Cascade and Lifting Structures in the Spectral Domain for Bipartite Graph Filter Banks

David B. H. Tay* and Antonio Ortega† Aamir Anis†
* School of Information Technology, Deakin University, Australia.
E-mail: david.tay@deakin.edu.au
† Signal & Image Processing Institute, Ming Hsieh Department of Electrical Engineering,
University of Southern California, USA.
E-mail: ortega@sipi.usc.edu; aanis@usc.edu

***Abstract*—In classical multirate filter bank systems, the cascade (product) of simple polyphase matrices is an important technique for the theory, design and implementation of filter banks. A particularly important class of cascades uses elementary matrices and leads to the well known lifting scheme in wavelets. In this paper the theory and principles of cascade and lifting structures for bipartite graph filter banks are developed. Accurate spectral characterizations of these structures using equivalent subgraphs will be presented. Some features of the structures in the graph case, that are not present in the classical case, will be discussed.**

***Index Terms*—Polyphase Structures, Graph Filter Banks, Spectral Graph Wavelets.**

I. Introduction

Applications where the data are defined over irregular domains, e.g. sensor, social and transportation networks, require a new generation of signal processing techniques that are adapted to signals over graphs. Earlier reviews of graph signal processing (GSP) are found in [1], [2]. More recent developments are found in [3]–[5]. In the classical regular domain, the wavelet transform and its variants [6]–[9] are perhaps the most popular and successful transforms in a plethora of applications. There have been several approaches to extend and generalize the wavelet transform to graph signals [10]–[18]. Many graph transforms are however not critically sampled and/or do not have an explicit spectral representation. Graph transforms that are based on two-channel critically sampled perfect reconstruction (PR) filter banks (FB) were first proposed by Narang and Ortega [16], [17]. The graph FB (GFB) in [16], [17] is defined for undirected graphs and the 'base' matrix for filtering is the normalized Laplacian matrix. Extension and generalization of the FB to directed graphs with more general base matrices are found in [19].

In the classical regular domain case, the cascade polyphase structures and lifting structures are important in the theory, design and implementation of FBs [7]–[9]. Here, one usually does not need to distinguish between the structure for design and the structure for implementation. The classical Noble identity allows one to easily move filtering operations from the higher sampling rate to the lower sampling rate, i.e. $H(z^2) \to H(z)$, thus facilitating the derivation of the efficient polyphase implementation structure. For graph signals the relationship between the design and implementation structures is however not simple. The polyphase representation matrix (PRM), introduced in [20], is an alternative and more succinct way to represent the GFB functions and the PR conditions. PRMs are useful for filter design but do not provide a direct representation of the filtering operations in the downsampled domain. In order to address this shortcoming the concept of the polyphase transform matrix (PTM) was then developed in [19] as a way to represent the filtering operations in the downsampled domain. The PRM can be viewed as a spectral domain representation of the GFB whereas and the PTM can be viewed as a vertex domain representation. The concept of a cascade (product) of PRM was introduced in [20] but no equivalent result for PTM has been published so far. Ladder structures, which are a special case of PRM cascades and are lifting-like, were used for filter design in [20]. However the equivalent lifting PTM, corresponding to this class of cascades, was not considered in [19].

In this paper we develop the cascade and lifting implementation structures for bipartite GFB. Lifting based transforms have also been previously proposed in [13], [14] for graph signals. However the filters in [13], [14] are defined in the vertex domain and cannot be readily interpreted in the spectral domain. The filters in this paper are however defined in the spectral domain and therefore allow us to control the spectral characteristics, e.g. low-pass, of the filters used in the ladder structures. Furthermore, only two lifting steps were considered in [13], [14] but the number of steps considered here is arbitrary. This paper will formally derive the lifting implementation structures and provide accurate spectral characterization of the signals and filters w.r.t. equivalent subgraphs.

II. Definitions and Preliminaries

A very brief overview of some graph signal processing concepts that are relevant to this work is presented here. More details are found in [1], [15]–[17], [19]. A graph $G = (V, E)$ is defined by the set of vertices V and edges E. The adjacency matrix $\mathbf{A}$ is an $N \times N$ matrix whose element $a_{i,j}$ $(i, j = 1, \ldots, N)$ is positive real and gives the weight of the directed edge from vertex j to vertex i. For undirected graphs $a_{i,j} = a_{j,i}$ (symmetric) but we will consider the general non-symmetric case. 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 $\mathbf{f} = [f(1) \cdots f(N)]^T$. A graph filter can be defined in terms of the spectral filter function $h(\mu)$ where μ is the spectral variable. Although $h(\mu)$, in principle, can be any transcendental function, in practice it is usually a polynomial function (in the variable μ) for efficient implementation and the localization property. For the rest of this paper we will *assume polynomial* $h(\mu)$. When μ is substituted with an $N \times N$ base matrix $\tilde{\mathbf{A}}$, we have the transformation matrix $h(\tilde{\mathbf{A}})$ that can be used in the filtering process, i.e. the filtered output $\mathbf{f}_{out} = h(\tilde{\mathbf{A}})\,\mathbf{f}$. Commonly used base matrices are the adjacency and the Laplacian.

The critically sampled two-channel filter bank (FB) proposed in [16], [17] 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. In [16], [17] only undirected graphs were considered. Generalizations to directed graphs were presented in [19]. The analysis and synthesis FBs are shown in Figs. 1 and 2 respectively.

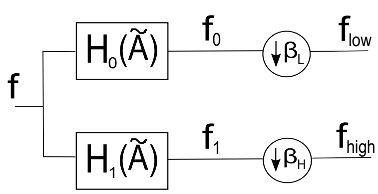

Fig. 1. Analysis filter bank.

