Two-Channel Critically-Sampled Graph Wavelets With Spectral Domain Sampling

Akie Sakiyama, Student Member, IEEE, Kana Watanabe, Yuichi Tanaka, Member, IEEE, and Antonio Ortega, Fellow, IEEE

Abstract—We propose two-channel critically-sampled wavelet transforms for signals on undirected graphs that utilize spectral domain sampling. Unlike conventional approaches based on vertex domain sampling, our transforms have the following desirable properties: 1) perfect reconstruction regardless of the characteristics of the underlying graphs and graph variation operators and 2) a symmetric structure; i.e., both analysis and synthesis filter banks are built using similar building blocks. The relationship between the proposed wavelets and those using vertex domain sampling are also described. The effectiveness of our approach is evaluated by comparing its performance in nonlinear approximation and denoising with that of conventional graph wavelets and filter banks.

Index Terms—Graph signal processing, spectral graph wavelet, spectral graph filter bank, spectral domain sampling

I. INTRODUCTION

A. Motivation

Graph signal processing focuses on graph signals, discrete signals defined on the vertices of a graph [1]–[8]. Graph signals can represent a broad range of irregularly structured data, such as signals on brain, sensor, social, and traffic networks, point cloud attributes, and images/videos. Developing sparse representations for these signals by using appropriate bases or frames is important, because these signals are often high-dimensional. Promising applications for such sparse representations of graph signals include feature extraction [4]–[6], denoising [7]–[10], compression [10]–[13], and others in many different areas [14]–[19].

As is the case with classical signal processing, multiscale transforms or dictionaries are important tools for achieving sparse representations of graph signals. Sampling strategies are crucial for controlling the level of redundancy in the graph signal in a multiscale signal representation. Undecimated transforms [20]–[22] require a significant storage overhead for the transformed coefficients. Other approaches, which can achieve different trade-offs in terms of redundancy, performance, computation cost and storage, are oversampled [21], [23]–[25], critically sampled (CS) [21], [26]–[33] and undersampled transforms [34].

For time domain signals, downsampling by a factor of two followed by upsampling by a factor of two corresponds to replacing every other sample by zero. In the frequency domain, the resulting signal has two components, the original frequency content of the signal and an added aliasing term (a modulated version of the original spectrum) [35]–[37]. In contrast, in the case of graph signals, downsampling/upsampling in the vertex domain (i.e., replacing some of the values on the graph vertices by zero) does not preserve the shape of the original graph signal spectrum [38], with the sole exception of bipartite graphs.

To date, most graph transforms that make use of sampling apply it in the vertex domain. These approaches are limited to specific graph types and variation operators. For example, some designs require bipartite graphs and symmetric normalized graph Laplacians are needed for perfect reconstruction [21], [26], [27], [29]. Some graph transforms are not restricted to specific graph types or variation operators but have other limitations. For example, in [30], [32], [39] the transform operates on a graph different from the original one, while in [31], [40] the original graph is used, but even though the decomposition can use efficient polynomial graph filters, the reconstruction requires significantly more complex global interpolation. The choices of sampling leading to perfect reconstruction are not unique in [31], [40], and for any given sampling choice the interpolator is obtained via matrix inversion, which can lead to numerical issues.

In this paper, we propose novel CS graph wavelet transforms (GWTs), using the recently introduced spectral domain sampling [38], that overcome the problems described above. These GWTs have the following properties:

- Perfect reconstruction is guaranteed for any graph and for any variation operator as long as the operator is diagonalizable and has real eigenvalues.
- The (frequency domain) sampling that leads to perfect reconstruction is unique, while the analysis and synthesis operators have the same complexity and a matrix inversion is not required to compute the reconstruction operator.

Moreover, we show that the GWTs obtained in the vertex domain and those obtained using spectral domain sampling are identical in some special cases. We also assess their performance through experiments on denoising and nonlinear approximation. This paper significantly extends our preliminary study [41] by adding rigorous proofs to the theoretical results and providing much more comprehensive experiments.
The rest of the paper is organized as follows. We review related work in Section II. Sampling methods in the vertex and spectral domains are introduced in Section III Section IV reviews the conventional CS GWTs. The proposed CS GWTs are presented in Section V along with the octave-band structure and polyphase representation. The relationship between the vertex and spectral domain sampling approaches is studied in Section VI. Section VII presents a few potential applications of the proposed CS GWTs, together with comparisons with the conventional methods. Finally, Section VIII is the conclusion.

B. Notation

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of edges $\mathcal{E}$ and vertices $\mathcal{V}$, where the number of vertices is $N = |\mathcal{V}|$. We consider undirected graphs without self-loops and nonnegative edge weights. A graph signal is a function $f : \mathcal{V} \rightarrow \mathbb{R}$, and it can be represented in vector form $\mathbf{f} \in \mathbb{R}^N$, whose $n$th sample $f[n]$ is regarded as a signal value on the $n$th vertex of the graph.

A $\mathbb{R}^{N \times N}$ is an adjacency matrix of the graph whose $(i, j)$th-element $a_{ij}$ represents the weight of the edge between the $i$th and $j$th vertices. $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal degree matrix whose elements are defined as $d_{ii} = \sum_j a_{ij}$. The combinatorial and symmetric normalized graph Laplacians are defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$ and $\mathcal{L} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$, respectively. Since a graph Laplacian is a real symmetric matrix, the eigendecomposition of $\mathbf{L}$ (or $\mathcal{L}$) can always be represented as $\mathbf{L} = \mathbf{U} \Lambda \mathbf{U}^{-1}$, where $\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}]$ is an eigenvector matrix, $\Lambda = \text{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{N-1})$ is an eigenvalue matrix having eigenvalues $\lambda_i$ $(i = 0, 1, \ldots, N-1)$ of $\mathbf{L}$ as diagonal elements, and $^{-1}$ represents the transpose of a matrix.

For a symmetric normalized graph Laplacian, its eigenvalues are bounded in $\lambda_i \in [0, 2]$. In addition, the maximum eigenvalue becomes $\lambda_{\text{max}} = 2$ and the eigenvalues are distributed symmetrically with respect to $\lambda = 1$ only for the bipartite case.

The graph Fourier transform (GFT) is defined as

$$\tilde{\mathbf{f}}[i] = \langle \mathbf{u}_i, \mathbf{f} \rangle = \sum_{n=0}^{N-1} u_i[n] f[n], \quad (1)$$

while the other definitions of the GFT, such as those in [ ], can be used as long as the GFT matrix is nonsingular.

II. RELATED WORK

Several CS GWTs using vertex domain sampling have been proposed for signals on bipartite graphs. They can be used on non-bipartite graphs by dividing the original graph into several bipartite graphs and then using a “multidimensional” decomposition. Filter design methods for this class of CS GWT include: graphQMF [26], which utilizes quadrature mirror filters; graphBior [27] a biorthogonal and polynomial filter solution with spectral factorization; a frequency conversion method [21] that transforms time domain filters into graph spectral filters; near-orthogonal polynomial filter design methods proposed in [28], [29]. Oversampled graph filter banks were introduced in [24], [25] as an extension of CS GWTs for bipartite graphs.

