laplacian matrix in our definition of GFT.

We note that the graph Laplacian can be used to describe the total variation of the signal with respect to the graph; *i.e.*, for any signal  $\mathbf{x} \in \mathbb{R}^N$  residing on the vertices of a graph with the graph Laplacian  $\mathcal{L}$ , we can write [35]

$$\mathbf{x}^T \mathcal{L} \mathbf{x} = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N W_{i,j} (x_i - x_j)^2. \quad (1)$$

$\mathbf{x}^T \mathcal{L} \mathbf{x}$  is small when  $\mathbf{x}$  has similar values at each pair of vertices  $i$  and  $j$  connected by an edge, *or* when the weight  $W_{i,j}$  is small for an edge connecting  $i$  and  $j$  with dissimilar

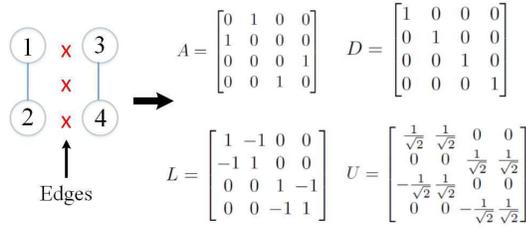


Fig. 3. An example of constructing GFT from a  $2 \times 2$  pixel block. The vertical contour separates pixel 1 and 2 from pixel 3 and 4, and a graph is constructed by connecting pixels on each side of the contour. The corresponding adjacency matrix  $\mathbf{W}$ , degree matrix  $\mathbf{D}$ , Laplacian matrix  $\mathcal{L}$  as well as the computed GFT  $\mathbf{U}$  are shown on the right.

values. Thus, a signal is smooth (mostly low-frequency components) with respect to a graph if the edge weights capture the similarity of connected pixels in the signal. Since  $\mathbf{x}^T \mathcal{L} \mathbf{x}$  is small if a signal is smooth on the graph (thus a sparse representation in the GFT domain), we will use  $\mathbf{x}^T \mathcal{L} \mathbf{x}$  to select graphs that lead to good compression results in later sections. In particular, we will discuss how we use (1) as the rate proxy of transform coefficients during the search for optimal GFTs in Section VI.

The eigenvectors  $\{\psi_l\}_{l=0,1,\dots,N-1}$  of the Laplacian matrix are then used to define the GFT. Formally, for any signal  $\mathbf{x} \in \mathbb{R}^N$  residing on the vertices of  $\mathcal{G}$ , its GFT  $\hat{\mathbf{x}} \in \mathbb{R}^N$  is defined in [9] as

$$\hat{\mathbf{x}}(l) = \langle \psi_l, \mathbf{x} \rangle = \sum_{n=1}^N \psi_l^*(n) \mathbf{x}(n), \quad l = 0, 1, \dots, N - 1. \quad (2)$$

The inverse GFT follows as

$$\mathbf{x}(n) = \sum_{l=0}^{N-1} \hat{\mathbf{x}}(l) \psi_l(n), \quad n = 1, 2, \dots, N. \quad (3)$$

Having defined the GFT, we give an example of how to construct a GFT given an unweighted graph defined for a pixel block. Given a  $\sqrt{N} \times \sqrt{N}$  pixel block, we first treat each pixel in the block as a vertex in a graph  $\mathcal{G}$  and connect it to its four immediate neighbors, resulting in a 4-connectivity graph. See Fig. 3 for an illustration. Second, if there is a large discrepancy in values between two neighboring pixels, we eliminate their connection. Given the connectivity graph, we can define the adjacency matrix  $\mathbf{W}$ , where  $W_{i,j} = W_{j,i} = 1$  if pixel positions  $i$  and  $j$  are connected, and 0 otherwise. The degree matrix  $\mathbf{D}$  can then be computed. In the third step, using computed  $\mathbf{W}$  and  $\mathbf{D}$ , we compute the graph Laplacian matrix  $\mathcal{L} = \mathbf{D} - \mathbf{W}$ . We then stack pixels in the  $\sqrt{N} \times \sqrt{N}$  patch into a length- $N$  vector  $\mathbf{x}$  and compute the GFT according to (2).

## V. OPTIMAL GRAPH FOURIER TRANSFORMS FOR PIECEWISE SMOOTH IMAGES

We now define a notion of optimality of GFT for compression of PWS images. Towards a formal definition of an optimization problem, we then define the search space for

GFT as a discrete set of graph supports with edge weights drawn from a small discrete set. The weights are later derived assuming a specific statistical model for PWS images, where we demonstrate that the derived GFT approximates the KLT under this model.

### A. Optimality Definition for Graph Fourier Transforms

In lossy image compression, different coding systems are compared based on their RD performance, which describes the trade-off between the coding rate and total induced distortion. Specifically, a typical system design seeks to minimize a weighted sum of rate and distortion for chosen weighting parameters. However, assuming high bit rate, [36] shows that a uniform quantizer yields the following expected distortion:

$$D = \frac{Nq^2}{12}, \quad (4)$$

where  $q$  is the quantization step size employed for each coefficient, and  $N$  is the total number of coefficients, which is the same for different orthogonal transforms. This indicates that the expected distortion does not change when considering different transforms under the same assumptions.

Hence, we only need to consider the total coding rate. Given a pixel block  $\mathbf{x} \in \mathbb{R}^N$ , the GFT representation derived from the graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$  constructed on  $\mathbf{x}$  has two representation costs: i) the cost of transform coefficient vector  $\alpha$  denoted by  $R_\alpha(\mathbf{x}, \mathbf{W})$ , and ii) the cost of transform description  $T$  denoted by  $R_T(\mathbf{W})$ . We thus arrive at the following definition of optimality for GFTs:

*Definition:* For a given image block  $\mathbf{x} \in \mathbb{R}^N$  under fixed uniform quantization at high bit rate, an *optimal* GFT is the one that minimizes the total rate, *i.e.*:

$$\begin{aligned} \min_{\mathbf{W}} \quad & R_\alpha(\mathbf{x}, \mathbf{W}) + R_T(\mathbf{W}) \\ \text{s.t.} \quad & W_{i,j} \in \mathcal{C} \quad \forall i, j \in \mathcal{V} \end{aligned} \quad (5)$$

where  $\mathcal{C}$  is the feasible set of edge weights.

Note that while an edge weight could in general take on any non-negative real value, we restrict weights to a small discrete set  $\mathcal{C}$  in order to enable low description cost  $R_T$  for GFT. This is further discussed next.

### B. Definition of Search Space for GFTs

To lower the description cost  $R_T$  for GFT in optimization (5), instead of a full graph we assume a simpler 4-connectivity graph, where each vertex is only connected to its four adjacent neighbors. Further, we view pixels that are neighbors in the 4-connectivity graph as random variables, and consider a discrete set of possible weights to be chosen for each edge connecting two pixels. These weights correspond to three representative classes of the correlation between two pixels: 1) *strong correlation* between the pixels, which models pixel pairs occurring in smooth regions of the foreground/background; 2) *zero correlation* between the two pixels, occurring when they straddle sharp boundaries between the foreground and background; and 3) *weak correlation* between the pixels that correspond to distinctly different parts

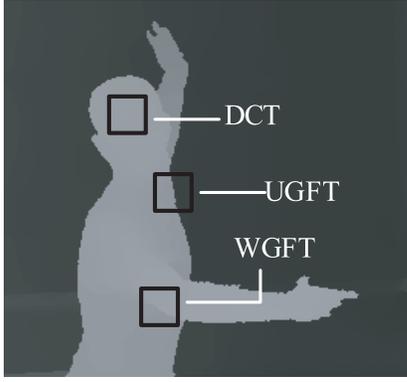


Fig. 4. An intuitive illustration of how different transforms adapt to pixel blocks with different pixel correlations in the depth map of Ballet.

of the same foreground/background object. See Fig. 4 for an illustration. Correspondingly, the weights are assigned to be  $\mathcal{C} = \{1, 0, c\}$ , where  $c$  is a carefully chosen real value between 0 and 1 (to be formally derived next).