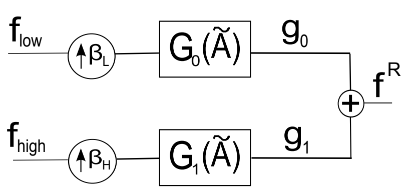

Fig. 2. Synthesis filter bank.

Theorem 1 ([19]): The analysis filter bank in Fig. 1 and the synthesis bank in Fig. 2 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

$$G_0(\mu) = H_1(-\mu), \quad G_1(\mu) = H_0(-\mu) \tag{1}$$

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

and the base matrix $\tilde{\mathbf{A}}$ is an *admissible matrix* of the form:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0}_{|L|} & \mathbf{A}_1 \\ \mathbf{A}_2 & \mathbf{0}_{|H|} \end{bmatrix}. \tag{3}$$

An alternative and more succinct way to express the PR conditions is via the polyphase representation of the filter functions [20]. In this approach each filter function is partitioned into an *even* part and an *odd* part, e.g., for filter $H_0(\mu)$, we have

$$H_0^e(\mu) \equiv \tfrac{1}{2}(H_0(\mu) + H_0(-\mu)) = \sum_k h_0(2k)\mu^{2k} \tag{4}$$

and

$$H_0^o(\mu) \equiv \tfrac{1}{2}(H_0(\mu) - H_0(-\mu)) = \sum_k h_0(2k+1)\mu^{2k+1}. \tag{5}$$

The analysis *polyphase representation matrix* (PRM) is then defined as

$$\mathbf{P}_a(\mu) = \begin{bmatrix} H_0^e(\mu) & H_0^o(\mu) \\ H_1^o(\mu) & H_1^e(\mu) \end{bmatrix}. \tag{6}$$

The following symmetric properties can be verified.

Property 1 (Symmetry):

$$\begin{aligned} \mathbf{P}_a(\mu) + \mathbf{P}_a(-\mu) &= 2\,\mathrm{diag}(H_0^e(\mu), H_1^e(\mu)), \\ \mathbf{P}_a(\mu) - \mathbf{P}_a(-\mu) &= 2\,\mathbf{I}_a\,\mathrm{diag}(H_0^o(\mu), H_1^o(\mu)). \end{aligned}$$

where $\mathbf{I}_a \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is the anti-diagonal unit matrix. The *synthesis PRM* $\mathbf{P}_s(\mu)$ can be similarly defined:

$$\mathbf{P}_s(\mu) = \begin{bmatrix} G_0^e(\mu) & G_0^o(\mu) \\ G_1^o(\mu) & G_1^e(\mu) \end{bmatrix}. \tag{7}$$

The next Lemma gives the equivalent PR conditions.

Lemma 1 ([20]): (1) is satisfied if

$$\mathbf{P}_s(\mu) = \mathbf{I}_a\,\mathbf{P}_a(-\mu)\,\mathbf{I}_a. \tag{8}$$

(2) is satisfied if

$$\det \mathbf{P}_a(\mu) = 1. \tag{9}$$

A matrix satisfying Property 1 and (9) is called a *valid matrix*.

III. Cascade Polyphase Structures

A. Dowsampled Filtering Structures

Some pertinent results from [19] are first presented. Some of the mathematical expressions used here may appear different from [19] but both are equivalent, and this is done for the convenience in later development. In Figs. 1 and 2 the signals and filters are in the upsampled domain. In [19] the polyphase analysis of the filter bank (FB) yielded equivalent filtering structures in the downsampled domain. The filtering in the downsampled domain is w.r.t. the subgraphs G_α and G_β with adjacency matrices $\mathbf{A}_\alpha \equiv \mathbf{A}_1\mathbf{A}_2$ and $\mathbf{A}_\beta \equiv \mathbf{A}_2\mathbf{A}_1$, respectively, followed by either the $\mathbf{A}_1$ or $\mathbf{A}_2$ projection operator. Figs. 3 and 4, respectively, show the equivalent analysis and synthesis structures [19].

The analysis subfilters are defined by the submatrices [19]:

$$\hat{\mathbf{H}}_0^{e,u} \equiv H_0^e(\mathbf{A}_\alpha) = \sum_k h_0(2k)(\mathbf{A}_\alpha)^k \tag{10}$$

$$\hat{\mathbf{H}}_1^{o,l} \equiv H_1^o(\mathbf{A}_\alpha) = \sum_k h_1(2k+1)(\mathbf{A}_\alpha)^k \tag{11}$$

$$\hat{\mathbf{H}}_0^{o,u} \equiv H_0^e(\mathbf{A}_\beta) = \sum_k h_0(2k+1)(\mathbf{A}_\beta)^k \tag{12}$$

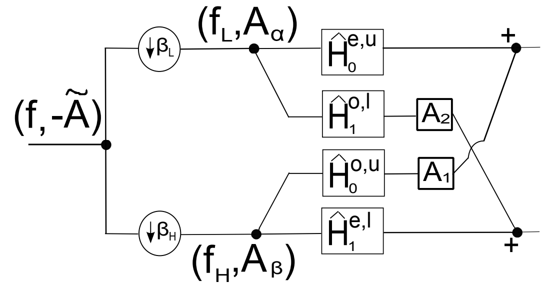

Fig. 3. Equivalent analysis polyphase structure.

