INVITED PAPER

# Graph Signal Processing: Overview, Challenges, and Applications

*This article presents methods to process data associated to graphs (graph signals) extending techniques (transforms, sampling, and others) that are used for conventional signals.*

By Antonio Ortega, *Fellow IEEE*, Pascal Frossard, *Fellow IEEE*, Jelena Kovačević, *Fellow IEEE*, José M. F. Moura, *Fellow IEEE*, and Pierre Vandergheynst

**ABSTRACT** | Research in graph signal processing (GSP) aims to develop tools for processing data defined on irregular graph domains. In this paper, we first provide an overview of core ideas in GSP and their connection to conventional digital signal processing, along with a brief historical perspective to highlight how concepts recently developed in GSP build on top of prior research in other areas. We then summarize recent advances in developing basic GSP tools, including methods for sampling, filtering, or graph learning. Next, we review progress in several application areas using GSP, including processing and analysis of sensor network data, biological data, and applications to image processing and machine learning.

**KEYWORDS** | Graph signal processing (GSP); network science and graphs; sampling; signal processing

## I. INTRODUCTION AND MOTIVATION

Data is all around us, and massive amounts of it. Almost every aspect of human life is now being recorded at all levels: from the marking and recording of processing inside the cells starting with the advent of fluorescent markers, to our personal data through health monitoring devices and apps, financial and banking data, our social networks, mobility and traffic patterns, marketing preferences, fads, and many more. The complexity of such networks [1] and interactions means that the data now reside on irregular and complex structures that do not lend themselves to standard tools.

Manuscript received November 25, 2017; revised March 10, 2018; accepted March 23, 2018. Date of current version April 24, 2018. *(Corresponding author: Antonio Ortega.)*
**A. Ortega** is with the University of Southern California, Los Angeles, CA 90089, USA (e-mail: antonio.ortega@sipi.usc.edu).
**P. Frossard,** and **P. Vandergheynst** are with EPFL, Lausanne, Switzerland-1015, Lausanne
**J. Kovačević,** and **J. M. F. Moura** are with Carnegie Mellon University, Pittsburgh, PA 15213, USA.

Digital Object Identifier: 10.1109/JPROC.2018.2820126

Graphs offer the ability to model such data and complex interactions among them. For example, users on Twitter can be modeled as nodes while their friend connections can be modeled as edges. This paper explores adding attributes to such nodes and modeling those as signals on a graph; for example, year of graduation in a social network, temperature in a given city on a given day in a weather network, etc. Doing so requires us to extend classical signal processing concepts and tools such as Fourier transform, filtering, and frequency response to data residing on graphs. It also leads us to tackle complex tasks such as sampling in a principled way. The field that gathers all these questions under a common umbrella is graph signal processing (GSP) [2], [3].

While the precise definition of a graph signal will be given later in the paper, let us assume for now that a graph signal is a set of values residing on a set of nodes. These nodes are connected via (possibly weighted) edges. As in classical signal processing, such signals can stem from a variety of domains; unlike in classical signal processing, however, the underlying graphs can tell a fair amount about those signals through their structure. Different types of graphs model different types of networks that these nodes represent.

Typical graphs that are used to represent common real-world data include Erdős–Rényi graphs, ring graphs, random geometric graphs, small-world graphs, power-law graphs, nearest-neighbor graphs, scale-free graphs, and many others. These model networks with random connections (Erdős–Rényi graphs), networks of brain neurons (small-world graphs), social networks (scale-free graphs), and others.

As in classical signal processing, graph signals can have properties, such as smoothness, that need to be appropriately defined. They can also be represented via basic atoms and can have a spectral representation. In particular, the graph Fourier transform allows us to develop the intuition gathered in the classical setting and extend it to graphs; we can talk about the notions of frequency and bandlimitedness, for example. We can filter graph signals. They can be sampled, a notoriously hard problem; with GSP, one gains access to principled tools mimicking the classical ones. We can denoise graph signals, we can learn their underlying structure, we can model them. If the graphs cannot be directly observed, we can also learn their structure from data. All of these topics will be explored in more detail in what follows.

As illustration, consider what smoothness of graph signals may represent in urban settings. First, however, we have to understand what smoothness means on graphs. For example, we can think of smooth graph signals in the vertex domain, that is, signals where neighboring nodes tend to have similar values. We can also think of the smoothness of graph signals in the spectral domain, typically called bandlimitedness. Different types of smoothness are possible in the spectral domain where, instead of a sharp cutoff, frequency content may decay according to some law.

Fig. 1 illustrates how a piecewise-smooth signal model can be used to approximate the taxi-pickup distribution in Manhattan. Fig. 1(a) shows the number of taxi pickups (with blue as low numbers up to red for high numbers) from 6 P.M. to 7 P.M. on January 4, 2015, projected to the nearest intersection. What one can observe is that the busy shopping/entertainment areas such as Times Square in New York City show similar mobility patterns, as illustrated by the number of taxi pickups. In other words, neighboring intersections around shopping areas will exhibit similar (homogeneous) mobility patterns, likely corresponding to similar lifestyle behaviors, while the intersections around residential areas will exhibit different, yet still homogeneous mobility patterns and lifestyle behaviors; so, shopping areas will be very busy during the regular business hours and after work, but perhaps less so late in the evening, while residential areas will be busy on weekends and at times parents drop off and pick up their kids from school. Similarly, in social networks, within a given social circle, users' profiles tend to be homogeneous, while within a different social circle they will be different, yet still homogeneous; for example, friends from a high school in New York City will probably have similar taste in entertainment, while friends from a high school in Lausanne will also have similar tastes but different from those from teenagers in New York. We can model such data as piecewise-smooth graph signals [see, for example, how a piecewise-smooth signal in Fig. 1(b) provides a good approximation to the actual measurements of Fig. 1(a)], as they capture large variations between pieces and small variations within pieces.

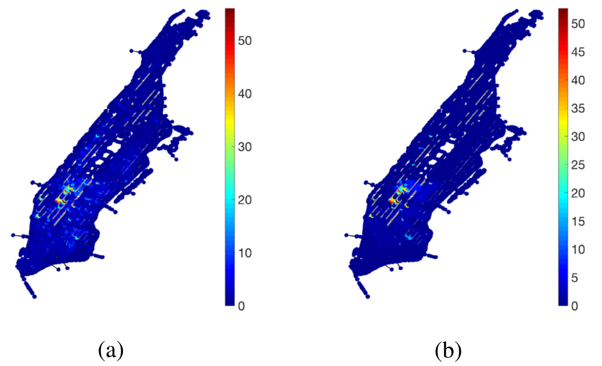



**Fig. 1.** ***Piecewise-smooth graph signals approximate irregular, nonsmooth graph signals by capturing both large variations at boundaries as well as small variations within pieces. (a) Data captured in Manhattan (13 679 intersections). (b) Piecewise-smooth approximation to the data with 50 coefficients (from [4] with permission).***

A number of communities consider similar questions as GSP. In particular, the machine learning community also considers graph structure/data and, at times, uses similar methods as GSP does. For example, the graph Fourier basis is related to Laplacian eigenvectors and graph signal recovery is related to semisupervised learning with graphs [5]. There are, however, a few differences. 1) GSP defines a framework that allows the extension of classical signal processing concepts. 2) Sampling data associated with graphs is rarely studied in the machine learning community and the sampling problem on graphs is generally a hard one; graph-structured data representations (such as graph filter banks and graph dictionary learning) are also rarely studied in the machine learning community. 3) Given that GSP extends classical signal processing concepts, it is in a position to consider low-level processing such as denoising, inpainting, and compression. 4) Machine learning typically considers a graph as a discrete version of a manifold. In many real-world applications that are associated with graphs, this assumption is, however, not true. On the contrary, GSP does not make this assumption. For example, there is no underlying manifold for online social networks. In summary, as will be discussed next, and illustrated throughout this paper, GSP has strong connections to several theoretical and practical research domains; its promise lies in its ability to develop new tools and approach existing problems from different perspectives.

### A. Related Work in Other Areas

*1) Network Science:* This area addresses issues such as uncovering community relations, perceived alliances, quantifying connectedness, or determining the relevance of specific agents [6]–[9]. Thus, much of this work does not concentrate on the data but rather their structure. It determines, for example, the size of the giant component, distribution of component sizes, degree and clique distributions, clustering coefficients, betweenness and closeness centralities, path length, and network diameter [1], [10], [11]. Connections to GSP are primarily due to graph spectra that GSP builds upon, which is strongly related to the structure of the graph [12]. As an example, spectral clustering methods use the low-frequency eigenvectors of the Laplacian [13] and can thus be addressed from a GSP perspective as well [14].

*2) Network Processes:* The aim is to model propagation over networks, including such phenomena as diffusion of diseases and epidemics, spread of (fake) news, memes, fads, voting trends, imitation and social influence, propagation of failures, and blackouts. Common models are similar to stochastic automata where the states of the nodes (the "data") evolve through local rules, i.e., according to exogenous (external to the network) and endogenous (internal to the network) effects. For example, using terminology from epidemics, nodes of the graph representing agents or individuals of a population can be infected (adopt an opinion or spread a rumor), or susceptible (open to adopt an opinion or spread a rumor). Infected nodes can heal and become susceptible again; susceptible nodes can become infected either by an action external to the network or by an action of infected neighbors [15], [16]. As the analysis of such network processes is difficult, traditionally, the network is abstracted out, assuming that any node can infect any other node (full mixing or complete network). To account for the impact of the network [17], resorting to numerical studies is precluded except for very small networks since the network state space $\{0,1\}^N$ grows exponentially fast ($2^N$, for $N$ agents). To study these processes [18], [19], one usually considers one of two asymptotic regimes: 1) long-term behavior (time asymptotics), attempting to find the equilibrium distribution of the process [20], [21]; or 2) large network asymptotics (mean-field approximation) [22] leading to the study of the qualitative behavior of nonlinear ordinary differential equations [23]. Because asymptotic behavior can be seen to depend on the eigenstructure of the underlying graph, GSP representations as those discussed in Section III-B can be used to characterize the evolution of a system. As an example, several papers have explored the use of GSP techniques to improve the efficiency of value function estimation in a reinforcement learning scenario [24], [25].

*3) Graphical Models:* The focus in this area is on inference and learning from large data sets [26]–[30]. The data are modeled as a set of random variables described by a family of Gibbs probability distributions, and the underlying graph (whose nodes label the variables) captures statistical dependence and conditional independence among the data. Acyclic graphs [31], [32] represent Bayesian networks, and undirected graphs represent Markov random fields [33]–[35]. Graphical models exploit factorizations of the joint distribution to develop efficient message passing algorithms for inference and find application in many areas such as modeling texture and other features in image processing [36]–[41]; see [42] for illustrative applications in several domains. Recent work on learning graph from data [43], [44], which makes use of Markov random field models to define optimality criteria for the learned graphs, connects graphical models to GSP.

## B. Historical Perspective on Graph Signal Processing

We now briefly review some of the prior work that is more directly connected and in the spirit of signal processing on graphs, [2], [3]. We organize the discussion along two main lines; some parts of the exposition follow closely [2], [45].

*1) From Algebraic Signal Processing to Graph Signal Processing:* The sequence of papers [46]–[50] introduced algebraic signal processing (ASP), an axiomatic approach to time signal processing. ASP starts from a signal model $\Omega$. Many signal models are possible, and a relevant question is to determine which one is more appropriate for a given application or should be associated with a given linear transform. Under appropriate conditions, the signal model is generated from a simple filter, the shift, which then determines filtering, convolution, the Fourier transform, frequency, and spectral analysis among other common concepts, and constructs from traditional digital signal processing. Such formalism allowed for a systematic and uniform framework for variations of classical signal processing. ASP, after appropriately defining a space line-graph signal model [49], can be used to show that the DCT plays the same role for that signal model as the one the DFT plays for the time (cyclic) model. ASP led to the introduction of, possibly weighted, graph adjacency matrices as shifts that generate the graph signal model for signals indexed by nodes of an arbitrary directed or undirected graph [2], [51]. This choice is satisfying in the sense that, when the signal model is the classical time signal model, the shift and the graph signal model revert to the classical time shift (delay) and signal model [48] (see Section II). Subsequently, authors the authors have proposed other shifts obtained from the adjacency matrix of the graph [52], [53] that attempt to preserve isometry of the shift, but in some cases lose the locality of the adjacency matrix shift [52]

*2) From Graph Laplacian Spectral Clustering to Laplacian-Based GSP:* References [54]–[57] develop low-dimensional representations for large high-dimensional data through spectral graph theory [56], [58] and the graph Laplacian [12], by projecting the data on a low-dimensional subspace generated by a small subset of the Laplacian eigenbasis [13]. The use of the graph Laplacian is justified by assuming the data are smooth on the data space (manifold). References [59]–[61] choose discrete approximations to other continuous operators, for example, a conjugate to an elliptic Schrödinger-type operator, and obtain other spectral bases for the characterization of the geometry of the manifold underlying the data.

Coming from another angle, motivated by processing data collected by sensor networks where sensors are irregularly placed, different authors develop regression algorithms [62], wavelet decompositions [61], [63]–[66], filter banks on graphs [67], [68], denoising [69], and compression schemes using the graph Laplacian [70]. Some of these references consider distributed processing of data from sensor fields, while others study localized processing of signals on graphs in a multiresolution fashion by representing data using wavelet-like bases with varying "smoothness" or defining transforms based on node neighborhoods. For example, [66] uses the graph Laplacian and its eigenbasis to define a spectrum and a Fourier transform of a signal on a graph. Besides using the graph Laplacian, these works apply to data indexed by undirected graphs with real, nonnegative edge weights. This approach is more fully developed in [3], which adopts the graph Laplacian as basic building block to develop GSP for data supported by undirected graphs.

*3) Image Processing, Computer Graphics, and GSP:* In addition, graph-based approaches have been widely used in signal processing contexts. For example, several authors represent images as graphs for segmentation [71], [72] and popular image-dependent filtering methods can be interpreted from a graph perspective [73]. Models used in computer graphics applications can often be viewed as graphs (e.g., meshes where vertices form triangles to which attributes are associated) and graph-based filtering, processing, and multiresolution representations can be developed [74]–[76].

