Appl. Comput. Harmon. Anal. 30 (2011) 129–150

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Wavelets on graphs via spectral graph theory

David K. Hammond [a,*,1], Pierre Vandergheynst [b,2], Rémi Gribonval [c]

[a] NeuroInformatics Center, University of Oregon, Eugene, USA
[b] Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[c] INRIA, Rennes, France

ARTICLE INFO

Article history:
Received 14 November 2009
Revised 21 April 2010
Accepted 25 April 2010
Available online 28 April 2010
Communicated by Charles K. Chui

Keywords:
Graph theory
Wavelets
Spectral graph theory
Overcomplete wavelet frames

ABSTRACT

We propose a novel method for constructing wavelet transforms of functions defined on the vertices of an arbitrary finite weighted graph. Our approach is based on defining scaling using the graph analogue of the Fourier domain, namely the spectral decomposition of the discrete graph Laplacian $\mathcal{L}$. Given a wavelet generating kernel g and a scale parameter t, we define the scaled wavelet operator $T_g^t = g(t\mathcal{L})$. The spectral graph wavelets are then formed by localizing this operator by applying it to an indicator function. Subject to an admissibility condition on g, this procedure defines an invertible transform. We explore the localization properties of the wavelets in the limit of fine scales. Additionally, we present a fast Chebyshev polynomial approximation algorithm for computing the transform that avoids the need for diagonalizing $\mathcal{L}$. We highlight potential applications of the transform through examples of wavelets on graphs corresponding to a variety of different problem domains.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

Many interesting scientific problems involve analyzing and manipulating structured data. Such data often consist of sampled real-valued functions defined on domain sets themselves having some structure. The simplest such examples can be described by scalar functions on regular Euclidean spaces, such as time series data, images or videos. However, many interesting applications involve data defined on more topologically complicated domains. Examples include data defined on network-like structures, data defined on manifolds or irregularly shaped domains, and data consisting of "point clouds", such as collections of feature vectors with associated labels. As many traditional methods for signal processing are designed for data defined on regular Euclidean spaces, the development of methods that are able to accommodate complicated data domains is an important problem.

Many signal processing techniques are based on transform methods, where the input data is represented in a new basis before analysis or processing. One of the most successful types of transforms in use is wavelet analysis. Wavelets have proved over the past 25 years to be an exceptionally useful tool for signal processing. Much of the power of wavelet methods comes from their ability to simultaneously localize signal content in both space and frequency. For signals whose primary information content lies in localized singularities, such as step discontinuities in time series signals or edges in images, wavelets can provide a much more compact representation than either the original domain or a transform with global basis elements such as the Fourier transform. An enormous body of literature exists for describing and exploiting this

* Corresponding author.
E-mail addresses: hammond@uoregon.edu (D.K. Hammond), pierre.vanderhheynst@epfl.ch (P. Vandergheynst), remi.gribonval@inria.fr (R. Gribonval).

[1] This work was performed while DKH was at EPFL.

[2] This work was supported in part by the EU Framework 7 FET-Open project FP7-ICT-225913-SMALL: Sparse Models, Algorithms and Learning for Large-Scale Data.

1063-5203/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.acha.2010.04.005

wavelet sparsity. We include a few representative references for applications to signal compression [1–5], denoising [6–10], and inverse problems including deconvolution [11–15]. As the individual waveforms comprising the wavelet transform are self-similar, wavelets are also useful for constructing scale invariant descriptions of signals. This property can be exploited for pattern recognition problems where the signals to be recognized or classified may occur at different levels of zoom [16]. In a similar vein, wavelets can be used to characterize fractal self-similar processes [17].

The demonstrated effectiveness of wavelet transforms for signal processing problems on regular domains motivates the study of extensions to irregular, non-Euclidean spaces. In this paper, we describe a flexible construction for defining wavelet transforms for data defined on the vertices of a weighted graph. Our approach uses only the connectivity information encoded in the edge weights, and does not rely on any other attributes of the vertices (such as their positions as embedded in 3d space, for example). As such, the transform can be defined and calculated for any domain where the underlying relations between data locations can be represented by a weighted graph. This is important as weighted graphs provide an extremely flexible model for approximating the data domains of a large class of problems.

Some data sets can naturally be modeled as scalar functions defined on the vertices of graphs. For example, computer networks, transportation (road, rail, airplane) networks or social networks can all be described by weighted graphs, with the vertices corresponding to individual computers, cities or people respectively. The graph wavelet transform could be useful for analyzing data defined on these vertices, where the data is expected to be influenced by the underlying topology of the network. As a mock example problem, consider rates of infection of a particular disease among different population centers. As the disease may be expected to spread by people traveling between different areas, the graph wavelet transform based on a weighted graph representing the transportation network may be helpful for this type of data analysis.

Weighted graphs also provide a flexible generalization of regular grid domains. By identifying the grid points with vertices and connecting adjacent grid points with edges with weights inversely proportional to the square of the distance between neighbors, a regular lattice can be represented with weighted graph. A general weighted graph, however, has no restriction on the regularity of vertices. For example points on the original lattice may be removed, yielding a "damaged grid", or placed at arbitrary locations corresponding to irregular sampling. In both of these cases, a weighted graph can still be constructed that represents the local connectivity of the underlying data points. Wavelet transforms that rely upon regular spaced samples will fail in these cases, however transforms based on weighted graphs may still be defined.

Similarly, weighted graphs can be inferred from mesh descriptions for geometrical domains. An enormous literature exists on techniques for generating and manipulating meshes; such structures are widely used in applications for computer graphics and numerical solution of partial differential equations. The transform methods we will describe thus allow the definition of a wavelet transform for data defined on any geometrical shape that can be described by meshes.

Weighted graphs can also be used to describe the similarity relationships between "point clouds" of vectors. Many approaches for machine learning or pattern recognition problems involve associating each data instance with a collection of feature vectors that hopefully encapsulate sufficient information about the data point to solve the problem at hand. For example, for machine vision problems dealing with object recognition, a common preprocessing step involves extracting keypoints and calculating the Scale Invariant Feature Transform (SIFT) features [18]. In many automated systems for classifying or retrieving text, word frequencies counts are used as feature vectors for each document [19]. After such feature extraction, each data point may be associated to a feature vector $v_m \in \mathbb{R}^N$, where N may be very large depending on the application. For many problems, the local distance relationships between data points are crucial for successful learning or classification. These relationships can be encoded in a weighted graph by considering the data points as vertices and setting the edge weights equal to a distance metric $A_{m,n} = d(v_m, v_n)$ for some function $d : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$. The spectral graph wavelets in this setting could find a number of uses for analysis of data defined on such point clouds. They may be useful for regularization of noisy or corrupted data on a point cloud, or could serve as a building blocks for building a hypothesis function for learning problems.

Classical wavelets are constructed by translating and scaling a single "mother" wavelet. The transform coefficients are then given by the inner products of the input function with these translated and scaled waveforms. Directly extending this construction to arbitrary weighted graphs is problematic, as it is unclear how to define scaling and translation on an irregular graph.

We approach this problem by working in the spectral graph domain, i.e. using the basis consisting of the eigenfunctions of the graph Laplacian $\mathcal{L}$. This tool from spectral graph theory [20], provides an analogue of the Fourier transform for functions on weighted graphs. In our construction, the wavelet operator at unit scale is given as an operator-valued function $T_g = g(\mathcal{L})$ for a generating kernel g. Scaling is then defined in the spectral domain, i.e. the operator T_g^t at scale t is given by $g(t\mathcal{L})$. Applying this operator to an input signal f gives the wavelet coefficients of f at scale t. These coefficients are equivalent to inner products of the signal f with the individual graph wavelets. These wavelets can be calculated by applying this operator to a delta impulse at a single vertex, i.e. $\psi_{t,m} = T_g^t \delta_m$. We show that this construction is analogous to the 1d wavelet transform for a symmetric wavelet, where the transform is viewed as a Fourier multiplier operator at each wavelet scale.

In this paper we introduce this spectral graph wavelet transform and study several of its properties. We show that in the fine scale limit, for sufficiently regular g, the wavelets exhibit good localization properties. With continuously defined spatial scales, the transform is analogous to the continuous wavelet transform, and we show that it is formally invertible subject to an admissibility condition on the kernel g. Sampling the spatial scales at a finite number of values yields a redundant, invertible transform with overcompleteness equal to the number of spatial scales chosen. We show that in this

case the transform defines a frame, and give a condition for computing the frame bounds depending on the selection of spatial scales.

While we define our transform in the spectral graph domain, directly computing it via fully diagonalizing the Laplacian operator is infeasible for problems with size exceeding a few thousand vertices. We introduce a method for approximately computing the forward transform through operations performed directly in the vertex domain that avoids the need to diagonalize the Laplacian. By approximating the kernel g with a low-dimensional Chebyshev polynomial, we may compute an approximate forward transform in a manner which accesses the Laplacian only through matrix-vector multiplication. This approach is computationally efficient if the Laplacian is sparse, as is the case for many practically relevant graphs.

We show that computation of the pseudoinverse of the overcomplete spectral graph wavelet transform is compatible with the Chebyshev polynomial approximation scheme. Specifically, the pseudoinverse may be calculated by an iterative conjugate gradient method that requires only application of the forward transform and its adjoint, both of which may be computed using the Chebyshev polynomial approximation methods.

Our paper is structured as follows. Related work is discussed in Section 1.1. We review the classical wavelet transform in Section 2, and highlight the interpretation of the wavelet acting as a Fourier multiplier operator. We then set our notation for weighted graphs and introduce spectral graph theory in Section 4. The spectral graph wavelet transform is defined in Section 4. In Section 5 we discuss and prove some properties of the transform. Section 6 is dedicated to the polynomial approximation and fast computation of the transform. The inverse transform is discussed in Section 7. Finally, several examples of the transform on domains relevant for different problems are shown in Section 8.

1.1. Related work

Since the original introduction of wavelet theory for square integrable functions defined on the real line, numerous authors have introduced extensions and related transforms for signals on the plane and higher-dimensional spaces. By taking separable products of one-dimensional wavelets, one can construct orthogonal families of wavelets in any dimension [21]. However, this yields wavelets with often undesirable bias for coordinate axis directions. Additionally, this approach yields a number of wavelets that is exponential in the dimension of the space, and is unsuitable for data embedded in spaces of large dimensionality. A large family of alternative multiscale transforms has been developed and used extensively for image processing, including Laplacian pyramids [22], steerable wavelets [23], complex dual-tree wavelets [24], curvelets [25], and bandlets [26]. Wavelet transforms have also been defined for certain non-Euclidean manifolds, most notably the sphere [27,28] and other conic sections [29].

Previous authors have explored wavelet transforms on graphs, albeit via different approaches to those employed in this paper. Crovella and Kolaczyk [30] defined wavelets on unweighted graphs for analyzing computer network traffic. Their construction was based on the n-hop distance, such that the value of a wavelet centered at a vertex n on vertex m depended only on the shortest-path distance between m and n. The wavelet values were such that the sum over each n-hop annulus equaled the integral over an interval of a given zero-mean function, thus ensuring that the graph wavelets had zero mean. Their results differ from ours in that their construction made no use of graph weights and no study of the invertibility or frame properties of the transform was done. Smalter et al. [31] used the graph wavelets of Crovella and Kolaczyk as part of a larger method for measuring structural differences between graphs representing chemical structures, for machine learning of chemical activities for virtual drug screening.

Jansen et al. [32] develop a multiscale scheme for data on graphs based on lifting. Their scheme requires distances to be assigned to each edge, which for their examples are inferred from Euclidean distances when the graph vertices correspond to irregularly sampled points of Euclidean space. The lifting procedure is generated by using the weighted average of the graph neighbors of each vertex for the lifting prediction step. The main difference with our method is that this lifting is performed directly in the vertex domain, as opposed to our spectral domain approach for defining scaling.

Other authors have considered analogues of the wavelet transform for data defined on tree structures. Murtagh [33] developed a Haar wavelet transform for rooted binary trees, known as dendrograms. This concept was expanded upon by Lee et al. [34] who developed the treelet transform, incorporating automatic construction of hierarchical trees for multivariate data.

Maggioni and Coifman [35] introduced "diffusion wavelets", a general theory for wavelet decompositions based on compressed representations of powers of a diffusion operator. The diffusion wavelets were described with a framework that can apply on smooth manifolds as well as graphs. Their construction interacts with the underlying graph or manifold space through repeated applications of a diffusion operator T, analogously to how our construction is parametrized by the choice of the graph Laplacian $\mathcal{L}$. The largest difference between their work and ours is that the diffusion wavelets are designed to be orthonormal. This is achieved by running a localized orthogonalization procedure after applying dyadic powers of T at each scale to yield nested approximation spaces, wavelets are then produced by locally orthogonalizing vectors spanning the difference of these approximation spaces. While an orthogonal transform is desirable for many applications, notably operator and signal compression, the use of the orthogonalization procedure complicates the construction of the transform, and somewhat obscures the relation between the diffusion operator T and the resulting wavelets. In contrast our approach is conceptually simpler, gives a highly redundant transform, and affords finer control over the selection of wavelet scales.

Maggioni and Mhaskar have developed a theory of "diffusion polynomial frames" [36] that is closely related to the present work, in a more general quasi-metric measure space setting. While some of their localization results are similar

to ours, they do not provide any algorithms for efficient computation of the frames. Geller and Mayeli [37] studied a construction for wavelets on compact differentiable manifolds that is formally similar to our approach on weighted graphs. In particular, they define scaling using a pseudodifferential operator tLe^{-tL}, where L is the manifold Laplace–Beltrami operator and t is a scale parameter, and obtain wavelets by applying this to a delta impulse. They also study the localization of the resulting wavelets, however the methods and theoretical results in their paper are different as they are in the setting of smooth manifolds.

