POLITECNICO DI TORINO
Repository ISTITUZIONALE

Graph Spectral Image Processing

(Article begins on next page)

28 April 2024

Graph Spectral Image Processing

Gene Cheung, *Senior Member, IEEE,* Enrico Magli, *Fellow, IEEE,* Yuichi Tanaka, *Member, IEEE,*
and Michael Ng, *Senior Member, IEEE*

Abstract—**Recent advent of graph signal processing (GSP) has spurred intensive studies of signals that live naturally on irregular data kernels described by graphs (*e.g.*, social networks, wireless sensor networks). Though a digital image contains pixels that reside on a regularly sampled 2D grid, if one can design an appropriate underlying graph connecting pixels with weights that reflect the image structure, then one can interpret the image (or image patch) as a signal on a graph, and apply GSP tools for processing and analysis of the signal in graph spectral domain. In this article, we overview recent graph spectral techniques in GSP specifically for image / video processing. The topics covered include image compression, image restoration, image filtering and image segmentation.**

Index Terms—**Image processing, graph signal processing**

I. INTRODUCTION

Graph signal processing (GSP) is the studies of signals that live on irregularly structured data kernels described by graphs [1], such as social networks and wireless sensor networks. The underlying graph typically reveals signal structures; an edge (i, j) with large weight $w_{i,j}$ connecting nodes i and j means that the signal samples at i and j are expected to be similar or correlated. Though a digital image contains pixels that reside on a regularly sampled 2D grid, one can nonetheless interpret an image (or an image patch) as a signal on a graph, with edges that connect each pixel to its neighborhood of pixels. By choosing an appropriate graph that reflects the intrinsic image structure, a spectrum of graph frequencies can be defined through eigen-decomposition of the graph Laplacian matrix [2], and notions like transforms [3]–[8], wavelets [9]–[11], smoothness [12]–[16] etc can be correspondingly derived. Then a target image (or image patch) can be decomposed and analyzed spectrally on the chosen graph using developed GSP tools—analogous to frequency decomposition of square pixel blocks via known transforms like discrete cosine transform (DCT). Recently, this graph spectral interpretation of traditional 2D images has led to new insights and understanding, resulting in optimization of both the underlying graph and the graph-based processing tools that shows demonstrable gain in a number of traditional image

processing areas, including image compression, restoration, filtering and segmentation[1].

For image compression, a Fourier-like transform for graph-signals called *graph Fourier transform* (GFT) [1] and many variants [5]–[8], [18], [19] have been used as adaptive transforms for coding of piecewise smooth (PWS) and natural images. Because the underlying graph used to define GFT can be different for each code block, the cost of describing the graph as well as the cost of coding GFT coefficients to represent the signal must both be taken into consideration. For wavelets on graphs [9]–[11], where the conventional notion of "downsampling by 2" is ill-defined for irregular data kernels, how to define critically sampled perfect reconstruction filterbanks (with (bi)orthogonal conditions) using appropriate downsamplers has been a challenge. We review proposals in designs of graph transforms and wavelets for image / video compression in Section III.

For image restoration such as denoising and deblurring, how to design appropriate signal priors to regularize otherwise ill-posed problems is a major challenge. Notions of sparsity [20] and signal smoothness [13], [16], [21], [22] can also be generalized to the graph-signal domain. Wiener filtering for graph-signals, which first requires a proper definition of wide sense stationarity for irregular graph data kernels, was recently developed [23]. We review popular graph-based restoration techniques in Section IV.

Spectral filtering is a fundamental image processing operation. It turns out that the well-known bilateral filter for image denoising [24] can be interpreted as a linear low-pass filter for a specific graph [25]. Other diffusion and edge-preserving smoothing operators are also discussed in Section V. Popular applications such as image retargeting and non-photorealistic rendering of images are also overviewed. Finally, fast implementation of graph filters using Chebyshev polynomial approximation is discussed.

Image segmentation is an old computer vision problem, and there is a long history of graph-based approaches such as graph cuts [26], [27]. More recent models such as the Mumford-Shah model [28] and graph biLaplacian [29] are discussed in Section VI.

II. PRELIMINARIES

A. Graph Definition

We first introduce common definitions and concepts in GSP for use in later sections. A graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ contains a

[1]We note that while graph has been used extensively as an abstraction for image processing in the past [17], we focus in this article in particular recent developed techniques that process or analyze image signals in appropriately chosen graph spectral domains.

set $\mathcal{V}$ of N nodes and a set $\mathcal{E}$ of M edges. Each existing edge $(i,j) \in \mathcal{E}$ is undirected and contains an edge weight $w_{i,j}$, which is typically positive; a large positive $w_{i,j}$ would mean that samples at nodes i and j are expected to be similar / correlated. Common for images, weight $w_{i,j}$ of an edge connecting nodes (pixels) i and j is computed using a Gaussian kernel, as done in the bilateral filter [24]:

$$w_{i,j} = \exp\left(-\frac{\|\mathbf{l}_i - \mathbf{l}_j\|_2^2}{\sigma_l^2}\right) \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma_x^2}\right) \quad (1)$$

where $\mathbf{l}_i$ is the location of pixel i on the 2D image grid, x_i is the intensity of pixel i, and σ_l^2 and σ_x^2 are two parameters. Hence $0 \le w_{i,j} \le 1$. Larger geometric and/or photometric distances between pixels i and j would mean a smaller weight $w_{i,j}$. Edge weights can alternatively be defined based on local pixel patches, features, etc [30]. To a large extent, the appropriate definition of edge weight (inter-node similarity) is application-dependent; we will introduce various definitions for different applications in the sequel.

More generally, a suitable graph can be constructed from a machine learning perspective—given multiple signal observations, identify a graph structure that best fits the observed data given a fitting criterion or model assumptions [31]–[34]. For example, *graphical lasso* in [31] computes a sparse inverse covariance matrix (precision matrix) assuming a Gaussian Markov Random Field (GMRF) model and a sparse graph. Graph learning is a fundamental problem in GSP and is discussed extensively in another article in this special issue.

A *graph-signal* $\mathbf{x}$ on $\mathcal{G}$ is a discrete signal of dimension N—one sample $x_i \in \mathbb{R}$ for each node[2] i in $\mathcal{V}$. Assuming nodes are appropriately labeled from 1 to N, we can treat a graph-signal simply as a vector $\mathbf{x} \in \mathbb{R}^N$.

B. Graph Spectrum

Given an edge weight matrix $\mathbf{W}$ where $W_{i,j} = w_{i,j}$, we define a diagonal *degree matrix* $\mathbf{D}$, where $d_{i,i} = \sum_j W_{i,j}$. A *combinatorial graph Laplacian matrix* $\mathbf{L}$ is $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [1]. Because $\mathbf{L}$ is symmetric, one can show via the Spectral Theorem that it can be eigen-decomposed into:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (2)$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing real eigenvalues λ_k along the diagonal, and $\mathbf{U}$ is an eigen-matrix composed of orthogonal eigenvectors $\mathbf{u}_i$ as columns. If all edge weights $w_{i,j}$ are restricted to be positive, then graph Laplacian $\mathbf{L}$ can be proved to be *positive semi-definite* (PSD) [2][3], meaning that $\lambda_k \ge 0, \forall k$ and $\mathbf{x}^T\mathbf{L}\mathbf{x} \ge 0, \forall \mathbf{x}$. Non-negative eigenvalues λ_k can be interpreted as *graph frequencies*, and eigenvectors $\mathbf{U}$ interpreted as corresponding graph frequency components. Together they define the *graph spectrum* for graph $\mathcal{G}$.

The set of eigenvectors $\mathbf{U}$ for $\mathbf{L}$ collectively form the *graph Fourier Transform* (GFT) [1], which can be used to decompose a graph-signal $\mathbf{x}$ into its frequency components via $\boldsymbol{\alpha} = \mathbf{U}^T\mathbf{x}$, similar to known discrete transforms such as DCT. In fact, one can interpret GFT as a generalization of known transforms like DCT; see Shuman et al. [1] for details.

Note that if the multiplicity m_k of an eigenvalue λ_k is larger than 1, then the set of eigenvectors that span the corresponding eigen-subspace of dimension m_k is non-unique. In this case it is necessary to specify the graph spectrum as the collection of eigenvectors $\mathbf{U}$ themselves. See more discussion on this issue in the compression context in Section III.

If we consider also negative edge weights $w_{i,j}$ that reflect inter-pixel dissimilarity / anti-correlation, then graph Laplacian $\mathbf{L}$ can be indefinite. We discuss a few recent works that employ negative edges in Section IV.

C. Variation Operators

Closely related to the combinatorial graph Laplacian $\mathbf{L}$ are other variants of Laplacian operators, each with its own unique spectral properties. A *normalized graph Laplacian* $\mathbf{L}_n = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ is a symmetric normalized variant of $\mathbf{L}$. In contrast, a *random walk graph Laplacian* $\mathbf{L}_r = \mathbf{D}^{-1}\mathbf{L}$ is an asymmetric normalized variant of $\mathbf{L}$. A *generalized graph Laplacian* $\mathbf{L}_g = \mathbf{L} + \text{diag}(d_{i,i})$ is a graph Laplacian with self-loops $d_{i,i}$ at nodes i—called *loopy graph Laplacian* in [35]—resulting in a general symmetric matrix with non-positive off-diagonal entries [36]. Eigen-decomposition can also be performed on these operators to acquire a set of graph frequencies and frequency components. For example, normalized variants $\mathbf{L}_n$ and $\mathbf{L}_r$ share the same eigenvalues between 0 and 2. While $\mathbf{L}$ and $\mathbf{L}_n$ are both symmetric, $\mathbf{L}_n$ does not have the constant vector as an eigenvector. Asymmetric $\mathbf{L}_r$ can be symmetrized via left and right diagonal matrix multiplications [37]. We will discuss different choices of variation operators in the sequel for different applications[4].

D. Graph-Signal Priors

Traditionally, for graph $\mathcal{G}$ with positive edge weights, signal $\mathbf{x}$ is considered *smooth* if each sample x_i on node i is similar to samples x_j on neighboring nodes j with large $w_{i,j}$. In the graph frequency domain, it means that $\mathbf{x}$ contains mostly low graph frequency components; *i.e.*, coefficients $\boldsymbol{\alpha} = \mathbf{U}^T\mathbf{x}$ are zeros for high frequencies. The smoothest signal is the constant vector—the first eigenvector $\mathbf{u}_1$ for $\mathbf{L}$ corresponding to the smallest eigenvalue $\lambda_1 = 0$.

---

[2] If a graph node represents a pixel in an image, each pixel would typically have three color components. For simplicity, one can treat each color component separately as a different graph-signal.

[3] One can prove that a graph $\mathcal{G}$ with positive edge weights has PSD graph Laplacian $\mathbf{L}$ via the Gershgorin circle theorem: each Gershgorin disc corresponding to a row in $\mathbf{L}$ is located in the non-negative half-space, and since all eigenvalues reside inside the union of all discs, they are non-negative.

[4] In [38], a general denoising regularization term is proposed where the penalty is proportional to the inner product between the signal $\mathbf{x}$ and its denoised residual $\mathbf{x} - f(\mathbf{x})$; $\mathbf{x}^T\mathbf{L}\mathbf{x}$ being an example if $\mathbf{I} - \mathbf{L}$ is interpreted as a denoising operator. The main goal of [38] is to show this regularization term can be used as an engine for more general inverse problems, similar to *plug-and-play priors* (P^3) [39]. In contrast, our goal here is to show different graph variation operators have different characteristics that are suitable for different applications.

Mathematically, we can write that a signal $\mathbf{x}$ is smooth if its *graph Laplacian regularizer* $\mathbf{x}^T \mathbf{L} \mathbf{x}$ is small [14]–[16]. Graph Laplacian regularizer can be expressed as:

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{(i,j) \in \mathcal{E}} w_{i,j} (x_i - x_j)^2 = \sum_k \lambda_k \, \alpha_k^2 \qquad (3)$$

Because $\mathbf{L}$ is PSD, $\mathbf{x}^T \mathbf{L} \mathbf{x}$ is lower-bounded by 0, achieved when $\mathbf{x} = c\mathbf{u}_1$ for some scalar constant c.

In [12] the adjacency matrix $\mathbf{W}$ is interpreted as a shift operator, and thus graph-signal smoothness is defined instead as the difference between a smooth signal $\mathbf{x}$ and its shifted version $\mathbf{W}\mathbf{x}$. Specifically, graph total variation based on l_p-norm is:

$$\mathrm{TV}_{\mathbf{W}}(\mathbf{x}) = \left\| \mathbf{x} - \frac{1}{|\lambda_{\max}|} \mathbf{W} \mathbf{x} \right\|_p^p \qquad (4)$$

where p is a chosen integer. More specifically, a quadratic smoothness prior is defined in [13] (also in [38]):

$$S_2(\mathbf{x}) = \frac{1}{2} \| \mathbf{x} - \mathbf{W} \mathbf{x} \|_2^2 \qquad (5)$$

Besides smoothness, sparsity of graph-signals with respect to a trained graph dictionary can also be used as a prior [40]. Specifically, to effectively represent signals on different graph topologies, graph atoms are constructed as polynomials of the graph Laplacian. Preliminary results in [40] show its potential, but we will not discuss this further in the sequel.

III. GRAPH-BASED IMAGE COMPRESSION

Image compression refers to the process of encoding an image $\mathbf{x}$ onto a codeword $c(\mathbf{x})$, minimizing distortion in the reconstructed image $\widehat{\mathbf{x}}$ for a given target bit-rate R_T, i.e.

$$\min D(\mathbf{x}, \widehat{\mathbf{x}}) \ \text{subject to} \ R(c(\mathbf{x})) \le R_T, \qquad (6)$$

where $R(c(\mathbf{x}))$ is the average codeword length. Traditionally, lossy compression employs a 2D transform (denoted as $\mathbf{U}$) to produce a new image representation where image pixels are at least approximately uncorrelated. This process typically generates a vector of transform coefficients as $\boldsymbol{\alpha} = \mathbf{U}^{-1}\mathbf{x}$, such that only few coefficients of $\boldsymbol{\alpha}$ are significantly different from zero. This is critical to achieve good compression performance, and such coefficients can often be interpreted in terms of a frequency representation.

A. Adaptive transforms for compression

More in detail, as in Fig. 1, the first step consists of the linear transform generating coefficients $\boldsymbol{\alpha}$. Such coefficients are subsequently quantized, and the quantization indexes are losslessly coded using some data compression algorithm such as Huffman or arithmetic coding. There exist plenty of variations on this scheme, and the interested reader is referred to textbooks on the subject for details, e.g. [41]. For our discussion, it is important to note that, if the transform to be used is *not* known in advance at the encoder and decoder but it is computed adaptively at the encoder in order to optimize the compression process, then some ancillary information has to be communicated to the decoder, in order to reconstruct the
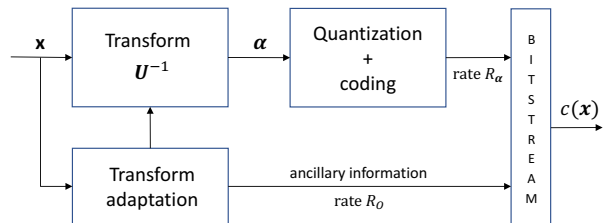

Fig. 1. Block diagram of lossy compression scheme. Coefficients are transformed, quantized and entropy coded. The transform is signal-adaptive, and the bitstream is composed of the coded transform coefficients (rate $R_{\boldsymbol{\alpha}}$) plus the ancillary information (rate R_O), for a total rate $R(c(\mathbf{x})) \le R_T$.

corresponding inverse transform to correctly decode the image. The rate term in (6) can be written as $R(c(\mathbf{x})) = R_{\boldsymbol{\alpha}} + R_O$, i.e. the rate needed to encode the transform coefficients plus the overhead rate due to the ancillary information; both terms may depend on $\mathbf{x}$, making the design of adaptive transforms a challenging problem.

In this section we focus on the transform stage, as spectral graph theory provides innovative tools to design transforms for image compression. Seeking an "optimal" transform has proved to be an elusive goal except for a few rather simple image models. The Karhunen-Loève transform (KLT) is based on the eigendecomposition of the covariance matrix of the input process and is very similar to the principal component analysis; it has been shown to be optimal for a Gaussian source under mean square error metric and fixed-rate coding [42]. The DCT [43] is asymptotically equivalent to the KLT for a first-order autoregressive process [44]. However, these models fail to capture the complex and nonstationary behavior typically occurring in digital images, and transform design is still an active research area. While many commonly used transforms, such as the DCT and wavelets [45], employ a fixed set of basis vectors that need not be communicated to the decoder, the KLT is a signal-adaptive transform. Adaptivity allows to match the basis vectors (the columns of $\mathbf{U}$) to a class of signals of interest, but the transform matrix has to be known at both the encoder and decoder; moreover, the resulting transform has no structure and hence lacks any fast algorithm. These issues have limited the practical use of the KLT for signal compression.