Having defined the edge weight set, for ease of computation we further divide the optimization problem (5) into two sub-problems with two corresponding non-overlapping GFT search spaces:

- *Unweighted GFT* (UGFT), with  $\mathcal{C} = \{1, 0\}$  for blocks that can be well described by pixel pairs with strong and zero correlations only; and
- *Weighted GFT* (WGFT), with  $\mathcal{C} = \{1, c\}$  for blocks that can be well described by pixel pairs with strong and weak correlations only.

If a block can be well described by pixel pairs with strong correlation only (*i.e.*, a smooth block), then the GFT defaults to the DCT. See Fig. 4 for an illustration. In a nutshell, the WGFT considers only graphs that describe a single connected component (*i.e.*, only one DC coefficient in any WGFT representation of the signal). The UGFT considers graphs of multiple connected components, where each component is connected by edges with weight 1. (A connected component with a pair of disconnected neighboring pixels appears rarely in practice, and thus is not considered in the UGFT optimization.) Both search spaces are searched for a given input coding block for the best possible signal representation. The rationale for this division is twofold. First, if the coding block is sufficiently small, then the likelihood of a block containing pixel pairs with all the three classes of correlations is very small. Second, division into two independent searches for the UGFT and WGFT leads to efficient search algorithms, which will be presented in Section VI.

The key question now is: what is the most suitable value of  $c$ , one that correctly models weak correlation in the third class of pixel pairs, so that an optimal GFT in the WGFT search space has the best possible performance? We next derive the optimal  $c$  from statistical analysis under a model designed for PWS signals, where we demonstrate that the derived GFT approximates the KLT for the corresponding class of signals.

### C. Derivation of Optimal Edge Weights for Weak Correlation

For simplicity, we consider the derivation of the optimal edge weight  $c$  in one dimension. We note, however, that the optimality of our derived edge weight  $c$ —by extension the optimality of the corresponding GFT—carries over to the more general 2D setting where the locations of smooth and weak transition regions  $\mathcal{S}$  and  $\mathcal{P}$  are known deterministically, and the statistics of the random variables involved in the model are known probabilistically.

Given the unique characteristics of PWS images, we assume a piecewise first-order autoregressive process  $\mathbf{x} = [x_1, \dots, x_N]^T$  with *independent and identically distributed* (i.i.d.) standard normal noise  $e_k \sim \mathcal{N}(0, 1)$ ,  $k = 1, \dots, N$  [37]:

$$x_k = \begin{cases} \eta, & k = 1 \\ x_{k-1} + e_k, & 1 < k \leq N, [k-1, k] \in \mathcal{S} \\ x_{k-1} + g + e_k, & 1 < k \leq N, [k-1, k] \in \mathcal{P} \end{cases} \quad (6)$$

where we assume the first variable  $x_1$  to be  $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ . In the given smooth region  $\mathcal{S}$ ,  $x_k$  is assumed to be the sum of  $x_{k-1}$  and a standard normal noise, while across the known weak transition region  $\mathcal{P}$  (*e.g.*, from one smooth region of the foreground to another, as shown in Fig. 4),  $x_k$  is modeled as the sum of  $x_{k-1}$ , a random gap  $g \sim \mathcal{N}(m_g, \sigma_g^2)$ , and a standard normal noise.

Further, we assume there exists only one weak transition in a code block. Since experimentally we use a sufficiently small block size, it is very likely to have at most one weak transition in a block in each dimension. Assuming the only weak transition exists between  $x_{k-1}$  and  $x_k$ , then (6) can be expanded as:

$$\begin{aligned} x_1 &= \eta \\ x_2 - x_1 &= e_2 \\ &\dots \\ x_k - x_{k-1} &= g + e_k \\ &\dots \\ x_N - x_{N-1} &= e_N \end{aligned} \quad (7)$$

and further written into the matrix form:

$$\mathbf{F}\mathbf{x} = \mathbf{b}, \quad (8)$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ e_2 \\ \vdots \\ e_k \\ \vdots \\ e_N \end{bmatrix} + \begin{bmatrix} \eta \\ 0 \\ \vdots \\ g \\ \vdots \\ 0 \end{bmatrix} \quad (9)$$



the weight (17) of the edge that connects two pixels in the transition region  $\mathcal{P}$ .

## VI. ADAPTIVE SELECTION OF GRAPH FOURIER TRANSFORMS

Having derived the optimal edge weight  $c$  for weakly correlated pixel pair in a coding block—and by extension the optimal WGFT for decorrelation, ideally one can simply use that optimal WGFT for transform coding. However, this does not account for the cost of transform description  $R_T$  in (5). To account for both the signal representation cost in the GFT domain and the description cost of the chosen GFT, as discussed in Section V, we search for the optimal GFT that minimizes (5), and further divide the search space into two subspaces: 1) WGFT on weighted and connected graphs, and 2) UGFT on unweighted and disconnected graphs. In this section we present the respective problem formulations and algorithms in the two subspaces in detail.

### A. Weighted Graph Fourier Transform

The WGFT is derived from a weighted and connected graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ , where the feasible weight set is  $\mathcal{C} = \{1, c\}$ . For ease of optimization, we first propose plausible proxies for the two rate terms in (5), *i.e.*, the rates for transform coefficients and transform description. Then we cast the optimization problem of WGFT as a *separation-deviation* (SD) problem, in order to leverage on the well known and efficient algorithm developed for the SD problem in [13].

1) *Rate Proxy of Transform Coefficients*: Transform coefficients consist of DC and AC coefficients. Since the WGFT is constructed on a *connected* graph (*i.e.*, only one connected component in the graph), the zero eigenvalue is of multiplicity one, and each WGFT produces only one DC coefficient for a given block  $\mathbf{x}$ . The cost of DC coefficients of the WGFT is thus the same. We then approximate the cost of quantized AC coefficients (corresponding to non-zero eigenvalues), for a given quantization step size  $q$  and without consideration of rounding for simplicity, as follows:

$$\begin{aligned} \mathbf{x}^T \mathcal{L} \mathbf{x} / q^2 &= \mathbf{x}^T \left( \sum_{l=0}^{N-1} \lambda_l \psi_l \psi_l^T \right) \mathbf{x} / q^2 \\ &= \sum_{l=0}^{N-1} \lambda_l (\mathbf{x}^T \psi_l) (\psi_l^T \mathbf{x}) / q^2 \\ &= \sum_{l=0}^{N-1} \lambda_l (\alpha_l / q)^2, \end{aligned} \quad (18)$$

where  $\alpha_l$ ,  $l = 1, \dots, N-1$ , is the  $l$ -th transform coefficient. In words,  $\mathbf{x}^T \mathcal{L} \mathbf{x} / q^2$  is an eigenvalue-weighted sum of squared quantized transform coefficients. This also means that the DC coefficient is not reflected in (18). By minimizing (18), we suppress high-frequency coefficients. Recall that  $\mathbf{x}^T \mathcal{L} \mathbf{x}$  can also be written in the form in (1) in terms of edge weights and adjacent pixel differences. We thus propose the following

