Bipartite Graph Filter Banks: Polyphase Analysis and Generalization

David B. H. Tay*,+, Antonio Ortega++
+Dept. of Engineering, LaTrobe University, Bundoora, Victoria 3086, Australia.
++ Signal & Image Processing Institute, Ming Hsieh Department of Electrical Engineering,
University of Southern California, Los Angeles, CA 90089, USA

***Abstract*— [1] The work by Narang and Ortega [1], [2] laid the foundations for the two-channel critically sampled perfect reconstruction filter bank for signals defined on undirected graphs. The basic filter bank proposed is applicable only to bipartite graphs but using the notion of separable filtering, the basic filter bank can be applied to any arbitrary undirected graphs. In this work several new theoretical results are presented. In particular, the proposed polyphase analysis yields filtering structures in the downsampled domain that is equivalent to those before downsampling and thus can be exploited for efficient implementation. These theoretical results also provide new insights that can be exploited in the design of these systems. These insights allow us to generalize these filter banks to directed graphs and to using a variety of graph base matrices, while also providing a link to the DSP$_G$ framework of Sandryhaila and Moura [3], [4]. Experiments show evidence that better non-linear approximation results may be obtained by a better select of these base matrices.**

I. Introduction

Recently there has been great interest amongst signal processing researchers to develop techniques for processing signal defined on graphs, i.e. graph signal processing (GSP). This is mainly spurred by the proliferation of applications where data is defined over domains that are irregular, e.g. social, sensor and transportation networks. In [5], [6] good overviews of recent developments and avenues for future work in GSP are presented. One view of GSP is that it is a merging of graph theory with concepts and techniques from regular domain signal processing. Spectral graph theory [7] provides a natural extension of the notion of frequency and frequency domain to the spectral domain for undirected graph signals [5]. The eigenvalues of the Laplacian matrix give the discrete spectral frequencies with this approach [5]. The DSP$_G$ approach of Sandryhaila and Moura [3], [4] uses the adjacency matrix as a generalization of the one sample shift (delay) in 1-D signals and the filtering is based on polynomials of the adjacency matrix.

Many signal processing systems use some form of transform to map the data from the original domain to another. In regular domain signal processing the wavelet transform (and its extensions and generalizations) is perhaps one of the most popular and commonly used transforms, as it provides a decomposition that has time and frequency localization properties [8]–[10]. There have been several approaches proposed in the literature [1], [2], [11]–[17] to extend and adapt the wavelet transform for graph signals. Reference [1] provides a comparison of many of the techniques proposed. For regular domain signals, wavelet transforms based on the two-channel critically sampled perfect reconstruction filter banks are perhaps one of the most popular. Critical sampling allows for efficient non-redundant representation of the signal. However many graph transforms are not critically sampled.

Narang and Ortega [1], [2] laid the foundations for a graph transform that is based on two-channel critically sampled filter banks. The graph filter banks in [1], [2] can be implemented efficiently using distributed processing without the need for any eigendecomposition. The basic filter bank is applicable only to undirected bipartite graphs. However using the notion of the separable filter bank, the basic filter bank can be applied to any arbitrary undirected graph. Other approaches such as in [14], [15] also provide critical sampling but unlike [1], [2] do not have an easy frequency interpretation. More recently, using graph sampling theory, a two-channel critically sampled system was proposed in [18] but its implementation requires eigendecomposition, which can have significant complexity. In [19], a critically sampled system was also proposed. The analysis bank in [19] can be implemented with distributed processing but inversion of a potentially large matrix, which can be computationally costly, is required in the synthesis bank. Extension to M-channel graph filter banks with critical sampling was recently proposed in [20], [21]. One of the main differences between the approach in [21] and those in [1], [2] is in the downsampling operation. In [1], [2] the downsampling sets can be arbitrary, and an exact solution can be found for any bipartite graph. Instead, the approach in [20], [21] requires downsampling to be closely linked to the graph shift operator, in order to replicate the behavior of multi-rate systems in regular domain. Specifically, in a 2-channel filterbank, the two polyphase terms would be obtained by downsampling on the *same set of nodes*, first the original signal and then the shifted original signal (see [20]), while in this paper the original (or filtered) signal is downsampled directly on two disjoint sets of nodes (as in [1], [2]).

The design of bipartite graph filter banks (GFB), which involves the determination of four polynomial spectral filter functions, can be found in [1], [2], [22]–[26]. The technique in [23] was developed using a theoretical framework that is based on the notion of a polyphase representation and ladder structures adapted to graph filter banks (GFB). The

[1]submitted to IEEE Trans. Signal Processing

work in [23] however was focussed on the construction of filters. Here we present new theoretical results that extend the work in [23]. The main contributions are: i) a new polyphase framework for the analysis of the GFB, ii) based on this new framework, a derivation of filtering structures in the downsampled domain (filtering after downsampling) that are equivalent to filtering operations in the original graph (filtering before downsampling), a lead to significantly reduced implementation complexity, iii) a study of the connections between the characteristics of the downsampled graph, and its corresponding filters and signals, and those of the original graph, filters and signals, which sheds some light on the relationship between original and downsampled domain, and could be exploited to design improved multirate systems, and iv) a generalization of the approaches in [1], [2] to directed graphs, and other graph matrices, as well as a link to the DSP$_G$ framework of Sandryhaila and Moura [3], [4]. We provide experimental results to demonstrate that replacing the symmetric normalized Laplacian as "base matrix" in the framework proposed by Narang and Ortega [1], [2] by other base matrices derived from the adjacency but with different normalizations can lead to better filtering.

An overview of the paper is as follows. A review of the fundamentals of graph signal processing and bipartite GFB is presented in Section II. The polyphase analysis is presented in Section III, where efficient polyphase structures are derived. Section IV presents results that lead to a spectral interpretation of the polyphase analysis. Section V explores the relationship between the vertex and spectral domains. Generalizations are presented in Section VI where it is shown that the GFB can also be applied to directed graphs. In Section VII the application to non-linear approximation of graph signals is considered and the effect of using different base matrices is considered. Concluding remarks are found in Section VIII.

II. Definitions and Preliminaries

A brief review of graph theory, graph signal processing and graph filter banks is first presented. We refer the reader to [1], [2], [5], [16] for more details.

A. Spectral Graph Theory

A graph $G = (V, E)$ is defined by the set of vertices V and edges E. A graph can have directed or undirected edges. Initially we will consider only undirected graphs but later some of the results will be generalized to directed graphs as well. The number of vertices is denoted as $N = |V|$ and the vertices are labelled as $1, \ldots, N$. An edge $e \in (i, j) \in E$ connects the two vertices i and j. The adjacency matrix $\mathbf{A}$ is the $N \times N$ symmetric matrix whose element $a_{i,j}$ $(i, j = 1, \ldots, N)$ is positive real and gives the weight of the edge connecting vertices i and j. If $a_{i,j} = 0$ there is no edge connecting vertices i and j. The degree of vertex i is defined as $d_i \equiv \sum_j a_{i,j}$ and the diagonal matrix $\mathbf{D} \equiv \text{diag}(d_i)$. The combinatorial graph Laplacian matrix is defined as $\mathbf{L} \equiv \mathbf{D} - \mathbf{A}$ and the *symmetric normalized* graph Laplacian, with respect to $\mathbf{D}$, is defined as

$$\mathcal{L} \equiv \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

where $\mathbf{I}$ is the identity matrix. Since $\mathcal{L}$ is a real symmetric matrix, it can be decomposed as

$$\mathcal{L} = \sum_{i=1}^{N} \lambda_i \mathbf{u}_i \mathbf{u}_i^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_i)$ and $\mathbf{U} \equiv [\mathbf{u}_1 | \mathbf{u}_2 | \cdots | \mathbf{u}_N]$, with λ_i being the eigenvalue of $\mathcal{L}$ and $\mathbf{u}_i$ the corresponding eigenvector. Now $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ (orthogonal matrix), i.e. the eigenvectors $\mathbf{u}_1, \cdots, \mathbf{u}_N$ form an orthonormal set. The set of eigenvalues $\sigma(G) \equiv \{\lambda_1 \leq \lambda_2 \cdots \leq \lambda_N\}$ is the spectrum of graph G. The eigenvalues are bounded in the interval [0,2].

B. Graph Signal Filtering

A signal over a graph G is a function that maps each vertex i to a numerical value $f(i)$. The graph signal can be represented as the vector $\mathbf{f} = [f(1) \cdots f(N)]^T$. The graph Fourier transform is defined as

$$\hat{f}(\lambda_l) \equiv \sum_{n=1}^{N} f(n) u_l(n) = \mathbf{f}^T \mathbf{u}_l$$

where $\mathbf{u}_l \equiv [u_l(1) \ \cdots \ u_l(N)]^T$ (for $l = 1, \cdots, N$) are the Laplacian eigenvectors. The equation can be written as $\hat{\mathbf{f}} = \mathbf{U}^T\mathbf{f}$ where $\hat{\mathbf{f}} = [\hat{f}(\lambda_1) \cdots \hat{f}(\lambda_N)]^T$ is the vector of spectral components at the graph frequencies. The inverse graph Fourier transform is therefore $\mathbf{f} = \mathbf{U}\hat{\mathbf{f}}$ or in scalar form $f(n) = \sum_{l=1}^{N} \hat{f}(\lambda_l)\, u_l(n)$. Filtering in the spectral domain is defined as $\hat{f}_{out}(\lambda_l) = h(\lambda_l)\hat{f}(\lambda_l)$ where $h(\lambda)$ is the spectral filter in the continuous spectral variable λ. The inverse transform of $\hat{f}_{out}(\lambda_l)$ gives the output in the vertex domain as $f_{out}(n) = \sum_{l=1}^{N} \hat{f}(\lambda_l) h(\lambda_l) u_l(n)$. In vector/matrix form $\mathbf{f}_{out} = \mathbf{H}\,\mathbf{f}$, where $\mathbf{H} = h(\mathcal{L}) \equiv \sum_{i=1}^{N} h(\lambda_i)\mathbf{u}_i\mathbf{u}_i^T = \mathbf{U}\,\text{diag}\{h(\lambda_i)\}\,\mathbf{U}^T$ is the transformation matrix. In general the knowledge of the eigenvalues/eignevectors of the underlying graph G is required for implementing the filtering operation. An eigendecomposition of a large graph is however computationally expensive. When the spectral filter is a polynomial function given by $h(\lambda) = \sum_{k=0}^{K} b_k \lambda^k$, it can be readily shown (using the identity $\mathcal{L}^k = \mathbf{U}\,\text{diag}\{\lambda_i^k\}\,\mathbf{U}^T$) that the transformation matrix is given by $\mathbf{H} = \sum_{k=0}^{K} b_k\, \mathcal{L}^k$. The eigendecomposition of the Laplacian is therefore not required for implementing the filter. Only powers of the Laplacian are required and filtering is equivalent to repeated application of the Laplacian on the input signal which has much lower computationally complexity. Another important property of polynomial filters is localization. A K-hop local neighbourhood for vertex i, denoted by $\mathcal{N}(i, K)$, is defined as the set of (other) vertices that are connected to vertex i by no more than K edges. A filter $h(\mathcal{L})$ is *K-hop localized* if the output $f_{out}(i)$ is determined **only** by input values $f(j)$ in $\mathcal{N}(i, K)$. It can be shown that a degree K polynomial filter is K-hop localized [5], [16] and can be implemented with distributed processing.