Like the KLT, the GFT is also based on an eigenvector decomposition. The GFT interprets a signal as being defined on a graph, and calculates the eigenvector decomposition of the corresponding graph Laplacian as in (2). Thus, while the KLT takes a statistical approach, describing correlations among image pixels through estimates of their linear correlation coefficients, the GFT employs a more flexible approach, in which pixel similarities are encoded into the weights of an undirected graph, where each node of the graph represents a pixel, and each edge weight represents the "similarity" of the two pixels at the ends of the edge. The two transforms are related to each other; in particular, [46] shows that the GFT approximates the KLT for a piece-wise first-order autoregressive process, while [5] shows that the GFT is optimal for decorrelation of an image following a Gauss-Markov random field (GMRF) model. Both
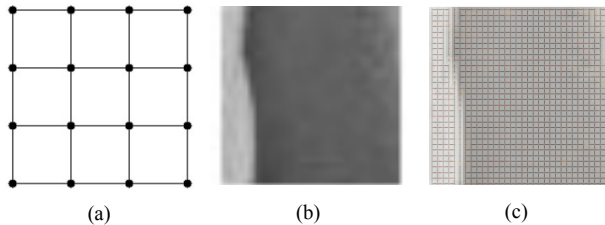
Fig. 2. (a) Square grid graph; (b) a 32x32 image block, and (c) an example of graph superimposed onto it. Graph edges with white color denote weak pixel similarity, and the figure shows that graph edges indeed encode structural information about the image.

the KLT and GFT can be interpreted in terms of kernels. In the KLT, the covariance matrix (which is PSD) is obtained from a (PSD) linear kernel, whereas the GFT is obtained from the graph Laplacian matrix, which is also PSD.

In practice, however, a graph can be computed for each individual image, making the GFT a more flexible framework for transform design. Roughly speaking, the graph in the GFT encodes image structures, as opposed to statistical correlations. This is useful because one can decide the degree of accuracy with which structures are represented in the graph, providing means to reduce the overhead of signaling the transform to the decoder. Referring to Fig. 1, using the GFT in a transform coding scheme requires communicating to the decoder a description of the graph as ancillary information; the relatively high overhead calls for descriptions of the graph that are optimized in a rate-distortion (RD) sense, i.e. they are sufficiently informative to yield effective transforms, without requiring a large overhead.

B. Graph Fourier Transform and graph design

As has been seen in Sec. II-B, the graph topology and set of weights $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ fully define the graph Laplacian matrix, from which the GFT is computed[5]. Hence, obtaining a "good" GFT amounts to selecting the topology and weights yielding the best compression performance in an RD sense as in (6). Regarding the topology, given that 2D images are typically defined on a square grid, a square grid graph is typically employed as in Fig. 2-a, where each pixel is connected to its four horizontal and vertical neighbors. In principle, one may decide to add graph edges corresponding to diagonal neighbors, or connecting pixels whose distance is larger than one. However, this may greatly increase the overhead of communicating the graph, unless edges are carefully selected e.g. as proposed in [48]. Fig. 2-b shows a 32x32 image block with the corresponding graph superimposed onto it, emphasizing the fact that the graph encodes image structures.

1) Choosing edge weights: The weight $w_{i,j}$ on each edge of the graph is conventionally computed as a function of the difference in pixel values x_i and x_j connected by that edge—i.e. the photometric distance—as computed in (1).

However, it is easy to realize that real-valued graph weights are too expensive in terms of overhead. In [3], [49], [50]

the weights are constrained to be in the set $\{1, 0\}$, implying that the graph only describes strong or zero correlation; the weights are chosen from detected image edges [18], using a greedy optimization algorithm [49], or from the output of an image segmentation algorithm, where an independent graph with weights equal to one is associated to each region and the resulting GFT plays the role of a shape-adaptive transform [50]. In [51] the difference $|x_i - x_j|$ is quantized to two values using a pdf-optimized uniform quantizer, yielding a graph that is always connected by construction; although weight binarization leads to suboptimal compression efficiency, it is shown that a suitably designed quantizer makes the performance loss very small. In [46] two sets of weight values are used, i.e. $w_{i,j} \in \{1, 0\}$ for image blocks characterized by strong or zero correlation, and $w_{i,j} \in \{1, c\}$ for blocks with strong or *weak* correlation. The constant c is optimized using a model suitable for piecewise smooth signals, and very good results are obtained in the compression of depth map images. The overhead incurred by the graph, however, makes it harder to obtain significant gains on natural images. This problem is addressed in [51], where edge prediction followed by coding is used to reduce the overhead, leading to performance gains between 1 and 3 dB in peak signal-to-noise ratio (PSNR) over the DCT. More sophisticated graph coding techniques may also be devised, e.g. one might in principle apply contour coding techniques as in [52], [53] to reduce the cost of representing the graph. Moreover, in [53] directional graph weight prediction modes are proposed, which avoid transmitting any overhead information to the decoder.

2) Graph learning: Defining a good graph from data observations is so important in many applications, and particularly in compression, that more structured methods have been developed to this purpose; this problem is referred to as *graph learning*. In [54], the authors formulate the graph learning problem as a precision matrix estimation with generalized Laplacian constraints. In [55], a sparse combinatorial Laplacian matrix is estimated from the data samples under a smoothness prior. In [56], a new class of transforms called graph template transform is proposed; the authors use a graph template to impose a sparsity pattern and approximate the empirical inverse covariance based on that template.

While the methods above are effective at deriving a graph from data, none of them takes into account the actual cost of representing, and thus coding, the graph, which is clearly a major problem for image compression. In [57] a novel graph-based framework is proposed, explicitly accounting for the cost of transmitting the graph. The authors treat the edge weights $w_{i,j}$ as a graph signal that lies on the dual graph. They compute the GFT of the weights graph and code its quantized transform coefficients. The choice of the graph is posed as a RD optimization problem.

3) Reducing GFT complexity: Besides the cost required to represent and encode the graph, the complexity of solving (2) to obtain the GFT matrix may outweigh any obtained coding gain. Indeed, applying the GFT to large blocks may quickly become infeasible. In [46] the authors propose to use a lookup table storing the GFTs for the most commonly used graphs, so that only the index of the corresponding

---

[5]Other approaches are also possible, e.g. [47] defines a GFT obtained from the eigendecomposition of the graph adjacency matrix.
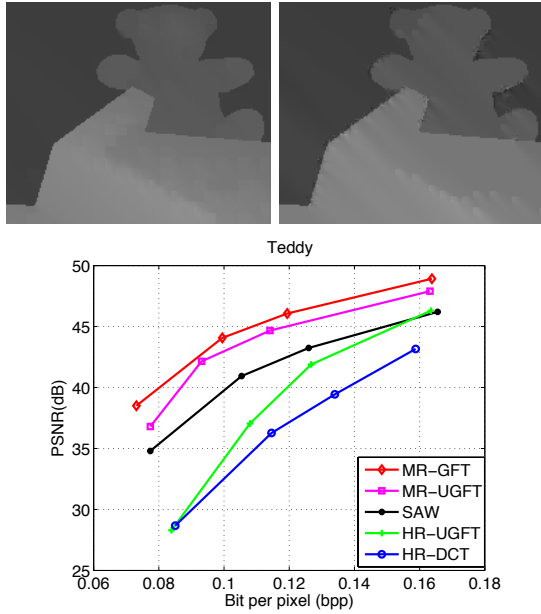
Fig. 3. Compression performance on the *Teddy* depth image, employing the MR-GFT [46], the MR-UGFT [4], H.264/AVC in intra mode (HR-DCT), the HR-UGFT [3], and the shape-adaptive wavelet (SAW, [60]). The top line shows reconstruction using the MR-GFT (left) and HR-DCT (right) at 0.1 bpp.

Fig. 4. (a) 2D-DCT basis vectors represented in matrix form (with $n = 8$): the corresponding two eigenvectors of an eigenvalue with multiplicity 2 are highlighted in red, the $n-1$ eigenvectors corresponding to $\lambda = 4$ are highlighted in blue and the $n-1$ eigenvectors corresponding to the eigenvalues with algebraic multiplicity 1 are highlighted in green. (b) Basis vectors of steered 2D-DCT with $\theta_{k,l} = \frac{\pi}{4} \ \forall k,l$.

chosen transform has to be transmitted; this has been shown to work well for relatively small block sizes. Moreover, in [4], [46] it is proposed to apply the GFT to a low-resolution version of the image, and to employ edge-adaptive filtering to restore the original resolution. In [58] graph-based separable transforms are proposed, where the transform is optimized separately along rows and columns. In [59] symmetric line-graph transforms are proposed, in which symmetries are exploited to reduce the number of operations needed to compute the transform.

4) Compression performance: Several authors have applied the GFT for image and video compression. In a practical setting, GFT coefficients over different image blocks may correspond to different frequencies (eigenvalues), making entropy coding somewhat more difficult. A possible solution employs bit-plane coding of coefficient significance, which depends only on the energy distribution of the transform coefficients. In Fig. 3 we report a comparison on compression of depth images. As can be seen, the MR-GFT codec [46] outperforms the other transforms in rate-distortion sense; in particular, gains between 5 and 10 dB are obtained with respect to the corresponding DCT-based coder. Correspondingly, at the same bit-rate the MR-GFT yields a depth image with less evident artifacts.

C. Steerable transforms from GFT

In many cases much of the content of an image block can be described by few main structures, employing a simplified image model with much fewer parameters, leading to reduced overhead. In particular, the directional model has become rather popular, e.g. in directional intra prediction modes [61]
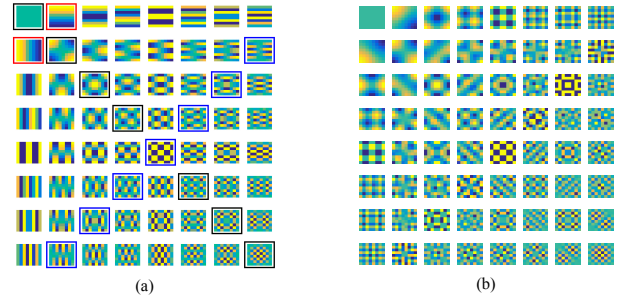
and directional transforms [62]–[68], including more sophisticated transforms such as bandelets [69] and anisotropic transforms [70]. The GFT framework can also be employed to design simplified adaptive transforms. As has been seen in Sec. II, the DCT is the GFT of the line graph with all weights equal to 1. In the same way, the basis vectors of the 2D-DCT are eigenvectors of the Laplacian of a square grid graph as in Fig. 2 [5], but the solution to (2) for a square grid graph is not unique because the eigenvalues of $\mathbf{L}$ do not all have algebraic multiplicity equal to one [71]. Using the pair (k,l) with $k,l \in [1,n]$ instead of index i in order to emphasize the bidimensionality of the basis vectors corresponding to the eigenvectors of $\mathbf{L}$ in the 2D case, it is easy to show that $\lambda_{k,l} = \lambda_{l,k}$ for $k \neq l$, i.e. these eigenvalues have multiplicity 2. Moreover, $\lambda_{k,n-k} = 4$ for $1 \leq k \leq n-1$, i.e. this eigenvalue has multiplicity $n-1$. Graphically this is shown in Fig. 4(a), where the basis vectors highlighted in red represent an example of eigenvectors corresponding to the same eigenvalue.

Therefore, the set of all possible eigenbases satisfying (2) for the Laplacian of a square grid graph can be represented as

$$\begin{bmatrix} \mathbf{u}'_{(k,l)} \\ \mathbf{u}'_{(l,k)} \end{bmatrix} = \begin{bmatrix} \cos\theta_{k,l} & \sin\theta_{k,l} \\ -\sin\theta_{k,l} & \cos\theta_{k,l} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{(k,l)} \\ \mathbf{u}_{(l,k)} \end{bmatrix}, \qquad (7)$$

where $\mathbf{u}_{(k,l)}$ are the eigenvectors corresponding to the basis vectors of the separable 2D-DCT. Indeed, (7) applies a rotation of an arbitrary angle $\theta_{k,l}$ to each pair of basis vectors $\mathbf{u}_{(k,l)}$ and $\mathbf{u}_{(l,k)}$. The new transform is defined by the new eigenvectors $\mathbf{u}'_{(k,l)}$, or equivalently and more handily by the original eigenvectors $\mathbf{u}_{(k,l)}$ plus the set of rotation angles $\theta_{k,l}$. Fig. 4(b) shows the resulting set of 2D basis vectors when $\theta_{k,l} = \frac{\pi}{4} \ \forall k,l$.

Such angles must be chosen to match the directional characteristics of the image block the transform is applied to, and to minimize the overhead of transmitting the angles. In [71] the same angle is chosen for all pairs of basis vectors with multiplicity equal to 2, i.e. $\theta_{k,l} = \theta$. In [57] $\theta_{k,l}$ is chosen individually for each pair, almost halving the number of DCT coefficients to be transmitted. In [72] the angles are chosen in a RD optimized fashion. In terms of implementation, in [72] it is noted that the coefficients of the steered transform can be

obtained from the coefficients of the separable 2D-DCT of the image block, followed by the application of a sparse rotation matrix; this makes the complexity only marginally higher than that of the separable 2D-DCT. Interestingly, the same principle can be applied to other transforms as well. In [8] it is shown that steerable 1D and 2D Discrete Fourier transforms (DFT) can be obtained. In the one-dimensional case, rotations change the balance of signal energy between the real and imaginary parts of the DFT; the resulting transform is related to the DCT, the discrete sine transform and the Hilbert transform. In 2D, rotations indeed correspond to geometric rotations.

D. Applications

We have previously mentioned applications of GFT to the compression of depth maps and natural images. Other authors have applied various types of GFT to video compression. In [73] the GFT is optimized for intra-prediction residues, while in [74] the authors propose a block-based lifting transform on graphs for intra-predicted video coding. A graph-based method for inter-predicted video coding has been introduced in [75], where the authors design a set of simplified graph templates capturing basic statistical characteristics of inter-predicted residual blocks. In [59] symmetric line-graph transforms are proposed for predictive video coding. In [76] a new edge coding method is introduced with application to intra prediction residuals.

Applications of GFT to other types of data have also been presented. In [77] a graph-based representation is applied to the problem of interactive multiview streaming, while in [78] a weighted GFT is employed for compression of light fields. In [79] the time-varying geometry of 3D point cloud sequences is represented as a set of graphs on which motion estimation is performed, whereas in [80] the graph representation is used to encode luminance information in multiview video.

Finally, a few works have employed graph wavelets to image/video coding problems. In [81] a graph wavelet transform has been proposed for image compression. In [82]–[84] the authors propose a complete video encoder based on lifting-based wavelet transforms on graphs; constructing a graph in which any pixel could be linked to several spatial and temporal neighbors, they jointly exploit spatial and temporal correlation. In [85] lifting-based graph wavelets are applied to compression of depth maps. In [86] graph wavelets are employed for the compression of hyperspectral images. These 3D images are characterized by a significant amount of correlation among images at different wavelengths, as well as spatial correlation, which are exploited constructing a spatial-spectral graph for groups of bands.

IV. GRAPH-BASED IMAGE RESTORATION

Image restoration is an inverse problem; given a noise-corrupted and/or degraded observation $\mathbf{y}$, one is tasked with restoring the original signal $\mathbf{x}$. Examples of restoration problems include image denoising, interpolation, super-resolution, deblurring, etc. An example generic image formation model is:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \qquad (8)$$

where $\mathbf{H}$ is a degradation matrix that performs down-sampling, blurring etc., and $\mathbf{z}$ is an additive noise.

Image restoration is an ill-posed problem, and thus prior knowledge about the sought signal $\mathbf{x}$ is required to regularize the problem. In this section, we describe recent graph-signal priors and their usages in the literature for image restoration.

A. Image Denoising

We start with image denoising, which is the most basic image restoration problem with image formation model (8) where $\mathbf{H} = \mathbf{I}$, and $\mathbf{z}$ is typically assumed to be an *additive white Gaussian noise* (AWGN).

Using a Bayesian approach, a typical *maximum a posteriori* (MAP) formulation has the following form:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu\, R(\mathbf{x}) \qquad (9)$$

where $R(\mathbf{x})$ is the negative log of a signal prior or regularization term for candidate signal $\mathbf{x}$, and μ is a weight parameter. The crux is to define a prior $R(\mathbf{x})$ that discriminates target signal $\mathbf{x}$ against other candidates, while keeping optimization (9) computationally efficient. There have been many priors $R(\mathbf{x})$ proposed with a varying degree of success; *e.g. total variation* (TV) [87], kernel regression [88], nonlocal means (NLM) [89], sparsity with respect to a pre-defined over-complete dictionary [90], etc. We discuss popular graph-signal priors in the literature, where the underlying graphs are often signal-adaptive.