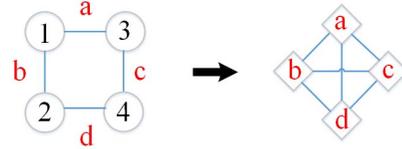


Fig. 6. An illustration for converting edges in original graph to vertices in the dual graph for SD formulation.

proxy for the cost of quantized AC coefficients:

$$\begin{aligned} R_\alpha(\mathbf{x}, \mathbf{W}) &= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N W_{i,j} (x_i - x_j)^2 / q^2 \\ &= \frac{1}{2} \rho \sum_{i=1}^N \sum_{j=1}^N W_{i,j} (x_i - x_j)^2, \end{aligned} \quad (19)$$

where  $\rho = 1/q^2$ . We note that [38] proposed to use (19) as the cost function for *all* GFT coefficients, even for disconnected graphs. As discussed, (19) excludes the cost of DC coefficients, which means that (19) fails to capture the variable cost of DC coefficients in cases where the graph contains a variable number of connected components. In our WGFT search space, we avoid this problem since our search space contains only graphs that have a single connected component, because 0 is excluded as a possible choice for an edge weight. Note that without this restriction, the all zero connection graph would be chosen as the optimal choice, which obviously does not lead to an optimal GFT in general.

Further, for ease of later algorithm development, we convert edges in the graph  $\mathcal{G}$  to nodes in a dual graph<sup>3</sup>  $\mathcal{G}^d$ , and define the AC coefficient cost (19) on  $\mathcal{G}^d$  instead. More specifically, we map each edge  $e$  that connects vertices  $v_1(e)$  and  $v_2(e)$  in  $\mathcal{G}$  to a *node*<sup>4</sup>  $v_e^d$  in  $\mathcal{G}^d$ , and assign node  $v_e^d$  with the value  $W_e = W_{v_1(e), v_2(e)}$ . A *link* is drawn between two nodes in the dual graph if the two corresponding edges in the original graph share the same vertex as an endpoint, or traverse the same contour in the coding block. See Fig. 6 for an example of the conversion.

Given the node set  $\mathcal{V}^d$  in the dual graph, (19) can now be rewritten as

$$R_\alpha(\mathbf{x}, \mathbf{W}) = \rho \sum_{e \in \mathcal{V}^d} W_e (x_{v_1(e)} - x_{v_2(e)})^2, \quad (20)$$

where a *label*  $W_e$  assigned to node  $e$  in the dual graph  $\mathcal{G}^d$  is the edge weight  $W_{v_1(e), v_2(e)}$  in the original graph  $\mathcal{G}$ . A label assignment  $W$  to a node  $e$  in a graph inducing difference cost is typically called a *deviation* cost in an SD problem [13].

2) *Rate Proxy of Transform Description*: Weighted graphs for a pixel block are defined by edges assigned weak correlation  $c$ ; the rest of the edges are assigned weight 1. Further, edges of weak correlation tend to collectively outline an object's contour, *i.e.*, fractional weight edges tend to appear consecutively. Hence, a rare WGFT—one selected with low

<sup>3</sup>The dual graph is also termed the *line graph* in the mathematical discipline of graph theory.

<sup>4</sup>To avoid confusion, we use the terminology of *vertex* and *edge* in the original graph, and *node* and *link* in the dual graph.

probability  $p$  or high entropy cost  $-\rho \log p$ —will have weak correlation edges scattered throughout the block. We thus propose a rate proxy to penalize these rare GFTs. Specifically, we use the label differential cost of nodes connected by links  $\mathcal{E}^d$  in the dual graph  $\mathcal{G}^d$  as the rate proxy of WGFT, *i.e.*,  $\sum_{(e,s) \in \mathcal{E}^d} |W_e - W_s|$ . Assigning different labels to nodes connected by a link will induce a *separation* cost in an SD problem [13].

Further, since pixels in a PWS image are more likely to be strongly correlated, we penalize a block with larger number of weak correlation edges by adding another term to the rate proxy:

$$R_T(\mathbf{W}) = \sum_{(e,s) \in \mathcal{E}^d} |W_e - W_s| + \sum_{e \in \mathcal{V}^d} \gamma \rho (1 - W_e), \quad (21)$$

where  $\rho$  is the same one as in (19), and  $\gamma$  is used to assign the importance of the second term relative to the first term. While the first term specifies the cost of encoding a given distribution of edges (consecutive fractional weight edges consume fewer bits as discussed earlier), the second term essentially counts the number of fractional weight edges in the block to encode. The actual relative importance between these two terms, *i.e.*, the assignment of  $\gamma$ , depends on the particular edge encoding scheme.

3) *Problem Formulation for WGFT*: Collecting the two proposed rate proxies together, we can now formulate the optimal WGFT problem in the dual graph as an SD problem as follows:

$$\begin{aligned} \min_{\mathbf{W}} \rho \sum_{e \in \mathcal{V}^d} [W_e (x_{v_1(e)} - x_{v_2(e)})^2 + \gamma (1 - W_e)] \\ + \sum_{(e,s) \in \mathcal{E}^d} |W_e - W_s| \\ \text{s.t. } W_e \in \{1, c\} \quad \forall e \in \mathcal{V}^d. \end{aligned} \quad (22)$$

The problem is an SD problem, because the first and second term can be interpreted as a deviation cost, while the third term can be interpreted as a separation cost. We thus employ the algorithm in [13] to efficiently solve the SD problem in (22). Since the defined deviation term is linear and the separation term is bi-linear in (22), it is solvable in strongly polynomial time<sup>5</sup> using the algorithm in [13] based on minimum graph cuts. Specifically, the running time of this algorithm for a  $4 \times 4$  block (adopted in our coding system) is on the order of  $10^{-2}$  seconds on an Intel Core(TM) i5-4570 CPU 3.20GHz machine (around 2.7 minutes for a  $512 \times 512$  image).

### B. Unweighted Graph Fourier Transform

Unlike WGFT, in the UGFT case we do not have closed form expressions for  $R_\alpha(\mathbf{x}, \mathbf{W})$  and  $R_T(\mathbf{W})$  in terms of  $\mathbf{W}$ . The reason is that it is difficult to approximate the first term in (5)—the rate of UGFT coefficients—using a simple proxy as done in WGFT. As discussed earlier, the rate proxy

in (19) captures only the cost of AC coefficients, ignoring the cost of DC coefficients, which could be *variable* for UGFTs corresponding to graphs with several disconnected components. Due to this combinatorial nature, we develop a greedy algorithm based on spectral clustering for the UGFT search sub-problem, which computes the *actual rate* via a divide-and-conquer strategy. We first formulate the UGFT search problem, and then elaborate on the proposed greedy algorithm.