2. Classical wavelet transform

We first give an overview of the classical Continuous Wavelet Transform (CWT) for $L^2(\mathbb{R})$, the set of square integrable real-valued functions. We will describe the forward transform and its formal inverse, and then show how scaling may be expressed in the Fourier domain. These expressions will provide an analogue that we will later use to define the spectral graph wavelet transform.

In general, the CWT will be generated by the choice of a single "mother" wavelet ψ. Wavelets at different locations and spatial scales are formed by translating and scaling the mother wavelet. We write this by

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right) \tag{1}$$

This scaling convention preserves the L^1 norm of the wavelets. Other scaling conventions are common, especially those preserving the L^2 norm, however in our case the L^1 convention will be more convenient. We restrict ourselves to positive scales $s > 0$.

For a given signal f, the wavelet coefficient at scale s and location a is given by the inner product of f with the wavelet $\psi_{s,a}$, i.e.

$$W_f(s,a) = \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx \tag{2}$$

The CWT may be inverted provided that the wavelet ψ satisfies the admissibility condition

$$\int_0^{\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega}\,d\omega = C_\psi < \infty \tag{3}$$

This condition implies, for continuously differentiable ψ, that $\hat{\psi}(0) = \int \psi(x)\,dx = 0$, so ψ must be zero mean.

Inversion of the CWT is given by the following relation [38]

$$f(x) = \frac{1}{C_\psi}\int_0^{\infty}\int_{-\infty}^{\infty} W_f(s,a)\psi_{s,a}(x)\frac{da\,ds}{s} \tag{4}$$

This method of constructing the wavelet transform proceeds by producing the wavelets directly in the signal domain, through scaling and translation. However, applying this construction directly to graphs is problematic. For a given function $\psi(x)$ defined on the vertices of a weighted graph, it is not obvious how to define $\psi(sx)$, as if x is a vertex of the graph there is no interpretation of sx for a real scalar s. Our approach to this obstacle is to appeal to the Fourier domain. We will first show that for the classical wavelet transform, scaling can be defined in the Fourier domain. The resulting expression will give us a basis to define an analogous transform on graphs.

For the moment, we consider the case where the scale parameter is discretized while the translation parameter is left continuous. While this type of transform is not widely used, it will provide us with the closest analogy to the spectral graph wavelet transform. For a fixed scale s, the wavelet transform may be interpreted as an operator taking the function f and returning the function $T^s f(a) = W_f(s,a)$. In other words, we consider the translation parameter as the independent variable of the function returned by the operator T^s. Setting

$$\bar{\psi}_s(x) = \frac{1}{s}\psi^*\left(\frac{-x}{s}\right) \tag{5}$$

we see that this operator is given by convolution, i.e.

$$\begin{aligned}\left(T^s f\right)(a) &= \int_{-\infty}^{\infty} \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)\,dx = \int_{-\infty}^{\infty} \bar{\psi}_s(a-x) f(x)\,dx \\ &= (\bar{\psi}_s \star f)(a)\end{aligned} \tag{6}$$

Taking the Fourier transform and applying the convolution theorem yields

$$\widehat{T^s f}(\omega) = \hat{\bar{\psi}}_s(\omega)\hat{f}(\omega) \tag{7}$$

Using the scaling properties of the Fourier transform and the definition (5) gives

$$\hat{\bar{\psi}}_s(\omega) = \hat{\psi}^*(s\omega) \tag{8}$$

Combining these and inverting the transform we may write

$$\big(T^s f\big)(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega x}\hat{\psi}^*(s\omega)\hat{f}(\omega)\,d\omega \tag{9}$$

In the above expression, the scaling s appears only in the argument of $\hat{\psi}^*(s\omega)$, showing that the scaling operation can be completely transferred to the Fourier domain. The above expression makes it clear that the wavelet transform at each scale s can be viewed as a Fourier multiplier operator, determined by filters that are derived from scaling a single filter $\hat{\psi}^*(\omega)$. This can be understood as a band-pass filter, as $\hat{\psi}(0) = 0$ for admissible wavelets. Expression (9) is the analogue that we will use to later define the spectral graph wavelet transform.

Translation of the wavelets may be defined through "localizing" the wavelet operator by applying it to an impulse. Writing $\delta_a(x) = \delta(x-a)$, one has

$$\big(T^s\delta_a\big)(x) = \frac{1}{s}\psi^*\left(\frac{a-x}{s}\right) \tag{10}$$

For real-valued and even ψ this reduces to $(T^s\delta_a)(x) = \psi_{a,s}(x)$.

3. Weighted graphs and spectral graph theory

The previous section showed that the classical wavelet transform could be defined without the need to express scaling in the original signal domain. This relied on expressing the wavelet operator in the Fourier domain. Our approach to defining wavelets on graphs relies on generalizing this to graphs; doing so requires the analogue of the Fourier transform for signals defined on the vertices of a weighted graph. This tool is provided by spectral graph theory. In this section we fix our notation for weighted graphs, and motivate and define the graph Fourier transform.

3.1. Notation for weighted graphs

A weighted graph $G = \{E, V, w\}$ consists of a set of vertices V, a set of edges E, and a weight function $w : E \to \mathbb{R}^+$ which assigns a positive weight to each edge. We consider here only finite graphs where $|V| = N < \infty$. The adjacency matrix A for a weighted graph G is the $N \times N$ matrix with entries $a_{m,n}$ where

$$a_{m,n} = \begin{cases} w(e) & \text{if } e \in E \text{ connects vertices } m \text{ and } n \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

In the present work we consider only undirected graphs, which correspond to symmetric adjacency matrices. We do not consider the possibility of negative weights.

A graph is said to have loops if it contains edges that connect a single vertex to itself. Loops imply the presence of nonzero diagonal entries in the adjacency matrix. As the existence of loops presents no significant problems for the theory we describe in this paper, we do not specifically disallow them.

For a weighted graph, the degree of each vertex m, written as $d(m)$, is defined as the sum of the weights of all the edges incident to it. This implies $d(m) = \sum_n a_{m,n}$. We define the matrix D to have diagonal elements equal to the degrees, and zeros elsewhere.

Every real-valued function $f : V \to \mathbb{R}$ on the vertices of the graph G can be viewed as a vector in $\mathbb{R}^N$, where the value of f on each vertex defines each coordinate. This implies an implicit numbering of the vertices. We adopt this identification, and will write $f \in R^N$ for functions on the vertices of the graph, and $f(m)$ for the value on the mth vertex.

Of key importance for our theory is the graph Laplacian operator $\mathcal{L}$. The non-normalized Laplacian is defined as $\mathcal{L} = D - A$. It can be verified that for any $f \in \mathbb{R}^N$, $\mathcal{L}$ satisfies

$$(\mathcal{L}f)(m) = \sum_{m\sim n} a_{m,n}\cdot\big(f(m) - f(n)\big) \tag{12}$$

where the sum over $m \sim n$ indicates summation over all vertices n that are connected to the vertex m, and $a_{m,n}$ denotes the weight of the edge connecting m and n.

For a graph arising from a regular mesh, the graph Laplacian corresponds to the standard stencil approximation of the continuous Laplacian (with a difference in sign). Consider the graph defined by taking vertices $v_{m,n}$ as points on a

regular two-dimensional grid, with each point connected to its four neighbors with weight $1/(\delta x)^2$, where δx is the distance between adjacent grid points. Abusing the index notation, for a function $f = f_{m,n}$ defined on the vertices, applying the graph Laplacian to f yields

$$(\mathcal{L}f)_{m,n} = (4f_{m,n} - f_{m+1,n} - f_{m-1,n} - f_{m,n+1} - f_{m,n-1})/(\delta x)^2 \tag{13}$$

which is the standard 5-point stencil for approximating $-\nabla^2 f$.

Some authors define and use an alternative, normalized form of the Laplacian, defined as

$$\mathcal{L}^{norm} = D^{-1/2}\mathcal{L}D^{-1/2} = I - D^{-1/2}AD^{-1/2} \tag{14}$$

It should be noted that $\mathcal{L}$ and $\mathcal{L}^{norm}$ are not similar matrices, in particular their eigenvectors are different. As we shall see in detail later, both operators may be used to define spectral graph wavelet transforms, however the resulting transforms will not be equivalent. Unless noted otherwise we will use the non-normalized form of the Laplacian, however much of the theory presented in this paper is identical for either choice. We consider that the selection of the appropriate Laplacian for a particular problem should depend on the application at hand.

For completeness, we note the following. The graph Laplacian can be defined for graphs arising from sampling points on a differentiable manifold. The regular mesh example described previously is a simple example of such a sampling process. With increasing sampling density, by choosing the weights appropriately the normalized graph Laplacian operator will converge to a differential operator on the manifold. In the case of sampling from a uniform distribution, the limit will be the intrinsic Laplace–Beltrami operator; otherwise the limit will depend on the underlying distribution. Several authors have studied this limiting process in detail, notably [39–41].

3.2. Graph Fourier transform

On the real line, the complex exponentials $e^{i\omega x}$ defining the Fourier transform are eigenfunctions of the one-dimensional Laplacian operator $\frac{d^2}{dx^2}$. The inverse Fourier transform

$$f(x) = \frac{1}{2\pi}\int \hat{f}(\omega)e^{i\omega x}\,d\omega \tag{15}$$

can thus be seen as the expansion of f in terms of the eigenfunctions of the Laplacian operator.

The graph Fourier transform is defined in precise analogy to the previous statement. As the graph Laplacian $\mathcal{L}$ is a real symmetric matrix, it has a complete set of orthonormal eigenvectors. We denote these by χ_ℓ for $\ell = 0, \ldots, N-1$, with associated eigenvalues λ_ℓ

$$\mathcal{L}\chi_\ell = \lambda_\ell \chi_\ell \tag{16}$$

As $\mathcal{L}$ is symmetric, each of the λ_ℓ's are real. For the graph Laplacian, it can be shown that the eigenvalues are all non-negative, and that 0 appears as an eigenvalue with multiplicity equal to the number of connected components of the graph [20]. Henceforth, we assume the graph G to be connected, we may thus order the eigenvalues such that

$$0 = \lambda_0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{N-1} \tag{17}$$

For any function $f \in \mathbb{R}^N$ defined on the vertices of G, its graph Fourier transform $\hat{f}$ is defined by

$$\hat{f}(\ell) = \langle \chi_\ell, f\rangle = \sum_{n=1}^{N} \chi_\ell^*(n) f(n) \tag{18}$$

where we adopt the convention that the inner product be conjugate-linear in the first argument. The inverse transform reads as

$$f(n) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(n) \tag{19}$$

The Parseval relation holds for the graph Fourier transform, in particular for any $f, h \in \mathbb{R}^N$

$$\langle f, h\rangle = \langle \hat{f}, \hat{h}\rangle \tag{20}$$

4. Spectral graph wavelet transform

Having defined the analogue of the Fourier transform for functions defined on the vertices of weighted graphs, we are now ready to define the spectral graph wavelet transform (SGWT). The transform will be determined by the choice of a kernel function $g : \mathbb{R}^+ \to \mathbb{R}^+$, which is analogous to Fourier domain wavelet $\hat{\psi}^*$ in Eq. (9). This kernel g should behave as a band-pass filter, i.e. it satisfies $g(0) = 0$ and $\lim_{x\to\infty} g(x) = 0$. We will defer the exact specification of the kernel g that we use until later.

4.1. Wavelets

The spectral graph wavelet transform is generated by wavelet operators that are operator-valued functions of the Laplacian. One may define a measurable function of a bounded self-adjoint linear operator on a Hilbert space using the continuous functional calculus [42]. This is achieved using the spectral representation of the operator, which in our setting is equivalent to the graph Fourier transform defined in the previous section. In particular, for our spectral graph wavelet kernel g, the wavelet operator $T_g = g(\mathcal{L})$ acts on a given function f by modulating each Fourier mode as

$$\widehat{T_g f}(\ell) = g(\lambda_\ell)\hat{f}(\ell) \tag{21}$$

Employing the inverse Fourier transform yields

$$(T_g f)(m) = \sum_{\ell=0}^{N-1} g(\lambda_\ell)\hat{f}(\ell)\chi_\ell(m) \tag{22}$$

The wavelet operators at scale t is then defined by $T_g^t = g(t\mathcal{L})$. It should be emphasized that even though the "spatial domain" for the graph is discrete, the domain of the kernel g is continuous and thus the scaling may be defined for any positive real number t.

The spectral graph wavelets are then realized through localizing these operators by applying them to the impulse on a single vertex, i.e.

$$\psi_{t,n} = T_g^t \delta_n \tag{23}$$

Expanding this explicitly in the graph domain shows

$$\psi_{t,n}(m) = \sum_{\ell=0}^{N-1} g(t\lambda_\ell)\chi_\ell^*(n)\chi_\ell(m) \tag{24}$$

Formally, the wavelet coefficients of a given function f are produced by taking the inner product with these wavelets, as

$$W_f(t,n) = \langle \psi_{t,n}, f \rangle \tag{25}$$

Using the orthonormality of the $\{\chi_\ell\}$, it can be seen that the wavelet coefficients can also be achieved directly from the wavelet operators, as

$$W_f(t,n) = \left(T_g^t f\right)(n) = \sum_{\ell=0}^{N-1} g(t\lambda_\ell)\hat{f}(\ell)\chi_\ell(n) \tag{26}$$

Note that from Eq. (24), it can be seen that the wavelets $\psi_{t,n}$ depend on the values of $g(tx)$ only for x in the spectrum of $\mathcal{L}$. This implies that selection of scales appropriate for a particular problem requires some knowledge of the spectrum. As we shall see later in Section 8.1, this will be done using an upper bound on the largest eigenvalue of $\mathcal{L}$.