Note that one may choose not to pose a MAP optimization like (9) at all; [30] argued it is more direct to address image denoising as a filtering problem:

$$\mathbf{x} = \mathbf{D}^{-1}\mathbf{W}\mathbf{y} \qquad (10)$$

where $\mathbf{D}^{-1}\mathbf{W}$ is row-stochastic, and filter coefficients in $\mathbf{W}$ are designed adaptively based on local / non-local statistics[6]. While graph-based filters derived from (9) can often be casted in the same framework in [30], we instead focus on the introduction of graph-based priors $R(\mathbf{x})$ for (9). We refer interested readers in image denoising using (10) to the extensive overview paper [30].

1) Sparsity of GFT Coefficients: One conventional approach is to map an observed signal $\mathbf{y}$ to a pre-selected transform domain, and assuming sparse signal representation in the domain, perform hard / soft thresholding on the transform coefficients [92]. Instead of pre-determined transforms and wavelets, one can use graph transforms and wavelets as basis and perform coefficient thresholding subsequently. Probabilistically, [93] showed that the graph Laplacian can be roughly interpreted as an inverse covariance matrix of a *Gaussian Markov Random Field* (GMRF), and thus the corresponding GFT is equivalent to the *Karhunen Loève Transform* (KLT) that deccorelates an input random signal. Hence it is reasonable to assume that an appropriately chosen GFT can sparsify a signal, resulting in a smaller l_0-norm.

As a concrete implementation, *non-local graph based transform* (NLGBT) [20] used GFT for depth image denoising as

---

[6]Instead of explicitly normalization, a recent work [91] shows that an image filter can be approximately normalized with lower complexity.

follows. Assuming self-similarity in images as done in NLM [89] and BM3D [94], $N-1$ similar patches $\mathbf{y}_i$, $i \geq 2$, to a target patch $\mathbf{y}_1$ are first searched in the depth image, in order to compute an average patch $\bar{\mathbf{y}}$. Assuming a four-connected graph that connects each pixel to its four nearest neighbors, the weight $w_{i,j}$ of an edge connecting pixels i and j is computed using (1). Note that the edge weights are computed using photometric distance, making the resulting filter signal-adaptive, thus improving its performance [30].

It is legitimate to ask how sensitive would the computed edge weights in (1) are to noise in observations. If one uses a pre-filtered version of the observation to compute edge weights using (1) [30], then it is shown that the computed eigenvectors are robust to noise [95]. In [96], the authors performed low-pass filtering on computed edge weights in a dual graph as pre-filtering. In [97], for piecewise smooth (PWS) images, the authors minimize the total variation of edge weights in a dual graph. In [20], the averaging over N patches effectively constitutes one low-pass pre-filtering.

Given graph Laplacian $\mathbf{L}$ computed for the constructed graph, GFT $\mathbf{U}^T$ is computed as the basis that spans the signal space. The N similar patches are denoised jointly as follows:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^N \|\mathbf{y}_i - \mathbf{U}\boldsymbol{\alpha}_i\|_2^2 + \tau \sum_{i=1}^N \|\boldsymbol{\alpha}_i\|_0 \qquad (11)$$

where the weight parameter τ can be estimated using *Stein Unbiased Risk Estimator* (SURE) [98]. Soft thresholding is used to iteratively minimize the second term. Shrinkage of transform coefficients for image denoising is common [99]. If the l_0-norm is replaced by a convex l_1-norm, then fast algorithms such as the split Bregman method [100] can be used.

(11) is solved iteratively, where between iterations edge weights are updated in (1) using computed solution in (11). [20] showed that for PWS images, the performance can outperform state-of-the-art algorithms like BM3D [94].

2) Graph Laplacian regularizer: Another common graph-signal prior is the *graph Laplacian regularizer* $R(\mathbf{x}) = \mathbf{x}^T \mathbf{L}\mathbf{x}$; it can be interpreted as a *Tikhonov regularizer* $\|\Gamma\mathbf{x}\|_2^2$ where $\Gamma = \mathbf{U}\Lambda^{1/2}\mathbf{U}^T$ given $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T$. From (3), minimizing $\mathbf{x}^T \mathbf{L}\mathbf{x}$ means that connected pixel pairs (i,j) by large edge weights $w_{i,j}$ will have similar sample values, or that the energy of the signal resides mostly in the low frequencies. $\mathbf{x}^T \mathbf{L}\mathbf{x}$ for restoration is prevalent across many fields, such as graph-based classifier in machine learning [21].

Using $R(\mathbf{x}) = \mathbf{x}^T \mathbf{L}\mathbf{x}$ in (9) leads to the following optimal solution $\mathbf{x}^*$:

$$\mathbf{x}^* = \mathbf{U}\text{diag}\left(\frac{1}{(1+\mu\lambda_1)}, \ldots, \frac{1}{(1+\mu\lambda_N)}\right)\mathbf{U}^T\mathbf{y} \qquad (12)$$

The resulting low-pass filter on $\mathbf{y}$ in GFT domain—smaller filter coefficient $(1+\mu\lambda_i)^{-1}$ for larger λ_i—can be implemented efficiently using Chebychev polynomial approximation, as discussed in Section V.

Alternatively, [13] defined signal smoothness using (5), and, assuming that the Hermetian of the weight matrix $\mathbf{W}^* = h(\mathbf{W})$ is a polynomial of $\mathbf{W}$, then the optimal MAP denoising filter with a l_2-norm fidelity term is derived as $g(\lambda_n)$ without matrix inversion:

$$g(\lambda_n) = \frac{1}{1 + \mu(1-\lambda_n)^2} \qquad (13)$$

See [13] for details.

It is known that the graph Laplacian can be derived from sample points of a differentiable manifold, and if the samples are randomly distributed, then the graph Laplacian operator converges to the Laplace-Beltrami operator in continuous manifold space when the number of samples tends to infinity [101]. The graph Laplacian regularizer can also be interpreted from a continuous manifold perspective [16], with additional insights that connect the prior to TV. Because edge weights $w_{i,j}$ in (1) are typically defined signal-adaptively, it is appropriate to write the prior as $\mathbf{x}^T \mathbf{L}(\mathbf{x})\mathbf{x}$. More generally, $w_{i,j}$ can be computed as the Gaussian of the difference in a set of pre-defined *exemplar functions* $f()$ evaluated at node i and j. Examples of $f()$ can be the x- and y-coordinates of a pixel, and intensity value of the pixel.

If we view the graph-signal as samples on a continuous manifold, then as the number of samples tends to infinity and the distances among neighboring samples go to 0, $\mathbf{x}^T \mathbf{L}(\mathbf{x})\mathbf{x}$ converges to a continuous functional [16],

$$\int_\Omega \nabla x^T \mathbf{G}^{-1} \nabla x \left(\sqrt{\det(\mathbf{G})}\right)^{2\gamma-1} d\mathbf{s} \qquad (14)$$

where $\mathbf{G}$ is defined as follows:

$$\mathbf{G} = \sum_{n=1}^n \nabla f_n \nabla f_n^T \qquad (15)$$

$\mathbf{G}$ can be viewed as the *structure tensor* of the gradient of the exemplar functions $\{\nabla f_n\}_{n=1}^N$. For convenience, define now $\mathbf{D} = \mathbf{G}^{-1}\left(\sqrt{\det(\mathbf{G})}\right)^{2\gamma-1}$. [16] then showed that the solution to the continuous counterpart of optimization (9) can be implemented as an anisotropic diffusion:

$$\partial_t x = \text{div}\left(\mathbf{D}\nabla x^*\right) \qquad (16)$$

$\mathbf{D}$ in this context is also the *diffusivity* that determines how fast an image is being diffused. For $\gamma < 1$, through eigen-analysis of $\mathbf{D}$ one can show that the diffusion process is divided into two steps: i) a forward diffusion process that *smooths* along an image edge, and ii) a backward diffusion process that *sharpens* perpendicular to an image edge. When $\gamma = 1$, the diffusion process is analogous to TV in the continuous domain. This explains why denoising using the graph Laplacian regularizer $\mathbf{x}^T \mathbf{L}(\mathbf{x})\mathbf{x}$ works particularly well for PWS images, such as depth images shown in Fig. 5.

Orthogonally, [102] proposed a fast graph Laplacian implementation of *low dimensional manifold model* (LDMM) [103]. In particular, LDMM [103] assumes that the size-d pixel patches of an image are points in a d-dimensional space that lie in a low-dimensional manifold, commonly called *patch manifold*. Thus the dimensionality of the manifold can be used as a prior to regularize an inverse problem:

$$\min_{\mathbf{x}\in\mathbb{R}^{m\times n}, \mathcal{M}\subset\mathbb{R}^d} \dim(\mathcal{M}) \quad \text{s.t. } \mathbf{y} = \mathbf{x} + \mathbf{z}, \quad P(\mathbf{x}) \subset \mathcal{M} \quad (17)$$

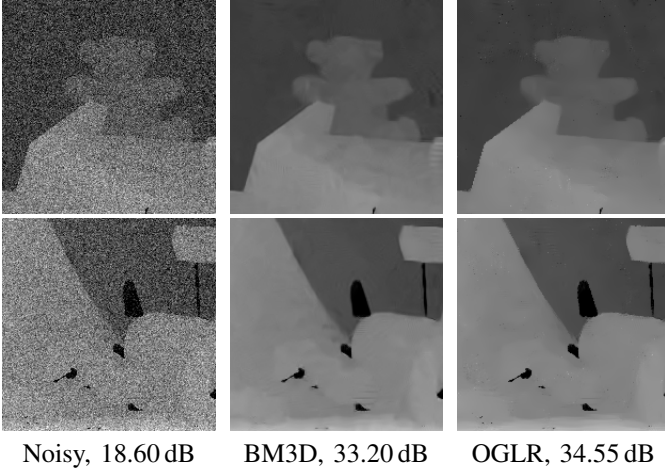Noisy, 18.60 dB   BM3D, 33.20 dB   OGLR, 34.55 dB

Fig. 5. Denoising of the depth image *Teddy*, where the original image is corrupted by AWGN with $\sigma_{\mathcal{I}} = 30$. Two cropped fragments of each image are presented for comparison.

where $\mathbf{y}$, $\mathbf{x}$ and $\mathbf{z}$ are the observed image, target image and noise respectively, $P(\mathbf{x})$ are the patches of image $\mathbf{x}$, and $\mathcal{M}$ is the patch manifold. [103] showed that the dimensionality of the manifold can be written as a sum of *coordinate functions*. For any $\mathbf{x} \in \mathcal{M}$:

$$\dim(\mathcal{M}) = \sum_{j=1}^{d} \|\nabla_{\mathcal{M}} \alpha_j(\mathbf{x})\|^2 \qquad (18)$$

where $\alpha_i(\mathbf{x})$ is the i-th coordinate function. (17) can be solved iteratively, where the key step to solve for the new image $\mathbf{x}^{k+1}$ and coordinate functions α_i^{k+1} requires a point integral method [104] that is computationally complex. Instead, [102] proposed to use a *weighted graph Laplacian* (WGL) method to replace the point integral method:

$$\min_{u} \sum_{\mathbf{x} \in P \setminus S} \left(\sum_{\mathbf{y} \in P} w(\mathbf{x}, \mathbf{y})(u(\mathbf{x}) - u(\mathbf{y}))^2 \right)$$
$$+ \frac{|p|}{|S|} \sum_{\mathbf{x} \in S} \left(\sum_{\mathbf{y} \in P} w(\mathbf{x}, \mathbf{y})(u(\mathbf{x}) - u(\mathbf{y}))^2 \right) \qquad (19)$$

The corresponding Euler-Lagrange equation is a linear system that is symmetric and positive definite. This is much easier to solve than the point integral method.

3) Graph Total Variation: Instead of the graph total variation (4) defined in [12], there exist works [22], [105]–[107] that defined and optimized TV for graphs in a more traditional manner as the seminal work [87][7]. Specifically, *local gradient* $\nabla_i \mathbf{x} \in \mathbb{R}^N$ at a node $i \in \mathcal{N}$ is first defined:

$$(\nabla_i \mathbf{x})_j = (x_j - x_i)W_{i,j} \qquad (20)$$

Then the (isotropic) graph total variation is defined as follows:

$$\|\mathbf{x}\|_{\text{TV}} = \sum_{i \in \mathcal{V}} \|\nabla_i \mathbf{x}\|_2 = \sum_{i \in \mathcal{V}} \sqrt{\sum_{j \in \mathcal{V}} (x_j - x_i)^2 W_{i,j}^2} \qquad (21)$$

---

[7] [107] actually defines a more general notion called *dual constrained total variation* (DCTV) that includes TV as a special case, and proposed a parallel proximal algorithm as solution.

Because the TV-norm is convex but non-smooth, there exist specialized algorithms that minimize it with a fidelity term, such as proximal gradient algorithms [106], [107].

As an illustrative example, in [22] a signal reconstruction given noise samples $\mathbf{y}$ from sampling matrix $\mathbf{S}$ is formulated as:

$$\min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x}\|_{\text{TV}} \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{S}\mathbf{x}\|_2 \le \epsilon \qquad (22)$$

To solve (22), the authors first convert the L_1-norm to its convex conjugate—a L_∞-norm ball—leading to a saddle point formulation, similarly done in [106]. Then they use a first-order primal-dual algorithm [108], since the new formulation has proximal operators that are much easier to compute. A distributed version of the algorithm is also provided when handling a large graph. Experimental results show that optimization of this graph TV norm (21) has better performance that earlier defined smoothness notions (3) and (4). See [22] for details.

4) Wiener Filter: More recently, instead of relying on a MAP formulation with sparsity or smoothness priors for regularization, one can approach the denoising problem from a statistical point of view and design a *Wiener filter* that minimizes the mean square error (MSE) instead [23], [109]. In particular, [23] first generalizes the notion of *wide-sense stationarity* (WSS) for graph-signals (with generalized translation and modulation operators on graphs [110]), estimates the *power spectral density* (PSD), and computes the minimal MSE (MMSE) graph Wiener filter. There are several advantages to employ a Wiener filter approach. First, unlike the smoothness prior that assumes implicitly a GMRF signal model, as long as the PSD can be robustly estimated, the Wiener filtering approach is more general and does not require a Gaussian assumption. Second, there is no need to tune a weight parameter (μ in (9)) to trade off the fidelity term with the prior term. Third, the specificity of the estimated PSD per graph frequency can be exploited during denoising.

Instead of executing the computed graph Wiener filter in the GFT domain, there exist fast methods based on Chebyshev polynomials [111] or Lanczos method [112] so that processing can be carried out locally in the vertex domain. Graph-signal filtering will be covered in more details in Section V.

5) Other Graph-based Image Denoising Approaches: We overview a few other notable approaches in graph-based image denoising. [113] performed image denoising by projecting an observed signal to a low-dimensional Krylov subspace of the graph Laplacian via a conjugate gradient method, resulting a fast image filtering operation that is competitive with Chebyshev polynomial approximation for the same order. As an extension, [114] performs edge sharpening using a graph with negative edges, implemented using the same projection method via conjugate gradient. [115] proposed a fast graph construction to mimic the performance of an edge-preserving bilateral filter (BF), where the computed sparse graph has eigenvectors in the graph spectral domain that are very close to the original BF. Edge-preserving smoothing is also considered in [116] via multiple Laplacians of affinity weights, each of which avoids computation-expensive normalization.

B. Image Deblurring

Image deblurring is more challenging than denoising, where the image model (8) has a blurring operator $\mathbf{H}$, which may or may not be known. Among many proposals in the literature [117], [118] is [119], which elects a graph-based approach. The unique aspect in [119] is that the similarity matrix $\mathbf{W}$ is first pre- and post-multiplied by a diagonal matrix $\mathbf{C}$, so that the resulting matrix $\mathbf{K}$ is both row- and column-stochastic:

$$\mathbf{K} = \mathbf{C}^{-1/2}\mathbf{W}\mathbf{C}^{-1/2} \tag{23}$$

$\mathbf{C}$ is computed using a fast implementation [120] of the Sinkhorn-Knop matrix scaling algorithm [121]. The resulting normalized Laplacian $\mathcal{L} = \mathbf{I} - \mathbf{K}$ is symmetric, positive semi-definite, and has the constant vector associated with eigenvalue 0. This results in the following objective ([119], eq.(16)):

$$\min_{\mathbf{x}} (\mathbf{y} - \mathbf{H}\mathbf{x})^T \{\mathbf{I} + \beta(\mathbf{I} - \mathbf{K})\} (\mathbf{y} - \mathbf{H}\mathbf{x})$$
$$+ \eta\mathbf{x}^T(\mathbf{I} - \mathbf{K})\mathbf{x} \tag{24}$$

where $\beta \geq -1$ and $\eta > 0$ are parameters. Note that formulation (24) is useful for any linear inverse problems. The solution $\mathbf{x}^*$ of (24) can be obtained by solving a system of linear equations via conjugate gradient. Similarity matrix $\mathbf{W}$ is then updated using computed $\mathbf{x}^*$, and the process is repeated for several iteration to remove blur.