1) *Problem Reformulation*: Since UGFTs are defined on disconnected and unweighted graphs with the feasible weight set  $\mathcal{C} = \{1, 0\}$ , we modify the constraint in (5) for the UGFT search problem as follows:

$$\begin{aligned} \min_{\mathbf{W}} R_\alpha(\mathbf{x}, \mathbf{W}) + R_T(\mathbf{W}) \\ \text{s.t. } W_{i,j} \in \{1, 0\} \quad \forall i, j \in \mathcal{V}. \end{aligned} \quad (23)$$

2) *A Greedy Algorithm for UGFT*: Instead of exhaustive search, we develop a greedy algorithm combined with spectral clustering in order to efficiently solve (23). Spectral clustering [35], identification of clusters of similar data, takes advantage of the eigenvectors of the graph Laplacian derived from a similarity graph of the data. Among the family of spectral clustering algorithms, normalized cuts [14] is a very popular one. We hence employ normalized cuts to identify *clusters of similar pixels* for UGFT.

For a given pixel block  $\mathbf{x}$ , the normalized cuts algorithm consists of four steps: i) construct a similarity graph  $\mathcal{G}^s = \{\mathcal{V}^s, \mathcal{E}^s, \mathbf{W}^s\}$  on  $\mathbf{x}$ , which measures the similarity between pixels  $i$  and  $j$  using  $W_{i,j}^s$ ; ii) acquire the generalized eigenvectors  $\psi$  of the graph Laplacian by solving  $(\mathbf{D}^s - \mathbf{W}^s)\psi = \lambda \mathbf{D}^s \psi$ ; iii) bipartition  $\mathbf{x}$  using the eigenvector with the smallest non-zero eigenvalue; iv) recursively repartition the segmented clusters if necessary.

Leveraging on normalized cuts, we develop a greedy algorithm to search for a locally optimal UGFT. The key idea is to recursively partition the pixel block using normalized cuts until the resulting representation cost increases. For each iteration, we perform the following three steps on each pixel block  $\mathbf{x}$ .

First, we construct a similarity graph  $\mathcal{G}^s = \{\mathcal{V}^s, \mathcal{E}^s, \mathbf{W}^s\}$  on  $\mathbf{x}$ . Each weight  $W_{i,j}^s$  is defined as in [35]:

$$W_{i,j}^s = \exp \left\{ \frac{-|x_i - x_j|^2}{\sigma_w^2} \right\}, \quad (24)$$

which describes the similarity in pixel intensities.<sup>6</sup>  $\sigma_w$  controls the sensitivity of the similarity measure to the range of intensity differences. From this graph, we use normalized cuts to partition the block into two clusters.

Second, we build a graph  $\mathcal{G}^u$  where pixels in the same cluster are connected with weight 1 while those in different clusters are disconnected. The UGFT is then derived from  $\mathcal{G}^u$ .

Thirdly, we compute the total representation cost as defined in the objective of (23). We encode transform coefficients via entropy coding to compute the coding rate, and encode

<sup>5</sup>“Strongly polynomial time” means that the computation complexity of an algorithm in question is polynomial in the size of the input bits that describe the instance of the optimization problem. See [39] for details.

<sup>6</sup>Note that the similarity graph  $\mathcal{G}^s$  used for normalized cuts is a different graph than the ones used to define GFTs, and thus the edge weights are not restricted to be in a small discrete set  $\mathcal{C}$ .

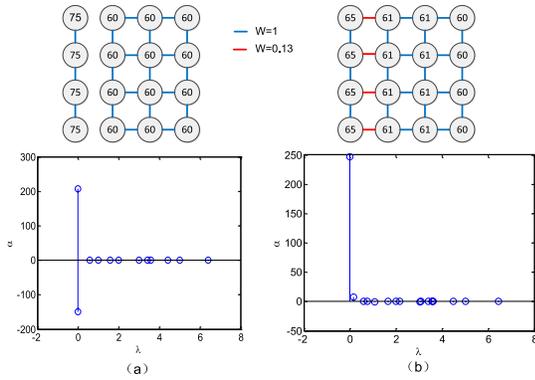


Fig. 7. An illustration of UGFT and WGFT. (a) An example block containing pixel pairs with strong and zero correlations, and its signal representation in the UGFT domain. (b) An example block containing pixel pairs with strong and weak correlations, and its signal representation in the WGFT domain.

disconnected edges in the graph via AEC to compute the transform description cost. Note that, instead of using a proxy as in the search for WGFT, we compute the actual cost of transform description, since it is more accurate while only inducing moderate complexity to the greedy algorithm. If the current representation cost is smaller than that of the previous iteration, then further repartition is performed.

In practice, few iterations are required due to the small size of the coding block, and hence this greedy algorithm is computation-efficient.

### C. Illustrative Examples: Optimal GFTs for Different Blocks & at Different Rates

Having elaborated on the problem formulations and algorithms of WGFT and UGFT, we provide examples to illustrate optimal GFTs for different classes of blocks and at different rates.

We first provide an example in Fig. 7 to show how we employ the two flavors of GFTs to capture various correlations in pixel blocks. Fig. 7(a) shows an example block containing pixel pairs with strong and zero correlations. UGFT is chosen during mode decision based on RD costs, where the block is divided into two separate partitions and a connected graph is constructed in each partition. The resulting transform representation is one DC coefficient for the description of each partition, as shown at the bottom of Fig. 7(a). Fig. 7(b) shows an example block containing pixel pairs with strong and weak correlations. Accordingly WGFT is chosen from mode decision, where a graph containing edge weights  $c$  (here  $c = 0.13$ ) is constructed on the block. The resulting transform coefficients consist of one large DC term and one small AC term, shown at the bottom of Fig. 7(b).

Next, we provide an illustrative example of optimal UGFTs for the same pixel block but at different target bit rates. Given a pixel block with three smooth regions as shown in Fig. 8, the minimal representation cost at high bit rates is achieved when it is divided into three connected components corresponding to the three smooth regions. This is because this derived UGFT results in only three DC coefficients, leading to the minimal and significantly smaller cost of quantized

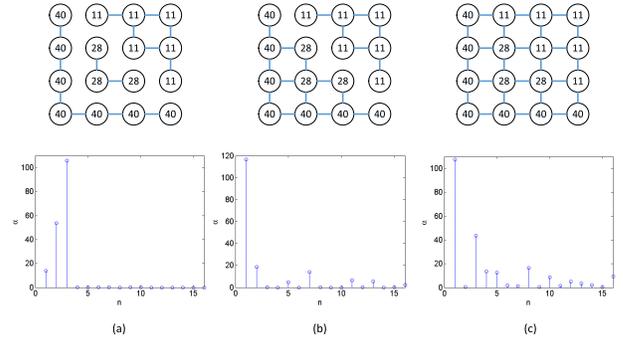


Fig. 8. An illustration of optimal UGFTs at different rates. For a given pixel block, the figure shows the optimal graph construction (all the edges connecting pixels are assigned weight 1) and resulting transform coefficients at (a) high bit rate, (b) medium bit rate and (c) low bit rate.

transform coefficients (compared to other UGFTs resulting in many high frequency coefficients) at high bit rates, which together with the description cost results in the minimal representation cost. At medium rates, the larger quantization parameter (QP) quantizes more coefficients to zero, and a simpler UGFT with similar quantized transform coefficient cost but smaller transform description cost would be more attractive, resulting in the graph in Fig. 8(b). Finally, at low rates, the large QP quantizes most coefficients to zero, and the simplest UGFT is best, as shown in Fig. 8(c).

