LocLets: Localized Graph Wavelets for Processing Frequency Sparse Signals on Graphs

Basile de Loynes, Fabien Navarro, Baptiste Olivier

LocLets: Localized Graph Wavelets for Processing Frequency Sparse Signals on Graphs

Basile de Loynes*, Fabien Navarro†, Baptiste Oliver‡

June 18, 2019

Abstract

In this article, a new family of graph wavelets, abbreviated LocLets for *Loc*alized graph wave*Lets*, is introduced. These wavelets are localized in the Fourier domain on subsets of the graph Laplacian spectrum. LocLets are built upon the Spectral Graph Wavelet Transform (SGWT) and adapt better to signals that are sparse in the Fourier domain than standard SGWT. In fact, as a refinement of SGWT, LocLets benefits from the Chebyshev's machinery to ensure the LocLets transform remains an efficient and scalable tool for signal processing on large graphs. In addition, LocLets exploits signals sparsity in various ways: compactness, efficiency and ease of use of the transform are improved for sparse signals in the Fourier domain. As typical examples of such sparse signals, there are smooth and highly non-smooth signals. For these latter signals, their mixtures or even a wider class of signals, it is shown in this paper that LocLets provide substantial improvements in standard noise reduction tasks compared to advanced graph-wavelet based methods.

Keywords: Graph Signal Processing; Spectral Graph Theory; Wavelet; Chebyshev Approximations; Graph Fourier Transform; Frame; Denoising.

1 Introduction

Graph Signal Processing (GSP) aims to generalize the standard framework of signal analysis to signals defined on graphs by combining the theoretical concepts of algebraic and spectral graphs with harmonic analysis (see for instance [4, 1] and references therein). Such an extension becomes necessary to process structured data. We refer the reader to [16] for an introduction to this emerging field and to [14] for an overview of recent developments, challenges and applications. In general, two types of problems can be distinguished according to whether the underlying graph is known or unknown. The first case corresponds to the set up of a sampled signal at certain irregularly spaced points (intersections of a transportation network, nodes in a computer network, ...). In the second case, a graph is constructed from the data itself, it is generally interpreted as a noisy realization of one or several distributions supported by a submanifold of the Euclidean space. In this latter context, the theoretical submanifold is somehow approximated using standard methods such as k-NN, ε-graph and their Gaussian weighted versions. In any of these cases, the framework is actually similar: it consists of a graph (given by the application or by the data) and signals are real valued functions defined on the vertices of the graph.

*Basile de Loynes
ENSAI, France, E-mail: basile.deloynes@ensai.fr
†Fabien Navarro
CREST, ENSAI, France, E-mail: fabien.navarro@ensai.fr
‡Baptiste Oliver
Orange Labs, France. E-mail: baptiste.olivier@orange.com

Related Work

Basically, to any frame corresponds a signal analysis theory (see [2, 3] or [13, Chapter 5], for example). Therefore, the main challenge of signal analysis consists in finding an appropriate frame for the problem of interest. For instance, the spectral decomposition of the graph Laplacian gives rise to an orthonormal basis called the Fourier basis by analogy with the standard case. If this frame is well localized in the frequency by definition, it is no longer verified in the graph domain. This phenomenon is illustrated by the fact that the eigenvectors corresponding to the upper part of Laplacian spectrum tend to be more oscillating than those from the bottom of the spectrum (see for example [19, Fig. 1.6, p. 28] for an illustration). To overcome this problem, [11] developed a fairly general construction of a frame enjoying the usual properties of standard wavelets: each vector of the frame is localized both in the graph domain and the spectral domain. The transform associated with this frame is named Spectral Graph Wavelet Transform (SGWT), its precise definition is recalled in Section 2. Other choices were also suggested (we refer to [5, 12, 18, 10]). Among them authors of [12, 10] ensure the wavelet transform is orthogonal. This property implies in particular that the inverse transform is simply the adjoint of the direct transform.

Contribution

With the notion of graph Laplacian comes a notion of regularity of a signal $f \in \mathbb{R}^V$. Intuitively, a smooth signal will not vary much between two vertices that are close in the graph. This regularity property can be read in the Fourier domain: a very smooth signal will be correctly represented in the Fourier domain with a small number of eigenvectors associated with the lowest spectral values; on the contrary, non-smooth signals (*i.e.* highly oscillating) are represented with eigenvectors corresponding to the upper part of the spectrum. Both types of signal are said *frequency sparse* (see Section 3 for the precise definition).

Basically, LocLets, defined in Section 2, aims to take advantage of this prior information on the signal. More precisely, LocLets provides a richer collection of wavelets by applying SGWT not to the original Laplacian but to its projections on several pairwise orthogonal eigenspaces. Thus, it is expected to obtain sparser wavelet representation when the signal is localized in the Fourier domain, which reveals a specific regularity. It should be noted that, in some areas of application, signal regularity can be a reasonable assumption: for example, physical measurements are often diffusive, some features are naturally homogeneous inside communities of social networks... Intuitively, LocLets induce frames that are better suited to these sparse frequency signal subclasses.

Experimental results[1] are gathered in Section 5. First, a toy example (a small weighted graph of size $n = 10$ is generated) is considered and the performance of LocLets is compared with the denoising procedure introduced in [10]. As a second test, the well-known "Swiss roll" data set (with $n = 1000$ points) is considered. Again, our procedure shows fairly better performance than the one in [10].

2 Localized Wavelet Transforms

The main idea behind LocLets is to apply SGWT not to the signal itself, but separately to each element of the signal decomposition with respect to some pairwise orthogonal eigenspaces of the graph Laplacian.

[1] Source code available at https://bitbucket.org/batistbucket/loclets/src/master/

2.1 Fast Spectral Graph Wavelet Transform

Let G be an undirected weighted graph, with set of vertices V, and weights $(w_{ij})_{i,j\in V}$ with the property $w_{ij} = w_{ji}$ for $i, j \in V$. The size of the graph is the number of nodes $n = |V|$. The (unnormalized) graph Laplacian matrix $L \in \mathbb{R}^{V\times V}$ associated with G is the symmetric matrix defined as $L = D - W$, where W is the matrix of weights with coefficients $(w_{ij})_{i,j\in V}$, and D the diagonal matrix with diagonal coefficients $D_{ii} = \sum_{j\in V} w_{ij}$. A signal f on the graph G is a function $f : V \to \mathbb{R}$.

