David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst

# The Emerging Field of Signal Processing on Graphs

© ISTOCKPHOTO.COM/JAMIE FARRANT

## Extending high-dimensional data analysis to networks and other irregular domains

In applications such as social, energy, transportation, sensor, and neuronal networks, high-dimensional data naturally reside on the vertices of weighted graphs. The emerging field of signal processing on graphs merges algebraic and spectral graph theoretic concepts with computational harmonic analysis to process such signals on graphs. In this tutorial overview, we outline the main challenges of the area, discuss different ways to define graph spectral domains, which are the analogs to the classical frequency domain, and highlight the importance of incorporating the irregular structures of graph data domains when processing signals on graphs. We then review methods to generalize fundamental operations such as filtering, translation, modulation, dilation, and downsampling to the graph setting and survey the localized, multiscale transforms that have been proposed to efficiently extract information from high-dimensional data on graphs. We conclude with a brief discussion of open issues and possible extensions.

*Digital Object Identifier 10.1109/MSP.2012.2235192*
*Date of publication: 5 April 2013*

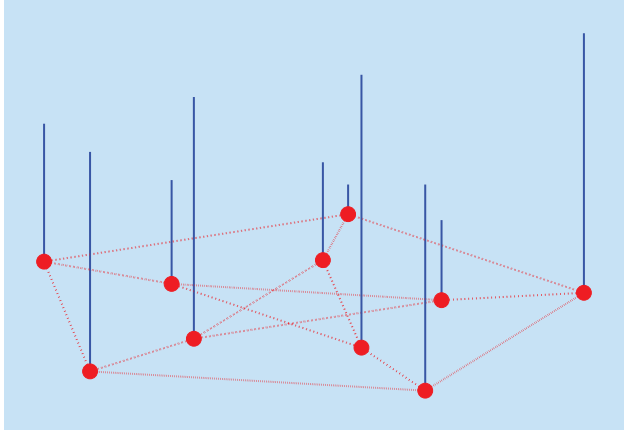
### INTRODUCTION

Graphs are generic data representation forms that are useful for describing the geometric structures of data domains in numerous applications, including social, energy, transportation, sensor, and neuronal networks. The weight associated with each edge in the graph often represents the similarity between the two vertices it connects. The connectivities and edge weights are either dictated by the physics of the problem at hand or inferred from the data. For instance, the edge weight may be inversely proportional to the physical distance between nodes in the network. The data on these graphs can be visualized as a finite collection of samples, with one sample at each vertex in the graph. Collectively, we refer to these

samples as a graph signal. An example of a graph signal is shown in Figure 1.

We find examples of graph signals in many different engineering and science fields. In transportation networks, we may be interested in analyzing epidemiological data describing the spread of disease, census data describing human migration patterns, or logistics data describing inventories of trade goods (e.g., gasoline or grain stocks). In brain imaging, it is now possible to noninvasively infer the anatomical connectivity of distinct functional regions of the cerebral cortex [1], and this connectivity can be represented by a weighted graph with the vertices corresponding to the functional regions of interest. Thus, noisy functional magnetic resonance imaging images can be viewed as signals on weighted graphs. Weighted graphs are also commonly used to represent similarities between data points in statistical learning problems for applications such as machine vision [2] and automatic text classification [3]. In fact, much of the literature on graph-based data analysis techniques emanates from the statistical learning community, as graph-based methods became especially popular for the semisupervised learning problem where the objective is to classify unknown data with the help of a few labeled samples (e.g., [4]–[9]). In image processing, there has been a recent spike in graph-based filtering methods that build nonlocal and semilocal graphs to connect the pixels of the image based not only on their physical proximity, but also on noisy versions of the image to be processed (e.g., [10]–[12] and references therein). Such methods are often able to better recognize and account for image edges and textures.

Common data processing tasks in these applications include filtering, denoising, inpainting, and compressing graph signals. How can data be processed on irregular data domains such as arbitrary graphs? What are the best ways to efficiently extract information, either statistically or visually, from this high-dimensional data, for the purposes of storage, communication, and analysis? Is it possible to use operators or algorithms from the classical digital signal processing toolboxes? These are just a few of the questions that underlie the field of signal processing on graphs.



**[FIG1] A random positive graph signal on the vertices of the Petersen graph. The height of each blue bar represents the signal value at the vertex.**

### *THE MAIN CHALLENGES OF SIGNAL PROCESSING ON GRAPHS*

The ability of wavelet, time-frequency, curvelet, and other localized transforms to sparsely represent different classes of high-dimensional data such as audio signals and images that lie on regular Euclidean spaces has led to a number of resounding successes in the aforementioned signal processing tasks (see, e.g., [13, Sec. II] for a recent survey of transform methods).

Both a signal on a graph with $N$ vertices and a classical discrete-time signal with $N$ samples can be viewed as vectors in $\mathbb{R}^N$. However, a major obstacle to the application of the classical signal processing techniques in the graph setting is that processing the graph signal in the same ways as a discrete-time signal ignores key dependencies arising from the irregular data domain. (Throughout this article, we refer to signal processing concepts for analog or discrete-time signals as "classical," to differentiate them from concepts defined in the graph signal framework.) Moreover, many extremely simple yet fundamental concepts that underlie classical signal processing techniques become significantly more challenging in the graph setting.

- To translate an analog signal $f(t)$ to the right by 3, we simply perform a change of variable and consider $f(t-3)$. However, it is not immediately clear what it means to translate a graph signal "to the right by 3." The change of variable technique will not work as there is no meaning to $f(\circ - 3)$ in the graph setting. One naive option would be to simply label the vertices from 1 to $N$ and define $f(\circ - 3) := f(\mathrm{mod}(\circ - 3, N))$, but it is not particularly useful to define a generalized translation that depends heavily on the order in which we (arbitrarily) label the vertices. The unavoidable fact is that weighted graphs are irregular structures that lack a shift-invariant notion of translation. (The exception is the class of highly regular graphs such as a ring graph that have circulant graph Laplacians. Grady and Polimeni [14, p. 158] refer to such graphs as shift invariant graphs.)
- Modulating a signal on the real line by multiplying by a complex exponential corresponds to translation in the Fourier domain. However, the analogous spectrum in the graph setting is discrete and irregularly spaced, and it is therefore nontrivial to define an operator that corresponds to translation in the graph spectral domain.
- We intuitively downsample a discrete-time signal by deleting every other data point, for example. Yet, what does it mean to downsample the signal on the vertices of the graph shown in Figure 1? There is not an obvious notion of "every other vertex" of a weighted graph.
- Even when we do fix a notion of downsampling, to create a multiresolution on graphs, we need a method to generate a coarser version of the graph that somehow captures the structural properties embedded in the original graph.

In addition to dealing with the irregularity of the data domain, the graphs in the previously mentioned applications can feature a

large number of vertices and therefore many data samples. To scale well with the size of the data, signal processing techniques for graph signals should employ localized operations that compute information about the data at each vertex by using data from a small neighborhood of vertices close to it in the graph.

Therefore, the overarching challenges of processing signals on graphs are 1) in cases where the graph is not directly dictated to us by the application, deciding how to construct a weighted graph that captures the geometric structure of the underlying data domain; 2) incorporating the graph structure into localized transform methods; 3) at the same time, leveraging invaluable intuitions developed from years of signal processing research on Euclidean domains; and 4) developing computationally efficient implementations of the localized transforms, to extract information from high-dimensional data on graphs and other irregular data domains.

To address these challenges, the emerging field of signal processing on graphs merges algebraic and spectral graph theoretic concepts with computational harmonic analysis. There is an extensive literature in both algebraic graph theory (e.g., [15]) and spectral graph theory (e.g., [16] and [17] and references therein); however, the bulk of the research prior to the past decade focused on analyzing the underlying graphs, as opposed to signals on graphs.

Finally, we should note that researchers have also designed localized signal processing techniques for other irregular data domains such as polygonal meshes and manifolds. This work includes, for example, low-pass filtering as a smoothing operation to enhance the overall shape of an object [18], transform coding based on spectral decompositions for the compression of geometry data [19], and multiresolution representations of large meshes by decomposing one surface into multiple levels with different details [20]. There is no doubt that such work has inspired and will continue to inspire new signal processing techniques in the graph setting.

## THE GRAPH SPECTRAL DOMAINS

Spectral graph theory has historically focused on constructing, analyzing, and manipulating graphs, as opposed to signals on graphs. It has proved particularly useful for the construction of expander graphs [21], graph visualization [17, Sec. 16.7], spectral clustering [22], graph coloring [17, Sec. 16.9], and numerous other applications in chemistry, physics, and computer science (see, e.g., [23] for a recent review).

In the area of signal processing on graphs, spectral graph theory has been leveraged as a tool to define frequency spectra and expansion bases for graph Fourier transforms. In this section, we review some basic definitions and notations from spectral graph theory, with a focus on how it enables us to extend many of the important mathematical ideas and intuitions from classical Fourier analysis to the graph setting.

### *WEIGHTED GRAPHS AND GRAPH SIGNALS*

We are interested in analyzing signals defined on an undirected, connected, weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, which consists of a finite set of vertices $\mathcal{V}$ with $|\mathcal{V}| = N$, a set of edges $\mathcal{E}$, and a weighted adjacency matrix $\mathbf{W}$. If there is an edge $e = (i, j)$ connecting vertices $i$ and $j$, the entry $W_{i,j}$ represents the weight of the edge; otherwise, $W_{i,j} = 0$. If the graph $\mathcal{G}$ is not connected and has $M$ connected components ($M > 1$), we can separate signals on $\mathcal{G}$ into $M$ pieces corresponding to the $M$ connected components, and independently process the separated signals on each of the subgraphs.

When the edge weights are not naturally defined by an application, one common way to define the weight of an edge connecting vertices $i$ and $j$ is via a thresholded Gaussian kernel weighting function

$$W_{i,j} = \begin{cases} \exp\left(-\dfrac{[\mathrm{dist}(i,j)]^2}{2\theta^2}\right) & \text{if } \mathrm{dist}(i,j) \le \kappa \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

for some parameters $\theta$ and $\kappa$. In (1), $\mathrm{dist}(i, j)$ may represent a physical distance between vertices $i$ and $j$, or the Euclidean distance between two feature vectors describing $i$ and $j$, the latter of which is especially common in graph-based semisupervised learning methods. A second common method is to connect each vertex to its $k$-nearest neighbors based on the physical or feature space distances. For other graph construction methods, see, e.g., [14, Ch. 4].

A signal or function $f : \mathcal{V} \to \mathbb{R}$ defined on the vertices of the graph may be represented as a vector $\mathbf{f} \in \mathbb{R}^N$, where the $i$th component of the vector $\mathbf{f}$ represents the function value at the $i$th vertex in $\mathcal{V}$. The graph signal in Figure 1 is one such example.

### *THE GRAPH LAPLACIAN*

The (unnormalized) graph Laplacian, also called the combinatorial graph Laplacian, is defined as $\mathcal{L} := \mathbf{D} - \mathbf{W}$, where the degree matrix $\mathbf{D}$ is a diagonal matrix whose $i$th diagonal element $d_i$ is equal to the sum of the weights of all the edges incident to vertex $i$. The graph Laplacian is a difference operator, as, for any signal $\mathbf{f} \in \mathbb{R}^N$, it satisfies

$$(\mathcal{L}f)(i) = \sum_{j \in \mathcal{N}_i} W_{i,j}[f(i) - f(j)],$$

where the neighborhood $\mathcal{N}_i$ is the set of vertices connected to vertex $i$ by an edge. More generally, we denote by $\mathcal{N}(i, k)$ the set of vertices connected to vertex $i$ by a path of $k$ or fewer edges.

Because the graph Laplacian $\mathcal{L}$ is a real symmetric matrix, it has a complete set of orthonormal eigenvectors, which we denote by $\{\mathbf{u}_\ell\}_{\ell = 0,1,\ldots,N-1}$. Note that there is not necessarily a unique set of graph Laplacian eigenvectors, but we assume throughout that a set of eigenvectors is chosen and fixed. These eigenvectors have associated real, nonnegative eigenvalues $\{\lambda_\ell\}_{\ell = 0,1,\ldots,N-1}$ satisfying $\mathcal{L}\mathbf{u}_\ell = \lambda_\ell \mathbf{u}_\ell$, for $\ell = 0, 1, \ldots, N-1$. Zero appears as an eigenvalue with multiplicity equal to the number of connected components of the graph [16], and thus, since we consider connected graphs, we assume that the graph Laplacian eigenvalues are ordered as $0 = \lambda_0 < \lambda_1 \le \lambda_2 \cdots \le \lambda_{N-1} := \lambda_{\max}$. We denote the entire spectrum by $\sigma(\mathcal{L}) := \{\lambda_0, \lambda_1, \ldots, \lambda_{N-1}\}$.