The above methods are for designing filters in the graph frequency domain. There are also CS graph filter banks whose filters are designed in the vertex domain. For example, a lifting-based transform [39] divides the original graph into even and odd-indexed vertices and performs vertex domain filtering. The subgraph-based biorthogonal filter bank [30] decomposes the original graph into several partitions. Wavelets on a balanced tree [42] provide CS perfect reconstruction transforms using vertex domain filtering. The spline-based graph wavelet is a CS perfect reconstruction for cyclic graphs [32]. There is a CS graph filter bank for a specific class of graphs, called $\Omega$-structures [33]. However, all of these methods require simplifying the graph, i.e., eliminating some of the edges in the original graph, in order to ensure critical sampling and invertibility.

CS graph filter banks can also be designed with careful vertex domain sampling. An $M$-channel CS graph filter bank [31] was designed that selects sampled vertices for each subband in order to satisfy the uniqueness set condition. In the context of sampling theory of graph signals, band limiting the input graph signal followed by vertex domain sampling has been proposed as a graph filter bank [40]. However, such approaches have several limitations. First, they have to select an appropriate sampling set for perfect reconstruction. In other words, arbitrarily selected sampling sets do not generally lead to a perfect reconstruction transform. Second, the sampling set is not unique; different sampling sets significantly affect the overall performance of the graph transforms in applications. Here, an efficient vertex selection was proposed in [43], where the filter bank system has a symmetric structure. However, the perfect reconstruction condition is only guaranteed for bipartite graphs [43]. Third, many approaches are perfect reconstruction only if ideal filters are used in the analysis transform. That means there is no flexibility in the design of the filter. However, non-ideal filters are sometimes preferred when the eigenvalue distribution of the variation operator is irregular (described in Section VII-D). Fourth, they often need to calculate the reconstruction operator for the synthesis side [31], [40]. This leads to a matrix inversion with a high computational cost.

The performance of CS GWTs with vertex domain sampling varies according to the graph reduction method used. Graph coloring [26], [27], [44], [45], Kron reduction [11], [46], maximum spanning trees [47], weighted max-cut [48], and graph coarsening using algebraic distance [49] are examples of the various graph reduction methods.

The properties of the existing and proposed GWTs are summarized in Table I. It should be emphasized that all of the existing approaches have limitations on their design, e.g., eligible graphs/variation operators, sampling set guaranteeing perfect reconstruction, or filter design. Our approach overcomes the limitations by employing a novel sampling in the graph frequency domain, and it is the only approach that has
all of the following features: i) spectral domain filtering, ii) orthogonality, iii) perfect reconstruction, and iv) applicability to any graph.

III. SAMPLING OF GRAPH SIGNALS

This section describes the sampling methods of graph signals in the vertex and graph frequency domains.

A. Sampling in Vertex Domain

The conventional and widely used method for sampling graph signals in the vertex domain is defined as follows:

**Definition 1. (Downsampling of graph signals in vertex domain).** Let \( G_0 = (V_0, E_0) \) and \( G_1 = (V_1, E_1) \) be the original and reduced-size graphs, respectively, where every vertex in \( G_1 \) has a one-to-one correspondence to one of the vertices in \( G_0 \). The original signal is \( f \in \mathbb{R}^{V_0} \). In the vertex domain, downsampling of \( f \) to \( \tilde{f}_d \in \mathbb{R}^{V_1} \) is defined as follows.

\[
(GD1) \quad \text{Keeping samples in } V_1. \quad \tilde{f}_d[n] = f[n'] \quad \text{if } v_{0,n'} \in V_0 \text{ corresponds to } v_{1,n} \in V_1. \quad (2)
\]

This is illustrated in Fig. [1](b).

**Definition 2. (Upsampling in vertex domain).** \( G_0 \) and \( G_1 \) are the same as in Definition [4]. The original signal at this time is \( f \in \mathbb{R}^{V_1} \), and its sample is associated with \( G_1 \). Upsampling in the vertex domain, i.e., mapping from \( f \) to \( \tilde{f}_u \in \mathbb{R}^{V_0} \), is defined as follows.

\[
(GU1) \quad \text{Placing samples on } V_1 \text{ into the corresponding vertices in } G_0. \quad f_u[n] = \begin{cases} f[n'] & \text{if } v_{u,n'} \in V_1 \text{ corresponds to } v_{n} \in V_0 \\ 0 & \text{otherwise.} \end{cases} \quad (3)
\]

This is illustrated in Fig. [2](b).

B. Sampling in Graph Frequency Domain

Next, we describe sampling of graph signals defined in the graph frequency domain [38]. Note that there are a number of slightly different definitions in the literature. Please refer to [38] for other definitions besides the ones used here.

**Definition 3. (Downsampling of graph signals in graph frequency domain).** Let \( L_0 \in \mathbb{R}^{N \times N} \) and \( L_1 \in \mathbb{R}^{N/2 \times N/2} \) respectively be graph Laplacians for the original and reduced-size graphs, respectively, and assume that their eigendecompositions are given as \( L_0 = U_0 \Lambda_0 U_0^\top \) and \( L_1 = U_1 \Lambda_1 U_1^\top \), where \( \Lambda_1 = \text{diag}(\lambda_{1,\max}, \lambda_{2,\max}, \ldots, \lambda_{N,\max}) \). The downsampling graph signal in the graph frequency domain \( \tilde{f}_d \in \mathbb{R}^{N/2} \) is defined as follows.

\[
(GD2) \quad \text{Fold & add, Warping spectrum.} \quad \tilde{f}[i] = \tilde{f}_d[i] + \tilde{f}[N - i - 1], \quad (4)
\]

where \( i = 0, \ldots, N/2 - 1 \). The above equation is easily represented in matrix form:

\[
f_d = U_1 \tilde{S}_d U_0^\top f, \quad (5)
\]

where \( \tilde{S}_d = [I_{N/2} J_{N/2}] \), in which \( I \) and \( J \) are the identity and counter-identity matrices, respectively. This downsampling strategy is illustrated in Fig. [1](c).

**Definition 4. (Upsampling of graph signals in graph frequency domain).** Let \( L_0 \in \mathbb{R}^{N \times N} \) and \( L_2 \in \mathbb{R}^{2N \times 2N} \) be the graph Laplacians for the original and increased-size graphs, respectively. The upsampling graph signal in the graph frequency domain \( \tilde{f}_u \in \mathbb{R}^{2N} \) is defined as follows.

\[
(GU2) \quad \text{Repeating the original and flipped spectra alternatively.} \quad \tilde{f}_u[i] = \begin{cases} \tilde{f}[i] & i = 0, \ldots, N - 1 \\ \tilde{f}[2N - i - 1] & i = N, \ldots, 2N - 1. \end{cases} \quad (6)
\]

The above equation is easily represented in matrix form:

\[
f_u = U_2 \tilde{S}_u U_0^\top f, \quad (7)
\]

where \( \tilde{S}_u = [I_N J_N]^\top \) and \( U_2 \) is the eigenvector matrix of \( L_2 \).