Let $\mathfrak{F} = \{r_i\}_{i\in I}$ be a frame of vectors of $\mathbb{R}^V$, that is a family of vectors in $\mathbb{R}^V$ such that there exist $A, B > 0$ such that for all $f \in \mathbb{R}^V$

$$A\|f\|_2^2 \leq \sum_{i\in I} |\langle f, r_i\rangle|^2 \leq B\|f\|_2^2. \tag{1}$$

The linear map $T_{\mathfrak{F}} : \mathbb{R}^V \to \mathbb{R}^I$ defined for $f \in \mathbb{R}^V$ by $T_{\mathfrak{F}} f = (\langle f, r_i\rangle)_{i\in I}$ is called the *analysis* operator. The *synthesis* operator is the adjoint of $T_{\mathfrak{F}}$, it is the linear map $T_{\mathfrak{F}}^* : \mathbb{R}^I \to \mathbb{R}^V$ defined for a vector of coefficients $(c_i)_{i\in I}$ by $T_{\mathfrak{F}}^*(c_i)_{i\in I} = \sum_{i\in I} c_i r_i$. As a frame is in particular a generating family of $\mathbb{R}^V$, a signal $f \in \mathbb{R}^V$ can be recovered from its coefficients $T_{\mathfrak{F}} f$ with the help of the synthesis operator.

2.1.1 Spectral Information of *L*nd Functional Calculus

Let us recall some useful definitions and facts related to the functional calculus applied to symmetric matrix L. Recall that L has a spectral decomposition

$$L = \sum_{\ell} \lambda_\ell \langle \chi_\ell, \cdot\rangle \chi_\ell,$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0$ denote the (ordered) eigenvalues of matrix L, and $(\chi_\ell)_{1\leq \ell\leq n}$ are the associated normalized and pairwise orthogonal eigenvectors. Note that the operator $P_\ell = \langle \chi_\ell, \cdot\rangle \chi_\ell$ is a projection onto the 1-dimensional subspace generated by the eigenvector χ_ℓ. For any function $f : \mathrm{sp}(L) \to \mathbb{R}$ defined on the spectrum $\mathrm{sp}(L)$ of matrix L, we define

$$f(L) = \sum_{\ell} f(\lambda_\ell) \langle \chi_\ell, \cdot\rangle \chi_\ell \quad \text{(functional calculus)}.$$

For any subset $I \subset \mathbb{R}^+$, $P_I(L)$ stands for the projection onto the subspace generated by $(\chi_\ell)_{\lambda_\ell\in I}$, that is

$$P_I(L) = \sum_{\ell:\ \lambda_\ell\in I} \langle \chi_\ell, \cdot\rangle \chi_\ell.$$

In particular, $L_I = LP_I(L) = P_I(L)L$ is essentially the projection of matrix L onto the range of projections $P_I(L)$. It is worth noting that $\mathrm{sp}(L_I) = \mathrm{sp}(L) \cap I$.

2.1.2 Discrete SGWT

The authors of [11] have introduced a new form of graph wavelets. Our LocLets are designed upon the definition of these graph wavelets. Let's recall the basic definitions related to the framework in [11]. Let $\varphi, \psi : \mathbb{R} \to \mathbb{R}$ be the scaling and kernel functions respectively. Fix scales $(s_j)_{1\leq j\leq J}$. The discrete SGWT is defined as follows:

$$\mathcal{W}f = (\varphi(L)f^T, \psi(s_1L)f^T, \ldots, \psi(s_JL)f^T)^T.$$

Such a formula can be decomposed on the following wavelet functions:

$$\psi_{j,m} = \psi(s_jL)\delta_m, \quad \text{and} \quad \phi_m = \varphi(L)\delta_m,$$

for $1 \leq m \leq n$.

Therefore, the discrete SGWT can be written in terms of the wavelet coefficients of the form $\mathcal{W}_f(j,m) = \langle \psi_{j,m}, f \rangle$ and coefficients $\langle \varphi_m, f \rangle$. The adjoint matrix $\mathcal{W}^*$ to $\mathcal{W}$ is:

$$\mathcal{W}^*(\eta_0^T, \eta_1^T, \ldots)^T = \varphi(L)\eta_0 + \sum_{j \geq 1} \psi(s_j L)\eta_j.$$

From $\mathcal{W}$ and $\mathcal{W}^*$ a reconstruction formula is obtained by applying $(\mathcal{W}^*\mathcal{W})^{-1}\mathcal{W}^*$ to the wavelet coefficients.

In order to evaluate the quality of the reconstruction (*e.g.* frame bounds in [11]), the following function G defined below plays a central role. We will also discuss the role of G in the LocLets framework.

$$G(\lambda) = \varphi(\lambda)^2 + \sum_{1 \leq j \leq J} \psi(s_j \lambda)^2 \quad \text{for} \quad \lambda \in \mathbb{R}^+.$$

In fact, [11, Theorem 5.6] asserts that $(\varphi_m, \psi_{j,m})_{j,m}$ is a frame $\mathfrak{F}$ whose bounds inequalities (1)

$$A = \min_{\lambda \in \mathrm{sp}(L)} G(\lambda) \quad \text{and} \quad B = \max_{\lambda \in \mathrm{sp}(L)} G(\lambda).$$

2.1.3 Fast Computation of SGWT

An important challenge raised by the definition of SGWT in [11] is the practical calculation of the transform without considering the whole spectral decomposition of L that is known to be poorly scalable. Fortunately, the only quantities required are of the form $g(L)f$, for some functions g defined on the spectrum $\mathrm{sp}(L)$, and a signal f. In [11], $g(L)f$ is approximated with $T_i(L)f$, where T_i is a Chebyshev polynomial of the first kind approximating the g function correctly on the compact set $\mathrm{sp}(L)$. The computation of the vectors $T_i(L)f$ only involves i matrix-vector multiplications possibly exploiting the sparsity of L. This efficient method was coined fast SGWT.

2.1.4 Reconstruction Formula Using Chebyshev Approximation

In [11], a reconstruction formula is given using pseudo-inverse $(\mathcal{W}^*\mathcal{W})^{-1}\mathcal{W}^*$ of $\mathcal{W}$. The reconstruction of a signal f from its coefficients c is given by $(\mathcal{W}^*\mathcal{W})^{-1}\mathcal{W}^*c$. To compute the reconstructed signal, the authors of [11] propose to solve the square matrix equation $\mathcal{W}^*\mathcal{W}f = \mathcal{W}^*c$ using fast gradient descent algorithms. As already noted in [12], an alternative approach involving Chebyshev approximation is also possible. However, in the latter paper, the authors restrict themselves to the case of Parseval's frames *i.e.* $A = B = 1$ in Equation (1).