4.2. Scaling functions

By construction, the spectral graph wavelets $\psi_{t,n}$ are all orthogonal to the null eigenvector χ_0, and nearly orthogonal to χ_ℓ for λ_ℓ near zero. In order to stably represent the low frequency content of f defined on the vertices of the graph, it is convenient to introduce a second class of waveforms, analogous to the low-pass residual scaling functions from classical wavelet analysis. These spectral graph scaling functions have an analogous construction to the spectral graph wavelets. They will be determined by a single real-valued function $h : \mathbb{R}^+ \to \mathbb{R}$, which acts as a low-pass filter, and satisfies $h(0) > 0$ and $h(x) \to 0$ as $x \to \infty$. The scaling functions are then given by $\phi_n = T_h\delta_n = h(\mathcal{L})\delta_n$, and the coefficients by $S_f(n) = \langle \phi_n, f \rangle$.

Introducing the scaling functions helps ensure stable recovery of the original signal f from the wavelet coefficients when the scale parameter t is sampled at a discrete number of values t_j, i.e. so that small perturbations in the wavelet coefficients cannot lead to large changes in recovered f. As we shall see in detail in Section 5.3, stable recovery will be assured if the quantity $G(\lambda) = h(\lambda)^2 + \sum_{j=1}^{J} g(t_j\lambda)^2$ is bounded away from zero on the spectrum of $\mathcal{L}$. Representative choices for h and g are shown in Fig. 1; the exact specification of h and g is deferred to Section 8.1.

Note that the scaling functions defined in this way are present merely to smoothly represent the low frequency content on the graph. They do not generate the wavelets ψ through the two-scale relation as for traditional orthogonal wavelets. The design of the scaling function generator h is thus uncoupled from the choice of wavelet kernel g, provided reasonable tiling for G is achieved.

5. Transform properties

In this section we detail several properties of the spectral graph wavelet transform. We first show an inverse formula for the transform analogous to that for the continuous wavelet transform. We examine the small-scale and large-scale limits, and show that the wavelets are localized in the limit of small scales. Finally we discuss discretization of the scale parameter and the resulting wavelet frames.

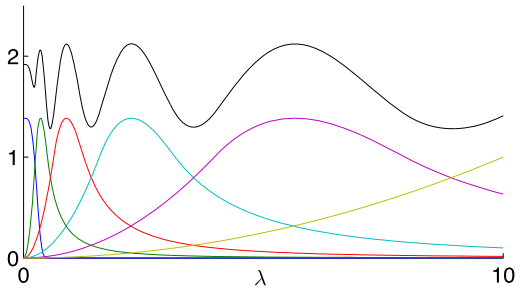

Fig. 1. Scaling function $h(\lambda)$ (blue curve), wavelet generating kernels $g(t_j\lambda)$, and sum of squares G (black curve), for $J = 5$ scales, $\lambda_{max} = 10$, $K = 20$. Details in Section 8.1. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

5.1. Continuous SGWT inverse

In order for a particular transform to be useful for signal processing, and not simply signal analysis, it must be possible to reconstruct a signal corresponding to a given set of transform coefficients. We will show that the spectral graph wavelet transform admits an inverse formula analogous to (4) for the continuous wavelet transform.

Intuitively, the wavelet coefficient $W_f(t,n)$ provides a measure of "how much of" the wavelet $\psi_{t,n}$ is present in the signal f. This suggests that the original signal may be recovered by summing the wavelets $\psi_{t,n}$ multiplied by each wavelet coefficient $W_f(t,n)$. The reconstruction formula below shows that this is indeed the case, subject to a non-constant weight dt/t.

Lemma 5.1. *If the SGWT kernel g satisfies the admissibility condition*

$$\int_0^\infty \frac{g^2(x)}{x}\,dx = C_g < \infty \tag{27}$$

and $g(0) = 0$, then

$$\frac{1}{C_g}\sum_{n=1}^{N}\int_0^\infty W_f(t,n)\psi_{t,n}(m)\frac{dt}{t} = f^{\#}(m) \tag{28}$$

where $f^{\#} = f - \langle \chi_0, f\rangle \chi_0$. In particular, the complete reconstruction is then given by $f = f^{\#} + \hat{f}(0)\chi_0$.

Proof. Using (24) and (26) to express $\psi_{t,n}$ and $W_f(t,n)$ in the graph Fourier basis, the l.h.s. of the above becomes

$$\begin{aligned}&\frac{1}{C_g}\int_0^\infty \frac{1}{t}\sum_n\left(\sum_\ell g(t\lambda_\ell)\chi_\ell(n)\hat{f}(\ell)\sum_{\ell'} g(t\lambda_{\ell'})\chi^*_{\ell'}(n)\chi_{\ell'}(m)\right)dt\\ &\quad= \frac{1}{C_g}\int_0^\infty \frac{1}{t}\left(\sum_{\ell,\ell'} g(t\lambda_{\ell'})g(t\lambda_\ell)\hat{f}(\ell)\chi_{\ell'}(m)\sum_n \chi^*_{\ell'}(n)\chi_\ell(n)\right)dt\end{aligned} \tag{29}$$

The orthonormality of the χ_ℓ implies $\sum_n \chi^*_{\ell'}(n)\chi_\ell(n) = \delta_{\ell,\ell'}$, inserting this above and summing over ℓ' gives

$$= \frac{1}{C_g}\sum_\ell\left(\int_0^\infty \frac{g^2(t\lambda_\ell)}{t}\,dt\right)\hat{f}(\ell)\chi_\ell(m) \tag{30}$$

If g satisfies the admissibility condition, then the substitution $u = t\lambda_\ell$ shows that $\int \frac{g^2(t\lambda_\ell)}{t}\,dt = C_g$ independent of ℓ, except for when $\lambda_\ell = 0$ at $\ell = 0$ when the integral is zero. The expression (30) can be seen as the inverse Fourier transform evaluated at vertex m, where the $\ell = 0$ term is omitted. This omitted term is exactly equal to $\langle \chi_0, f\rangle\chi_0 = \hat{f}(0)\chi_0$, which proves the desired result. □

Note that for the non-normalized Laplacian, χ_0 is constant on every vertex and $f^{\#}$ above corresponds to removing the mean of f. Formula (28) shows that the mean of f may not be recovered from the zero-mean wavelets. The situation is different from the analogous reconstruction formula (4) for the CWT, which shows the somewhat counterintuitive result

that it is possible to recover a nonzero-mean function by summing zero-mean wavelets. This is possible on the real line as the Fourier frequencies are continuous; the fact that it is not possible for the SGWT should be considered a consequence of the discrete nature of the graph domain.

While it is of theoretical interest, we note that this continuous scale reconstruction formula may not provide a practical reconstruction in the case when the wavelet coefficients may only be computed at a discrete number of scales, as is the case for finite computation on a digital computer. We shall revisit this and discuss other reconstruction methods in Sections 5.3 and 7.

5.2. Localization in small-scale limit

One of the primary motivations for the use of wavelets is that they provide simultaneous localization in both frequency and time (or space). It is clear by construction that if the kernel g is localized in the spectral domain, as is loosely implied by our use of the term band-pass filter to describe it, then the associated spectral graph wavelets will all be localized in frequency. In order to be able to claim that the spectral graph wavelets can yield localization in both frequency and space, however, we must analyze their behavior in the space domain more carefully.

For the classical wavelets on the real line, the space localization is readily apparent: if the mother wavelet $\psi(x)$ is well localized in the interval $[-\epsilon, \epsilon]$, then the wavelet $\psi_{t,a}(x)$ will be well localized within $[a - \epsilon t, a + \epsilon t]$. In particular, in the limit as $t \to 0$, $\psi_{t,a}(x) \to 0$ for $x \neq a$. The situation for the spectral graph wavelets is less straightforward to analyze because the scaling is defined implicitly in the Fourier domain. We will nonetheless show that, for g sufficiently regular near 0, the normalized spectral graph wavelet $\psi_{t,j}/\|\psi_{t,j}\|$ will vanish on vertices sufficiently far from j in the limit of fine scales, i.e. as $t \to 0$. This result will provide a quantitative statement of the localization properties of the spectral graph wavelets.

One simple notion of localization for $\psi_{t,n}$ is given by its value on a distant vertex m, e.g. we should expect $\psi_{t,n}(m)$ to be small if n and m are separated, and t is small. Note that $\psi_{t,n}(m) = \langle \psi_{t,n}, \delta_m \rangle = \langle T_g^t \delta_n, \delta_m \rangle$. The operator $T_g^t = g(t\mathcal{L})$ is self-adjoint as $\mathcal{L}$ is self-adjoint. This shows that $\psi_{t,n}(m) = \langle \delta_n, T_g^t \delta_m \rangle$, i.e. a matrix element of the operator T_g^t.

Our approach is based on approximating $g(t\mathcal{L})$ by a low order polynomial in $\mathcal{L}$ as $t \to 0$. As is readily apparent by inspecting Eq. (22), the operator T_g^t depends only on the values of $g_t(\lambda)$ restricted to the spectrum $\{\lambda_\ell\}_{\ell=0}^{N-1}$ of $\mathcal{L}$. In particular, it is insensitive to the values of $g_t(\lambda)$ for $\lambda > \lambda_{N-1}$. If $g(\lambda)$ is smooth in a neighborhood of the origin, then as t approaches 0 the zoomed in $g_t(\lambda)$ can be approximated over the entire interval $[0, \lambda_{N-1}]$ by the Taylor polynomial of g at the origin. In order to transfer the study of the localization property from g to an approximating polynomial, we will need to examine the stability of the wavelets under perturbations of the generating kernel. This, together with the Taylor approximation will allow us to examine the localization properties for integer powers of the Laplacian $\mathcal{L}$.

In order to formulate the desired localization result, we must specify a notion of distance between points m and n on a weighted graph. We will use the shortest-path distance, i.e. the minimum number of edges for any paths connecting m and n:

$$\begin{aligned} &d_G(m,n) = \operatorname*{argmin}_s \{k_1, k_2, \ldots, k_s\} \\ &\text{s.t.} \quad m = k_1, \quad n = k_s, \quad \text{and} \quad a_{k_r,k_{r+1}} > 0 \quad \text{for } 1 \leqslant r < s \end{aligned} \tag{31}$$

Note that as we have defined it, d_G disregards the values of the edge weights. In particular it defines the same distance function on G as on the binarized graph where all of the nonzero edge weights are set to unit weight.

We now state the localization result for integer powers of the Laplacian.

Lemma 5.2. *Let G be a weighted graph, $\mathcal{L}$ the graph Laplacian (normalized or non-normalized) and $s > 0$ an integer. For any two vertices m and n, if $d_G(m,n) > s$ then $(\mathcal{L}^s)_{m,n} = 0$.*

Proof. First note that $\mathcal{L}_{i,j} = 0$ if i and j are distinct vertices that are not connected by a nonzero edge. By repeatedly expressing matrix multiplication with explicit sums, we have

$$\left(\mathcal{L}^s\right)_{m,n} = \sum \mathcal{L}_{m,k_1} \mathcal{L}_{k_1,k_2} \ldots \mathcal{L}_{k_{s-1},n} \tag{32}$$

where the sum is taken over all $s-1$ length sequences $k_1, k_2, \ldots, k_{s-1}$ with $1 \leqslant k_r \leqslant N$. Assume for contradiction that $(L^s)_{m,n} \neq 0$. This is only possible if at least one of the terms in the above sum is nonzero, i.e. there exist $k_1, k_2, \ldots, k_{s-1}$ such that $\mathcal{L}_{m,k_1} \neq 0, L_{k_1,k_2} \neq 0, \ldots, \mathcal{L}_{k_{s-1}} \neq 0$. After removing possibly repeated values of the k_r's, this implies the existence of a path of length less than or equal to s from m to n, so that $d(m,n) \leqslant s$, which contradicts the hypothesis. □

We now proceed to examining how perturbations in the kernel g affect the wavelets in the vertex domain. If two kernels g and $\tilde{g}$ are close to each other in some sense, then the resulting wavelets should be close to each other. More precisely, we have

Lemma 5.3. *Let $\psi_{t,n} = T_g^t \delta_n$ and $\tilde{\psi}_{t,n} = T_{\tilde{g}}^t \delta_n$ be the wavelets at scale t generated by the kernels g and $\tilde{g}$. If $|g(t\lambda) - \tilde{g}(t\lambda)| \leqslant M(t)$ for all $\lambda \in [0, \lambda_{N-1}]$, then $|\psi_{t,n}(m) - \tilde{\psi}_{t,n}(m)| \leqslant M(t)$ for each vertex m. Additionally, $\|\psi_{t,n} - \tilde{\psi}_{t,n}\|_2 \leqslant \sqrt{N} M(t)$.*

Proof. First recall that $\psi_{t,n}(m) = \langle \delta_m, g(t\mathcal{L})\delta_n \rangle$. Thus,

$$\begin{aligned} \left|\psi_{t,n}(m) - \tilde{\psi}_{t,n}(m)\right| &= \left|\left\langle \delta_m, \left(g(t\mathcal{L}) - \tilde{g}(t\mathcal{L})\right)\delta_n \right\rangle\right| \\ &= \left|\sum_\ell \chi_\ell(m)\left(g(t\lambda_\ell) - \tilde{g}(t\lambda_\ell)\right)\chi_\ell^*(n)\right| \\ &\leqslant M(t)\sum_\ell \left|\chi_\ell(m)\chi_\ell(n)^*\right| \end{aligned} \tag{33}$$

where we have used the Parseval relation (20) on the second line. By Cauchy–Schwartz, the above sum over ℓ is bounded by 1 as

$$\sum_\ell \left|\chi_\ell(m)\chi_\ell^*(n)\right| \leqslant \left(\sum_\ell \left|\chi_\ell(m)\right|^2\right)^{1/2}\left(\sum_\ell \left|\chi_\ell^*(n)\right|^2\right)^{1/2} \tag{34}$$

and $\sum_\ell |\chi_\ell(m)|^2 = 1$ for all m, as the χ_ℓ form a complete orthonormal basis.[3] Using this bound in (33) proves the first statement.