This upsampling strategy is illustrated in Fig. [2](c).

It is worth noting that, in general, the graph frequency components are not localized in the vertex domain after the spectral domain sampling; the main and aliasing components in the graph frequency domain are mixed in the vertex domain.

IV. TWO-CHANNEL CS GWTs WITH VERTEX DOMAIN SAMPLING

A. Framework and Perfect Reconstruction Condition

The most popular CS GWTs are designed for bipartite graphs [26], [27]. They are perfect reconstruction if the
TABLE I
LIST OF CRITICALLY-SAMPLED GRAPH WAVELETS AND FILTER BANKS. PROPERTIES ARE DESCRIBED AT THE BOTTOM OF THE TABLE.

| GWTs / Properties                      | Analysis\(^1\) | Synthesis\(^2\) | Filter\(^3\) | Graphs\(^4\) | VO\(^5\) | Orth.\(^6\) | Comp.\(^7\) | PR\(^8\) |
|----------------------------------------|---------------|----------------|-------------|-------------|---------|------------|-----------|---------|
| GraphQMF [26]                          | Filt. → VS    | VS → Filt.     | S           | Bipartite   | SNL     | O          | ✓         | ✓       |
| GraphBior [27]                         | Filt. → VS    | VS → Filt.     | S           | Bipartite   | SNL     | O, B       | ✓         | ✓       |
| Frequency conversion [21]              | Filt. → VS    | VS → Filt.     | S           | Bipartite   | SNL     | B          | ✓         | ✓       |
| Nearyth [29]                           | Filt. → VS    | VS → Filt.     | S           | Bipartite   | SNL     | B          | ✓         | ✓       |
| ω-structure [33]                       | Filt. → VS    | Interpolation  | S           | Cyclic      | NAD     | N/A        | ✓         | ✓       |
| Generalized Spline [32]                | Filt. → VS    | Interpolation  | S           | Any         | V       | Any        | ✓         | ✓       |
| Lifting [30]                           | Filt. → VS    | VS → Filt.     | V           | Tree        | Any     | B          | ✓         | ✓       |
| Wavelets on balanced tree [42]         | Filt. → VS    | VS → Filt.     | V           | Any         | Any     | LoS        | ✓         | ✓       |
| Subgraph [30]                          | Filt. → VS    | VS → Filt.     | S           | Any         | Any     | N/A        | ✓         | ✓       |
| Qualified sampling [40]                | Filt. → VS    | Interpolation  | S           | Any         | Any     | N/A        | ✓         | ✓       |
| Uniqueness set [31]                    | Filt. → VS    | Interpolation  | S           | Any         | Any     | N/A        | ✓         | ✓       |
| Sampling set selection [43]            | Filt. → VS    | VS → Filt.     | S           | Any         | SNL     | O, B       | ✓         | ✓       |
| Proposed                               | Filt. → SS    | SS → Filt.     | S           | Any         | Any     | O, B       | ✓         | ✓       |

\(^1\) Building blocks for the analysis transform. Filt. → VS: Filtering then vertex domain downsampling. Filt. → SS: Filtering then spectral domain downsampling.

\(^2\) Building blocks for the synthesis transform. VS → Filt.: Vertex domain upsampling then filtering. SS → Filt.: Spectral domain upsampling then filtering. Interpolation: Interpolation operator that cannot be separated into VS → Filt.

\(^3\) Domain for filter design. S: Graph frequency domain. V: Vertex domain.

\(^4\) Applicable variation operators. SNL: symmetric normalized graph Laplacian. NAD: normalized adjacency matrix. LoS: Laplacian on subgraphs.

\(^5\) Orthogonality. O: Orthogonal. B: Biorthogonal.

\(^6\) Compact support.

\(^7\) Perfect reconstruction property.

\(^8\) The constraint on the ω-structure is described in [33] that includes M-block cyclic graphs. Precisely, this constraint can be relaxed by using a similarity transformation [23].

\(^9\) Perfect reconstruction is possible only for bipartite graphs.

Fig. 2. Upsampling of signals on graphs. The signal is upsampled by two. The shaded areas represent different signals. (a) Original graph signal. (b) (GU1): vertex domain upsampling. (c) (GU2): spectral domain upsampling.

Fig. 3. CS GWTs with vertex domain sampling.

underlying graph is bipartite and the variation operator is a symmetric normalized graph Laplacian or normalized random walk graph Laplacian. Non-bipartite graphs should be simplified to bipartite ones before transformation by the CS GWTs to guarantee the perfect reconstruction condition.

Fig. 3 illustrates the entire transformation for one bipartite graph. In the figure, \(H_k := U H_k(\Lambda) U^\top\) is the \(k\)th filter in the analysis filter bank and \(G_k := U G_k(\Lambda) U^\top\) is the \(k\)th filter in the synthesis filter bank, in which

\[
H_k(\Lambda) = \text{diag}(H_k(\lambda_0), H_k(\lambda_1), \ldots, H_k(\lambda_{N-1}))
\]

\[
G_k(\Lambda) = \text{diag}(G_k(\lambda_0), G_k(\lambda_1), \ldots, G_k(\lambda_{N-1})).
\]

Let \(\mathcal{B} = (\mathcal{L}, \mathcal{H}, \mathcal{E})\) be a bipartite graph only having edges between vertex sets \(\mathcal{L}\) and \(\mathcal{H}\). The number of samples in each channel is determined on the basis of the graph-coloring result. Down- and upsampling in the vertex domain for \(\mathcal{B}\), represented in (2) and (3), is defined in matrix notation as follows:

\[
S_{d,0} = I_L \in \{0,1\}^{|\mathcal{L}| \times N}, \quad S_{u,0} = S_{d,0}^T
\]

\[
S_{d,1} = I_H \in \{0,1\}^{|\mathcal{H}| \times N}, \quad S_{u,1} = S_{d,1}^T
\]

where \(I_L\) and \(I_H\) are submatrices of \(I_N\) whose rows correspond to the indices of \(\mathcal{L}\) and \(\mathcal{H}\), respectively. That is, the sampled signal can be represented as \(f_{d,0} = S_{d,0} f\) and so on.