First, note that $\mathcal{W}^*\mathcal{W}$ is a diagonal matrix in the basis $(\chi_\ell)_\ell$ of eigenvectors of L. More precisely, the following proposition holds.

Proposition 1. *Let* $G(\lambda) = \varphi(\lambda)^2 + \sum_j \psi(s_j\lambda)^2$. *Then*

$$\mathcal{W}^*\mathcal{W}f = G(L)f = \sum_\ell G(\lambda_\ell)\langle f, \chi_\ell \rangle \chi_\ell.$$

Proof. It is sufficient to verify $\mathcal{W}^*\mathcal{W}\chi_l = G(\lambda_l)\chi_l$ for all l. Let $1 \leq l \leq n$. We have

$$\mathcal{W}\chi_l = (\varphi(\lambda_l)\chi_l^T, \psi(s_1\lambda_l)\chi_l^T, \ldots)^T.$$

From the definition of $\mathcal{W}^*$, it follows $\mathcal{W}^*\mathcal{W}\chi_l = G(\lambda_l)\chi_l$. □

In other words, assuming $G(\lambda) > 0$ for all $\lambda \in \mathrm{sp}(L)$ (or equivalently $\mathcal{W}^*\mathcal{W}$ is invertible), we have $(\mathcal{W}^*\mathcal{W})^{-1} f = (G(L))^{-1} f$ for any signal f. In particular, the latter quantity can be approached using Chebyshev approximation, and matrix-vector multiplications of L with vectors. To go further, we also have a closed formula for the pseudo-inverse:

$$(\mathcal{W}^*\mathcal{W})^{-1}\mathcal{W}^*(\eta_0^T, \eta_1^T, \ldots)^T = \frac{\varphi}{G}(L)\eta_0 + \sum_{j\geq 1} \frac{\psi(s_j.)}{G}(L)\eta_j. \tag{2}$$

Hence, the reconstruction formula can be realized in practice with Chebyshev approximations of the terms $\frac{\varphi}{G}(L)g$, $\frac{\psi(s_j.)}{G}(L)g$ for some vectors g.

For Parseval's frame, $G \equiv 1$ so that $\mathcal{W}^*\mathcal{W} = \mathrm{Id}$. In this case, Equation (2) with $G \equiv 1$ is nothing but the one obtained for the reconstruction in [12].

2.2 Loclets: Localized Graph Wavelets and LocLets Transform

2.2.1 LocLets: Basic Definitions

In this section, the main object of this article is presented *a.k.a.* localized graph wavelets abbreviated as LocLets. First, let us fix some subset $I \subset \mathbb{R}^+$. Recall that L_I stands for the projected Laplacian $P_I(L)L$. The LocLet discrete transform at I is nothing but the discrete SGWT, applied to L_I in place of L. Its formula and that of its adjoint are given below:

$$\mathcal{W}^I f = (\varphi(L_I)f^T, \psi(s_1 L_I)f^T, \ldots)^T \quad \text{and} \quad (\mathcal{W}^I)^*(\eta_0^T, \eta_1^T, \ldots)^T = \varphi(L_I)\eta_0 + \sum_{j\geq 1} \psi(s_j L_I)\eta_j.$$

LocLets functions and coefficients are defined similarly to their SGWT analog, for $1 \leq j \leq J$, and $1 \leq m \leq n$:

$$\psi_{j,m,I} = \psi(s_j L_I)\delta_m, \quad \varphi_{m,I} = \varphi(L_I)\delta_m \quad \text{and} \quad \mathcal{W}_f^I(j,m) = \langle \psi_{j,m,I}, f\rangle.$$

Now assume that one has a partition of the smallest interval $I_L \subset \mathbb{R}^+$ containing the spectrum $\mathrm{sp}(L)$, in disjoint subsets $(I_k)_{1\leq k\leq K}$, that is $I_L = \sqcup_k I_k$. Then the LocLet transform of a signal f with respect to partition $(I_k)_{1\leq k\leq K}$ is given by

$$\mathcal{W}^{\mathrm{LocLet}} f = (\mathcal{W}^{I_k} f)_{1\leq k\leq K}.$$

There are several motivations for introducing LocLets objects:

- localization in the Fourier domain (*i.e.* on $\mathrm{sp}(L)$) is closely related to smoothness properties of the signal f;
- this transform is more suitable for signals whose support in the Fourier domain is sparse. For example, smooth signals on the graph have a representation in the Fourier domain mainly supported in an interval I containing only the smallest eigenvalues of L;
- in the case of large graphs but smooth signal, the dimension of the problem can be considerably reduced;
- many theoretical results (*e.g.* frame bounds, errors in Lanczos procedure,...) can be expressed in terms of functional over the spectrum $\mathrm{sp}(L)$, and can be improved by means of restrictions on sub-intervals $I \subset I_L$.

In spite of its apparent similarity to SGWT from [11], we show that LocLets significantly improve the accuracy over the usual SGWT (and other concurrent techniques) in some standard tasks handled by graph wavelets.

2.2.2 Fast Computation of Loclets

Optimization in fast computation of SGWT relies on exploiting matrix-vector multiplications as much as possible. We fit with this approach to propose a fast computation of LocLets. Our goal is to approach vectors of the form $g(L_I)f$ for some subset $I \subset I_L$ and some function g. Indeed, we observe that we have

$$g(L_I)f = g(L)f_I \text{ where } f_I = P_I(L)f.$$

We therefore propose to first calculate f_I, then to chain with the machinery of [11]. It is crucial to consider functional calculus (here $g(L)$ and $P_I(L)$) with respect to matrix L, and not L_I: that way, we can still benefit from sparsity properties of matrix L in matrix-vector calculations.

Our fast computation of $f_I = P_I(L)f$ also uses Chebyshev approximation of a relevant function: we just notice that we have $\mathbb{1}_I(L) = P_I(L)$, where $\mathbb{1}_I$ is the indicator function of the interval I, and thus approaching $\mathbb{1}_I$ on I_L by Chebyshev polynomials provides an efficient method for calculating f_I.