### A GRAPH FOURIER TRANSFORM AND NOTION OF FREQUENCY

The classical Fourier transform

$$\hat{f}(\xi) := \langle f, e^{2\pi i \xi t} \rangle = \int_{\mathbb{R}} f(t) e^{-2\pi i \xi t} dt$$

is the expansion of a function $f$ in terms of the complex exponentials, which are the eigenfunctions of the one-dimensional (1-D) Laplace operator

$$-\Delta(e^{2\pi i \xi t}) = -\frac{\partial^2}{\partial t^2} e^{2\pi i \xi t} = (2\pi\xi)^2 e^{2\pi i \xi t}. \tag{2}$$

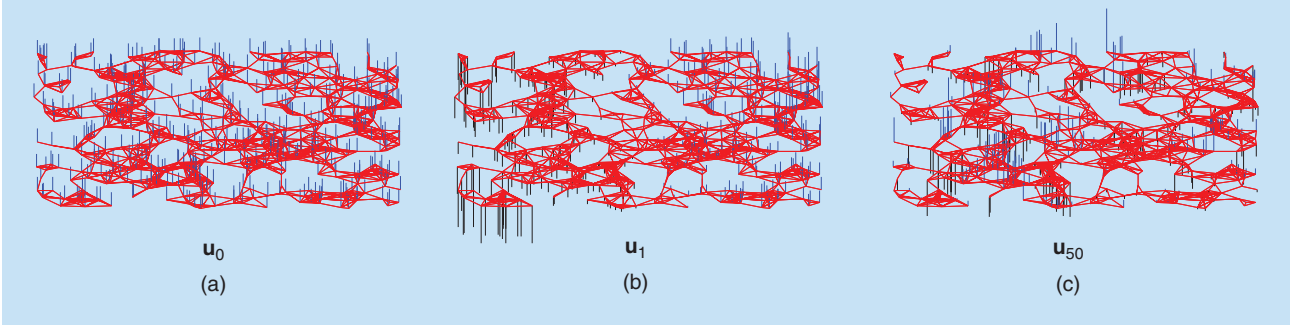
Analogously, we can define the graph Fourier transform $\hat{\mathbf{f}}$ of any function $\mathbf{f} \in \mathbb{R}^N$ on the vertices of $\mathcal{G}$ as the expansion of $\mathbf{f}$ in terms of the eigenvectors of the graph Laplacian:

$$\hat{f}(\lambda_\ell) := \langle \mathbf{f}, \mathbf{u}_\ell \rangle = \sum_{i=1}^{N} f(i) u_\ell^*(i). \tag{3}$$

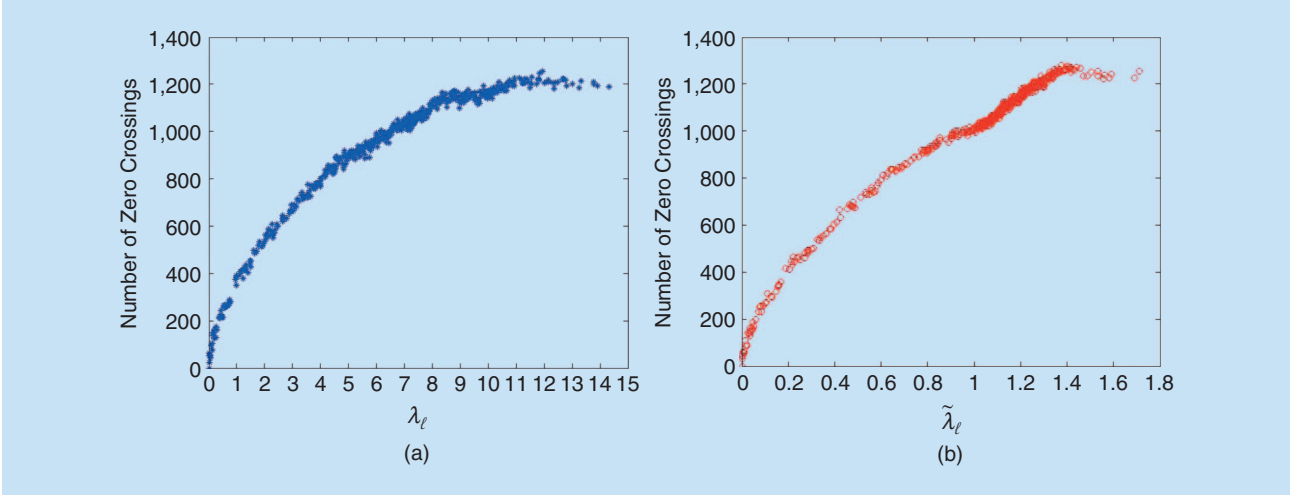
The inverse graph Fourier transform is then given by

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\lambda_\ell) u_\ell(i). \tag{4}$$

In classical Fourier analysis, the eigenvalues $\{(2\pi\xi)^2\}_{\xi \in \mathbb{R}}$ in (2) carry a specific notion of frequency: for $\xi$ close to zero (low frequencies), the associated complex exponential eigenfunctions are smooth, slowly oscillating functions, whereas for $\xi$ far from zero (high frequencies), the associated complex exponential eigenfunctions oscillate much more rapidly. In the graph setting, the graph Laplacian eigenvalues and eigenvectors provide a similar notion of frequency. For connected graphs, the Laplacian eigenvector $\mathbf{u}_0$ associated with the eigenvalue 0 is constant and equal to $1/\sqrt{N}$ at each vertex. The graph Laplacian eigenvectors associated with low frequencies $\lambda_\ell$ vary slowly across the graph, i.e., if two vertices are connected by an edge with a large weight, the values of the eigenvector at those locations are likely to be similar. The eigenvectors associated with larger eigenvalues oscillate more rapidly and are more likely to have dissimilar values on vertices connected by an edge with high weight. This is demonstrated in both Figure 2, which shows different graph Laplacian eigenvectors for a random sensor network graph, and Figure 3, which shows the number $|\mathcal{Z}_\mathcal{G}(\cdot)|$ of zero crossings of each graph Laplacian eigenvector. The set of zero crossings of a signal $\mathbf{f}$




**[FIG2] (a)–(c) Three graph Laplacian eigenvectors of a random sensor network graph. The signals' component values are represented by the blue (positive) and black (negative) bars coming out of the vertices. Note that $\mathbf{u}_{50}$ contains many more zero crossings than the constant eigenvector $\mathbf{u}_0$ and the smooth Fiedler vector $\mathbf{u}_1$.**

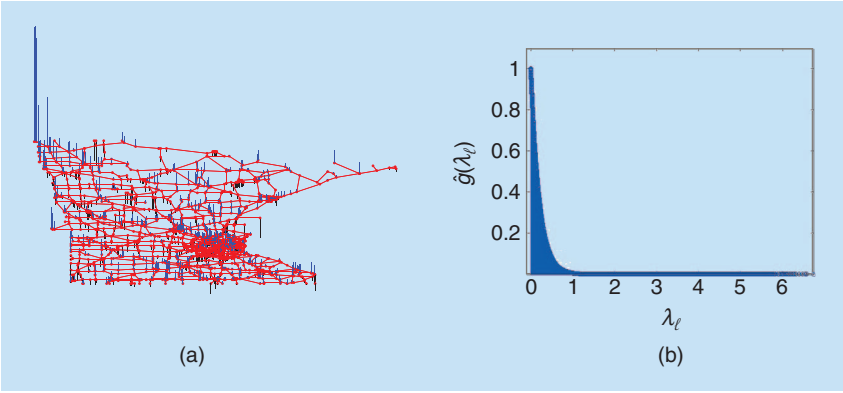



**[FIG3] The number of zero crossings, $|\mathcal{Z}_\mathcal{G}(\mathbf{u}_\ell)|$ in (a) and $|\mathcal{Z}_\mathcal{G}(\tilde{\mathbf{u}}_\ell)|$ in (b), of the unnormalized and normalized graph Laplacian eigenvectors for the random sensor network graph of Figure 2, respectively (the latter of which is defined in the section "Other Graph Matrices"). In both cases, the Laplacian eigenvectors associated with larger eigenvalues cross zero more often, confirming the interpretation of the graph Laplacian eigenvalues as notions of frequency.**

on a graph $\mathcal{G}$ is defined as the set of edges connecting a vertex with a positive signal to a vertex with a negative signal: $\mathcal{Z}_{\mathcal{G}}(\mathbf{f}) := \{e = (i, j) \in \mathcal{E} : f(i)f(j) < 0\}$.

### GRAPH SIGNAL REPRESENTATIONS IN TWO DOMAINS

The graph Fourier transform (3) and its inverse (4) give us a way to equivalently represent a signal in two different domains: the vertex domain and the graph spectral domain. While we often start with a signal $\mathbf{g}$ in the vertex domain, it may also be useful to define a signal $\hat{\mathbf{g}}$ directly in the graph spectral domain. We refer to such signals as kernels. In Figure 4(a) and (b), one such signal, a heat kernel, is shown in both domains. Analogously to the classical analog case, the graph Fourier coefficients of a smooth signal such as the one shown in Figure 4 decay rapidly. Such signals are compressible as they can be closely approximated by just a few graph Fourier coefficients (see, e.g., [24]–[26] for ways to exploit this compressibility).




**[FIG4] Equivalent representations of a graph signal in the vertex and graph spectral domains. (a) A signal $g$ that resides on the vertices of the Minnesota road graph [27] with Gaussian edge weights as in (1). The signal's component values are represented by the blue (positive) and black (negative) bars coming out of the vertices. (b) The same signal in the graph spectral domain. In this case, the signal is a heat kernel, which is actually defined directly in the graph spectral domain by $\hat{g}(\lambda_\ell) = e^{-5\lambda_\ell}$. The signal plotted in (a) is then determined by taking an inverse graph Fourier transform (4) of $\hat{\mathbf{g}}$.**

### DISCRETE CALCULUS AND SIGNAL SMOOTHNESS WITH RESPECT TO THE INTRINSIC STRUCTURE OF THE GRAPH

When we analyze signals, it is important to emphasize that properties such as smoothness are with respect to the intrinsic structure of the data domain, which in our context is the weighted graph. Whereas differential geometry provides tools to incorporate the geometric structure of the underlying manifold into the analysis of continuous signals on differentiable manifolds, discrete calculus provides a "set of definitions and differential operators that make it possible to operate the machinery of multivariate calculus on a finite, discrete space" [14, p. 1].

To add mathematical precision to the notion of smoothness with respect to the intrinsic structure of the underlying graph, we briefly present some of the discrete differential operators defined in [4], [6]–[8], [14], and [28]–[30]. Note that the names of many of the discrete calculus operators correspond to the analogous operators in the continuous setting. In some problems, the weighted graph arises from a discrete sampling of a smooth manifold. In that situation, the discrete differential operators may converge—possibly under additional assumptions—to their namesake continuous operators as the density of the sampling increases. For example, [31]–[34] examine the convergence of discrete graph Laplacians (normalized and unnormalized) to continuous manifold Laplacians.

The edge derivative of a signal $\mathbf{f}$ with respect to edge $e = (i, j)$ at vertex $i$ is defined as

$$\left.\frac{\partial \mathbf{f}}{\partial e}\right|_i := \sqrt{W_{i,j}}[f(j) - f(i)],$$

and the graph gradient of $\mathbf{f}$ at vertex $i$ is the vector

$$\nabla_i \mathbf{f} := \left[\left\{\left.\frac{\partial \mathbf{f}}{\partial e}\right|_i\right\}_{e \in \mathcal{E} \text{ s.t. } e=(i,j) \text{ for some } j \in \mathcal{V}}\right].$$

Then the local variation at vertex $i$

$$\begin{aligned}\|\nabla_i \mathbf{f}\|_2 &:= \left[\sum_{e \in \mathcal{E} \text{ s.t. } e=(i,j) \text{ for some } j \in \mathcal{V}} \left(\left.\frac{\partial \mathbf{f}}{\partial e}\right|_i\right)^2\right]^{\frac{1}{2}} \\ &= \left[\sum_{j \in \mathcal{N}_i} W_{i,j}[f(j) - f(i)]^2\right]^{\frac{1}{2}}\end{aligned}$$

provides a measure of local smoothness of $\mathbf{f}$ around vertex $i$, as it is small when the function $\mathbf{f}$ has similar values at $i$ and all neighboring vertices of $i$.