C. Biorthogonal Graph Filter Banks

The critically sampled two-channel filter bank proposed in [1], [2] is defined on bipartite graphs. A bipartite graph $G =$

(L, H, E) is a graph whose vertices can be partitioned into two disjoint subsets, i.e. $V = L \bigcup H$ and $L \bigcap H = \emptyset$, such that every edge connects one vertex from L to one vertex from H. Downsampling of a bipartite graph signal *retains only* vertices in L (or H) and discards the other vertices in H (or L). Upsampling inserts the discarded nodes but replaces the signal values with zeros. There are 4 spectral filters in the filter bank: (i) $h_0(\lambda)$ (low-pass analysis), (ii) $h_1(\lambda)$ (high-pass analysis), (iii) $g_0(\lambda)$ (low-pass synthesis) and (iv) $g_1(\lambda)$ (high-pass synthesis). The perfect reconstruction (PR) condition can be given by the following theorem:

Theorem 1: Given *any* bipartite graph $G = (L, H, E)$ and *any* input signal $\mathbf{f}$, the filter bank achieves PR, i.e. the reconstructed signal $\mathbf{f}^R = \mathbf{f}$, *if and only if* for $0 \leq \lambda \leq 2$

$$h_0(\lambda)g_0(\lambda) + h_1(\lambda)g_1(\lambda) = 2 \tag{1}$$

$$h_0(2-\lambda)g_0(\lambda) - h_1(2-\lambda)g_1(\lambda) = 0 \tag{2}$$

The proof of Theorem 1 above can be found in Section III.B in [1]. The next lemma (from [23]) gives the PR condition with polynomial filters.

Lemma 1: With polynomial functions for $h_0(\lambda)$, $g_0(\lambda)$, $h_1(\lambda)$ and $g_1(\lambda)$, conditions (1) and (2) in Theorem 1 are satisfied *if and only if*

$$g_0(\lambda) = h_1(2-\lambda), \quad g_1(\lambda) = h_0(2-\lambda) \tag{3}$$

$$h_0(\lambda)h_1(2-\lambda) + h_0(2-\lambda)h_1(\lambda) = 2 \tag{4}$$

The PR conditions in Lemma 1 can be expressed in alternate form using the polyphase representation proposed in [23]. To achieve this the filter functions are expressed as polynomials of the shifted-frequency variable $\mu = \lambda - 1 \in [-1, 1]$. Define $H_0(\mu) \equiv h_0(\mu+1)$, $H_1(\mu) \equiv h_1(\mu+1)$, $G_0(\mu) \equiv g_0(\mu+1)$ and $G_0(\mu) \equiv g_0(\mu+1)$. Then $h_0(\lambda) = H_0(\lambda-1) = \sum_{k=0}^{K} a_{H0}(k)(\lambda-1)^k$, i.e., a polynomial in the shifted-frequency variable. Now $\mu = \lambda - 1$ corresponds to

$$\tilde{\mathbf{A}} \equiv \mathcal{L} - \mathbf{I} = -\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{5}$$

We shall refer to $-\tilde{\mathbf{A}}$ as to the *normalized adjacency matrix*. A polynomial filter function $H(\mu)$ (in the variable μ) has the corresponding transform matrix given by $H(\tilde{\mathbf{A}})$. This transform matrix $H(\tilde{\mathbf{A}})$ is the same as the transform matrix obtained from the matrix polynomial of the Laplacian, i.e. $H(\tilde{\mathbf{A}}) = h(\mathcal{L})$ where $H(\mu) = h(\mu+1)$. The filter function $H_0(\mu)$ is then partitioned into an *even part* and an *odd* part, which are defined as:

$$H_0^e(\mu) \equiv \tfrac{1}{2}(H_0(\mu) + H_0(-\mu)) \tag{6}$$

$$H_0^o(\mu) \equiv \tfrac{1}{2}(H_0(\mu) - H_0(-\mu)) \tag{7}$$

A similar definition applies for the other filters. Now $H_0^e(\mu)$ and $H_0^o(\mu)$ are polynomials with *only* even and odd powers, respectively. They are even and odd functions respectively, i.e. $H_0^e(\mu) = H_0^e(-\mu)$ and $H_0^o(\mu) = -H_0^o(-\mu)$. The analysis and synthesis *polyphase representation matrices* are defined as

$$\mathbf{P}_a(\mu) = \begin{bmatrix} H_0^e(\mu) & H_0^o(\mu) \\ H_1^o(\mu) & H_1^e(\mu) \end{bmatrix}, \tag{8}$$

$$\mathbf{P}_s(\mu) = \begin{bmatrix} G_0^e(\mu) & G_0^o(\mu) \\ G_1^o(\mu) & G_1^e(\mu) \end{bmatrix}. \tag{9}$$

Note the different ordering in the first and second rows. The following symmetry properties of the matrix can be easily verified:

Property 1:

$$\mathbf{P}_a(\mu) + \mathbf{P}_a(-\mu) = 2\,\mathrm{diag}(H_0^e(\mu), H_1^e(\mu)).$$

Property 2:

$$\mathbf{P}_a(\mu) - \mathbf{P}_a(-\mu) = 2\,\mathbf{I}_a\,\mathrm{diag}(H_0^o(\mu), H_1^o(\mu)).$$

where $\mathbf{I}_a \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is the anti-diagonal unit matrix. The next Lemma and Corollary are proved in [23].

Lemma 2: Equation (3) in Lemma 1 is satisfied if

$$\mathbf{P}_s(\mu) = \mathbf{I}_a\,\mathbf{P}_a(-\mu)\,\mathbf{I}_a \tag{10}$$

Equation (4) in Lemma 1 is satisfied if

$$\det \mathbf{P}_a(\mu) = 1 \tag{11}$$

III. Polyphase Analysis and Structures

The results in section II-C are mainly pertinent for constructing the filter functions. Polyphase analysis deals with the signals and filters in the downsampled domain. Efficient structures in the downsampled domain are derived here. The main results are stated in Theorem 2 and Theorem 3 below.

A. Canonical adjacency and matrix polynomials

For a given bipartite graph $G = (L, H, E)$, without loss of generality suppose the nodes in subgraph L are labelled contiguously so that $L = \{n : n = 1, \ldots, |L|\}$. The complement set is then $H = \{n : n = |L|+1, \ldots, |L|+|H|\}$. The (negative) normalized adjacency matrix can then be written in *canonical form* as

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{|L|} & \mathbf{A}_1 \\ \mathbf{A}_2 & \mathbf{0}_{|H|} \end{bmatrix}. \tag{12}$$

The rectangular matrix $\mathbf{A}_1$ (size $|L| \times |H|$) represents the connections from one subgraph to another. For an undirected graph $\mathbf{A}_2 = \mathbf{A}_1^T$ (size $|H| \times |L|$) but for the development to follow we will assume that $\mathbf{A}_2$ is not neccesarily equal to $\mathbf{A}_1^T$. Any adjacency matrix of a bipartite graph $\tilde{\mathbf{A}}^{nc}$ that is not canonical can be transformed into a canonical form using permutation matrices [27, p. 85].

The next Lemma provides explicit expressions of powers of $\tilde{\mathbf{A}}$, a result that will be important later on.

Lemma 3: For $k = 0, 1, \ldots$

$$\tilde{\mathbf{A}}^{2k} = \begin{bmatrix} (\mathbf{A}_1\mathbf{A}_2)^k & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & (\mathbf{A}_2\mathbf{A}_1)^k \end{bmatrix}, \tag{13}$$

$$\tilde{\mathbf{A}}^{2k+1} = \begin{bmatrix} \mathbf{0}_{|L|} & \mathbf{A}_1(\mathbf{A}_2\mathbf{A}_1)^k \\ \mathbf{A}_2(\mathbf{A}_1\mathbf{A}_2)^k & \mathbf{0}_{|H|} \end{bmatrix}, \tag{14}$$

where the zeroth power of a matrix gives the identity matrix of the same size, and the subscripts (e.g. $|L| \times |H|$) denote the dimension of the submatrices.

Proof: Equation (13) will be proven first using induction. It can be easily verified that

$$\tilde{\mathbf{A}}^2 = \begin{bmatrix} \mathbf{A}_1\mathbf{A}_2 & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & \mathbf{A}_2\mathbf{A}_1 \end{bmatrix}.$$

This is (13) with $k = 1$. Suppose (13) applies for $k-1$, i.e.

$$\tilde{\mathbf{A}}^{2(k-1)} = \begin{bmatrix} (\mathbf{A}_1\mathbf{A}_2)^{k-1} & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & (\mathbf{A}_2\mathbf{A}_1)^{k-1} \end{bmatrix}.$$

Then by explicit multiplication of the two matrices above

$$\tilde{\mathbf{A}}^{2k} = \tilde{\mathbf{A}}^2\tilde{\mathbf{A}}^{2(k-1)} = \begin{bmatrix} (\mathbf{A}_1\mathbf{A}_2)^{k} & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & (\mathbf{A}_2\mathbf{A}_1)^{k} \end{bmatrix}.$$

By using induction from $k = 1$, equation (13) applies for any $k \geq 1$. Explicit multiplication of equations (12) and (13) gives

$$\tilde{\mathbf{A}}^{2k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^{2k} = \begin{bmatrix} \mathbf{0}_{|L|} & \mathbf{A}_1(\mathbf{A}_2\mathbf{A}_1)^{k} \\ \mathbf{A}_2(\mathbf{A}_1\mathbf{A}_2)^{k} & \mathbf{0}_{|H|} \end{bmatrix}.$$

■

Lemma 3 shows that *even and odd powers* of $\tilde{\mathbf{A}}$ result in *block diagonal matrices and block anti-diagonal matrices*, respectively.