As will be seen in our experiments, this simple method already gives interesting results. However, it is well known that a non-regular function such as $\mathbb{1}_I$ is subject to certain boundary phenomena (*e.g.* Gibbs phenomenon) at discontinuities (here the ends of the interval I). Some works proposed efficient computations of $P_I(L)$ based on the approximation of more regular functions: see for instance [15, Section 3], where contours integral are used; another reference of interest is [6].

3 LocLets and Frequency Sparse Signal

A signal $f : V \to \mathbb{R}$ is *frequency sparse* if its Fourier transform is sparse. Smooth signals are substantial examples of frequency sparse signals. Intuitively, a signal is smooth if it contains only low frequencies. Mathematically, signal smoothness is related to its Fourier spectrum *via* the Dirichlet's formula (see for instance [20, p.15]) recalled below:

$$f^T L f = \sum_l \lambda_\ell |\langle f, \chi_\ell \rangle|^2 = \frac{1}{2} \sum_{i \sim j} w_{ij} |f_i - f_j|^2.$$

The rightmost expression of the smoothness modulus exhibits a weighted sum of squares of the gradient function of f making the connection with its oscillation properties. For this reason, the quantity $S_L(f) = f^T L f$ is called the smoothness modulus of signal f (with respect to Laplacian L). As a special case, we have $S_L(f) = 0$ if and only if f is constant. Similarly, the modulus $S_L(f)$ tends to be small as signal f is smooth on the graph, that is with few oscillations. LocLets are designed to exploit the connections expressed in the Dirichlet's formula between the smoothness of a signal and the importance of its spectral projections.

The following result states that the LocLet transform with respect to a single subset $I \subset \mathbb{R}_+$ of a signal f is nothing but the SGWT of $f_I = P_I(L)f$.

Proposition 2. *Let $I \subset \mathbb{R}_+$ be any subset. Then, $\mathcal{W}^I f = \mathcal{W} f_I$.*

Proof. From functional calculus properties, we have $\psi(s_j L_I) = P_I(L)\psi(s_j L)$, and so $\psi_{j,m,I} = P_I(L)\psi_{s_j,m}$. It follows

$$\langle f_I, \psi_{j,m} \rangle = \langle P_I(L)f, \psi_{j,m} \rangle = \langle f, P_I(L)\psi_{j,m} \rangle = \langle f, \psi_{j,m,I} \rangle.$$

Hence the equality $\mathcal{W}_{f_I}(j,m) = \mathcal{W}_f^I(j,m)$ holds for each coefficient, and then $\mathcal{W} f_I = \mathcal{W}_I f$. □

Consequently, consider $\sqcup_{1 \le k \le K} I_k = I_L$ an ordered partition of I_L meaning that I_k contains larger values than $I_{k'}$ for $k < k'$. Then, by definition of LocLet transform and Proposition 2, it follows

$$\mathcal{W}^{\text{LocLet}} f = (\mathcal{W} f_{I_k})_k.$$

Hence, LocLets simply perform SGWT component-wise on the decomposition $f = \sum_k f_{I_k}$ along different levels of smoothness ranging from the smoothest part f_{I_K} to most oscillating one f_{I_1}.

The following proposition provides an error bound on the error made when a LocLet transform is performed in place of the SGWT.

Proposition 3. *Let $I \subset I_L$ be an arbitrary subset. Let $f = f_I + f_{\overline{I}}$ be the orthogonal decomposition of signal f associated with subsets $I, \overline{I}$. Defining*

$$B_{\overline{I}} = \max_{\lambda \in \overline{I}} G_{|\overline{I}}(\lambda),$$

where $G_{|\overline{I}}$ is the restriction of G to subset $\overline{I}$, the following estimate holds

$$\sum_{j,n} \left| \mathcal{W}_f(j,n) - \mathcal{W}_{f_I}(j,n) \right|^2 \leq B_{\overline{I}} \, \|f_{\overline{I}}\|_2^2 \, \left| \{\ell, \lambda_\ell \in \mathrm{sp}(L) \cap \overline{I}\} \right|.$$

Proof. We have $\mathcal{W}_f(s,n) = \mathcal{W}_{f_I}(s,n) + \mathcal{W}_{f_{\overline{I}}}(s,n)$ for all s, n. Moreover,

$$\begin{aligned} \sum_{j,n} \left\| \mathcal{W}_{f_{\overline{I}}} \right\|_2^2 = \sum_{j,n} \left| \langle \psi_{s_j,n}, f_{\overline{I}} \rangle \right|^2 &\leq \sum_{j,n} \left\| \psi^{\overline{I}}_{s_j,n} \right\|_2^2 \, \left\| f_{\overline{I}} \right\|_2^2 \\ &= \left\| f_{\overline{I}} \right\|_2^2 \sum_{\ell: \ \lambda_\ell \in \mathrm{sp}(L) \cap \overline{I}} G_{|\overline{I}}(\lambda_\ell) \ \leq \left| \{\ell : \lambda_\ell \in \mathrm{sp}(L) \cap \overline{I}\} \right| \ B_{\overline{I}} \, \left\| f_{\overline{I}} \right\|_2^2. \end{aligned}$$

□

As an illustration, exploiting the sparsity in the frequency domain of a signal f, one might choose some subset $I \subset I_L$ in such a way that $\|f_{\overline{I}}\|_2 \leq \varepsilon$ for some small $\varepsilon > 0$. Then, Proposition 3 states that the ℓ^2-error made when LocLet transform is performed in place of SGWT is of order

$$\varepsilon^2 \ B_{\overline{I}} \left| \{\ell, \lambda_\ell \in \mathrm{sp}(L) \cap \overline{I}\} \right|.$$

Therefore, if f is sufficiently smooth so that $\varepsilon^2 \ll \left[B_{\overline{I}} \left| \{\ell, \lambda_\ell \in \mathrm{sp}(L) \cap \overline{I}\} \right| \right]^{-1}$, then compared to the standard SGWT transform $\mathcal{W}f$, LocLet transform $\mathcal{W}^I f$ provides a sufficiently accurate approximation while space-time complexity is reduced. These better performances are mainly due to the fact that the matrix L_I has a lower rank than the matrix L.

4 Denoising Frequency Sparse Signals with LocLets

Given an observed noisy signal $\tilde{f}$ of the form $\tilde{f} = f + Z$ where Z is a n-dimensional Gaussian vector distributed as $N(0, \sigma^2 \mathrm{Id})$, the aim of denoising is to provide an estimator of the *a priori* unknown signal f.