### C. Outline of the Paper

The outline of the paper is as follows. Section II starts by presenting the framework and key ingredients of GSP. It explains how the concepts from classical signal processing such as signals, filters, and Fourier transform, among others, extend to complex structures where data are indexed by nodes on a graph. Section III covers some state-of-the-art topics and associated challenges, such as the definition of frequency, graph learning, sampling and representations. Section IV follows up with applications of GSP in sensor networks, biological networks, 3-D point cloud processing, and machine learning. Section V gives some conclusions.

## II. KEY INGREDIENTS OF GRAPH SIGNAL PROCESSING

In this section, we introduce basic GSP concepts. While more formal derivations of GSP can be developed, e.g., from the signal model introduced in the algebraic signal processing (ASP) [46]–[49] or from the spectral perspective developed in [3] and [66] based on spectral graph theory [12], we choose a more intuitive presentation by first reviewing the concept of shift in digital signal processing (DSP) (Section II-A) in order to emphasize connections between DSP and GSP. We then develop a corresponding notion of shift for GSP (Section II-B). This in turns leads to the definition of frequencies for graph signals (Section II-C) and their interpretation (Section II-D). We focus on tools derived from the adjacency or Laplacian matrices of the graphs, as these are by far the most widely used. However, we note that each of these approaches have their own limitations and there are active research efforts to build GSP tools on alternative definitions of frequency (see Section III-A).

### A. The Role of Shifts in DSP

DSP [77]–[81] studies time signals. GSP[1] [2], [3], [45] extends DSP to signal samples indexed by nodes of a graph. At a very high level, DSP and, therefore, GSP study: 1) signals and their representations; 2) systems that process signals, usually referred to as filters; 3) signal transforms, including two very important ones, namely, the $z$-transform and the Fourier transform; and 4) sampling of signals, as well as other more specialized topics.

Consider $N$ samples of a signal $s_n$, $n = 0, 1, \ldots, N-1$. We restrict ourselves to signals with a finite number $N$ of samples and to filters with finite impulse response (FIR) filters. The $z$-transform $s(z)$ of the (real or complex valued) time signal $s = \{s_n : n = 0, 1, \ldots, N-1\}$ organizes its samples $s_n$ into an ordered set of time samples, where sample $s_n$ at time $n$ precedes $s_{n+1}$ at time $n+1$ and succeeds $s_{n-1}$ at time $n-1$. In other words, the signal is given by the $N$-tuple $s = (s_0, s_1, \ldots, s_{N-1})$. This representation is achieved by using a formal variable, say $z^{-1}$, called the shift (or delay), so that the signal of $N$-samples is represented by

$$s(z) = \sum_{n=0}^{N-1} s_n z^{-n}. \tag{1}$$

The $z$-transform $s(z)$ provides a (formal)[2] polynomial representation of the signal that is useful in studying how signals are processed by filters. Clearly, given $s(z)$ we can recover the signal $s$ [80], [81].

The discrete Fourier transform (DFT) of the signal $s$ is $\hat{s} = \{\hat{s}_k : k = 0, \ldots, N-1\}$ given by

$$\hat{s}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} s_n e^{-j\frac{2\pi}{N}kn}. \tag{2}$$

The $\hat{s}_k$ are the Fourier coefficients of the signal. The DFT represents the signal $s$ in the dual or frequency domain, leading to concepts such as frequency, spectrum, low-, band-, and high-pass signals. The discrete frequencies are $\Omega_k = (2\pi k/N)$, $k = 0, 1, \ldots, N-1$, and the $N$ signals $(x_k[n])$

$$\left\{x_k[n] = \frac{1}{\sqrt{N}} e^{-j\frac{2\pi}{N}kn} : n = 0, 1, \ldots, N-1\right\}_{k=0}^{N-1}$$

are the spectral components.

The signal is recovered from its Fourier coefficients by the inverse DFT

$$s_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \hat{s}_k e^{j\frac{2\pi}{N}kn}, s = 0, 1, \ldots, N-1. \tag{3}$$

In DSP, besides signals, we also have filters $h$. A FIR filter is also represented by a polynomial in $z^{-1}$

$$h(z) = \sum_{n=0}^{N-1} h_n z^{-n} \tag{4}$$

[1]We consider here only linear GSP.

[2]While in DSP $z$ is a complex variable, which leads to the DFT when restricted to the unit circle as in (2), here we establish the link to GSP by viewing $z$ as a placeholder for each sample of the signal.

so that the output $s_{\text{out}}$ of filter $h$ applied to signal $s_{\text{in}}$ is

$$s_{\text{out}}(z) = h(z) \cdot s_{\text{in}}(z). \tag{5}$$

Because we are only considering finite time signals, and the product above could result in $s_{\text{out}}(z)$ being a polynomial in $z^{-1}$ of degree greater than $N-1$, we have to consider boundary conditions (b.c.). For simplicity, we consider periodic extensions of the signal, i.e., the signal sample $s_N$ is equal to the signal sample $s_0$; more generally, $s_n = s_{n \bmod N}$. In other words, the real line is folded around the circle. Defining the shift or delay filter

$$h_{\text{shift}}(z) = z^{-1}$$

and applying it to a signal $s_{\text{in}} = (s_0, s_1, \ldots, s_{N-1})$ gives an output

$$s_{\text{out}} = h_{\text{shift}} \cdot s_{\text{in}} = (s_{N-1}, s_0, s_1, \ldots, s_{N-2}).$$

By (4), any filter $h$ in DSP is a polynomial in the shift, i.e., it is built from series and parallel combinations of shifts. Thus, the shift is the basic building block in DSP, from which we can build more complicated filters.

A second very important DSP property that is adopted in GSP is shift invariance. This readily follows from

$$z^{-1} \cdot h(z) = h(z) \cdot z^{-1}. \tag{6}$$

In words, the series combination of filters is commutative, a filter commutes with the shift filter—delaying the input signal $s_{\text{in}}$ and then filtering the delayed input signal leads to the same signal as first filtering the input signal $s_{\text{in}}$ and then delaying the filtered output.

> Restating for emphasis, both (1) and (4) show the principal role played by the shift $z^{-1}$ in DSP. We represent signals by (finite degree) polynomials in $z^{-1}$ and build filters also as polynomials in $z^{-1}$.

## B. Defining Shifts in Graph Signal Processing

We now extend the above concepts and tools to graph signals, i.e., signals whose samples are indexed by the nodes of arbitrary graphs. To do so, we start by reinterpreting the finite signals from the previous section as vectors rather than tuples or sequences.

Rewrite the signal $s = (s_0, s_1, \ldots, s_{N-1})$ as the vector

$$\boldsymbol{s} = [s_0\, s_1 \cdots s_{N-1}]^{\top} \in \mathbb{C}^N$$

where for generality we allow the signal to be complex valued. Using this notation, a filter $h$ is represented by a matrix $\mathbf{H}$ and (5) can be simply written as a matrix–vector multiplication

$$\boldsymbol{s}_{\text{out}} = \mathbf{H} \cdot \boldsymbol{s}_{\text{in}}$$

where filters are represented by matrices, while signals are represented by vectors. In particular, the shift filtering operation corresponds to multiplication by a circulant matrix $\mathbf{A}_c$

$$[s_{N-1}\, s_0 \ldots s_{N-2}]^{\top} = \mathbf{A}_c \cdot [s_0\, s_1 \ldots s_{N-1}]^{\top}$$

given by the cyclic shift

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix}. \tag{7}$$

A graph interpretation for the DSP concepts of Section II-A can be achieved by viewing the 0-1 shift matrix $\mathbf{A}_c$ of (7) as the adjacency matrix of a graph. Labeling the rows and columns of $\mathbf{A}_c$ from 0 to $N-1$, define the graph $G_c = (V, E)$ with node set $V = \{0, 1, \ldots, N-1\}$. Row $n$ of $\mathbf{A}_c$ represents the set of in-edges of node $n$ in $G_c$—if there is an entry 1 at column $\ell$, $\mathrm{A}_{c,n\ell} = 1$, then there is an edge from $\ell$ to $n$. $\mathbf{A}_c$ is then the adjacency matrix of the cycle graph in Fig. 2.

> The key point we make is the dual role of the matrix $\mathbf{A}_c$ in (7), which represents both the shift $z^{-1}$ in DSP and the adjacency matrix of the associated time graph in Fig. 2.

This graph interpretation of DSP can be extended to develop a linear time shift invariant GSP [2]. Consider now a graph signal $\boldsymbol{s} \in \mathbb{C}^N$, where the entries of the signal $\boldsymbol{s}$ are indexed by the $N$ nodes of an arbitrary graph $G = (V, E)$, $v_1, \ldots, v_N$. Assuming that the graph has edge weights $w_{ij}$, denote an edge of weight $w_{ij}$ going from $v_j$ to $v_i$, then we can define the following algebraic representations associated to $G$.

*Definition 1 (Algebraic Representations of Graphs):* The adjacency matrix is a matrix $\mathbf{A}$, such that $(\mathbf{A})_{ij} = w_{ij}$. In the particular case where the graph is undirected, we have $w_{ij} = w_{ji}$, $\mathbf{A}$ is now symmetric, and we also define the degree matrix of $G$, a diagonal matrix $\mathbf{D}$, with entries $(\mathbf{D})_{ii} = \sum_{j=1}^{N} (\mathbf{A})_{ij}$ and $(\mathbf{D})_{ij} = 0$ for $i \neq j$, the combinatorial graph Laplacian defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, and the symmetric normalized Laplacian $\boldsymbol{\mathcal{L}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$.

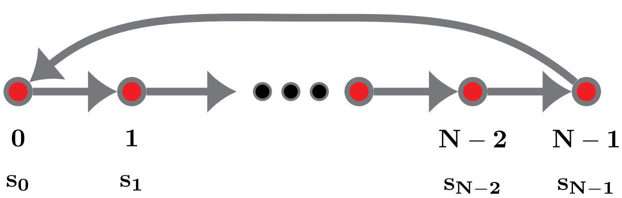



**Fig. 2.** ***Time graph: Cycle graph $G_c$.***

The adjacency matrix $\mathbf{A}$ can be adopted as the shift [2] for this general graph. Other choices have been proposed, including the Laplacians [3], or variations of these matrices [52], [53]. Different choices for the shift present different tradeoffs. The adjacency matrix $\mathbf{A}$ reduces to the shift in classical time DSP and applies to directed and undirected graphs,[3] while the graph Laplacian applies only to undirected graphs, so that $\mathbf{L}$ is symmetric and positive semidefinite, which avoids a certain number of analytical and numerical difficulties that may arise when choosing $\mathbf{A}$. Furthermore, graph Laplacian spectra have been widely studied in the field of spectral graph theory [12]. In specific applications, one should consider the various definitions and choose the one that leads to the best tradeoff for the problem being considered [82]. This choice is further discussed in Sections II-E and III-A.

For time signals, as discussed with respect to (1), the basis $\{z^{-n}\}_{n=0}^{N-1}$ orders the samples of the signal by increasing order of the time labels (nodes in time graph). Rewriting (1), we get

$$s(z) = \left[\left(z^{-1}\right)^0 z^{-1} \ldots z^{-(N-1)}\right] [s_0\, s_1 \ldots s_{N-1}]^\top.$$

In GSP, ordering the samples corresponds to labeling the nodes of the graph. This labeling or numbering fixes the adjacency matrix of the graph, and hence the graph shift. The columns of the graph shift provide a basis and a representation for the graph signals. Other bases could be used, leading to different signal representations. We note that relabeling the nodes of the graph by a permutation $\Pi$ conjugates the shift by $\Pi$

$$A_\Pi = \Pi A \Pi^\top.$$

Following the analogy with DSP, we can now define the notion of shift invariance and polynomial filters for arbitrary graphs. A filter represented by $\mathbf{H}$ will be shift invariant if it commutes with the shift

$$\mathbf{AH} = \mathbf{HA}.$$

As proven in [2], if the characteristic polynomial $p_A(z)$ and the minimum polynomial[4] $m_A(z)$ of $\mathbf{A}$ are equal, then every filter commuting with $\mathbf{A}$ is a polynomial in $\mathbf{A}$, i.e.,

$$\mathbf{H} = h(\mathbf{A}).$$

For equality $p_A(z) = m_A(z)$, to each eigenvalue of $\mathbf{A}$ there corresponds a single eigenvector.[5] A simpler condition is for the eigenvalues of $\mathbf{A}$ to be distinct. To keep the discussion simple, unless otherwise stated, we assume $\mathbf{A}$ has $N$ distinct eigenvalues and hence a complete set of eigenvectors.

By the Cayley–Hamilton theorem of linear algebra [83], [84]

$$\text{degree}(h(z)) = \text{degree}(p_A(z)) \le N-1.$$

In fact, $\text{degree}(h(z)) \le \text{degree}(m_A(z)) \le \text{degree}(p_A(z))$. In words, shift invariant filters are polynomials with degree at most $\text{degree}(m_A(z))$.