The two-channel GWT shown in Fig. 3 is designed to satisfy the following perfect reconstruction condition.

\[
T_p = G_0 S_{u,0} S_{d,0} H_0 + G_1 S_{u,1} S_{d,1} H_1 = \hat{c} I_N,
\]

where \(c \in \mathbb{R}\). [10] is further represented as the condition for spectral graph filters as follows.

\[
G_0(\lambda) H_0(\lambda) + G_1(\lambda) H_1(\lambda) = c^2
\]

\[
G_0(\lambda) H_0(2 - \lambda) - G_1(\lambda) H_1(2 - \lambda) = 0.
\]
A. Framework and Perfect Reconstruction Condition

The framework of the proposed transform is shown in Fig. 5, where $H_k(A)$ and $G_k(A)$ are the same as (8). It seems to be similar to the existing CS GWTs, but it differs in that all operations are performed in the graph frequency domain.

The sampling matrices are defined as follows:

$$
\tilde{S}_{d,0} = [I_{N/2} \ J_{N/2}], \quad \tilde{S}_{u,0} = \tilde{S}_{d,0}^T,
\tilde{S}_{d,1} = [I_{N/2} - J_{N/2}], \quad \tilde{S}_{u,1} = \tilde{S}_{d,1}^T.
$$

The down- and up-sampling operations for the low-pass branch are, respectively, the same as (GD2) and (GU2) introduced in Section III-B whereas those in the high-pass branch are modulated versions of (GD2) and (GU2).

By using this structure, the following theorem gives the perfect reconstruction condition.

**Theorem 1.** The two-channel CS GWT with spectral domain sampling shown in Fig. 5 is a perfect reconstruction transform if the graph spectral responses of the filters satisfy the following relationship:

$$
G_0(\lambda_i)H_0(\lambda_i) + G_1(\lambda_i)H_1(\lambda_i) = c^2 \quad (18)
$$

$$
G_0(\lambda_i)H_0(\lambda_{N-i-1}) - G_1(\lambda_i)H_1(\lambda_{N-i-1}) = 0. \quad (19)
$$

**Proof.** The output signal $\tilde{t}$ in Fig. 5 is represented as

$$
\tilde{t} = U_0G_0(A)\tilde{S}_{u,0}\tilde{S}_{d,0}H_0(A)U_0^Tf + U_0G_1(A)\tilde{S}_{u,1}\tilde{S}_{d,1}H_1(A)U_0^Tf.
$$

Since $U_0$ is an orthogonal matrix, if the transfer matrix

$$
\tilde{T}_s = G_0(A)\tilde{S}_{u,0}\tilde{S}_{d,0}H_0(A) + G_1(A)\tilde{S}_{u,1}\tilde{S}_{d,1}H_1(A)
$$

is the identity matrix, the whole transform becomes a perfect reconstruction system. By substituting (17) into (21), $\tilde{T}_s$ becomes

$$
\tilde{T}_s = G_0(A)\begin{bmatrix} I_{N/2} & I_{N/2} \ J_{N/2} & J_{N/2} \end{bmatrix} H_0(A) + G_1(A)\begin{bmatrix} I_{N/2} & -J_{N/2} \ J_{N/2} & -I_{N/2} \end{bmatrix} H_1(A)
$$

$$
= G_0(A) (I_N + J_N) H_0(A) + G_1(A) (I_N - J_N) H_1(A)
$$

$$
= (G_0(A)H_0(A) + G_1(A)H_1(A))
$$

$$
+ (G_0(A)H_0(A') - G_1(A)H_1(A')).J_N,
$$

where $A' = \text{diag}(\lambda_{N-1}, \ldots, \lambda_0)$. If the filters satisfy (18) and (19), we have

$$
G_0(A)H_0(A) + G_1(A)H_1(A) = c^2 I_N \quad (23)
$$

$$
G_0(A)H_0(A') - G_1(A)H_1(A') = 0_N. \quad (24)
$$

This leads to $\tilde{T}_s = c^2 I_N$. \hfill \square
Let us define $A_u = \text{diag}(\lambda_0, \ldots, \lambda_{N-1})$ and $A'_u = \text{diag}(\lambda_{N-1}, \ldots, \lambda_0)$. The matrix of the analysis transform $H \in \mathbb{R}^{N \times N}$ is represented as

$$H = \begin{bmatrix} \tilde{S}_{d,0} & \tilde{S}_{d,1} \end{bmatrix} \begin{bmatrix} H_0(A) & H_1(A) \end{bmatrix} \begin{bmatrix} U_0^T \\ U_0 \end{bmatrix}. \quad (27)$$

By looking at (25) and doing some elementary calculations, (27) can be rewritten as follows.

$$H = \begin{bmatrix} H_0(A_u) & H_0(A'_u) \\ H_1(A_u) & -H_1(A'_u) \end{bmatrix} \begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} U_0^T. \quad (28)$$

Hence, the polyphase matrix in the graph frequency domain $H_{\text{poly}}$ is

$$H_{\text{poly}} = \begin{bmatrix} H_0(A_u) & H_0(A'_u) \\ H_1(A_u) & -H_1(A'_u) \end{bmatrix}. \quad (29)$$

Similarly, the synthesis transform matrix $G \in \mathbb{R}^{N \times N}$ is represented as

$$G = \begin{bmatrix} U_0 & U_0 \end{bmatrix} \begin{bmatrix} G_0(A) & G_1(A) \\ \tilde{S}_{u,0} & \tilde{S}_{u,1} \end{bmatrix}. \quad (30)$$

After a calculation similar to that for the analysis side, we obtain the following equivalent expression,

$$G_{\text{poly}} = \begin{bmatrix} G_0(A_u) & G_1(A_u) \\ G_0(A'_u) & -G_1(A'_u) \end{bmatrix}. \quad (32)$$

The polyphase representation in the graph frequency domain is illustrated in Fig. 7.

As in the traditional polyphase structure, it is clear that the graph signal is perfectly recovered when the product of the analysis and synthesis polyphase matrices becomes the identity matrix, i.e., $G_{\text{poly}} H_{\text{poly}} = I_N$.