Simple observations highlighted in the following proposition demonstrate the benefits of using LocLets for noise reduction tasks, when the signal f is sparse in the Fourier domain.

Proposition 4. *Assume $f = f_I$ for some subset $I \subset I_L$. Then*

$$\mathbb{E}\left[\left\| f - \tilde{f}_I \right\|_2^2 \right] = \mathbb{E}\left[\left\| f - \tilde{f} \right\|_2^2 \right] - \sigma^2 |\overline{I} \cap \mathrm{sp}(L)|.$$

In particular, denoising of $\tilde{f}$ boils down to denoising of $\tilde{f}_I = f_I + Z_I$.

Proof. First, the following equalities hold:

$$f - \tilde{f} = f - \tilde{f}_I + \tilde{f}_I - \tilde{f} = (f - \tilde{f})_I + \tilde{f}_{\overline{I}}.$$

As $(f - \tilde{f})_I$ and $\tilde{f}_{\overline{I}}$ are orthogonal vectors, it follows that

$$\left\| f - \tilde{f} \right\|_2^2 = \left\| f - \tilde{f}_I \right\|_2^2 + \left\| \tilde{f}_{\overline{I}} \right\|_2^2.$$

It remains to notice that $\mathbb{E}(||\tilde{f}_{\overline{I}}||^2) = \sigma^2 |\overline{I} \cap \mathrm{sp}(L)|$. □

While Proposition 4 asserts a trivial denoising solution in the Fourier domain, *i.e.* simply destroying the projection $\tilde{f}_{\overline{I}} = Z_{\overline{I}}$, this approach is no longer as immediate whence based on the graph domain observations. Hence LocLets, with their possibilities to localize implicitly in the frequency domain, seem to be the suitable tools for frequency sparse signals.

In practice, it is needed to estimate the support of the Fourier transform in the frequency domain in an efficient way *i.e.* without computing the whole eigendecomposition of matrix L. Based on the χ^2-statistics and the fact the Fourier transform is orthogonal, Algorithm 1 is designed for this purpose.

Algorithm 1: Support approximation in the Fourier domain

Input: noisy signal $\tilde{f}$, a subdivision $I_1, I_2, \ldots, I_{K_0}$, estimated $|I_k \cap \mathrm{sp}L|$, $k = 1, \ldots, K_0$, threshold $\alpha \in (0, 1)$.

Output: $\tilde{f}_I = P_I(L)\tilde{f}$, where I is an approximation of the Fourier support of $\tilde{f}$.

for $k = 1, \ldots, K_0$ **do**

  Compute $\|f_{I_k}\|_2^2 = \|P_{I_k}(L)f\|_2^2$;

  Compute

$$p_k = \mathbb{P}(\Gamma_k > \|f_{I_k}\|_2^2) \quad \text{and} \quad \Gamma_k \sim \chi^2(|I_k \cap \mathrm{sp}(L)|);$$

end

Compute $\tilde{f}_I = \sum_{k:\ p_k \leq \alpha} P_{I_k} \tilde{f}$;

Heuristically, if I contains the support of the Fourier transform of f, on the complementary subset $\overline{I}$ we only observe pure white Gaussian noise so that $\|P_{\overline{I}}\tilde{f}\|_2^2 = \|\tilde{f}_{\overline{I}}\|_2^2$ is distributed as $\sigma^2\chi^2(d_I)$ with $d_I = |\overline{I} \cap \mathrm{sp}(L)|$. On the other hand, on I the square of the Euclidean norm of a non-centered Gaussian vector is observed. Consequently, the quantity $\mathbb{P}\left(\chi^2(d_I) > \sigma^{-2}\|P_I\tilde{f}\|_2^2\right)$ is typically very close to zero whereas $\mathbb{P}\left(\chi^2(n - d_I) > \sigma^{-2}\|P_{\overline{I}}\tilde{f}\|_2^2\right)$ remains away from 0. To put it in a nutshell, sliding a window along the spectrum of L, Algorithm 1 performs a series of χ^2-test.

However, for a subset $I \subset I_L$, the quantity d_I remains unknown. To overcome this problem, we rely on an estimate computed by the algorithm introduced in [7]. More precisely, the interval I_L is decomposed into intervals $I_1, \ldots, I_{K_0}$ of equal size and the counting algorithm is performed on each sub-interval. This counting process requires K_0 Chebyshev approximations. If $K_0 \ll n$, the complexity of our method remains acceptable. Furthermore, let us stress that the numbers of the eigenvalues in each I_k depends only on the Laplacian and should be estimated only once, independently of signals on the graph.

The second step gives an estimate of the original signal using a thresholding procedure on each element $\tilde{f}_I$ and $\tilde{f}_{\overline{I}}$.

On the one hand, the methodology developed in [10] is prohibitive in terms of time and space complexity as soon as the underlying graphs become moderately large. On the other hand, the fast SWGT remains an approximating procedure. If a signal happens to be very

Algorithm 2: LocLets thresholding estimation procedure

Input: $\tilde{f}$, α, $(I_k)_{k=1,\dots,K_0}$, estimated $|I_k \cap \mathrm{sp}(L)|$, thresholds t_1, t_2
Output: estimator $\hat{f}$ of signal f

1 Apply Algorithm 1 with $\tilde{f}$, α, $(I_k)_{k=1,\dots K_0}$, estimated $|I_k \cap \mathrm{sp}(L)|$; it outputs $\tilde{f}_I$ and $\tilde{f}_{\overline{I}}$;
2 Apply soft-thresholding with threshold t_1 to $\mathcal{W}^I \tilde{f}$ and t_2 to $\mathcal{W}^{\overline{I}} \tilde{f}$;
3 Apply the inverse LocLet transform to the soft-thresholded coefficients to obtain $\hat{f}_I, \hat{f}_{\tilde{I}}$;
4 Compute the estimator $\hat{f} = \hat{f}_I + \hat{f}_{\overline{I}}$;

sparse in frequency, then an even more optimal strategy is possible: first, the support I in the frequency domain is approximated with the help of Algorithm 1; then, the procedure of [10] is applied to $P_I f$ (the low rank part) and LocLets on $P_{\overline{I}} f$. This idea is made precisely in Algorithm 3. As shown in our experiments in Section 5, it turns out that a fair gain is obtained applying this procedure.