In another approach, [122] extends the SURE-LET image deblurring framework in [118] for point cloud attributes (e.g., texture on 3D models). The key idea is to use graph to represent irregular 3D-point structures in a point cloud, so that subband decomposition and Wiener-like filtering via thresholding can be performed before reconstructing the signal. The blur kernel is replaced by Tikhonov regularized inverse for better condition number. See [122] for details.

C. Soft Decoding of JPEG Encoded Images

The graph Laplacian regularizer, which promotes PWS behavior in the reconstructed signal when used iteratively [16], [20], can be used in combination with other priors for image restoration; an earlier work [123], [124] combined the graph Laplacian regularizer with a kernel method for image restoration. To illustrate how different priors can be combined, we discuss the problem of soft decoding of JPEG images [125]. JPEG remains the prevalent image compression format worldwide, and thus optimizing image reconstruction from the compressed format remains important. Recall that in JPEG, each 8×8 pixel block is transformed via DCT to coefficients Y_i, each of which is scalar quantized:

$$q_i = \text{round}\,(Y_i/Q_i) \tag{25}$$

where Q_i is the quantization parameter (QP) for coefficient i. The quantized coefficients of different blocks are subsequently entropy-coded into the JPEG compressed format.

At the decoder, one must decide which coefficient value Y_i to reconstruct within the indexed quantization bin before inverse DCT to recover the pixel block:

$$q_iQ_i \leq Y_i < (q_i + 1)Q_i \tag{26}$$

To choose Y_i within the bin constraint (26), one must rely on signal priors. In [125], the authors used a combination of three priors that complement each other: Laplacian distribution for DCT coefficients, sparse representation given a compact pre-trained dictionary, and a new graph-signal smoothness prior. For initialization of the first solution, the first prior assumes that each DCT coefficient i follows a Laplacian distribution with parameter μ_i [126]. The second prior assumes that a pixel patch can be approximated by a sparse linear combination $\boldsymbol{\alpha}$ of atoms from an over-complete dictionary $\boldsymbol{\Phi}$ [90]. [125] shows that if $\boldsymbol{\Phi}$ is constrained in size due to computation cost, then the reconstructed patch would lack high DCT frequencies, resulting in blurs.

Finally, a new graph-signal smoothness prior using the *Left Eigenvectors of Random walk Graph Laplcian* (LERaG) is proposed. As previously discussed, iterative graph Laplacian regularizer promotes PWS behavior, thus recovering lost high DCT frequencies in a PWS pixel patch and complementing the restoration abilities of the aforementioned sparse coding using a small over-complete dictionary.

Further, for patch-based restoration, it is desirable in general to apply the *same* filtering strength when processing different patches in the image. Using previously described regularizer $\mathbf{x}^T\mathbf{L}\mathbf{x}$ where $\mathbf{L}$ is unnormalized, however, would mean that the strength of the resulting filtering depends on the total degree of the constructed graph. One alternative is to use the symmetric normalized graph Laplacian $\mathcal{L}$ to define smoothness prior $\mathbf{x}^T\mathcal{L}\mathbf{x}$. However, because the constant signal is not an eigenvector of $\mathcal{L}$, prior $\mathbf{1}^T\mathcal{L}\mathbf{1} > 0$, and the prior does not preserve constant signals that are common in natural images.

(a) initialization, LERaG = 76453.02, 2nd Eigenvalue = 0.001079

(b) iter = 1, LERaG = 64827.99, 2nd Eigenvalue = 315e-6

(c) iter = 2, LERaG = 14057.09, 2nd Eigenvalue = 35e-6
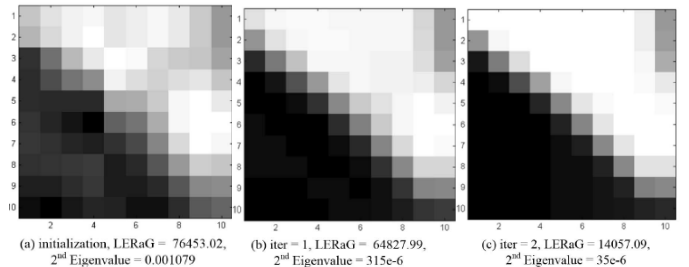
Fig. 6. 2nd eigenvalue of normalized graph Laplacian is monotonically decreasing with iteration numbers.

Instead, [125] proposed to use $\mathbf{x}^T\mathcal{L}_r^T\mathcal{L}_r\mathbf{x}$, computed efficiently as $\mathbf{x}^T(d_{\min}^{-1})\mathbf{L}\mathbf{D}^{-1}\mathbf{L}\mathbf{x}$, as the graph-signal smoothness prior, where $\mathcal{L}_r = \mathbf{D}^{-1}\mathbf{L}$ is the random walk graph Laplacian matrix. Like $\mathcal{L}$, $\mathcal{L}_r$ is normalized so the filter strength of the derived processing is the same for different patches. Yet unlike $\mathcal{L}$, $\mathbf{1}^T\mathcal{L}_r^T\mathcal{L}_r\mathbf{1} = 0$, and hence the prior can well preserve constant signals in natural images. Compared to the new normalized graph Laplacian matrix computed from a doubly stochastic similarity matrix as discussed earlier [119], [125] showed that LERaG outperforms this approach with a lower computation cost (see [125] for detailed comparisons).

Combining the sparsity prior and LERaG, we arrive at the

following optimization for soft decoding of a pixel patch $\mathbf{x}$:

$$\underset{\{\mathbf{x},\boldsymbol{\alpha}\}}{\arg\min} \|\mathbf{x} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda_1 \|\boldsymbol{\alpha}\|_0 + \lambda_2 \mathbf{x}^T (d_{\min}^{-1}) \mathbf{L}\mathbf{D}^{-1}\mathbf{L}\mathbf{x},$$
$$\text{s.t. } \mathbf{q}\mathbf{Q} \preceq \mathbf{T}\mathbf{M}\mathbf{x} \prec (\mathbf{q}+1)\mathbf{Q} \qquad (27)$$

where λ_1 and λ_2 are weight parameters, $\mathbf{q}$ and $\mathbf{Q}$ are the quantization bin indices and QP's, and both signal $\mathbf{x}$ and its sparse code $\boldsymbol{\alpha}$ are unknown. $\mathbf{x}$ and $\boldsymbol{\alpha}$ are solved alternately while holding the other variable fixed.

As suggested in [30], to improve filtering performance, (27) is computed iteratively, each time the edge weights in the graph are updated from the last computed solution $\mathbf{x}$. Due to the diffusion taking place as discussed previously, the filtered patch will increasingly become more PWS, as shown in Fig. 6. Note also that the second eigenvalue of the normalized graph Laplacian becomes increasingly smaller, resulting in a smaller prior cost. As shown in Fig. 7, the soft-decoded JPEG image `Butterfly` has higher quality than competing schemes.
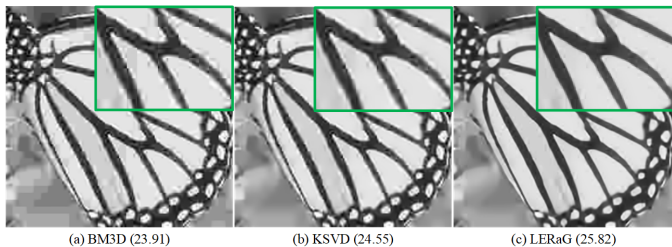

(a) BM3D (23.91) (b) KSVD (24.55) (c) LERaG (25.82)

Fig. 7. Comparison of tested methods in visual quality on *Butterfly* at QF = 5. The corresponding PSNR values are also given as references.

D. Other Graph-based Image Restorations

To show the breadth of applications using graph-based image restoration techniques, we briefly overview a few notable works. [127] first interpolates a full color image from Bayer-patterned samples, then based on the interpolated values computes edge weights for graph Laplacian regularization towards image demosaicking. For a stereo image pair with heterogeneous qualities, the higher-quality view image and corresponding disparity map are used to construct appropriate graph for bilateral filtering of the lower-quality view image in order to suppress noise [128]. Similar in concept, leveraging on the information provided by the high-resolution color image, resolution of the low-resolution depth image is enhanced via joint bilateral upsampling [129]. An image bit-depth enhancement scheme using a graph-signal smoothness assumption for the AC component in an image patch was proposed in [130].

V. GRAPH-BASED IMAGE FILTERING

As in image denoising and other inverse imaging, extracting smooth components of the image, i.e., low-graph-frequency components, is a critical issue since many image filtering applications utilize edge-preserving image smoothing as a key ingredient. This section introduces various image filtering methods using graph spectral analysis and shows relationships among them.

A. Smoothing and Diffusion in Graph Spectral Domain

One of the seminal works on smoothing using graph spectral analysis is 3-D mesh processing from computer graphics community [131], [132][8]. It determines edge weights of the graph as Euclidean distance between vertices (of the 3-D mesh) and smooths the 3-D mesh shape using a graph low-pass filter with a binary response. That is, the spectral response of the filter is

$$\widetilde{h}(\lambda_k) = \begin{cases} 1 & \text{if } k \leq T_k, \\ 0 & \text{otherwise,} \end{cases} \qquad (28)$$

where T_k is the user-defined bandwidth, i.e., how many eigenvalues are passed. Clearly, we can define an arbitrary response according to the purpose. This kind of naive approaches has been used in several computer graphics/vision tasks [133]–[137]. The filter in (28) actually smoothes out high-graph-frequency components, however, as the number of vertices grows, it is difficult to compute graph Fourier basis via eigendecomposition.

Heat kernel in the spectral domain has also been proposed in [138]. In this work, the weight of the edges of the graph is computed according to photometric distance, i.e., large weights are assigned to the edges whose both ends have similar pixel values and vice versa. Additionally, its graph spectral filter is defined as a solution of the heat equation on the graph as follows:

$$\widetilde{h}(\lambda) = e^{-t\lambda}, \qquad (29)$$

where $t > 0$ is an arbitrary parameter to control the spreading speed due to diffusion. By implementing it with the naive approach, it still needs a large computation cost due to eigendecomposition of graph Laplacian. However, (29) can also be represented by using Taylor series around the origin as

$$e^{-t\lambda} = \sum_{k=0}^{\infty} \frac{t^k}{k!}(-\lambda)^k. \qquad (30)$$

By truncating the above equation with an arbitrary order, we can approximate it as a finite-order polynomial [1], [111]. In [138], the Krylov subspace method is used along with (30) to approximate the graph filter.

However, as shown in Fig. 8, its approximation accuracy significantly gets worse for large λ. Since the maximum eigenvalue $\lambda_{\max}$ highly depends on the graph used, it is better to use different approximation methods (introduced in Section V-E).

B. Edge-Preserving Smoothing

As previously mentioned, edge-preserving smoothing is widely used for various image filtering tasks as well as image restoration [24], [139]–[146]. Image restoration aims to get the ground-truth image (approximately) from its degraded version, whereas edge-preserving smoothing is used to yield a user-desired image from the original one; It is either noisy or noise-free.

---

[8]The term "graph signal" was first introduced in [132], to the best of our knowledge.
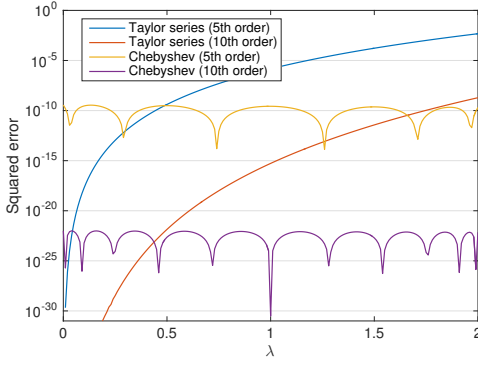
Fig. 8. Approximation error comparison of $\widetilde{h}(\lambda) = e^{-\lambda}$. The error is calculated as $E(\lambda) = (\widetilde{h}(\lambda) - \widetilde{h}_K^{\text{approx}}(\lambda))^2$, where $\widetilde{h}_K^{\text{approx}}(\lambda)$ is the Kth-order approximated response.

In the graph setting, we often need to define pixel-wise or patch-wise relationships as a distance between pixels or patches, and it is used to construct a graph. Three distances are considered in general [30]: 1) geometric distance, 2) photometric distance, and 3) their combination. Furthermore, especially for image filtering other than restoration, we often employ 4) *saliency* of the image/region/pixel, which simulates perceptual behavior [147], [148].

The graph spectral representation of bilateral filter [25] introduces that the bilateral filter can be regarded as a combination of graph Fourier basis and a graph low-pass filter. The filter coefficients of the bilateral filter are represented in (1). Since its weights clearly depend on the geometric and photometric distances, it is a pixel-dependent filter.

In a classical sense, the frequency domain representation of the bilateral filter cannot be calculated straightforwardly. In contrast, the bilateral filter can be considered as a graph filter by considering a weight matrix $\mathbf{W}$ where $[\mathbf{W}]_{ij} = w_{i,j}$ as an adjacency matrix of the graph. (1) is rewritten as

$$\widehat{\mathbf{x}} = \mathbf{D}^{-1}\mathbf{W}\mathbf{x} \tag{31}$$

where $\mathbf{D} = \text{diag}(d_0, d_1, \ldots, d_{N-1})$ in which $d_i = \sum_j w_{i,j}$. It is further rewritten as a graph spectral filter by

$$\widehat{\mathbf{x}} = \mathbf{D}^{-1/2}\mathbf{U}_n(\mathbf{I} - \mathbf{\Lambda}_n)\mathbf{U}_n^T\mathbf{D}^{1/2}\mathbf{x} \tag{32}$$

where we utilize the fact $\mathbf{W} = \mathbf{D} - \mathbf{L}$ and $\mathbf{L}_n = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. When we define a degree-normalized signal as $\underline{\mathbf{x}} = \mathbf{D}^{-1/2}\mathbf{x}$, the above equation is represented as

$$\widehat{\underline{\mathbf{x}}} = \mathbf{U}_n\widetilde{h}_{\text{BF}}(\mathbf{\Lambda}_n)\mathbf{U}_n^T\underline{\mathbf{x}}, \tag{33}$$

where $\widetilde{h}_{\text{BF}}(\lambda_n) = 1 - \lambda_n$. Since $\lambda_n \in [0, 2]$, it acts as a graph low-pass filter.

The above-mentioned representation of the bilateral filter suggests that the original bilateral filter implicitly designs the graph Fourier basis and the graph spectral filter simultaneously. For example, consider the following spectral response:

$$\widetilde{h}(\lambda) = \frac{1}{1 + \rho\widetilde{h}_r(\lambda)}, \tag{34}$$

where $\widetilde{h}_r(\lambda)$ is a graph high-pass filter and $\rho > 0$ is a parameter. Clearly, $\widetilde{h}(\lambda)$ works as a graph low-pass filter. It is the optimal solution to the following denoising problem [25]:

$$\arg\min_{\underline{\mathbf{x}}} ||\underline{\mathbf{y}} - \underline{\mathbf{x}}||_2^2 + \rho||\mathbf{H}_r\underline{\mathbf{x}}||_2^2, \tag{35}$$

where $\mathbf{y} = \mathbf{x} + \mathbf{e}$ in which $\mathbf{e}$ is a zero-mean i.i.d. Gaussian noise and $\mathbf{H}_r = \mathbf{U}\widetilde{h}_r(\mathbf{\Lambda})\mathbf{U}^T$.

It is known that the bilateral filter sometimes needs many iterations to smooth out details for textured and/or noisy images. To boost up the smoothing effect, the trilateral filter method [149] first smoothes gradients of the image, and then the smoothed gradient is utilized to smooth intensities. Its counterpart in the graph spectral domain has also been proposed in [150] with the parameter optimization method for ρ in (34) which minimizes MSE after denoising.

Other than the bilateral filter, non-local filters can also be interpreted as graph spectral filters like (33) while variational operators are not restricted to the symmetric normalized graph Laplacian and sometimes they are permitted to have negative-weighted edges. For example, [116], [151] introduce graph spectral filters based on non-local means [152] with random-walk graph Laplacian.

The power of graph spectral analysis for image filtering is that it is able to consider the prior information of the image, e.g., edges, textures, and saliencies, as a graph, separately from the user-desired information as a graph spectral response. That is, we can (and need to) design good graphs as well as graph filters for the desired image filtering effects. The design methods of graph spectral filters for various image processing tasks have been discussed in [116], [133], [151], [153]–[155] and references therein.

The above approach is generally represented as

$$\widehat{\mathbf{x}} = \mathbf{W}\widetilde{h}(\mathbf{\Lambda})\mathbf{W}^{-1}\mathbf{x}, \tag{36}$$

where $\mathbf{W} \in \mathbb{R}^{N \times N}$ is an arbitrary dictionary which sparsely represents the image $\mathbf{x}$ [9] and $\widetilde{h}(\mathbf{\Lambda}) = \text{diag}(\widetilde{h}(\lambda_0), \widetilde{h}(\lambda_1), \ldots)$ is a filter in the spectral domain (not restricted to the spectrum of the graph). This general form is considered in a modern image processing tasks [30] where $\mathbf{W}$ and $\mathbf{\Lambda}$ are obtained from a (symmetrized) matrix whose elements are come from arbitrary image processing. However, in this paper, we focus on a specific form of (36) in graph setting, where the dictionary is so-called graph Fourier basis and the spectral response is designed for the eigenvalues of the graph.