Consider the generic polynomial

$$F(\mu) \equiv \sum_{k\geq 0} \hat{f}(k)\mu^k$$

which can represent any of the spectral filters H_0, H_1 G_0 or G_1. The even and odd parts are then given by:

$$\begin{aligned} F^e(\mu) &= \sum_{k\geq 0} \hat{f}(2k)\mu^{2k}, \\ F^o(\mu) &= \sum_{k\geq 0} \hat{f}(k+1)\mu^{2k+1}. \end{aligned}$$

The next Lemma shows explicitly the structure of the matrix polynomial $F(\tilde{\mathbf{A}})$ in terms of the even and odd parts. The result will be used later to derive the polyphase structures.

Lemma 4: The matrix polynomial $F(\tilde{\mathbf{A}}) \equiv \sum_{k\geq 0} \hat{f}(k)(\tilde{\mathbf{A}})^k$ can be expressed as

$$F(\tilde{\mathbf{A}}) = F^e(\tilde{\mathbf{A}}) + F^o(\tilde{\mathbf{A}})$$

where

$$F^e(\tilde{\mathbf{A}}) = \begin{bmatrix} F^{e,u}(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & F^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \tag{15}$$

$$F^o(\tilde{\mathbf{A}}) = \begin{bmatrix} \mathbf{0}_{|L|} & F^{o,u}(\tilde{\mathbf{A}}) \\ F^{o,l}(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{bmatrix} \tag{16}$$

and the submatrices are defined as

$$F^{e,u}(\tilde{\mathbf{A}}) \equiv \sum_{k\geq 0} \hat{f}(2k)(\mathbf{A}_1\mathbf{A}_2)^k \tag{17}$$

$$F^{e,l}(\tilde{\mathbf{A}}) \equiv \sum_{k\geq 0} \hat{f}(2k)(\mathbf{A}_2\mathbf{A}_1)^k \tag{18}$$

$$F^{o,u}(\tilde{\mathbf{A}}) \equiv \sum_{k\geq 0} \hat{f}(2k+1)\mathbf{A}_1(\mathbf{A}_2\mathbf{A}_1)^k \tag{19}$$

$$F^{o,l}(\tilde{\mathbf{A}}) \equiv \sum_{k\geq 0} \hat{f}(2k+1)\mathbf{A}_2(\mathbf{A}_1\mathbf{A}_2)^k \tag{20}$$

and $\mathbf{A}_1$ and $\mathbf{A}_2$ are submatrices in (12).

Proof: By definition $F^e(\tilde{\mathbf{A}}) = \sum_k \hat{f}(2k)\ (\tilde{\mathbf{A}})^{2k}$. Using Lemma 3 for $(\tilde{\mathbf{A}})^{2k}$

$$F^e(\tilde{\mathbf{A}}) = \sum_k \hat{f}_0(2k) \begin{bmatrix} (\mathbf{A}_1\mathbf{A}_2)^{k} & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & (\mathbf{A}_2\mathbf{A}_1)^{k} \end{bmatrix}.$$

and the result for $F^e(\tilde{\mathbf{A}})$ follows. By definition $F^o(\tilde{\mathbf{A}}) = \sum_k \hat{f}(2k+1)\ (\tilde{\mathbf{A}})^{2k+1}$. Using Lemma 3 for $(\tilde{\mathbf{A}})^{2k+1}$

$$F^o(\tilde{\mathbf{A}}) = \sum_k \hat{f}_0(2k+1) \begin{bmatrix} \mathbf{0}_{|L|} & \mathbf{A}_1(\mathbf{A}_2\mathbf{A}_1)^{k} \\ \mathbf{A}_2(\mathbf{A}_1\mathbf{A}_2)^{k} & \mathbf{0}_{|H|} \end{bmatrix}.$$

and the result for $F^o(\tilde{\mathbf{A}})$ follows. ■

Note that $F^e(\tilde{\mathbf{A}})$ is *block diagonal* whereas $F^o(\tilde{\mathbf{A}})$ is *block anti-diagonal*. Furthermore the submatrices $F^{e,u}(\tilde{\mathbf{A}})$ and $F^{e,l}(\tilde{\mathbf{A}})$ are square but the submatrices $F^{o,u}(\tilde{\mathbf{A}})$ and $F^{o,l}(\tilde{\mathbf{A}})$ are rectangular in general. The superscripts u and l are used to denote upper and lower respectively.

B. Analysis Polyphase Structure

The analysis filter bank is shown in Fig. 1. The signals, e.g. $\mathbf{f}$, are to be treated as column vectors. Signals before the downsampler are of dimension $N = |L| + |H|$ and signals after the downsampler are of dimension $|L|$ or $|H|$. This is analogous to the full sampling rate and downsampled rate in 1-D multirate systems. The rectangular blocks, e.g. $H_0(\tilde{\mathbf{A}})$ are to be treated as transform matrices and are matrix polynomials of $\tilde{\mathbf{A}}$. Consider now Fig. 2 which shows the bipartite decomposition of a full rate signal $\mathbf{f}$ into two lower rate signals $\mathbf{f}_L$ and $\mathbf{f}_H$. Assuming the contiguous ordering (labelling) of the nodes of the bipartite graph as described in section III-A, the downsampling operation can be expressed compactly as:

$$\mathbf{f}_L = \mathbf{B}_L\mathbf{f}, \qquad \mathbf{f}_H = \mathbf{B}_H\mathbf{f}, \tag{21}$$

where

$$\mathbf{B}_L \equiv \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{0}_{|L|\times|H|} \end{bmatrix}, \ \mathbf{B}_H \equiv \begin{bmatrix} \mathbf{0}_{|H|\times|L|} & \mathbf{I}_{|H|} \end{bmatrix} \tag{22}$$

and $\mathbf{I}_{|L|}$ ($\mathbf{I}_{|H|}$) is the size $|L|$ ($|H|$) identity matrix. Concatenating the column vectors $\mathbf{f}_L$ with $\mathbf{f}_H$ gives $\mathbf{f}$ as

$$\begin{bmatrix} \mathbf{f}_L \\ \mathbf{f}_H \end{bmatrix} = \begin{bmatrix} \mathbf{B}_L\mathbf{f} \\ \mathbf{B}_H\mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & \mathbf{I}_{|H|} \end{bmatrix}\mathbf{f} = \mathbf{f}$$

The next theorem formally gives the equivalent downsampled filtering for the analysis filter bank.

Theorem 2: The filtered and downsampled signals in Fig. 1 are given by

$$\mathbf{f}_{sub} \equiv \begin{bmatrix} \mathbf{f}_{low} \\ \mathbf{f}_{high} \end{bmatrix} = \mathbf{T}_A \begin{bmatrix} \mathbf{f}_L \\ \mathbf{f}_H \end{bmatrix} \equiv \mathbf{T}_A\mathbf{f}$$

where $\mathbf{f}_L$ and $\mathbf{f}_H$ are defined in (21). The *analysis polyphase transform matrix* (size $N \times N$) is defined as

$$\mathbf{T}_A \equiv \begin{bmatrix} H_0^{e,u}(\tilde{\mathbf{A}}) & H_0^{o,u}(\tilde{\mathbf{A}}) \\ H_1^{o,l}(\tilde{\mathbf{A}}) & H_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix}. \tag{23}$$

The submatrices

1) $H_0^{e,u}(\tilde{\mathbf{A}})$ and $H_0^{o,u}(\tilde{\mathbf{A}})$ are as defined in Lemma 4 but with F replaced with H_0 and $\hat{f}(k)$ replaced with $\hat{h}_0(k)$ (coefficients of spectral filter $H_0(\mu)$).
2) $H_1^{e,l}(\tilde{\mathbf{A}})$ and $H_1^{o,l}(\tilde{\mathbf{A}})$ are as defined Lemma 4 but with F replaced with H_1 and $\hat{f}(k)$ replaced with $\hat{h}_1(k)$ (coefficients of spectral filter $H_1(\mu)$).

Proof: With reference to Fig. 1, $\mathbf{f}_0 = H_0(\tilde{\mathbf{A}})\mathbf{f}$ and $\mathbf{f}_{low} = \mathbf{B}_L\mathbf{f}_0$ using (21). Therefore

$$\mathbf{f}_{low} = \mathbf{B}_L H_0(\tilde{\mathbf{A}})\mathbf{f} = \mathbf{B}_L(H_0^e(\tilde{\mathbf{A}}) + H_0^o(\tilde{\mathbf{A}}))\mathbf{f}$$

Using Lemma 4 we have

$$\mathbf{B}_L H_0^e(\tilde{\mathbf{A}})\mathbf{f} = \left[\begin{array}{cc} \mathbf{I}_{|L|} & \mathbf{0}_{|L|\times|H|} \end{array}\right] \cdot \left[\begin{array}{cc} H_0^{e,u}(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & H_0^{e,l}(\tilde{\mathbf{A}}) \end{array}\right] \left[\begin{array}{c} \mathbf{f}_L \\ \mathbf{f}_H \end{array}\right] = H_0^{e,u}(\tilde{\mathbf{A}})\mathbf{f}_L$$

$$\mathbf{B}_L H_0^o(\tilde{\mathbf{A}})\mathbf{f} = \left[\begin{array}{cc} \mathbf{I}_{|L|} & \mathbf{0}_{|L|\times|H|} \end{array}\right] \cdot \left[\begin{array}{cc} \mathbf{0}_{|L|} & H_0^{o,u}(\tilde{\mathbf{A}}) \\ H_0^{o,l}(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{array}\right] \left[\begin{array}{c} \mathbf{f}_L \\ \mathbf{f}_H \end{array}\right] = H_0^{o,u}(\tilde{\mathbf{A}})\mathbf{f}_H$$

Adding the two result above gives

$$\mathbf{f}_{low} = H_0^{e,u}(\tilde{\mathbf{A}})\mathbf{f}_L + H_0^{o,u}(\tilde{\mathbf{A}})\mathbf{f}_H \tag{24}$$

Using Fig. 1 and similar steps to the derivation for $\mathbf{f}_{low}$, the high-pass signal is given by

$$\mathbf{f}_{high} = H_1^{o,l}(\tilde{\mathbf{A}})\mathbf{f}_L + H_1^{e,l}(\tilde{\mathbf{A}})\mathbf{f}_H \tag{25}$$

■

Comment: For 1-D regular domain multirate systems, the noble identity allows one to move the filtering operation from a higher sampling rate to a lower one and vice versa [9], [10]. The noble identity can be exploited to derive efficient implementation of 1-D filter bank systems. The result above is analogous to the 1-D polyphase structure where the filtering is performed on the downsampled signals, i.e. first downsample, then filter. The approach by Teke and Vaidyanathan [20], [21] is to mimic the behaviour of delays in 1-D using the shift operator. In particular, signals in [20], [21] are downsampled using a single set of nodes. As an example, in a "lazy" filterbank, the original signal is sampled on those nodes, then shifted by multiplying it by the shift operator (the adjacency matrix) and sampled again in those same nodes [20]. In contrast, we sample the signal in two disjoint subsets (see (22)) with no shifts involved, and our approach is applicable to any bipartite graph.