2) Bipartite Case: Here, suppose the underlying graph is a bipartite one with $|\mathcal{L}| = |\mathcal{H}|$ and the variation operator is a symmetric normalized graph Laplacian. Without loss of generality, we can assume that the vertices of the first half correspond to $\mathcal{L}$ and those of the second half correspond to $\mathcal{H}$. In this case, the normalized graph Laplacian can be written as

$$L = \begin{bmatrix} L_{\mathcal{L}\mathcal{L}} & L_{\mathcal{L}\mathcal{H}} \\ L_{\mathcal{H}\mathcal{L}} & L_{\mathcal{H}\mathcal{H}} \end{bmatrix} = \begin{bmatrix} U_{\mathcal{L}\mathcal{L}} & U_{\mathcal{L}\mathcal{H}} \\ U_{\mathcal{H}\mathcal{L}} & -U_{\mathcal{H}\mathcal{H}} \end{bmatrix} \begin{bmatrix} \Lambda_{\mathcal{L}} & 0 \\ 0 & 2I - \Lambda_{\mathcal{L}} \end{bmatrix} \begin{bmatrix} U_{\mathcal{L}\mathcal{L}}^T & U_{\mathcal{L}\mathcal{H}}^T \\ U_{\mathcal{H}\mathcal{L}}^T & -U_{\mathcal{H}\mathcal{H}}^T \end{bmatrix}.$$

$$= \begin{bmatrix} 2U_{\mathcal{H}\mathcal{L}}(1 + \Lambda_{\mathcal{L}})U_{\mathcal{L}\mathcal{L}}^T & I \\ I & 2U_{\mathcal{L}\mathcal{L}}(1 + \Lambda_{\mathcal{L}})U_{\mathcal{H}\mathcal{H}}^T \end{bmatrix}. \quad (33)$$

We can rewrite the signal before it is transformed using $H_{\text{poly}}$ as follows:

$$\begin{bmatrix} I_{N/2} \\ J_{N/2} \end{bmatrix} \begin{bmatrix} U_{\mathcal{L}\mathcal{L}}^T & U_{\mathcal{L}\mathcal{H}}^T \\ U_{\mathcal{H}\mathcal{L}}^T & -U_{\mathcal{H}\mathcal{H}}^T \end{bmatrix} \begin{bmatrix} f_u \\ f_l \end{bmatrix} = \begin{bmatrix} U_{\mathcal{L}\mathcal{L}}^T f_u + U_{\mathcal{H}\mathcal{H}}^T f_l \\ J(U_{\mathcal{H}\mathcal{L}} f_u - U_{\mathcal{H}\mathcal{H}} f_l) \end{bmatrix}. \quad (34)$$

$^2$Note that the order of the eigenvalues in $\Lambda_{\mathcal{H}}$ is flipped.
Fig. 6. L-level octave-band analysis filter bank with spectral domain sampling. The direct structure (a) and its equivalent structure after merging the filters and sampling operators (b). $H_k$ and $\hat{S}_k$ are the $k$th merged filters and sampling matrices, respectively.

![GraphSS-I](image1) ![GraphSS-O](image2) ![GraphSS-B](image3)

Fig. 7. Polyphase representation of analysis filter bank with spectral domain sampling, where $f_u$ and $f_l$ are the upper and lower halves of $f$, respectively.

![GraphSS-1](image4) ![GraphSS-O](image5) ![GraphSS-B](image6)

Fig. 8. Proposed CS GWTs with spectral domain sampling. Light blue and yellow dashed lines represent (18) and (19), respectively.

where $f_u$ and $f_l$ are the upper and lower halves of $f$, respectively. Note that we can compute $U_{LL}^T f_u$ and $U_{HL}^T f_l$ separately; the downsampling operation can be performed before the graph Fourier transform as in the classical case.

D. Design Methods

By using a method similar to [21], both orthogonal and biorthogonal filters can be designed on the basis of those used in classical signal processing.

1) Orthogonal Solution: Similarly to the design of the graph-QMF [25], the proposed orthogonal CS GWT uses one prototype filter $H_0(\lambda_i)$. The remaining $G_0(\lambda_i), H_1(\lambda_i),$ and $G_1(\lambda_i)$ are calculated from $H_0(\lambda_i)$, i.e., $H_k(\lambda_i) = G_k(\lambda_i)$ and $H_k(\lambda_i) = H_0(\lambda_{N-1-i})$. $H_0(\lambda_i)$ has to satisfy the following condition to ensure perfect reconstruction:

$$H_0^2(\lambda_i) + H_0^2(\lambda_{N-1-i}) = c^2.$$  (35)

We utilize the frequency responses of the time domain filters to design $H_0(\lambda_i)$. First, a real-valued function $H_{\text{freq}}(\omega)$, where $\omega \in [0, \pi]$, is obtained from the time-domain filter; then $H_0(\lambda_i)$ is calculated according to the eigenvalue distribution of the graph Laplacian. That is,

$$H_0(\lambda_i) = H_{\text{freq}}\left(\pi i \frac{\lambda_i}{N}\right).$$  (36)

2) Biorthogonal Solution: Similar to graphBior [27] and wavelets in the time domain, the high-pass filters used in the proposed biorthogonal CS GWT are defined from the low-pass filters as

$$H_1(\lambda_i) = G_0(\lambda_{N-1-i}), \quad G_1(\lambda_i) = H_0(\lambda_{N-1-i}).$$  (37)

This leads to the following condition:

$$H_0(\lambda_i)G_0(\lambda_i) + H_0(\lambda_{N-1-i})G_0(\lambda_{N-1-i}) = c^2.$$  (38)

VI. RELATIONSHIP BETWEEN VERTEX AND SPECTRAL DOMAIN SAMPLING-BASED APPROACHES

This section examines the relationship between CS GWTs with vertex domain sampling and those with spectral domain sampling.

A. Reduced-Size Bipartite Graphs with Kron Reduction

Kron reduction is a widely used method to reduce the size of the graph Laplacian, especially in multiscale transforms of graph signals [46]. The following theorem describes the condition under which vertex domain sampling is identical to spectral domain sampling.

**Theorem 2.** If the original graph is bipartite with $|L| = |H| = N/2$ and the reduced graph is obtained by Kron reduction [46], the signal downsampled by (GD2) is equivalent to the one downsampled by (GD1) when the symmetric normalized graph Laplacian is used as the variation operator.

**Proof.** Without loss of generality, we can assume the vertices in the first half correspond to $\hat{L}$ and in the second half to $\hat{H}$. The Kron reduction of the symmetric normalized graph Laplacian is represented as

$$\mathcal{L}_{\text{reduced}} = \mathcal{L}_{\hat{L}\hat{L}} - \mathcal{L}_{\hat{L}\hat{H}}\mathcal{L}^{-1}_{\hat{H}\hat{H}}\mathcal{L}_{\hat{H}\hat{L}} = \mathbf{I} - 4\mathbf{U}_{\hat{L}\hat{L}}(1 + \mathbf{A}_L)\mathbf{U}_{\hat{H}\hat{L}}^\top(1 + \mathbf{A}_H)\mathbf{U}_{\hat{L}\hat{L}}^\top = \mathbf{I} - 2\mathbf{U}_{\hat{L}\hat{L}}(1 + \mathbf{A}_L)(1 + \mathbf{A}_H)^\top - 2\mathbf{U}_{\hat{L}\hat{L}}(2\mathbf{A}_L + \mathbf{A}_L^2)\mathbf{U}_{\hat{L}\hat{L}}^\top.$$  (39)
where we have used the same notation as in (33). Therefore, the eigenvector and eigenvalue matrices of the reduced graph are $\sqrt{2}U_{LL}^{\top}$ and $-(2A_L + \Lambda^2)$, respectively.