The second statement follows immediately as

$$\|\psi_{t,n} - \tilde{\psi}_{t,n}\|_2^2 = \sum_m \left(\psi_{t,n}(m) - \tilde{\psi}_{t,n}(m)\right)^2 \leqslant \sum_m M(t)^2 = NM(t)^2 \qquad \square \tag{35}$$

We will prove the final localization result for kernels g which have a zero of integer multiplicity at the origin. Such kernels can be approximated by a single monomial for small scales.

Lemma 5.4. *Let g be $K+1$ times continuously differentiable, satisfying $g(0) = 0$, $g^{(r)}(0) = 0$ for all $r < K$, and $g^{(K)}(0) = C \neq 0$. Assume that there is some $t' > 0$ such that $|g^{(K+1)}(\lambda)| \leqslant B$ for all $\lambda \in [0, t'\lambda_{N-1}]$. Then, for $\tilde{g}(t\lambda) = (C/K!)(t\lambda)^K$ we have*

$$M(t) = \sup_{\lambda\in[0,\lambda_{N-1}]} \left|g(t\lambda) - \tilde{g}(t\lambda)\right| \leqslant t^{K+1}\frac{\lambda_{N-1}^{K+1}}{(K+1)!}B \tag{36}$$

for all $t < t'$.

Proof. As the first $K-1$ derivatives of g are zero, Taylor's formula with remainder shows, for any values of t and λ,

$$g(t\lambda) = C\frac{(t\lambda)^K}{K!} + g^{(K+1)}(x^*)\frac{(t\lambda)^{K+1}}{(K+1)!} \tag{37}$$

for some $x^* \in [0, t\lambda]$. Now fix $t < t'$. For any $\lambda \in [0, \lambda_{N-1}]$, we have $t\lambda < t'\lambda_{N-1}$, and so the corresponding $x^* \in [0, t'\lambda_{N-1}]$, and so $|g^{(K+1)}(x^*)| \leqslant B$. This implies

$$\left|g(t\lambda) - \tilde{g}(t\lambda)\right| \leqslant B\frac{t^{K+1}\lambda^{K+1}}{(K+1)!} \leqslant B\frac{t^{K+1}\lambda_{N-1}^{K+1}}{(K+1)!} \tag{38}$$

As this holds for all $\lambda \in [0, \lambda_{N-1}]$, taking the sup over λ gives the desired result. $\square$

We are now ready to state the complete localization result. Note that due to the normalization chosen for the wavelets, in general $\psi_{t,n}(m) \to 0$ as $t \to 0$ for all m and n. Thus a non-vacuous statement of localization must include a renormalization factor in the limit of small scales.

Theorem 5.5. *Let G be a weighted graph with Laplacian $\mathcal{L}$. Let g be a kernel satisfying the hypothesis of Lemma 5.4, with constants t' and B. Let m and n be vertices of G such that $d_G(m,n) > K$. Then there exist constants D and t'', such that*

$$\frac{\psi_{t,n}(m)}{\|\psi_{t,n}\|} \leqslant Dt \tag{39}$$

for all $t < \min(t', t'')$.

[3] Orthonormality typically reads as $\sum_m \chi_\ell^*(m)\chi_{\ell'}(m) = \delta_{\ell,\ell'}$. To see the desired statement with the sum over ℓ, set the matrix $U_{i,j} = \chi_j(i)$. Orthonormality implies $U^*U = I$. As matrices commute with their inverses, also $UU^* = I$ which implies $\sum_\ell \chi_l(m)\chi_l^*(n) = \delta_{m,n}$.

Proof. Set $\tilde{g}(\lambda) = \frac{g^{(K)}(0)}{K!}\lambda^K$ and $\tilde{\psi}_{t,n} = T_{\tilde{g}}^t \delta_n$. We have

$$\tilde{\psi}_{t,n}(m) = \frac{g^{(K)}(0)}{K!} t^K \langle \delta_m, \mathcal{L}^K \delta_n \rangle = 0 \tag{40}$$

by Lemma 5.2, as $d_G(m,n) > K$. By the results of Lemmas 5.3 and 5.4, we have

$$\left|\psi_{t,n}(m) - \tilde{\psi}_{t,n}(m)\right| = \left|\psi_{t,n}(m)\right| \leqslant t^{K+1} C' \tag{41}$$

for $C' = \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B$. Writing $\psi_{t,n} = \tilde{\psi}_{t,n} + (\psi_{t,n} - \tilde{\psi}_{t,n})$ and applying the triangle inequality shows

$$\|\tilde{\psi}_{t,n}\| - \|\psi_{t,n} - \tilde{\psi}_{t,n}\| \leqslant \|\psi_{t,n}\| \tag{42}$$

We may directly calculate $\|\tilde{\psi}_{t,n}\| = t^K \frac{g^{(K)}(0)}{K!} \|\mathcal{L}^K \delta_n\|$, and we have $\|\psi_{t,n} - \tilde{\psi}_{t,n}\| \leqslant \sqrt{N} t^{K+1} \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B$ from Lemma 5.4. These imply together that the l.h.s. of (42) is greater than or equal to $t^K (\frac{g^{(K)}(0)}{K!} \|\mathcal{L}^K \delta_n\| - t\sqrt{N} \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B)$. Together with (41), this shows

$$\frac{\psi_{t,n}(m)}{\|\psi_{t,n}\|} \leqslant \frac{tC'}{a - tb} \tag{43}$$

with $a = \frac{g^{(K)}(0)}{K!} \|\mathcal{L}^K \delta_n\|$ and $b = \sqrt{N} \frac{\lambda_{N-1}^{K+1}}{(K+1)!} B$. An elementary calculation shows $\frac{C't}{a-tb} \leqslant \frac{2C'}{a} t$ if $t \leqslant \frac{a}{2b}$. This implies the desired result with $D = \frac{2C'K!}{g^{(K)}(0)\|\mathcal{L}^K \delta_n\|}$ and $t'' = \frac{g^{(K)}(0)\|\mathcal{L}^K \delta_n\|(K+1)}{2\sqrt{N}\lambda_{N-1}^{K+1} B}$. □

Remark. As this localization result uses the shortest-path distance defined without using the edge weights, it is only directly useful for sparse weighted graphs where a significant number of edge weights are exactly zero. Many large-scale graphs which arise in practice are sparse, however, so the class of sparse weighted graphs is of practical significance.

5.3. Spectral graph wavelet frames

The spectral graph wavelets depend on the continuous scale parameter t. For any practical computation, t must be sampled to a finite number of scales. Choosing J scales $\{t_j\}_{j=1}^J$ will yield a collection of NJ wavelets $\psi_{t_j,n}$, along with the N scaling functions ϕ_n.

It is a natural question to ask how well behaved this set of vectors will be for representing functions on the vertices of the graph. We will address this by considering the wavelets at discretized scales as a frame, and examining the resulting frame bounds.

We will review the basic definition of a frame. A more complete discussion of frame theory may be found in [43] and [44]. Given a Hilbert space $\mathcal{H}$, a set of vectors $\Gamma_k \in \mathcal{H}$ form a frame with frame bounds A and B if the inequality

$$A\|f\|^2 \leqslant \sum_k \left|\langle f, \Gamma_k \rangle\right|^2 \leqslant B\|f\|^2 \tag{44}$$

holds for all $f \in \mathcal{H}$.

The frame bounds A and B provide information about the numerical stability of recovering the vector f from inner product measurements $\langle f, \Gamma_k \rangle$. These correspond to the scaling function coefficients $S_f(n)$ and wavelet coefficients $W_f(t_j, n)$ for the frame consisting of the scaling functions and the spectral graph wavelets with sampled scales. As we shall see later in Section 7, the speed of convergence of algorithms used to invert the spectral graph wavelet transform will depend on the frame bounds.

Theorem 5.6. *Given a set of scales $\{t_j\}_{j=1}^J$, the set $F = \{\phi_n\}_{n=1}^N \cup \{\psi_{t_j,n}\}_{j=1\ n=1}^{J\ N}$ forms a frame with bounds A, B given by*

$$\begin{aligned} A &= \min_{\lambda \in [0, \lambda_{N-1}]} G(\lambda) \\ B &= \max_{\lambda \in [0, \lambda_{N-1}]} G(\lambda) \end{aligned} \tag{45}$$

where $G(\lambda) = h^2(\lambda) + \sum_j g(t_j\lambda)^2$.

Proof. Fix f. Using expression (26), we see

$$\begin{aligned} \sum_n \left|W_f(t,n)\right|^2 &= \sum_n \sum_\ell g(t\lambda_\ell) \chi_\ell(n) \hat{f}(\ell) \sum_{\ell'} \left(g(t\lambda_{\ell'}) \chi_{\ell'}(n) \hat{f}(\ell')\right)^* \\ &= \sum_\ell \left|g(t\lambda_\ell)\right|^2 \left|\hat{f}(\ell)\right|^2 \end{aligned} \tag{46}$$

upon rearrangement and using $\sum_n \chi_\ell(n)\chi^*_{\ell'}(n) = \delta_{\ell,\ell'}$. Similarly,

$$\sum_n \left|S_f(n)\right|^2 = \sum_\ell \left|h(\lambda_\ell)\right|^2 \left|\hat{f}(\ell)\right|^2 \tag{47}$$

Denote by Q the sum of squares of inner products of f with vectors in the collection F. Using (46) and (47), we have

$$Q = \sum_\ell \left(\left|h(\lambda_\ell)\right|^2 + \sum_{j=1}^{J} \left|g(t_j\lambda_\ell)\right|^2 \right) \left|\hat{f}(\ell)\right|^2 = \sum_\ell G(\lambda_\ell) \left|\hat{f}(\lambda_\ell)\right|^2 \tag{48}$$

Then by the definition of A and B, we have

$$A \sum_{\ell=0}^{N-1} \left|\hat{f}(\ell)\right|^2 \leqslant Q \leqslant B \sum_{\ell=0}^{N-1} \left|\hat{f}(\ell)\right|^2 \tag{49}$$

Using the Parseval relation $\|f\|^2 = \sum_\ell |\hat{f}(\ell)|^2$ then gives the desired result. □

6. Polynomial approximation and fast SGWT

We have defined the SGWT explicitly in the space of eigenfunctions of the graph Laplacian. The naive way of computing the transform, by directly using Eq. (26), requires explicit computation of the entire set of eigenvectors and eigenvalues of $\mathcal{L}$. This approach scales poorly for large graphs. General purpose eigenvalue routines such as the QR algorithm have computational complexity of $O(N^3)$ and require $O(N^2)$ memory [45]. Direct computation of the SGWT through diagonalizing $\mathcal{L}$ is feasible only for graphs with fewer than a few thousand vertices. In contrast, problems in signal and image processing routinely involve data with hundreds of thousands or millions of dimensions. Clearly, a fast transform that avoids the need for computing the complete spectrum of $\mathcal{L}$ is needed for the SGWT to be a useful tool for practical computational problems.

We present a fast algorithm for computing the SGWT that is based on approximating the scaled generating kernels g by low order polynomials. Given this approximation, the wavelet coefficients at each scale can then be computed as a polynomial of $\mathcal{L}$ applied to the input data. These can be calculated in a way that accesses $\mathcal{L}$ only through repeated matrix-vector multiplication. This results in an efficient algorithm in the important case when the graph is sparse, i.e. contains a small number of edges.

We first show that the polynomial approximation may be taken over a finite range containing the spectrum of $\mathcal{L}$.

Lemma 6.1. *Let $\lambda_{max} \geqslant \lambda_{N-1}$ be any upper bound on the spectrum of $\mathcal{L}$. For fixed $t > 0$, let $p(x)$ be a polynomial approximant of $g(tx)$ with L_∞ error $B = \sup_{x\in[0,\lambda_{max}]} |g(tx) - p(x)|$. Then the approximate wavelet coefficients $\tilde{W}_f(t,n) = (p(\mathcal{L})f)_n$ satisfy*

$$\left|W_f(t,n) - \tilde{W}_f(t,n)\right| \leqslant B\|f\| \tag{50}$$

Proof. Using Eq. (26) we have

$$\begin{aligned} \left|W_f(t,n) - \tilde{W}_f(t,n)\right| &= \left| \sum_\ell g(t\lambda_\ell)\hat{f}(\ell)\chi_\ell(n) - \sum_\ell p(\lambda_\ell)\hat{f}(\ell)\chi_\ell(n) \right| \\ &\leqslant \sum_l \left|g(t\lambda_\ell) - p(\lambda_\ell)\right| \left|\hat{f}(\ell)\chi_\ell(n)\right| \\ &\leqslant B\|f\| \end{aligned} \tag{51}$$

The last step follows from using Cauchy–Schwartz and the orthonormality of the χ_ℓ's. □

Remark. The results of the lemma hold for any $\lambda_{max} \geqslant \lambda_{N-1}$. Computing extremal eigenvalues of a self-adjoint operator is a well-studied problem, and efficient algorithms exist that access $\mathcal{L}$ only through matrix-vector multiplication, notably Arnoldi iteration or the Jacobi–Davidson method [45,46]. In particular, good estimates for λ_{N-1} may be computed at far smaller cost than that of computing the entire spectrum of $\mathcal{L}$.