For notions of global smoothness, the discrete $p$-Dirichlet form of $\mathbf{f}$ is defined as

$$S_p(\mathbf{f}) := \frac{1}{p}\sum_{i \in V} \|\nabla_i \mathbf{f}\|_2^p = \frac{1}{p}\sum_{i \in V}\left[\sum_{j \in \mathcal{N}_i} W_{i,j}[f(j) - f(i)]^2\right]^{\frac{p}{2}}. \quad (5)$$

When $p = 1$, $S_1(\mathbf{f})$ is the total variation of the signal with respect to the graph. When $p = 2$, we have

$$\begin{aligned}S_2(\mathbf{f}) &= \frac{1}{2}\sum_{i \in V}\sum_{j \in \mathcal{N}_i} W_{i,j}[f(j) - f(i)]^2 \\ &= \sum_{(i,j) \in \mathcal{E}} W_{i,j}[f(j) - f(i)]^2 = \mathbf{f}^{\mathrm{T}}\mathcal{L}\mathbf{f}. \quad (6)\end{aligned}$$

$S_2(\mathbf{f})$ is known as the graph Laplacian quadratic form [17], and the seminorm $\|\mathbf{f}\|_{\mathcal{L}}$ is defined as

$$\|\mathbf{f}\|_{\mathcal{L}} := \|\mathcal{L}^{\frac{1}{2}}\mathbf{f}\|_2 = \sqrt{\mathbf{f}^{\mathrm{T}}\mathcal{L}\mathbf{f}} = \sqrt{S_2(\mathbf{f})}.$$

Note from (6) that the quadratic form $S_2(\mathbf{f})$ is equal to zero if and only if $\mathbf{f}$ is constant across all vertices (which is why $\|\mathbf{f}\|_{\mathcal{L}}$ is only a seminorm), and, more generally, $S_2(\mathbf{f})$ is small when the signal $\mathbf{f}$ has similar values at neighboring vertices

connected by an edge with a large weight, i.e., when it is smooth.

Returning to the graph Laplacian eigenvalues and eigenvectors, the Courant-Fischer Theorem [35, Theorem 4.2.11] tells us they can also be defined iteratively via the Rayleigh quotient as

$$\lambda_0 = \min_{\substack{f \in \mathbb{R}^N \\ \|\mathbf{f}\|_2 = 1}} \{\mathbf{f}^T \mathcal{L} \mathbf{f}\}, \tag{7}$$

$$\text{and } \lambda_\ell = \min_{\substack{f \in \mathbb{R}^N \\ \|\mathbf{f}\|_2 = 1 \\ \mathbf{f} \perp \text{span}\{\mathbf{u}_0, \ldots, \mathbf{u}_{\ell-1}\}}} \{\mathbf{f}^T \mathcal{L} \mathbf{f}\}, \quad \ell = 1, 2, \ldots, N-1, \tag{8}$$

where the eigenvector $\mathbf{u}_\ell$ is the minimizer of the $\ell$th problem. From (6) and (7), we see again why $\mathbf{u}_0$ is constant for connected graphs. Equation (8) explains why the graph Laplacian eigenvectors associated with lower eigenvalues are smoother, and provides another interpretation for why the graph Laplacian spectrum carries a notion of frequency.

In summary, the connectivity of the underlying graph is encoded in the graph Laplacian, which is used to define both a graph Fourier transform (via the graph Laplacian eigenvectors) and different notions of smoothness. "Example 1 (Importance of the Underlying Graph)" demonstrates how both the smoothness and the graph spectral content of a graph signal depend on the underlying graph.

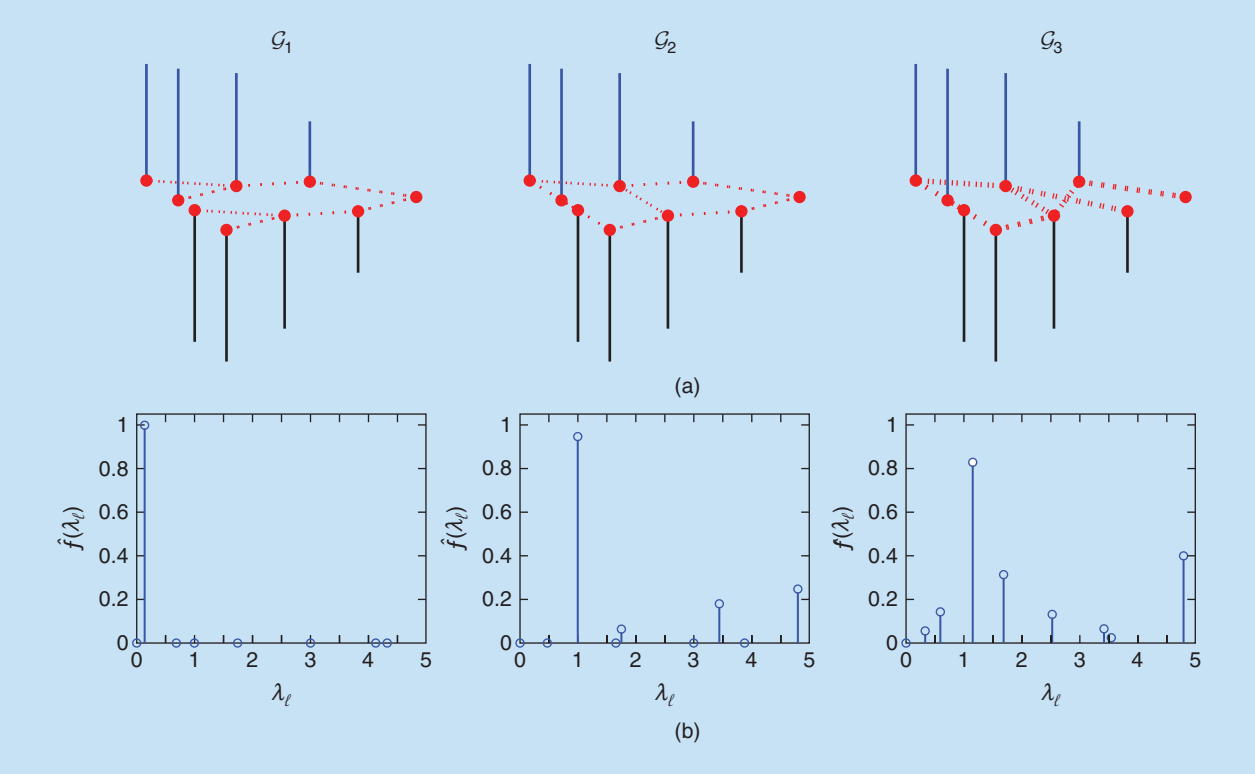
### *OTHER GRAPH MATRICES*

The basis $\{\mathbf{u}_\ell\}_{\ell=0,1,\ldots,N-1}$ of graph Laplacian eigenvectors is just one possible basis to use in the forward and inverse graph Fourier transforms (3) and (4). A second popular option is to normalize each weight $W_{i,j}$ by a factor of $1/\sqrt{d_i d_j}$. Doing so leads to the normalized graph Laplacian, which is defined as $\tilde{\mathcal{L}} := \mathbf{D}^{-1/2} \mathcal{L} \mathbf{D}^{-1/2}$, or, equivalently,

$$(\tilde{\mathcal{L}} f)(i) = \frac{1}{\sqrt{d_i}} \sum_{j \in \mathcal{N}_i} W_{i,j} \left[ \frac{f(i)}{\sqrt{d_i}} - \frac{f(j)}{\sqrt{d_j}} \right].$$

**EXAMPLE 1 (IMPORTANCE OF THE UNDERLYING GRAPH)**

In Figure S1, we plot the same signal $\mathbf{f}$ on three different unweighted graphs with the same set of vertices, but different edges. The smoothness and graph spectral content of the signal both depend on the underlying graph structure. In particular, the signal $\mathbf{f}$ is smoothest with respect to the intrinsic structure of $\mathcal{G}_1$, and least smooth with respect to the intrinsic structure of $\mathcal{G}_3$. This can be seen 1) visually; 2) through the Laplacian quadratic form, as $\mathbf{f}^T \mathcal{L}_1 \mathbf{f} = 0.14$, $\mathbf{f}^T \mathcal{L}_2 \mathbf{f} = 1.31$, and $\mathbf{f}^T \mathcal{L}_3 \mathbf{f} = 1.81$; and 3) through the graph spectral representations, where the signal has all of its energy in the low frequencies in the graph spectral plot of $\hat{\mathbf{f}}$ on $\mathcal{G}_1$, and more energy in the higher frequencies in the graph spectral plot of $\hat{\mathbf{f}}$ on $\mathcal{G}_3$.




**[FIGS1] The same graph signal plotted on three different unweighted graphs. (a) The signal in the vertex domains. (b) The signal in the respective graph spectral domains.**

The eigenvalues $\{\tilde{\lambda}_\ell\}_{\ell=0,1,\ldots,N-1}$ of the normalized graph Laplacian of a connected graph $\mathcal{G}$ satisfy

$$0 = \tilde{\lambda}_0 < \tilde{\lambda}_1 \leq \cdots \leq \tilde{\lambda}_{\max} \leq 2,$$

with $\tilde{\lambda}_{\max} = 2$ if and only if $\mathcal{G}$ is bipartite, i.e., the set of vertices $\mathcal{V}$ can be partitioned into two subsets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that every edge $e \in \mathcal{E}$ connects one vertex in $\mathcal{V}_1$ and one vertex in $\mathcal{V}_2$. We denote the normalized graph Laplacian eigenvectors by $\{\tilde{\mathbf{u}}_\ell\}_{\ell=0,1,\ldots,N-1}$. As seen in Figure 3(b), the spectrum of $\tilde{\mathcal{L}}$ also carries a notion of frequency, with the eigenvectors associated with higher eigenvalues generally having more zero crossings. However, unlike $\mathbf{u}_0$, the normalized graph Laplacian eigenvector $\tilde{\mathbf{u}}_0$ associated with the zero eigenvalue is not a constant vector.

The normalized and unnormalized graph Laplacians are both examples of generalized graph Laplacians [36, Sec. 1.6], also called discrete Schrödinger operators. A generalized graph Laplacian of a graph $\mathcal{G}$ is any symmetric matrix whose $i$, $j$th entry is negative if there is an edge connecting vertices $i$ and $j$, equal to zero if $i \neq j$ and $i$ is not connected to $j$, and may be anything if $i = j$.

A third popular matrix that is often used in dimensionality-reduction techniques for signals on graphs is the random walk matrix $\mathbf{P} := \mathbf{D}^{-1}\mathbf{W}$. Each entry $P_{i,j}$ describes the probability of going from vertex $i$ to vertex $j$ in one step of a Markov random walk on the graph $\mathcal{G}$. For connected, aperiodic graphs, each row of $\mathbf{P}^t$ converges to the stationary distribution of the random walk as $t$ goes to infinity. Closely related to the random walk matrix is the asymmetric graph Laplacian, which is defined as $\mathcal{L}_a := \mathbf{I}_N - \mathbf{P}$, where $\mathbf{I}_N$ is the $N \times N$ identity matrix. ($\mathcal{L}_a$ is not a generalized graph Laplacian due to its asymmetry.) Note that $\mathcal{L}_a$ has the same set of eigenvalues as $\tilde{\mathcal{L}}$, and if $\tilde{\mathbf{u}}_\ell$ is an eigenvector of $\tilde{L}$ associated with $\tilde{\lambda}_\ell$, then $\mathbf{D}^{-1/2}\tilde{\mathbf{u}}_\ell$ is an eigenvector of $\mathcal{L}_a$ associated with the eigenvalue $\tilde{\lambda}_\ell$.

As discussed in detail in the next section, both the normalized and unnormalized graph Laplacian eigenvectors can be used as filtering bases. There is not a clear answer as to when to use the normalized graph Laplacian eigenvectors, when to use the unnormalized graph Laplacian eigenvectors, and when to use some other basis. The normalized graph Laplacian has the nice properties that its spectrum is always contained in the interval $[0, 2]$ and, for bipartite graphs, the spectral folding phenomenon [37] can be exploited. However, the fact that the unnormalized graph Laplacian eigenvector associated with the zero eigenvalue is constant is a useful property in extending intuitions about the direct current (dc) components of signals from classical filtering theory.

## GENERALIZED OPERATORS FOR SIGNALS ON GRAPHS

In this section, we review different ways to generalize fundamental operations such as filtering, translation, modulation, dilation, and downsampling to the graph setting. These generalized operators are the ingredients used to develop the localized, multiscale transforms described in a later section.

### *FILTERING*

The first generalized operation we tackle is filtering. We start by extending the notion of frequency filtering to the graph setting, and then discuss localized filtering in the vertex domain.