C. Relationship between Edge-Preserving Smoothing and Retargeting

For various image processing tasks including the topics described in this paper, filtering methods combining geometric and photometric distances and saliency have been proposed (see [24], [30], [88], [89], [146], [152], [156]–[158] and references therein). Among the works, *domain transform* [159], [160] has a unique approach: It transforms the photometric distance into the geometric distance, then the nonuniformly

---

[9]Generally the number of atoms in the dictionary could be overcomplete, but we focus on the square and invertible $\mathbf{W}$ for the sake of simplicity.

distributed discrete signal is low-pass filtered. Finally, the nonuniformly distributed signal is warped to its original pixel position to obtain the resulting smoothed image.

Formally, the domain transform for a 1-D signal is performed in the following steps.

1) Compute a warped pixel position according to the geometric and photometric distances.

$$t_i = t_{i-1} + \alpha_g + \alpha_p \sum_{k=0}^{N_c-1} |x_i^{(k)} - x_{i-1}^{(k)}|, \qquad (37)$$

where t_i is the ith pixel position ($t_0 = 0$), α_g and α_p are the weights for the geometric and photometric distances (usually $\alpha_g = 1$), respectively, $x_i^{(k)}$ is the ith pixel value of the kth color component, and N_c is the number of color channels, e.g., $N_c = 3$ for RGB color images.

2) Place x_i onto t_i as $f(t_i) := x_i$. At this time, $f(t_i)$ can be regarded as a discrete signal after nonuniform sampling.

3) Perform a low-pass filter $h(t)$ to $f(t_i)$ to obtain $\widehat{f}(t) = h(t) * f(t)$ defined in the continuous domain $t \in [0, t_{N-1}]$.

4) Replace the filtered signal $\widehat{f}(t_i)$ back to its original coordinates, i.e., $\widehat{x}_i = \widehat{f}(t_i)$.

The motivation of the domain transform is clearly shared with that of the graph-based image processing; The relationship between signal values is determined first, then the low-pass filter is performed to obtain user-desired effects.

The deformed pixel position t_i can also be regarded as a solution of the following linear problem [155].

$$\boldsymbol{\Psi} \mathbf{t} = \boldsymbol{\tau}, \qquad (38)$$

where

$$[\boldsymbol{\Psi}]_{ij} = \begin{cases} 1 & i = j, \\ -1 & i = j - 1, \\ 0 & \text{otherwise} \end{cases} \qquad (39)$$

and $\tau_i = \alpha_g + \alpha_p \sum_{k=0}^{N_c-1} |x_i^{(k)} - x_{i-1}^{(k)}|$. (38) can be solved by simply taking the inverse of $\boldsymbol{\Psi}$ and it is represented as

$$\boldsymbol{\Psi}^{-1} = \begin{bmatrix} 1 & 0 & \cdots \\ 1 & 1 & \ddots \\ \vdots & \ddots & \ddots \end{bmatrix}. \qquad (40)$$

Here, let us consider the following optimization problem:

$$\arg\min_{\mathbf{t}} ||\boldsymbol{\tau} - \boldsymbol{\Psi}\mathbf{t}||_2^2. \qquad (41)$$

Its solution is $\boldsymbol{\Psi}^T \boldsymbol{\Psi} \mathbf{t} = \boldsymbol{\Psi}^T \boldsymbol{\tau}$ and (38) is obtained when we multiply $\boldsymbol{\Psi}^{-T}$ for both sides. Interestingly, the above linear problem can also be represented as

$$\mathbf{L}_{\text{path}} \mathbf{t} - \boldsymbol{\tau}' = \mathbf{0}, \qquad (42)$$

where $\mathbf{L}_{\text{path}} = \boldsymbol{\Psi}^T \boldsymbol{\Psi}$ is the graph Laplacian of the path graph and $\tau_i' = \alpha_p \sum_{k=0}^{N_c-1} |x_{i+1}^{(k)} - x_i^{(k)}| - |x_i^{(k)} - x_{i-1}^{(k)}|$. It means we can define a generalized version of the domain transform and the general form is very closely related to various mesh deformation methods [133], [154], [161]–[163].

Mesh deformation is widely used in computer graphics and vision as well as image processing. The simplest form of the optimization problem can be formulated as follows [163]–[165]:

$$\mathbf{L}_1 \mathbf{p}' - \mathbf{L}_0 \mathbf{p} = \mathbf{0} \qquad (43)$$

where $\mathbf{L}_0$ and $\mathbf{L}_1$ are, respectively, the graph Laplacians for the original and deformed vertices of the mesh and $\mathbf{p}$ and $\mathbf{p}'$ are the original and deformed vertex coordinates, respectively. Since a pure Laplacian is a singular matrix, (43) needs constraints to obtain robust solutions. One of the widely-used constraints is the boundary condition which keeps the deformed vertex positions on the boundary unchanged.

(42) is a special version of (43) since $\boldsymbol{\tau}'$ in (42) represents the second-order differentiation of the deformed pixel position. Conversely, if we can define a "good" distance (depending on applications) between a pair of pixels, its deformed pixel position would be determined by solving a linear equation having the form like (43), as long as the overall cost function is quadratic. The desired pixel values could be obtained by a graph-based filtering.

If we perform a low-pass filter to the nonuniformly distributed pixel $f(t_i')$ then move it back to its original uniform-interval position, the graph-based filtering is an edge-preserving smoothing. Instead of that, if we interpolate the uniform-interval pixels $\widehat{f}(t_i)$ from $f(t_i')$, it is so-called content-aware image resizing, also known as *image retargeting* [163], [166], [167]. Their relationship is illustrated in Fig. 9.

Note that the conventional approaches need to consider signal processing in a continuous domain as a counterpart of the discrete domain where image signals exist. It leads to that we have to estimate the appropriate continuous domain from the input signal or any other prior information. Generally, it is a difficult task and requires a large computation cost due to signal processing in the continuous domain. In contrast, graph-based methods are fully discretized; memory and computation costs are usually kept low. Additionally, the prior information is appropriately utilized to construct/learn a graph for purposes.

D. Non-Photorealistic Rendering of Images

Edge-preserving smoothing is also widely used in non-photorealistic rendering (NPR), which is one of the key tasks in computer graphics. With a combination of image processing techniques like thresholding (both in spatial and frequency domains) and segmentation, NPR accomplishes various artificial effects such as stylization, pencil drawing, and abstraction [155], [159], [168]–[172]. Examples of NPR are shown in Fig. 10.

Sometimes one needs NPR images with different degrees of artificiality. We can accomplish them by defining different graphs and filters for different artificialities, however, it is generally a cumbersome process. Instead, multiscale decomposition of images would be an alternative way.

Traditionally, each scale represents an image component which has a specific frequency range. In contrast, graph-based multiresolution has more flexibility on the preserved component in each scale; It can also reflect the structure
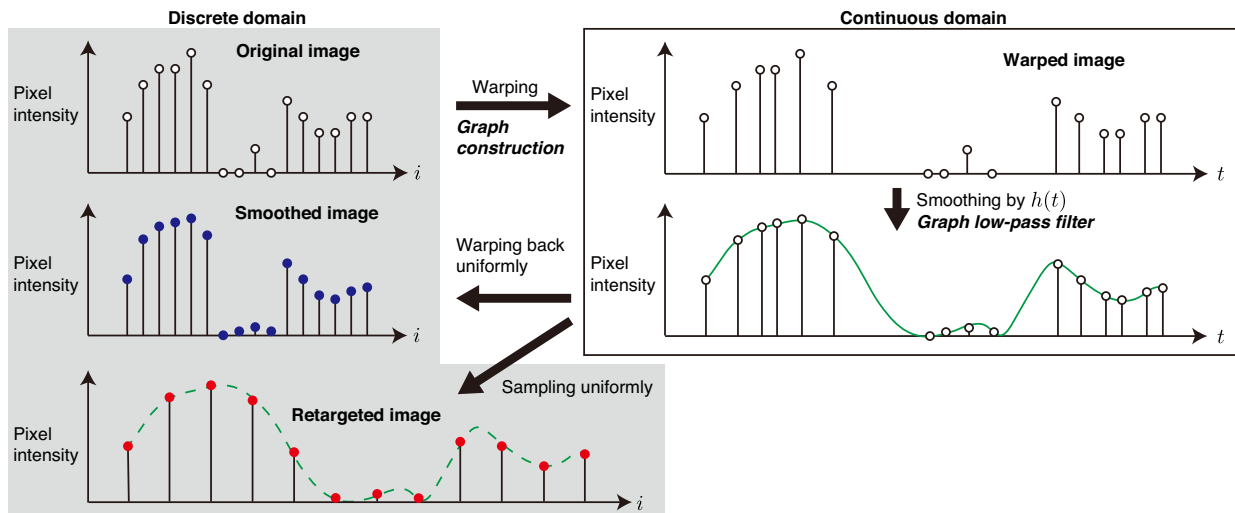
Fig. 9. Relationship between edge-preserving smoothing and image retargeting as signal processing on nonuniform grid. Graph signal processing corresponds to the construction of the appropriate graph and filtering of the nonuniformly sampled signal.

of pixels in each scale. For example, when we apply graph Laplacian pyramid [173] to an image with an appropriate multiscale graph and graph filters, high-graph-frequency component could represent pixel-level fine details, while the upper (coarser) level component would have region-level salient features. By changing functions to strengthen and weaken the transformed coefficients in each scale, we can obtain different NPR results from one multiscale representation of images. The multiscale representation has been proposed in the literature [133], [154], [155].

E. Fast Computation

Fast computation of graph spectral filtering is a key for its practical applications since the modern image filtering, including graph spectral filters, treats image signals as one long-vector $\mathbf{x} \in \mathbb{R}^{WH}$ where W and H are the width and height of the original image, respectively. The image resolution of digital broadcasting is becoming larger and larger; For example, 4K ultra-high-definition corresponds to $W = 3840$ and $H = 2160$ pixels, which leads to $WH > 8 \times 10^6$. As the naive approach, we have to construct a graph Laplacian of the size $WH \times WH$, then perform its eigendecomposition. This approach needs huge computational burden even in recent high-spec computers, and therefore, a workaround should basically be considered.

Although there are various methods to realize approximate graph spectral filtering, they can be divided into two approaches. One uses approximated eigenvectors (and eigenvalues) and the exact spectral filter response. The other uses an approximated spectral filter response and exact eigenvectors.

The first approach computes eigenvectors (or singular vectors for rectangular matrices) partially and/or approximately. Remaining eigenvectors are often approximated from the calculated eigenvectors. This approach can further be classified into two categories: Computing approximate eigenvectors from 1) graph Laplacian or other variation operators [174]–[176], and 2) pre-filtered images [151], [153]. Both can be applied to

Fig. 10. Examples of non-photorealistic rendering. Image stylization is shown in the left column and pencil drawing is shown in the right column. From top to bottom: Original, edge-preserving smoothing, and NPR results, respectively. The method in [153] is used for edge-preserving smoothing. For stylization, the edge image is combined with the smoothed image. For pencil drawing, edge detection is performed to the smoothed image and the edge image is combined with the high-frequency information in the image. Both test images are obtained from https://pixabay.com/.

any real symmetric matrices and the Nyström approximation method [177] plays a central role. They can drastically reduce the computation cost whereas it is required to decide how many eigenvectors are calculated prior to the decomposition.

The second approach uses the spectral response represented as a polynomial [116], [178]–[182]. Generally, if the filter response in the graph spectral domain is a Kth order polynomial function $\widetilde{h}(\lambda) = \sum_{k=0}^{K-1} a_k \lambda^k$, it can be represented as K

matrix-vector multiplications [1] since

$$\mathbf{U}\widetilde{h}(\mathbf{\Lambda})\mathbf{U}^T = \mathbf{U}\left(\sum_{k=0}^{K-1} a_k \mathbf{\Lambda}^k\right)\mathbf{U}^T = \sum_{k=0}^{K-1} a_k \mathbf{L}^k, \qquad (44)$$

where we utilize the fact $\mathbf{L}^k = \mathbf{U}\mathbf{\Lambda}^k\mathbf{U}^T$. This means we can use the exact full eigenvectors for filtering while the spectral response is approximated. It leads to that we have to choose good $\{a_k\}$. They can be determined empirically according to desired filtering effect [116], [153] or from a polynomial approximation of the desired spectral response [178]–[182].

The Kth order polynomial function on the graph can also be represented as the K-hop neighborhood transform in the vertex domain [1]. It leads to that the polynomial filters are localized in the vertex domain. Additionally, the output signal can be obtained from a distributed calculation.

Among the polynomial approximation methods, Chebyshev polynomial approximation [111], [182]–[185] is widely used for graph signal processing for some reasons. First, it can calculate with a recurrence relation; Memory requirement is small. Second, it produces low errors for the passband region. Third, it is very close to minimax polynomial and error bound can be calculated. Its approximation error for $\widetilde{h}(\lambda) = e^{-\lambda}$ is shown in Fig. 8. As compared with the Taylor series, the error by the Chebyshev approximation is bounded for all range of λ.

VI. GRAPH-BASED IMAGE SEGMENTATION

Image segmentation is an important and fundamental step in computer vision, image analysis and recognition [186]. It refers to partitioning an image into different regions where each region has its own meaning or characteristic in the image (e.g., the same color, intensity or texture). In the literature, there are a large number of image segmentation methods including threshold-based, edge-based, region-based and energy-based approaches; see the references in [187]. They have been applied to many image processing applications successfully, for example, in medical imaging, tracking and recognition.

For image segmentation methods, the energy-based approach is to develop and study an energy function which gives an optimum when the image is segmented into several regions according to the objective function criteria. This approach includes several techniques such as active contour (e.g., [188]) and graph cut (e.g., [26], [27]). The main advantage of using graph cut is that the associated energy function can be globally optimized whereas the other segmentation methods may not be guaranteed. In the graph cut segmentation, the energy function is constructed based on graphs where image pixels are mapped to graph vertices, and it can be optimized via graph-based algorithms and spectral graph theory results. By using the representation of graphs, morphological processing techniques can be applied to obtain many interesting image segmentation results, see for instance [189]. In this paper, we focus on the concept of graph cut segmentation and discuss its application to Mumford-Shah segmentation model.
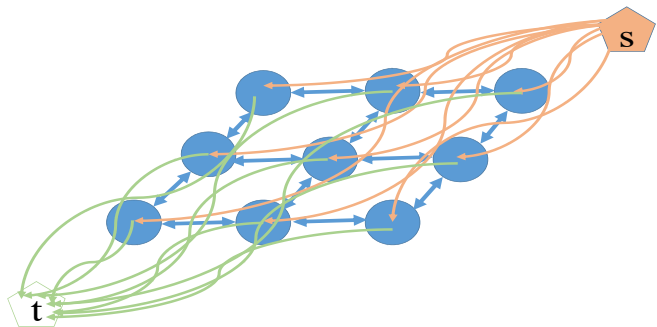

Fig. 11. An example of 3-by-3 image grid (blue circle: image pixel; blue arrowed line: pixel edge; brown arrowed line: an edge from the source vertex to pixel vertex; green arrowed line: an edge from pixel vertex to the sink vertex.

A. Graph Cut

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed of the vertex set $\mathcal{V}$ and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The vertex set $\mathcal{V}$ contains the nodes of a two-dimensional or three-dimensional image pixels together with two terminal vertices: the source vertex s and the sink vertex t. The edge set $\mathcal{E}$ contains two kinds of edges: (i) the edges $e = (i, j)$ where i and j are the image pixels except the source and the sink vertices; (ii) the terminal edges $e_s = (s, i)$ and $e_t = (i, t)$ where i is the image pixel except the source and the sink vertices. In two-dimensional or three-dimensional images, we usually assign an edge between two neighborhood pixels. We refer to Figure VI-A for a 3-by-3 for illustration. Moreover, the nonnegative cost $w_{i,j}$ is assigned to each edge $(i, j) \in \mathcal{E}$.

A cut on a graph is a partitioning of the vertices $\mathcal{V}$ into two disjoint and connected (through edges) sets $(\mathcal{V}_s, \mathcal{V}_t)$ such that $s \in \mathcal{V}_s$ and $t \in \mathcal{V}_t$. For each cut, the set of served edges $\mathcal{C}$ is defined as follows:

$$\mathcal{C}(\mathcal{V}_s, \mathcal{V}_t) = \{(i, j) \mid i \in \mathcal{V}_s, j \in \mathcal{V}_t \text{ and } (i, j) \in \mathcal{E}\}$$

We say that the graph cut uses the served edge (i, j) if (i, j) is contained in $\mathcal{C}$. Correspondingly, the cost of the cut is defined as follows:

$$\text{cost}(\mathcal{C}(\mathcal{V}_s, \mathcal{V}_t)) = \sum_{(i,j) \in \mathcal{C}(\mathcal{V}_s, \mathcal{V}_t)} w_{i,j}$$

In image segmentation, a cost function usually consists of the two terms: the region term and the boundary term [27]. The region term is used to give a cost function for a pixel assigned to a specific region. For example, the penalty can be referred to the difference between the intensity value of a pixel and the intensity model of the region. This term is usually used for the cost of edges between the source/sink vertex and pixel vertices. The boundary term is used to give a cost function when two neighborhood pixels are assigned to two different regions [26]. This term is usually used for the cost of edges between neighborhood pixels.