For fixed polynomial degree M, the upper bound on the approximation error from Lemma 6.1 will be minimized if p is the minimax polynomial of degree M on the interval $[0, \lambda_{max}]$. Minimax polynomial approximations are well known, in particular it has been shown that they exist and are unique [47]. Several algorithms exist for computing minimax polynomials, most notably the Remez exchange algorithm [48].

In this work, however, we will instead use a polynomial approximation given by the truncated Chebyshev polynomial expansion of $g(tx)$. It has been shown that for analytic functions in an ellipse containing the approximation interval, the

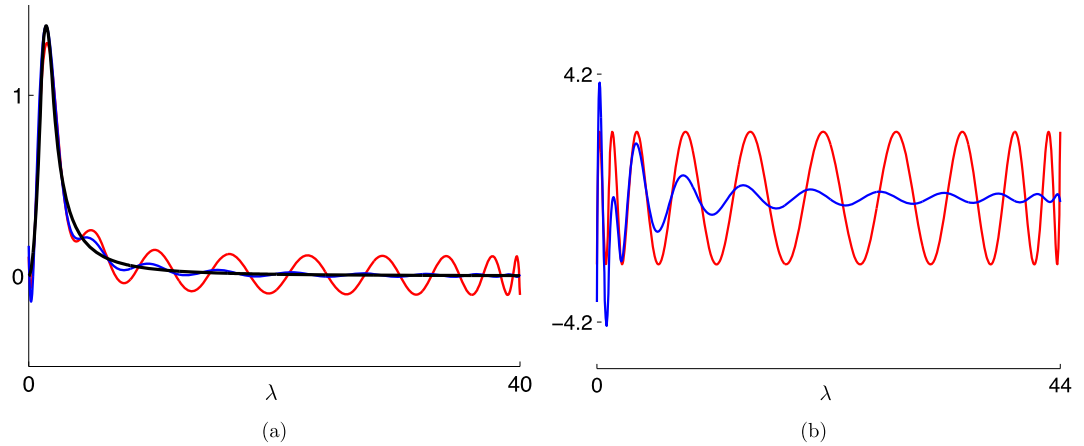

Fig. 2. (a) Wavelet kernel $g(\lambda)$ (black), truncated Chebyshev expansion (blue) and minimax polynomial approximation (red) for degree $m = 20$. Approximation errors shown in (b), truncated Chebyshev expansion has maximum error 0.206, minimax polynomial has maximum error 0.107. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

truncated Chebyshev expansion gives an approximate minimax polynomial [49]. Minimax polynomials of order m are distinguished by having their approximation error reach the same extremal value at $m+2$ points in their domain. As such, they distribute their approximation error across the entire interval. We have observed that for the wavelet kernels we use in this work, truncated Chebyshev expansions result in a maximum error only slightly higher than the true minimax polynomials, and have a much lower approximation error where the wavelet kernel to be approximated is smoothly varying. A representative example of this is shown in Fig. 2. We have observed that for small weighted graphs where the wavelets may be computed directly in the spectral domain, the truncated Chebyshev expansion approximations give slightly lower approximation error than the minimax polynomial approximations computed with the Remez algorithm.

For these reasons, we use approximating polynomials given by truncated Chebyshev expansions. In addition, we will exploit the recurrence properties of the Chebyshev polynomials for efficient evaluation of the approximate wavelet coefficients. An overview of Chebyshev polynomial approximation may be found in [50], we recall here briefly a few of their key properties.

The Chebyshev polynomials $T_k(y)$ may be generated by the stable recurrence relation $T_k(y) = 2yT_{k-1}(y) - T_{k-2}(y)$, with $T_0 = 1$ and $T_1 = y$. For $y \in [-1,1]$, they satisfy the trigonometric expression $T_k(y) = \cos(k \arccos(y))$, which shows that each $T_k(y)$ is bounded between -1 and 1 for $y \in [-1,1]$. The Chebyshev polynomials form an orthogonal basis for $L^2([-1,1], \frac{dy}{\sqrt{1-y^2}})$, the Hilbert space of square integrable functions with respect to the measure $dy/\sqrt{1-y^2}$. In particular they satisfy

$$\int_{-1}^{1} \frac{T_l(y)T_m(y)}{\sqrt{1-y^2}}\,dy = \begin{cases} \delta_{l,m}\pi/2 & \text{if } m,l > 0 \\ \pi & \text{if } m = l = 0 \end{cases} \tag{52}$$

Every $h \in L^2([-1,1], \frac{dy}{\sqrt{1-y^2}})$ has a convergent (in L^2 norm) Chebyshev series

$$h(y) = \frac{1}{2}c_0 + \sum_{k=1}^{\infty} c_k T_k(y) \tag{53}$$

with Chebyshev coefficients

$$c_k = \frac{2}{\pi}\int_{-1}^{1} \frac{T_k(y)h(y)}{\sqrt{1-y^2}}\,dy = \frac{2}{\pi}\int_0^{\pi} \cos(k\theta)h\big(\cos(\theta)\big)\,d\theta \tag{54}$$

We now assume a fixed set of wavelet scales t_n. For each n, approximating $g(t_n x)$ for $x \in [0, \lambda_{max}]$ can be done by shifting the domain using the transformation $x = a(y+1)$, with $a = \lambda_{max}/2$. Denote the shifted Chebyshev polynomials $\overline{T}_k(x) = T_k(\frac{x-a}{a})$. We may then write

$$g(t_n x) = \frac{1}{2}c_{n,0} + \sum_{k=1}^{\infty} c_{n,k}\overline{T}_k(x) \tag{55}$$

valid for $x \in [0, \lambda_{max}]$, with

$$c_{n,k} = \frac{2}{\pi} \int_0^{\pi} \cos(k\theta) g\big(t_n\big(a\big(\cos(\theta)+1\big)\big)\big)\, d\theta \tag{56}$$

For each scale t_j, the approximating polynomial p_j is achieved by truncating the Chebyshev expansion (55) to M_j terms. We may use exactly the same scheme to approximate the scaling function kernel h by the polynomial p_0.

Selection of the values of M_j may be considered a design problem, posing a trade-off between accuracy and computational cost. The fast SGWT approximate wavelet and scaling function coefficients are then given by

$$\begin{aligned} \tilde{W}_f(t_j, n) &= \left(\frac{1}{2} c_{j,0} f + \sum_{k=1}^{M_j} c_{j,k} \overline{T}_k(\mathcal{L}) f \right)_n \\ \tilde{S}_f(n) &= \left(\frac{1}{2} c_{0,0} f + \sum_{k=1}^{M_0} c_{0,k} \overline{T}_k(\mathcal{L}) f \right)_n \end{aligned} \tag{57}$$

The utility of this approach relies on the efficient computation of $\overline{T}_k(\mathcal{L}) f$. Crucially, we may use the Chebyshev recurrence to compute this for each $k < M_j$ accessing $\mathcal{L}$ only through matrix-vector multiplication. As the shifted Chebyshev polynomials satisfy $\overline{T}_k(x) = \frac{2}{a}(x-1)\overline{T}_{k-1}(x) - \overline{T}_{k-2}(x)$, we have for any $f \in \mathbb{R}^N$,

$$\overline{T}_k(\mathcal{L}) f = \frac{2}{a}(\mathcal{L} - I)\big(\overline{T}_{k-1}(\mathcal{L}) f\big) - \overline{T}_{k-2}(\mathcal{L}) f \tag{58}$$

Treating each vector $\overline{T}_k(\mathcal{L}) f$ as a single symbol, this relation shows that the vector $\overline{T}_k(\mathcal{L}) f$ can be computed from the vectors $\overline{T}_{k-1}(\mathcal{L}) f$ and $\overline{T}_{k-2}(\mathcal{L}) f$ with computational cost dominated by a single matrix-vector multiplication by $\mathcal{L}$.

Many weighted graphs of interest are sparse, i.e. they have a small number of nonzero edges. Using a sparse matrix representation, the computational cost of applying $\mathcal{L}$ to a vector is proportional to $|E|$, the number of nonzero edges in the graph. The computational complexity of computing all of the Chebyshev polynomials $T_k(\mathcal{L}) f$ for $k \leqslant M$ is thus $O(M|E|)$. The scaling function and wavelet coefficients at different scales are formed from the same set of $T_k(\mathcal{L}) f$, but by combining them with different coefficients $c_{j,k}$. The computation of the Chebyshev polynomials thus need not be repeated, instead the coefficients for each scale may be computed by accumulating each term of the form $c_{j,k} T_k(\mathcal{L}) f$ as $T_k(\mathcal{L}) f$ is computed for each $k \leqslant M$. This requires $O(N)$ operations at scale j for each $k \leqslant M_j$, giving an overall computational complexity for the fast SGWT of $O(M|E| + N\sum_{j=0}^{J} M_j)$, where J is the number of wavelet scales. In particular, for classes of graphs where $|E|$ scales linearly with N, such as graphs of bounded maximal degree, the fast SGWT has computational complexity $O(N)$. Note that if the complexity is dominated by the computation of the $T_k(\mathcal{L}) f$, there is little benefit to choosing M_j to vary with j.

Applying the recurrence (58) requires memory of size $3N$. The total memory requirement for a straightforward implementation of the fast SGWT would then be $N(J+1) + 3N$.

6.1. Fast computation of adjoint

Given a fixed set of wavelet scales $\{t_j\}_{j=1}^{J}$, and including the scaling functions ϕ_n, one may consider the overall wavelet transform as a linear map $W : \mathbb{R}^N \to \mathbb{R}^{N(J+1)}$ defined by $Wf = ((T_h f)^T, (T_g^{t_1} f)^T, \ldots, (T_g^{t_J} f)^T)^T$. Let $\tilde{W}$ be the corresponding approximate wavelet transform defined by using the fast SGWT approximation, i.e. $\tilde{W} f = ((p_0(\mathcal{L}) f)^T, (p_1(\mathcal{L}) f)^T, \ldots, (p_J(\mathcal{L}) f)^T)^T$. We show that both the adjoint $\tilde{W}^* : \mathbb{R}^{N(J+1)} \to \mathbb{R}^N$ and the composition $W^* W : R^N \to R^N$ can be computed efficiently using Chebyshev polynomial approximation. This is important as several methods for inverting the wavelet transform or using the spectral graph wavelets for regularization can be formulated using the adjoint operator, as we shall see in detail later in Section 7.

For any $\eta \in \mathbb{R}^{N(J+1)}$, we consider η as the concatenation $\eta = (\eta_0^T, \eta_1^T, \ldots, \eta_J^T)^T$ with each $\eta_j \in \mathbb{R}^N$ for $0 \leqslant j \leqslant J$. Each η_j for $j \geqslant 1$ may be thought of as a subband corresponding to the scale t_j, with η_0 representing the scaling function coefficients. We then have

$$\begin{aligned} \langle \eta, Wf \rangle_{N(J+1)} &= \langle \eta_0, T_h f \rangle + \sum_{j=1}^{J} \big\langle \eta_j, T_g^{t_j} f \big\rangle_N \\ &= \big\langle T_h^* \eta_0, f \big\rangle + \left\langle \sum_{j=1}^{J} \big(T_g^{t_j}\big)^* \eta_j, f \right\rangle_N = \left\langle T_h \eta_0 + \sum_{j=1}^{J} T_g^{t_j} \eta_j, f \right\rangle_N \end{aligned} \tag{59}$$

as T_h and each $T_g^{t_j}$ are self-adjoint. As (59) holds for all $f \in \mathbb{R}^N$, it follows that $W^*\eta = T_h\eta_0 + \sum_{j=1}^J T_g^{t_j}\eta_n$, i.e. the adjoint is given by re-applying the corresponding wavelet or scaling function operator on each subband, and summing over all scales.

This can be computed using the same fast Chebyshev polynomial approximation scheme in Eq. (57) as for the forward transform, e.g. as $\tilde{W}^*\eta = \sum_{j=0}^J p_j(\mathcal{L})\eta_j$. Note that this scheme computes the *exact* adjoint of the approximate forward transform, as may be verified by replacing T_h by $p_0(\mathcal{L})$ and $T_g^{t_j}$ by $p_j(\mathcal{L})$ in (59).

We may also develop a polynomial scheme for computing $\tilde{W}^*\tilde{W}$. Naively computing this by first applying $\tilde{W}$, then $\tilde{W}^*$ by the fast SGWT would involve computing $2J$ Chebyshev polynomial expansions. By precomputing the addition of squares of the approximating polynomials, this may be reduced to application of a single Chebyshev polynomial with twice the degree, reducing the computational cost by a factor J. Note first that

$$\tilde{W}^*\tilde{W}f = \sum_{j=0}^J p_j(\mathcal{L})\big(p_j(\mathcal{L})f\big) = \left(\sum_{j=0}^J \big(p_j(\mathcal{L})\big)^2\right) f \tag{60}$$

Set $P(x) = \sum_{j=0}^J (p_j(x))^2$, which has degree $M^* = 2\max\{M_j\}$. We seek to express P in the shifted Chebyshev basis as $P(x) = \frac{1}{2}d_0 + \sum_{k=1}^{M^*} d_k\overline{T}_k(x)$. The Chebyshev polynomials satisfy the product formula

$$T_k(x)T_l(x) = \frac{1}{2}\big(T_{k+l}(x) + T_{|k-l|}(x)\big) \tag{61}$$

which we will use to compute the Chebyshev coefficients d_k in terms of the Chebyshev coefficients $c_{j,k}$ for the individual p_j's.