## VII. FAST IMPLEMENTATION OF GRAPH FOURIER TRANSFORMS

In this section we propose a fast implementation of GFT for practical deployment. As stated earlier, the online eigen-decomposition for the construction of GFT is a hurdle to real-time implementation. We avoid this by pre-computing and storing most popular GFTs in a table for simple lookups. Graph isomorphism is further exploited to optimize the table. We first detail the construction of GFT lookup table, and then compare the complexity of table lookup against that of online eigen-decomposition.

The table size could be very large if we simply store all the used GFTs. A large table would lead to high requirement in storage *and* expensive lookup. Hence, we propose to construct a GFT lookup table of relatively small size. Specifically, we have three strategies:

- 1) We perform GFT on a LR block ( $4 \times 4$ ), which admits a smaller space of GFT variants to begin with.
- 2) Only the most frequently used LR-GFTs are stored.
- 3) Exploiting graph isomorphism, only one LR-GFT is stored for all isomorphic graphs.

Due to self-similarity in images, the same or similar structures are likely to recur throughout. Hence, the underlying LR-GFTs with respect to those structures are frequently used. We thus store only the most popular LR-GFTs in our lookup table, while covering a large fraction of the total used LR-GFTs.

We store one GFT for all isomorphic graphs. Two graphs  $\mathcal{G}$  and  $\mathcal{H}$  are isomorphic if there exists such a function mapping

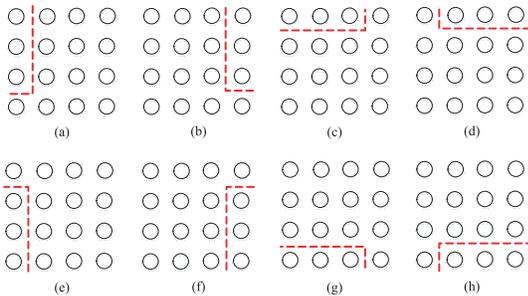


Fig. 9. An illustration of the 8 underlying isomorphic graphs with respect to the same graph structure, i.e., the same GFT after appropriate permutation of vertex labels.

vertices of  $\mathcal{G}$  to vertices of  $\mathcal{H}$ :

$$f : \mathcal{V}(\mathcal{G}) \rightarrow \mathcal{V}(\mathcal{H}), \quad (25)$$

that any two vertices  $i$  and  $j$  of  $\mathcal{G}$  are adjacent in  $\mathcal{G}$  if and only if  $f(i)$  and  $f(j)$  are adjacent in  $\mathcal{H}$ . Intuitively, the structures of isomorphic graphs are the same.

For 4-connected graphs constructed on a pixel block, we can construct 8 isomorphic graphs with respect to the same graph structure, thus resulting in the same GFT after appropriate permutation of vertex labels. See Fig. 9 for an illustration. The corresponding mapping functions  $f$  include *reflection* and *rotation* of graph vertices. Specifically, a graph can be mapped to four of its isomorphic graphs by reflection with respect to the central horizontal, vertical, diagonal and anti-diagonal axes respectively, and three of its isomorphic graphs by different degrees of rotation. Hence, we can reduce the table size by up to a factor of 8 via graph isomorphism. During the encoding and decoding, the actual GFT, if not directly stored, is retrieved by mapping from the stored GFT.

With all the above factors considered, the lookup table can be reduced to a manageable size. Experimentally, we collect LR-GFTs from ten PWS images with diverse structures to build the lookup table. Note that, the training images are different from the testing images in the final experimentation. Statistical results show that 126 most popular LR-GFTs out of a total 3749 LR-GFTs cover 88% of actual computed optimal LR-GFTs. We thus set the table size  $L$  to 126 to store the most popular LR-GFTs. Further, we encode the table indices by Huffman Coding [40] based on the used frequency estimated from test images, which is known at both the encoder and decoder.

Having constructed the lookup table, during encoding we search for the GFT (including UGFT and WGFT) for a given block by finding out the one that gives the best RD performance. Different from previous optimization (5), we consider the resulting distortion for a given GFT during table lookup. Recall that the expected distortion remains the same in a *statistical* sense when we design GFTs for particular classes of statistical signals. In contrast, during run-time when the designed GFTs are fixed in the table, we can consider the distortion of each block *deterministically*. We then transmit the table index losslessly to indicate which LR-GFT is employed for the given block, so that the decoder is able to identify the correct inverse transform.

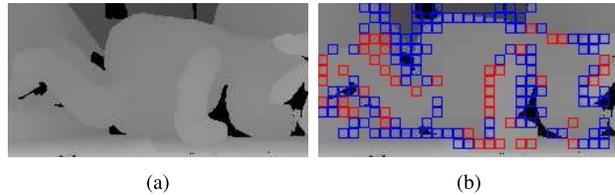


Fig. 10. The selection of transform modes for a portion of *Teddy*. Blocks in red choose WGFT and blocks in blue choose UGFT. (a) The original. (b) Transform modes.

We now compare the complexity of using eigen-decomposition for the derivation of GFT and that of table lookup. The computational complexity of eigen-decomposition at both the encoder and decoder for a  $\sqrt{N} \times \sqrt{N}$  ( $N = 16$  in our case) coding block is  $O(N^3)$ . In contrast, the complexity of table lookup is  $O(L)$  ( $L = 126$  in our setting) at the encoder and  $O(1)$  at the decoder. Hence, table lookup leads to significant reduction in complexity compared against on-line eigen-decomposition at both the encoder and decoder.

## VIII. EXPERIMENTAL RESULTS

### A. Experimental Setup

To evaluate our proposed MR-GFT coding system for PWS signals, we use four test image sequences:  $448 \times 368$  depth maps of *Teddy* and *Cones*,<sup>7</sup> and graphics images of *Dude* ( $800 \times 480$ ) and *Tsukuba*<sup>8</sup> ( $640 \times 480$ ).

For the three transform candidates in the proposed coding system, the block size of HR-DCT is  $8 \times 8$ , and that of LR-WGFT and LR-UGFT is  $4 \times 4$ . The weighting parameter  $\gamma$  in (22) is empirically assigned 3 in all the experiments. We note that our experimental results are not very sensitive to the specific choice of  $\gamma$ . For the calculation of the fractional edge weight  $c$ , we collect pixel pairs of weak correlation from the training images, and compute  $c$  to be 0.13 via (17).

We compare coding performance of our proposed scheme against four compression schemes: H.264 intra (HR-DCT) [2], the GFT coding (HR-UGFT) in [24], the shape-adaptive wavelet (SAW) in [21], and our previous work MR-UGFT in [34]. Note that intra prediction is used for all schemes.