Algorithm 3: LocLets support approximation, and low-rank Parseval Frame thresholding procedure

Input: $\tilde{f}$, α, $(I_k)_{k=1,\dots,K_0}$, estimated $|I_k \cap \mathrm{sp}(L)|$, thresholds t_1, t_2
Output: estimator $\hat{f}$ of signal f

1 Apply Algorithm 1 with $\tilde{f}$, α, $(I_k)_{k=1,\dots K_0}$, estimated $|I_k \cap \mathrm{sp}(L)|$; it outputs $\tilde{f}_I$ and $\tilde{f}_{\overline{I}}$;
2 Compute Parseval Frame for L_I;
3 Apply Parseval Frame thresholding with threshold t_1 to $\tilde{f}_I$; it outputs $\hat{f}_I$;
4 Apply soft-thresholding wtih threshold t_2 to $\mathcal{W}^{\overline{I}} \tilde{f}$;
5 Apply the inverse LocLet transform to the soft-thresholded coefficients to obtain $\hat{f}_{\tilde{I}}$;
6 Compute the estimator $\hat{f} = \hat{f}_I + \hat{f}_{\overline{I}}$;

5 Experimental Validation

5.1 General Settings

In this section, we compare the numerical performance of LocLets with that of state-of-the-art classical wavelet frame introduced in [10]. Let us recall that the latter frame can be considered as a special case of SGWT as defined in [11].

However, the denoising method in [10] requires the computation of the whole spectral decomposition of the Laplacian which does not scale to large graphs. We stress here that we provide a fair comparison with [10], only in terms of denoising performance, and with no computational considerations. Moreover, we choose for LocLets to use the most naive thresholding procedure by considering a global and scale independent threshold scale. The discussion of possible improvements is postponed in Section 6.

For all the experiments below, the SGWT and LocLets are built upon the scale and kernel functions giving rise to the Parseval frame of [10]. More precisely, set respectively $\varphi = \sqrt{\zeta_0}$ and $\psi = \sqrt{\zeta_1}$ for the scale and kernel functions where

$$\zeta_0(x) = \omega(x) \quad \text{and} \quad \zeta_1(x) = \omega(b^{-1}x) - \omega(x),$$

where we choose $b = 2$ and ω is piecewise linear, vanishes on $[1, \infty)$ and is constant equal to one on $(-\infty, b^{-1}]$. The scales are of the form $s_j = b^{-j+1}$ for $j = 1, \dots, J$ where J is chosen similarly to [10].

In the sequel, 'PF' stands for Parseval Frame and refers to the estimator of [10]; the estimators implemented by Algorithm 2 and Algorithm 3 are referred to as 'LLet' and 'LLet+PF' respectively. Below, the latter methodology is shown to outperform all the others for very frequency sparse signals.

It is worth recalling that 'LLet+PF' benefits from the dimension reduction property of LocLets. More precisely, whereas the whole eigendecomposition of L is required to apply 'PF', for Parseval frame denoising in the context of 'LLet+PF', only a low-rank spectral decomposition is needed, namely the decomposition of L_I.

For all our experiments, we set $\alpha = 0.001$ for Algorithm 1. Our experiments are carried out on a grid of the parameters (f, w, t_1, t_2) with f the signal at stake, w the size of the sliding window in Algorithm 1 (so approximately a number of intervals equal to $w \times K_0 \sim |I_L|$), and t_1, t_2 the thresholds in Algorithms 2, 3. First, we compute the best SNR (Signal-to-Noise Ratio) result r_D over a large grid of values (t_1, t_2), and for each denoising method $D \in \{$'PF','LLet','LLet+PF'$\}$. Then, we calculate two metrics: the maximum M_D and average value μ_D of the values r_D over 10 random perturbations of the signal f. The same SNR evaluation is performed on the raw noisy signal without any denoising, and is referred in our results by 'SNR_{in}'. We recall that a good quality in denoising is reflected by a large value of the SNR metric.

We sample randomly signals whose support are sparse in the Fourier domain. We will use the notation f_{i-j} for normalized signals supported on a sub-interval of I_L containing exactly the eigenvalues $\lambda_i, \lambda_{i+1}, \ldots, \lambda_j$. As an example, f_n is a constant signal while f_{1-2} is a highly non-smooth signal supported on the eigenspaces of large eigenvalues λ_1, λ_2. For experiments, the signals were calculated from the knowledge of $\mathrm{sp}(L)$, and relevant projections of random functions on the graph.

5.2 Low-Dimensional Experiment

We first provide a toy experiment on a small randomly selected graph. We simulate a graph of size $n = 10$ by randomly sampling a Laplacian matrix $L \in \mathbb{R}^{n \times n}$, with a non-zero coefficient ratio almost equal to 0.3 (to simulate a certain sparsity in the graph connections). For the graph in our experiments, we have $\lambda_1 = 32.0$, and $I_L = [0.0, 32.0]$.

Then, we randomly sample the signals $(f_{i-(i+1)})_{1 \leq i \leq 9}$ on the graph with variable and sparse supports in the Fourier domain. The noise reduction task is parametrized by the white noise standard deviation $\sigma = 0.05$. For each function $f_{i-(i+1)}$, ten noisy realizations are simulated. The SNR performances for denoising functions $f_{i-(i+1)}$ are computed over these ten samples, and the results for functions $f_{1-2}, f_{5-6}, f_{9-10}$ are shown in Table 1. Since $w \times K_0 \sim |I_L|$, a sliding window $w = 1$ will produce $K_0 = 32$ intervals in Algorithm 1, while a sliding window $w = 10$ only 4 intervals.

The first obvious observation is that 'LLet+PF' performs better than its concurrent in almost all situations. The gain is very significant since there is sometimes a gap of 4dB in μ_D-metric between 'LLet+PF' and its closest concurrent, and up to 8dB in M_D-metric.

The low-dimensional regime has some specificities, which will not be observed for the case $n = 1000$. First, the results show a large variance over random realizations. This is easily seen from a comparison of the best results in Table 1 (left) obtained for SNR values close to 33 to the best in Table 1 (right) attaining a 24 SNR value. Secondly, we would expect to have worse results for larger sliding windows, as the estimation of the Fourier support of f is less accurate. But this fact is not observed, even for the case $K_0 = 4$ ($w = 10$).