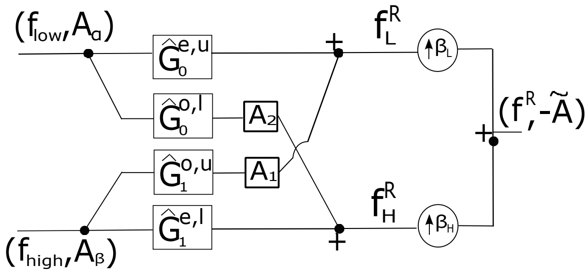

Fig. 4. Equivalent synthesis polyphase structure.

$$\hat{\mathbf{H}}_1^{e,l} \equiv H_1^o(\mathbf{A}_\beta) = \sum_k h_1(2k)(\mathbf{A}_\beta)^k \tag{13}$$

The synthesis subfilters are defined by the submatrices [19]:

$$\hat{\mathbf{G}}_0^{e,u} \equiv G_0^e(\mathbf{A}_\alpha) = \sum_k g_0(2k)(\mathbf{A}_\alpha)^k \tag{14}$$

$$\hat{\mathbf{G}}_0^{o,l} \equiv G_0^o(\mathbf{A}_\alpha) = \sum_k g_0(2k+1)(\mathbf{A}_\alpha)^k \tag{15}$$

$$\hat{\mathbf{G}}_1^{o,u} \equiv G_1^o(\mathbf{A}_\beta) = \sum_k g_1(2k+1)(\mathbf{A}_\beta)^k \tag{16}$$

$$\hat{\mathbf{G}}_1^{e,l} \equiv G_1^e(\mathbf{A}_\beta) = \sum_k g_1(2k)(\mathbf{A}_\beta)^k \tag{17}$$

The filtering in the downsampled domain can be defined succinctly using transform matrices of size $N \times N$. The analysis *polyphase transform matrix (PTM)* is defined as

$$\mathbf{T}_A \equiv \begin{bmatrix} \mathbf{H}_0^{e,u} & \mathbf{A}_1\mathbf{H}_0^{o,u} \\ \mathbf{A}_2\mathbf{H}_1^{o,l} & \mathbf{H}_1^{e,l} \end{bmatrix}. \tag{18}$$

and the *synthesis PTM* is defined as

$$\mathbf{T}_S \equiv \begin{bmatrix} \mathbf{G}_0^{e,u} & \mathbf{A}_1\mathbf{G}_1^{o,u} \\ \mathbf{A}_2\mathbf{G}_0^{o,l} & \mathbf{G}_1^{e,l} \end{bmatrix} \tag{19}$$

There is a relationship between the PRM and the PTM but it is important to note that the two matrices are not identical - see [19] for details. Furthermore replacing μ with $\tilde{\mathbf{A}}$ in $\mathbf{P}_a(\mu)$ and $\mathbf{P}_s(\mu)$ *does not* give the transform matrices defined above.

A simple and effective way to construct spectral filters is by a cascade (product) of PRMs.

Corollary 1 ([20]): The product of two valid matrices $\mathbf{P}_{a,1}(\mu)$ and $\mathbf{P}_{a,2}(\mu)$ is another valid matrix, i.e.

$$\mathbf{P}_a(\mu) \equiv \mathbf{P}_{a,2}(\mu)\mathbf{P}_{a,1}(\mu)$$

satisfy Property 1 and (9).

B. Cascade of Transform Matrices

The ability to express a given transform matrix as a product of simpler matrices has the following advantages:

1) Each simple factor (matrix) can (usually) be associated with a fundamental operation on the signal undergoing processing, e.g. simple averaging or differencing of neighborhood samples. This facilitates the understanding of the effect of the transform on the input signal.
2) Fast or computationally efficient implementation can be derived using the factorization results, e.g. lifting structure and 'in-place' computations.

The next two Lemmas can be used to derive cascade of transform matrices for GFBs.

Lemma 2: Given two analysis PRMs

$$\mathbf{P}_{a,i}(\mu) = \begin{bmatrix} H_{0,i}^e(\mu) & H_{0,i}^o(\mu) \\ H_{1,i}^o(\mu) & H_{1,i}^e(\mu) \end{bmatrix} \quad i = 1, 2$$

with the corresponding analysis PTMs $\mathbf{T}_{A,1}$ and $\mathbf{T}_{A,2}$ respectively. The product of PRMs

$$\mathbf{P}_a(\mu) \equiv \mathbf{P}_{a,2}(\mu)\mathbf{P}_{a,1}(\mu)$$

has a corresponding analysis PTM given by

$$\mathbf{T}_A = \mathbf{T}_{A,2}\,\mathbf{T}_{A,1} \tag{20}$$

The proof of Lemma 2 is found in Appendix A. A similar result for the synthesis bank is given by the next Lemma.

Lemma 3: Given two synthesis PRMs

$$\mathbf{P}_{s,i}(\mu) = \begin{bmatrix} G_{0,i}^e(\mu) & G_{0,i}^o(\mu) \\ G_{1,i}^o(\mu) & G_{1,i}^e(\mu) \end{bmatrix} \quad i = 1, 2$$

with the corresponding synthesis PTMs $\mathbf{T}_{S,1}$ and $\mathbf{T}_{S,2}$ respectively. The product of PRMs

$$\mathbf{P}_s(\mu) \equiv \mathbf{P}_{s,2}(\mu)\mathbf{P}_{s,1}(\mu)$$

has a corresponding synthesis PTM given by

$$\mathbf{T}_S = \mathbf{T}_{S,1}\,\mathbf{T}_{S,2} \tag{21}$$

The outline of the proof of Lemma 3 is found in Appendix B. Note that the ordering of the matrix product in (20) is opposite to that in (21).