### B. Selection of Transform Modes

We first investigate the selection of transform modes for test images. Fig. 10 shows an example of the mode decision in *Teddy*. It can be observed that blocks containing pixel pairs with weak correlation (*weak boundaries*) in red, generally choose LR-WGFT, while blocks containing pixel pairs with zero correlation (*strong boundaries*) in blue, generally choose LR-UGFT. This verifies our design of WGFT and UGFT for the representation of blocks containing pixels pairs with different correlations. Note that some blocks with strong/weak boundaries do not choose LR-WGFT/LR-UGFT, e.g., the blocks containing black holes. This is because those blocks lose much energy during the low-pass filtering if LR-WGFT

<sup>7</sup>Available at <http://vision.middlebury.edu/stereo/data/scenes2003/>

<sup>8</sup>Available at <http://www.cvlab.cs.tsukuba.ac.jp/dataset/tsukubastereo.php>

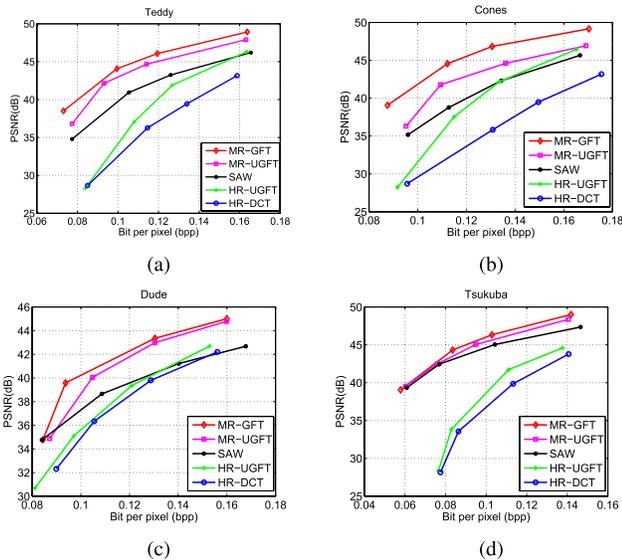


Fig. 11. RD performance comparison among different compression schemes for PWS images.

or LR-UGFT is selected, which greatly degrades the reconstruction quality. In these cases, HR-DCT will be chosen for better preservation of details.

Further, we observe that GFTs corresponding to contiguous boundaries are used more frequently, which verifies our assumption for the proxy of transform description in (21).

### C. Objective Comparison in RD Performance

We now compare the performance of the proposed MR-GFT scheme against SAW, HR-UGFT, HR-DCT and MR-UGFT for PWS images. Fig. 11 presents the RD performance of these schemes for four test images with a typical PSNR range. The proposed MR-GFT achieves significant gain in PSNR over a wide range of bit rate. On average we have 6.8 dB gain over HR-DCT, 5.9 dB gain over HR-UGFT, 2.5 dB gain over SAW, and 1.2 dB gain over MR-UGFT.

The gain comes from three improvements: 1) the added edge weight  $c$ , which leads to sparser GFT-domain representation via proper characterization of weak correlation; 2) minimization of the total representation cost, which results in minimal rate; and 3) the MR scheme, which down-samples each HR block to LR and thus reducing coding bits.

### D. Subjective Comparison

Fig. 12 demonstrates images reconstructed from different schemes for Teddy, Cones and Dude. We observe that our MR-GFT produces the cleanest images with the sharpest boundaries among all methods, which validates the effectiveness of the edge-preserving MR-GFT approach. In the images reconstructed by HR-DCT, boundaries are severely corrupted since the assumption of the near unity inter-pixel correlation in the DCT is not valid along strong boundaries in PWS images. Though GFT is employed in HR-UGFT, some boundaries in the restored images by HR-UGFT are still corrupted. This is because the DCT is chosen in those regions as GFT consumes

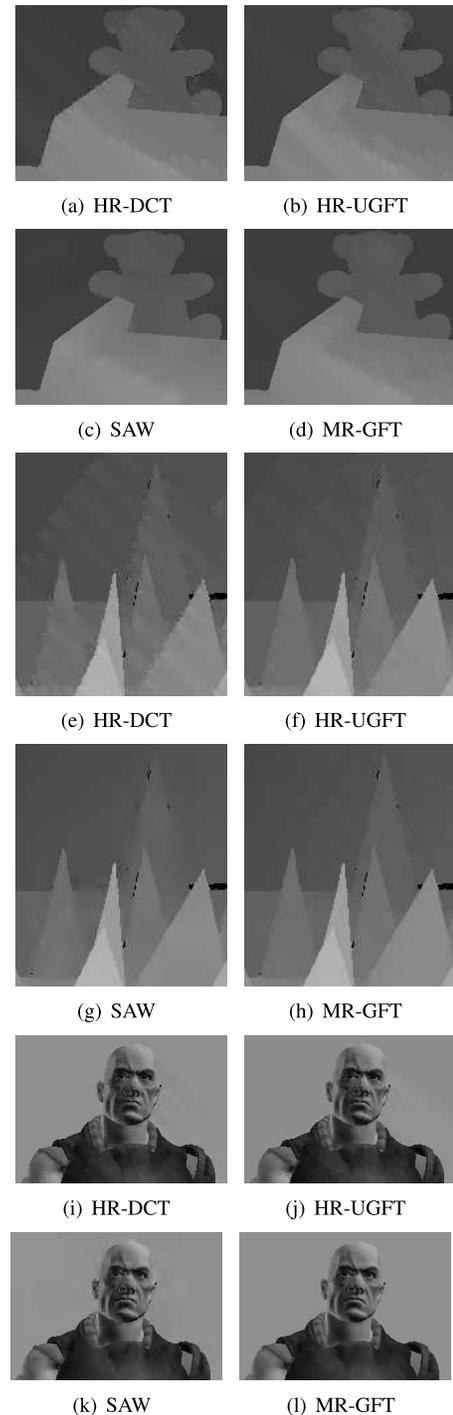


Fig. 12. The subjective quality comparison among different compression schemes. (a)~(d) Teddy at 0.10 bpp; (e)~(h) Cones at 0.13 bpp; (i)~(l) Dude at 0.13 bpp.

more bits due to the boundary coding. Images compressed by SAW remain sharp along strong boundaries. However, weak boundaries are often blurred. In contrast, MR-GFT preserves both strong and weak boundaries, since we properly model weak correlation using the fractional weight  $c$ .

### E. Application to Depth-Image-Based Rendering

We conduct further experiments on depth maps, which is an important class of PWS images. Instead of being

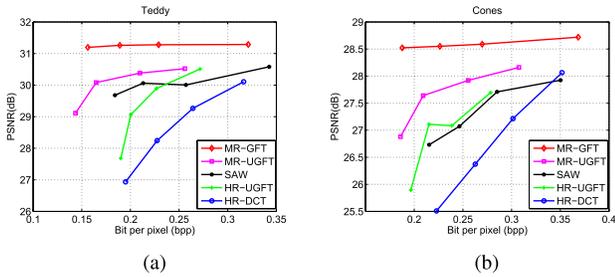


Fig. 13. RD performance comparison among different compression schemes for depth maps tailored for DIBR.

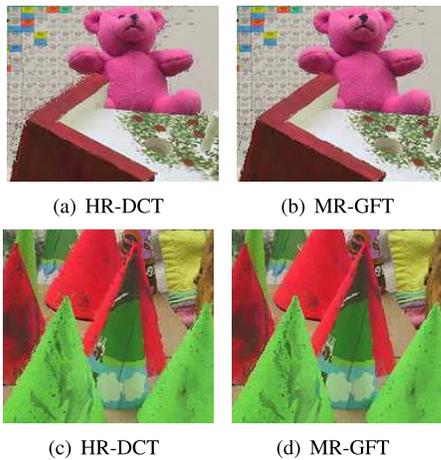


Fig. 14. Subjective quality comparison of DIBR-synthesized images among different compression schemes. (a) (b) *Teddy*; (c) (d) *Cones*.

observed directly, depth maps generally facilitate various end applications, such as virtual view synthesis via *depth-image-based rendering* (DIBR). Note that in this case, one can replace the depth map distortion metric in (4) with a synthesized view distortion metric. The synthesized view distortion still does not change using different transforms, because the position error in the synthesized view is a linear function of the distortion in the depth map under some assumptions [41]. Hence, the GFT training remains the same for depth map coding for DIBR. During run-time, one can replace the depth map distortion with a synthesized view distortion metric during table lookup.