Recall that the downsampling matrix in the vertex domain can be represented as

$$S_d = \begin{bmatrix} \mathbf{I}_{|\mathcal{C}|} & 0_{|\mathcal{H}|} \end{bmatrix},$$

and the downsampling matrix created by spectral domain sampling in this case is

$$U_{LL} \begin{bmatrix} \mathbf{I}_{|\mathcal{C}|} & 0_{|\mathcal{H}|} \end{bmatrix} \begin{bmatrix} U_{LL}^{\top} & U_{LL} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{|\mathcal{C}|} & 0_{|\mathcal{H}|} \end{bmatrix}.$$  \hspace{1cm} (41)

This completes the proof.

B. Perfect Reconstruction Condition for Bipartite Graphs

The following theorem reveals that the perfect reconstruction condition for the proposed method coincides with that of vertex domain sampling in a special case.

**Theorem 3.** If the underlying graph is a bipartite graph and the symmetric normalized graph Laplacian is used as the variation operator, the perfect reconstruction condition for the two-channel CS GWT using the vertex domain sampling is identical to that using spectral domain sampling.

**Proof.** The eigenvalue distribution of the normalized graph Laplacian of a bipartite graph is symmetric with respect to $\lambda = 1$, and the maximum eigenvalue is 2. Therefore, $\lambda_{N-1} = 2 - \lambda_1$, which implies that the perfect reconstruction condition (18) and (19) is identical to (11) and (12). \hspace{1cm} \Box

VII. EXPERIMENTS

Nonlinear approximations and denoising for synthetic graph signals were selected as target applications of the numerical performance comparisons. The bases of the comparisons were existing graph wavelets. Hereafter, we abbreviate the proposed filter bank as graphSS, where SS refers to spectral sampling.

A. Setup

The prototype filters used for graphSSs were:

- The ideal filters (denoted as GraphSS-I). The set of ideal filters clearly satisfies (18) and (19).
- The orthogonal filter set designed with the Meyer kernel as $H_{\text{first}}(\omega)$ (denoted as GraphSS-O).
- The biorthogonal filter set based on 9/7-CDF filters (denoted as GraphSS-B).

Their spectral characteristics are shown in Figs. 8(a)–(c). As described above, the proposed transforms can be applied to both combinatorial and symmetric normalized graph Laplacians, so we decided to examine both cases. In what follows, the combinatorial version is specified by (C), e.g., GraphSS-O(C), whereas the normalized one is specified by (N).

The compared methods were GraphQMF [26], GraphBior [27], GraphFC [21]. Note that the existing CS GWTs require

3Note that $\mathcal{L}_{\text{reduced}}$ would have self-loops and thus, its eigenvalues are not always positive in general.

bipartition of the underlying graph. We used the coloring-based bipartition [26], [27], [45] as suggested by the authors. All wavelets decomposed graph signals into two-level octave bands.

The three synthetic graph signals shown in Fig. 9 were used in the experiments. They have different characteristics; the signal shown in Fig. 9(a) is localized in the spectral domain (see Fig. 9(d)). We designed the signal by using the method in (22). Specifically, $f = \sum_{j=1}^{4} f_j / ||f_j||_{\infty}$, where

$$f_j[i] = \mathbb{1}_{\{\text{vertex } i \text{ is in cluster } j\}} \sum_{\ell=0}^{N-1} u_\ell[i] \mathbb{1}_{\{\ell \leq \lambda_4 \leq \tau_j\}}.$$  \hspace{1cm} (42)

We took the sequence $\{\tau_j\}_{j=1,\ldots,4}$ to be $[\lambda_9, \lambda_{29}, \lambda_{59}, \lambda_{79}, [\lambda_{149}, \lambda_{169}], [\lambda_{299}, \lambda_{319}]$. The second graph signal shown in Fig. 9(b) is smooth in the graph frequency domain, where $f[j] = \exp(-\lambda_4/4)$, and its spectrum is shown in Fig. 9(e). The third one is the sum of the spectral localized signal and the exponential one, as shown in Fig. 9(f).

B. Nonlinear Approximation

In nonlinear approximation, we keep the fraction of the transformed coefficients having high absolute values and set the remaining coefficients to zero. Figs. 10(a)–(c) show the results. The wavelets with the combinatorial Laplacian gave better SNRs than the conventional methods did and the ones with the normalized graph Laplacian.

When the spectrum was highly localized, graphSSs significantly outperformed the existing methods, as shown in Fig. 10(a). For the smooth signal, the gap between the SNRs was smaller, but the proposed method still had better reconstruction quality. Interestingly, GraphSS-I was not always the best among the proposed transforms (see Fig. 10(b)).

C. Denoising

Hard thresholding with a threshold of $3\sigma$ was performed on each subband except the lowest to remove additive white Gaussian noise with variance $\sigma^2$. The SNRs after denoising are compared in Table 11. The proposed methods with the combinatorial graph Laplacian outperformed the conventional methods in most cases. The graphSSs outperformed the conventional methods for all signals and noise levels. GraphSS-I(C) was good for the signals on the community graph and the Swiss roll graph, whereas GraphSS-B(C) was better than the ideal for the sensor graph.

The denoised signals for the sensor graph are shown in Fig. 11 where $\sigma = 1/4$. As in the numerical comparison, GraphSS-B(C) presented cleaner signals than others.

D. Discussion: Ideal vs. Non-Ideal Filters

An interesting and somewhat counter-intuitive result of the experiments was that, in some cases, non-ideal filters (GraphSS-O and GraphSS-B) outperformed the ideal filter (GraphSS-I). To understand why this happened, we should note that our approach involves spectral folding of frequencies
based on their indices, i.e., their ordering in the frequency domain. Thus, each ideal filter removes exactly half the frequencies, but this may correspond to very different ranges of variation, as will be seen next. Also, non-ideal filters are more localized in the vertex domain. We discuss these two advantages of non-ideal filters in what follows.

1) Passband Widths: In the case of bipartite graphs whose variation operator is the symmetric normalized Laplacian, the frequencies are naturally symmetric around $\lambda = 1$, which is not only the center of symmetry for the frequencies but also the middle point of the range of frequency variation.

Instead, in our setting, our spectral folding imposes symmetry. For an arbitrary graph, the exact distribution of frequencies does not exhibit symmetry and there could be more low (high) variation eigenvectors than low (high) variation ones. Thus, the “ideal” low-pass filter, i.e., the one preserving the first $N/2$ frequencies, can represent very different variation ranges for different graphs, and thus, it is not guaranteed to be always the best choice of low-pass filter.

As an example, let us assume that the frequency range $[0, \lambda_{\text{max}}/2]$ contains more than half the frequencies (eigenvalues). In this case, an ideal low-pass filter passing through the frequency range $[0, \lambda_{N/2}]$ would eliminate some of the frequencies in that range, since its cutoff frequency will be at $\lambda_{N/2} < \lambda_{\text{max}}/2$. Non-ideal filters, in contrast, can use a more natural range of frequencies in the decomposition because of the overlapping of the frequency responses for the low-pass and high-pass filters. This allows the low-pass channel to include more of the “natural” low frequencies, i.e., those having lower variation.