#### FREQUENCY FILTERING

In classical signal processing, frequency filtering is the process of representing an input signal as a linear combination of complex exponentials and amplifying or attenuating the contributions of some of the component complex exponentials

$$\hat{f}_{\text{out}}(\xi) = \hat{f}_{\text{in}}(\xi)\hat{h}(\xi), \tag{9}$$

where $\hat{h}(\cdot)$ is the transfer function of the filter. Taking an inverse Fourier transform of (9), multiplication in the Fourier domain corresponds to convolution in the time domain

$$f_{\text{out}}(t) = \int_{\mathbb{R}} \hat{f}_{\text{in}}(\xi)\hat{h}(\xi)e^{2\pi i\xi t}d\xi \tag{10}$$

$$= \int_{\mathbb{R}} f_{\text{in}}(\tau)h(t-\tau)d\tau =: (f_{\text{in}} * h)(t). \tag{11}$$

Once we fix a graph spectral representation, and thus our notion of a graph Fourier transform (in this section, we use the eigenvectors of $\mathcal{L}$, but $\tilde{\mathcal{L}}$ can also be used), we can directly generalize (9) to define frequency filtering, or graph spectral filtering, as

$$\hat{f}_{\text{out}}(\lambda_\ell) = \hat{f}_{\text{in}}(\lambda_\ell)\hat{h}(\lambda_\ell), \tag{12}$$

or, equivalently, taking an inverse graph Fourier transform,

$$f_{\text{out}}(i) = \sum_{\ell=0}^{N-1} \hat{f}_{\text{in}}(\lambda_\ell)\hat{h}(\lambda_\ell)u_\ell(i). \tag{13}$$

Borrowing notation from the theory of matrix functions [38], we can also write (12) and (13) as $\mathbf{f}_{\text{out}} = \hat{h}(\mathcal{L})\mathbf{f}_{\text{in}}$, where

$$\hat{h}(\mathcal{L}) := \mathbf{U}\begin{bmatrix} \hat{h}(\lambda_0) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \hat{h}(\lambda_{N-1}) \end{bmatrix}\mathbf{U}^*. \tag{14}$$

The basic graph spectral filtering (12) can be used to implement discrete versions of well-known continuous filtering techniques such as Gaussian smoothing, bilateral filtering, total variation filtering, anisotropic diffusion, and nonlocal means filtering (see, e.g., [39] and references therein). In particular, many of these filters arise as solutions to variational problems to regularize ill-posed inverse problems such as denoising, inpainting, and super resolution. One example is the discrete regularization framework

$$\min_{\mathbf{f}} \{\|\mathbf{f} - \mathbf{y}\|_2^2 + \gamma S_p(\mathbf{f})\}, \tag{15}$$

where $S_p(\mathbf{f})$ is the $p$-Dirichlet form of (5). References [4]–[11], [14, Ch. 5], and [28]–[30] discuss (15) and other energy minimization models in detail, as well as specific filters that arise

as solutions, relations between these discrete graph spectral filters and filters arising out of continuous partial differential equations, and applications such as graph-based image processing, mesh smoothing, and statistical learning. In "Example 2 (Tikhonov Regularization)," we show one particular image denoising application of (15) with $p = 2$.

## FILTERING IN THE VERTEX DOMAIN

To filter a signal in the vertex domain, we simply write the output $f_{\text{out}}(i)$ at vertex $i$ as a linear combination of the components of the input signal at vertices within a $K$-hop local neighborhood of vertex $i$

$$f_{\text{out}}(i) = b_{i,i} f_{\text{in}}(i) + \sum_{j \in \mathcal{N}(i,K)} b_{i,j} f_{\text{in}}(j), \tag{16}$$

for some constants $\{b_{i,j}\}_{i,j \in \mathcal{V}}$. Equation (16) just says that filtering in the vertex domain is a localized linear transform.

We now briefly relate filtering in the graph spectral domain (frequency filtering) to filtering in the vertex domain. When the frequency filter in (12) is an order $K$ polynomial $\hat{h}(\lambda_\ell) = \sum_{k=0}^{K} a_k \lambda_\ell^k$ for some constants $\{a_k\}_{k=0,1\ldots,K}$, we can also interpret the filtering equation (12) in the vertex domain. From (13), we have

$$\begin{aligned} f_{\text{out}}(i) &= \sum_{\ell=0}^{N-1} \hat{f}_{\text{in}}(\lambda_\ell) \hat{h}(\lambda_\ell) u_\ell(i) \\ &= \sum_{j=1}^{N} f_{\text{in}}(j) \sum_{k=0}^{K} a_k \sum_{\ell=0}^{N-1} \lambda_\ell^k u_\ell^*(j) u_\ell(i) \\ &= \sum_{j=1}^{N} f_{\text{in}}(j) \sum_{k=0}^{K} a_k (\mathcal{L}^k)_{i,j}. \end{aligned} \tag{17}$$

**EXAMPLE 2 (TIKHONOV REGULARIZATION)**

We observe a noisy graph signal $\mathbf{y} = \mathbf{f}_0 + \boldsymbol{\eta}$, where $\boldsymbol{\eta}$ is uncorrelated additive Gaussian noise, and we wish to recover $\mathbf{f}_0$. To enforce a priori information that the clean signal $\mathbf{f}_0$ is smooth with respect to the underlying graph, we include a regularization term of the form $\mathbf{f}^{\mathsf{T}} \mathcal{L} \mathbf{f}$, and, for a fixed $\gamma > 0$, solve the optimization problem

$$\operatorname*{argmin}_{\mathbf{f}} \{ \|\mathbf{f} - \mathbf{y}\|_2^2 + \gamma \mathbf{f}^{\mathsf{T}} \mathcal{L} \mathbf{f} \}. \tag{S1}$$

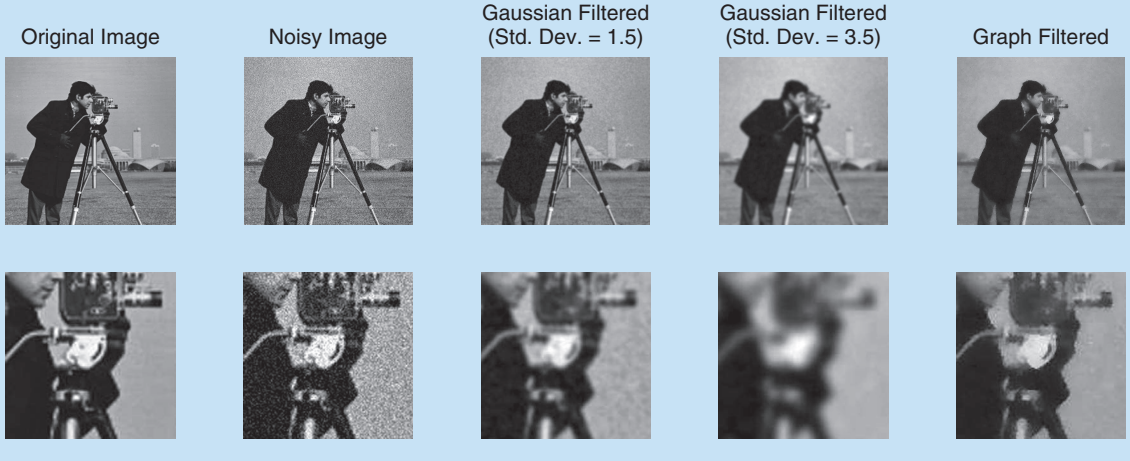
The first-order optimality conditions of the convex objective function in (S1) show that (see, e.g., [4], [29, Sec. III-A], and [40, Prop. 1]) the optimal reconstruction is given by

$$f_*(i) = \sum_{\ell=0}^{N-1} \left[ \frac{1}{1 + \gamma \lambda_\ell} \right] \hat{y}(\lambda_\ell) u_\ell(i), \tag{S2}$$

or, equivalently, $\mathbf{f} = \hat{h}(\mathcal{L})\mathbf{y}$, where $\hat{h}(\lambda) := 1/(1 + \gamma\lambda)$ can be viewed as a low-pass filter.

As an example, in Figure S2, we take the $512 \times 512$ cameraman image as $\mathbf{f}_0$ and corrupt it with additive Gaussian noise with mean zero and standard deviation 0.1 to get a noisy signal $\mathbf{y}$. We then apply two different filtering methods to denoise the signal. In the first method, we apply a symmetric two-dimensional Gaussian low-pass filter of size $72 \times 72$ with two different standard deviations: 1.5 and 3.5. In the second method, we form a semilocal graph on the pixels by connecting each pixel to its horizontal, vertical, and diagonal neighbors, and setting the Gaussian weights (1) between two neighboring pixels according to the similarity of the noisy image values at those two pixels; i.e., the edges of the semilocal graph are independent of the noisy image, but the distances in (1) are the differences between the neighboring pixel values in the noisy image. For the Gaussian weights in (1), we take $\theta = 0.1$ and $\kappa = 0$. We then perform the low-pass graph filtering (S2) with $\gamma = 10$ to reconstruct the image. This method is a variant of the graph-based anisotropic diffusion image smoothing method of [11].

In all image displays in Figure S2, we threshold the values to the [0,1] interval. The images in (b) comprise zoomed-in versions of the images in (a). Comparing the results of the two filtering methods, we see that to smooth sufficiently in smoother areas of the image, the classical Gaussian filter also smooths across the image edges. The graph spectral filtering method does not smooth as much across the image edges, as the geometric structure of the image is encoded in the graph Laplacian via the noisy image.




**[FIGS2] Image denoising via classical Gaussian filtering and graph spectral filtering.**

Yet, $(\mathcal{L}^k)_{i,j} = 0$ when the shortest-path distance $d_{\mathcal{G}}(i, j)$ between vertices $i$ and $j$ (i.e., the minimum number of edges comprising any path connecting $i$ and $j$) is greater than $k$ [41, Lemma 5.2]. Therefore, we can write (17) exactly as in (16), with the constants defined as

$$b_{i,j} := \sum_{k=d_{\mathcal{G}}(i,j)}^{K} a_k (\mathcal{L}^k)_{i,j}.$$

So when the frequency filter is an order $K$ polynomial, the frequency filtered signal at vertex $i$, $f_{\text{out}}(i)$, is a linear combination of the components of the input signal at vertices within a $K$-hop local neighborhood of vertex $i$. This property can be quite useful when relating the smoothness of a filtering kernel to the localization of filtered signals in the vertex domain.

### CONVOLUTION

We cannot directly generalize the definition (11) of a convolution product to the graph setting, because of the term $h(t-\tau)$. However, one way to define a generalized convolution product for signals on graphs is to replace the complex exponentials in (10) with the graph Laplacian eigenvectors [42]

$$(f * h)(i) := \sum_{\ell=0}^{N-1} \hat{f}(\lambda_\ell)\hat{h}(\lambda_\ell)u_\ell(i), \tag{18}$$

which enforces the property that convolution in the vertex domain is equivalent to multiplication in the graph spectral domain.

### TRANSLATION

The classical translation operator is defined through the change of variable $(T_\upsilon f)(t) := f(t-\upsilon)$, which, as discussed earlier, we cannot directly generalize to the graph setting. However, we can also view the classical translation operator $T_\upsilon$ as a convolution with a delta centered at $\upsilon$, i.e., $(T_\upsilon f)(t) = (f * \delta_\upsilon)(t)$ in the weak sense. Thus, one way to define a generalized translation operator $T_n : \mathbb{R}^N \to \mathbb{R}^N$ is via generalized convolution with a delta centered at vertex $n$ [41], [42]:

$$(T_n g)(i) := \sqrt{N}\,(g * \delta_n)(i) \overset{(20)}{=} \sqrt{N} \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i), \tag{19}$$

where

$$\delta_n(i) = \begin{cases} 1 & \text{if } i = n \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

A few remarks about the generalized translation (19) are in order. First, we do not usually view it as translating a signal $\mathbf{g}$ defined in the vertex domain, but rather as a kernelized operator acting on a kernel $\hat{g}(\cdot)$ defined directly in the graph spectral domain. To translate this kernel to vertex $n$, the $\ell$th component of the kernel is multiplied by $u_\ell^*(n)$, and then an inverse graph Fourier transform is applied. Second, the normalizing constant $\sqrt{N}$ in (19) ensures that the translation operator preserves the mean of a signal, i.e., $\sum_{i=1}^N (T_n g)(i) = \sum_{i=1}^N g(i)$. Third, the smoothness of the kernel $\hat{g}(\cdot)$ controls the localization of $T_n\mathbf{g}$ around the center vertex $n$; that is, the magnitude $(T_n g)(i)$ of the translated kernel at vertex $i$ decays as the distance between $i$ and $n$ increases [41]. This property can be seen in Figure 5, where we translate a heat kernel around to different locations of the Minnesota graph. Finally, unlike the classical translation operator, the generalized translation operator (19) is not generally an isometric operator ($\|T_n\mathbf{g}\|_2 \neq \|\mathbf{g}\|_2$), due to the possible localization of the graph Laplacian eigenvectors ($\mu > 1/\sqrt{N}$).