C. Synthesis Polyphase Structure

The synthesis filter bank is shown in Fig. 3. The signals are vectors with dimension of either $|L|$, $|H|$ or $|L|+|H|$. The blocks are transform matrices and are matrix polynomial of $\tilde{\mathbf{A}}$. Contiguous ordering (labelling) of the nodes of the bipartite graph as described in section III-A is assumed. The upsampling operations can then be described compactly as

$$(\mathbf{f}_{low})_{\uparrow\beta_L} = \left[\begin{array}{c} \mathbf{f}_{low} \\ \mathbf{0}_{|H|\times 1} \end{array}\right], \qquad (\mathbf{f}_{high})_{\uparrow\beta_H} = \left[\begin{array}{c} \mathbf{0}_{|L|\times 1} \\ \mathbf{f}_{high} \end{array}\right] \tag{26}$$

Note that the reverse of the bipartite decomposition in Fig. 2 is given by

$$(\mathbf{f}_L)_{\uparrow\beta_L} + (\mathbf{f}_H)_{\uparrow\beta_H} = \left[\begin{array}{c} \mathbf{f}_L \\ \mathbf{f}_H \end{array}\right] = \mathbf{f}$$

The next Lemma gives the equivalent filtering for the synthesis filter bank.

Lemma 5: The interpolated (upsampled and filtered) signals in Fig. 3 are given by

$$\mathbf{g}_0 = \left[\begin{array}{c} G_0^{e,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) \end{array}\right]\mathbf{f}_{low}, \qquad \mathbf{g}_1 = \left[\begin{array}{c} G_1^{o,u}(\tilde{\mathbf{A}}) \\ G_1^{e,l}(\tilde{\mathbf{A}}) \end{array}\right]\mathbf{f}_{high} \tag{27}$$

where

1) $G_0^{e,u}(\tilde{\mathbf{A}})$ and $G_0^{o,l}(\tilde{\mathbf{A}})$ are as defined in Lemma 4 but with F replaced with G_0 and $\hat{f}(k)$ replaced with $\hat{g}_0(k)$ (coefficients of spectral filter $G_0(\mu)$).
2) $G_1^{o,u}(\tilde{\mathbf{A}})$ and $G_1^{e,l}(\tilde{\mathbf{A}})$ are as defined in Lemma 4 but with F replaced with G_1 and $\hat{f}(k)$ replaced with $\hat{g}_1(k)$ (coefficients of spectral filter $G_1(\mu)$).

Proof: With reference to Fig. 3 and using (26)

$$\mathbf{g}_0 = G_0(\tilde{\mathbf{A}})(\mathbf{f}_{low})_{\uparrow\beta_L} = (G_0^e(\tilde{\mathbf{A}}) + G_0^o(\tilde{\mathbf{A}}))\left[\begin{array}{c} \mathbf{f}_{low} \\ \mathbf{0}_{|H|\times 1} \end{array}\right]$$

Using Lemma 4 we have

$$G_0^e(\tilde{\mathbf{A}})\left[\begin{array}{c} \mathbf{f}_{low} \\ \mathbf{0}_{|H|\times 1} \end{array}\right] = \left[\begin{array}{cc} G_0^{e,u}(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & G_0^{e,l}(\tilde{\mathbf{A}}) \end{array}\right]\left[\begin{array}{c} \mathbf{f}_{low} \\ \mathbf{0}_{|H|\times 1} \end{array}\right] = \left[\begin{array}{c} G_0^{e,u}(\tilde{\mathbf{A}})\mathbf{f}_{low} \\ \mathbf{0}_{|H|\times 1} \end{array}\right]$$

$$G_0^o(\tilde{\mathbf{A}})\left[\begin{array}{c} \mathbf{f}_{low} \\ \mathbf{0}_{|H|\times 1} \end{array}\right] = \left[\begin{array}{cc} \mathbf{0}_{|L|} & G_0^{o,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{array}\right]\left[\begin{array}{c} \mathbf{f}_{low} \\ \mathbf{0}_{|H|\times 1} \end{array}\right] = \left[\begin{array}{c} \mathbf{0}_{|L|\times 1} \\ G_0^{o,l}(\tilde{\mathbf{A}})\mathbf{f}_{low} \end{array}\right]$$

Adding the two expressions above gives the result for $\mathbf{g}_0$. With reference to Fig. 3 and using (26)

$$\mathbf{g}_1 = G_1(\tilde{\mathbf{A}})(\mathbf{f}_{high})_{\uparrow\beta_H} = (G_1^e(\tilde{\mathbf{A}}) + G_1^o(\tilde{\mathbf{A}}))\left[\begin{array}{c} \mathbf{0}_{|L|\times 1} \\ \mathbf{f}_{high} \end{array}\right]$$

Using similar steps to the derivation for $\mathbf{g}_0$, the result for $\mathbf{g}_1$ in (27) follows. ■

The reconstructed signal is $\mathbf{f}^R = \mathbf{g}_0 + \mathbf{g}_1$. The bipartite (polyphase) decomposition of $\mathbf{f}^R$, analogous to (21), is defined as

$$\mathbf{f}_L^R = \mathbf{B}_L\mathbf{f}^R, \qquad \mathbf{f}_H^R = \mathbf{B}_H\mathbf{f}^R \tag{28}$$

Concatenating $\mathbf{f}_L^R$ with $\mathbf{f}_H^R$, which is equivalent to reversing the decomposition, gives $\mathbf{f}^R$ i.e.

$$\left[\begin{array}{c} \mathbf{f}_L^R \\ \mathbf{f}_H^R \end{array}\right] = (\mathbf{f}_L^R)_{\uparrow\beta_L} + (\mathbf{f}_H^R)_{\uparrow\beta_H} = \mathbf{f}^R.$$

The next theorem gives explicit expressions for $\mathbf{f}_L^R$ and $\mathbf{f}_H^R$.

Theorem 3: The bipartite (polyphase) decomposition components of $\mathbf{f}^R$ as defined in (28) are given by

$$\mathbf{f}^R \equiv \left[\begin{array}{c} \mathbf{f}_L^R \\ \mathbf{f}_H^R \end{array}\right] = \mathbf{T}_S\left[\begin{array}{c} \mathbf{f}_{low} \\ \mathbf{f}_{high} \end{array}\right] = \mathbf{T}_S\mathbf{f}_{sub}$$

where the *synthesis polyphase transform matrix* (size $N \times N$) is defined as

$$\mathbf{T}_S \equiv \begin{bmatrix} G_0^{e,u}(\tilde{\mathbf{A}}) & G_1^{o,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) & G_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \tag{29}$$

and the submatrices $G_0^{e,u}(\tilde{\mathbf{A}})$, etc. are defined in Lemma 5.

Proof: Now $\mathbf{f}^R = (\mathbf{g}_0 + \mathbf{g}_1)$. Using Lemma 5

$$\mathbf{f}^R = \begin{bmatrix} G_0^{e,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) \end{bmatrix} \mathbf{f}_{low} + \begin{bmatrix} G_1^{o,u}(\tilde{\mathbf{A}}) \\ G_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \mathbf{f}_{high}$$

$$= \begin{bmatrix} G_0^{e,u}(\tilde{\mathbf{A}}) & G_1^{o,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) & G_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \begin{bmatrix} \mathbf{f}_{low} \\ \mathbf{f}_{high} \end{bmatrix}$$

Using (28) for $\mathbf{f}_L^R$

$$\mathbf{f}_L^R = \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{0}_{|L|\times|H|} \end{bmatrix} \begin{bmatrix} G_0^{e,u}(\tilde{\mathbf{A}}) & G_1^{o,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) & G_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \begin{bmatrix} \mathbf{f}_{low} \\ \mathbf{f}_{high} \end{bmatrix}$$

$$= \begin{bmatrix} G_0^{e,u}(\tilde{\mathbf{A}}) & G_1^{o,u}(\tilde{\mathbf{A}}) \end{bmatrix} \begin{bmatrix} \mathbf{f}_{low} \\ \mathbf{f}_{high} \end{bmatrix}$$

$$\mathbf{f}_L^R = G_0^{e,u}(\tilde{\mathbf{A}})\mathbf{f}_{low} + G_1^{o,u}(\tilde{\mathbf{A}})\mathbf{f}_{high} \tag{30}$$

Using (28) for $\mathbf{f}_H^R$

$$\mathbf{f}_H^R = \begin{bmatrix} \mathbf{0}_{|H|\times|L|} & \mathbf{I}_{|H|} \end{bmatrix} \begin{bmatrix} G_0^{e,u}(\tilde{\mathbf{A}}) & G_1^{o,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) & G_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \begin{bmatrix} \mathbf{f}_{low} \\ \mathbf{f}_{high} \end{bmatrix}$$

$$= \begin{bmatrix} G_0^{o,l}(\tilde{\mathbf{A}}) & G_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \begin{bmatrix} \mathbf{f}_{low} \\ \mathbf{f}_{high} \end{bmatrix}$$

$$\mathbf{f}_H^R = G_0^{o,l}(\tilde{\mathbf{A}})\mathbf{f}_{low} + G_1^{e,l}(\tilde{\mathbf{A}})\mathbf{f}_{high} \tag{31}$$

■

To summarize: Theorem 2 gives the analysis polyphase structure as shown in Fig. 4. Theorem 3 gives the synthesis polyphase structure as shown in Fig. 5. These results are built on top of Lemmas 3 and 4 which hold for bipartite graphs. Implementing the full-rate filter bank in Fig. 1 requires the multiplication of the two transform matrices $H_0(\tilde{\mathbf{A}})$ and $H_1(\tilde{\mathbf{A}})$ (dimension $N \times N$) with $\mathbf{f}$ (dimension N). Implementing the polyphase structure in Fig. 4 requires the multiplication of the transform matrices $\mathbf{T}_A$ (dimension $N \times N$) with $\mathbf{f}$. We therefore have a computational complexity reduction by a factor of two which is the same as in 1-D systems.