Expressing this explicitly is slightly complicated by the convention that the $k = 0$ Chebyshev coefficient is divided by 2 in the Chebyshev expansion (55). For convenience in the following, set $c'_{j,k} = c_{j,k}$ for $k \geqslant 1$ and $c'_{j,0} = \frac{1}{2}c_{j,0}$, so that $p_j(x) = \sum_{k=0}^{M_n} c'_{j,k}\overline{T}_k(x)$. Writing $(p_j(x))^2 = \sum_{k=0}^{2*M_n} d'_{j,k}\overline{T}_k(x)$, and applying (61), we compute

$$d'_{j,k} = \begin{cases} \frac{1}{2}({c'_{j,0}}^2 + \sum_{i=0}^{M_n} {c'_{j,i}}^2) & \text{if } k = 0 \\ \frac{1}{2}(\sum_{i=0}^{k} c'_{j,i}c'_{j,k-i} + \sum_{i=0}^{M_j-k} c'_{j,i}c'_{j,k+i} + \sum_{i=k}^{M_j} c'_{j,i}c'_{j,i-k}) & \text{if } 0 < k \leqslant M_j \\ \frac{1}{2}(\sum_{i=k-M_j}^{M_j} c'_{j,i}c'_{j,k-i}) & \text{if } M_j < k \leqslant 2M_j \end{cases} \tag{62}$$

Finally, setting $d_{n,0} = 2d'_{j,0}$ and $d_{j,k} = d'_{j,k}$ for $k \geqslant 1$, and setting $d_k = \sum_{j=0}^J d_{j,k}$ gives the Chebyshev coefficients for $P(x)$. We may then compute

$$\tilde{W}^*\tilde{W}f = P(\mathcal{L})f = \frac{1}{2}d_0 f + \sum_{k=1}^{M^*} d_k\overline{T}_k(\mathcal{L})f \tag{63}$$

following (57).

7. Reconstruction

For most interesting signal processing applications, merely calculating the wavelet coefficients is not sufficient. A wide class of signal processing applications are based on manipulating the coefficients of a signal in a certain transform, and later inverting the transform. For the SGWT to be useful for more than simply signal analysis, it is important to be able to recover a signal corresponding to a given set of coefficients.

The SGWT is an overcomplete transform as there are more wavelets $\psi_{t_j,n}$ than original vertices of the graph. Including the scaling functions ϕ_n in the wavelet frame, the SGWT maps an input vector f of size N to the $N(J+1)$ coefficients $c = Wf$. As is well known, this means that W will have an infinite number of left-inverses M s.t. $MWf = f$. A natural choice among the possible inverses is to use the pseudoinverse $L = (W^*W)^{-1}W^*$. The pseudoinverse satisfies the minimum-norm property

$$Lc = \operatorname*{argmin}_{f\in\mathbb{R}^N} \|c - Wf\|_2 \tag{64}$$

For applications which involve manipulation of the wavelet coefficients, it is very likely to need to apply the inverse to a set of coefficients which no longer lie directly in the image of W. The above property indicates that, in this case, the pseudoinverse corresponds to orthogonal projection onto the image of W, followed by inversion on the image of W.

Given a set of coefficients c, the pseudoinverse will be given by solving the square matrix equation $(W^*W)f = W^*c$. This system is too large to invert directly. Solving it may be performed using any of a number of iterative methods, including the classical frame algorithm [43], and the faster conjugate gradients method [51]. These methods have the property that each step of the computation is dominated by application of W^*W to a single vector. We use the conjugate gradients method, employing the fast polynomial approximation (63) for computing application of $\tilde{W}^*\tilde{W}$.

8. Implementation and examples

In this section we first give the explicit details of the wavelet and scaling function kernels used, and how we select the scales. We then show examples of the spectral graph wavelets on several different real and synthetic data sets.

8.1. SGWT design details

Our choice for the wavelet generating kernel g is motivated by the desire to achieve localization in the limit of fine scales. According to Theorem 5.5, localization can be ensured if g behaves as a monic power of x near the origin. We choose g to be exactly a monic power near the origin, and to have power-law decay for large x. In between, we set g to be a cubic spline such that g and g' are continuous. Our g is parametrized by the integers α and β, and x_1 and x_2 determining the transition regions:

$$g(x;\alpha,\beta,x_1,x_2) = \begin{cases} x_1^{-\alpha}x^{\alpha} & \text{for } x < x_1 \\ s(x) & \text{for } x_1 \leqslant x \leqslant x_2 \\ x_2^{\beta}x^{-\beta} & \text{for } x > x_2 \end{cases} \tag{65}$$

Note that g is normalized such that $g(x_1) = g(x_2) = 1$. The coefficients of the cubic polynomial $s(x)$ are determined by the continuity constraints $s(x_1) = s(x_2) = 1$, $s'(x_1) = \alpha/x_1$ and $s'(x_2) = -\beta/x_2$. All of the examples in this paper were produced using $\alpha = \beta = 2$, $x_1 = 1$ and $x_2 = 2$; in this case $s(x) = -5 + 11x - 6x^2 + x^3$.

The wavelet scales t_j are selected to be logarithmically equispaced between the minimum and maximum scales t_J and t_1. These are themselves adapted to the upper bound λ_{max} of the spectrum of $\mathcal{L}$. The placement of the maximum scale t_1 as well as the scaling function kernel h will be determined by the selection of $\lambda_{min} = \lambda_{max}/K$, where K is a design parameter of the transform. We then set t_1 so that $g(t_1 x)$ has power-law decay for $x > \lambda_{min}$, and set t_J so that $g(t_J x)$ has monic polynomial behavior for $x < \lambda_{max}$. This is achieved by $t_1 = x_2/\lambda_{min}$ and $t_J = x_2/\lambda_{max}$.

For the scaling function kernel we take $h(x) = \gamma \exp(-(\frac{x}{0.6\lambda_{min}})^4)$, where γ is set such that $h(0)$ has the same value as the maximum value of g.

This set of scaling function and wavelet generating kernels, for parameters $\lambda_{max} = 10$, $K = 20$, $\alpha = \beta = 2$, $x_1 = 1$, $x_2 = 2$, and $J = 4$, are shown in Fig. 1.

8.2. Illustrative examples: spectral graph wavelet gallery

As a first example of building wavelets in a point cloud domain, we consider the spectral graph wavelets constructed on the "Swiss roll". This example data set consists of points randomly sampled on a 2d manifold that is embedded in $\mathbb{R}^3$. The manifold is described parametrically by $\vec{x}(s,t) = (t\cos(t)/4\pi, s, t\sin(t)/4\pi)$ for $-1 \leqslant s \leqslant 1$, $\pi \leqslant t \leqslant 4\pi$. For our example we take 500 points sampled uniformly on the manifold.

Given a collection x_i of points, we build a weighted graph by setting edge weights $a_{i,j} = \exp(-\|x_j - x_j\|^2/2\sigma^2)$. For larger data sets this graph could be sparsified by thresholding the edge weights, however we do not perform this here. In Fig. 3 we show the Swiss roll data set, and the spectral graph wavelets at four different scales localized at the same location. We used $\sigma = 0.1$ for computing the underlying weighted graph, and $J = 4$ scales with $K = 20$ for computing the spectral graph wavelets. In many examples relevant for machine learning, data are given in a high-dimensional space that intrinsically lie on some underlying lower-dimensional manifold. This figure shows how the spectral graph wavelets can implicitly adapt to the underlying manifold structure of the data, in particular notice that the support of the coarse scale wavelets diffuses locally along the manifold and does not "jump" to the upper portion of the roll.

A second example is provided by a transportation network. In Fig. 4 we consider a graph describing the road network for Minnesota. In this dataset, edges represent major roads and vertices their intersection points, which often but not always correspond to towns or cities. For this example the graph is unweighted, i.e. the edge weights are all equal to unity independent of the physical length of the road segment represented. In particular, the spatial coordinates of each vertex are used only for displaying the graph and the corresponding wavelets, but do not affect the edge weights. We show wavelets constructed with $K = 100$ and $J = 4$ scales.

Graph wavelets on transportation networks could prove useful for analyzing data measured at geographical locations where one would expect the underlying phenomena to be influenced by movement of people or goods along the transportation infrastructure. Possible example applications of this type include analysis of epidemiological data describing the spread of disease, analysis of inventory of trade goods (e.g. gasoline or grain stocks) relevant for logistics problems, or analysis of census data describing human migration patterns.

Another promising potential application of the spectral graph wavelet transform is for use in data analysis for brain imaging. Many brain imaging modalities, notably functional MRI, produce static or time series maps of activity on the cortical surface. Functional MRI imaging attempts to measure the difference between "resting" and "active" cortical states, typically by measuring MRI signal correlated with changes in cortical blood flow. Due to both constraints on imaging time and the very indirect nature of the measurement, functional MRI images typically have a low signal-to-noise ratio. There is

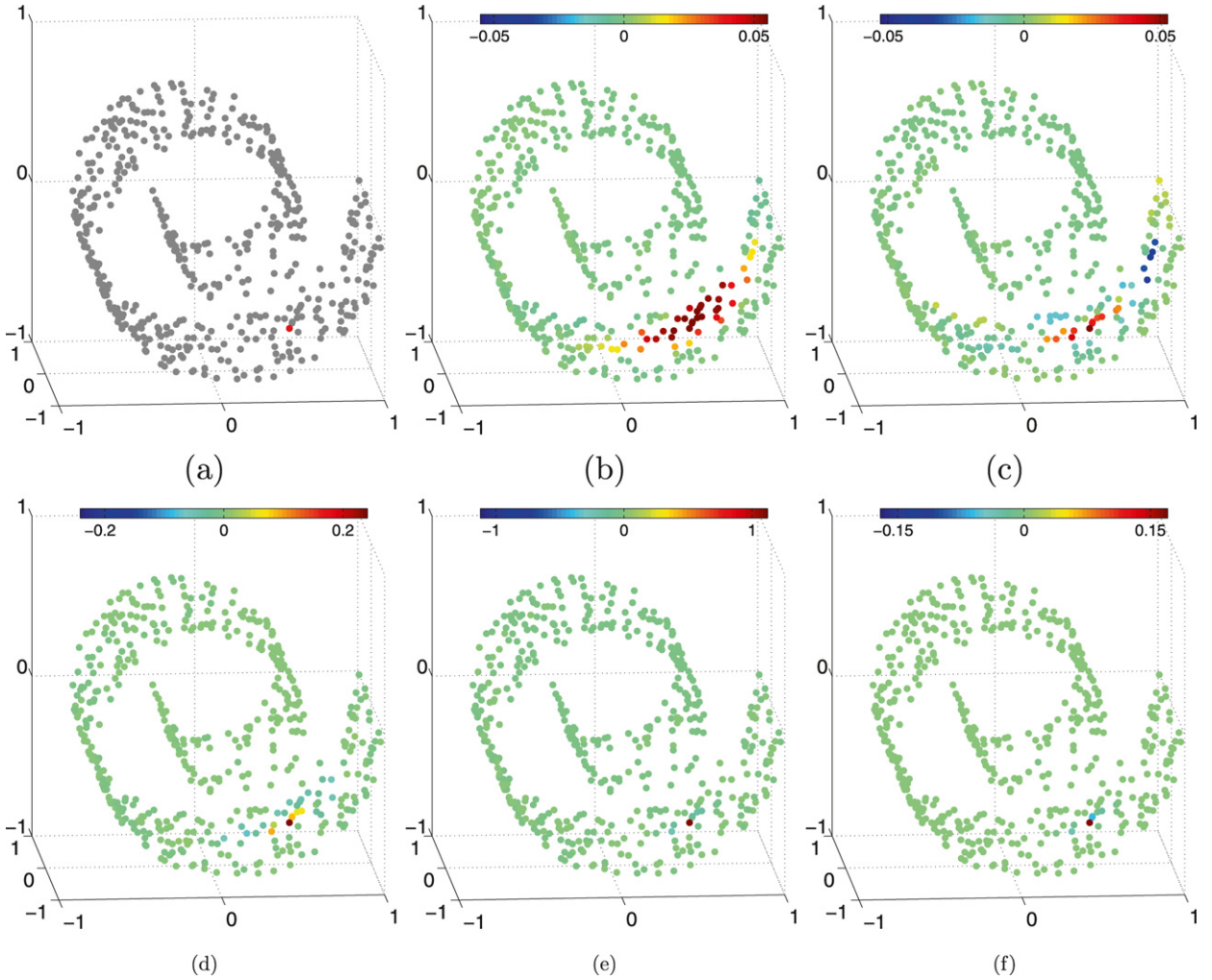

Fig. 3. Spectral graph wavelets on Swiss roll data cloud, with $J = 4$ wavelet scales. (a) Vertex at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

thus a need for techniques for dealing with high levels of noise in functional MRI images, either through direct denoising in the image domain or at the level of statistical hypothesis testing for defining active regions.

Classical wavelet methods have been studied for use in fMRI processing, both for denoising in the image domain [52] and for constructing statistical hypothesis testing [53,54]. The power of these methods relies on the assumption that the underlying cortical activity signal is spatially localized, and thus can be efficiently represented with localized wavelet waveforms. However, such use of wavelets ignores the anatomical connectivity of the cortex.