Basically, regional and edge information are used in graph cut. By incorporating shape information of the object into graph cut, image segmentation results can be improved. The main idea is to revise the region term and the boundary term in

cost function such that specific image segmentation results can be obtained. For instance, a distance function can be employed to represent some shapes for image segmentation [190] and surface segmentation [191].

1) Max-Flow and Min-Cut: A minimum cut is the cut that have the minimum cost called min-cut. As an example, in foreground-background segmentation application, V_s contains vertices that corresponds to the foreground region in an image and V_t contains vertices that corresponds to the background region in an image. We would like to to find a minimum cut containing two sets V_s and V_t such that the foreground and the background regions can be identified.

We note that each edge can be interpreted as a pipe and its edge cost can be considered as the capacity of this pipe. The max-flow problem is to find the largest amount of flow allowed to pass from the source vertex to the sink vertex subject to pipe capacity constraints and conservation of flows in the graph. By the duality theorem [192], the max-flow problem is equivalent to the min-cut problem. A globally optimum solution for min-cut can be found by using the max-flow algorithm (e.g., [192], [193]). Other graph cut implementations include push-relabel [193] and pseudo-flows [194], [195] techniques. They can be shown to be an iterative algorithm by generating a sequence of cuts such that the sequence converges to a global optimum solution. These iterative approaches can be interpreted as a splitting and merging method for finding an optimal graph partition. Such efficient graph cut algorithms [27], [196] are developed for image segmentation purpose. Their numerical examples have shown that the performance of these algorithms is significantly better than that of the standard max-flow technique. The main idea is to avoid combinatorial computational and introduce an iterative approach for finding an optimal solution. We will discuss this approach for solving Mumford-Shah segmentation model.

2) Normalized Cuts: In the literature, we know that a minimum cut may favour giving regions with a small number of vertices, see for instance [26], [197]. To avoid such situation for partitioning out small regions, the use of the normalized cut is proposed by Shi and Malik [26]. The cost of a cut is defined as a fraction of the total edge connections to all the vertices in the graph:

$$\text{cost}_n(\mathcal{C}) = \frac{\text{cost}(\mathcal{C}(\mathcal{V}_s, \mathcal{V}_t))}{\text{cost}(\mathcal{C}(\mathcal{V}_s, \mathcal{V}))} + \frac{\text{cost}(\mathcal{C}(\mathcal{V}_s, \mathcal{V}_t))}{\text{cost}(\mathcal{C}(\mathcal{V}_t, \mathcal{V}))},$$

where $\text{cost}(\mathcal{C}(\mathcal{V}_s, \mathcal{V}))$ is the sum of the cost of the edges between $\mathcal{V}_s$ and the whole set of vertices and $\text{cost}(\mathcal{C}(\mathcal{V}_s, \mathcal{V}))$ can be defined similarly. It is shown in [26] that the resulting optimization problem can be relaxed to solving an eigenvalue problem:

$$(\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})\mathbf{y} = \lambda\mathbf{y}.$$

The coefficient matrix is called a normalized Laplacian matrix. We note that spectral graph theory [2] can be used to study such normalized Laplacian matrix. The eigenvector corresponding to the second smallest eigenvalue of normalized Laplacian matrix provides a normalized cut. The eigenvector corresponding to the third smallest eigenvalue of normalized Laplacian matrix provides a partition of the first two regions

identified by the normalized cut. In practice, we can restart solving the partitioning problem on each subregion individually.

In the literature, other cuts are proposed and studied for image segmentation, for instance mean cut [198], ratio cut [199] and ratio regions approach [200]. In [187], some comparisons are presented for different graph cut approaches. Recently, an exact l_1 relaxation of the Cheeger ratio cut problem for multi-class transductive learning is studied in [201]. In general, the problem of finding a cut (min-cut, normalized cut, ratio cut, mean cut and ratio region) in an arbitrary graph is NP-hard. Definitely, efficient approximations to their solutions are required for image segmentation.

In some applications, a small number of pixels with known labels (foreground or background), the technique of random walks can be employed to assign each pixel to the label for which the largest probability is calculated. The framework can be interpreted as discrete potential theory an electrical circuits and the algorithm can be implemented on graphs that are constructed in Section II, see [202], [203]. A bilaterally constrained optimization model arising from the semi-supervised multiple-class image segmentation problem was developed in [204], [205].

B. The Mumford-Shah Model

In [28], Boykov et al. showed an interesting connection between graph cuts and level sets [206], and discussed how combinatorial graph cuts algorithms can be used for solving variational image segmentation problems such as Mumford-Shah functionals [207]. In [208], Yuan et al. further investigated novel max-flow and min-cut models in the spatially continuous setting, and showed that the continuous max-flow models correspond to their respective continuous min-cut models as primal and dual problems.

The Mumford-Shah model is an image segmentation model with a wide range of applications in imaging sciences. Let f be the target image. We would like to seek a partition $\{\Omega_i\}_{i=1}^n$ of the image domain Ω, and an approximation image u which minimizes the functional

$$J(u, \{\Gamma_i\}_{i=1}n) = \int_\Omega (u - f)^2 dx + \beta \int_{\Omega \setminus \cup_i \Gamma_i} |\nabla u|^2 dx + \nu \sum_{i=1}^n \int_{\Gamma_i} ds \qquad (45)$$

where $\{\Gamma_i\}_{i=1}^n$ denotes the interphases between the regions $\{\Omega_i\}_{i=1}^n$. It is interesting to note that when u is assumed to be constant within each Ω_i. The second term in (45) disappears and the resulting functional is given as follows:

$$J(u, \{\Gamma_i\}_{i=1}^n) = \int_\Omega (u - f)^2 dx + \nu \sum_{i=1}^n \int_{\Gamma_i} ds, \qquad (46)$$

where

$$u = \sum_{i=1}^n c_i \xi_i \qquad (47)$$

and ξ_i is the characteristic function of Ω_i.

In [209], Chan and Vese proposed to use level set functions to represent the above functional and solve the resulting optimization problem via the gradient descent method. Piecewise constant level set functions are used in [210]:

$$\phi = i \quad \text{in } \Omega_i, \ 1 \le i \le n.$$

The relationship between the characteristic function and the level set function is given as follows:

$$\xi_i = \frac{1}{\alpha_i} \prod_{j=1, j \ne i} (\phi - j) \quad \text{with } \alpha_i = \prod_{k=1, k \ne i} (i - k).$$

The length term in (46) can be approximated by the total variation of the level set function itself. The resulting Mumford-Shah functional becomes

$$J(u, \phi) = \int_{\Omega} (u - f)^2 dx + \nu \int_{\Omega} |\nabla \phi| dx. \tag{48}$$

Many research works have been studied to minimize (48) by continuous optimization methods such as the augmented Lagrangian method [210], [211] with the integer-valued constraint: $\prod_{i=1}^{n}(\phi - i) = 0$. Note that there are some variants of the total variation regularization term in the two-dimensional domain (x_1, x_2) setting. The isotropic form is given by $\int_{\Omega} \sqrt{|\phi_{x_1}|^2 + |\phi_{x_2}|^2} dx_1 dx_2$. The anisotropic form is given by $\int_{\Omega}(|\phi_{x_1}| + |\phi_{x_2}|)dx_1 dx_2$, and its modified form is given by $\int_{\Omega}(|\phi_{x_1}| + |\phi_{x_2}| + |R\phi_{x_1}| + |R\phi_{x_2}|)dx_1 dx_2$, where $R(\cdot)$ is the counterclockwise rotated gradient by $\pi/4$ radians used for creating more isotropic version.

1) Discrete Models: In [212], Bae et al. solved the minimization problem by graph cuts. They discretized the variational problem (48) on a grid, and the discrete energy function can be written as follows:

$$J_d(u, \phi) = \sum_i (u_i - f_i)^2 + \nu \sum_i \sum_{j \in \mathcal{N}(i)} w_{i,j} |\phi_i - \phi_j|, \tag{49}$$

where i and j refer to the grid points, the weights $w_{i,j}$ are given by $w_{i,j} = \frac{1}{k \times distance(i,j)}$, $distance(i,j)$ is the distance between the two grid points i and j, and k refers to the neighbourhood numbers in the discretization of different total variation forms.

2) Graph Cuts Minimization: For fixed values of $\{c_i\}_{i=1}^n$, the minimizer of (49) can be solved by finding the minimum cut over a constructed graph. It is not necessary to impose integer constraints in (49) to obtain integer-valued level set function ϕ_i. According to the optimization problem in (49), the set of vertices and the corresponding edges with their cost function can be constructed suitably. The work on graph cuts for the two regions Mumford-Shah model can be found in [213], [214].

For multiple regions, Bae et al. [212] designed multiple layers to deal with multiple regions. We refer to Figure 12 for one-dimensional example of five grid points and three regions segmentation for illustration. The graph consists of three layers referring to three regions segmentation ($n = 3$). Each layer contains five grid points as vertices (blue circles). The edges between grid points refer to their neighbourhoods (blue arrowed lines). The cost of these edges is related to the total variation regularization term (or the boundary term for
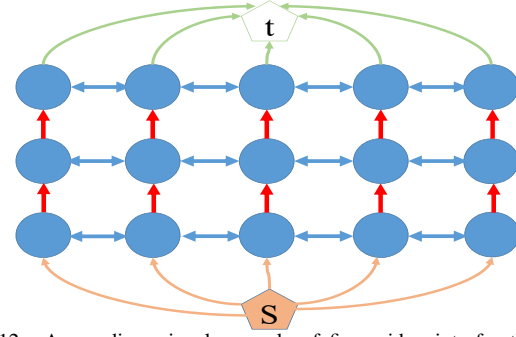

Fig. 12. A one-dimensional example of five grid points for three regions segmentation (blue circle: image pixel; blue arrowed line: an edge between two grid point vertices; brown arrowed line: an edge from the source vertex to a grid point vertex; green arrowed line: an edge from a grid point vertex to the sink vertex; red arrowed line: an edge from one region to another region.

the discontinuity of the two neighbourhood grid points). The source vertex and the sink vertex are also constructed in the graph. The cost of the edges between the source vertex to the vertices in the top layer, and between the sink vertex to the vertices in the bottom layer, refer to the region penalty term. It was shown in [212] that for any piecewise constant level set function ϕ taking values in $\{1, 2, \cdots, n\}$, there exists a unique admissible cut on the constructed graph. The level set function ϕ corresponds to a minimum cut in the constructed graph. After the level set function ϕ is determined, the values $\{c_i\}_{i=1}^n$ can be minimized by using the first term of (49), and they are given by

$$c_i = \frac{\sum_j f_j \xi_i(j)}{\sum_j \xi_i(j)}, \quad i = 1, 2, \cdots, n.$$

Numerical results in [212]–[214] have shown that this graph cut approach for solving the Mumford-Shah segmentation model is superior in efficiency compared to the partial differential equation based approach. Alternatively convex approaches to segmentation with active contours have also been considered, see for instance [215], [216].

This graph cut approach can be used to address a class of multi-labeling problems over a spatially continuous image domain, where the data fitting term can be of any bounded function, see [208], [217], [218]. It can also be extended to the convex relaxation of Pott's model [219] describing a partition of the continuous domain into disjoint subdomains as the minimum of a weighted sum of data fitting and the length of the partition boundaries. Recent research development along this direction include multi-class transductive learning based on L_1 relaxations of Cheeger cut and Mumford-Shah-Potts Model [201], and image segmentation by using the Ambrosio-Tortorelli functional and Discrete Calculus [220].

C. Graph BiLaplacian

Graph Laplacian matrix plays a leading role in these graph-based optimization methods. For example, Levin et al. [221] proposed a semi-supervised image matting method with closed form solution. Also Levin et al. [222] proposed a spectral matting method based on the spectral analysis of the matting

Laplacian matrix derived in [221]. Note that the matting Laplacian matrix can be viewed as a generalization of the graph Laplacian. Inspired by graph-based methods and their good performance, the graph Laplacian can be generalized to its second-order graph Laplacian, namely graph biLaplacian [29]. In particular, in an image when a vertex is only connected with its four neighbourhood vertices with equal edge weight, the graph biLaplacian is a finite difference approximation to the biharmonic operator in a continuous setting.

The i−th component of graph Laplacian of $u \in \mathbb{R}^n$ is

$$[\Delta_w u]_i = \sum_{j \in \mathcal{N}_i} w_{i,j}(u_i - u_j)$$

where $\mathcal{N}_i$ denotes the neighbourhood of the i vertex (all the vertices connected with the vertex i). The i-th component of the graph biLaplacian of u can be considered as follows:

$$(\Delta_w^2 u)_i = \sum_{j \in \mathcal{N}_i} w_{i,j}([\Delta_w u]_i - [\Delta_w u]_j).$$

The elements of the i-th row of the graph biLaplacian matrix Δ_w^2 are:

$$
\begin{aligned}
(\Delta_w^2)_{i,i} &= \sum_{j \in \mathcal{N}_i} w_{i,j}^2 + \sum_{j \in \mathcal{N}_i} \sum_{l \in \mathcal{N}_i} w_{i,j} w_{i,l}, \\
(\Delta_w^2)_{i,j} &= -\sum_{k \in \mathcal{N}_j} w_{i,j} w_{j,k} - \sum_{k \in \mathcal{N}_i} w_{i,j} w_{i,k}, \ j \in \mathcal{N}_i, \\
(\Delta_w^2)_{i,k} &= \sum_{j \in \mathcal{N}_i} w_{i,j} w_{j,k}, \ k \in \mathcal{N}_j, \ j \in \mathcal{N}_i \ k \neq i.
\end{aligned}
$$

The normalized graph biLaplacian matrix can be defined similarly. The spectral properties of graph biLaplacian and normalized graph biLaplacian can be found in [29].

We remark that the above formulation of graph Laplacian and graph biLaplacian is equivalent to the discretization of the harmonic and biharmonic PDE equation with Neumann boundary condition respectively. The harmonic equation is given by

$$\Delta u = 0, \text{in } \Omega, \quad \frac{\partial u}{\partial n}|_{\partial \Omega} = 0.$$

The biharmonic equation is

$$\Delta^2 u = 0, \text{in } \Omega, \quad \frac{\partial u}{\partial n}|_{\partial \Omega} = 0.$$

which comes from minimizing the following total squared curvature

$$\min \int_\Omega |\Delta u|^2 dx.$$

Harmonic and biharmonic equations and their numerical schemes are widely studied and applied in data interpolation, computer vision and image inpainting problems, see [223]–[228] and the references therein.

VII. Conclusion

Though graph signal processing (GSP) for large data networks has been studied intensively the last few years, applications of graph spectral techniques to image processing have received comparatively less attention. In this article, we overview recent developments of graph spectral algorithms for image compression, restoration, filtering and segmentation.

Because a digital image lives naturally on a discrete 2D grid, one key challenge for graph-based image processing is the appropriate selection of the underlying graph that describes the image structure for the graph-based tools that operate on top. For compression, the description of the graph translates to side information coding overhead. For restoration, filtering and segmentation, edge weights convey local signal similarity information, or a priori higher-level contextual information (e.g. saliency) that assist global processing operation. For future work, one focus is to design application-specific graph structures that target specific tasks like image enhancement, while trading off performance with computation complexity.

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

He was a senior researcher in Hewlett-Packard Laboratories Japan, Tokyo, from 2000 till 2009. He is now an associate professor in National Institute of Informatics in Tokyo, Japan. He has been an adjunct associate professor in the Hong Kong University of Science & Technology (HKUST) since 2015.

His research interests include 3D imaging and graph signal processing. He has served as associate editor for IEEE Transactions on Multimedia (2007–2011), DSP Applications Column in IEEE Signal Processing Magazine (2010–2014), SPIE Journal of Electronic Imaging (2014–2016) and IEEE Transactions on Circuits and Systems for Video Technology (2016–2017). He currently serves as associate editor for IEEE Transactions on Image Processing (2015–present) and APSIPA Journal on Signal & Information Processing (2011–present), and as area editor for EURASIP Signal Processing: Image Communication (2011–present). He served as a member of the Multimedia Signal Processing Technical Committee (MMSP-TC) in IEEE Signal Processing Society (2012–2014), and a member of the Image, Video, and Multidimensional Signal Processing Technical Committee (IVMSP-TC) (2015–2017, 2018-2020). He is a co-author of the best student paper award in IEEE Workshop on Streaming and Media Communications 2011 (in conjunction with ICME 2011), ICIP 2013, ICIP 2017 and IVMSP 2016, best paper runner-up award in ICME 2012, best paper finalists in ICME 2011, ICIP 2011 and ICME 2015, and IEEE Signal Processing Society (SPS) Japan best paper award 2016.