IV. Equivalent Subgraph Signals and Filters

Results are derived in this section that allow a spectral interpretation of the polyphase analysis of the previous section. The explicit expressions of the submatrices/subfilters in (23) and (29) can be obtained by using (17) - (20), e.g. $H_0^{e,u}(\tilde{\mathbf{A}})$ is given by (17) with $\hat{h}_0(2k)$ replacing $\hat{f}(2k)$. The subfilters are functions of $\tilde{\mathbf{A}}$ but are not simple polynomials of $\tilde{\mathbf{A}}$. Define the following matrices:

$$\mathbf{A}_\alpha \equiv \mathbf{A}_1\mathbf{A}_2 \qquad \mathbf{A}_\beta \equiv \mathbf{A}_2\mathbf{A}_1$$

If the original graph (adjacency $\tilde{\mathbf{A}}$) is undirected ($\mathbf{A}_2 = \mathbf{A}_1^T$), then $\mathbf{A}_\alpha \equiv \mathbf{A}_1\mathbf{A}_1^T$ and $\mathbf{A}_\beta \equiv \mathbf{A}_1^T\mathbf{A}_1$. Note that in this case, both $\mathbf{A}_\alpha$ and $\mathbf{A}_\beta$ are symmetric but $\mathbf{A}_\alpha \neq \mathbf{A}_\beta$ in general. The subfilters in (17) - (18) can then be expressed as:

$$F^{e,u}(\tilde{\mathbf{A}}) = \hat{F}^{e,u}(\mathbf{A}_\alpha) \equiv \sum_{k\geq 0} \hat{f}(2k)(\mathbf{A}_\alpha)^k$$

$$F^{e,l}(\tilde{\mathbf{A}}) = \hat{F}^{e,l}(\mathbf{A}_\beta) \equiv \sum_{k\geq 0} \hat{f}(2k)(\mathbf{A}_\beta)^k$$

$$F^{o,u}(\tilde{\mathbf{A}}) = \mathbf{A}_1\hat{F}^{o,u}(\mathbf{A}_\beta) \equiv \mathbf{A}_1 \sum_{k\geq 0} \hat{f}(2k+1)(\mathbf{A}_\beta)^k$$

$$F^{o,l}(\tilde{\mathbf{A}}) = \mathbf{A}_2\hat{F}^{o,l}(\mathbf{A}_\alpha) \equiv \mathbf{A}_2 \sum_{k\geq 0} \hat{f}(2k+1)(\mathbf{A}_\alpha)^k$$

Let G_α and G_β be subgraphs with adjacency matrix $\mathbf{A}_\alpha$ and $\mathbf{A}_\beta$ respectively. The vertices of G_α and G_β are the subsets L and H respectively (L and H are bipartitions from the original graph). The subfilters $\hat{F}^{e,u}(\mathbf{A}_\alpha)$ and $\hat{F}^{o,l}(\mathbf{A}_\alpha)$ are polynomials of $\mathbf{A}_\alpha$ and therefore can be viewed as spectral filters w.r.t. to the subgraph G_α. The subfilters $\hat{F}^{e,l}(\mathbf{A}_\beta)$ and $\hat{F}^{o,u}(\mathbf{A}_\beta)$ are polynomials of $\mathbf{A}_\beta$ and therefore can be viewed as spectral filters w.r.t. to the subgraph G_β. The subfilter $F^{o,u}(\tilde{\mathbf{A}})$ ($F^{o,l}(\tilde{\mathbf{A}})$) consists of spectral filtering with $\hat{F}^{o,u}(\mathbf{A}_\beta)$ ($\hat{F}^{o,l}(\mathbf{A}_\alpha)$) followed by the projection operator $\mathbf{A}_1$ ($\mathbf{A}_2$). Note that even though the original bipartite graph does not have any self-loops (zero values along the diagonal of $\tilde{\mathbf{A}}$), the resulting subgraphs G_α and G_β may have self loops (non-zero diagonals for $\mathbf{A}_\alpha$ and $\mathbf{A}_\beta$).

Since we are now considering signals on multiple different graphs, we introduce a notation that makes it explicit which graph a signal lives. Thus $(\mathbf{f}, -\tilde{\mathbf{A}})$ represents the input signal, where $\mathbf{f}$ is the signal vector and $-\tilde{\mathbf{A}}$ the adjacency matrix of the graph. The signals from the bipartite decomposition in Fig. 2 can therefore be represented as $(\mathbf{f}_L, \mathbf{A}_\alpha)$ and $(\mathbf{f}_H, \mathbf{A}_\beta)$, i.e. signals on equivalent subgraphs. The polyphase structure in Fig. 4 has therefore an equivalent structure shown in Fig. 6. The signal $(\mathbf{f}_L, \mathbf{A}_\alpha)$ is filtered with the spectral filters $\hat{H}^{e,u}(\mathbf{A}_\alpha)$ and $\hat{H}^{o,l}(\mathbf{A}_\alpha)$. The output from $\hat{H}^{o,l}(\mathbf{A}_\alpha)$ is then applied to the projection operator $\mathbf{A}_2$ which maps a signal on subgraph G_α to a signal on subgraph G_β. Something similar occurs for the signal $(\mathbf{f}_H, \mathbf{A}_\beta)$. An equivalent structure to Fig. 5, for the synthesis side, showing the spectral filtering process and projection operators, can similarly be obtained.

The next Lemma gives the relationship between the eigenvalues/eigenvector of the bipartite original graph (adjacency $-\tilde{\mathbf{A}}$) and the subgraphs (adjacencies $\mathbf{A}_\alpha$ and $\mathbf{A}_\beta$).

Lemma 6: Let $\tilde{\mathbf{A}}\mathbf{y} = \mu\mathbf{y}$ where μ and $\mathbf{y}$ are the eigenvalues and eigenvector respectively. The eigenvector can be partitioned into $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T \\ \mathbf{y}_2^T \end{bmatrix}^T$, where $\mathbf{y}_1$ and $\mathbf{y}_2$ are associated with set of vertices in L and H respectively (bipartition). Then

1) The eigenvalues of $\mathbf{A}_\alpha$ are μ^2 with eigenvector $\mathbf{y}_1$.
2) The eigenvalues of $\mathbf{A}_\beta$ are μ^2 with eigenvector $\mathbf{y}_2$

Proof: By definition

$$\tilde{\mathbf{A}}\mathbf{y} = \begin{bmatrix} \mathbf{0}_{|L|} & \mathbf{A}_1 \\ \mathbf{A}_2 & \mathbf{0}_{|H|} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \mu \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$$

Taking each component separately gives

$$\mathbf{A}_1\mathbf{y}_2 = \mu\mathbf{y}_1, \qquad \mathbf{A}_2\mathbf{y}_1 = \mu\mathbf{y}_2$$

Using these relations, we get

$$\mathbf{A}_\alpha \mathbf{y}_1 = \mathbf{A}_1\mathbf{A}_2\mathbf{y}_1 = \mu\mathbf{A}_1\mathbf{y}_2 = \mu^2\mathbf{y}_1$$
$$\mathbf{A}_\beta \mathbf{y}_2 = \mathbf{A}_2\mathbf{A}_1\mathbf{y}_2 = \mu\mathbf{A}_2\mathbf{y}_1 = \mu^2\mathbf{y}_2$$

■

If $\tilde{\mathbf{A}}$ is normalized as in (5), then by definition $-\tilde{\mathbf{A}}\mathbf{1}_N = \mathbf{1}_N$, where $\mathbf{1}_N$ is the dimension N row vector with unity elements. Using the partitioned form of $\tilde{\mathbf{A}}$ in (12), it can be easily shown that $\mathbf{A}_1\mathbf{1}_H = -\mathbf{1}_L$ and $\mathbf{A}_2\mathbf{1}_L = -\mathbf{1}_H$. Using these results we have $\mathbf{A}_\alpha\mathbf{1}_L = \mathbf{A}_1\mathbf{A}_2\mathbf{1}_L = \mathbf{1}_L$ and $\mathbf{A}_\beta\mathbf{1}_H = \mathbf{A}_2\mathbf{A}_1\mathbf{1}_H = \mathbf{1}_H$, i.e. the adjacency matrices of the subgraphs are also normalized. Note that there is no negative sign in the normalization for $\mathbf{A}_\alpha$ and $\mathbf{A}_\beta$, i.e. they are the positive version of the adjacency matrix. The corresponding Laplacians are therefore

$$\mathcal{L}_\alpha = \mathbf{I}_L - \mathbf{A}_\alpha, \qquad \mathcal{L}_\beta = \mathbf{I}_H - \mathbf{A}_\beta$$

Let λ denote the eigenvalue of the original graph Laplacian $\mathcal{L} = \mathbf{I}_H + \tilde{\mathbf{A}}$ (from (5)). Then $\mu = \lambda - 1$ (μ eigenvalue of $\tilde{\mathbf{A}}$). Since the eigenvalues of $\mathbf{A}_\alpha$ and $\mathbf{A}_\beta$ are the same and equal to μ^2 by Lemma 6, the eigenvalues of $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$ are also the same and are given by:

$$\lambda_{\alpha,\beta} = 1 - \mu^2 = 1 - (\lambda - 1)^2 = 2\lambda - \lambda^2$$

This equation gives the mapping of the spectral frequencies from the upsampled domain to the downsampled domain. The next Lemma gives explicit relationships between the Laplacians in both domains.

Lemma 7: The subgraphs Laplacian $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$ are related to the original bipartite graph Laplacian $\mathcal{L}$ as follows:

$$\mathcal{L}_\alpha = \mathbf{B}_L(\mathcal{L}\mathbf{Q})^2\mathbf{B}_L^T, \qquad \mathcal{L}_\beta = \mathbf{B}_H(\mathcal{L}\mathbf{Q})^2\mathbf{B}_H^T$$

where $\mathbf{B}_L$ and $\mathbf{B}_H$ are defined in (22) and $\mathbf{Q} \equiv \begin{bmatrix} \mathbf{B}_L \\ -\mathbf{B}_H \end{bmatrix}$.

Proof: Now $\mathcal{L} = \mathbf{I}_N + \tilde{\mathbf{A}}$ from (5). Using (12) for $\tilde{\mathbf{A}}$

$$\mathcal{L}\mathbf{Q} = \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{A}_1 \\ \mathbf{A}_2 & \mathbf{I}_{|H|} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & -\mathbf{I}_{|H|} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{A}_1 \\ -\mathbf{A}_2 & -\mathbf{I}_{|H|} \end{bmatrix}$$

$$(\mathcal{L}\mathbf{Q})^2 = \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{A}_1 \\ -\mathbf{A}_2 & -\mathbf{I}_{|H|} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{A}_1 \\ -\mathbf{A}_2 & -\mathbf{I}_{|H|} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{|L|} - \mathbf{A}_1\mathbf{A}_2 & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & \mathbf{I}_{|H|} - \mathbf{A}_2\mathbf{A}_1 \end{bmatrix}$$

The diagonal blocks of $(\mathcal{L}\mathbf{Q})^2$ give $\mathcal{L}_\alpha$ and $\mathcal{L}_\beta$. These blocks can be extracted from $(\mathcal{L}\mathbf{Q})^2$ by pre-multiplication with $\mathbf{B}_L$ (or $\mathbf{B}_H$) and post-multiplication with $\mathbf{B}_L^T$ (or $\mathbf{B}_H^T$). ■

Comment: As shown in [1] the bipartite graph filter bank can be applied to any arbitrary graph using the notion of separable processing. This would require the decomposition of an arbitrary graph into a sequence of bipartite graphs. The decomposition however is not-unique and developing ways to give a good decomposition is still an open area of research, see for example [28]. The results above provide an accurate characterization of transformed signals in terms of the equivalent subgraphs and can potentially be used to develop improved multirate decomposition techniques. This is beyond the scope of this paper and is left for future work.