## C. Frequency Representations for Graph Signals

In DSP and in linear systems, we are interested in signals that are invariant when processed by a (linear) filter, i.e.,

$$h \cdot s_{\text{in}} = \alpha s_{\text{in}}$$

where $\alpha$ is a scalar (from the base field). Such $s_{\text{in}}$ are, of course, the eigensignals of the filter $h$. In GSP, we define filters as matrices and thus the eigensignals of $h$ are the eigenvectors of the corresponding $\mathbf{H}$. More interestingly, since shift-invariant filters are polynomials of a single matrix, the shift $\mathbf{A}$, we only need to consider the eigenvectors of $\mathbf{A}$. Then, write

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1} \tag{8}$$

where $\mathbf{V} = [v_0 \ldots v_{N-1}]$ is the matrix of the $N$ eigenvectors of $\mathbf{A}$, and $\boldsymbol{\Lambda} = \text{diag}[\lambda_0 \ldots \lambda_{N-1}]$ is the matrix of distinct eigenvalues of $\mathbf{A}$. Because we assume that $\mathbf{A}$ has a complete set of eigenvectors, $\mathbf{V}$ is invertible. Then, it is straightforward to verify that for each (polynomial) filter

$$\begin{aligned}\mathbf{H} &= h(\mathbf{A})\\ &= h\left(\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}\right)\\ &= \sum_{m=0}^{M-1} h_m \left(\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}\right)^m\\ &= \mathbf{V}h(\boldsymbol{\Lambda})\mathbf{V}^{-1}\end{aligned} \tag{9}$$

where $h(\boldsymbol{\Lambda})$ is the diagonal matrix

$$h(\boldsymbol{\Lambda}) = \text{diag}\left[h(\lambda_0) \ldots h(\lambda_{N-1})\right]. \tag{10}$$

We can promptly verify that the eigenvectors of $\mathbf{A}$ are the eigenfunctions of the (polynomial) filter

$$\begin{aligned}\mathbf{H}v_m &= \mathbf{V}h(\boldsymbol{\Lambda})\mathbf{V}^{-1}v_m\\ &= \mathbf{V}h(\boldsymbol{\Lambda})e_m\\ &= h(\lambda_m)v_m\end{aligned} \tag{11}$$

where $e_m$ is the zero vector except for entry $m$ that is a one. Equation (11) is the GSP counterpart to the classical DSP fact that exponentials are eigenfunctions of linear systems. As such the response of the filter to an exponential is the same exponential amplified or attenuated by a gain that is the frequency response of the filter at the frequency

[3]Note that the graph defined by $\mathbf{A}_c$ in (7) is directed in order to match exactly the behavior of shifts in time in DSP, which are always directed, i.e., we either move forward or backward in time. But, in general, the notion of a graph shift applies to any adjacency matrix, whether corresponding to a directed or an undirected graph. In what follows both directed and undirected graphs are considered.

[4]For a matrix $\mathbf{A}$, the minimal polynomial $m_A(z)$ is the polynomial of minimal degree having $\mathbf{A}$ as a root.

[5]In other words, the Jordan form of $\mathbf{A}$ has single blocks for each distinct eigenvalue.

of the exponential. We refer to this as the invariance property of exponentials with respect to linear systems in DSP. Accordingly, (11) shows the invariance of the eigenvectors of the shift operator $\mathbf{A}$ with respect to graph filters.

Finally, we can introduce the Fourier transform for graph signals. The cyclic shift in (7) can be written as

$$\mathbf{A}_c = \mathbf{DFT}_N^{-1} \begin{pmatrix} e^{-j\frac{2\pi \cdot 0}{N}} & & \\ & \ddots & \\ & & e^{-j\frac{2\pi \cdot (N-1)}{N}} \end{pmatrix} \mathbf{DFT}_N \tag{12}$$

where $\mathbf{DFT}_N = (1/\sqrt{N})\left[\omega_N^{kn}\right]$, $\omega_N = \exp^{-j(2\pi/N)}$, is the discrete Fourier matrix. The inverse $\mathbf{DFT}_N^{-1} = \mathbf{DFT}_N^H$ is the matrix of eigenvectors of $\mathbf{A}_c$. The eigenvalues of $\mathbf{A}_c$ are $e^{-j(2\pi n/N)}$, $n = 0, \ldots, N-1$, the diagonal entries of the middle matrix in (12). The graph Fourier transform (GFT) follows by analogy with (12). From the eigendecomposition of $\mathbf{A}$ in (8), the graph Fourier transform is the inverse of the matrix $\mathbf{V}$ of eigenvectors of the shift $\mathbf{A}$

$$\mathbf{F} = \mathbf{V}^{-1}. \tag{13}$$

The eigenvectors of the shift $\mathbf{A}$, columns of $\mathbf{V}$, are the graph spectral components, and the eigenvalues of $\mathbf{A}$, the diagonal entries $\lambda_k$ of matrix $\Lambda$ in (8), are the graph frequencies. The graph frequencies are complex valued for a general nonsymmetric (directed graph) shift $\mathbf{A}$.

The graph Fourier transform of graph signal $\boldsymbol{s}$ is given by the graph Fourier analysis decomposition

$$\hat{\boldsymbol{s}} = \mathbf{F}\boldsymbol{s} = \mathbf{V}^{-1}\boldsymbol{s} = \left[f_0\,\boldsymbol{s} \ldots f_{N-1}\,\boldsymbol{s}\right]^\top \tag{14}$$

where $f_k$ is a row vector, the $k$-th row of $\mathbf{F}$. The graph Fourier coefficients or graph spectral coefficients of signal $\boldsymbol{s}$ are computed using the inner product as $\hat{\boldsymbol{s}}(\lambda_k) = \hat{s}_k = f_k\,\boldsymbol{s} = \langle f_k^H, \boldsymbol{s}\rangle$. Then, the Fourier spectral decomposition of the signal is obtained by the graph inverse Fourier transform. Equivalently, it is given by the graph Fourier synthesis expression

$$\begin{aligned} \boldsymbol{s} &= \mathbf{F}^{-1}\hat{\boldsymbol{s}} = \mathbf{V}\hat{\boldsymbol{s}} \\ &= \sum_{k=0}^{N-1} \hat{\boldsymbol{s}}_k v_k \\ &= \sum_{k=0}^{N-1} \langle f_k^H, \boldsymbol{s}\rangle v_k \\ &= \mathbf{V}\left[\langle f_0^H, \boldsymbol{s}\rangle \ldots \langle f_{N-1}^H, \boldsymbol{s}\rangle\right]^\top. \end{aligned} \tag{15}$$

The eigenvectors $v_k$ of $\mathbf{A}$, columns of $\mathbf{V}$, are the spectral components. Equation (15) synthesizes the original signal $\boldsymbol{s}$ from the spectral components $v_k$; the coefficients $\hat{s}_k$ of the decomposition are the spectral coefficients of $\boldsymbol{s}$.

## D. Interpreting Graph Frequencies

We can now interpret filtering a graph signal (i.e., multiplying the corresponding vector by $\mathbf{H}$) in the spectral domain. From (9), the output of $\mathbf{s}_{\text{in}}$ to filter $h$ is successively

$$\begin{aligned} s_{\text{out}} &= \mathbf{H} \cdot \boldsymbol{s}_{\text{in}} \\ &= \mathbf{V}h(\Lambda)\underbrace{\left(\mathbf{V}^{-1}\boldsymbol{s}_{\text{in}}\right)}_{\text{Fourier transf.}} \qquad (16) \\ &= \mathbf{V}\underbrace{\operatorname{diag}\left[h(\lambda_0)\ldots h(\lambda_{N-1})\right]\hat{\boldsymbol{s}}_{\text{in}}}_{\text{Filtering in graph Fourier space}} \\ &= \underbrace{\mathbf{V}\operatorname{diag}\left[h(\lambda_0)\,\hat{\boldsymbol{s}}_{\text{in}_0}\ldots h(\lambda_{N-1})\,\hat{\boldsymbol{s}}_{\text{in}_{N-1}}\right]^\top}_{\text{Inverse Fourier transf.}}. \qquad (17) \end{aligned}$$

Thus, according to (16), filtering by $\mathbf{H}$ can be performed by first taking the graph Fourier transform of the input $(\mathbf{V}^{-1}\boldsymbol{s}_{\text{in}})$, followed by pointwise multiplication in the frequency domain of the graph Fourier transform signal $\hat{\boldsymbol{s}}_{\text{in}}$ by the filter frequency response $\operatorname{diag}\left[h(\lambda_0)\ldots h(\lambda_{N-1})\right]^\top$ given by (17). Finally, an inverse graph Fourier transform computes the output back in the graph node domain. This is the graph Fourier filtering theorem that reduces graph filtering to two graph Fourier transforms and a pointwise multiplication in the spectral domain [2].

With a notion of frequency we can now consider the GSP equivalents to classical concepts of low-, high-, and bandpass signals or filters, as well as the question of efficient filter design. In the classical time domain, these concepts are directly related to values of frequency. In the time domain, the frequency is actually defined from the eigenvalues of the cyclic shift $\mathbf{A}_c$ in (12) as

$$\Omega_k = \frac{2\pi k}{N},\ k = 0, 1, \ldots, N-1.$$

These frequencies are directly related to the degree of variation of the spectral components. For example, the lowest frequency $\Omega_0 = 0$ corresponds to the least varying spectral component, the constant or DC-spectral component, the next frequency $\Omega_1 = (2\pi/N)$ represents a higher variation spectral component, and so on. There is a nice one-to-one correspondence between the ordered value of the frequency and the corresponding degree of variation or complexity of the time spectral component.

In GSP, the frequencies are defined by the eigenvalues of the shift. We can order the graph frequencies by relating them to the complexity of the spectral component. For example, this can be measured by the total variation of the associated spectral component through

$$\mathbf{TV}_{\mathrm{G}}(\boldsymbol{v}_k) = \left\|\boldsymbol{v}_k - \mathbf{A}^{\text{norm}}\boldsymbol{v}_k\right\|_1$$

where $\|\cdot\|_1$ is norm 1, and $\mathbf{A}^{\text{norm}} = (1/|\lambda_{\max}|)\mathbf{A}$. Other norms could be used to define the total variation; see [3] and [51]. Using this, graph frequency $\lambda_m$ is larger than graph frequency $\lambda_\ell$ if

$$\mathbf{TV}_{\mathrm{G}}(\boldsymbol{v}_m) > \mathbf{TV}_{\mathrm{G}}(\boldsymbol{v}_\ell).$$

Assuming that the graph frequencies have been ordered from low to high, graph signal $\boldsymbol{s}$ is low-pass if its graph

Fourier coefficients are zero for $\Omega_k$, $k > \ell$, for some $\ell$, $0 \leq \ell < N-1$. We can similarly define band- and high-pass signals and filters.[6]

### E. Frequency Representations Based on the Laplacian

The notions of frequency that arise in conventional signal processing provide a sound mathematical and intuitive basis for analyzing signals. While it is mathematically possible, as just discussed, to define notions of frequency for graph signals, developing a corresponding intuition to understand these elementary frequencies is not as straightforward. For the total variation criterion it has been shown experimentally and justified theoretically that the frequency bases obtained from the shift operator tend to be ordered [51].

Up to this point, we have focused primarily on frequency representations derived from the adjacency matrix of a graph, an approach that can be applied to both directed and undirected graphs, and can be linked to DSP concepts in the case of the cycle graph. A frequency representation can be similarly built on top of the Laplacian matrix of an undirected graph. Since this matrix is positive semidefinite, all the eigenvalues are real and nonnegative, and a full set of orthogonal eigenvectors can be obtained, so that we can write

$$\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^\top \tag{18}$$

with $\mathbf{U}$ the GFT matrix, which is real valued and orthogonal in this case. Because the eigenvalues are real, they provide a natural way to order the GFT basis vectors in terms of frequency (the variations of their values on the graph). In this case, the eigenvalue/eigenvector pairs can be viewed as successive optimizers of the Rayleigh quotient, where the $k$-th pair $\lambda_k, \mathbf{u}_k$ solves

$$\mathbf{u}_k = \arg_{\mathbf{x}^\top \mathbf{u}_{k'}=0,\, k'=0,\ldots,k-1} \min \frac{\mathbf{x}^\top \mathbf{L}\mathbf{x}}{\mathbf{x}^\top \mathbf{x}} \tag{19}$$

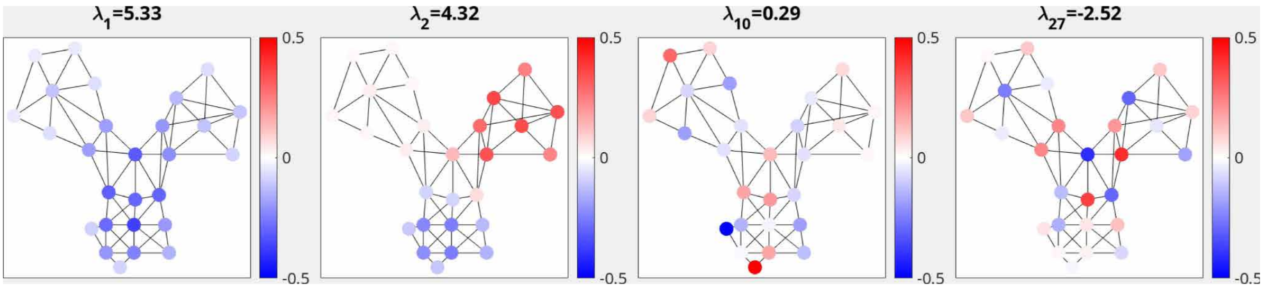
with $\lambda_k = \mathbf{u}_k^\top \mathbf{L}\mathbf{u}_{\mathbf{u}_k}$, if $\mathbf{u}_k$ is normalized. Thus, for the explicit variation metric induced by the Laplacian quadratic form, the GFT provides an orthogonal basis with increased variation, and such that, from (19), each additional basis vector minimizes the increase in variation while guaranteeing orthogonality. More generally, the relationships between eigenvectors and eigenvalues of the Laplacian and the structure of a graph are part of a deep and beautiful domain of mathematics known as spectral graph theory [12]. When graphs have structures closely related to those used in DSP (e.g., circulant adjacency matrices [85]) frequency interpretation is clear. If the graph is more general than the ring graph, part of the intuition remains, as illustrated by Fig. 3. Indeed, eigenvectors $u_i$ are oscillating over the vertex set. As the eigenvalue index $i$ increases, the number of oscillations tends to increase as well [86]. However, the irregular nature of graphs means that the analogies to DSP cannot always be extended easily. For example, the spacing between frequencies (as measured by the eigenvalues of the Laplacian, for example) can be highly irregular, or some frequencies may have high multiplicity. Also, the high-frequency eigenvectors of irregular graphs can be highly localized [87], [88]. This potentially indicates that a direct ordering of frequencies may be insufficient to fully understand signal decompositions induced by current GSP techniques. To be complete, note that, while Laplacians can be easily defined for undirected graphs, there has been work to introduce definitions appropriate for directed graphs as well [89], [90]. In summary, a full understanding of the best frequency representation for a specific GSP application, as a function of the type of graph considered, is still an active research topic. This is discussed further in Section III-A.