Enrico Magli (F'07–S97–M01–SM07) received the M.Sc. and Ph.D. degrees from Politecnico di Torino, Torino, Italy, in 1997 and 2001, respectively. He is currently a Full Professor with Politecnico di Torino, Torino, Italy. His research interests are in the field of graph signal processing, compressive sensing, image and video coding, and vision. He is an associate editor of the IEEE Transactions on Circuits and Systems for Video technology, the IEEE Transactions on Multimedia, and the EURASIP Journal on image and video processing. He is an IEEE Fellow and has been an IEEE Distinguished Lecturer for 2015-2016, and a corecipient of the IEEE Geoscience and Remote Sensing Society 2011 Transactions Prize Paper Award.

Yuichi Tanaka (S06–M07) received the B.E., M.E. and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2003, 2005 and 2007, respectively. He was a Postdoctoral Scholar at Keio University, Yokohama, Japan, from 2007 to 2008, and supported by the Japan Society for the Promotion of Science (JSPS). From 2006 to 2008, he was also a visiting scholar at the University of California, San Diego (Video Processing Group supervised by Prof. T. Q. Nguyen). From 2008 to 2012, he was an Assistant Professor in the Department of Information Science, Utsunomiya University, Tochigi, Japan. Since 2012, he has been an Associate Professor in Graduate School of BASE, Tokyo University of Agriculture and Technology, Tokyo, Japan. Currently he has a cross appointment as a PRESTO Researcher, Japan Science and Technology Agency. His current research interests are in the field of multidimensional signal processing which includes: graph signal processing, image and video processing with computer vision techniques, distributed video coding, objective quality metric, and effective spatial-frequency transform design.

Dr. Tanaka has been an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING since 2016 and served as an Associate Editor of *IEICE Transactions on Fundamentals* from 2013 to 2017. Currently he is an elected member of the APSIPA SIPTM (Signal and Information Processing Theory and Methods) and IVM (Image, Video and Multimedia) Technical Committees. He was a recipient of the Yasujiro Niwa Outstanding Paper Award in 2010, the TELECOM System Technology Award in 2011, and Ando Incentive Prize for the Study of Electronics in 2015. He also received IEEE Signal Processing Society Japan Best Paper Award in 2016 and Best Paper Awards in APSIPA ASC 2014 and 2015.

Michael K. Ng is a Chair Professor in the Department of Mathematics at the Hong Kong Baptist University. He obtained his B.Sc. degree in 1990 and M.Phil. degree in 1992 at the University of Hong Kong, and Ph.D. degree in 1995 at Chinese University of Hong Kong. He was a Research Fellow of Computer Sciences Laboratory at Australian National University (1995-1997), and an Assistant/Associate Professor (1997-2005) of the University of Hong Kong before joining Hong Kong Baptist University. His research interests include bioinformatics, image processing, scientific computing and data mining, and he serves on the editorial boards of international journals, see http://www.math.hkbu.edu.hk/~mng.

REFERENCES

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

[2] F. Chung, "Spectral graph theory," in *CBMS Regional Conference Series in Mathematics*, 1997.

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

[4] 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 International Conference on Image Processing*, Orlando, FL, September 2012.

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

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

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

[8] G. Fracastoro and E. Magli, "Steerable discrete Fourier transform," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 319–323, March 2017.

[9] I. Ram, M. Elad, and I. Cohen, "Generalized tree-based wavelet transform," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4199–4209, Sept 2011.

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

[11] S. K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4673–4685, Oct 2013.

[12] S. Chen, A. Sandryhaila, J. Moura, and J. Kovacevic, "Signal recovery on graphs: Variation minimization," in *IEEE Transactions on Signal Processing*, September 2015, vol. 63, no.17, pp. 4609–4624.

[13] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," in *IEEE Global Conference on Signal and Information Processing*, Austin, TX, December 2014.

[14] J. Pang, G. Cheung, W. Hu, and O. C. Au, "Redefining self-similarity in natural images for denoising using graph signal gradient," in *APSIPA ASC*, Siem Reap, Cambodia, December 2014.

[15] J. Pang, G. Cheung, A. Ortega, and O. C. Au, "Optimal graph Laplacian regularization for natural image denoising," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2015.

[16] J. Pang and G. Cheung, "Graph Laplacian regularization for inverse imaging: Analysis in the continuous domain," in *IEEE Transactions on Image Processing*, April 2017, vol. 26, no.4, pp. 1770–1785.

[17] O. Lezoray and L. Grady, *Image Processing and Analysis with Graphs: Theory and Practice*, CRC Press, 2017.

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

[19] W. Hu, G. Cheung, X. Li, and O. Au, "Depth map compression using multi-resolution graph-based transform for depth-image-based rendering," in *2012 19th IEEE International Conference on Image Processing*, Sept 2012, pp. 1297–1300.

[20] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *IEEE International Workshop on Multimedia Signal Processing*, Pula, Italy, October 2013.

[21] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semisupervised learning on large graphs," in *Shawe-Taylor J., Singer Y. (eds) Learning Theory, COLT 2004, Lecture Notes in Computer Science*, 2004, vol. 3120, pp. 624–638.

[22] P. Berger, G. Hannak, and G. Matz, "Graph signal recovery via primal-dual algorithms for total variation minimization," in *IEEE Journal on Selected Topics in Signal Processing*, September 2017, vol. 11, no.6, pp. 842–855.

[23] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," in *arXiv:1601.02522v5*, April 2017.

[24] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE International Conference on Computer Vision*, Bombay, India, 1998.

[25] A. Gadde, S. K. Narang, and A. Ortega, "Bilateral filter: Graph spectral interpretation and extensions," in *IEEE International Conference on Image Processing*, Melbourne, Australia, September 2013.

[26] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, vol. 22, pp. 888–905.

[27] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001, vol. 23, pp. 1222–1239.

[28] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," in *International Journal of Computer Vision*, 2006, pp. 109–131.

[29] F. Li and M. Ng, "Image colorization by using graph bilaplacian," in *Preprint*, 2017.

[30] P. Milanfar, "A tour of modern image filtering," in *IEEE Signal Processing Magazine*, January 2013, vol. 30, no.1, pp. 106–128.

[31] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," in *Biostatistics*, 2008, vol. 9, no.3, pp. 432–441.

[32] S. Daitch, J. Kelner, and D. Spielman, "Fitting a graph to vector data," in *ACM International Conference on Machine Learning*, Montreal, Canada, June 2009.

[33] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang, "Learning with l^1-graph for image analysis," in *IEEE Transactions on Image Processing*, April 2010, vol. 19, no.4, pp. 858–866.

[34] H. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," in *IEEE Journal on Selected Topics in Signal Processing*, September 2017, vol. 11, no.6, pp. 825–841.

[35] F. Dörfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, January 2013, vol. 60, no.1, pp. 150–163.

[36] T. Biyikoglu, J. Leydold, and P. F. Stadler, "Nodal domain theorems and bipartite subgraphs," in *Electronic Journal of Linear Algebra*, November 2005, vol. 13, pp. 344–351.

[37] P. Milanfar, "Symmetrizing smoothing filters," in *SIAM J. Imaging Science*, 2013, vol. 6, no.1, pp. 263–284.

[38] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," in *SIAM J. Imaging Sciences*, 2017, vol. 10, no.4, pp. 1804–1844.

[39] S. Venkatakrishnan, C. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, TX, December 2013.

[40] D. Thanou and P. Frossard, "Multi-graph learning of spectral graph dictionaries," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2015.

[41] K. Sayood, *Introduction to data compression, 3rd edition*, Morgan Kaufman, 2006.

[42] V. Goyal, J. Zhuang, and M. Vetterli, "Transform coding with backward adaptive updates," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1623–1633, 2000.

[43] G. Strang, "The discrete cosine transform," *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.

[44] A.K. Jain, "A sinusoidal family of unitary transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 4, pp. 356–365, 1979.

[45] R.A. DeVore, B. Jawerth, and B.J. Lucier, "Image compression through wavelet transform coding," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 719–746, 1992.

[46] W. Hu, G. Cheung, A. Ortega, and O. Au, "Multi-resolution graph Fourier transform for compression of piecewise smooth images," in *IEEE Transactions on Image Processing*, January 2015, vol. 24, no.1, pp. 419–433.

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

[48] I. Rotondo, G. Cheung, A. Ortega, and H. E. Egilmez, "Designing sparse graphs via structure tensor for block transform coding of images," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Dec 2015, pp. 571–574.

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

[50] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven graph transform for image compression," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 2631–2635.

[51] G. Fracastoro and E. Magli, "Predictive graph construction for image compression," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 2204–2208.

[52] A. Zheng, G. Cheung, and D. Florencio, "Context tree-based image contour coding using a geometric prior," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 574–589, Feb 2017.

[53] F. Verdoja and M. Grangetto, "Directional graph weight prediction for image compression," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 1517–1521.

[54] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," in *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2016, pp. 6350–6354.

[55] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[56] E. Pavez, H. E. Egilmez, Y. Wang, and A. Ortega, "GTT: Graph Template Transforms with applications to image coding," in *Proc. Picture Coding Symposium (PCS)*, 2015, pp. 199–203.

[57] G. Fracastoro, D. Thanou, and P. Frossard, "Graph transform learning for image compression," in *Proc. Picture Coding Symposium (PCS)*, 2016.

[58] H. E. Egilmez, Y. H. Chao, A. Ortega, B. Lee, and S. Yea, "Gbst: Separable transforms based on line graphs for predictive video coding," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 2375–2379.

[59] Keng-Shih Lu and Antonio Ortega, "Symmetric line graph transforms for inter predictive video coding," in *Proc. Picture Coding Symposium (PCS)*, 2016.

[60] M. Maitre and M.N. Do, "Depth and depth-color coding using shape-adaptive wavelets," *Journal of Visual Communication and Image Representation*, vol. 21, no. 5, pp. 513–522, 2010.

[61] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[62] Jizheng Xu, Bing Zeng, and Feng Wu, "An overview of directional transforms in image coding.," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 3036–3039.

[63] H. Xu, J. Xu, and F. Wu, "Lifting-based directional DCT-like transform for image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1325–1335, 2007.

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

[65] C. L. Chang and B. Girod, "Direction-adaptive partitioned block transform for image coding," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2008, pp. 145–148.

[66] F. Kamisli and J. S. Lim, "Transforms for the motion compensation residual," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, pp. 789–792.

[67] R. A. Cohen, S. Klomp, A. Vetro, and H. Sun, "Direction-adaptive transforms for coding prediction residuals," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2010, pp. 185–188.

[68] A. Drémeau, C. Herzet, C. Guillemot, and J.J. Fuchs, "Sparse optimization with directional dct bases for image compression," in *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 1290–1293.

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

[70] X. Peng, J. Xu, and F. Wu, "Directional filtering transform for image/intra-frame compression," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2935–2946, Nov 2010.

[71] G. Fracastoro and E. Magli, "Steerable discrete cosine transform," in *Proc. of IEEE International Workshop on Multimedia Signal Processing*, 2015, pp. 1–6.

[72] G. Fracastoro, S.M. Fosson, and E. Magli, "Steerable discrete cosine transform," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 303–314, 2015.

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

[74] Y. H. Chao, A. Ortega, and S. Yea, "Graph-based lifting transform for intra-predicted video coding," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 1140–1144.

[75] Hilmi E Egilmez, Amir Said, Yung-Hsuan Chao, and Antonio Ortega, "Graph-based transforms for inter predicted video coding," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 3992–3996.

[76] Y. H. Chao, H. E. Egilmez, A. Ortega, S. Yea, and B. Lee, "Edge adaptive graph-based transforms: Comparison of step/ramp edge models for video compression," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 1539–1543.

[77] B. Motz, G. Cheung, and P. Frossard, "Graph-based representation and coding of 3d images for interactive multiview navigation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 1155–1159.

[78] X. Su, M. Rizkallah, T. Maugey, and C. Guillemot, "Graph-based light fields representation and coding using geometry information," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2017.

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

[80] T. Maugey, Y. H. Chao, A. Gadde, A. Ortega, and P. Frossard, "Luminance coding in graph-based representation of multiview images," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 130–134.

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

[82] Eduardo Martínez-Enríquez and Antonio Ortega, "Lifting transforms on graphs for video coding," in *Proc. Data Compression Conference (DCC)*, 2011, pp. 73–82.

[83] Eduardo Martínez-Enríquez, Fernando Díaz-de María, and Antonio Ortega, "Video encoder based on lifting transforms on graphs," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 3509–3512.

[84] Eduardo Martinez-Enriquez, Jesus Cid-Sueiro, Fernando Diaz-De-Maria, and Antonio Ortega, "Directional transforms for video coding based on lifting on graphs," *IEEE Trans. Circuits Syst. Video Technol.*, 2016.

[85] Yung-Hsuan Chao, Antonio Ortega, Wei Hu, and Gene Cheung, "Edge-adaptive depth map coding with lifting transform on graphs," in *Proc. Picture Coding Symposium (PCS)*, 2015, pp. 60–64.

[86] J. Zeng, G. Cheung, Y.-H. Chao, I. Blanes, J. Serra-Sagristá, and A. Ortega, "Hyperspectral i mage coding using graph wavelets," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2017.

[87] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," in *Physica D: Nonlinear Phenomena*, November 1992, vol. 60, no.1, pp. 259–268.

[88] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," in *IEEE Transactions on Image Processing*, February 2007, vol. 16, no.3, pp. 349–366.

[89] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, CA, June 2005.

[90] M. Elad and M. Aharon, "Image denoising via sparse and redundant representation over learned dictionaries," in *IEEE Transactions on Image Processing*, December 2006, vol. 15, no.12.

[91] P. Milanfar and H. Talebi, "A new class of image filters without normalization," in *IEEE International Conference on Image Processing*, Phoenix, AZ, September 2016.

[92] D. Donoho and J. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," in *Biometrika*, August 2007, vol. 81, no.3.

[93] C. Zhang, D. Florencio, and P. A. Chou, "Graph signal processing–a probabilistic framework," in *Microsoft Res., Tech. Rep. MSR-TR-2015-31*, Redmond, WA, USA.

[94] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," in *IEEE Transactions on Image Processing*, August 2007, vol. 16, no.8, pp. 2080–2095.

[95] F.G. Meyer and X. Shen, "Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising," in *Appl. Comput. Harmon. Anal.*, March 2014, vol. 36, no.2, pp. 326–334.

[96] X. Liu, G. Cheung, and X. Wu, "Joint denoising and contrast enhancement of images using graph laplacian operator," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2015.

[97] W. Hu, G. Cheung, and M. Kazui, "Graph-based dequantization of block-compressed piecewise smooth images," in *IEEE Signal Processing Letters*, February 2016, vol. 23, no.2, pp. 242–246.

[98] F. Luisier, T. Blu, and M. Unser, "A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding," in *IEEE Transactions on Image Processing*, March 2007, vol. 16, no.3.

[99] Y. Iizuka and Y. Tanaka, "Depth map denoising using collaborative graph wavelet shrinkage on connected image patches," in *IEEE International Conference on Image Processing*, Paris, France, October 2014.

[100] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems," in *SIAM J. IMAGING SCIENCES*, 2009, vol. 2, no.2, p. 323343.

[101] M. Belkin and P. Niyogi, "Towards a theoretical foundation for Laplacian-based manifold methods," in *J. Comput. System Sci.*, 2005, vol. 74, pp. 1289–1308.

[102] Z. Shi, S. Osher, and W. Zhu, "Low dimensional manifold model with semi-local patches," in *UCLA CAM Tech Report 16-63*, 2016.

[103] S. Osher, Z. Shi, and W. Zhu, "Low dimensional manifold model for image processing," in *UCLA CAM Tech Report 16-04*, 2016.

[104] Z. Li and Z. Shi, "A convergent point integral method for isotropic elliptic equations on point cloud," in *SIAM : Multiscale Modeling Simulation*, 2016, vol. 14, pp. 874–905.

[105] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," in *IEEE Transactions on Image Processing*, July 2008, vol. 17, no.7, pp. 1047–1060.

[106] M. Hidane, o. Lezoray, and A. Elmoataz, "Nonlinear multilayered representation of graph-signals," in *Journal of Mathematical Imaging and Vision*, February 2013, vol. 45, no.2, pp. 114–137.

[107] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, and H. Talbot, "Dual constrained TV-based regularization on graphs," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2013, vol. 6, no.3, pp. 1246–1273.

[108] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," in *J. Math. Imag. Vis.*, 2011, vol. 40, no.1, pp. 120–145.

[109] A. C. Yagan and M. T. Ozgen, "A spectral graph wiener filter in graph fourier domain for improved image denoising," in *IEEE Global Conference on Signal and Information Processing*, Washington, DC, December 2016.