V. Vertex and Spectral Relationships

For a given representation matrix $\mathbf{P}_a(\mu)$ (dimension 2×2) defined in (8), there is a corresponding transform matrix $\mathbf{T}_A$ (dimension $2N \times 2N$) which can be obtained by using Lemma 4 and Theorem 2. The former can be viewed as a spectral domain description and the later can be viewed as a vertex domain description, as will be justified next by comparisons with 1-D systems.

A. Analogy with 1-D systems

The bipartite graph filter bank (GFB) considered in this work is analogous to the classical 1-D two-channel multirate filter. In section 3.2 in [29] three types of domains are used for the analysis/description of the 1-D filter bank: time-domain, modulation-domain and polyphase domain. In the time-domain the analysis filter bank is described by the analysis matrix $\mathbf{T}_a^{1D}$ which is a matrix with a block Toeplitz structure (see (3.2.2), page 109 in [29]). In the polyphase-domain the analysis filter bank is described by the analysis polyphase matrix $\mathbf{H}_p^{1D}(z)$ which is a 2×2 matrix of Laurent-polynomials in the variable $z = e^{j\omega}$ (see (3.2.22), page 114 in [29]). The polyphase-domain can also be considered as a frequency-domain description as it involves the frequency variable ω. For the GFB the transform matrix $\mathbf{T}_A$ is analogous to the analysis matrix $\mathbf{T}_a^{1D}$. Both $\mathbf{T}_A$ and $\mathbf{T}_a^{1D}$ can be used to transform the signal by matrix/vector multiplication. The matrix $\mathbf{T}_A$ can therefore be considered a vertex domain description as it allows a vertex domain implementation. The $\mathbf{P}_a(\mu)$ in GFB is analogous to $\mathbf{H}_p^{1D}(z)$ as both involve frequency variables. The analogy however are not without differences. The ordering of the even and odd components in $\mathbf{P}_a(\mu)$ is different to that in $\mathbf{H}_p^{1D}(z)$. The variable $\mu = \lambda - 1$ is real-valued but the variable z is complex-valued. The analogy and comparison are summarized in Table I. Similar analogies exist for the synthesis filter bank.

B. Perfect Reconstruction

The analysis in sections III and IV makes no assumption about any constraint imposed on the spectral filters or the elements of the representation matrices in (8) and (9). The perfect reconstruction (PR) condition in terms of the spectral filters is given by Lemma 1, while the corresponding condition in terms of the representation matrices is given by Lemma 2. Equivalent conditions in terms on the transform matrices (23) and (29) are derived next.

Lemma 8: If the synthesis representation matrix $\mathbf{P}_s(\mu)$ is given by (10) in Lemma 2, then the synthesis transform matrix is given by

$$\mathbf{T}_S \equiv \begin{bmatrix} G_0^{e,u}(\tilde{\mathbf{A}}) & G_1^{o,u}(\tilde{\mathbf{A}}) \\ G_0^{o,l}(\tilde{\mathbf{A}}) & G_1^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} = \begin{bmatrix} H_1^{e,u}(\tilde{\mathbf{A}}) & -H_0^{o,u}(\tilde{\mathbf{A}}) \\ -H_1^{o,l}(\tilde{\mathbf{A}}) & H_0^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix} \tag{32}$$

Proof: Using (8) and (9) in (10) gives the following relation between the element of the representation matrices

$$\begin{bmatrix} G_0^e(\mu) & G_0^o(\mu) \\ G_1^o(\mu) & G_1^e(\mu) \end{bmatrix} = \begin{bmatrix} H_1^e(\mu) & -H_1^o(\mu) \\ -H_0^o(\mu) & H_o^e(\mu) \end{bmatrix}$$

By substituting μ with $\tilde{\mathbf{A}}$ we obtain 4 matrix equations, e.g. $G_0^e(\tilde{\mathbf{A}}) = H_1^e(\tilde{\mathbf{A}})$. The matrices in each equation have the block-form in either (15) or (16). Taking the appropriate block will give the required submatrix in (32), e.g. block $(1,1)$ from $G_0^e(\tilde{\mathbf{A}}) = H_1^e(\tilde{\mathbf{A}})$ gives $G_0^{e,u}(\tilde{\mathbf{A}}) = H_1^{e,u}(\tilde{\mathbf{A}})$. ■

The next Lemma gives the equivalent PR condition in terms of the transform matrices.

Lemma 9: If (10) and (11) in Lemma 2 are satisfied then the transform matrices satisfy

$$\mathbf{T}_S \mathbf{T}_A = \mathbf{I}_N \tag{33}$$

Proof: Using Lemma 8 the product $\mathbf{T}_S\mathbf{T}_A$ is given explicitly by

$$\mathbf{T}_S\mathbf{T}_A = \begin{bmatrix} H_1^{e,u} & -H_0^{o,u} \\ -H_1^{o,l} & H_0^{e,l} \end{bmatrix} \begin{bmatrix} H_0^{e,u} & H_0^{o,u} \\ H_1^{o,l} & H_1^{e,l} \end{bmatrix}$$

$$= \begin{bmatrix} H_1^{e,u}H_0^{e,u} - H_0^{o,u}H_1^{o,l} & H_1^{e,u}H_0^{o,u} - H_0^{o,u}H_1^{e,l} \\ H_0^{e,l}H_1^{o,l} - H_1^{o,l}H_0^{e,u} & H_0^{e,l}H_1^{e,l} - H_1^{o,l}H_0^{o,u} \end{bmatrix} \tag{34}$$

where the explicit dependence on $\tilde{\mathbf{A}}$ is not shown for brevity. If (11) is satisfied, it is shown in Appendix A that the diagonal-blocks, e.g. block $(1,1)$, are equal to the identity matrix (of appropriate size) and the anti-diagonal-blocks, e.g. block $(2,1)$, are equal to the zero matrix (of appropriate size). Therefore (34) is equal to the identity matrix and the proof is complete. ■

Equation (33) shows that $\mathbf{T}_S$ is the inverse of $\mathbf{T}_A$ (and vice versa). However no explicit inversion of $\mathbf{T}_A$ is required as long as $\mathbf{T}_S$ satisfies (32), i.e. the inverse is implicit. The result in Lemma 9 is analogous to the equivalence relationship in Theorem 3.7 in [29] (page 122) for 1-D systems and will be needed to prove the generalizations in section VI.

VI. Generalizations

The original formulation of the two-channel graph filter bank of Narang and Ortega [1], [2] was for undirected graphs. The spectral filters used are polynomial functions and the equivalent transformation matrices are matrix polynomials. We define the term *base matrix* as the independent matrix variable in the matrix polynomials. The base matrix used in [1], [2] is the symmetrically normalized Laplacian $\mathcal{L}$ and allows the spectral folding phenomena in bipartite graphs to be exploited. As explained in section II-C, an alternate but equivalent formulation uses the normalized adjacency matrix $\tilde{\mathbf{A}}$ defined in (5) as the base matrix. The spectral filters are then polynomials in the variable $\mu = \lambda - 1$ with the following associations: $\mu \to \tilde{\mathbf{A}}$ and $\lambda \to \mathcal{L}$.

For undirected graphs the matrices $\mathcal{L}$ and $\tilde{\mathbf{A}}$ are symmetric. If we examine carefully the analysis presented in sections III and V no symmetry assumption is made about the base matrix $\tilde{\mathbf{A}}$. The only requirement is that the base matrix is of the block anti-diagonal form shown in (12) which is equivalent to the underlying graph being bipartite. We refer to any such base matrix as an *admissible matrix*. This means that the base matrix could be

1) the un-normalized adjacency matrix $\mathbf{A}$,
2) a non-symmetrically normalized adjacency matrix, such as the random-walk matrix $\mathbf{D}^{-1}\mathbf{A}$. Other examples are considered in Section VII.

Furthermore since no assumption of matrix symmetry is made, the bipartite graph could be directed. This extension / generalization can be formally stated in the following theorem:

Theorem 4: The analysis filter bank in Fig. 1 and the synthesis bank in Fig. 3 form a perfect reconstruction (PR) system, i.e. $\mathbf{f} = \mathbf{f}^R$ if the spectral filters $H_0(\mu)$, $H_1(\mu)$, $G_0(\mu)$ and $G_0(\mu)$ are polynomial functions satisfying

$$G_0(\mu) = H_1(-\mu), \quad G_1(\mu) = H_0(-\mu) \tag{35}$$

$$H_0(\mu)H_1(-\mu) + H_0(-\mu)H_1(\mu) = 2 \tag{36}$$

and the base matrix $\tilde{\mathbf{A}}$ is an admissible matrix, i.e. of the form in (12).

Proof: Since $\mu = \lambda - 1$, equations (35) and (36) implies equations (3) and (4) respectively. Using Lemma 2, equations (10) and (11) are satisfied. Using Lemma 9, PR is achieved. ■

The symbol $\tilde{\mathbf{A}}$ was initially defined as the symmetrically normalized adjacency matrix in (5). However from here on $\tilde{\mathbf{A}}$ is used to denote any matrix of the form in (12). The domain of the polynomial filters is the disc $|\mu| \leq \rho(\tilde{\mathbf{A}})$ where $\rho(\tilde{\mathbf{A}})$ is the spectral radius. If $\tilde{\mathbf{A}}$ is normalized such that $\rho(\tilde{\mathbf{A}}) = 1$ then the filters are graph independent. Furthermore if the eigenvalues are real then $-1 \leq \mu \leq 1$. e.g. (5).

The DSP$_G$ framework of Sandryhaila and Moura [3], [4] for graph signal processing is based on the polynomial of the adjacency matrix of a graph which could be undirected or directed. Theorem 4 shows that the bipartite graph filter bank of Narang and Ortega [1], [2] can be extended for the DSP$_G$ framework. For an arbitrary (non-bipartite) undirected graph, Narang and Ortega [1], [2] proposed a decomposition into a series of bipartite graphs which is not unique. For Theorem 4 to be applicable to arbitrary (non-bipartite) directed graphs, some form of decomposition needs also to be performed. The development of techniques for directed graph decomposition is beyond the scope of this paper and left for future work.