Lemmas 2 and 3 can applied recursively to the product of any number (M) of representation matrices:

1) The transform matrix corresponding to

$$\prod_{i=M}^{1} \mathbf{P}_{a,i}(\mu) \quad \text{is} \quad \prod_{i=M}^{1} \mathbf{T}_{A,i}.$$

2) The transform matrix corresponding to

$$\prod_{i=M}^{1} \mathbf{P}_{s,i}(\mu) \quad \text{is} \quad \prod_{i=1}^{M} \mathbf{T}_{S,i}.$$

IV. Lifting Structures

Corollary 1 can be exploited for constructing spectral filter functions via the product of simple PRMs [20]. The simple PRMs are defined as follows:

Definition 1 (Type I valid matrix):

$$\mathbf{U}_i(\mu) \equiv \begin{bmatrix} 1 & \tilde{L}_i(\mu) \\ 0 & 1 \end{bmatrix} \tag{22}$$

Definition 2 (Type II valid matrix):

$$\mathbf{D}_i(\mu) \equiv \begin{bmatrix} 1 & 0 \\ \tilde{L}_i(\mu) & 1 \end{bmatrix} \tag{23}$$

where the kernels $\tilde{L}_i(\mu)$ $(i = 0, 1, \dots)$ are odd polynomial functions satisfying

$$\tilde{L}_i(\mu) \equiv \sum_{k \geq 0} \tilde{l}_i(2k+1)\mu^{2k+1} = -\tilde{L}_i(-\mu) \tag{24}$$

and $\tilde{l}_i(2k+1)$ denotes the coefficient. It is easy to verify that the Types I and II matrices above are valid.

The PTMs corresponding to the simple PRMs in (22) and (23) are obtained next. There are four cases to consider. The results are summarized in Table I. The steps for deriving the results are given below.

1) When $\mathbf{P}_a(\mu) = \mathbf{U}_i(\mu)$, $H_0^e(\mu) = H_1^e(\mu) = 1$, $H_1^o(\mu) = 0$ and $H_o^o(\mu) = \tilde{L}_i(\mu)$ in (6). Using (10), (11), (12) and (13) in (18) gives the Type I analysis PTM.
2) When $\mathbf{P}_a(\mu) = \mathbf{D}_i(\mu)$, $H_0^e(\mu) = H_1^e(\mu) = 1$, $H_0^o(\mu) = 0$ and $H_1^o(\mu) = \tilde{L}_i(\mu)$ in (6). Using (10), (11), (12) and (13) in (18) gives the Type II analysis PTM.
3) When $\mathbf{P}_a(\mu) = \mathbf{U}_i(\mu)$ (Type I), using (8), $\mathbf{P}_s(\mu) = \mathbf{D}_i(-\mu)$ (Type II). The subfilters in (7) are $G_0^e(\mu) = G_1^e(\mu) = 1$, $G_0^o(\mu) = 0$ and $G_1^o(\mu) = \tilde{L}_i(-\mu) = -\tilde{L}_i(\mu)$. Using (14), (15), (16) and (17) in (19) gives the Type II synthesis PTM.
4) When $\mathbf{P}_a(\mu) = \mathbf{D}_i(\mu)$ (Type II), using (8), $\mathbf{P}_s(\mu) = \mathbf{U}_i(-\mu)$ (Type I). The subfilters in (7) are $G_0^e(\mu) = G_1^e(\mu) = 1$, $G_0^o(\mu) = \tilde{L}_i(-\mu) = -\tilde{L}_i(\mu)$ and $G_1^o(\mu) = 0$. Using (14), (15), (16) and (17) in (19) gives the Type I synthesis PTM.

Note that a Type I analysis PRM gives a Type I analysis PTM but a Type I synthesis PRM gives a Type II synthesis PTM.

The next Corollory gives the analysis lifting structure shown in Fig. 5.

Corollary 2: If the analysis PRM is

$$\mathbf{P}_a(\mu) = \prod_{k=M}^{1} \mathbf{D}_{2k}(\mu)\mathbf{U}_{2k-1}(\mu) \tag{25}$$

then the equivalent PTM is given by

$$\mathbf{T}_A = \prod_{k=M}^{1} \mathbf{T}_{\mathbf{D}}^A(2k)\mathbf{T}_{\mathbf{U}}^A(2k-1) \tag{26}$$

Proof: Use first and second rows of Table I to obtain the constituent PTMs. Then apply Lemma 2 recursively. ■

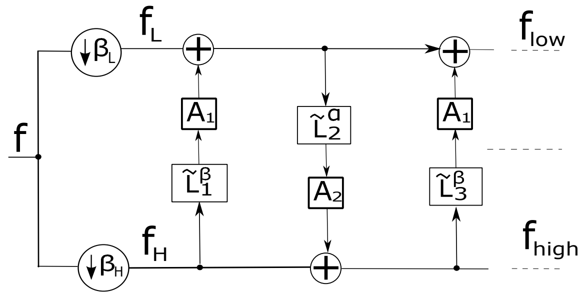

Fig. 5. Analysis lifting structure. $\tilde{\mathbf{L}}_i^\alpha \equiv \tilde{L}_i(\mathbf{A}_\alpha)$. $\tilde{\mathbf{L}}_i^\beta \equiv \tilde{L}_i(\mathbf{A}_\beta)$.

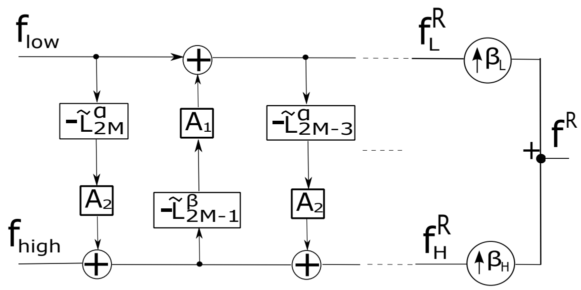

Fig. 6. Synthesis lifting structure. $\tilde{\mathbf{L}}_i^\alpha \equiv \tilde{L}_i(\mathbf{A}_\alpha)$. $\tilde{\mathbf{L}}_i^\beta \equiv \tilde{L}_i(\mathbf{A}_\beta)$.