### F. Implementation

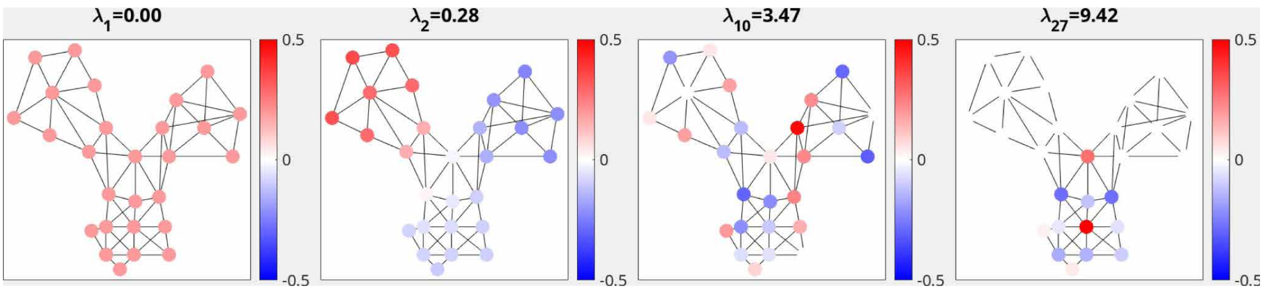
Finally, let us quickly touch on the issue of computational complexity of the filtering operation. A straightforward algorithm would consist in computing the GFT matrix $\mathbf{V}$ and explicitly applying it to the input signal as in (16). This is simple and accurate for small graphs thanks to fast SVD algorithms. Partial SVD can also be used if the filter $h$ should only be evaluated on the top or bottom eigenvalues [91]. In general, and for large graphs, it is better to avoid computing even a partial SVD. One efficient possibility is to compute a polynomial approximation to $h$ with Chebyshev filters [66]. For large but sparse graphs, this reduces computations to sparse matrix–vector multiply, which is very efficient.

Furthermore, filter implementation via polynomial approximation can be interpreted in terms of localization in the vertex domain. Note that when the input signal is a perfect impulse located at a given vertex $\mathbf{s} = \mathbf{e}_i$, the filtered signal depends only on the graph filter and the vertex location in the graph: $\mathbf{f}_i = \mathbf{H}\mathbf{e}_i$. Even though $\mathbf{f}_i$ changes with the chosen vertex, it was proved in [92] that this signal is localized around $i$ in a way that only depends on the smoothness of the filter $h$. This is interesting because it allows to design filters that act locally and in a controlled way over the vertex set. After a filter $h(\lambda)$ is chosen, one can choose an approximation $h_k(\lambda)$, a polynomial of degree $k$ in $\lambda$. Note that $h_k(\lambda)$ can then be implemented as shown in (9) by applying a polynomial of the shift operator. This does not require knowing the eigenvalues and eigenvectors associated to the graph, so that it is possible to process signals on very large graphs locally, by processing $k$-hop neighborhoods of nodes in the vertex domain, without a need to find the graph spectrum first.

[6]The total variation is the $l1$-norm of a vector multiplied by $\mathbf{I} - \mathbf{A}^{\text{norm}}$. Assume for simplicity that the graph is undirected, then the largest eigenvalue of $\mathbf{I} - \mathbf{A}^{\text{norm}}$, and thus the largest TV, will be $1 - \lambda_{\min}/|\lambda_{\max}|$, where $\lambda_{\min}$ is the smallest eigenvalue of A, which intuitively, as seen in Fig. 3, corresponds to high variation in the eigenvector.




(a)




(b)

**Fig. 3.** ***Example of elementary frequencies obtained from different algebraic representations of the same graph. (a) Adjacency matrix. (b) Laplacian matrix. In each case, four different frequencies are shown, corresponding to different eigenvalues, ranging from lowest frequency to highest frequency. In the Laplacian case, the lowest frequency is $\lambda = 0$, representing a constant value throughout the graph, and the highest is $\lambda = 4.53$, where we can observe a large number of sign changes across graph edges. Note that for any given graph with N nodes we will have N eigenvectors that can be ordered in terms of their variation covering the whole range of frequencies for that graph. In this example, the graph is unweighted. Unlike conventional signal processing, some of the eigenvectors can be localized in the graph (e.g., the highest frequency eigenvector of the Laplacian).***

## III. STATE OF THE ART AND CHALLENGES

### A. Frequency Definition

One can guarantee the existence of an orthogonal basis for any undirected graph. Thus, once a graph has been chosen (see Section III-E) a definition of frequency is readily available, which allows us to address other questions considered in this section (sampling, signal representation, etc.). Multiple choices are possible, as a function of the graph type, the selected shift operator and its normalization, etc. Making these choices appropriately for a given application remains an open question, which is actively being investigated.

As an example, the eigenvalues of the chosen operator matrix (Laplacian or adjacency) can have high multiplicity. In this situation, a graph with $N$ nodes will have fewer than $N$ unique frequencies. A particular concern is that one can choose any set of orthogonal vectors within the subspace corresponding to this frequency, leading to different GFTs and thus potentially irreproducible results. As a way to address this scenario, recent work [93] suggests using oblique projections to measure the energy within such a subspace, using this information to represent the overall energy at that frequency.

For directed graphs, additional problems arise given that a full set of eigenvectors may not exist. Results for directed graphs are often restricted to cases where the adjacency matrix is invertible and eigenvectors do exist (as discussed in Section II-C). If these conditions do not hold, the Jordan canonical form is used to obtain the GFT [2], but this is well known to be a numerically unstable procedure. As an alternative, some authors have proposed to approximate directed graphs by undirected ones, using approaches such as the hub authority model [94], [95]. Recent work has also considered alternative definitions of frequency. For example, the work in [96] advocates using the random walk Laplacian normalization, while in [97], the authors propose alternative choices of a graph signal inner product and explore the resulting frequency definitions. Other techniques make use of explicit optimization to choose a set of graph frequencies. As an example, the work in [98] uses an optimization procedure to construct explicitly an orthogonal basis set that minimizes a quantity related to the cut size. With this approach, successive eigenvectors provide increasingly higher frequencies in the sense of corresponding to higher cut costs, while being orthogonal to those eigenvectors previously selected. The work in [99] also uses optimization techniques with a different criterion to define a set of frequencies associated to a graph. In summary, this is a very active area of research, and the best approach to define a set of frequencies for graphs in a specific application remains to some extent an open question.

### B. Representations

Designing representations for graph signals having desirable properties (e.g., localization, critical sampling, orthogonality, etc.) has been one of the first and most important research goals in GSP. Pioneering contributions [61] and [100] provided early examples of designs based on vertex domain and spectral domain characteristics, respectively. Vertex domain designs such as [100] or [101] have the advantage of defining exactly localized basis functions on the graph, but do not have a clear spectral interpretation. Conversely, diffusion wavelets [61] are defined in the spectral domain, but do not guarantee exact vertex domain localization (only energy decay properties). The spectral graph wavelet transform design [66] was the first to combine a spectral design with vertex-domain localization, by defining smooth filter kernels in the spectral domain and approximating these with polynomials.

The filterbanks developed in [66] were not critically sampled, unlike [61] or [101]. Thus, much recent work has focused on developing critically sampled filterbanks having both a spectral interpretation and vertex localized implementation. These types of filterbanks have been designed for bipartite graphs [68], [102], thus requiring the graph to be decomposed into a series of bipartite subgraphs [68], [103]. An alternative approach proposed in [85] and [104] can be applied to circulant graphs, for which the GFT corresponds to the DFT. Recent work [105], [106] has shown that similar filterbank designs can be developed for directed graphs, where these designs are only possible for $M$-block cyclic graphs, which play a similar role to that of bipartite graphs in the undirected case. Note that in all these cases, critical sampling combined with polynomial analysis and synthesis filtering is restricted to specific types of graphs (bipartite, $M$-block cyclic, and circulant.) Note also that critical sampling with polynomial analysis and synthesis filters on undirected graphs can only be achieved in the bipartite case [107].[7] Ongoing work is focusing on 1) providing better tools to characterize $M$-block cyclic graphs, including, for example, the definition of polyphase representations [105], [106], [108], [109]; 2) development of improved filters by exploiting conventional filter designs and/or relaxing the critical sampling requirement [110]–[113]; and 3) novel approaches for downsampling, e.g., frequency domain techniques [114], that allow extending critically sampled filterbanks to nonbipartite graphs.

[7]Under some conditions on the analysis filters, critical sampling and perfect reconstruction can be achieved for any graph, but this requires a synthesis operation corresponding to an $N \times N$ matrix multiplication, which may not be practical for large graphs [107]. As an example, the approach in [85] guarantees invertibility but reconstruction is nonpolynomial.

While much of the work to date has focused on representations with bases functions selected in terms of frequency content (e.g., low-pass versus high-pass bases), some recent work is also exploring representations for piecewise smooth signal models [4]. The design of representations that adapt to the specific properties of graph signal classes has further been addressed from the viewpoint of dictionary learning [115]–[117]. The main objective is to design dictionary of atoms that are able to sparsely represent signals on graphs while incorporating the structure of the graph.

### C. Sampling

The problem of sampling signals on graphs is modeled on the corresponding problem in conventional signal processing. The basic idea is to define a class of signals (for example, signals that are bandlimited to the first $K$ frequencies of the GFT) and then define necessary and sufficient conditions to reconstruct a signal in that class from its samples. The first problem formulation and a sufficient condition for unique recovery were presented in [118]. A necessary and sufficient condition for unique recovery in undirected graphs was introduced in [119], and, subsequently, several papers proposed solutions for different aspects of the problem [120]–[122]. In particular, sampling results have been generalized to directed graphs [121], [122] and to other classes of signals such as piecewise smooth signals [123].

A key difference when comparing sampling in conventional signal processing and in the context of graph signals is the lack of "regular" sampling patterns in the latter. The lack of regularity in the graph itself prevents us from defining the idea of sampling "every other node." Thus, multiple approaches have been suggested to identify the most informative vertices on a graph so that these can be sampled. While the sampling problem is formalized based on the assumption that signals to be sampled belong to a certain class (e.g., bandlimited), in practice, these can never be guaranteed and thus the observed signals will be noisy and, in general, will not belong to the prespecified class. To address this problem, several methods approach the problem of sampling set selection from an experiment design perspective [121], [122], [124], setting as a goal to identify a set of vertices that minimizes some measure of worst case reconstruction error in cases where noise or model mismatch is present. The measure can also be mean squared reconstruction error instead of worst case in the experiment design paradigm [122].

Complexity is a key challenge in sampling set identification, especially for large-scale graphs. Some techniques require computing and storing the first $K$ basis vectors of the GFT [121]. For larger graph sizes, where this may not be practical, the approach in [122] uses spectral proxies instead of exact graph frequencies leading to lower complexity. To reduce complexity even further, the work in [125] proposes a random sampling technique where the probability of selecting a given vertex is based on a locally computed metric. This leads to significantly lower complexity but, as a random sampling technique, it may not always lead to performance comparable to those of more complex greedy optimization methods such as [121] and [122].

Given the samples of a graph signal, the next objective is to reconstruct an estimation of the signal at the nodes that were not sampled (observed). Reconstruction algorithms based on polynomial filters approximating ideal reconstruction filters have been proposed in order to reconstruct an estimated signal on the whole graph based on the observed vertex measurements [126], [127].

While theoretical aspects of graph signal sampling are by now well understood, the relevance of proposed techniques to practical applications is still an open question. A key challenge in this regard is to identify what are relevant signal models for real data sets, while potentially adapting proposed generic sampling methods to specific types of graphs (e.g., exploiting properties of nearly regular graphs).

### D. Extending Conventional Signal Processing to Graph Signals

Challenges in extending ideas and concepts from conventional signal processing to signal processing on graphs can be further exemplified by research into notions of stationarity and localization. For conventional time signals, a test for stationarity can be based on determining whether time shifts affect the statistical properties of a signal or, equivalently, observing a signal at different times. However, these two views are not equivalent for finite dimension graphs: we can observe a given signal at different nodes, but this is not necessarily the same as "shifting" the signal while observing it at always at the same node. For graphs with $N$ vertices, shifting can be defined via a spectral domain operator [66]; or, instead, the graph shift based on the adjacency matrix can be used. Some authors have proposed a definition of stationarity based on spectral properties of the vertex shift operator [128]. To overcome challenges associated to existing shift operators, one solution, first proposed by [129], is to introduce alternative graph shift operators (see also [52]) or localization operators that have both a spectral interpretation and vertex domain localization [130], [131]. Notions of stationarity can help develop probabilistic GSP methods leading to graph-based Wiener filtering [131], [132].

A study of vertex/spectral localization and uncertainty principles was first developed by [133], where it was shown that, in general, it is not possible to achieve arbitrarily good localization in both spectral and vertex domains simultaneously. However, a limitation in this study was that bounds had to be derived for individual vertices. More recently, [134] has shown that for graph signals it is in fact possible to have compact support in both spectral and vertex domain (something that can never occur in conventional signal processing). As was already noted in Section II-E, this occurs due to the irregular nature of graphs: for example, a graph consisting of several loosely connected clusters is likely to lead to some columns of $\mathbf{V}$ having nonzero entries only in some of the clusters. Other contributions, such as [135] and [136], have also explored the challenges in directly extending the concept of an uncertainty principle to graph signals, while other recent work considers alternative frequency representations that can take into consideration the specific localization properties encountered in irregular graphs [137]–[139]

Work in these two areas shows that direct extensions of signal processing concepts to graphs are not straightforward, and thus further research is still needed to develop techniques that can provide insights about graph signal behavior (localization, stationarity) while accommodating key characteristics of graphs (e.g., irregular node connectivity and spectral characteristics).