VII. Application

We consider a simple application where we represent the same signal with filter banks based on different base matrices, and their respective performance in terms of non linear approximation. The effect of using different admissible base matrices is investigated. The base matrix considered in [1], [2] is the symmetrically normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{A}^S \equiv \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. Another normalization which leads to a non-symmetric matrix is $\tilde{\mathbf{A}} = \mathbf{A}^{RW} \equiv \mathbf{D}^{-1}\mathbf{A}$, commonly known as a random walk (RW) matrix. The RW matrix is row stochastic as the sum of every row is equal to unity, i.e. $\mathbf{A}^{RW}\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the all ones column vector. Using $\mathbf{A}^{RW}$ is equivalent to pre-scaling the input signal $\mathbf{f}$ by $\mathbf{D}^{1/2}$ before the analysis filter bank and post-scaling of

the reconstructed signal $\mathbf{f}$ by $\mathbf{D}^{-1/2}$ after the synthesis filter bank. This scaling prevents DC leakage of the high-pass signal into the low-pass band if $H_1(-1) = 0$ (zero DC response in the high-pass filter) and the system is known as a zeroDC filterbank [2].

In the context of adaptive image filtering, Milanfar [30] unified various well known filters such as the bilateral and NLM (Non-Local-Means) filters using a matrix representation. The matrix, denoted by $\mathbf{W}$, can be intepreted as an adjacency matrix of correlation weights between pixels. The relation between the bilateral filter and spectral graph filtering is also explored in [31]. With these image filters a normalization, equivalent to post-multiplication with $\mathbf{D}^{-1}$, is usually applied to ensure the pixels value are within a suitable range. The equivalent overall filtering matrix $\mathbf{D}^{-1}\mathbf{W}$ is therefore a RW matrix but is non-symmetric. The lack of symmetry has motivated Milanfar [32], [33] to propose a symmetrization procedure to the non-symmetric matrix. The symmetrization procedure is known as the Sinkhorn-Knopp (SK) balancing algorithm [34], [35]. At each step of the SK algorithm, column normalization followed by row normalization is applied to the matrix. Provided the original non-symmetric matrix satisfies some conditions (see [30], [34], [35] for details), the algorithm converges to a symmetric matrix $\mathbf{W}^{DS}$ that is doubly stochastic, i.e. sum of every row or column is unity. Improvement in performance was observed in [32], [33] when using $\mathbf{W}^{DS}$ instead of $\mathbf{D}^{-1}\mathbf{W}$.

More recently however, it was demonstrated in [36] that even with just one step of the SK algorithm, significant performance improvement is made when compared to using $\mathbf{D}^{-1}\mathbf{W}$. It was also observed in [36] that applying the full SK algorithm (that converge to a doubly stochastic matrix) does not always give the best result. In some cases one or two step(s) gave the best result and in other cases the improvement is marginal after a few steps. We now consider using the finite steps SK algorithm on the adjacency matrix $\mathbf{A}$ to give the base matrix $\tilde{\mathbf{A}}$ of the spectral filters. The symbol $\mathbf{A}_R^{SK}$ is used to denote the matrix from the R steps SK algorithm. Each step of the algorithm is given by the equation

$$\mathbf{A}_k^{SK} = \mathbf{D}_r^{-1}\mathbf{A}_{k-1}^{SK}\mathbf{D}_c^{-1} \quad \text{for} \quad k = 1, \ldots, R$$

where $\mathbf{D}_c \equiv \text{diag}\{(\mathbf{A}_{k-1}^{SK})^T\mathbf{1}\}$ and $\mathbf{D}_r \equiv \text{diag}\{\mathbf{A}_{k-1}^{SK}\mathbf{D}_c^{-1}\mathbf{1}\}$. The algorithm is initialized with the adjacency matrix, i.e $\mathbf{A}_0^{SK} = \mathbf{A}$. Note that $\mathbf{A}_R^{SK}$ in general is non-symmetric but is row-stochastic ($\mathbf{A}_k^{SK}\mathbf{1} = \mathbf{1}$) due to the $\mathbf{D}_r^{-1}$ normalization. Therefore the filter bank will still have the zeroDC property discussed above. Multiplication by $\mathbf{D}_c^{-1}$ or $\mathbf{D}_r^{-1}$ scales every element of the matrix $\mathbf{A}_{k-1}^{SK}$ independently. Elements that are zero-valued will remain zero-valued and the sparsity of the matrix is not affected by the SK steps. This also means that if $\mathbf{A}_0^{SK} = \mathbf{A}$ is admissible, i.e. of the form in (12), the zero submatrix blocks $\mathbf{0}_{|L|}$ and $\mathbf{0}_{|H|}$ will be unaffected by the SK steps. Therefore $\mathbf{A}_k^{SK}$ is admissible for all k.

The graph signal considered is the Minnesota traffic graph signal from [1], [2] and is shown in Fig. 7. The graph is 3-colorable and can be decomposed into a series of 2 bipartite graphs using the Harary algorithm [37]. There are 3 channels (LL, LH and HH channels) from the decomposition using the separable filter bank. The graphBior(5,5) filter pair from [2] is used in the experiments here. In the reconstruction all low-pass (LL) coefficients and a certain percentage of the largest magnitude high-pass coefficients are used. The simulations were performed with the aid of the GraphBior toolbox that accompanies the work in [2] and the SGWT toolbox that accompanies the work in [16]. The PSNR values of the reconstructed signal with different base matrices are shown in Table II. The result using the $\mathbf{A}^{RW}$ is significantly better than using $\mathbf{A}^S$ as DC leakage is not present in the former but present in the latter. Note that since $\mathcal{L} = \mathbf{A}^S + \mathbf{I}$ (see (5)), the result with $\mathbf{A}^S$ is the same as the result using the Laplacian based approach in [2]. There is substantial improvement using $\mathbf{A}_R^{SK}$ and the PSNR increases with the number of SK steps R. However there is a tapering-off in the rate of increase in PSNR with R. Practically therefore, it may not be worth applying the full SK algorithm to give a doubly stochastic base matrix.

VIII. Conclusion

The polyphase structure in the downsampled domain, which is computationally efficient, for critically sampled bipartite graph filter banks was derived in this work. It was shown that the signals in the downsampled domain are defined on equivalent subgraphs and the filtering can be defined as spectral filters with respect to these subgraphs. The results from the polyphase analysis reveal analogies of the graph filter banks with 1-D filter banks but some differences also exist. The theoretical results allowed the generalization of the filter bank to bipartite directed graphs and filters with more general base matrices. It was also demonstrated that the use of alternative base matrices can lead to improvement in non-linear approximation applications.

Future work can include developing techniques for decomposing an arbitrary directed graph into a series of bipartite directed graphs. Another direction is to develop techniques to find a good admissible base matrix for a particular situation or application. This is related to the open issue of "Other Graph Matrices" discussed in [5].

Appendix

A. Submatrices arising in Lemma 9

We show here that the submatrices in (34) are equal to either the identity matrix or the zero matrix if (11) is satisfied. The proof for the two anti-diagonal-blocks (diagonal-blocks) are similar to each other and only one will be shown.

The following are trivial scalar equations: (i) $H_1^e(\mu)H_0^o(\mu) - H_0^o(\mu)H_1^e(\mu) = 0$; (ii) $H_0^e(\mu)H_1^o(\mu) - H_1^o(\mu)H_0^e(\mu) = 0$. Subtituting μ with $\tilde{\mathbf{A}}$ in the equation in (i) gives $H_1^e(\tilde{\mathbf{A}})H_0^o(\tilde{\mathbf{A}}) - H_0^o(\tilde{\mathbf{A}})H_1^e(\tilde{\mathbf{A}}) = \mathbf{0}_N$. Using the results in Appendix B on this equation and extracting the $(2,1)$ block gives the following identity $H_1^{e,u}(\tilde{\mathbf{A}})H_0^{o,u}(\tilde{\mathbf{A}}) - H_0^{o,u}(\tilde{\mathbf{A}})H_1^{e,l}(\tilde{\mathbf{A}}) = \mathbf{0}_{|L|\times|H|}$.

Equation (11) can be written as either (i) $\det\mathbf{P}_a(\mu) = H_1^e(\mu)H_0^e(\mu) - H_0^o(\mu)H_1^o(\mu) = 1$, or (ii) $\det\mathbf{P}_a(\mu) = H_0^e(\mu)H_1^e(\mu) - H_1^o(\mu)H_0^o(\mu) = 1$. Subtituting μ with $\tilde{\mathbf{A}}$ in the equation in (i) gives $H_1^e(\tilde{\mathbf{A}})H_0^e(\tilde{\mathbf{A}}) - H_0^o(\tilde{\mathbf{A}})H_1^o(\tilde{\mathbf{A}}) = \mathbf{I}_N$. Using the results in Appendix B on this equation

and extracting the $(1,1)$ block gives the following identity $H_1^{e,u}(\tilde{\mathbf{A}})H_0^{e,u}(\tilde{\mathbf{A}}) - H_0^{o,u}(\tilde{\mathbf{A}})H_1^{o,l}(\tilde{\mathbf{A}}) = \mathbf{I}_{|L|}$.

B. Matrix products

Expressions of the matrix products, such as $H_0^e(\tilde{\mathbf{A}})H_0^e(\tilde{\mathbf{A}})$, in term of the submatrices, such as $H_0^{e,u}(\tilde{\mathbf{A}})$ or $H_0^{e,u}(\tilde{\mathbf{A}})$ are derived here. The matrices have the form as shown in (15) or (16). Generic symbols for the matrices, such as $H_i^e(\tilde{\mathbf{A}})$, and submatrices, such as $H_i^{e,l}(\tilde{\mathbf{A}})$, where $i = 0, 1$, will be used. By explicit multiplication of matrices of the form shown in (15) or (16), the following types of products can be obtained:

$$H_i^e(\tilde{\mathbf{A}})H_j^e(\tilde{\mathbf{A}}) =$$

$$\begin{bmatrix} H_i^{e,u}(\tilde{\mathbf{A}})H_j^{e,u}(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & H_i^{e,l}(\tilde{\mathbf{A}})H_j^{e,l}(\tilde{\mathbf{A}}) \end{bmatrix}$$

$$H_i^e(\tilde{\mathbf{A}})H_j^o(\tilde{\mathbf{A}}) =$$

$$\begin{bmatrix} \mathbf{0}_{|L|} & H_i^{e,u}(\tilde{\mathbf{A}})H_j^{o,u}(\tilde{\mathbf{A}}) \\ H_i^{e,l}(\tilde{\mathbf{A}})H_j^{o,l}(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{bmatrix}$$

$$H_i^o(\tilde{\mathbf{A}})H_j^e(\tilde{\mathbf{A}}) =$$

$$\begin{bmatrix} \mathbf{0}_{|L|} & H_i^{o,u}(\tilde{\mathbf{A}})H_j^{e,l}(\tilde{\mathbf{A}}) \\ H_i^{o,l}(\tilde{\mathbf{A}})H_j^{e,u}(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{bmatrix}$$

$$H_i^o(\tilde{\mathbf{A}})H_j^o(\tilde{\mathbf{A}}) =$$

$$\begin{bmatrix} H_i^{o,u}(\tilde{\mathbf{A}})H_j^{o,l}(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & H_i^{o,l}(\tilde{\mathbf{A}})H_j^{o,u}(\tilde{\mathbf{A}}) \end{bmatrix}$$

for $i, j = 0, 1$. Note that two of the matrices are block diagonal and the other two are block anti-diagonal.