The synthesis lifting structure shown in Fig. 6 is given by next Corollary.

Corollary 3: If the synthesis PRM is

$$\mathbf{P}_s(\mu) = \prod_{k=M}^{1} \mathbf{U}_{2k}(-\mu)\mathbf{D}_{2k-1}(-\mu) \tag{27}$$

then the equivalent PTM is given by

$$\mathbf{T}_S = \prod_{k=1}^{M} \mathbf{T}_{\mathbf{D}}^S(2k-1)\mathbf{T}_{\mathbf{U}}^S(2k) \tag{28}$$

Proof: Use third and fourth rows of Table Ito obtain the constituent PTMs. Then apply Lemma 3 recursively. ■

Corollary 4: The PTMs in Corollaries 2 and 3, i.e. (26) and (28), form a perfect reconstruction system, i.e.

$$\mathbf{T}_S\mathbf{T}_A = \mathbf{I}_N.$$

Proof: Firstly it can be verified that $\mathbf{T}_{\mathbf{U}}^S(i)\mathbf{T}_{\mathbf{D}}^A(i) = \mathbf{I}$ by explicit multiplication of $\mathbf{T}_{\mathbf{U}}^S(i)$ and $\mathbf{T}_{\mathbf{D}}^A(i)$ in Table I. Similarly, it can be verified that $\mathbf{T}_{\mathbf{D}}^S(i)\mathbf{T}_{\mathbf{U}}^A(i) = \mathbf{I}$ by using the result in Table I. Using these results on the explicit product of (28) and (26) gives

$$\mathbf{T}_{\mathbf{D}}^S(1)\mathbf{T}_{\mathbf{U}}^S(2)\dots\mathbf{T}_{\mathbf{D}}^S(2M-1)\overbrace{\mathbf{T}_{\mathbf{U}}^S(2M)\mathbf{T}_{\mathbf{D}}^A(2M)}^{=\mathbf{I}}$$
$$.\mathbf{T}_{\mathbf{U}}^A(2M-1)\dots\mathbf{T}_{\mathbf{D}}^A(2)\mathbf{T}_{\mathbf{U}}^A(1)$$
$$= \mathbf{T}_{\mathbf{D}}^S(1)\mathbf{T}_{\mathbf{U}}^S(2)\dots\overbrace{\mathbf{T}_{\mathbf{D}}^S(2M-1)\mathbf{T}_{\mathbf{U}}^A(2M-1)}^{=\mathbf{I}}$$

TABLE I
LIFTING POLYPHASE TRANSFORM MATRICES (PTM) FROM CORRESPONDING POLYPHASE REPRESENTATION MATRICES (PRM)

Case and Type	PRM	PTM
1) Type I Analysis	$\mathbf{P}_a(\mu) = \begin{bmatrix} 1 & \tilde{L}_i(\mu) \\ 0 & 1 \end{bmatrix}$	$\mathbf{T}^A_{\mathbf{U}}(i) \equiv \begin{bmatrix} \mathbf{I}_{\vert L\vert} & \mathbf{A}_1\tilde{L}_i(\mathbf{A}_\beta) \\ \mathbf{0}_{\vert H\vert\times\vert L\vert} & \mathbf{I}_{\vert H\vert} \end{bmatrix}$
2) Type II Analysis	$\mathbf{P}_a(\mu) = \begin{bmatrix} 1 & 0 \\ \tilde{L}_i(\mu) & 1 \end{bmatrix}$	$\mathbf{T}^A_{\mathbf{D}}(i) \equiv \begin{bmatrix} \mathbf{I}_{\vert L\vert} & \mathbf{0}_{\vert L\vert\times\vert H\vert} \\ \mathbf{A}_2\tilde{L}_i(\mathbf{A}_\alpha) & \mathbf{I}_{\vert N\vert} \end{bmatrix}$
3) Type II Synthesis	$\mathbf{P}_s(\mu) = \begin{bmatrix} 1 & 0 \\ -\tilde{L}_i(\mu) & 1 \end{bmatrix}$	$\mathbf{T}^S_{\mathbf{U}}(i) \equiv \begin{bmatrix} \mathbf{I}_{\vert L\vert} & \mathbf{0}_{\vert L\vert\times\vert H\vert} \\ -\mathbf{A}_2\tilde{L}_i(\mathbf{A}_\alpha) & \mathbf{I}_{\vert N\vert} \end{bmatrix}$
4) Type I Synthesis	$\mathbf{P}_s(\mu) = \begin{bmatrix} 1 & -\tilde{L}_i(\mu) \\ 0 & 1 \end{bmatrix}$	$\mathbf{T}^S_{\mathbf{D}}(i) \equiv \begin{bmatrix} \mathbf{I}_{\vert L\vert} & -\mathbf{A}_1\tilde{L}_i(\mathbf{A}_\beta) \\ \mathbf{0}_{\vert H\vert\times\vert L\vert} & \mathbf{I}_{\vert H\vert} \end{bmatrix}$

$$\ldots \mathbf{T}^A_{\mathbf{D}}(2)\mathbf{T}^A_{\mathbf{U}}(1)$$

By continued simplification of the product of two middle terms, a pair at a time, the final result is $\mathbf{T}_S\mathbf{T}_A = \mathbf{I}_N$. ■

A. Discussion

When designing lifting filters in the regular spatial domain, one may not know readily what the overall filterbank response would look like, but one will have a good interpretation of the prediction and update filters. In the graph case, for the designs that are based on (22) and (23), it is not immediately obvious what the equivalent prediction and update filters are. The results above provide an accurate characterization of the equivalent prediction and update filters.