### E. Graph Learning

Much recent work on GSP assumes that the graph is given or can be defined in a reasonable way based on the nature of the application. As an example, in communication or social networks, the connectivity of the network (directed or undirected) can be used to define the graph. In other applications, edge weights between nodes can be chosen as a decreasing function of distance, e.g., physical distance between sensors in the case of sensor networks or distance in feature space in the case of learning applications [5], [140], [141].

Recent work has been considering alternative techniques where the goal is to learn graphs from data. This work is motivated by scenarios where 1) no reasonable initial graph exists or 2) it is desirable to modify a known graph (based on network connectivity for example) by selecting weights derived from data. The key idea in these approaches is to select a graph such that the most likely vectors in the data (the graph signals) correspond to the lowest frequencies of the GFT or to the more likely signals generated by Gauss Markov random field (GMRF) related to the graph.

Examples of approaches based on smoothness include [43], [142], and [143], while representative methods based on the GMRF model are [44] and [144]. The basic idea in the latter approaches is to identify GMRF models such that the inverse covariance (precision) matrix has the form of a graph Laplacian (e.g., combinatorial or generalized). Note that this work extends popular approaches for graph learning (e.g., graphical Lasso [145]) to precision matrices restricted to have a Laplacian form (corresponding to a graph with positive edge weights). Other approaches have addressed graph selection under the assumption that the observed data were obtained through graph-based diffusion. Examples of these approaches include [146]–[149]. While not explicitly a graph learning problem, the related question of blind identification of graph filters has also been studied [150].

There remain several major challenges in the development of graph learning methods. Graphs derived from data are essentially models, and as such the "right" graph model should be selected based on the number of parameters it uses, its data fit, and its ability to provide useful interpretations. While a sparsity criterion addresses some of these requirements, other constraints may also be important. For example, it will be useful to develop methods to select graphs with specific topology properties [151], spectral properties (eigenvalue distribution, eigenvector localization), or even computational properties (e.g., leading to GFTs with lower computation cost.)

## IV. GRAPH SIGNAL PROCESSING APPLICATIONS

Networks are present in very different application domains, where graphs can provide a generic representation of the structure present in the data sets. In this section, we discuss a wide set of applications where the GSP framework has been used. We consider four different types of scenarios, where both the scale and the domain of the networks considered are very different. We start with physical networks, including both large scale networks (sensor networks in Section IV-A) and human-scale ones (biological networks in Section IV-B), where the goal is to use measurements to better understand physical phenomena. We then consider "logical" networks, where GSP is introduced as an alternative for existing processing techniques for conventional signals (images and point clouds in Section IV-C), or as a tool to analyze large scale data sets (machine learning and data science applications in Section IV-D). In each of these cases, we provide a few, nonexhaustive, examples to highlight the different types of domains and graph representations that have been studied. More detailed discussion of graph-based techniques in specific domains are considered in other papers in this special issue [152].

### A. Sensor Networks

One of the most natural applications of GSP is in the context of sensor networks. A graph represents the relative positions of sensors in the environment, and the application goals include compression, denoising, reconstruction, or distributed processing of sensor data. Indeed, some of the initial explorations of graph-based processing focused on sensor networks [64], [65], [153], [154].

A first approach to define a graph associated to a sensor network is to choose edge weights as a decreasing function of distance between nodes (sensors). Then, data observations that are similar at neighboring nodes lead naturally to a smooth (low-pass) graph signal. Such a smooth graph signal model makes it possible to detect outliers or abnormal values by high-pass filtering and thresholding [51], [155], or to build effective signal reconstruction methods from sparse set of sensor readings, as in [156]–[158], which can potentially lead to significant savings in energy resources, bandwidth, and latency in sensor network applications.

A second scenario is where the graph to be used for data analysis is given by the application. For example, urban data processing relies on data that naturally live on networks, such as energy, transportation, or road networks. In these applications cases, GSP has been used to monitor urban air pollution [159], or to monitor and analyze power consumption [160], for example. Some works such as [161]–[163] have used GSP tools for analyzing traffic and mobility in large cities. For example, wavelets on graphs can serve to extract useful traffic patterns to detect disruptive traffic events such as congestion [164]. Graph wavelet coefficients at different scales permit to infer useful information such as origin, propagation, and the span of traffic congestion.

In some cases, relations between sensor readings are not exclusively explained by the distance between sensor locations, or by some actual network constraints. Other factors can influence the data values observed at the sensor readings such as the presence of geographical obstacles (e.g., in temperature measurements), or the interaction between networks of different types (e.g., how proximity to a freeway affects pollution measurements in a city). In some cases, the phenomena that can explain these relations between measurements are latent and this leads to the challenging problem of learning a graph (see also Section III-E) that can explain the data observations under signal smoothness or other signal model assumptions [43], [142], [149]. This allows inferring system features and behaviors that are hidden in the measured data sets (e.g., ozone data sets in [165]).

Finally, several of the GSP operators presented in this paper are amenable to distributed implementations that are particularly interesting for large sensor networks, and which motivated some of the early work mentioned at the beginning of this section. For example, the graph multiplier operators can be approximated by Chebyshev polynomials in distributed implementation of smoothing, denoising, inverse filtering, or semisupervised learning tasks [166]. The work in [167], for example, studies the problem of distributed reconstruction of time-varying bandlimited graph signals recorded by a subset of temperature sensor nodes. There is, however, still a lot of opportunities for the development of distributed GSP algorithms that are able to extend to large-scale networks and big data applications.

### B. Biological Networks

Biological networks have also proved to be a popular application domain for GSP, with recent research works focusing on the analysis of data from systems known to have a network structure, such as the human brain, and also on the inference of *a priori* unknown biological networks.

Several works have studied human brain networks using the GSP framework. For example, it has been observed that human brain activity signals can be mapped on a network (graph) where each node corresponds to a brain region. The network links (edge weights) are considered to be known *a priori* and represent the structural connectivity or the functional coherence between brain regions [170], [171]. GSP tools such as the graph signal representations described in Section III-B can then be used to analyze the brain activity signal on the functional or structural brain network. For example, low frequencies in the graph signal represent similar activities in regions that are highly connected in the functional brain networks, while high frequencies denote very different activities in such brain regions.

These ideas have been used to analyze brain signals, leading to biologically plausible observations about the behavior the human cognitive system, as in, for example, [168] and [169]. Fig. 4 illustrates the signal distribution of different graph frequency components in an active motor learning task. Interestingly, regions with strong signal in low and high graph frequency components overlap well with the regions known to contribute to better motor learning [172]. Additionally, it has been observed that there is a strong association between the actual brain networks (characterized by their spectral properties) and the level of exposure of subject to different tasks [173]. Some works further build on the multiresolution properties of spectral graph wavelet transforms to capture subtle connected patterns of brain activity or provide biologically meaningful decompositions of functional magnetic resonance imaging (fMRI) data [174]–[176]. Interestingly, it is also possible to combine different sources of information in the analysis of the brain networks. For example, the work in [177] integrates infra-slow neural oscillations and anatomical-connectivity maps derived from functional and diffusion magnetic resonance imaging (MRI), in a multilayer-graph framework that captures transient networks of spatiotemporal connectivity. These networks group anatomically wired and temporary synchronized brain regions and encode the propagation of functional activity on the structural connectome, which contributes to a deeper understanding of the important structure–function relationships in the human brain.

The GSP framework has also been proposed for the classification of brain graph signals [178] and the analysis of anomalies or diseases [179], [180]. For example, source localization algorithms based on sparse regularization can be used to localize the possible origins of Alzheimer's disease based on a large set of repeated MRI scans. This can help understand the dynamics and origin of dementia, which is an important step toward developing effective treatment of neurodegenerative diseases [181]. The growing number of publications studying brain activity or brain network features from a GSP perspective indicates that these are promising applications for the methods described in this paper.

It should finally be noted that brain networks are not the only biological networks where GSP offers promising solutions. GSP elements and biological priors are combined to infer networks and discover meaningful interactions in gene regulatory networks, as in [182] and [183]. The inference of the structure of protein interaction networks has also been addressed with help of spectral graph templates [148]. In particular, the observed matrix of mutual information can be approximated by some (unknown) analytic matrix function of the unobserved structure to be recovered. Observed data are then used to obtain the eigenvectors of the matrix representation of the graph model, and then the eigenvalues are estimated with the help of sparsity priors. The above examples are only some illustrations of the recent works that attempt to infer structures of biological networks using a GSP perspective. Biological networks that cannot be explicitly recorded and measured are potentially good applications for graph learning and inference methods in particular, which can uncover unknown interactions in the biological data.

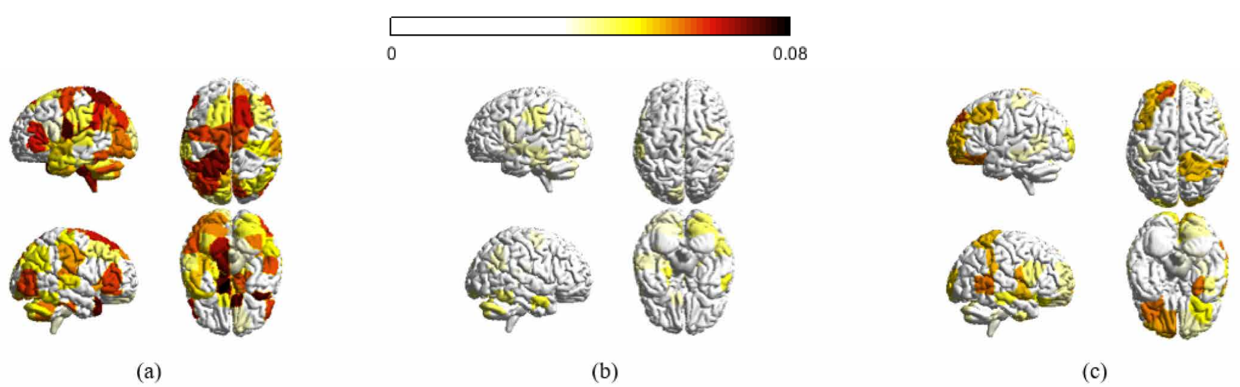



**Fig. 4.** ***Distribution of decomposed signals. (a)–(c) Absolute magnitudes for all brain regions with respect to graph low frequencies, graph middle frequencies, and graph high frequencies, respectively. Higher concentration in graph low frequency results in better learning performance, when subjects are unfamiliar with the task [168]. Concentration in graph low frequency also helps faster response in switching attention between actions [169]. From [168], with permission.***

### C. Image and 3-D Point Cloud Processing

While GSP is often applied to data sets that naturally exhibit irregular structures, it has also been applied to other data sets where conventional signal processing has been used for many years, including, for example, images and video sequences. An image to be processed can be viewed as a set of pixels, each associated to a vertex, forming a regular graph with all edge weights equal to 1 (e.g., a line graph or a grid graph). Indeed processing using the discrete Fourier transform or the discrete cosine transform (DCT) can be shown to have a simple interpretation in terms of the frequencies associated to those regular graphs [184] (see also Section I-B). Instead, recent work uses regular line and grid graph topologies, but with unequal edge weights that can adapt to the specific characteristic of an image or a set of images.

A first set of approaches associates a different graph to each image, by associating smaller edge weights to connect pixels that are on opposite sides of an image contour. This type of image-dependent graph representation is strongly connected to popular image processing techniques, such as the bilateral filter and related methods [73], which also apply signal-dependent filtering and are widely used in applications such as image restoration or denoising. Graphs are used to capture the geometric structure in images, such as contours that carry crucial visual information, in order to avoid blurring them during the filtering process. In addition to works that effectively extend image priors such as total variation (TV) minimization to graph representations (e.g., [185] and [186]), other works such as [187] or [188] use more specific GSP operators for denoising or filtering. In particular, the authors in [187] use graph spectral denoising methods to enhance the quality of images, while the work in [188] uses graph-based filters that influence the strength and direction of filtering for effective enhancement of natural images.

A second avenue of research has considered situations where a graph is constructed as an efficient representation for a set of images, in particular, in the context of image and video compression applications. The Karhunen–Loève transform (KLT) is known to provide the best transform coding gains under the assumption that the signals can be modeled as stationary Gaussian processes (which is often a good assumption for images). Indeed, extensive use of the DCT is often justified because it is optimal for a GMRF with correlation 1, which is an appropriate model for natural images. The inverse covariance matrix, or precision matrix, then corresponds to a line graph with equal weights. From this perspective, graph learning approaches can be used to learn precision matrices with structures and weights that capture statistics of specific types of images. For example, piecewise smooth images can be compressed using suitable graph Fourier transforms (GFTs), which can be adapted to different types of image pixel blocks [189], [190]. Graph-based transforms have also been used to code motion-compensated residuals in predictive video coding [191] with effective rate-distortion performance.

New visual modalities such as 3-D meshes or 3-D point clouds where data are sampled in irregular locations in 3-D space, lend themselves naturally to graph representations. The color or 3-D information supported by nodes or voxels is connected to its nearest neighbors to form a graph. Graph-based transforms can then be used to compress the resulting graph signals in static or dynamic point clouds [192], [193]. In particular, the temporal redundancy between 3-D point cloud frames at different instants can be effectively estimated with help of graph spectral features [192], as illustrated in Fig. 5. Graph-based transforms permit to properly exploit both the spatial correlation inside each frame and the temporal correlation between the frames, which eventually results in effective compression. Compression, however, is not the only application of GSP in 3-D point clouds. Fast resampling methods, which are important in processing, registering, or visualizing large point clouds, can also be built on graph-based randomized strategies to select representative subsets of points while preserving application-dependent features [194].