To validate the above discussion numerically, the low-pass filtered signals obtained from graphSS were compared with those given by the ideal low-pass filter based on the frequency
TABLE II

DENOISING RESULTS: SNR (dB), AVERAGE OF 100 RUNS.

| Graph        | GraphQMF | GraphBior | GraphFC | GraphSS-O(N) | GraphSS-O(C) | GraphSS-B(N) | GraphSS-B(C) | GraphSS-I(N) | GraphSS-I(C) |
|--------------|----------|-----------|---------|--------------|--------------|--------------|--------------|--------------|--------------|
| **Community**|          |           |         |              |              |              |              |              |              |
| Original     | 11.69    | 11.93     | 11.59   | 11.34        | 12.95        | 11.29        | 13.95        | 11.80        | 13.45        |
| Noisy       | 9.37     | 7.33      | 3.60    | 7.73         | 10.25        | 7.72         | 10.69        | 9.29         | 5.81         |
| **Sensor**   |          |           |         |              |              |              |              |              |              |
| Original     | 10.53    | 10.58     | 10.33   | 9.13         | 12.54        | 9.20         | 12.35        | 9.29         | 12.87        |
| Noisy       | 8.80     | 6.11      | 2.56    | 5.17         | 8.38         | 5.11         | 8.74         | 5.23         | 8.49         |
| **Swiss Roll**|         |           |         |              |              |              |              |              |              |
| Original     |          |           |         |              |              |              |              |              |              |
| Noisy       | 11.80    | 5.81      | -0.21   | 5.17         | 8.38         | 5.11         | 8.74         | 5.23         | 8.49         |

Fig. 11. Denoising results.

TABLE III

DIFFERENCES BETWEEN \( \bar{f}_{\text{value}} \) AND \( \bar{f}_{0,X} \) (AVERAGE OF 100
INDEPENDENT RUNS)

| Graph signal | \(||f_{\text{value}} - f_{0,1}|||^2| | 0.42 | 0.54 |
|--------------|-------------------|-------------|-------------|
|                \(||f_{\text{value}} - f_{0,0}|||^2| | 0.42 | 0.27 |
|                \(||f_{\text{value}} - f_{0,0}|||^2| | 0.49 | 0.23 |

\( E_X[i] := (\bar{f}_{\text{value}}[i] - \bar{f}_{0,X}[i])^2 \), are also shown in Figs. 12(e) and (f) for a clear visualization.

For the signal on the regularly distributed sensor network, all graphSSs present similar results in their passband, but the non-ideal filters have sidelobes in the transition bands. In contrast, \( \bar{f}_{0,1} \) is far from \( \bar{f}_{\text{value}} \) for the graphs with irregularly distributed sensors, whereas the non-ideal filters have smaller maximum errors than those of GraphSS-I in that case. The errors are numerically compared in Table III. For the signal of Fig. 12(a), all low-pass filters of graphSS show comparable errors with respect to \( f_{\text{value}} \), whereas GraphSS-I has a larger error than those of GraphSS-O and GraphSS-O for the signal of Fig. 12(b).

2) Filter Localization: Additionally, we compared the filter localizations in the vertex domain. As described in Section V-D, the filters for graphSS are designed on the basis of frequency values. In general, the filters are global operators in the vertex domain, but their spreads differ depending on the prototype filters.

Fig. 13 shows examples of the low-pass filter atoms at the

\[ H_{\text{value}}(\Lambda) = \text{diag}(1, \ldots, 1, 0, \ldots, 0), \]  
\[ \#(\lambda_{\text{max}}/2) \]

where \( \#(\lambda_{\text{max}}/2) \) represents the number of eigenvalues smaller than \( \lambda_{\text{max}}/2 \). In the following, the signals that were low-pass filtered by \( H_{\text{value}}(\Lambda) \) of GraphSS-X (X ∈ \{I, O, B\}) are specified as \( f_{0,X} \), while those filtered with \( H_{\text{value}}(\Lambda) \) are represented as \( \bar{f}_{\text{value}} \).

Signals on two sensor networks with \( N = 100 \) were used in the experiments. These networks had different sensor distributions, as shown in Fig. 12(a) and (b). That is, the vertices of the graph shown in Fig. 12(a) are randomly distributed, whereas those in Fig. 12(b) have a concentrated region at the bottom left. Hence, the eigenvalue distributions of their combinatorial graph Laplacian are different. Specifically, the graph shown in Fig. 12(a) has \( \lambda_{N/2} = 7.89 \) and \( \lambda_{\text{max}}/2 = 7.50 \), while that in Fig. 12(b) has \( \lambda_{N/2} = 5.73 \) and \( \lambda_{\text{max}}/2 = 15.40 \), i.e., \( \#(\lambda_{\text{max}}/2) > N/2 \).

Despite the difference between the eigenvalue distributions, the spectra of both signals are defined similarly on the basis of the frequency value:

\[ f[\epsilon] = \exp(\lambda_i/4) + \epsilon, \]

where \( \epsilon \) is zero-mean i.i.d. Gaussian noise with standard deviation \( \sigma = 0.05 \).

The original and filtered spectra are shown in Figs. 12(c) and (d). The squared differences between \( f_{\text{value}} \) and \( f_{0,X} \), i.e.,
also showed the theoretical relationship between the conventional graph wavelets with vertex domain sampling and the proposed ones. In experiments on nonlinear approximation and denoising, our CS GWTs outperformed several other methods. Future work will include devising a fast computation method, $M$-channel filter bank design, and dictionary learning with spectral domain sampling.

**REFERENCES**

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

[2] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 1644–1656, Jun. 2013.

[3] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing,” *Proc. IEEE, in press*, 2018.

[4] N. Leonard and D. Van De Ville, “ Tight wavelet frames on multislice graphs,” *IEEE Trans. Signal Process.*, vol. 16, no. 13, pp. 3357–3367, Jul. 2013.

[5] B. A. Miller, M. S. Beard, P. J. Wolfe, and N. T. Bliss, “A spectral framework for anomalous subgraph detection,” *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4191–4206, Aug. 2015.

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

[7] K. Yamamoto, M. Onuki, and Y. Tanaka, “Deblurring of point cloud attributes in graph spectral domain,” in *Proc. Int. Conf. Image Process.*, 2016, pp. 1559–1563.

[8] G. Cheung, E. Magli, Y. Tanaka, and M. Ng, “Graph spectral image processing,” *Proc. IEEE, in press*, 2018.

[9] K. Yamamoto, M. Onuki, and Y. Tanaka, “Deblurring of point cloud attributes in graph spectral domain,” in *Proc. Int. Conf. Image Process.*, 2016, pp. 1559–1563.