The structures in Figs. 5 and 6 are reminiscent of the classical lifting for regular domain signals [21]. Just like in the classical case, for the graph case, it can be readily shown there is a computational complexity reduction by a factor of two compared to the full rate implementation [19]. However, there are important features in the graph case that are not present in the classical case. In the classical case the 'split' operation results in polyphase signals with equal number of samples. In the graph case, the equivalent bipartite decomposition operation results in polyphase signals with an unequal number of samples in general. The predict/update (also known as dual-lift/lift) filters in the classical case, as far as the domains are concerned, are indistinguishable. Both filters are defined over the same equivalent regular line graph. For the graph case however, the domain for the 'up' filters (Type I analysis and Type II synthesis) is subgraph G_α (with adjacency $\mathbf{A}_\alpha$). For the 'down' filters (Type II analysis and Type I synthesis) the domain is subgraph G_β (with adjacency $\mathbf{A}_\beta$). Finally, projection operators (either $\mathbf{A}_1$ or $\mathbf{A}_2$) are needed in the graph case to map the filtered output from one domain to another but not in the classical case.

V. CONCLUSION

Cascade and lifting structures for bipartite graph filter banks were derived in this work. The filtering is in the downsampled domains and is w.r.t. equivalent subgraphs. Projection operators are needed to map signals from one subgraph to another subgraph. Going from the design structure to the implementation structure is significantly more complicated for the graph case compared to the classical case.

APPENDIX

A. Proof of Lemma 2

It will be shown explicitly that the R.H.S. of equation (20) is equal to the L.H.S. of the equation. Explicit multiplication of $\mathbf{P}_{a,1}(\mu)$ and $\mathbf{P}_{a,2}(\mu)$ gives

$$\begin{bmatrix} H^e_{0,2}(\mu) & H^o_{0,2}(\mu) \\ H^o_{1,2}(\mu) & H^e_{1,2}(\mu) \end{bmatrix} \begin{bmatrix} H^e_{0,1}(\mu) & H^o_{0,1}(\mu) \\ H^o_{1,1}(\mu) & H^e_{1,1}(\mu) \end{bmatrix}$$
$$= \begin{bmatrix} H^e_{0,2}H^e_{0,1} + H^o_{0,2}H^o_{1,1} & H^e_{0,2}H^o_{0,1} + H^o_{0,2}H^e_{1,1} \\ H^o_{1,2}H^e_{0,1} + H^e_{1,2}H^o_{1,1} & H^o_{1,2}H^o_{0,1} + H^e_{1,2}H^e_{1,1} \end{bmatrix}.$$
$$= \begin{bmatrix} H^e_0(\mu) & H^o_0(\mu) \\ H^o_1(\mu) & H^e_1(\mu) \end{bmatrix} \tag{29}$$

The last line shows explicitly the symbol of each element in the product. There are 4 elements in the matrix equation above and each element results in scalar equation involving polynomials in the variable μ, e.g. $H^e_0(\mu) = H^e_{0,2}(\mu)H^e_{0,1}(\mu) + H^o_{0,2}(\mu)H^o_{1,1}(\mu)$ from element $(1,1)$. Each of the scalar equation becomes a matrix equation if μ is replaced with the adjacency matrix $\tilde{\mathbf{A}}$, e.g. from element $(1,1)$

$$H^e_0(\tilde{\mathbf{A}}) = H^e_{0,2}(\tilde{\mathbf{A}})H^e_{0,1}(\tilde{\mathbf{A}}) + H^o_{0,2}(\tilde{\mathbf{A}})H^o_{1,1}(\tilde{\mathbf{A}}) \tag{30}$$

Lemma 4 in [19] shows that matrices with the superscript e are block diagonal with the form

$$\begin{aligned} H^e_\square(\tilde{\mathbf{A}}) &= \begin{bmatrix} H^{e,u}_\square(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & H^{e,l}_\square(\tilde{\mathbf{A}}) \end{bmatrix} \\ &\equiv \begin{bmatrix} H^e_\square(\mathbf{A}_\alpha) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & H^e_\square(\mathbf{A}_\beta) \end{bmatrix} \end{aligned} \tag{31}$$

The symbol '$\square$' can be any subscript, e.g. $\square = 0$ or $\square = 0,2$. Note that $H^e_\square(\bullet)$ ($H^o_\square(\bullet)$) is the even (odd) part of $H_\square(\bullet)$ as defined in (4) ((5)). Lemma 4 in [19] shows that matrices with the superscript o are block anti-diagonal with the form

$$\begin{aligned} H^o_\square(\tilde{\mathbf{A}}) &= \begin{bmatrix} \mathbf{0}_{|L|} & H^{o,u}_\square(\tilde{\mathbf{A}}) \\ H^{o,l}_\square(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{bmatrix} \\ &\equiv \begin{bmatrix} \mathbf{0}_{|L|} & \mathbf{A}_1 H^o_\square(\mathbf{A}_\beta) \\ \mathbf{A}_2 H^o_\square(\mathbf{A}_\alpha) & \mathbf{0}_{|H|} \end{bmatrix}. \end{aligned} \tag{32}$$