### D. Machine Learning and Data Science

Graph methods have long played an important role in machine learning applications, as they provide a natural way to represent the structure of a data set. In this context, each vertex represents one data point to which a label can be associated, and a graph can be formed by connecting vertices with edge weights that are assigned based on a decreasing function of the distance between data points in the feature space. GSP then enables different types of processing, learning, or

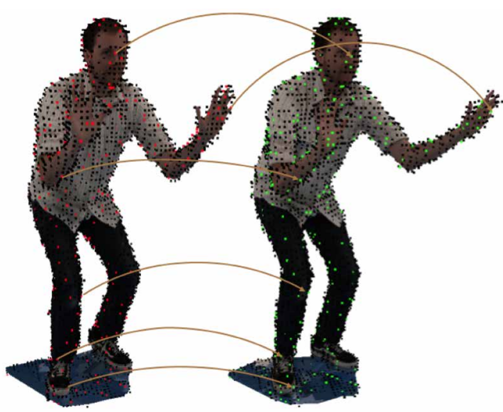


**Fig. 5.** ***Example of motion estimation in a 3-D point cloud sequence. Each frame is represented as a graph signal that captures the color and the geometry information of each voxel. Graph spectral features at each voxel capture the local graph signal properties and permit to find correspondences between frames at different instances. A subset of the correspondences between the target (red) and the reference frame (green) are highlighted between small cubes that correspond to voxels. From [192], with permission.***

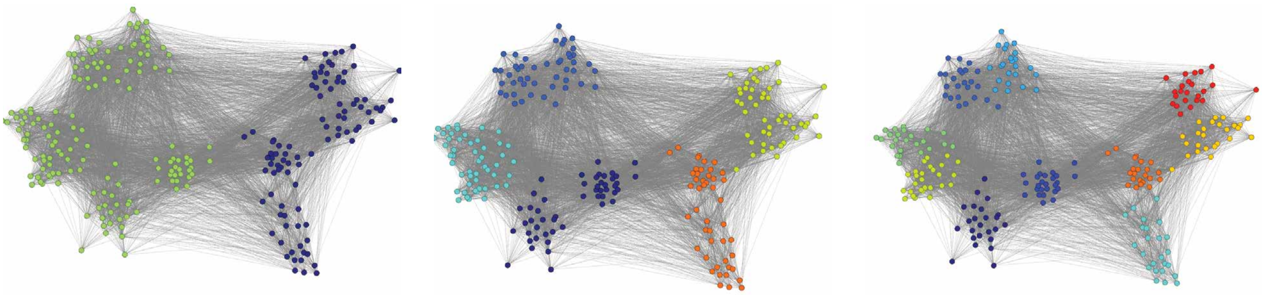


**Fig. 6.** ***Multiscale community structures in a graph of social interactions between children in a primary school. The different figures show the partition of the original social network in two, five, and ten communities, respectively. From [199], with permission.***

filtering operations on values associated to graph vertices. In a different context, GSP elements can be helpful to construct architectures to classify signals that live on irregular structures. We give below some examples of machine learning applications in both contexts.

When data labels are presented as signals on a (nearest-neighbor) graph, graph signal regularization techniques can be used in the process of estimating labels [5], optimizing the prediction of unknown labels in classification [51] or semisupervised learning problems [195]. Furthermore, as labeled samples are often a scarce and expensive resource in semisupervised learning applications, graph sampling strategies such as those presented in Section III-C can be helpful in determining the actual needs for labeled data and develop effective active learning algorithms [141].

Graphs can also be constructed to describe similarities between users or items in recommendation systems that assist customers in making decisions by collecting information about how other users rate particular services or items [196]. Leveraging the notions of graph frequency and graph filters, classical collaborative filtering methods (such as $k$-nearest-neighbors strategies), can then be implemented with specific band-stop graph filters on graphs [197]. Furthermore, linear latent factor models, such as low-rank matrix completion, can be viewed as bandlimited interpolation algorithms that operate in a frequency domain given by the spectrum of a joint user and item network. This can serve to design effective graph filtering algorithms that lead to enhanced rating prediction in video recommendation applications, for example, [197]. Content-based recommendation can also be addressed as an online learning problem solved with spectral bandit algorithms [198]. The key idea is to represent the reward function in an online recommendation system as a linear combination of the eigenvectors of the similarity graph that connects the different items. With this representation it is possible to optimize the reward function by favoring smoothness on the graph, which has been shown to be effective in video recommendation examples [198].

Data clustering or community detection can also benefit from tools developed under the GSP framework. For example, graph transforms, and especially graph wavelets, have been used to solve the classical problem of community detection [199]. The problem of detecting multiscale community in networks is cast as the problem of clustering nodes based on graph wavelets features. This allows the introduction of a notion of scale in the analysis of the network, as well as a sort of "egocentered" view of how a particular node "sees" the network (see Fig. 6). Furthermore, the extension of clustering or community detection tasks to large-scale systems generally relies on sampling or randomized strategy where GSP methods can also be very helpful. For example, fast graph-based filtering of random signals can be used to estimate the graph structure, and in particular to approximate eigenvectors that are often crucial in the design of clustering algorithms and other machine learning tasks. One of the initial works in this direction [200] proposes to use power methods (that can be shown to be related to graph filter operators) to speed up the computation of eigenvectors used in spectral clustering applications. More recently, a fast graph clustering algorithm that is provably as good as spectral clustering has been developed based on random signal filtering techniques [14]. Related ideas have been used in sketching [201], [202], data visualization applications on large real-world data sets of millions of nodes [194], [203], or in analysis of dynamic networks [204]. These examples provide evidence for the potential benefits of using GSP principles in big data applications.

Finally, the GSP framework can also be used to design architectures to analyze or classify whole graph signals that originally live on irregular structures. In particular, the GSP toolbox has been extensively used to extend convolutional deep learning techniques to data defined on graphs. The convolutional neural network paradigm has been generalized with help of GSP elements for the extraction of feature descriptors for 3-D shapes [205], [206]. A localized spectral network architecture leveraging on localized vertex-frequency analysis has also been proposed in [207], and the use of heat kernels defined in the graph spectral domain has been developed in [208]. While the previous works mostly address the analysis of 3-D shapes, convolutional neural networks (CNNs) can actually be extended to many other signals in high-dimensional irregular domains, such as social networks, brain connectomes, or words embedding, by reformulation in the context of spectral graph theory. Here, the GSP framework leads to the development of fast localized convolutional filters on graphs [209] along with adapted pooling operators [210]. Unsurprisingly, deep network architectures for graphs signals have been actually tested in various applications domains, such as chemical molecule properties prediction [211], classification tasks on social networks [212], autism spectrum disorder classification [213], or traffic forecasting [214].

## V. CONCLUSION

While recent papers have developed key principles for signal processing of graph signals, and these have shown significant promise for some important applications, there remain significant challenges. On the theoretical front, work to date has focused on results that can be applied to arbitrary graphs. But given the significant differences between the spectral properties of graphs, there is strong current interest in developing tools that can take into consideration the particular characteristics of specific classes of graphs. On the application front, GSP is a good match for data sets exhibiting irregular relationships between samples that can be captured by a graph. However, additional research is needed within each application to further understand the best ways to combine GSP tools with existing techniques in order to achieve significant gains in terms of the metrics of interest for each application. Finally, it is worth mentioning that many of the basic GSP tools described here are available in several Matlab/Python toolboxes: GSPBox [215], GraSP [216], and PyGSP [217].

## Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments, which helped improve the final version of the manuscript; the authors who kindly helped in the description of their work and permitted reproduction of their figures; and Dr. B. Girault (University of Southern California) for his careful reading and suggestions, and for his help with some of the figures included in the paper. ■

## REFERENCES

[1] M. Newman, *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010.

[2] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[3] 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," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[4] S. Chen, A. Singh, and J. Kovačević (2017). "*Multiresolution representations for piecewise-smooth signals on graphs*." [Online]. Available: https://arxiv.org/abs/1803.02944

[5] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.

[6] Committee on Network Science for Future Army Applications, "Network science," Nat. Res. Council Nat. Acad., Washington, DC, USA, Tech. Rep., 2005.

[7] K. Börner, S. Sanyal, and A. Vespignani, "Network science," *Annu. Rev. Inf. Sci. Technol.*, vol. 41, no. 1, pp. 537–607, 2007.

[8] T. G. Lewis, *Network Science: Theory and Applications*. New York, NY, USA: Wiley, 2011.

[9] A.-L. Barabási, *Network Science*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[10] M. O. Jackson, *Social and Economic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[11] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[12] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI, USA: AMS, 1996.

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

[14] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, "Compressive spectral clustering," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, 2016.

[15] C. Castellano and R. Pastor-Satorras, "Competing activation mechanisms in epidemics on networks," *Sci. Rep.*, vol. 2, Apr. 2012, Art. no. 371.

[16] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Control Syst.*, vol. 36, no. 1, pp. 26–46, Feb. 2016.

[17] A. Ganesh, L. Massoulié, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 2. Mar. 2005, pp. 1455–1466.

[18] C. T. Butts, "Revisiting the foundations of network analysis," *Science*, vol. 325, no. 5939, pp. 414–416, 2009.

[19] V. Colizza, A. Barrat, M. Barthélemy, and A. Vespignani, "The role of the airline transportation network in the prediction and predictability of global epidemics," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 7, pp. 2015–2020, 2006.

[20] J. Zhang and J. M. F. Moura, "Diffusion in social networks as SIS epidemics: Beyond full mixing and complete graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 537–551, Aug. 2014.

[21] J. Zhang and J. M. F. Moura, "Role of subgraphs in epidemics over finite-size networks under the scaled SIS process," *J. Complex Netw.*, vol. 3, no. 4, pp. 584–605, 2015.

[22] M. Draief and L. Massouli, *Epidemics and Rumours in Complex Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[23] A. Santos, J. M. F. Moura, and J. M. F. Xavier, "Bi-Virus SIS epidemics over networks: Qualitative analysis," *IEEE Trans. Netw. Sci. Eng.*, vol. 2, no. 1, pp. 17–29, Jan. 2015.

[24] M. Levorato, U. Mitra, and A. Goldsmith, "Structure-based learning in wireless networks via sparse approximation," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 278, 2012.

[25] M. Levorato, S. Narang, U. Mitra, and A. Ortega, "Reduced dimension policy iteration for wireless network control via multiscale analysis," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 3886–3892.

[26] S. L. Lauritzen, *Graphical Models*, vol. 17. Oxford, U.K.: Clarendon, 1996.

[27] M. I. Jordan, *Learning in Graphical Models*, vol. 89. New York, NY, USA: Springer-Verlag, 1998.

[28] M. I. Jordan and M. J. Wainwright, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, nos. 1–2, pp. 1–305, 2007.

[29] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[30] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. Hoboken, NJ, USA: Wiley, 2009.

[31] J. Bang-Jensen and G. Z. Gutin, *Digraphs: Theory, Algorithms and Applications*. New York, NY, USA: Springer-Verlag, 2008.

[32] D. Edwards, *Introduction to Graphical Modelling*. New York, NY, USA: Springer-Verlag, 2012.

[33] Y. A. Rozanov, *Markov Random Fields*. New York, NY, USA: Springer-Verlag, 1982, pp. 55–102.

[34] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*, vol. 1. Providence, RI, USA: AMS, 1980.

[35] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2005.

[36] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Stat. Soc. Ser. B (Methodol.)*, pp. 192–236, 1974.

[37] R. Chellappa and A. Jain, Eds., *Markov Random Fields Theory and Application*. Boston, MA, USA: Academic, 1993.

[38] J. M. F. Moura and N. Balram, "Recursive structure of noncausal Gauss-Markov random fields," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 334–354, Feb. 1992.

[39] N. Balram and J. Moura, "Noncausal Gauss Markov random fields: Parameter structure and estimation," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1333–1355, Jul. 1993.

[40] A. S. Willsky, "Multiresolution Markov models for signal and image processing," *Proc. IEEE*, vol. 90, no. 8, pp. 1396–1458, Aug. 2002.

[41] D. Vats and J. M. F. Moura, "Finding non-overlapping clusters for generalized inference over graphical models," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6368–6381, Dec. 2012.

[42] M. Jordan, E. Sudderth, M. Wainwright, and A. Willsky, "Major advances and emerging developments of graphical models [from the guest editors]," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 17–138, Jun. 2010.

[43] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Nov. 2016.

[44] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Jun. 2017.

[45] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, Sep. 2014.

[46] M. Püschel and J. M. F. Moura, "The algebraic approach to the discrete cosine and sine transforms and their fast algorithms," *SIAM J. Comput.*, vol. 32, no. 5, pp. 1280–1316, 2003.

[47] M. Püschel and J. M. F. Moura (Dec. 2006). *Algebraic Signal Processing Theory*. [Online]. Available: http://arxiv.org/abs/cs.IT/0612077

[48] M. Puschel and J. M. F. Moura, "Algebraic signal processing theory: Foundation and 1-D time," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3572–3585, Aug. 2008.

[49] M. Puschel and J. M. F. Moura, "Algebraic signal processing theory: 1-D space," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3586–3599, Aug. 2008.

[50] M. Puschel and J. M. F. Moura, "Algebraic signal processing theory: Cooley-Tukey type algorithms for DCTs and DSTs," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1502–1521, Apr. 2008.

[51] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[52] B. Girault, P. Gonçalves, and É. Fleury, "Translation on graphs: An isometric shift operator," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2416–2420, Dec. 2015.

[53] A. Gavili and X. P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6303–6318, Dec. 2017.

[54] J. F. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dmensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[55] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[56] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[57] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 10, pp. 5591–5596, 2003.

[58] M. Belkin and P. Niyogi, "Using manifold structure for partially labelled classification," in *Proc. NIPS*, 2002, pp. 953–960.

[59] R. R. Coifman *et al.*, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 21, pp. 7426–7431, 2005.

[60] R. R. Coifman *et al.*, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 21, pp. 7432–7437, 2005.

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

[62] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proc. IPSN*, 2004, pp. 1–10.

[63] D. Ganesan, B. Greenstein, D. Estrin, J. Heidemann, and R. Govindan, "Multiresolution storage and search in sensor networks," *ACM Trans. Storage*, vol. 1, pp. 277–315, 2005.