### MODULATION AND DILATION

In addition to translation, many classical transform methods rely on modulation or dilation to localize signals' frequency content. The classical modulation operator

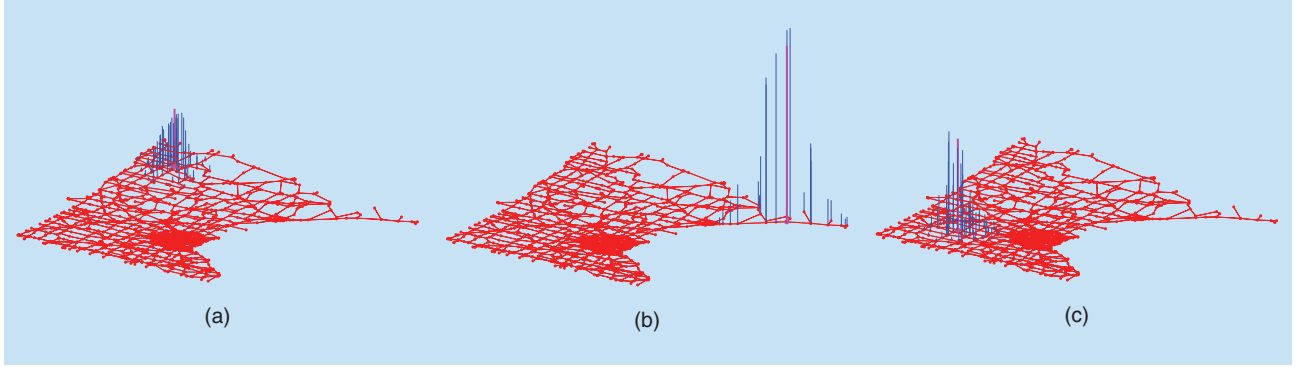
$$(M_\omega f)(t) := e^{2\pi i \omega t} f(t) \tag{21}$$

represents a translation in the Fourier domain:

$$\widehat{M_\omega f}(\xi) = \hat{f}(\xi - \omega), \quad \forall \xi \in \mathbb{R}.$$

One way to define generalized modulation in the graph setting is to replace the multiplication by a complex exponential (an eigenfunction of the 1-D Laplacian operator) in (21) with a multiplication by a graph Laplacian eigenvector:

$$(M_k g)(i) := \sqrt{N}\, u_k(i) g(i). \tag{22}$$




**[FIG5] The translated signals (a) $T_{100}\mathbf{g}$, (b) $T_{200}\mathbf{g}$, and (c) $T_{2,000}\mathbf{g}$, where $\mathbf{g}$ is the heat kernel shown in Figure 4(a) and (b).**

The generalized modulation (22) is not exactly a translation in the graph spectral domain due to the discrete and irregular nature of the spectrum; however, as shown in [42, Fig. 3], if a kernel $\hat{g}(\cdot)$ is localized around 0 in the graph spectral domain, then $\widehat{M_k \mathbf{g}}$ is localized around $\lambda_k$.

For $s > 0$, dilation or scaling of an analog signal $f$ in the time domain is given by

$$(\mathcal{D}_s f)(t) := \frac{1}{s} f\left(\frac{t}{s}\right). \tag{23}$$

We cannot directly generalize (23) to the graph setting, because $i/s$ is not likely to be in the domain $\mathcal{V}$ for all $i \in \mathcal{V}$. Instead, we can take the Fourier transform of (23)

$$\widehat{(\mathcal{D}_s f)}(\xi) = \hat{f}(s\xi), \tag{24}$$

and generalize (24) to the graph setting. Assuming we start with a kernel $\hat{g} : \mathbb{R}_+ \to \mathbb{R}$, we can define a generalized graph dilation by [41]

$$\widehat{(\mathcal{D}_s g)}(\lambda) := \hat{g}(s\lambda). \tag{25}$$

Note that, unlike the generalized modulation (22), the generalized dilation (25) requires the kernel $\hat{g}(\cdot)$ to be defined on the entire real line, not just on $\sigma(\mathcal{L})$ or $[0, \lambda_{\max}]$.

### *GRAPH COARSENING, DOWNSAMPLING, AND REDUCTION*

Many multiscale transforms for signals on graphs require successively coarser versions of the original graph that preserve properties of the original graph such as the intrinsic geometric structure (e.g., some notion of distance between vertices), connectivity, graph spectral distribution, and sparsity. The process of transforming a given (fine scale) graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ into a coarser graph $\mathcal{G}^{\text{reduced}} = \{\mathcal{V}^{\text{reduced}}, \mathcal{E}^{\text{reduced}}, \mathbf{W}^{\text{reduced}}\}$ with fewer vertices and edges, while also preserving the aforementioned properties, is often referred to as graph coarsening or coarse-graining [46].

This process can be split into two separate but closely related subtasks: 1) identifying a reduced set of vertices $\mathcal{V}^{\text{reduced}}$, and 2) assigning edges and weights, $\mathcal{E}^{\text{reduced}}$ and $\mathbf{W}^{\text{reduced}}$, to connect the new set of vertices. When an additional constraint that $\mathcal{V}^{\text{reduced}} \subset \mathcal{V}$ is imposed, the first subtask is often referred to as graph downsampling. The second subtask is often referred to as graph reduction or graph contraction.

In the special case of a bipartite graph, two subsets can be chosen so that every edge connects vertices in two different subsets. Thus, for bipartite graphs, there is a natural way to downsample by a factor of two, as there exists a notion of "every other vertex."

For nonbipartite graphs, the situation is far more complex, and a wide range of interesting techniques for the graph coarsening problem have been proposed by graph theorists, and, in particular, by the numerical linear algebra community. To mention just a few, Lafon and Lee [46] downsample based on diffusion distances and form new edge weights based on random walk transition probabilities; the greedy seed selection algorithm of Ron et al. [47] leverages an algebraic distance measure to downsample the vertices; recursive spectral bisection [48] repeatedly divides the graph into parts according to the polarity (signs) of the Fiedler vectors $\mathbf{u}_1$ of successive subgraphs; Narang and Ortega [49] minimize the number of edges connecting two vertices in the same downsampled subset; and another generally applicable method that yields the natural downsampling on bipartite graphs ([36, Ch. 3.6]) is to partition $\mathcal{V}$ into two subsets according to the polarity of the components of the graph Laplacian eigenvector $\mathbf{u}_{N-1}$ associated with the largest eigenvalue $\lambda_{\max}$. We refer readers to [47] and [50] and references therein for more thorough reviews of the graph coarsening literature.

There are also many interesting connections between graph coarsening, graph coloring [51], spectral clustering [22], and nodal domain theory [36, Ch. 3]. Finally, in a closely related topic, Pesenson (e.g., [52]) has extended the concept of bandlimited sampling to signals defined on graphs by showing that certain classes of signals can be downsampled on particular subgraphs and then stably reconstructed from the reduced set of samples.

## LOCALIZED, MULTISCALE TRANSFORMS FOR SIGNALS ON GRAPHS

The increasing prevalence of signals on graphs has triggered a recent influx of localized transform methods specifically designed to analyze data on graphs. These include wavelets on unweighted graphs for analyzing computer network traffic [53], diffusion wavelets and diffusion wavelet packets [24], [44], [45], the "top-down" wavelet construction of [54], graph dependent basis functions for sensor network graphs [55], lifting based wavelets on graphs [49], [56], multiscale wavelets on balanced trees [57], spectral graph wavelets [41], critically sampled two-channel wavelet filter banks [37], [58], and a windowed graph Fourier transform [42].

Most of these designs are generalizations of the classical wavelet filter banks used to analyze signals on Euclidean domains. The feature that makes the classical wavelet transforms so useful is their ability to simultaneously localize signal information in both time (or space) and frequency, and thus exploit the time-frequency resolution tradeoff better than the Fourier transform. In a similar vein, the desired property of wavelet transforms on graphs is to localize graph signal contents in both the vertex and graph spectral domains. In the classical setting, locality is measured in terms of the "spread" of the signal in time and frequency, and uncertainty principles (see [59, Sec. 2.6.2]) describe the tradeoff between time and frequency resolution. Whether such a tradeoff exists for graph signals remains an open question. However, some recent works have begun to define different ways to measure the "spread" of graph signals in both domains. For example, [60] defines the spatial spread of any signal $\mathbf{f}$ around a center vertex $i$ on a graph $\mathcal{G}$ as

$$\Delta^2_{\mathcal{G},i}(\mathbf{f}) := \frac{1}{\|\mathbf{f}\|_2^2} \sum_{j \in \mathcal{V}} [d_{\mathcal{G}}(i, j)]^2 [f(j)]^2. \tag{26}$$

Here, $\{[f(j)]^2 / \|\mathbf{f}\|^2\}_{j=1,2,\ldots,N}$ can be interpreted as a probability mass function (pmf) of signal $\mathbf{f}$, and $\Delta^2_{\mathcal{G},i}(\mathbf{f})$ is the variance of the

geodesic distance function $d_{\mathcal{G}}(i,.)\colon \mathcal{V} \to \mathbb{R}$ at node $i$, in terms of this spatial pmf. The spatial spread of a graph signal can then be defined as

$$\Delta_{\mathcal{G}}^2(\mathbf{f}) := \min_{i \in \mathcal{V}} \{\Delta_{\mathcal{G},i}^2(\mathbf{f})\}.$$

Similarly, the spectral spread of a graph signal can be defined as

$$\Delta_{\sigma}^2(\mathbf{f}) := \min_{\mu \in \mathbb{R}_+} \left\{ \frac{1}{\|\mathbf{f}\|_2^2} \sum_{\lambda \in \sigma(\mathcal{L})} [\sqrt{\lambda} - \sqrt{\mu}]^2 [\hat{f}(\lambda)]^2 \right\}, \tag{27}$$

where $\{[\hat{f}(\lambda)]^2/\|\mathbf{f}\|_2^2\}_{\lambda = \lambda_0, \lambda_1, \ldots, \lambda_{\max}}$ is the pmf of $\mathbf{f}$ across the spectrum of the Laplacian matrix, and $\sqrt{\mu}$ and $\Delta_{\sigma}^2(\mathbf{f})$ are the mean and variance of $\sqrt{\lambda}$, respectively, in the distribution given by this spectral pmf. Note that the definitions of spread presented here are heuristically defined and do not have a well-understood theoretical background. If the graph is not regular, the choice of which Laplacian matrix ($\mathcal{L}$ or $\tilde{\mathcal{L}}$) to use for computing spectral spreads also affects the results. The purpose of these definitions and the subsequent examples is to show that a tradeoff exists between spatial and spectral localization in graph wavelets. If we do not minimize over all $\mu$ but rather fix $\mu = 0$ and also use the normalized graph Laplacian matrix $\tilde{\mathcal{L}}$ instead of $\mathcal{L}$, the definition of spectral spread in (27) reduces to the one proposed in [60].

Depending on the application under consideration, other desirable features of a graph wavelet transform may include perfect reconstruction, critical sampling, orthogonal expansion, and a multiresolution decomposition [37].

In the remainder of this section, we categorize the existing graph transform designs and provide simple examples. The graph wavelet transform designs can broadly be divided into two types: vertex domain designs and graph spectral domain designs.

### VERTEX DOMAIN DESIGNS

The vertex domain designs of graph wavelet transforms are based on the spatial features of the graph, such as node connectivity and distances between vertices. Most of these localized transforms can be viewed as particular instances of filtering in the vertex domain, as in (16), where the output at each node can be computed from the samples within some $K$-hop neighborhood around the node. The graph spectral properties of these transforms are not explicitly designed. Examples of vertex domain designs include random transforms [55], graph wavelets [53], lifting-based wavelets [49], [61], [62], and tree wavelets [57].

The random transforms [55] for unweighted graphs compute either a weighted average or a weighted difference at each node in the graph with respect to a $k$-hop neighborhood around it. Thus, the filter at each node has a constant, nonzero weight $c$ within the $k$-hop neighborhood and zero weight outside, where the parameter $c$ is chosen so as to guarantee invertibility of the transform.

The graph wavelets of Crovella and Kolaczyk [53] are functions $\psi_{k,i}\colon \mathcal{V} \to \mathbb{R}$, localized with respect to a range of scale/location indices $(k, i)$, which at a minimum satisfy $\sum_{j \in \mathcal{V}} \psi_{k,i}(j) = 0$ (i.e., a zero dc response). This graph wavelet transform is described in more detail later in the section.