Note also that in (31) and (32), for convenience in the sequel, we define $H^{e,u}_\square(\tilde{\mathbf{A}}) \equiv H^e_\square(\mathbf{A}_\alpha)$, etc. There are 4 types of matrix products in the matrix version ($\mu \rightarrow \tilde{\mathbf{A}}$) of equation (29): (i) 'e' $\times$ 'e', (ii) 'e' $\times$ 'o', (iii) 'o' $\times$ 'e' and (iv) 'o' $\times$

'o'. The generic expressions of the 4 types of products in term of the submatrices (e.g. $H^{e,u}_{0,1}(\tilde{\mathbf{A}})$) are derived in Appendix C. Using the expressions from Appendix C on (30) (which is from element $(1,1)$) gives

$$H^e_0(\tilde{\mathbf{A}}) \equiv \begin{bmatrix} H^{e,u}_0(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & H^{e,l}_0(\tilde{\mathbf{A}}) \end{bmatrix} = \tag{33}$$

$$\begin{bmatrix} (H^{e,u}_{0,2}H^{e,u}_{0,1} + H^{o,u}_{0,2}H^{o,l}_{0,1}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & (H^{e,l}_{0,2}H^{e,l}_{0,1} + H^{o,l}_{0,2}H^{o,u}_{0,1}) \end{bmatrix}$$

where the dependence on $\tilde{\mathbf{A}}$ is not shown for brevity. Similarly with element $(1,2)$ we have:

$$H^o_0(\tilde{\mathbf{A}}) \equiv \begin{bmatrix} \mathbf{0}_{|L|} & H^{o,u}_0(\tilde{\mathbf{A}}) \\ H^{o,l}_0(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{bmatrix} = \tag{34}$$

$$\begin{bmatrix} \mathbf{0}_{|L|} & (H^{e,u}_{0,2}H^{o,u}_{0,1} + H^{o,u}_{0,2}H^{e,l}_{1,1}) \\ (H^{e,l}_{0,2}H^{o,l}_{0,1} + H^{o,l}_{0,2}H^{e,u}_{1,1}) & \mathbf{0}_{|H|} \end{bmatrix}$$

With the element $(2,1)$ we have:

$$H^o_1(\tilde{\mathbf{A}}) \equiv \begin{bmatrix} \mathbf{0}_{|L|} & H^{o,u}_1(\tilde{\mathbf{A}}) \\ H^{o,l}_1(\tilde{\mathbf{A}}) & \mathbf{0}_{|H|} \end{bmatrix} = \tag{35}$$

$$\begin{bmatrix} \mathbf{0}_{|L|} & (H^{o,u}_{1,2}H^{e,l}_{0,1} + H^{e,u}_{1,2}H^{o,u}_{1,1}) \\ (H^{o,l}_{1,2}H^{e,u}_{0,1} + H^{e,l}_{1,2}H^{o,l}_{1,1}) & \mathbf{0}_{|H|} \end{bmatrix}$$

With the element $(2,2)$ we have:

$$H^e_1(\tilde{\mathbf{A}}) \equiv \begin{bmatrix} H^{e,u}_1(\tilde{\mathbf{A}}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & H^{e,l}_1(\tilde{\mathbf{A}}) \end{bmatrix} = \tag{36}$$

$$\begin{bmatrix} (H^{o,u}_{0,2}H^{o,l}_{0,1} + H^{e,u}_{1,2}H^{e,u}_{1,1}) & \mathbf{0}_{|L|\times|H|} \\ \mathbf{0}_{|H|\times|L|} & (H^{o,l}_{1,2}H^{o,u}_{0,1} + H^{e,l}_{1,2}H^{e,l}_{1,1}) \end{bmatrix}$$

Using the equivalent symbols introduced in (31) and (32), e.g. $H^{e,u}_0(\tilde{\mathbf{A}}) \equiv H^e_0(\mathbf{A}_\alpha)$, the transform matrix $\mathbf{T}_A$ in (18) can be written as:

$$\mathbf{T}_A \equiv \begin{bmatrix} H^{e,u}_0(\tilde{\mathbf{A}}) & H^{o,u}_0(\tilde{\mathbf{A}}) \\ H^{o,l}_1(\tilde{\mathbf{A}}) & H^{e,l}_1(\tilde{\mathbf{A}}) \end{bmatrix}.$$

The expression for the submatrices in $\mathbf{T}_A$ above can be obtained using the 4 matrix equations above, e.g. $H^{e,u}_0(\tilde{\mathbf{A}}) = (H^{e,u}_{0,2}H^{e,u}_{0,1} + H^{o,u}_{0,2}H^{o,l}_{0,1})$ using equation (33). Using these expressions we have

$$\mathbf{T}_A \equiv \begin{bmatrix} H^{e,u}_0(\tilde{\mathbf{A}}) & H^{o,u}_0(\tilde{\mathbf{A}}) \\ H^{o,l}_1(\tilde{\mathbf{A}}) & H^{e,l}_1(\tilde{\mathbf{A}}) \end{bmatrix} =$$

$$\begin{bmatrix} (H^{e,u}_{0,2}H^{e,u}_{0,1} + H^{o,u}_{0,2}H^{o,l}_{0,1}) & (H^{e,u}_{0,2}H^{o,u}_{0,1} + H^{o,u}_{0,2}H^{e,l}_{1,1}) \\ (H^{o,l}_{1,2}H^{e,u}_{0,1} + H^{e,l}_{1,2}H^{o,l}_{1,1}) & (H^{o,l}_{1,2}H^{o,u}_{0,1} + H^{e,l}_{1,2}H^{e,l}_{1,1}) \end{bmatrix}$$

$$= \begin{bmatrix} H^{e,u}_{0,2}(\tilde{\mathbf{A}}) & H^{o,u}_{0,2}(\tilde{\mathbf{A}}) \\ H^{o,l}_{1,2}(\tilde{\mathbf{A}}) & H^{e,l}_{1,2}(\tilde{\mathbf{A}}) \end{bmatrix} \begin{bmatrix} H^{e,u}_{0,1}(\tilde{\mathbf{A}}) & H^{o,u}_{0,1}(\tilde{\mathbf{A}}) \\ H^{o,l}_{1,1}(\tilde{\mathbf{A}}) & H^{e,l}_{1,1}(\tilde{\mathbf{A}}) \end{bmatrix}$$

$$= \mathbf{T}_{A,2}\,\mathbf{T}_{A,1}$$

The equivalence between the second and third lines can be readily verified by explicit multiplication of the third line. Note that every symbol in the equation above are matrices (and not scalars).