[64] R. Wagner, H. Choi, R. Baraniuk, and V. Delouille, "Distributed wavelet transform for irregular sensor network grids," in *Proc. IEEE SSP Workshop*, Jul. 2005, pp. 1196–1201.

[65] R. Wagner, A. Cohen, R. G. Baraniuk, S. Du, and D. Johnson, "An architecture for distributed wavelet analysis and processing in sensor networks," in *Proc. IPSN*, 2006, pp. 243–250.

[66] 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.

[67] S. K. Narang and A. Ortega, "Local two-channel critically sampled filter-banks on graphs," in *Proc. ICIP*, 2010, pp. 333–336.

[68] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2786–2799, Jun. 2012.

[69] R. Wagner, V. Delouille, and R. G. Baraniuk, "Distributed wavelet de-noising for sensor networks," in *Proc. CDC*, 2006, pp. 373–379.

[70] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. ICASSP*, 2012, pp. 3921–3924.

[71] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1101–1113, Nov. 1993.

[72] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[73] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[74] I. Guskov, W. Sweldens, and P. Schröder, "Multiresolution signal processing for meshes," in *Proc. 26th Annu. Conf. Comput. Graph. Interactive Techn.*, 1999, pp. 325–334.

[75] K. Zhou, H. Bao, and J. Shi, "3D surface filtering using spherical harmonics," *Comput.-Aided Design*, vol. 36, no. 4, pp. 363–375, 2004.

[76] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interaction Techn.*, 1995, pp. 351–358.

[77] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1975.

[78] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1983.

[79] W. M. Siebert, *Circuits, Signals, and Systems*. Cambridge, MA, USA: MIT Press, 1986.

[80] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[81] S. K. Mitra, *Digital Signal Processing. A Computer-Based Approach*. New York, NY, USA: McGraw Hill, 1998.

[82] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sep. 2015.

[83] F. R. Gantmacher, *Matrix Theory*, vol. 21. New York, NY, USA: Chelsea, 1959.

[84] P. Lancaster and M. Tismenetsky, *The Theory of Matrices: With Applications*. Amsterdam, The Netherlands: Elsevier, 1985.

[85] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, "Spline-like wavelet filterbanks for multiresolution analysis of graph-structured data," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 268–278, Apr. 2015.

[86] E. BrianDavies, G. M. L. Gladwell, J. Leydold, and P. F. Stadler, "Discrete nodal domain theorems," *Linear Algebra Appl.*, vol. 336, nos. 1–3, pp. 51–60, 2001.

[87] S. Brooks and E. Lindenstrauss, "Non-localization of eigenfunctions on large regular graphs," *Israel J. Math.*, vol. 193, no. 1, pp. 1–14, 2013.

[88] N. Saito and E. Woei (2011). *On the Phase Transition Phenomenon of Graph Laplacian Eigenfunctions on Trees*. [Online]. Available: http://hdl.handle.net/2433/170938

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

[90] F. Bauer, "Normalized graph Laplacians for directed graphs," *Linear Algebra Appl.*, vol. 436, no. 11, pp. 4193–4222, 2012.

[91] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2012.

[92] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Appl. Comput. Harmon. Anal.*, Feb. 2015.

[93] J. A. Deri and J. M. F. Moura, "Spectral projector-based graph Fourier transforms," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 785–795, Sep. 2017.

[94] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[95] D. Zhou, T. Hofmann, and B. Schölkopf, "Semi-supervised learning on directed graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1633–1640.

[96] H. N. Mhaskar, "A unified framework for harmonic analysis of functions on directed graphs and changing data," *Appl. Comput. Harmon. Anal.*, vol. 44, no. 3, pp. 611–644, May. 2018.

[97] B. Girault, A. Ortega, and S. Narayanan (2018). "Irregularity-aware graph Fourier transforms." [Online]. Available: https://arxiv.org/abs/1802.10220

[98] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "On the graph Fourier transform for directed graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 796–811, Sep. 2017.

[99] R. Shafipour, A. Khodabakhsh, G. Mateos, and E. Nikolova (2017). "A digraph Fourier transform with spread frequency components." [Online]. Available: https://arxiv.org/abs/1705.10821

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

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

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

[103] J. Zeng, G. Cheung, and A. Ortega, "Bipartite approximation for graph wavelet signal decomposition," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5466–5480, Oct. 2017.

[104] M. S. Kotzagiannidis and P. L. Dragotti, "Splines and wavelets on circulant graphs," *Appl. Comput. Harmon. Anal.*, 2017.

[105] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs—Part I: Fundamentals," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 409–422, Jan. 2017.

[106] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs—Part II: M-channel filter banks," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 423–437, Jan. 2017.

[107] A. Anis and A. Ortega, "Critical sampling for wavelet filterbanks on arbitrary graphs," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 3889–3893.

[108] D. B. Tay and J. Zhang, "Techniques for constructing biorthogonal bipartite graph filter banks," *IEEE Trans. Signal Process.*, vol. 63, no. 21, pp. 5772–5783, Nov. 2015.

[109] D. B. H. Tay and A. Ortega, "Bipartite graph filter banks: Polyphase analysis and generalization," *IEEE Trans. Signal Process.*, vol. 65, no. 18, pp. 4833–4846, Sep. 2017.

[110] Y. Tanaka and A. Sakiyama, "M-channel oversampled graph filter banks," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3578–3590, Jul. 2014.

[111] A. Sakiyama and Y. Tanaka, "Oversampled graph Laplacian matrix for graph filter banks," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6425–6437, Dec. 2014.

[112] D. B. H. Tay and Z. Lin, "Design of near orthogonal graph filter banks," *IEEE Signal Process. Lett.*, vol. 22, no. 6, pp. 701–704, Jun. 2015.

[113] D. B. Tay, Y. Tanaka, and A. Sakiyama, "Critically sampled graph filter banks with polynomial filters from regular domain filter banks," *Signal Process.*, vol. 131, pp. 66–72, 2017.

[114] Y. Tanaka. (2017). "Spectral domain sampling of graph signals." [Online]. Available: https://arxiv.org/abs/1706.05147

[115] X. Zhang, X. Dong, and P. Frossard, "Learning of structured graph dictionaries," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 3373–3376.

[116] D. Thanou, D. I. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Trans. Signal Process.*, vol. 62, no. 15, pp. 3849–3862, Aug. 2014.

[117] Y. Yankelevsky and M. Elad, "Dual graph regularized dictionary learning," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 611–624, Apr. 2016.

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

[119] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2014, pp. 3864–3868.

[120] H. Shomorony and A. S. Avestimehr, "Sampling large data on graphs," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2014, pp. 933–936.

[121] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[122] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, 2016.

[123] S. Chen, R. Varma, A. Singh, and J. Kovačević (2015). "Signal representations on graphs: Tools and applications." [Online]. Available: https://arxiv.org/abs/1512.05406

[124] A. Gadde and A. Ortega, "A probabilistic interpretation of sampling theory of graph signals," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 3257–3261.

[125] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Appl. Comput. Harmon. Anal.*, 2016.

[126] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2013, pp. 5445–5449.

[127] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.

[128] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, Nov. 2017.

[129] B. Girault, "Stationary graph signals using an isometric graph translation," in *Proc. 23rd Eur. Signal Process. Conf. (EUSIPCO)*, 2015, pp. 1516–1520.

[130] B. Girault, P. Gončalves, S. S. Narayanan, and A. Ortega, "Localization bounds for the graph translation," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2016, pp. 331–335.

[131] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, Dec. 2017.

[132] B. Girault, P. Gončalves, E. Fleury, and A. S. Mor, "Semi-supervised learning for graph to signal mapping: A graph signal Wiener filter interpretation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2014, pp. 1115–1119.

[133] A. Agaskar and Y. M. Lu, "A spectral graph uncertainty principle," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4338–4356, Jul. 2013.

[134] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.

[135] B. Pasdeloup, R. Alami, V. Gripon, and M. Rabbat, "Toward an uncertainty principle for weighted graphs," in *Proc. 23rd Eur. Signal Process. Conf. (EUSIPCO)*, 2015, pp. 1496–1500.

[136] O. Teke and P. Vaidyanathan, "Uncertainty principles and sparse eigenvectors of graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5406–5420, Oct. 2017.

[137] H. Behjat, U. Richter, D. Van De Ville, and L. Sörnmo, "Signal-adapted tight frames on graphs," *IEEE Trans. Signal Process.*, vol. 64, no. 22, pp. 6017–6029, Nov. 2016.

[138] D. Van De Ville, R. Demesmaeker, and M. G. Preti, "When Slepian meets Fiedler: Putting a focus on the graph spectrum," *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 1001–1004, Jul. 2017.

[139] J. Irion and N. Saito, "Hierarchical graph Laplacian eigen transforms," *JSIAM Lett.*, vol. 6, pp. 21–24, 2014.

[140] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 441–448.

[141] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 492–501.

[142] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Artif. Intell. Stat.*, 2016, pp. 920–929.

[143] S. I. Daitch, J. A. Kelner, and D. A. Spielman, "Fitting a graph to vector data," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 201–208.

[144] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," *Cognit. Sci. Soc.*, vol. 32, no. 32, 2010.

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

[146] J. Mei and J. M. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, Aug. 2017.

[147] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Netw.*, to be published.

[148] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.

[149] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 484–499, Mar. 2017.

[150] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind identification of graph filters," *IEEE Trans. Signal Process.*, vol. 65, no. 5, pp. 1146–1159, Mar. 2017.

[151] E. Pavez, H. E. Egilmez, and A. Ortega, "Learning graphs with monotone topology properties and multiple connected components," *IEEE Trans. Signal Process.*, to be published.

[152] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, "Graph spectral image processing," *Proc. IEEE*, vol. 106, no. 5, May 2018, DOI: 10.1109/JPROC.2018.2799702.

[153] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 309–316.

[154] G. Shen and A. Ortega, "Joint routing and 2D transform optimization for irregular sensor network grids using wavelet lifting," in *Proc. 7th Int. Conf. Inf. Process. Sensor Netw.*, 2008, pp. 183–194.

[155] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2014, pp. 1085–1089.

[156] X. Zhu and M. Rabbat, "Graph spectral compressed sensing for sensor networks," in *Proc. IEEE ICASSP*, Mar. 2012, pp. 2865–2868.

[157] M. Kaneko, G. Cheung, W.-T. Su, and C.-W. Lin (Apr. 2017). "Graph-based joint signal/power restoration for energy harvesting wireless sensor networks." [Online]. Available: https://arxiv.org/abs/1704.06012

[158] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, "Efficient sensor position selection using graph signal sampling theory," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 6225–6229.

[159] R. K. Jain, J. M. F. Moura, and C. E. Kontokosta, "Big databig cities: Graph signals of urban air pollution [exploratory SP]," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 130–136, May 2014.

[160] K. He and L. Stanković, and J. Liao, "Non-intrusive load disaggregation using graph signal processing," *IEEE Trans. Smart Grid*, 2016, DOI: 10.1109/TSG.2016.2598872.

[161] P. Valdivia, F. Dias, F. Petronetto, C. T. Silva, and L. G. Nonato, "Wavelet-based visualization of time-varying data on graphs," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol. (VAST)*, Oct. 2015, pp. 1–8.

[162] S. Chen, Y. Yang, and J. M. F. Moura (Jul. 2016). "Localization, decomposition, and dictionary learning of piecewise-constant signals on graphs." [Online]. Available: https://arxiv.org/abs/1607.01100

[163] X. Dong, A. Ortega, P. Frossard, and P. Vandergheynst, "Inference of mobility patterns via spectral graph wavelets," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2013, pp. 3118–3122.

[164] D. M. Mohan, M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Wavelets on graphs with application to transportation networks," in *Proc. IEEE 17th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1707–1712.

[165] I. Jablonski, "Graph signal processing in applications to sensor networks, smart grids and smart cities," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7659–7666, 2017.

[166] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard (2017). "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal Process.*, to be published [Online]. Available: https://arxiv.org/abs/1111.5239

[167] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 728–740, Jun. 2015.

[168] W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Graph frequency analysis of brain signals," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1189–1203, Oct. 2016.

[169] J. D. Medaglia *et al.*, "Functional alignment with anatomical networks is associated with cognitive flexibility," *Nature Human Behav.*, vol. 2, no. 2, p. 156, 2018.

[170] E. Bullmore and O. Sporns, "The economy of brain network organization," *Nature Rev. Neurosci.*, vol. 13, pp. 336–349, May 2012.

[171] O. Sporns, *Networks of the Brain*. Cambridge, MA, USA: MIT Press, 2011.

[172] D. S. Bassett, N. F. Wymbs, M. A. Porter, P. J. Mucha, J. M. Carlson, and S. T. Grafton, "Dynamic reconfiguration of human brain networks during learning," *Proc. PNAS*, vol. 108, no. 18, pp. 7641–7646, 2011.

[173] L. Goldsberry, W. Huang, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Brain signal analytics from graph signal processing perspective," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP)*, Jun. 2017, pp. 851–855.

[174] H. Behjat, N. Leonardi, L. Sörnmo, and D. Van De Ville, "Anatomically-adapted graph wavelets for improved group-level fMRI activation mapping," *NeuroImage*, vol. 123, pp. 185–199, Dec. 2015.

[175] N. Leonardi and D. Van De Ville, "Tight wavelet frames on multislice graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3357–3367, Jul. 2013.

[176] S. Atasoy, I. Donnelly, and J. Pearson, "Human brain networks function in connectome-specific harmonic waves," *Nature Commun.*, vol. 7, p. 10340, 2016.

[177] A. Griffa *et al.*, "Transient networks of spatio-temporal connectivity map communication pathways in brain functional systems," *NeuroImage*, vol. 155, pp. 490–502, Jul. 2017.

[178] M. Ménoret, N. Farrugia, B. Pasdeloup, and V. Gripon, "Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Mar. 2017.