REFERENCES

[1] S. K. Narang and A. Ortega, "Perfect Reconstruction Two-Channel Wavelet Filter Banks for Graph Structured Data," *IEEE Trans. Signal Proc.*, vol. 60, no. 6, pp. 2786–2799, June 2012.

[2] ——, "Compact Support Biorthogonal Wavelet Filterbanks for Arbitrary Undirected Graphs," *IEEE Trans. Signal Proc.*, vol. 61, no. 19, pp. 4673–4685, October 2013.

[3] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs," *IEEE Trans. Signal Proc.*, vol. 61, no. 7, pp. 1644–1636, April 2013.

[4] ——, "Discrete Signal Processing on Graphs: Frequency Analysis," *IEEE Trans. Signal Proc.*, vol. 62, no. 12, pp. 3042–3054, June 2014.

[5] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The Emerging Field of Signal Processing on Graphs," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[6] A. Sandryhaila and J. M. Moura, "Big Data Analysis with Signal Processing on Graphs," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, September 2014.

[7] F. R. K. Chung, *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, No. 92, 1996.

[8] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Maths., 1992.

[9] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice-Hall, 1993.

[10] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

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

[12] W. Wang and K. Ramchandran, "Random multiresolution representations for arbitrary sensor network graphs," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, May 2006, p. IV.

[13] R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied Computational and Harmonic Analysis*, vol. 21, pp. 53–94, 2006.

[14] G. Shen and A. Ortega, "Optimized distributed 2D transforms for irregularly sampled sensor network grids using wavelet lifting," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, April 2008, pp. 2513–2516.

[15] S. K. Narang and A. Ortega, "Lifting based wavelet transforms on graphs," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. (APSIPAASC)*, October 2009, pp. 441–444.

[16] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied Computational and Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[17] Y. Tanaka and A. Sakiyama, "M-Channel Oversampled Graph Filter Banks," *IEEE Trans. Signal Proc.*, vol. 62, no. 14, pp. 3578–3590, July 2014.

[18] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, "Discrete Signal Processing on Graphs: Sampling Theory," *IEEE Trans. Signal Proc.*, vol. 63, no. 24, pp. 6510–6523, December 2015.

[19] V. N. Ekambaram, G. C. Fanti, B. Ayazifar, and K. Ramchandran, "Spline-like wavelet filterbanks for multiresolution analysis of graph-structured data," *IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS*, vol. 1, no. 4, pp. 268–278, December 2015.

[20] O. Teke and P. P. Vaidyanathan, "Fundamentals of Multirate Graph Signal Processing," in *Proc. 49th Asilomar Conference on Signals, Systems and Computers*, November 2015.

[21] ——, "Graph Filter Banks with M-Channels, Maximal Decimation, and Perfect Reconstruction," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4809 – 4093.

[22] D. B. H. Tay and Z. Lin, "Design of Near Orthogonal Graph Filter Banks," *IEEE Signal Proc. Letters*, vol. 22, no. 6, pp. 701–704, June 2015.

[23] D. B. H. Tay and J. Zhang, "Techniques for Constructing Biorthogonal Bipartite Graph Filter Banks," *IEEE Trans. Signal Proc.*, vol. 63, no. 21, pp. 5772–5783, November 2015.

[24] J.-Z. Jiang, F. Zhou, and P.-L. Shui, "Optimization Design of Two-Channel Biorthogonal Graph Filter Banks," *Circuits, Systems and Signal Processing*, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00034-015-0073-x

[25] A. Sakiyama, K. Watanabe, and Y. Tanaka, "Spectral Graph Wavelets and Filter Banks With Low Approximation Error," *IEEE Trans. Signal and Information Proc. over Networks*, vol. 2, no. 3, pp. 230–244, September 2016.

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

[27] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[28] J. Zeng, G. Cheung, and A. Ortega, "Bipartite subgraph decomposition for critically sampled wavelet filterbanks on arbitrary graphs," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6210 – 6214.

[29] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice-Hall, 1995.

[30] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, January 2013.

[31] A. Gadde, S. Narang, and A. Ortega, "Bilateral Filter: Graph Spectral Interpretation and Extensions," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2013, pp. 1222–1226.

[32] P. Milanfar, "Symmetrizing Smoothing Filters," *SIAM J. IMAGING SCIENCES*, vol. 6, no. 1, pp. 263–284, 2013.

[33] A. Kheradmand and P. Milanfar, "A general framework for regularized, similarity-based image restoration," *IEEE Trans. Image Proc.*, vol. 23, no. 12, pp. 5136–5151, December 2014.

[34] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific J. Math.*, vol. 21, p. 343348, 1967.

[35] P. A. Knight, "The sinkhornknopp algorithm: Convergence and applications," *SIAM J. Matrix Anal. Appl.*, vol. 30, p. 261275, 2008.

[36] S. H. Chan, T. Zickler, and Y. M. Lu, "Understanding symmetric smoothing filters via Gaussian mixtures," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2015, pp. 2500 – 2504.

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

TABLE I

COMPARISON BETWEEN 1-D AND GRAPH FILTER BANKS MATRICES

	1-D	Graph
Time / Vertex Domain	$\mathbf{T}_a^{1D}$	$\mathbf{T}_A$
Frequency / Spectral Domain Independent Variable	$\mathbf{H}_p^{1D}(z)$ z (complex-valued)	$\mathbf{P}_a(\mu)$ μ (real-valued)

TABLE II

PSNR OF RECONSTRUCTED SIGNAL USING VARIOUS BASE MATRICES

Base matrix $\tilde{\mathbf{A}}$	PSNR (dB) using 4% high-pass	PSNR (dB) using 2% high-pass
$\mathbf{A}^S$	14.96	14.10
$\mathbf{A}^{RW}$	32.10	25.89
$\mathbf{A}_1^{SK}$	35.06	27.37
$\mathbf{A}_3^{SK}$	36.89	28.37
$\mathbf{A}_5^{SK}$	38.10	28.67
$\mathbf{A}_{10}^{SK}$	39.42	28.72
$\mathbf{A}_{20}^{SK}$	40.39	28.78

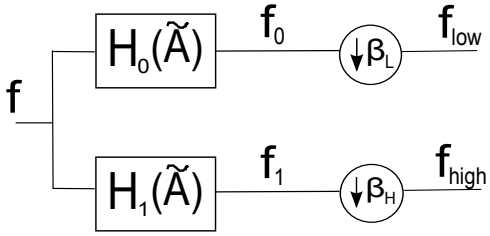

Fig. 1. Analysis filter bank.

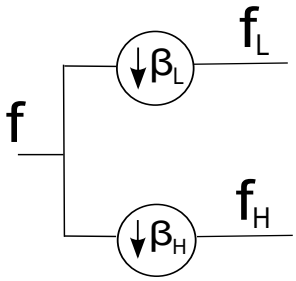

Fig. 2. Bipartite decomposition.

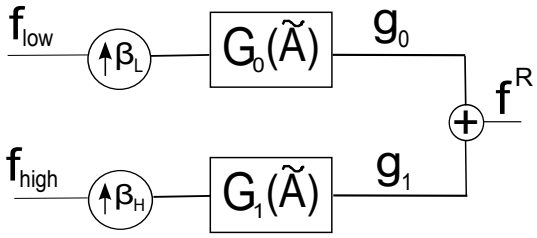

Fig. 3. Synthesis filter bank.

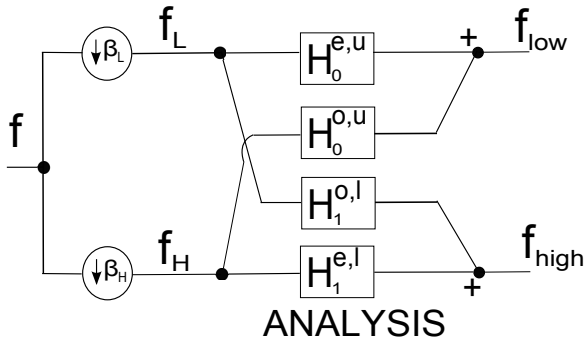

Fig. 4. Analysis polyphase structure.

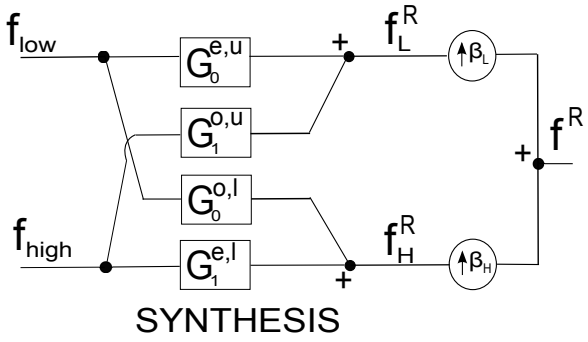

Fig. 5. Synthesis polyphase structure.

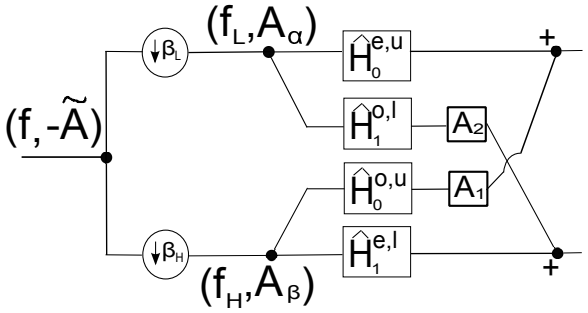

Fig. 6. Equivalent analysis polyphase structure.

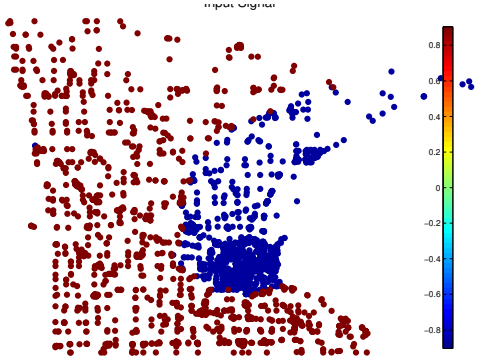

Fig. 7. Minnesota traffic graph signal.