B. Proof of Lemma 3

The proof is similar to the proof of Lemma 2 but with some important differences. Only a sketch is provided here with the emphasis on highlighting the differences. The steps are:

1) Explicit product of $\mathbf{P}_{s,2}(\mu)$ and $\mathbf{P}_{s,1}(\mu)$ to obtain 4 scalar equations in the variable μ that is similar to equation (29) but the symbol H replaced with G.
2) Convert each scalar equation to a matrix equation by the substitution $\mu \to \tilde{\mathbf{A}}$. There is however an important difference in the conversion here compared to the proof of Lemma 2. The best way to describe this difference is through an example equation. Element $(1,1)$ of equation (29), with $H \to G$, is $G^e_0(\mu) = G^e_{0,2}(\mu)G^e_{0,1}(\mu) + G^o_{0,2}(\mu)G^o_{1,1}(\mu)$. By changing the ordering of the products we have an equivalent $G^e_0(\mu) = G^e_{0,1}(\mu)G^e_{0,2}(\mu) + G^o_{1,1}(\mu)G^o_{0,2}(\mu)$. This might seem a trivial thing to do but when the scalar equation is converted to a matrix equation (via $\mu \to \tilde{\mathbf{A}}$) the result is different. The second form of the equation, where the ordering of the products are reversed, is used here. The matrix equation resulting from element $(1,1)$ scalar equation is therefore $G^e_0(\tilde{\mathbf{A}}) = G^e_{0,1}(\tilde{\mathbf{A}})G^e_{0,2}(\tilde{\mathbf{A}}) + G^o_{1,1}(\tilde{\mathbf{A}})G^o_{0,2}(\tilde{\mathbf{A}})$.
3) Explicit calculation using the 4 matrix equations from the previous step will give $G^e_0(\tilde{\mathbf{A}})$, etc. that is similar to equations (33) - (36).
4) Obtain the explicit expressions for the submatrices that make up $\mathbf{T}_S$ in (19) from the results in the previous step.
5) Compare the result from the previous step with the explicit multiplication of $\mathbf{T}_{S,1}$ and $\mathbf{T}_{S,2}$ to complete the proof.

C. Matrix products

Expressions of the matrix products, such as $H^e_{0,2}(\tilde{\mathbf{A}})H^e_{0,1}(\tilde{\mathbf{A}})$, in term of the submatrices, such as $H^{e,u}_{0,2}(\tilde{\mathbf{A}})$ or $H^{e,u}_{0,1}(\tilde{\mathbf{A}})$ are derived here. The matrices have the form as shown in (31) or (32). Generic symbols for the matrices, such as $H^e_{i,j}(\tilde{\mathbf{A}})$, and submatrices, such as $H^{e,l}_{i,j}(\tilde{\mathbf{A}})$, will be used. The first subscript i $(=0,1)$ denote whether the filter is low-pass $(i=0)$ or high-pass $(i=1)$. The second subscript j $(=1,2)$ denote whether the matrices are from the first polyphase matrix $\mathbf{P}_{a,1}(\mu)$ $(j=1)$ or second polyphase matrix $\mathbf{P}_{a,2}(\mu)$ $(j=2)$. By explicit multiplication of matrices of the form shown in (31) or (32), the following types of products can be obtained:

$$H^e_{i,j}(\tilde{\mathbf{A}})H^e_{m,n}(\tilde{\mathbf{A}}) =$$

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

$$H^e_{i,j}(\tilde{\mathbf{A}})H^o_{m,n}(\tilde{\mathbf{A}}) =$$

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

$$H^o_{i,j}(\tilde{\mathbf{A}})H^e_{m,n}(\tilde{\mathbf{A}}) =$$

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

$$H^{o}_{i,j}(\tilde{\mathbf{A}})H^{o}_{m,n}(\tilde{\mathbf{A}}) =$$

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

for $i, m = 0, 1$ and $j, n = 1, 2$. Note that two of the matrices are block diagonal and the other two are block anti-diagonal.

References

[1] 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.

[2] 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.

[3] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[4] "Cooperative Special Issue on Graph Signal Processing," *IEEE J. of Selected Topics in Signal Processing*, vol. 11, no. 6, September 2017.

[5] "Cooperative Special Issue on Graph Signal Processing," *IEEE Trans. Signal and Information Proc. over Networks*, vol. 3, no. 3, September 2017.

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

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

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

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

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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

[19] D. B. H. Tay and A. Ortega, "Bipartite Graph Filter Banks: Polyphase Analysis and Generalization," *IEEE Trans. Signal Proc.*, vol. 65, no. 18, pp. 4833–4846, September 2017.

[20] 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.

[21] W. Sweldens, "The lifting scheme: A custom-design of biorthogonal wavelets," *Applied Computational and Harmonic Analysis*, vol. 3, no. 2, pp. 186–200, 1996.