[179] C. Hu *et al.*, "A spectral graph regression model for learning brain connectivity of Alzheimer's disease," *PLoS ONE*, vol. 10, no. 5, May 2015, e0128136.

[180] C. Hu, J. Sepulcre, K. A. Johnson, G. E. Fakhri, Y. M. Lu, and Q. Li, "Matched signal detection on graphs: Theory and application to brain imaging data classification," *NeuroImage*, vol. 125, pp. 587–600, Jan. 2016.

[181] C. Hu, X. Hua, J. Ying, P. M. Thompson, G. E. Fakhri, and Q. Li, "Localizing sources of brain disease progression with network diffusion model," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1214–1225, Jul. 2016.

[182] A. Pirayre, C. Couprie, F. Bidard, L. Duval, and J.-C. Pesquet, "BRANE Cut: Biologically-related a priori network enhancement with graph cuts for gene regulatory network inference," *BMC Bioinf.*, vol. 16, no. 1, p. 368, Nov. 2015.

[183] A. Pirayre, C. Couprie, L. Duval, and J.-C. Pesquet, "BRANE Clust: Cluster-assisted gene regulatory network inference refinement," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2017, DOI: 10.1109/TCBB.2017.2688355.

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

[185] C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, and H. Talbot, "Dual constrained TV-based regularization on graphs," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1246–1273, Oct. 2013.

[186] L. Najman, "Extending the PowerWatershed framework thanks to Γ-convergence," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 2275–2292, 2017.

[187] D. Tian, H. Mansour, A. Knyazev, and A. Vetro, "Chebyshev and conjugate gradient filters for graph image denoising," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2014, pp. 1–6.

[188] J. Pang, G. Cheung, A. Ortega, and O. C. Au, "Optimal graph Laplacian regularization for natural image denoising," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. Hong Kong, Jan. 2015, pp. 2294–2298.

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

[190] G. Fracastoro, D. Thanou, and P. Frossard, "Graph transform learning for image compression," in *Proc. Picture Coding Symp.*, 2016.

[191] H. E. Egilmez, A. Said, Y.-H. Chao, and A. Ortega, "Graph-based transforms for inter predicted video coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3992–3996.

[192] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1765–1778, Apr. 2016.

[193] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compression with graph transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2066–2070.

[194] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, "Fast resampling of three-dimensional point clouds via graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 666–681, Feb. 2018.

[195] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovačević, "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2879–2893, Jun. 2014.

[196] K. Benzi, V. Kalofolias, X. Bresson, and P. Vandergheynst, "Song recommendation with non-negative matrix factorization and graph total variation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016, pp. 2439–2443.

[197] W. Huang, A. G. Marques, and A. Ribeiro, "Collaborative filtering via graph signal processing," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, 2017, pp. 1094–1098.

[198] M. Valko, "Spectral bandits for smooth graph functions," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Le Chesnay, France, Jan. 2014, pp. 1205–1215.

[199] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5227–5239, Oct. 2014.

[200] C. Boutsidis, A. Gittens, and P. Kambadur, "Spectral clustering via the power method—Provably," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 40–48.

[201] Y. Chi, "Compressive graph clustering from random sketches," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5466–5469.

[202] F. Gama, A. G. Marques, G. Mateos, and A. Ribeiro, "Rethinking sketching as sampling: A graph signal processing approach," Tech. Rep., Nov. 2016.

[203] J. Paratte and L. Martin (Nov. 2016). "Fast eigenspace approximation using random signals." [Online]. Available: https://arxiv.org/abs/1611.00938

[204] A. Dal Col, P. Valdivia, F. Petronetto, F. Dias, C. T. Silva, and L. G. Nonato, "Wavelet-based visual analysis of dynamic networks," *IEEE Trans. Vis. Comput. Graphics*, 2017, DOI: 10.1109/TVCG.2017.2746080.

[205] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, 2015, pp. 832–840.

[206] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," in *Proc. Eurograph. Symp. Geometry Process.*, Jun. 2015, pp. 1–11.

[207] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. ICLR*, 2014.

[208] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proc. NIPS*, 2016, pp. 3189–3197.

[209] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, 2016.

[210] R. Khasanova and P. Frossard, "Graph-based isometry invariant representation learning," in *Proc. ICML*, 2017.

[211] D. Duvenaud *et al.*, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. NIPS*, Dec. 2015, pp. 2224–2232.

[212] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf.*, New York, NY, USA, 2014, pp. 701–710.

[213] R. Anirudh and J. J. Thiagarajan (Apr. 2017). "Bootstrapping graph convolutional neural networks for autism spectrum disorder classification." [Online]. Available: https://arxiv.org/abs/1704.07487

[214] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in Proc. Int. Learning Representations (ICLR '18), Jul. 2017.

[215] N. Perraudin *et al.*, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014. [Online]. Available: https://epfl-lts2.github.io/gspbox-html/

[216] B. Girault, S. S. Narayanan, A. Ortega, P. Gonçalves, and E. Fleury, "Grasp: A MATLAB toolbox for graph signal processing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 6574–6575. [Online]. Available: https://gforge.inria.fr/projects/grasp

[217] M. Defferrard, L. Martin, R. Pena, and N. Perraudin *PYGSP: Graph Signal Processing in Python*. [Online]. Available: https://github.com/epfl-lts2/pygsp/

## ABOUT THE AUTHORS

**Antonio Ortega** (Fellow, IEEE) received the Telecommunications Engineering degree from the Universidad Politécnica de Madrid, Madrid, Spain, in 1989 and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 1994. At Columbia he was supported by a Fulbright scholarship.

In 1994, he joined the Electrical Engineering Department, University of Southern California (USC), Los Angeles, CA, USA, where he is currently a Professor. He has served as Associate Chair of EE-Systems and Director of the Signal and Image Processing Institute at USC. His research interests are in the areas of signal compression, representation, communication, and analysis. His recent work has focused on distributed compression, multiview coding, error-tolerant compression, information representation in wireless sensor networks, and graph signal processing. Over 40 Ph.D. students have completed their Ph.D. dissertations under his supervision at USC, and his work has led to over 400 publications in international conferences and journals, as well as several patents.

Prof. Ortega is a Fellow of the European Association for Signal Processing (EURASIP) and a member of the Association for Computing Machinery (ACM) and the Asia Pacific Signal and Information Processing Association (APSIPA). He has been Chair of the Image and Multidimensional Signal Processing (IMDSP) Technical Committee, and Chair of the SPS Big Data Special Interest Group. He has been technical program Co-Chair of MMSP 1998, ICME 2002, ICIP 2008, PCS 2013, PCS 2018, and DSW 2018; and a Senior Associate Editor for the IEEE Transactions on Image Processing (IEEE TIP) and IEEE Signal Processing Magazine, among others. He is the inaugural Editor-in-Chief of the *APSIPA Transactions on Signal and Information Processing* and an Associate Editor of the IEEE Transactions on Signal and Information Processing over Networks. He currently serves as a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society (SPS). He received the NSF CAREER award, the 1997 IEEE Communications Society Leonard G. Abraham Prize Paper Award, the IEEE Signal Processing Society 1999 Magazine Award, the 2006 *EURASIP Journal of Advances in Signal Processing* Best Paper Award, the ICIP 2011 best paper award, a best paper award at Globecom 2012, and the 2016 SPM Award. He was a plenary speaker at ICIP 2013, APSIPA ASC 2015, and CAMSAP 2017.

**Pascal Frossard** (Fellow, IEEE) received the M.S. and Ph.D. degrees 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, NY, USA, where he worked on media coding and streaming technologies. Since 2003, he has been a faculty member at EPFL, where he heads the Signal Processing Laboratory (LTS4). His research interests include graph signal processing, image representation and coding, visual information analysis, and distributed signal processing and communications.

Dr. Frossard has been the General Chair of IEEE ICME 2002 and Packet Video 2007. He has been the Technical Program Chair of IEEE ICIP 2014 and EUSIPCO 2008, and a member of the organizing or technical program committees of numerous conferences. He has been an Associate Editor of the IEEE Transactions on Signal Processing (2015–present), the IEEE Transactions on Big Data (2015–present), the IEEE Transactions on Image Processing (2010–2013), the IEEE Transactions on Multimedia (2004–2012), and the IEEE Transactions on Circuits and Systems for Video Technology (2006–2011). He is an elected member of the IEEE Multimedia Signal Processing Technical Committee (2004–2007, 2016–present), the IEEE Visual Signal Processing and Communications Technical Committee (2006–present), and the IEEE Multimedia Systems and Applications Technical Committee (2005–present). He has served as Chair of the IEEE Image, Video and Multidimensional Signal Processing Technical Committee (2014–2015), and Steering Committee Chair (2012–2014) and Vice-Chair (2004–2006) of the IEEE Multimedia Communications Technical Committee. He received the Swiss NSF Professorship Award in 2003, the IBM Faculty Award in 2005, the IBM Exploratory Stream Analytics Innovation Award in 2008, the IEEE Transactions on Multimedia Best Paper Award in 2011, and the IEEE Signal Processing Magazine Best Paper Award 2016.

**Jelena Kovačević** (Fellow, IEEE) received the Dipl. Electr. Eng. degree from the Electrical Engineering Department, University of Belgrade, Belgrade, Yugoslavia, in 1986 and the M.S. and Ph.D. degrees from Columbia University, New York, NY, USA, in 1988 and 1991, respectively.

From 1991 to 2002, she was with Bell Labs, Murray Hill, NJ, USA. She was a cofounder and Technical VP of xWaveforms, based in New York City and an Adjunct Professor at Columbia University. In 2003, she joined Carnegie Mellon University, Pittsburgh, PA, USA, where she is Hamerschlag University Professor and Head of Electrical and Computer Engineering, Professor of Biomedical Engineering, and was the Director of the Center for Bioimage Informatics. She coauthored the books *Wavelets and Subband Coding* (Englewood Cliffs, NJ, USA: Prentice-Hall, 1995) and *Foundations of Signal Processing* (Cambridge, U.K.: Cambridge Univ. Press, 2014). Her research interests include wavelets, frames, graphs, and applications to bioimaging and smart infrastructure.

Dr. Kovačević coauthored a top-10 cited paper in the *Journal of Applied and Computational Harmonic Analysis* and the paper for which A. Mojsilović received the Young Author Best Paper Award. Her paper on multidimensional filter banks and wavelets was selected as one of the Fundamental Papers in Wavelet Theory. She received the Belgrade October Prize in 1986, the E.I. Jury Award at Columbia University in 1991, the 2010 CIT Philip L. Dowd Fellowship Award from the College of Engineering at Carnegie Mellon University, and the 2016 IEEE SPS Technical Achievement Award. She is a past Editor-in-Chief of the IEEE Transactions on Image Processing, served as a guest coeditor on a number of special issues, and is/was on the editorial boards of several journals. She was a regular member of the NIH Microscopic Imaging Study Section and served as a Member-at-Large of the IEEE Signal Processing Society Board of Governors. She is a past Chair of the IEEE Signal Processing Society Bio Imaging and Signal Processing Technical Committee. She has been involved in organizing numerous conferences. She was a plenary/keynote speaker at a number of international conferences and meetings.

**José M. F. Moura** (Fellow, IEEE) received the Engenheiro Electrotécnico degree from the Instituto Superior Técnico (IST), Lisbon, Portugal and the M.Sc., E.E., and D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA.

He is the Philip L. and Marsha Dowd University Professor at Carnegie Mellon University (CMU), Pittsburgh, PA, USA. He was on the faculty at IST and has held visiting faculty appointments at MIT and New York University (NYU). He founded and directs a large education and research program between CMU and Portugal ( http://www.cmuportugal.org). His research interests are in data science, graph signal processing, and statistical and algebraic signal and image processing. He has published over 550 papers and holds 14 patents issued by the U.S. Patent Office. The technology of two of his patents (co-inventor A. Kavčić) are in over three billion disk drive read channel chips, about 60 % of all computers sold in the last 13 years worldwide, and was, in 2016, the subject of a 750 million U.S. dollars settlement between CMU and Marvell, the largest university settlement ever in the information technologies area.

Dr. Moura is the 2018 IEEE President Elect, he was the IEEE Technical Activities Vice-President (2016) and member of the IEEE Board of Directors. He served in several other capacities including IEEE Division IX Director, member of several IEEE Boards, President of the IEEE Signal Processing Society (SPS), the Editor-in-Chief for the IEEE Transactions in Signal Processing, and the interim Editor-in-Chief for IEEE Signal Processing Letters. He has received several awards, including the Technical Achievement Award and the Society Award from the IEEE Signal Processing Society. In 2016, he received the CMU College of Engineering Distinguished Professor of Engineering Award. He is a Fellow of the American Association for the Advancement of Science (AAAS), a corresponding member of the Academy of Sciences of Portugal, a Fellow of the U.S. National Academy of Inventors, and a member of the U.S. National Academy of Engineering.

**Pierre Vandergheynst** 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 at the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He was an Assistant Professor at EPFL (2002 2007), where he is now a Full Professor of Electrical Engineering and, by courtesy, of Computer and Communication Sciences. He serves as EPFL s Vice-President for Education. He is the author or coauthor of more than 70 journal papers, one monograph, and several book chapters. His research focuses on harmonic analysis, sparse approximations, and mathematical data processing in general with applications covering signal, image, and high-dimensional data processing, computer vision, machine learning, data science, and graph-based data processing.

Prof. Vandergheynst was the co-Editor-in-Chief of *Signal Processing* (2002–2006), an Associate Editor of the IEEE Transactions on Signal Processing (2007–2011), the flagship journal of the signal processing community, and currently serves as an Associate Editor of *Computer Vision and Image Understanding* and *SIAM Imaging Sciences*. He has been on the Technical Committee of various conferences, serves on the steering committee of the SPARS workshop, and was co-General Chairman of the EUSIPCO 2008 conference. He has received two IEEE best paper awards. He is a laureate of the Apple 2007 ARTS award and of the 2009–2010 De Boelpaepe prize of the Royal Academy of Sciences of Belgium.