Lifting-based transforms for graphs [49], [61], [62] are extensions of the lifting wavelets originally proposed for 1-D signals by Sweldens [63]. In this approach, the vertex set is first partitioned into sets of even and odd nodes, $\mathcal{V} = \mathcal{V}_{\mathcal{O}} \bigcup \mathcal{V}\varepsilon$. Each odd node computes its prediction coefficient using its own data and data from its even neighbors. Then each even node computes its update coefficients using its own data and the prediction coefficients of its neighboring odd nodes.

In [57], Gavish et al. construct tree wavelets by building a balanced hierarchical tree from the data defined on graphs, and then generating orthonormal bases for the partitions defined at each level of the tree using a modified version of the standard one-dimensional wavelet filtering and decimation scheme.

### GRAPH SPECTRAL DOMAIN DESIGNS

The graph spectral domain designs of graph wavelets are based on the spectral features of the graph, which are encoded, e.g., in the eigenvalues and eigenvectors of one of the graph matrices defined in the section "The Graph Spectral Domains." Notable examples in this category include diffusion wavelets [24], [44], spectral graph wavelets [41], and graph quadrature mirror filterbanks (graph-QMF) [37]. The general idea of the graph spectral designs is to construct bases that are localized in both the vertex and graph spectral domains.

The diffusion wavelets [24], [44], for example, are based on compressed representations of powers of a diffusion operator, such as the one discussed in "Example 3 (Diffusion Operators and Dilation)." The localized basis functions at each resolution level are downsampled and then orthogonalized through a variation of the Gram-Schmidt orthogonalization scheme.

The spectral graph wavelets of [41] are dilated, translated versions of a bandpass kernel designed in the graph spectral domain of the unnormalized graph Laplacian $\mathcal{L}$. They are discussed further in the next section.

Another graph spectral design is the two-channel graph-QMF filterbank proposed for bipartite graphs in [37]. The resulting transform is orthogonal and critically sampled and also yields perfect reconstruction. In this design, the analysis and synthesis filters at each scale are designed using a single prototype transfer function $\hat{h}(\tilde{\lambda})$, which satisfies

$$\hat{h}^2(\tilde{\lambda}) + \hat{h}^2(2 - \tilde{\lambda}) = 2, \tag{28}$$

where $\tilde{\lambda}$ is an eigenvalue in the normalized graph Laplacian spectrum. The design extends to any arbitrary graph via a bipartite subgraph decomposition.

### EXAMPLES OF GRAPH WAVELET DESIGNS

To build more intuition about graph wavelets, we present some examples using one vertex domain design and one graph spectral domain design.

For the vertex domain design, we use the graph wavelet transform of Crovella and Kolaczyk (CKWT) [53] as an example. These wavelets are based on the geodesic or shortest-path

**EXAMPLE 3 (DIFFUSION OPERATORS AND DILATION)**

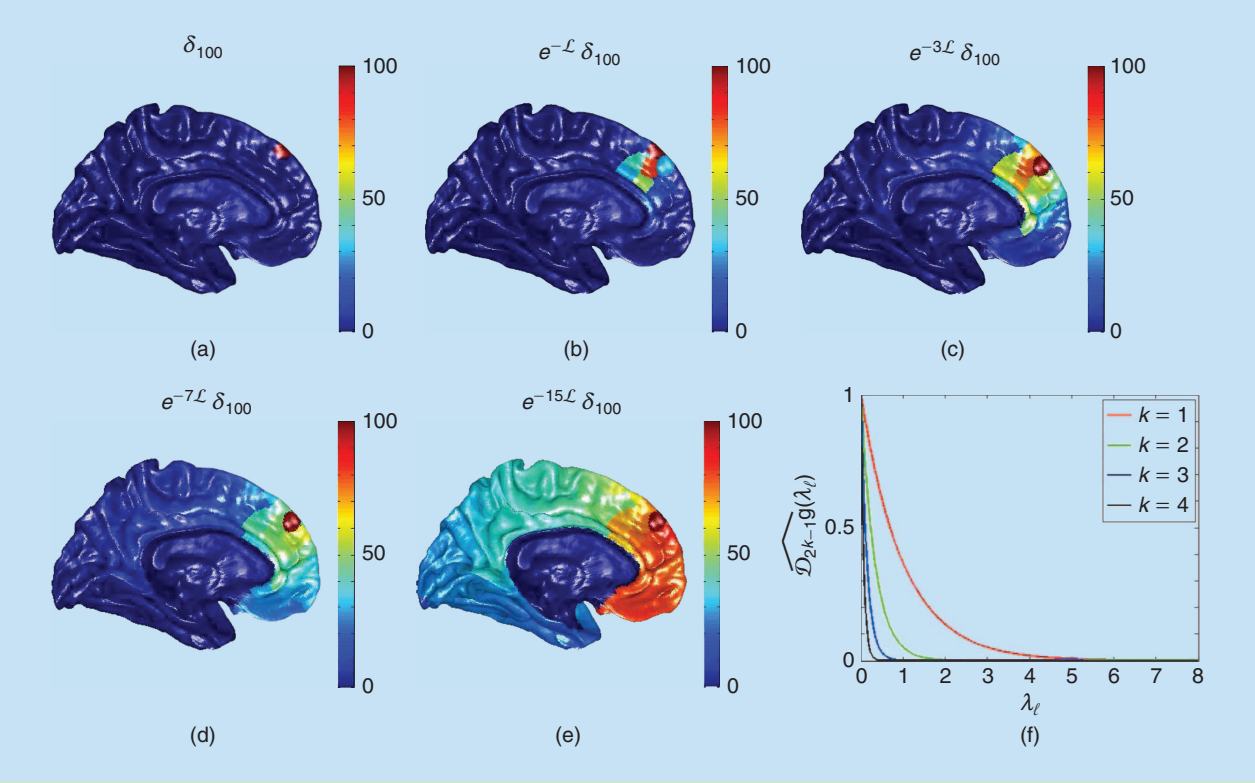
The heat diffusion operator $\mathbf{R} = e^{-\mathcal{L}}$ is an example of a discrete diffusion operator (see, e.g., [43] and [14, Sec. 2.5.5] for general discussions of discrete diffusions and [24, Sec. 4.1] for a formal definition and examples of symmetric diffusion semigroups). Intuitively, applying different powers $\tau$ of the heat diffusion operator to a signal $f$ describes the flow of heat over the graph when the rates of flow are proportional to the edge weights encoded in $\mathcal{L}$. The signal $f$ represents the initial amount of heat at each vertex, and $\mathbf{R}^{\tau}\mathbf{f} = (e^{-\tau\mathcal{L}})\,\mathbf{f}$ represents the amount of heat at each vertex after time $\tau$. The time variable $\tau$ also provides a notion of scale. When $\tau$ is small, the entry $(e^{-\tau\mathcal{L}})_{i,j}$ for two vertices that are far apart in the graph is very small, and therefore $((e^{-\tau\mathcal{L}})\,\mathbf{f})\,(i)$ depends primarily on the values $f(j)$ for vertices $j$ close to $i$ in the graph. As $\tau$ increases, $((e^{-\tau\mathcal{L}})\,\mathbf{f})\,(i)$ also depends on the values $f(j)$ for vertices $j$ farther away from $i$ in the graph. Zhang and Hancock [11] provide a more detailed mathematical justification behind this migration from domination of the local geometric structures to domination of the global structure of the graph as $\tau$ increases, as well as a nice illustration of heat diffusion on a graph in [11, Fig. 1].

Using our notations from (14) and (25), we can see that applying a power $\tau$ of the heat diffusion operator to any signal $\mathbf{f} \in \mathbb{R}^N$ is equivalent to filtering the signal with a dilated heat kernel

$$\mathbf{R}^{\tau}\mathbf{f} = (e^{-\tau\mathcal{L}})\,\mathbf{f} = \widehat{(\mathcal{D}_{\tau}g)}(\mathcal{L})\mathbf{f} = \mathbf{f} * (\mathcal{D}_{\tau}g),$$

where the filter is the heat kernel $\hat{g}(\lambda_\ell) = e^{-\lambda_\ell}$, similar to the one shown in Figure 4(b).

In Figure S3, we consider the cerebral cortex graph described in [41], initialize a unit of energy at the vertex 100 by taking $\mathbf{f} = \delta_{100}$, allow it to diffuse through the network for different dyadic amounts of time, and measure the amount of energy that accumulates at each vertex. Note that dyadic powers of diffusion operators of the form $\{\mathbf{R}^{2^{k-1}}\}_{k=1,2,\ldots}$ are of central importance to diffusion wavelets and diffusion wavelet packets [24], [44], [45], which we discuss in the section "Localized, Multiscale Transforms for Signals on Graphs."




**[FIGS3] Applying different powers of the heat diffusion operator can be interpreted as graph spectral filtering with a dilated kernel. (a) Original signal $\mathbf{f} = \delta_{100}$ on the cerebral cortex graph. Parts (b)–(e) show filtered signals $\{\mathbf{f} * (\mathcal{D}_{2^{k-1}}\mathbf{g})\}_{k=1,2,3,4} = \{\mathbf{R}^{2^{k-1}}\mathbf{f}\}_{k=1,2,3,4}$. (f) Different dilated kernels corresponding to the dyadic powers of the diffusion operator.**

distance $d_{\mathcal{G}}(i, j)$. Define $\partial\mathcal{N}(i, \tau)$ to be the set of all vertices $j \in \mathcal{N}$ such that $d_{\mathcal{G}}(i, j) = \tau$. Then the wavelet function $\psi_{k,i}^{CKWT}: \mathcal{V} \to \mathbb{R}$ at scale $k$ and center vertex $i \in \mathcal{V}$ can be written as

$$\psi_{k,i}^{\mathrm{CKWT}}(j) = \frac{a_{k,\tau}}{|\partial\mathcal{N}(i,\tau)|}, \quad \forall j \in \partial\mathcal{N}(i,\tau), \tag{29}$$

for some constants $\{a_{k,\tau}\}_{\tau=0,1,\ldots,k}$. Thus, each wavelet is constant across all vertices $j \in \partial\mathcal{N}(i, \tau)$ that are the same distance from the center vertex $i$, and the value of the wavelet at the vertices in $\partial\mathcal{N}(i, \tau)$ depends on the distance $\tau$. If $\tau > k$, $a_{k,\tau} = 0$, so that for any $k$, the function $\psi_{k,i}^{\mathrm{CKWT}}$ is exactly supported on a $k$-hop localized neighborhood around the center vertex $i$. The

constants $a_{k,\tau}$ in (29) also satisfy $\sum_{\tau=0}^{k} a_{k,\tau} = 0$, and can be computed from any continuous wavelet function $\psi^{[0,1)}(\cdot)$ supported on the interval $[0,1)$ by taking $a_{k,\tau}$ to be the average of $\psi^{[0,1)}(\cdot)$ on the sub-intervals $I_{k,\tau} = [\tau/(k+1), (\tau+1)/(k+1)]$. In our examples in Figures 6 and 7, we take $\psi^{[0,1)}(\cdot)$ to be the continuous Mexican hat wavelet. We denote the entire graph wavelet transform at a given scale $k$ as $\Psi_k^{\mathrm{CKWT}} := [\psi_{k,1}^{\mathrm{CKWT}}, \psi_{k,2}^{\mathrm{CKWT}}, \ldots \psi_{k,N}^{\mathrm{CKWT}}]$.