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

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

[12] X. Liu, G. Cheung, X. Wu, and D. Zhao, “Random walk graph Laplacian-based smoothness prior for soft decoding of JPEG images,” *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 509–524, Feb. 2017.

[13] J. Zhang and J. M. F. Moura, “Diffusion in social networks as SIS epidemics: Beyond full mixing and complete graphs,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 537–551, Aug. 2014.

[14] S. Ono, I. Yamada, and I. Kumazawa, “Total generalized variation for graph signals,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2015, pp. 5456–5460.

[15] H. Higashi, T. M. Rutkowsky, T. Tanaka, and Y. Tanaka, “Multilinear discriminant analysis with subspace constraints for single-trial classification of event-related potentials,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1295–1305, Oct. 2016.

[16] M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: Going beyond euclidean data,” *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[17] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, “Blind identification of graph filters,” *IEEE Trans. Signal Process.*, vol. 65, no. 5, pp. 1146–1159, Mar. 2016.

[18] R. Rustamov and L. Guibas, “Wavelets on graphs via deep learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 998–1006.

[19] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, Mar. 2011. [Online]. Available: [http://wiki.epfl.ch/gwlt](http://wiki.epfl.ch/gwlt)

[20] A. Sakiyama, K. Watanabe, and Y. Tanaka, “Spectral graph wavelets and filter banks with low approximation error,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 3, pp. 230–245, Sep. 2016.
[22] D. I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vanderghynst, “Spectrum-adapted tight graph wavelet and vertex-frequency frames,” IEEE Trans. Signal Process., vol. 63, no. 16, pp. 4223–4235, Aug. 2015. [Online]. Available: http://documents.epfl.ch/users/s/sh/shuman/www/publications.html

[23] D. B. H. Tay, Y. Tanaka, and A. Sakiyama, “Near orthogonal oversampled graph filter banks,” IEEE Trans. Signal Process., vol. 65, no. 14, pp. 3578–3590, Jul. 2014.

[24] S. K. Narang and A. Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” IEEE Trans. Signal Process., vol. 60, no. 6, pp. 2786–2799, Jun. 2012. [Online]. Available: http://biron.usc.edu/wiki/index.php/Graph_Filterbanks

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

[26] S. K. Narang and A. Ortega, “Perfect reconstruction two-channel wavelet filter banks for graph structured data,” IEEE Trans. Signal Process., vol. 60, no. 6, pp. 2786–2799, Jun. 2012. [Online]. Available: http://biron.usc.edu/wiki/index.php/Graph_Filterbanks

[27] Y. Jin and D. I. Shuman, “Critically sampled graph filter banks with polynomial filters from regular domain filter banks,” Signal Processing, vol. 131, pp. 66–72, Feb. 2017.

[28] N. Tremblay and P. Borgnat, “Subgraph-based filterbanks for graph signals,” IEEE Trans. Signal Process., vol. 64, no. 15, pp. 3827–3840, Aug. 2016.

[29] Y. Jin and D. I. Shuman, “An M-channel critically sampled filter bank for graph signals,” in Proc. IEEE Int. Conf. Acoust. Speech, Signal Process., 2017, pp. 3909–3913.

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

[31] O. Teke and P. P. Vaidyanathan, “Extending classical multirate signal processing theory to graphs—Part II: M-Channel filter banks,” IEEE Trans. Signal Process., vol. 65, no. 2, pp. 423–437, Jan. 2016.

[32] A. Sakiyama and Y. Tanaka, “Construction of undersampled graph filter banks via row subset selection,” in Proc. IEEE Global Conf. Signal Inf. Process., 2016, pp. 322–326.

[33] P. P. Vaidyanathan, Multirate Systems and Filter Banks. NJ: Prentice-Hall, 1993.

[34] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 3rd ed. Pearson, 2009.

[35] M. Vetterli, J. Kovacevic, and V. K. Goyal, Foundations of Signal Processing. Cambridge University Press, 2014.

[36] Y. Tanaka, “Spectral domain sampling of graph signals,” arXiv preprint arXiv:1706.05147, 2017.

[37] S. K. Narang and A. Ortega, “Lifting based wavelet transforms on graphs,” in Proc. Asia-Pacific Signal Inf. Process. Assoc. Annual Summit Conf., 2009, pp. 441–444.

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

[39] K. Watanabe, A. Sakiyama, Y. Tanaka, and A. Ortega, “Critically-sampled graph filter banks with spectral domain sampling,” in Proc. IEEE Int. Conf. Acoust. Speech, Signal Process., 2018, pp. 4054–4058.

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

[41] A. Anis and A. Ortega, “Critical sampling for wavelet filterbanks on arbitrary graphs,” in Proc. Int. Conf. Acoust. Speech, Signal Process., 2017, pp. 3889–3893.

[42] B. Aspvall and J. R. Gilbert, “Graph coloring using eigenvalue decomposition,” SIAM Journal on Algebraic Discrete Methods, vol. 5, no. 4, pp. 526–538, 1984.

[43] F. Harary, D. Hsu, and Z. Miller, “The biparticity of a graph,” J. Graph Theory, vol. 1, no. 2, pp. 131–133, 1977.

[44] F. Dorfler and F. Bullo, “Krone reduction of graphs with applications to electrical networks,” IEEE Trans. Circuits Syst. I, vol. 60, no. 1, pp. 150–163, Jan. 2013.

[45] H. Q. Nguyen and M. N. Do, “Downsampling of signals on graphs via maximum spanning trees,” IEEE Trans. Signal Process., vol. 63, no. 1, pp. 182–191, Jan. 2015.

[46] S. K. Narang and A. Ortega, “Local two-channel critically sampled filterbanks on graphs,” in Proc. Int. Conf. Image Process., 2010, pp. 333–336.

[47] D. Ron, I. Safro, and A. Brandt, “Relaxation-based coarsening and multiscale graph organization,” Multiscale Modeling & Simulation, vol. 9, no. 1, pp. 407–423, Sep. 2011.

[48] A. Cohen, I. Daubechies, and J.-C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” Communications on pure and applied mathematics, vol. 45, no. 5, pp. 485–560, Jun. 1992.

[49] G. Strang and T. Q. Nguyen, Wavelets and Filter Banks. MA: Wellesley-Cambridge, 1996.

[50] M. Bellanger, G. Bonnerot, and M. Coudreou, “Digital filtering by polyphase network: Application to sample-rate alteration and filter banks,” IEEE Trans. Acoust., Speech, Signal Process., vol. 24, no. 2, pp. 109–114, Apr. 1976.

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

[52] Y. Tanaka and A. Sakiyama, “A ‘polyphase’ structure of two-channel spectral graph wavelets and filter banks,” in Proc. IEEE Int. Conf. Acoust. Speech, Signal Process., 2017, pp. 4144–4148.