A common view of the cerebral cortex is that it is organized into distinct functional regions which are interconnected by tracts of axonal fibers. Recent advances in diffusion MRI imaging, notable diffusion tensor imaging (DTI) and diffusion spectrum imaging (DSI), have enabled measuring the directionality of fiber tracts in the brain. By tracing the fiber tracts, it is possible to non-invasively infer the anatomical connectivity of cortical regions. This raises an interesting question of whether knowledge of anatomical connectivity can be exploited for processing of image data on the cortical surface.

We[4] have begun to address this issue by implementing the spectral graph wavelets on a weighted graph which captures the connectivity of the cortex. Details of measuring the cortical connection matrix are described in [55]. Very briefly, the cortical surface is first subdivided into 998 Regions of Interest (ROI's). A large number of fiber tracts are traced, then the connectivity of each pair of ROI's is proportional to the number of fiber tracts connecting them, with a correction term depending on the measured fiber length. The resulting symmetric matrix can be viewed as a weighted graph where the vertices are the ROI's. Fig. 5 shows example spectral graph wavelets computed on the cortical connection graph, visualized by mapping the ROI's back onto a 3d model of the cortex. Only the right hemisphere is shown, although the wavelets are

[4] In collaboration with Dr Leila Cammoun and Prof. Jean-Philippe Thiran, EPFL, Lausanne, Dr Patric Hagmann and Prof. Reto Meuli, CHUV, Lausanne.

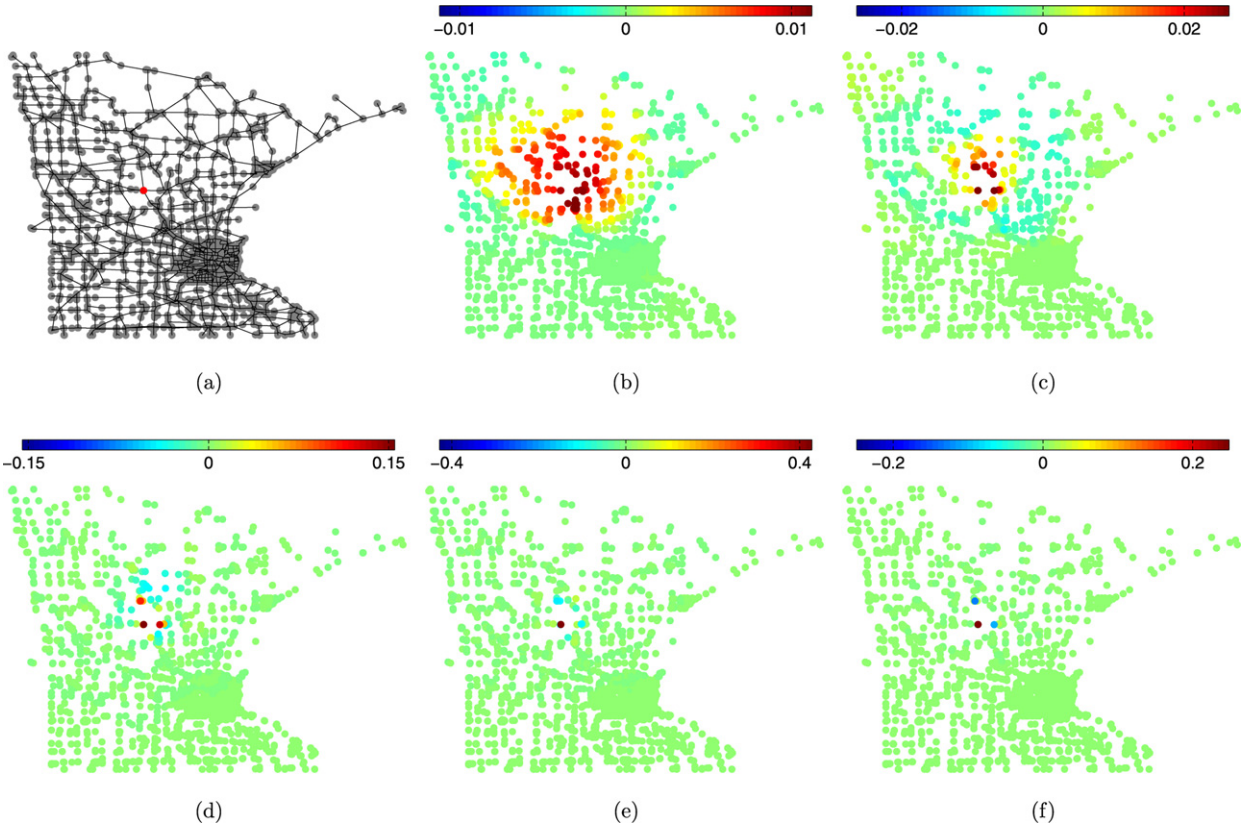

Fig. 4. Spectral graph wavelets on Minnesota road graph, with $K = 100$, $J = 4$ scales. (a) Vertex at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

defined on both hemispheres. For future work we plan to investigate the use of these cortical graph wavelets for use in regularization and denoising of functional MRI data.

A final interesting application for the spectral graph wavelet transform is the construction of wavelets on irregularly shaped domains. As a representative example, consider that for some problems in physical oceanography one may need to manipulate scalar data, such as water temperature or salinity, that is only defined on the surface of a given body of water. In order to apply wavelet analysis for such data, one must adapt the transform to the potentially very complicated boundary between land and water. The spectral wavelets handle the boundary implicitly and gracefully. As an illustration we examine the spectral graph wavelets where the domain is determined by the surface of a lake.

For this example the lake domain is given as a mask defined on a regular grid. We construct the corresponding weighted graph having vertices that are grid points inside the lake, and retaining only edges connecting neighboring grid points inside the lake. We set all edge weights to unity. The corresponding graph Laplacian is thus exactly the 5-point stencil (13) for approximating the continuous operator $-\nabla^2$ on the interior of the domain; while at boundary points the graph Laplacian is modified by the deletion of edges leaving the domain. We show an example wavelet on lake Geneva in Fig. 6. Shoreline data was taken from the GSHHS database [56] and the lake mask was created on a 256×153 pixel grid using an azimuthal equidistant projection, with a scale of 232 meters/pixel. The wavelet displayed is from the coarsest wavelet scale, using the generating kernel described in Section 8.1 with parameters $K = 100$ and $J = 5$ scales.

For this type of domain derived by masking a regular grid, one may compare the wavelets with those obtained by simply truncating the wavelets derived from a large regular grid. As the wavelets have compact support, the true and truncated wavelets will coincide for wavelets located far from the irregular boundary. As can be seen in Fig. 6, however, they are quite different for wavelets located near the irregular boundary. This comparison gives direct evidence for the ability of the spectral graph wavelets to adapt gracefully and automatically to the arbitrarily shaped domain.

We remark that the regular sampling of data within the domain may be unrealistic for problems where data are collected at irregularly placed sensor locations. The spectral graph wavelet transform could also be used in this case by constructing a graph with vertices at the sensor locations, however we have not considered such an example here.

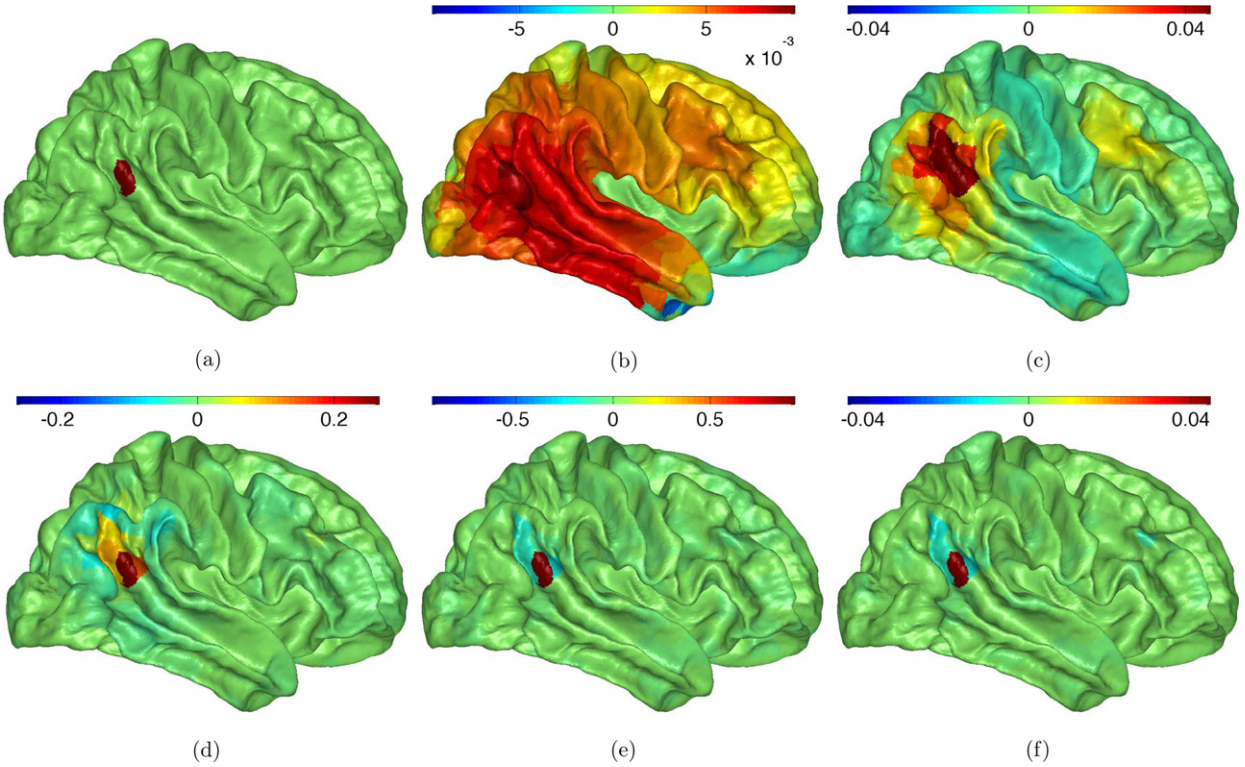

Fig. 5. Spectral graph wavelets on cerebral cortex, with $K = 50$, $J = 4$ scales. (a) ROI at which wavelets are centered, (b) scaling function, (c)–(f) wavelets, scales 1–4.

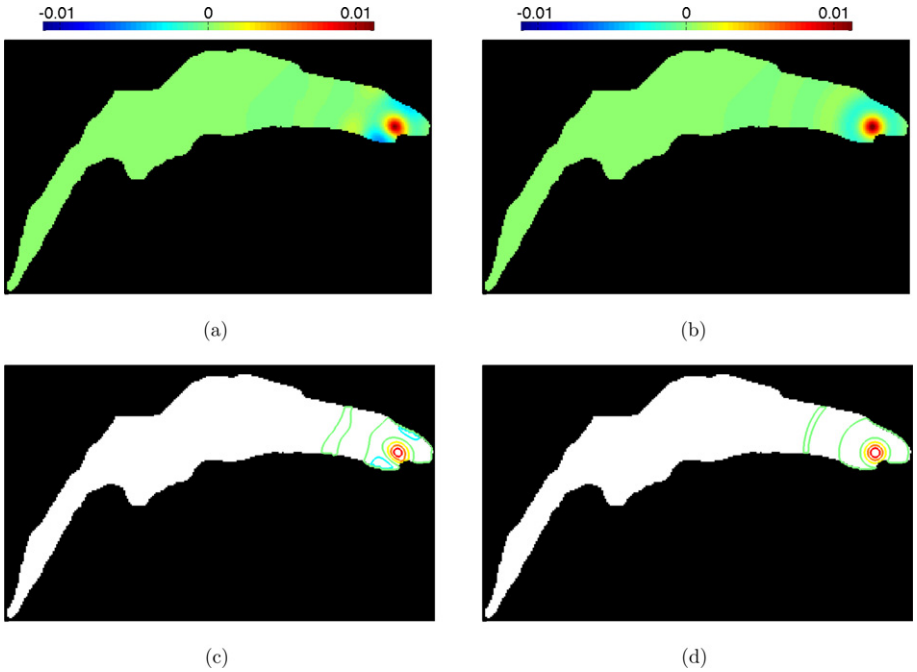

Fig. 6. Spectral graph wavelets on lake Geneva domain (spatial map (a), contour plot (c)); compared with truncated wavelets from graph corresponding to complete mesh (spatial map (b), contour plot (d)). Note that the graph wavelets adapt to the geometry of the domain.

9. Conclusions and future work

We have presented a framework for constructing wavelets on arbitrary weighted graphs. By analogy with classical wavelet operators in the Fourier domain, we have shown that scaling may be implemented in the spectral domain of the graph Laplacian. We have shown that the resulting spectral graph wavelets are localized in the small-scale limit, and form a frame with easily calculable frame bounds. We have detailed an algorithm for computing the wavelets based on Chebyshev polynomial approximation that avoids the need for explicit diagonalization of the graph Laplacian, and allows the application of the transform to large graphs. Finally we have shown examples of the wavelets on graphs arising from several different potential application domains.