For the graph spectral domain design, we use the spectral graph wavelet transform (SGWT) of [41] as an example. The SGWT consists of one scaling function centered at each vertex, and $K$ wavelets centered at each vertex, at scales $\{t_1, t_2, \ldots, t_K\} \in \mathbb{R}_+$. The scaling functions are translated low-pass kernels

$$\psi_{\mathrm{scal},i}^{\mathrm{SGWT}} := T_i \mathrm{h} = \hat{h}(\mathcal{L})\delta_i,$$

where the generalized translation $T_i$ is defined in (19), and the kernel $\hat{h}(\lambda)$ is a low-pass filter. The wavelet at scale $t_k$ and center vertex $i$ is defined as

$$\psi_{t_k,i}^{\mathrm{SGWT}} := T_i \mathcal{D}_{t_k} \mathrm{g} = \widehat{\mathcal{D}_{t_k} g}(\mathcal{L})\delta_i,$$

where the generalized dilation $\mathcal{D}_{t_k}$ is defined in (25), and $\hat{g}(\lambda)$ is a band-pass kernel satisfying $\hat{g}(0) = 0$, $\lim_{\lambda\to\infty} \hat{g}(\lambda) = 0$, and an admissibility condition [41]. We denote the SGWT transform at scale $t_k$ as

$$\Psi_{t_k}^{\mathrm{SGWT}} = [\psi_{t_k,1}^{\mathrm{SGWT}}, \psi_{t_k,2}^{\mathrm{SGWT}}, \ldots \psi_{t_k,N}^{\mathrm{SGWT}}],$$

so that the entire transform $\Psi^{\mathrm{SGWT}} : \mathbb{R}^N \to \mathbb{R}^{N(K+1)}$ is given by

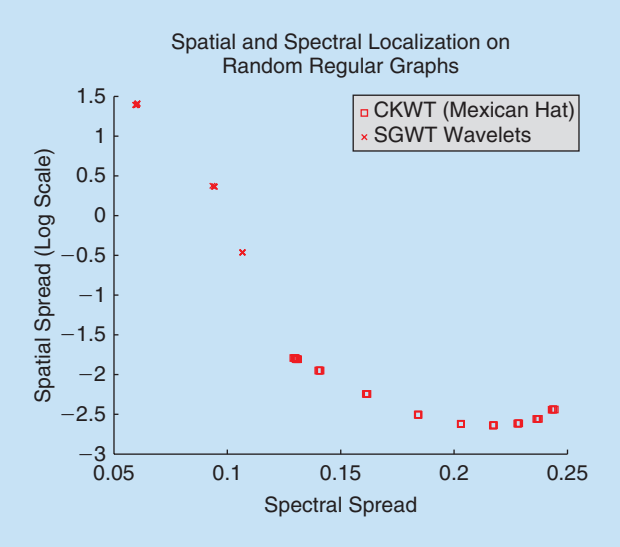
$$\Psi^{\mathrm{SGWT}} = [\Psi_{\mathrm{scal}}^{\mathrm{SGWT}}; \Psi_{t_1}^{\mathrm{SGWT}}; \ldots; \Psi_{t_K}^{\mathrm{SGWT}}].$$

We now compute the spatial and spectral spreads of the two graph wavelet transforms presented above. Unlike in the classical setting, the basis functions in a graph wavelet transform are not space-invariant, i.e., the spreads of two wavelets $\psi_{k,i_1}$ and $\psi_{k,i_2}$ at the same scale are not necessarily the same. Therefore, the spatial spread of a graph transform cannot be measured by computing the spreads of only one wavelet. In our analysis, we compute the spatial spread of a transform $\Psi_k$ at a given scale $k$ by taking an average over all scale $k$ wavelet (or scaling) functions of the spatial spreads (26) around each respective center vertex $i$. Therefore, we first compute

$$|\hat{\Psi}_k(\lambda)|^2 := \frac{1}{N}\sum_{i=1}^{N} |\hat{\psi}_{k,i}(\lambda)|^2, \tag{30}$$

and then take $|\hat{f}(\lambda)|^2 = |\hat{\Psi}_k(\lambda)|^2$ in (27) to compute the average spectral spread of $\Psi_k$.

The spatial and spectral spreads of both the CKWT and SGWT at different scales are shown in Figure 6. The graphs used in this example are random $d$-regular graphs. Observe that in Figure 6, the CKWT wavelets are located to the right of the SGWT wavelets on the horizontal (spectral) axis, and below them on the vertical (spatial) axis, which implies that, in this example, the CKWT wavelets are less localized spectrally and more localized spatially than the SGWT wavelets. This analysis provides an empirical understanding of the tradeoff between the spatial and spectral resolutions of signals defined on graphs.

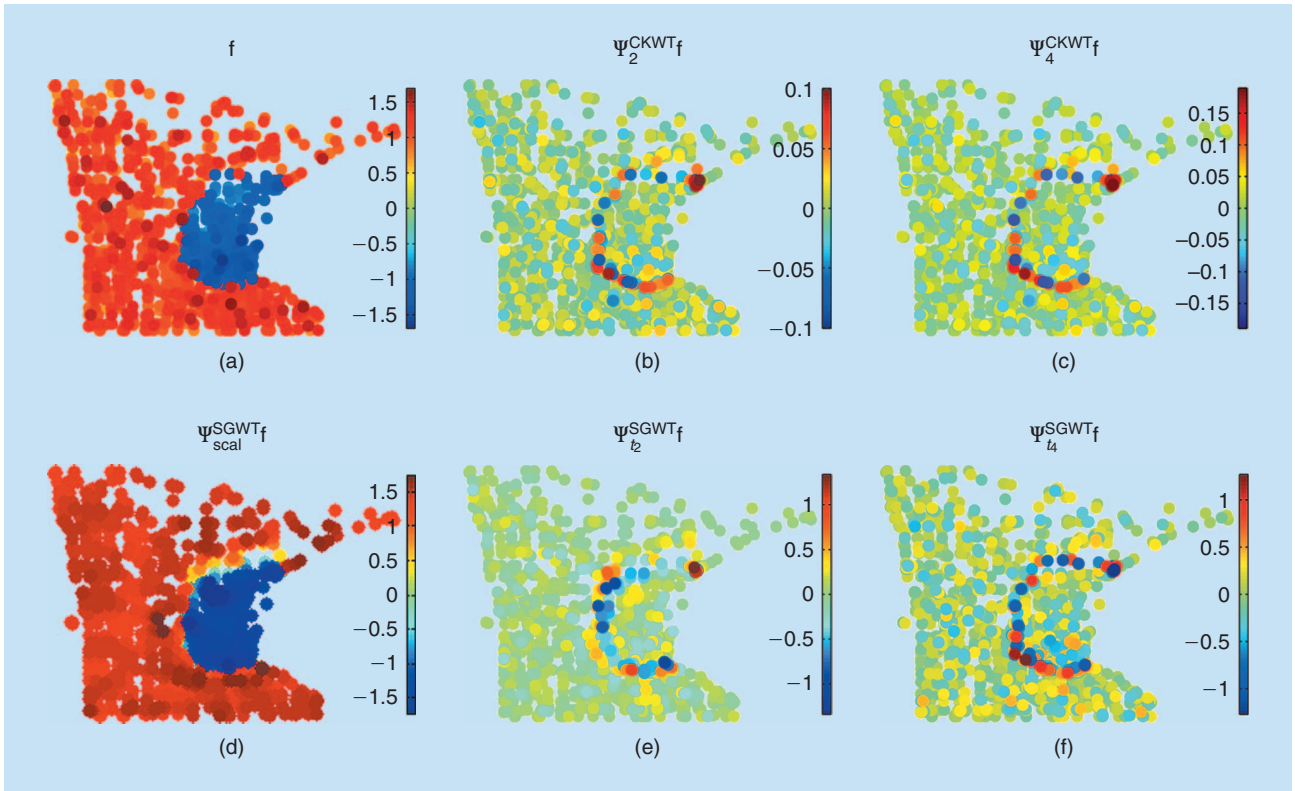



**[FIG6] The average spatial and spectral spreads of two example wavelet transforms on 5 instances of $d$-regular random graphs (size $N = 300$, degree $d = 5$). The coordinates of each point in this figure are the average spatial and spectral spreads across all wavelets at a given scale.**

Next, to empirically demonstrate the ability of these graph wavelet transforms to efficiently represent piecewise smooth signals on graphs, we compute the graph wavelet coefficients of the piecewise smooth signal with a sharp discontinuity shown in Figure 7(a) on the unweighted Minnesota road graph, where the color of a node represents the value of the signal at that vertex. We use the CKWT with scales $k = 1, 2, \ldots, 10$, and the SGWT with five wavelet scales as well as a scaling kernel. The bandpass wavelet kernel, scaling kernel, and values of the scales $t_1$, $t_2$, $t_3$, and $t_4$ are all designed by the SGWT toolbox [41]. The CKWT wavelet coefficients as Scales 2 and 4 are shown in Figure 7(b) and (c), and the SGWT scaling coefficients and wavelet coefficients at scales $t_2$ and $t_4$ are shown in Figure 7(d)–(f), respectively. Observe that for both transforms, the high-magnitude output coefficients are concentrated mostly near the discontinuity. This implies that these graph wavelet transforms are able to localize the high-pass information of the signal in the spatial domain, which the graph Fourier transform or other global transforms cannot do.

## SUMMARY, OPEN ISSUES, AND EXTENSIONS

We presented a generic framework for processing data on graphs, and we surveyed recent developments in the area of graph signal processing. In particular, we reviewed ways to generalize elementary operators such as filtering, convolution, and translation to graph setting. Such operations represent the core of graph signal processing algorithms, and they underlie the localized, multiscale transforms we discussed. For many of the generalized operators and localized, multiscale transforms, classical signal processing intuition from Euclidean spaces can be fairly directly




**[FIG7] (a) A piecewise smooth signal f with a severe discontinuity on the unweighted Minnesota graph. Parts (b) and (c) show wavelet coefficients of two scales of the CKWT. (d) Scaling coefficients of the SGWT. Parts (e) and (f) show wavelet coefficients of two scales of the SGWT. In both cases, the high-magnitude wavelet coefficients cluster around the discontinuity.**

extended to the graph setting. For example, we saw in the section "A Graph Fourier Transform and Notion of Frequency" how the notion of frequency extends nicely to the graph setting. However, signals and transforms on graphs can also have surprising properties due to the irregularity of the data domain. Moreover, these are by no means the only conceivable ways to generalize these operators and transforms to the graph setting. Thus, quite a few challenges remain ahead. In this section, we briefly mention a few important open issues and possible extensions.

### *OPEN ISSUES*

- Because all of the signal processing methods described in this article incorporate the graph structure in some way, construction of the underlying graph is extremely important. Yet, relatively little is known about how the construction of the graph affects properties of the localized, multiscale transforms for signals on graphs.
- As mentioned in the section "Other Graph Matrices," it is not always clear when or why we should use the normalized graph Laplacian eigenvectors, the unnormalized graph Laplacian eigenvectors, or some other basis as the graph spectral filtering basis.
- Similarly, in the vertex domain, a number of different distances, including the geodesic/shortest-path distance, the resistance distance [64], the diffusion distance [46], and algebraic distances [47], have useful properties, but it is not always clear which is the best to use in constructing or analyzing transform methods.
- Transform operators are only useful in high-dimensional data analysis if the computational complexity of applying the operator and its adjoint scales gracefully with the size of the signal. This fact is confirmed, for example, by the prevalence of fast Fourier transforms and other efficient computational algorithms throughout the signal processing literature. Most of the transforms for signals on graphs involve computations requiring the eigenvectors of the graph Laplacian or the normalized graph Laplacian. However, it is not practical to explicitly compute these eigenvectors for extremely large graphs, as the computational complexity of doing so does not scale gracefully with the size of the graph. Thus, an important area of research is approximate computational techniques for signal processing on graphs. Efficient numerical implementations for certain classes of graph operators have been suggested using polynomial approximations [4], [40], [41] and Krylov methods [11], but plenty of numerical issues remain open, including, e.g., a fast graph Fourier transform implementation.
- In Euclidean data domains, there is a deep mathematical theory of approximation linking properties of classes of signals to properties of their wavelet transform coefficients (see, e.g., [65]). A major open issue in the field of signal processing on graphs is how to link structural properties of graph signals and their underlying graphs to properties (such as

sparsity and localization) of the generalized operators and transform coefficients. Such a theory could inform transform designs, and help identify which transforms may be better suited to which applications. One issue at the heart of the matter is the need to better understand localization of signals in both the vertex and graph spectral domains. As discussed briefly in the section "Localized, Multiscale Transforms for Signals on Graphs," even defining appropriate notions of spreads in these domains is highly nontrivial. Moreover, unlike in the classical Euclidean settings, the graph Laplacian eigenvectors are often highly nonlocalized, making it more difficult to precisely identify the tradeoff between resolution in the vertex domain and resolution in the graph spectral domain. Agaskar and Lu [60] have begun to define such localization notions and study the resolution tradeoff.

### *EXTENSIONS*

The signal processing techniques we have described are focused on extracting information from a static signal on a static, weighted, undirected graph. Some clear extensions of this framework include: 1) considering directed graphs, as is done, for example, in [66], 2) considering time series of data on each vertex in a graph, and 3) considering a time-varying series of underlying graphs, as is done, for example, in [67], or any combination of these.

Finally, while the number of new analytic techniques for signals on graphs has been steadily increasing over the past decade, the application of these techniques to real science and engineering problems is still in its infancy. We believe the number of potential applications is vast, and hope to witness increased utilization of these important theoretical developments in the coming decade.


### ACKNOWLEDGMENTS

This work was supported in part by the European Commission under FETOpen grant 255931 UNLocX and in part by the National Science Foundation under grant CCF-1018977. The authors would like to thank the anonymous reviewers and Dorina Thanou for their constructive comments on earlier versions of this article.