[110] D. Shuman, D. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," in *Applied and Computational Harmonic Analysis*, March 2016, vol. 40, no.2, pp. 260–291.

[111] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," via *Applied and Computational Harmonic Analysis*, 2011, vol. 30, no.2, p. 129150.

[112] A. Susnjara1, N. Perraudin, D. Kressner1, and P. Vandergheynst, "Accelerated filtering on graphs using Lanczos method," in *unpublished, arXiv:1509.04537*, 2015.

[113] D. Tian, H. Mansour, A. Knyazev, and A. Vetro, "Chebyshev and conjugate gradient filters for graph image denoising," in *IEEE International Conference on Multimedia and Expo Workshops*, Chengdu, China, July 2014.

[114] A. Knyazev, "Edge-enhancing filters with negative weights," in *IEEE Global Conference on Signal and Information Processing*, Orlando, FL, December 2015.

[115] A. Gadde, M. Xu, and A. Ortega, "Sparse inverse bilateral filters for image processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, March 2017.

[116] H. Talebi and P. Milanfar, "Fast multilayer laplacian enhancement," in *IEEE Transactions on Computational Imaging*, December 2016, vol. 2, no.4, pp. 496–509.

[117] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring using regularized locally adaptive kernel regression," in *IEEE Transactions on Image Processing*, April 2008, vol. 17, no.4, pp. 550–563.

[118] F. Xue, F. Luisier, and T. Blu, "Multi-wiener SURE-LET deconvolution," in *IEEE Transactions on Image Processing*, May 2013, vol. 22, no.5, pp. 1954–1968.

[119] A. Kheradmand and P. Milanfar, "A general framework for regularized, similarity-based image restoration," in *IEEE Transactions on Image Processing*, October 2014, vol. 23, no.12, pp. 5136–5151.

[120] P. Knight and D. Ruiz, "A fast algorithm for matrix balancing," in *IMA Journal of Numerical Analysis*, 2012.

[121] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," in *Pacific J. Math*, 1967, vol. 21, pp. 343–348.

[122] K. Yamanoto, M. Onuki, and Y. Tanaka, "Deblurring of point cloud attributes in graph spectral domain," in *IEEE International Conference on Image Processing*, Phoenix, AZ, September 2016.

[123] D. Zhai, X. Liu, D. Zhao, H. Chang, and W. Gao, "Progressive image restoration through hybrid graph laplacian regularization," in *Data Compression Conference*, Snowbird, UT, March 2013.

[124] X. Liu, D. Zhai, D. Zhao, G. Zhai, and W. Gao, "Progressive image denoising through hybrid graph laplacian regularization: A unified framework," in *IEEE Transactions on Image Processing*, January 2013, vol. 23, no.4, pp. 1491–1503.

[125] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Random walk graph laplacian based smoothness prior for soft decoding of JPEG images," in *IEEE Transactions on Image Processing*, February 2017, vol. 26, no.2, pp. 509–524.

[126] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," in *IEEE Transactions on Image Processing*, October 2000, vol. 9, no.10, pp. 1661–1666.

[127] C. Hu, L. Cheng, and Y. Lu, "Graph-based regularization for color image demosaicking," in *IEEE International Conference on Image Processing*, Orlando, FL, September 2012.

[128] D. Tian, H. Mansour, A. Vetro, Y. Wang, and A. Ortega, "Depth-assisted stereo video enhancement using graph-based approaches," in *IEEE International Conference on Image Processing*, Paris, France, October 2014.

[129] Y. Wang, A. Ortega, D. Tian, and A. Vetro, "A graph-based joint bilateral approach for depth enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014.

[130] P. Wan, G. Cheung, D. Flrencio, C. Zhang, and O. Au, "Image bit-depth enhancement via maximum-a-posteriori estimation of AC signal," in *IEEE Transactions on Image Processing*, June 2016, vol. 25, no.6, pp. 2896–2909.

[131] G. Taubin, "A signal processing approach to fair surface design," in *Proc. SIGGRAPH'95*, 1995, pp. 351–358.

[132] G. Taubin, T. Zhang, and G. H. Golub, "Optimal surface smoothing as filter design," in *Proc. ECCV'96*, 1996, pp. 283–292.

[133] H. Zhang, O. Van Kaick, and R. Dyer, "Spectral mesh processing," in *Computer Graphics Forum*, 2010, vol. 29, pp. 1865–1894.

[134] B. Vallet and B. Lévy, "Spectral geometry processing with manifold harmonics," in *Computer Graphics Forum*, 2008, vol. 27, pp. 251–260.

[135] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 317–324.

[136] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM transactions on graphics (TOG)*, 2003, vol. 22, pp. 950–953.

[137] J.-H. Kim, J. Garcia, and A. Ortega, "Dependent bit allocation in multiview video coding," in *IEEE International Conference on Image Processing*, Genoa, Italy, September 2005.

[138] F. Zhang and E. Hancock, "Graph spectral image smoothing using the heat kernel," in *Pattern Recognition*, November 2008, vol. 41, no.11, pp. 3328–3342.

[139] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer Graphics and Image Processing*, vol. 9, no. 4, pp. 394–407, 1979.

[140] C. Pomalaza-Raez and C. McGillem, "An adaptative, nonlinear edge-preserving filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 3, pp. 571–576, 1984.

[141] J. Weickert, *Anisotropic diffusion in image processing*, Teubner Stuttgart, 1998.

[142] D. Barash, "Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 844–847, 2002.

[143] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *ACM transactions on graphics (TOG)*, 2002, vol. 21, pp. 257–266.

[144] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," in *ACM Transactions on Graphics (TOG)*, 2008, vol. 27, p. 67.

[145] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via $l0$ gradient minimization," in *ACM Transactions on Graphics (TOG)*, 2011, vol. 30, p. 174.

[146] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.

[147] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 1998, vol. 20, no.11, pp. 1254–1259.

[148] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Advances in Neural Information Processing Systems*, 2006.

[149] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," in *Eurographics Rendering Symposium*, 2003, pp. 186–196.

[150] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, 2016.

[151] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 755–768, 2014.

[152] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[153] H. Talebi and P. Milanfar, "Nonlocal image editing," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4460–4473, 2014.

[154] G. Rong, Y. Cao, and X. Guo, "Spectral mesh deformation," *The Visual Computer*, vol. 24, no. 7, pp. 787–796, 2008.

[155] S. Yagyu, A. Sakiyama, and Y. Tanaka, "Pyramidal image representation with deformation: Reformulation of domain transform and filter designs," in *Proc. ICIP'16*, 2016, pp. 3608–3612.

[156] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradientshop: A gradient-domain optimization framework for image and video filtering," *ACM Trans. Graph.*, vol. 29, no. 2, pp. 10, 2010.

[157] S. Paris, S. W. Hasinoff, and J. Kautz, "Local laplacian filters: Edge-aware image processing with a Laplacian pyramid," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 68–1, 2011.

[158] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, "Fast local laplacian filters: Theory and applications," *ACM Tran. Graph.*, vol. 33, no. 5, pp. 167, 2014.

[159] E. S. L. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," in *ACM Transactions on Graphics (ToG)*, 2011, vol. 30, p. 69.

[160] E. S. L. Gastal and M. M. Oliveira, "High-order recursive filtering of non-uniformly sampled signals for image and video processing," in *Computer Graphics Forum*, 2015, vol. 34, pp. 81–93.

[161] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 355–361, 2002.

[162] M. Desbrun, M. Meyer, and P. Alliez, "Intrinsic parameterizations of surface meshes," in *Computer Graphics Forum*, 2002, vol. 21, pp. 209–218.

[163] Y. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized scale-and-stretch for image resizing," *ACM Trans. Graph.*, vol. 27, no. 5, 2008.

[164] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proc the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 175–184.

[165] K. Zhou, J. Huang, J. Snyder, X. Liu, H. Bao, B. Guo, and H.-Y. Shum, "Large mesh deformation using the volumetric graph Laplacian," in *ACM transactions on graphics (TOG)*, 2005, vol. 24, pp. 496–503.

[166] L. Wolf, M. Guttmann, and D. Cohen-Or, "Non-homogeneous content-driven video-retargeting," in *Proc. ICCV'07*, 2007.

[167] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin, "A shape-preserving approach to image resizing," in *Computer Graphics Forum*, 2009, vol. 28, pp. 1897–1906.

[168] Georges Winkenbach and David H Salesin, "Computer-generated pen-and-ink illustration," in *Proc. SIGGRAPH'94*, 1994, pp. 91–100.

[169] T. Strothotte and S. Schlechtweg, *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*, Morgan Kaufmann, 2002.

[170] S. Rusinkiewicz, F. Cole, D. DeCarlo, and A. Finkelstein, "Line drawings from 3D models," in *SIGGRAPH 2008 Classes*, 2008, p. 39.

[171] S. Yagyu, A. Sakiyama, and Y. Tanaka, "Deforming pyramid: Multi-scale image representation using pixel deformation and filters for non-equispaced signals," *IEICE Trans. Fundam. Electron. Commun. and Comput. Sci.*, vol. 99, no. 9, pp. 1646–1654, 2016.

[172] H. Sadreazami, A. Asif, and A. Mohammadi, "Iterative graph-based filtering for image abstraction and stylization," *IEEE Trans. Circuits Syst. II*, 2017.

[173] D. I Shuman, M. J. Faraji, and P. Vandergheynst, "A multiscale pyramid transform for graph signals," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 2119–2134, 2016.

[174] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, 2004.

[175] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *J. Machine Learning Research*, vol. 6, no. Dec, pp. 2153–2175, 2005.

[176] T.-H. Oh, Y. Matsushita, Y.-W. Tai, and I. So Kweon, "Fast randomized singular value thresholding for nuclear norm minimization," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4484–4493.

[177] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2001, pp. 682–688.

[178] Y. Saad, "Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems," *Mathematics of Computation*, vol. 42, no. 166, pp. 567–588, 1984.

[179] Effrosini Kokiopoulou and Yousef Saad, "Polynomial filtering in latent semantic indexing for information retrieval," in *Proc. ACM SIGIR Conf. Res. Develop. Info. Retrieval*, 2004, pp. 104–111.

[180] Y. Zhou and Y. Saad, "A Chebyshev–Davidson algorithm for large symmetric eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 3, pp. 954–971, 2007.

[181] J.-F. Cai and S. Osher, "Fast singular value thresholding without singular value decomposition," *Methods Appl. Anal.*, vol. 20, no. 4, pp. 335–352, 2013.

[182] M. Onuki, S. Ono, K. Shirai, and Y. Tanaka, "Fast singular value shrinkage with Chebyshev polynomial approximation based on signal sparsity," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 6083–6096, 2017.

[183] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*, CRC Press, 2002.

[184] G. M. Phillips, *Interpolation and Approximation by Polynomials*, New York: Springer, 2003.

[185] D. I Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *Proc. DCOSS'11*, 2011, pp. 1–8.

[186] R. Gonzalez and R. Woods, "Digital image processing," 2018, Pearson.

[187] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approahces to image segmenttion," in *Pattern Recognition*, 2013, vol. 46, pp. 1020–1038.

[188] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[189] L. Najman and J. Cousty, "A graph-based mathematical morphology reader," in *Pattern Recognition Letters*, 2014, pp. 3–17.

[190] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, pp. 147–159.

[191] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong, "An integral equation as surface evolution pdes via geo-cuts," in *European Conference on Computer Vision*, 2006, pp. 409–422.

[192] L. Ford and D. Fulkerson, "Flows in networks," 1962, University Press.

[193] A. Goldberg and R. Tarjan, "A new approach to the maximum-flow problem," in *Journal of the Association for Computing Machinery*, 1988, pp. 921–940.

[194] D. Hochbaum, "Thr pseudoflow algorithm for the maximum flow problem," in *IPC098*, 1988, pp. 325–337.

[195] O. Juan and Y. Boykov, "Active graph cuts," in *IEEE Conference of Computer Vision and Pattern Recognition*, 2006, pp. 1023–1029.

[196] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, pp. 1124–1137.

[197] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993, pp. 1101–1113.

[198] S. Wang and J. Siskind, "Image segementation with minimum mean cut," in *International Conference on Computer Vision*, 2001, pp. 517–524.

[199] S. Wang and J. Siskind, "Image segmentation with ratio cut," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, pp. 675–690.

[200] I. Cox, S. Rao, and Y. Zhong, "Ratio regions: A technique for image segmentation," in *International Conference on Pattern Recognition*, 1986, pp. 557–564.

[201] X. Bresson, X. Tai, T. Chan, and A. Szlam, "Multi-class transductive learning based on l1 relaxations of cheeger cut and mumford-shah-potts model," in *Journal of Mathematical Imaging and Vision*, 2014, pp. 191–201.

[202] Leo Grady, "Random walks for image segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, pp. 1768–1783.

[203] A. Elmoataz X. Desquesnes and O. Lézoray, "Eikonal equation adaptation on weighted graphs: Fast geometric diffusion process for local and non-local image and data processing," in *Journal of Mathematical Imaging and Vision*, 2013, pp. 238–257.

[204] G. Qiu M. Ng and A. Yip, "Numerical methods for interactive multiple class image segmentation problems," in *International Journal of Imaging Systems and Technology*, 2010, pp. 191–201.

[205] M. Ng Y. Law, H. Lee and A. Yip, "A semi-supervised segmentation model for collections of images," in *IEEE Transactions on Image Processing*, pp. 2955–2968.

[206] J. Sethian, "Level set methods and fast marching methods," 1999, Cambridge University Press.

[207] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," in *Communications on Pure and Applied Mathematics*, 1989, pp. 577–685.

[208] J. Yuan, E. Bae, X. Tai, and Y. Boykov, "A continuous max-flow approach to potts model," in *European Conference on Computer Vision*, 2010, pp. 379–392.

[209] T. Chan and L. Vese, "Active contours without edges," in *IEEE Transactions on Image Processing*, 2001, pp. 266–277.

[210] J. Lie, M. Lysaker, and X. Tai, "A variant of the level set method and applications to image segmentation," in *Mathematics of Computation*, 2006, pp. 1155–1174.

[211] P. Lin X. Tai, O. Christiansen and I. Skjaelaaen, "Image segmentation using some piecewise constant level set methods with mbo type of projection," in *International Journal of Computer Vision*, 2007, pp. 61–76.

[212] E. Bae and X. Tai, "Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation," in *International Conference on Scale Space and Variational Methods in Computer Vision*, 2009, pp. 1–13.

[213] J. Darbon, "A note on the discrete binary mumuford-shah model," in *Computer Vision / Computer Graphics Collaboration Techniques and Applications*, 2007, pp. 283–294.

[214] N. Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby, "Graph cut optimization for the mumford-shah model," in *International Conference on Visualization, Imaging and Image Processing*, 2007, pp. 182–187.

[215] A. Elmoataz O. Lézoray and V. Ta, "Nonlocal pdes on graphs for active contours models with applications to image segmentation and data clustering," in *International Conference on Acoustics, Speech, and Signal Processing*, 2012, pp. 873–876.

[216] K. Drakopoulos and P. Maragos, "Active contours on graphs: Multiscale morphology and graphcuts," in *J. Sel. Topics Signal Processing*, 2012, pp. 780–794.

[217] H. Ishikawa, "Higher-order clique reduction in binary graph cut," in *Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2993–3000.

[218] E. Bae, J. Yuan, X. Tai, and Y. Boykov, "A fast continuous max-flow approach to non-convex multi-labeling problems," in *Efficient Algorithms for Global Optimization Methods in Computer Vision*, 2014, pp. 134–154.

[219] R. Potts, "Some generalized order disorder transformations," in *Cambridge Philosophical Society*, 1952, pp. 106–109.

[220] J. Lachaud M. Foare and H. Talbot, "Image restoration and segmentation using the ambrosio-tortorelli functional and discrete calculus," in *ICPR*, 2016, pp. 1418–1423.

[221] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics*, 2004, pp. 689–694.

[222] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, pp. 228–242.

[223] C. Gáspár, "Multi-level biharmonic and bi-helmholtz interpolation with application to the boundary element method," in *Engineering Analysis with Boundary Elements*, 2000, pp. 559–573.

[224] I. Briggs, "Machine contouring using minimum curvature," in *Geophysics*, 1974, pp. 39–48.

[225] P. Bjøstad, "Fast numerical solution of the biharmonic dirichlet problem on rectangles," in *SIAM Journal on Numerical Analysis*, 1983, pp. 59–71.

[226] S. Hoffmann, G. Plonka, and J. Weickert, "Discrete green's functions for harmonic and biharmonic inpainting with sparse atoms," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2015, pp. 169–182.

[227] D. Terzopoulos, "Multilevel computational processes for visual surface reconstruction," in *Computer Vision, Graphics, and Image Processing*, 1983, pp. 52–96.

[228] L. Grady and J. Polimeni, "Discrete calculus: Applied analysis on graphs for computational science," 2010, Springer.