5.3 Experiment on the Swiss Roll Graph

Below, a similar comparison is made for signals on a large Swiss roll graph. Following [11] for instance, this graph is generated by considering $(u_i, v_i)_{1 \leq i \leq n}$ uniformly random points in the

Table 1: SNR performance for $n = 10$.

id	w	SNR_{in}	M_{PF}	$\mathrm{M}_{\mathrm{LLet}}$	$\mathrm{M}_{\mathrm{LLet+PF}}$	μ_{PF}	μ_{LLet}	$\mu_{\mathrm{LLet+PF}}$
1-2	1	15.78	24.29	25.32	33.04	18.96	20.11	24.00
5-6	1	17.25	24.48	23.48	24.61	19.10	19.21	19.77
9-10	1	16.93	31.72	18.22	33.00	21.73	15.30	24.73
1-2	2	16.92	21.72	25.52	29.70	19.11	22.07	25.54
5-6	2	17.54	23.92	23.06	22.68	16.42	18.01	18.55
9-10	2	17.54	24.19	18.59	30.06	20.45	15.23	24.65
1-2	5	13.77	24.64	25.39	27.46	18.95	21.70	22.06
5-6	5	18.45	25.75	25.06	24.85	17.52	19.64	20.33
9-10	5	12.50	23.32	22.23	24.85	18.82	16.05	20.61
1-2	10	14.95	21.77	28.39	30.80	18.44	21.24	22.93
5-6	10	16.73	23.01	21.25	26.85	17.93	17.81	19.37
9-10	10	15.42	26.12	16.91	28.05	21.13	14.40	23.35

unit square. The size of the graph is $n = 1000$ in our experiments. Then, these random points are mapped to the Swiss roll via the parametrisation.

$$\begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \begin{pmatrix} \rho(u)\cos(\rho(u)) \\ \pi^2(\beta^2-\alpha^2)v/2 \\ \rho(u)\sin(\rho(u)) \end{pmatrix}$$

where $\rho(u) = \pi\sqrt{(\beta^2-\alpha^2)u+\alpha^2}$, $\alpha = 1$ and $\beta = 4$. Given this set of random points (p_i), a weighted graph is built by affecting a Gaussian weight $w_{i,j} = \exp(-\|p_i - p_j\|_2^2/2s^2)$ where $s > 0$ is some parameter ($s = 2$ in the experiments) between nodes p_i and p_j. Finally, this graph has been sparsify by thresholding the weights lower than 10^{-10}.

Table 2: SNR performance for $n = 1000$.

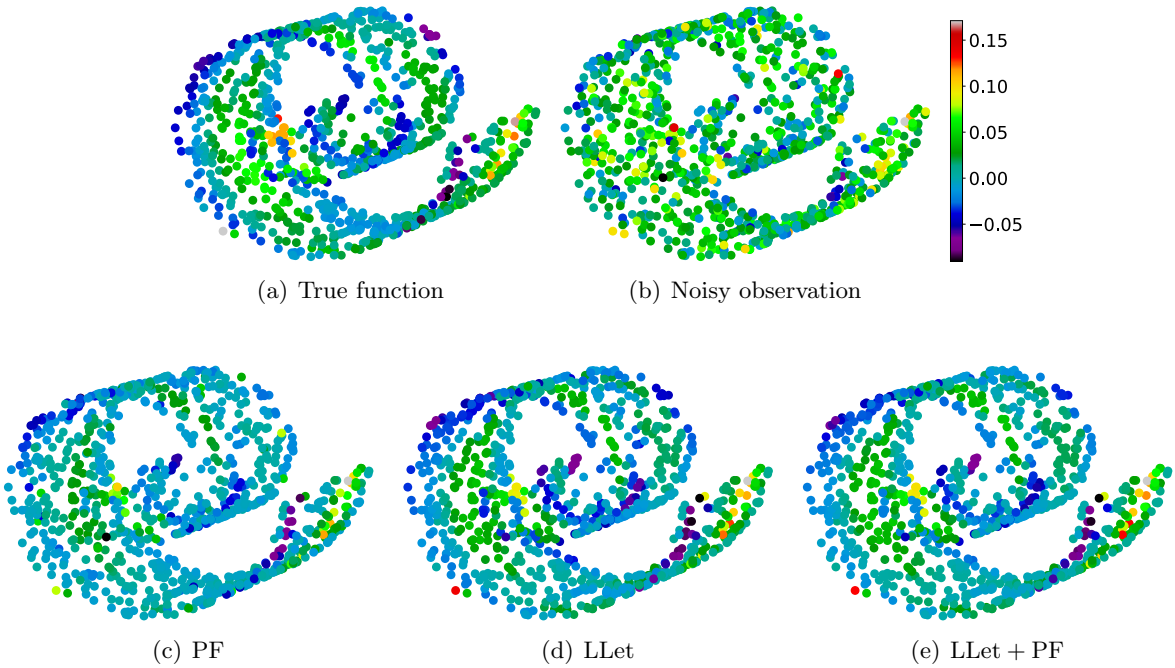
id	w	SNR_{in}	M_{PF}	$\mathrm{M}_{\mathrm{LLet}}$	$\mathrm{M}_{\mathrm{LLet+PF}}$	μ_{PF}	μ_{LLet}	$\mu_{\mathrm{LLet+PF}}$
951-1000	1	15.84	22.95	25.76	27.83	22.16	24.87	26.34
501-550	1	15.89	17.74	20.35	20.48	17.41	19.93	20.09
1-50	1	15.82	20.87	25.26	25.80	20.41	23.88	24.64
951-1000	2	15.93	22.67	24.32	26.16	21.98	23.53	24.96
501-550	2	15.99	17.60	20.23	20.52	17.38	19.78	19.90
1-50	2	15.70	20.99	23.12	24.20	20.48	22.72	23.43
951-1000	3	15.90	22.84	22.10	24.95	22.12	21.50	24.10
501-550	3	16.14	17.79	19.19	19.69	17.44	18.73	19.32
1-50	3	15.78	20.95	23.34	24.18	20.572	22.62	23.37
951-1000	4	16.15	22.96	20.78	24.56	22.28	19.94	23.60
501-550	4	16.16	17.52	18.67	18.75	17.30	18.34	18.44
1-50	4	15.98	21.14	8.74	8.91	20.68	8.67	8.84

We randomly sample the signals $(f_{i-(i+49)})_{1\leq i\leq 999}$. The noise reduction task is parametrized by the white noise standard deviation $\sigma = 0.005$. For each function $f_{i-(i+49)}$, ten noisy realizations are simulated. The results for functions $f_{1-50}, f_{501-550}, f_{951-1000}$ are presented in Table 2. Since the largest eigenvalue of L is $\lambda_1 = 8.8$, we obtain $I_L = [0.0, 8.8]$. The sliding window w

ranges from 1 to 4, producing a number of intervals K_0 ranging from 9 to 3 in Algorithm 1. For a single experiment, with a sample size $n = 1000$, we display the noisy observations and the unknown function f in Figure 1(a)–(b). A typical example of estimation for each of the three compared methods is also given in Figure 1(c)–(e).