### AUTHORS


*David I Shuman* (david.shuman@epfl.ch) received the B.A. degree in economics and the M.S. degree in engineering-economic systems and operations research from Stanford University, Stanford, California, in 2001 and the M.S. degree in electrical engineering: systems, the M.S. degree in applied mathematics, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, in 2006, 2009, and 2010, respectively. He is currently a postdoctoral researcher at the Institute of Electrical Engineering, Ecole Polytechnique Fédérale de Lausanne, Switzerland. His research interests include signal processing on graphs, computational harmonic analysis, and stochastic scheduling and resource allocation problems.

*Sunil K. Narang* (kumarsun@usc.edu) is a postdoctoral research associate working with Dr. Antonio Ortega in the Department of Electrical Engineering, University of Southern California (USC), Los Angeles. He received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Delhi, India, in 2005, and the Ph.D. degree in electrical engineering from USC, Los Angeles, in 2012. His research interests involve distributed compression in networks, transform-based analysis in graphs, and image and video processing.

*Pascal Frossard* (pascal.frossard@epfl.ch) received the M.S. and Ph.D. degrees, both in electrical engineering, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively. Between 2001 and 2003, he was a member of the research staff at the IBM T.J. Watson Research Center, Yorktown Heights, New York. He is now an associate professor at EPFL, where he heads the Signal Processing Laboratory (LTS4). His research interests include image representation and coding, visual information analysis, distributed image processing and communications, and media streaming systems. He is a Senior Member of the IEEE.

*Antonio Ortega* (antonio.ortega@sipi.usc.edu) received the telecommunications engineering degree from the Universidad Politecnica de Madrid, Spain, in 1989 and the Ph.D. degree in electrical engineering from Columbia University, New York, in 1994. He is a professor of electrical engineering at the University of Southern California and has served as an associate chair of the Department of Electrical Engineering-Systems. He is a Fellow of the IEEE and a member of ACM and Asia-Pacific Signal and Information Processing Association. He received the NSF CAREER Award and several paper awards, including most recently at ICIP 2011 and Globecom 2012. His research interests are in the areas of multimedia compression, communications, signal analysis, and graph signal processing.

*Pierre Vandergheynst* (pierre.vandergheynst@epfl.ch) received the M.S. degree in physics and the Ph.D. degree in mathematical physics from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1995 and 1998, respectively. From 1998 to 2001, he was a postdoctoral researcher with the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. From 2002 to 2007, he was an assistant professor at EPFL, where he is currently an associate professor. His research focuses on harmonic analysis, sparse approximations, and mathematical data processing, in general, with applications covering signal, image, and high-dimensional data processing; sensor networks; and computer vision.


### REFERENCES


[1] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. Honey, V. Wedeen, and O. Sporns, "Mapping the structural core of human cerebral cortex," *PLoS Biol.*, vol. 6, no. 7, pp. 1479–1493, 2008.

[2] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157.

[3] C. Apté, F. Damerau, and S. Weiss, "Automated learning of decision rules for text categorization," *ACM Trans. Inf. Syst.*, vol. 12, no. 3, pp. 233–251, 1994.

[4] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Proc. Ann. Conf. Comp. Learn. Theory*, 2003, LNCS, pp. 144–158.

[5] X. Zhu and Z. Ghahramani, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, Aug. 2003, pp. 912–919.

[6] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *Proc. ICML Workshop Stat. Relat. Learn. and Its Connections to Other Fields*, July 2004, pp. 132–137.

[7] D. Zhou and B. Schölkopf, "Regularization on discrete spaces," in *Pattern Recogn.* (Lect. Notes Comp. Sci., vol. 3663), W. G. Kropatsch, R. Sablatnig, and A. Hanbury, Eds. Berlin, Germany: Springer, 2005, pp. 361–368.

[8] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Learn. Theory* (Lect. Notes Comp. Sci.). Berlin, Germany: Springer, 2004, pp. 624–638.

[9] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Adv. Neural Inf. Process. Syst.*, vol. 16, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004, pp. 321–328.

[10] G. Peyré, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," in *Proc. ECCV'08*, 2008, Lect. Notes Comp. Sci., pp. 57–68.

[11] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognit.*, vol. 41, pp. 3328–3342, Nov. 2008.

[12] S. K. Narang, Y. H. Chao, and A. Ortega, "Graph-wavelet filterbanks for edge-aware image processing," in *Proc. IEEE Stat. Signal Process. Workshop*, Ann Arbor, MI, Aug. 2012, pp. 141–144.

[13] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, June 2010.

[14] L. J. Grady and J. R. Polimeni, *Discrete Calculus*. Berlin, Germany: Springer, 2010.

[15] C. Godsil and G. F. Royle, *Algebraic Graph Theory*. Berlin, Germany: Springer, 2001.

[16] F. K. Chung, *Spectral Graph Theory* (CBMS Regional Conference Series in Mathematics, vol. 92). Providence, RI: AMS Bookstore, 1997.

[17] D. Spielman, "Spectral graph theory," in *Combinatorial Scientific Computing*. Chapman and Hall/CRC Press, 2012.

[18] G. Taubin, "Geometric signal processing on polygonal meshes," in *Eurographics State of the Art Report*. Aug. 2000.

[19] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," in *Proc. ACM SIG-GRAPH Conf.*, New Orleans, LA, July 2000, pp. 279–286.

[20] I. Guskov, W. Sweldens, and P. Schroder, "Multiresolution signal processing for meshes," in *Proc. ACM SIG-GRAPH Conf.*, Los Angeles, CA, Aug. 1999, pp. 325–334.

[21] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bull. Amer. Math. Soc.*, vol. 43, pp. 439–561, Oct. 2006.

[22] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[23] D. Cvetković and I. Gutman, "Applications of graph spectra: An introduction to the literature," *Zbornik radova*, vol. 14, no. 22, pp. 9–34, 2011.

[24] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 53–94, 2006.

[25] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. IEEE Int. Conf. Acc., Speech, and Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 3921–3924.

[26] X. Zhu and M. Rabbat, "Graph spectral compressed sensing for sensor networks," in *Proc. IEEE Int. Conf. Acc., Speech, and Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 2865–2868.

[27] D. Gleich. *The MatlabBGL Matlab Library* [Online]. Available: http://www.cs.purdue.edu/homes/dgleich/packages/matlab_bgl/index.html

[28] S. Bougleux, A. Elmoataz, and M. Melkemi, "Discrete regularization on weighted graphs for image and mesh filtering," in *Scale Space Var. Methods Comp. Vision* (Lect. Notes Comp. Sci., vol. 4485), F. Sgallari, A. Murli, and N. Paragios, Eds. Berlin, Germany: Springer, 2007, pp. 128–139.

[29] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE Trans. Image Process.*, vol. 17, pp. 1047–1060, July 2008.

[30] S. Osher and J. Shen, "Digitized PDE method for data restoration," in *Handbook of Analytic-Computational Methods in Applied Mathematics*, G. Anastassiou, Ed. Boca Raton, FL: Chapman and Hall/CRC Press, 2000, pp. 751–771.

[31] M. Belkin and P. Niyogi, "Towards a theoretical foundation for Laplacian-based manifold methods," *J. Comp. Syst. Sci.*, vol. 74, no. 8, pp. 1289–1308, 2008.

[32] M. Hein, J.-Y. Audibert, and U. von Luxburg, "From graphs to manifolds: Weak and strong pointwise consistency of graph Laplacians," in *Learning Theory* (Lect. Notes Comp. Sci., vol. 3559), P. Auer and R. Meir, Eds. Berlin/Heidelberg, Germany: Springer, 2005, pp. 470–485.

[33] A. Singer, "From graph to manifold Laplacian: The convergence rate," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 128–134, July 2006.

[34] D. Ting, L. Huang, and M. I. Jordan, "An analysis of the convergence of graph Laplacians," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, June 2010, pp. 1079–1086.

[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, MA: Cambridge Univ. Press, 1990.

[36] T. Bıyıkoğlu, J. Leydold, and P. F. Stadler, "Chapter 1: Introduction," *Laplacian Eigenvectors of Graphs* (Lect. Notes in Math., vol. 1915). Berlin, Germany: Springer, 2007.

[37] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter-banks for graph structured data," *IEEE. Trans. Signal Process.*, vol. 60, pp. 2786–2799, June 2012.

[38] N. J. Higham, "Chapter 1: Theory of Matrix Functions," *Functions of Matrices*. Soc. for Industrial and Appl. Mathematics, 2008, pp. 1–29.

[39] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[40] D. I Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *Proc. Int. Conf. Distr. Comput. Sensor Syst.*, Barcelona, Spain, June 2011, pp. 1–8.

[41] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[42] D. I Shuman, B. Ricaud, and P. Vandergheynst, "A windowed graph Fourier transform," in *Proc. IEEE Stat. Signal Process. Workshop*, Ann Arbor, MI, Aug. 2012, pp. 133–136.

[43] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proc. Natl. Acad. Sci.*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[44] M. Maggioni, J. C. Bremer, R. R. Coifman, and A. D. Szlam, "Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs," in *Proc. SPIE Wavelet XI*, Sep. 2005, vol. 5914, pp. 59141M1–59141M13.

[45] J. C. Bremer, R. R. Coifman, M. Maggioni, and A. D. Szlam, "Diffusion wavelet packets," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 95–112, 2006.

[46] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1393–1403, Sept. 2006.

[47] D. Ron, I. Safro, and A. Brandt, "Relaxation-based coarsening and multiscale graph organization," *Multiscale Model. Simul.*, vol. 9, no. 1, pp. 407–423, Sept. 2011.

[48] S. T. Barnard and H. D. Simon, "Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems," *Concurrency: Practice Experience*, vol. 6, no. 2, pp. 101–117, Apr. 1994.

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

[50] C. Walshaw. *The Graph Partitioning Archive* [Online]. Available: http://staffweb.cms.gre.ac.uk/~wc06/partition/

[51] B. Aspvall and J. R. Gilbert, "Graph coloring using eigenvalue decomposition," *SIAM J. Alg. Disc. Meth.*, vol. 5, no. 4, pp. 526–538, 1984.

[52] I. Pesenson, "Sampling in Paley–Wiener spaces on combinatorial graphs," *Trans. Amer. Math. Soc*, vol. 360, no. 10, pp. 5603–5627, 2008.

[53] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *Proc. IEEE INFOCOM*, Mar. 2003, vol. 3, pp. 1848–1857.

[54] A. D. Szlam, M. Maggioni, R. R. Coifman, and J. C. Bremer, Jr., "Diffusion-driven multiscale analysis on manifolds and graphs: Top-down and bottom-up constructions," in *Proc. SPIE Wavelets*, Aug. 2005, vol. 5914, pp. 445–455.

[55] W. Wang and K. Ramchandran, "Random multiresolution representations for arbitrary sensor network graphs," in *Proc. IEEE Int. Conf. Acc., Speech, and Signal Process.*, May 2006, vol. 4, pp. 161–164.

[56] M. Jansen, G. P. Nason, and B. W. Silverman, "Multiscale methods for data on graphs and irregular multidimensional situations," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 71, no. 1, pp. 97–125, 2009.

[57] M. Gavish, B. Nadler, and R. R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, June 2010, pp. 367–374.

[58] S. K. Narang and A. Ortega, "Local two-channel critically sampled filter-banks on graphs," in *Proc. Int. Conf. Image Process.*, Hong Kong, Sept. 2010, pp. 333–336.

[59] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall, 1995.

[60] A. Agaskar and Y. M. Lu, "Uncertainty principles for signals defined on graphs: Bounds and characterizations," in *Proc. IEEE Int. Conf. Acc., Speech, and Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 3493–3496.

[61] R. Wagner, V. Delouille, and R. Baraniuk, "Distributed wavelet de-noising for sensor networks," in *Proc. IEEE Int. Conf. Dec. and Contr.*, San Diego, CA, Dec. 2006, pp. 373–379.

[62] G. Shen and A. Ortega, "Transform-based distributed data gathering," *IEEE. Trans. Signal Process.*, vol. 58, no. 7, pp. 3802–3815, July 2010.

[63] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, 1998.

[64] D. J. Klein and M. Randić, "Resistance distance," *J. Math. Chem.*, vol. 12, no. 1, pp. 81–95, 1993.

[65] D. L. Donoho, "Unconditional bases are optimal bases for data compression and for statistical estimation," *Appl. Comput. Harmon. Anal.*, vol. 1, no. 1, pp. 100–115, Dec. 1993.

[66] F. Chung, "Laplacians and the Cheeger inequality for directed graphs," *Ann. Combinatorics*, vol. 9, no. 1, pp. 1–19, 2005.

[67] J. D. Lee and M. Maggioni, "Multiscale analysis of time series of graphs," in *Proc. Int. Conf. Samp. Theory and Appl.*, Singapore, May 2011.

[SP]