There are many possible directions for future research for improving or extending the SGWT. One property of the transform presented here is that, unlike classical orthogonal wavelet transforms, we do not subsample the transform at coarser spatial scales. As a result the SGWT is overcomplete by a factor of $J + 1$ where J is the number of wavelet scales. Subsampling of the SGWT can be determined by selecting a mask of vertices at each scale corresponding to the centers of the wavelets to preserve. This is a more difficult problem on an arbitrary weighted graph than on a regular mesh, where one may exploit the regular geometry of the mesh to perform dyadic subsampling at each scale. An interesting question for future research would be to investigate an appropriate criterion for determining a good selection of wavelets to preserve after subsampling. As an example, one may consider preserving the frame bounds as much as possible under the constraint that the overall overcompleteness should not exceed a specified factor.

A related question is to consider how the SGWT would interact with graph contraction. A weighted graph may be contracted by partitioning its vertices into disjoint sets; the resulting contracted graph has vertices equal to the number of partitions and edge weights determined by summing the weights of the edges connecting any two partitions. Repeatedly contracting a given weighted graph could define a multiscale representation of the weighted graph. Calculating a single scale of the spectral graph wavelet transform for each of these contracted graphs would then yield a multiscale wavelet analysis. This proposed scheme is inspired conceptually by the fast wavelet transform for classical orthogonal wavelets, based on recursive filtering and subsampling. The question of how to automatically define the contraction at each scale on an arbitrary irregular graph is itself a difficult research problem.

The spectral graph wavelets presented here are not directional. In particular when constructed on regular meshes they yield radially symmetric waveforms. This can be understood as in this case the graph Laplacian is the discretization of the isotropic continuous Laplacian. In the field of image processing, however, it has long been recognized that directionally selective filters are more efficient at representing image structure. This raises the interesting question of how, and when, graph wavelets can be constructed which have some directionality. Intuitively, this will require some notion of local directionality, i.e. some way of defining directions of all of the neighbors of a given vertex. As this would require the definition of additional structure beyond the raw connectivity information, it may not be appropriate for completely arbitrary graphs. For graphs which arise from sampling a known orientable manifold, such as the meshes with irregular boundary used in Fig. 6, one may infer such local directionality from the original manifold.

For some problems it may be useful to construct graphs that mix both local and nonlocal connectivity information. As a concrete example consider the cortical graph wavelets shown in Fig. 5. As the vertices of the graph correspond to sets of MRI voxels grouped into ROI's, the wavelets are defined on the ROI's and thus cannot be used to analyze data defined on the scale of individual voxels. Analyzing voxel scale data with the SGWT would require constructing a graph with vertices corresponding to individual voxels. However, the nonlocal connectivity is defined only on the scale of the ROI's. One way of defining the connectivity for the finer graph would be as a sum $A^{nonlocal} + A^{local}$, where $A^{nonlocal}_{m,n}$ is the weight of the connection between the ROI containing vertex m and the ROI containing vertex n, and $A^{local}_{m,n}$ indexes whether m and n are spatial neighbors. Under this scheme we consider A^{local} as implementing a "default" local connectivity not arising from any particular measurement. Considering this raises interesting questions of how to balance the relative contributions of the local and nonlocal connectivities, as well as how the special structure of the hybrid connectivity matrix could be exploited for efficient computation.

The particular form of the wavelet generating kernel g used in the examples illustrating this work was chosen in a somewhat ad hoc manner. Aside from localization in the small-scale limit which required polynomial behavior of g at the origin, we have avoided detailed analysis of how the choice of g affects the wavelets. In particular, we have not chosen g and the choice of spatial scales to optimize the resulting frame bounds. More detailed investigation is called for regarding optimizing the design of g for different applications.

The fast Chebyshev polynomial approximation scheme we describe here could itself be useful independent of its application for computing the wavelet transform. One application could be for filtering of data on irregularly shaped domains, such as described in Fig. 6. For example, smoothing data on such a domain by convolving with a Gaussian kernel is confounded by the problem that near the edges the kernel would extend off of the domain. As an alternative, one could express the convolution as multiplication by a function in the Fourier domain, approximate this function with a Chebyshev polynomial, and then apply the algorithm described in this paper. This could also be used for band-pass or high-pass filtering of data on irregular domains, by designing appropriate filters in the spectral domain induced by the graph Laplacian.

The Chebyshev approximation scheme may also be useful for machine learning problems on graphs. Some recent work has studied using the "diffusion kernel" $K_t = e^{-t\mathcal{L}}$ for use with kernel-based machine learning algorithms [57]. The Cheby-

shev polynomial scheme provides a fast way to approximate this exponential that may be useful for large problems on unstructured yet sparse graphs.

Software

A MATLAB implementation of the SGWT is available online at wiki.epfl.ch/sgwt.

References

[1] J. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. Signal Process. 41 (12) (1993) 3445–3462.
[2] A. Said, W. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits Syst. Video Technol. 6 (3) (1996) 243–250.
[3] M. Hilton, Wavelet and wavelet packet compression of electrocardiograms, IEEE Trans. Biomed. Eng. 44 (5) (1997) 394–402.
[4] R. Buccigrossi, E. Simoncelli, Image compression via joint statistical characterization in the wavelet domain, IEEE Trans. Image Process. 8 (12) (1999) 1688–1701.
[5] D. Taubman, M. Marcellin, JPEG2000: Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, 2002.
[6] D.L. Donoho, I.M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (1994) 425–455.
[7] S. Chang, B. Yu, M. Vetterli, Adaptive wavelet thresholding for image denoising and compression, IEEE Trans. Image Process. 9 (9) (2000) 1532–1546.
[8] L. Sendur, I. Selesnick, Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency, IEEE Trans. Signal Process. 50 (11) (2002) 2744–2756.
[9] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. 12 (2003) 1338–1351.
[10] I. Daubechies, G. Teschke, Variational image restoration by means of wavelets: Simultaneous decomposition, deblurring, and denoising, Appl. Comput. Harmon. Anal. 19 (1) (2005) 1–16.
[11] J.-L. Starck, A. Bijaoui, Filtering and deconvolution by the wavelet transform, Signal Process. 35 (3) (1994) 195–211.
[12] D.L. Donoho, Nonlinear solution of linear inverse problems by wavelet-vaguelette decomposition, Appl. Comput. Harmon. Anal. 2 (2) (1995) 101–126.
[13] E. Miller, A.S. Willsky, A multiscale approach to sensor fusion and the solution of linear inverse problems, Appl. Comput. Harmon. Anal. 2 (2) (1995) 127–147.
[14] R. Nowak, E. Kolaczyk, A statistical multiscale framework for Poisson inverse problems, IEEE Trans. Inform. Theory 46 (5) (2000) 1811–1825.
[15] J. Bioucas-Dias, Bayesian wavelet-based image deconvolution: A GEM algorithm exploiting a class of heavy-tailed priors, IEEE Trans. Image Process. 15 (4) (2006) 937–951.
[16] R. Manthalkar, P.K. Biswas, B.N. Chatterji, Rotation and scale invariant texture features using discrete wavelet packet transform, Pattern Recognit. Lett. 24 (14) (2003) 2455–2462.
[17] P. Flandrin, Wavelet analysis and synthesis of fractional Brownian motion, IEEE Trans. Inform. Theory 38 (2) (1992) 910–917 (Part 2).
[18] D. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, 1999, pp. 1150–1157.
[19] C. Apté, F. Damerau, S.M. Weiss, Automated learning of decision rules for text categorization, ACM Trans. Inf. Syst. 12 (3) (1994) 233–251.
[20] F.K. Chung, Spectral Graph Theory, CBMS Reg. Conf. Ser. Math., vol. 92, AMS Bookstore, 1997.
[21] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, 1998.
[22] P.J. Burt, E.H. Adelson, The Laplacian pyramid as a compact image code, IEEE Trans. Commun. 31 (4) (1983) 532–540.
[23] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, D.J. Heeger, Shiftable multi-scale transforms, IEEE Trans. Inform. Theory 38 (2) (1992) 587–607, special issue on wavelets.
[24] N. Kingsbury, Complex wavelets for shift invariant analysis and filtering of signals, Appl. Comput. Harmon. Anal. 10 (3) (2001) 234–253.
[25] E. Candes, D. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities, Comm. Pure Appl. Math. 57 (2003) 219–266.
[26] G. Peyré, S. Mallat, Orthogonal bandlet bases for geometric images approximation, Comm. Pure Appl. Math. 61 (9) (2008) 1173–1212.
[27] J. Antoine, P. Vandergheynst, Wavelets on the 2-sphere: A group-theoretical approach, Appl. Comput. Harmon. Anal. 7 (3) (1999) 262–291.
[28] Y. Wiaux, J.D. McEwen, P. Vandergheynst, O. Blanc, Exact reconstruction with directional wavelets on the sphere, Mon. Not. R. Astron. Soc. 388 (2008) 770.
[29] J.-P. Antoine, I. Bogdanova, P. Vandergheynst, The continuous wavelet transform on conic sections, Int. J. Wavelets Multiresolut. Inf. Process. 6 (2) (2008) 137–156.
[30] M. Crovella, E. Kolaczyk, Graph wavelets for spatial traffic analysis, in: INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, 2003, pp. 1848–1857.
[31] A. Smalter, J. Huan, G. Lushington, Graph wavelet alignment kernels for drug virtual screening, J. Bioinform. Comput. Biol. 7 (2009) 473–497.
[32] M. Jansen, G.P. Nason, B.W. Silverman, Multiscale methods for data on graphs and irregular multidimensional situations, J. R. Stat. Soc. Ser. B Stat. Methodol. 71 (1) (2009) 97–125.
[33] F. Murtagh, The Haar wavelet transform of a dendrogram, J. Classification 24 (1) (2007) 3–32.
[34] A.B. Lee, B. Nadler, L. Wasserman, Treelets an adaptive multi-scale basis for sparse unordered data, Ann. Appl. Stat. 2 (2008) 435–471.
[35] R.R. Coifman, M. Maggioni, Diffusion wavelets, Appl. Comput. Harmon. Anal. 21 (2006) 53–94.
[36] M. Maggioni, H. Mhaskar, Diffusion polynomial frames on metric measure spaces, Appl. Comput. Harmon. Anal. 24 (3) (2008) 329–353.
[37] D. Geller, A. Mayeli, Continuous wavelets on compact manifolds, Math. Z. 262 (2009) 895–927.
[38] A. Grossmann, J. Morlet, Decomposition of Hardy functions into square integrable wavelets of constant shape, SIAM J. Math. Anal. 15 (4) (1984) 723–736.
[39] M. Hein, J. Audibert, U. von Luxburg, From graphs to manifolds – weak and strong pointwise consistency of graph Laplacians, in: P. Auer, R. Meir (Eds.), Proc. 18th Conf. Learning Theory (COLT), in: Lecture Notes in Comput. Sci., vol. 3559, Springer-Verlag, 2005, pp. 470–485.
[40] A. Singer, From graph to manifold Laplacian: The convergence rate, Appl. Comput. Harmon. Anal. 21 (1) (2006) 128–134, diffusion maps and wavelets.
[41] M. Belkin, P. Niyogi, Towards a theoretical foundation for Laplacian-based manifold methods, J. Comput. System Sci. 74 (8) (2008) 1289–1308, learning theory, 2005.
[42] M. Reed, B. Simon, Methods of Modern Mathematical Physics, vol. 1: Functional Analysis, Academic Press, 1980.
[43] I. Daubechies, Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, 1992.
[44] C.E. Heil, D.F. Walnut, Continuous and discrete wavelet transforms, SIAM Rev. 31 (4) (1989) 628–666.
[45] D. Watkins, The Matrix Eigenvalue Problem – GR and Krylov Subspace Methods, Society for Industrial and Applied Mathematics, 2007.
[46] G.L.G. Sleijpen, H.A.V. der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, SIAM J. Matrix Anal. Appl. 17 (2) (1996) 401–425.

[47] E. Cheney, Introduction to Approximation Theory, McGraw–Hill, New York, 1966.
[48] W. Fraser, A survey of methods of computing minimax and near-minimax polynomial approximations for functions of a single independent variable, J. Assoc. Comput. Mach. 12 (1965) 295–314.
[49] K.O. Geddes, Near-minimax polynomial approximation in an elliptical region, SIAM J. Numer. Anal. 15 (6) (1978) 1225–1233.
[50] G.M. Phillips, Interpolation and Approximation by Polynomials, CMS Books Math., Springer-Verlag, 2003.
[51] K. Grochenig, Acceleration of the frame algorithm, IEEE Trans. Signal Process. 41 (12) (1993) 3331–3340.
[52] S. Zaroubi, G. Goelman, Complex denoising of MR data via wavelet analysis: Application for functional MRI, Magnetic Resonance Imaging 18 (1) (2000) 59–68.
[53] U. Ruttimann, M. Unser, R. Rawlings, D. Rio, N. Ramsey, V. Mattay, D. Hommer, J. Frank, D. Weinberger, Statistical analysis of functional MRI data in the wavelet domain, IEEE Trans. Medical Imaging 17 (2) (1998) 142–154.
[54] D.V.D. Ville, T. Blu, M. Unser, Integrated wavelet processing and spatial statistical testing of fMRI data, Neuroimage 23 (4) (2004) 1472–1485.
[55] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C.J. Honey, V.J. Wedeen, O. Sporns, Mapping the structural core of human cerebral cortex, PLoS Comput. Biol. 6 (7) (2008) e159.
[56] P. Wessel, W.H.F. Smith, A global, self-consistent, hierarchical, high-resolution shoreline database, J. Geophys. Res. 101 (B4) (1996) 8741–8743, http://www.ngdc.noaa.gov/mgg/shorelines/gshhs.html.
[57] R.I. Kondor, J. Lafferty, Diffusion kernels on graphs and other discrete input spaces, in: Proceedings of the 19th International Conference on Machine Learning, 2002.