For reasonably small values of the sliding window w, 'LLet+PF' performs better than its concurrent in almost all situations. By contrast with the case $n = 10$, 'LLet' provides better performances than 'PF' more systematically. Indeed, signals are much more sparse in the Fourier domain for our experiment with $n = 1000$, which benefits to methods using LocLets.

As expected and observed for function f_{1-50} with $w = 4$, a size value too large w for intervals in Algorithm 1 decreases performances of LocLets methods. If the approximation of the Fourier support of the signal is too coarse, our LocLets lose their interesting frequency localization properties. Fortunately, this drop in performances occurs only for large values of w, here corresponding to values of $K_0 \leq 3$. The efficiency of Chebyshev approximations allow the use of larger values K_0.



(a) True function (b) Noisy observation

(c) PF (d) LLet (e) LLet + PF

Figure 1: Typical reconstructions from a single simulation with $n = 1000$.

6 Discussion and further improvements

In this paper we proposed a new wavelet transform called LocLets, which is a variation of SGWT designed for large graph signals sparse in the frequency domain. Basic theoretical properties of LocLets were provided and experiments were conducted on several denoising tasks applied to a large family of sparse signals. Empirical results have demonstrated that LocLets-based methods show significant improvements over the best prior denoising techniques. By contrast with their closest method in terms of denoising performances, which requires the computation of the whole spectral decomposition of the graph Laplacian, LocLets do not require any eigendecomposition, and scale much more easily with the size of the graph by means of standard Chebyshev approximations techniques.

LocLets transform suggests several interesting directions for future work. First, dimensionality reduction in the context of SGWT was already considered in some previous works, usually with connections to the Lanczos procedure [17]. As shown in the current paper with

method 'LLet+PF', one can benefit from both state-of-the-art denoising performances of Parseval frames and sparsity optimizations in the frequency domain of LocLets. We plan to investigate further the relationship between LocLets and low-rank Laplacian matrices.

In our experiments, we select the best denoising performance for each method with a grid search over thresholding values. Threshold selection was studied since a long time in the signal processing community, and several adaptative thresholding methods were proposed in the case of standard wavelets [8, 9]. LocLets introduced in this paper would certainly benefit from data-driven thresholds selection for practical use.

References

[1] Belkin, M., Niyogi, P.: Towards a theoretical foundation for laplacian-based manifold methods. Journal of Computer and System Sciences **74**(8), 1289–1308 (2008)

[2] Casazza, P.G., Kutyniok, G., Philipp, F.: Introduction to finite frame theory. In: Finite frames, pp. 1–53. Springer (2013)

[3] Christensen, O.: Frames and bases. Applied and Numerical Harmonic Analysis, Birkhäuser Boston, Inc., Boston, MA (2008). https://doi.org/10.1007/978-0-8176-4678-3, https://doi.org/10.1007/978-0-8176-4678-3, an introductory course

[4] Chung, F.R., Graham, F.C.: Spectral graph theory. No. 92, American Mathematical Soc. (1997)

[5] Crovella, M., Kolaczyk, E.: Graph wavelets for spatial traffic analysis. In: IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428). vol. 3, pp. 1848–1857. IEEE (2003)

[6] Di Napoli, E., Polizzi, E., Saad, Y.: Efficient estimation of eigenvalue counts in an interval. Numerical Linear Algebra with Applications **23**(4), 674–692 (2016)

[7] Di Napoli, E., Polizzi, E., Saad, Y.: Efficient estimation of eigenvalue counts in an interval. Numer. Linear Algebra Appl. **23**(4), 674–692 (2016). https://doi.org/10.1002/nla.2048, https://doi.org/10.1002/nla.2048

[8] Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. Journal of the american statistical association **90**(432), 1200–1224 (1995)

[9] Donoho, D.L., Johnstone, J.M.: Ideal spatial adaptation by wavelet shrinkage. biometrika **81**(3), 425–455 (1994)

[10] Göbel, F., Blanchard, G., von Luxburg, U.: Construction of tight frames on graphs and application to denoising, pp. 503–522. Springer (2018)

[11] Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis **30**(2), 129–150 (2011)

[12] Leonardi, N., Van De Ville, D.: Tight wavelet frames on multislice graphs. IEEE Transactions on Signal Processing **61**(13), 3357–3367 (2013)

[13] Mallat, S.: A wavelet tour of signal processing. Elsevier/Academic Press, Amsterdam, third edn. (2009), the sparse way, With contributions from Gabriel Peyré

[14] Ortega, A., Frossard, P., Kovačević, J., Moura, J.M., Vandergheynst, P.: Graph signal processing: Overview, challenges, and applications. Proceedings of the IEEE **106**(5), 808–828 (2018)

[15] Peter Tang, P.T., Polizzi, E.: Feast as a subspace iteration eigensolver accelerated by approximate spectral projection. SIAM Journal on Matrix Analysis and Applications **35**(2), 354–390 (2014)

[16] Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine **30**(3), 83–98 (2013)

[17] Susnjara, A., Perraudin, N., Kressner, D., Vandergheynst, P.: Accelerated filtering on graphs using lanczos method. arXiv preprint arXiv:1509.04537 (2015)

[18] Tanaka, Y., Sakiyama, A.: m-channel oversampled graph filter banks. IEEE Transactions on Signal Processing **62**(14), 3578–3590 (2014)

[19] Tremblay, N.: Networks and signal : signal processing tools for network analysis. Theses, Ecole normale supérieure de lyon - ENS LYON (Oct 2014), https://tel.archives-ouvertes.fr/tel-01078956

[20] Woess, W.: Random walks on infinite graphs and groups, vol. 138. Cambridge university press (2000)