We use a simple implementation of 3D warping [11] to perform DIBR. Fig. 13 presents the RD curves for *Teddy* and *Cones*, where the PSNR of synthesized views is evaluated at various total rates of stereo depth maps. On average we achieve 2.2dB gain over HR-DCT, 1.8dB gain over HR-UGFT, 1.2dB gain over SAW, and 1.0dB gain over MR-UGFT. Further, in Fig. 14 we show the virtual views of *Teddy* and *Cones* synthesized from stereo depth maps compressed using HR-DCT and MR-GFT at the same bit rate. MR-GFT is observed to produce more pleasant synthesized images, with fewer ringing artifacts and corrupted boundaries. The good performance is mostly due to the well-preserved depth map boundaries by MR-GFT, which plays a critical role in DIBR.

## IX. CONCLUSION

We propose a multi-resolution (MR) graph Fourier transform (GFT) coding scheme for compression of piecewise smooth (PWS) images. Unlike fixed transforms such as the DCT, the defined optimal GFT is adaptive to each local block by minimizing the total representation cost, considering both the sparsity of the signal's transform coefficients and the compactness of transform description. We develop efficient algorithms to search for optimal GFTs in a defined search space, based on graph optimization techniques such as spectral clustering and minimum graph cuts. Further, we introduce two techniques for practical implementation of GFT. One is the MR scheme where GFT is deployed over a low-pass filtered and down-sampled version of a high-resolution block. The other is the pre-computation of the most popular GFTs in a stored table for simple lookup instead of real-time eigen-decomposition. Experimental results show that the proposed scheme outperforms H.264 intra by 6.8dB in PSNR on average at the same bit rate. By extension, while we tailor our proposed approach for PWS image compression, it is possible to implement our proposal as a coding mode during compression of general images, so that when a code block is deemed PWS, our coding scheme can be deployed.

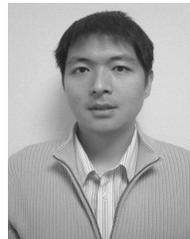
## REFERENCES

- [1] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 9–21, Sep. 2001.
- [2] T. Wiegand, G. J. 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.
- [3] J. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Trans. Commun. Syst.*, vol. 11, no. 3, pp. 289–296, Sep. 1963.
- [4] V. K. Goyal, J. Zhuang, and M. Vetterli, "Transform coding with backward adaptive updates," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1623–1633, Jul. 2000.
- [5] S. Jana and P. Moulin, "Optimality of KLT for high-rate transform coding of Gaussian vector-scale mixtures: Application to reconstruction, estimation, and classification," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4049–4067, Sep. 2006.
- [6] M. Effros, H. Feng, and K. Zeger, "Suboptimality of the Karhunen–Loeve transform for transform coding," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1605–1619, Aug. 2004.
- [7] A. K. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.
- [8] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, Jul. 2003.
- [9] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2010.
- [10] P. Wan, Y. Feng, G. Cheung, I. V. Bajic, O. C. Au, and Y. Ji, "3D motion in visual saliency modeling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 1831–1835.
- [11] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," *Proc. SPIE*, vol. 7443, p. 74430T, Feb. 2009.
- [12] S.-C. Han and C. I. Podilchuk, "Video compression with dense motion fields," *IEEE Trans. Image Process.*, vol. 10, no. 11, pp. 1605–1612, Nov. 2001.
- [13] D. S. Hochbaum, "An efficient and effective tool for image segmentation, total variations and regularization," in *Proc. 3rd Int. Conf. Scale Space Variational Methods Comput. Vis.*, vol. 6667, 2011, pp. 338–349.
- [14] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

- [15] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, "Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 343–359, Mar. 2005.
- [16] Y. Morvan, P. H. N. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," *Proc. SPIE*, vol. 6055, Jan. 2006, p. 60550K.
- [17] P. Merkle *et al.*, "The effects of multiview depth video compression on multiview rendering," *Signal Process., Image Commun.*, vol. 24, nos. 1–2, pp. 73–88, Jan. 2009.
- [18] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep. 2012, pp. 1541–1544.
- [19] 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.
- [20] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Wavelet-domain approximation and compression of piecewise smooth images," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1071–1087, May 2006.
- [21] M. Maitre and M. N. Do, "Depth and depth-color coding using shape-adaptive wavelets," *J. Vis. Commun. Image Represent.*, vol. 21, nos. 5–6, pp. 513–522, Jul./Aug. 2010.
- [22] E. J. Candes and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise  $C^2$  singularities," *Commun. Pure Appl. Math.*, vol. 57, no. 2, pp. 219–266, Feb. 2002.
- [23] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.
- [24] G. Shen, W.-S. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *Proc. IEEE Picture Coding Symp.*, Nagoya, Japan, Dec. 2010, pp. 566–569.
- [25] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process. (MMSP)*, Pula, Italy, Sep./Oct. 2013, pp. 001–006.
- [26] C. Zhang and D. Florencio, "Analyzing the optimality of predictive transform coding using graph-based models," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 106–109, Jan. 2013.
- [27] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," *Proc. SPIE*, vol. 7543, p. 75430B, Jan. 2010.
- [28] N.-M. Cheung, D. Tian, A. Vetro, and H. Sun, "On modeling the rendering error in 3D video," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep./Oct. 2012, pp. 3021–3024.
- [29] G. Cheung, A. Kubota, and A. Ortega, "Sparse representation of depth maps for efficient transform coding," in *Proc. IEEE Picture Coding Symp.*, Nagoya, Japan, Dec. 2010, pp. 298–301.
- [30] G. Cheung, J. Ishida, A. Kubota, and A. Ortega, "Transform domain sparsification of depth maps using iterative quadratic programming," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, Sep. 2011, pp. 129–132.
- [31] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," *Signal Process. Image Commun.*, vol. 24, nos. 1–2, pp. 89–100, Jan. 2009.
- [32] D. B. Graziosi, N. M. M. Rodrigues, S. M. M. de Faria, D. Tian, and A. Vetro, "Analysis of depth map resampling filters for depth-based 3d video coding," in *Proc. Int. Conf. Telecommun.*, Castelo Branco, Portugal, May 2013.
- [33] G. Shen, W.-S. Kim, A. Ortega, J. Lee, and H. Wey, "Edge-aware intra prediction for depth-map coding," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Hong Kong, Sep. 2010, pp. 3393–3396.
- [34] W. Hu, G. Cheung, X. Li, and O. Au, "Depth map compression using multi-resolution graph-based transform for depth-image-based rendering," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep./Oct. 2012, pp. 1297–1300.
- [35] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
- [36] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.
- [37] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.
- [38] W.-S. Kim, S. K. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 813–816.
- [39] K. S. C. Papadimitriou, *Combinatorial Optimization: Algorithm and Complexity*. New York, NY, USA: Dover, 1998, p. 165.
- [40] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.
- [41] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 721–724.



**Wei Hu** (SM'11) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2010. She is currently pursuing the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Hong Kong. Her research interests include graph signal processing, 2D/3D image and video representation, coding, and processing.



**Gene Cheung** (M'00–SM'07) received the B.S. degree in electrical engineering from Cornell University, Ithaca, NY, USA, in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 1998 and 2000, respectively.

He was a Senior Researcher with Hewlett-Packard Laboratories Japan, Tokyo, Japan, from 2000 to 2009. He is currently an Associate Professor with the National Institute of Informatics, Tokyo.

Prof. Cheung research interests include image and video representation, immersive visual communication, and graph signal processing. He has authored over 140 international conference and journal publications. He has served as an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA from 2007 to 2011, and serves as an Associate Editor of the Digital Signal Processing Applications Column in the *IEEE Signal Processing Magazine*, the *APSIPA Journal on Signal and Information Processing*, and the *SPIE Journal of Electronic Imaging*, and an Area Editor of the *EURASIP Signal Processing: Image Communication*. He serves as the lead Guest Editor of the Special Issue on Interactive Media Processing for Immersive Communication in the IEEE JOURNAL ON SPECIAL TOPICS ON SIGNAL PROCESSING, target published date in 2015. He served as a member of the Multimedia Signal Processing (MMSP) Technical Committee in the IEEE Signal Processing Society from 2012 to 2014. He served as the Area Chair of the IEEE International Conference on Image Processing (ICIP) in 2010 and from 2012 to 2013, the Technical Program Co-Chair of the International Packet Video Workshop in 2010, the Track Co-Chair of the Multimedia Signal Processing Track in the IEEE International Conference on Multimedia and Expo (ICME) in 2011, the Symposium Co-Chair of the Communications Software, Services, and Multimedia Applications Symposium in the IEEE Global Communications Conference in 2012, and an Area Chair of the ICME in 2013. He was invited as a Plenary Speaker of the IEEE MMSP in 2013 on the topic 3D Visual Communication: Media Representation, Transport and Rendering. He is a co-author of best student paper award in IEEE Workshop on Streaming and Media Communications 2011 (in conjunction with ICME 2011), best paper finalists in ICME 2011 and ICIP 2011, best paper runner-up award in ICME 2012, and best student paper award in ICIP 2013).



**Antonio Ortega** (S'91–M'95–SM'00–F'07) received the Telecommunications Engineering degree from the Universidad Politecnica de Madrid, Madrid, Spain, in 1989, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 1994, where he was supported by a Fulbright Scholarship.

He has been with the Department of Electrical Engineering Systems, University of Southern California (USC), Los Angeles, CA, USA, since 1994, where he is currently a Professor. He was the

Director of the Signal and Image Processing Institute and is the Associate Chair of Electrical Engineering Systems at USC. His work with USC has been funded by agencies, such as the National Science Foundation, the National Aeronautics and Space Administration, the Department of Energy, and a number of companies. Over 25 Ph.D. students have completed their thesis work under his supervision, and his work has led to over 250 publications in international conferences and journals. His research interests are multimedia compression, communications, and signal analysis. His recent work is focusing on distributed compression, multiview coding, error tolerant compression, wavelet-based signal analysis, and information representation in wireless sensor networks.

Dr. Ortega is a member of the Association for Computing Machinery. He has been the Chair of the Image and Multidimensional Signal Processing Technical Committee and a member of the Board of Governors of the IEEE Signal Processing Society in 2002. He was the Technical Program Co-chair of the IEEE International Conference on Image Processing in 2008, the IEEE International Workshop on Multimedia Signal Processing in 1998, and the IEEE International Conference on Multimedia and Expo in 2002. He has been an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and the *EURASIP Journal on Advances in Signal Processing*. He received the National Science Foundation CAREER Award, the IEEE Communications Society Leonard G. Abraham Prize Paper Award in 1997, the IEEE Signal Processing Society Magazine Award in 1999, and the *EURASIP Journal of Advances in Signal Processing* Best Paper Award in 2006.



**Oscar C. Au** (F'12) received the B.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1986, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1988 and 1991, respectively. After being a Post-Doctoral Fellow in Princeton for one year, he joined the Hong Kong University of Science and Technology (HKUST), Hong Kong, as an Assistant Professor, in 1992. He is/was a Professor with the Department of Electronic and Computer Engineering, the Director of the Multimedia Technology Research

Center, and the Director of Computer Engineering, HKUST. He is currently a Core Member of the State Key Laboratory on Advanced Displays and Optoelectronic Technology.

His main research contributions are in video/image coding and processing, watermarking/light weight encryption, and speech/audio processing. His research topics include fast motion estimation for H.261/3/4/5, MPEG-1/2/4, AVS, optimal and fast suboptimal rate control, mode decision, transcoding, denoising, deinterlacing, post-processing, multiview coding, view interpolation, depth estimation, 3DTV, scalable video coding, distributed video coding, subpixel rendering, JPEG/JPEG2000, HDR imaging, compressive sensing, halftone image data hiding, GPU-processing, and software–hardware co-design. He has authored 70 technical journal papers, over 370 conference papers, three book chapters, and over 70 contributions to international standards. His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. His lightweight encryption and error resilience algorithms are accepted into the China AVS standard. He was the Chair of the Screen Content Coding Ad Hoc Group in JCTVC for HEVC. He holds over 25 granted U.S. patents and is applying for over 70 on his signal processing techniques. He has performed forensic investigation and stood as an Expert Witness in Hong Kong courts many times.

Dr. Au is a fellow of the Hong Kong Institution of Engineers. He is/was an Associate Editor and a Senior Editor of several IEEE journals (the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, *Statistics and Probability Letters*, the *Journal of Selected Topics in Signal Processing*, and the IEEE Signal Processing Society Online Video Library) and non-IEEE journals (the *Journal of Visual Communication and Image Representation*, the *Journal of Signal Processing Systems*, *APSIPA Transactions on Signal and Image Processing*, the *Journal of Micromechanics and Microengineering*, the *Journal of Financial Intermediation*, and *Small Wars Journal*). He is a Guest Editor of some special issues in the *Journal of Selected Topics in Signal Processing* and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is/was a BoG Member and the Vice President Technical Activity of the Asia-Pacific Signal and Information Processing Association. He is/was the Chair of three technical committees (TCs), such as the IEEE CAS MSA TC, the IEEE SPS MMSP TC, and APSIPA IVM TC. He is a member of five other TCs, such as the IEEE CAS VSPC TC, DSP TC, the IEEE SPS IVMSM TC, IFS TC, and the IEEE ComSoc MMC TC. He served on two steering committees, such as the IEEE TMM and the IEEE ICME. He also served on organizing committee of many conferences, including ISCAS 1997, ICASSP 2003, ISO/IEC 71st MPEG 2005, and ICIP 2010. He was/will be the General Chair of several conferences, such as PCM 2007, ICME 2010, PV 2010, MMSP 2015, APSIPA ASC 2015, and ICME 2017. He received five best paper awards, such as SIPS 2007, PCM 2007, MMSP 2012, ICIP 2013, and MMSP 2013. He was the IEEE Distinguished Lecturer (DL) in 2009 and 2010 and APSIPA DL in 2013 and 2014, and has been a Keynote Speaker multiple times.