DESIGN OF SPECTRAL GRAPH WAVELETS AND FILTER BANKS

Akie Sakiyama

Department of Bio-Functions and Systems Science

Graduate School of Bio-Applications and Systems Engineering

Tokyo University of Agriculture and Technology

March 2018

Copyright 2018 Akie Sakiyama

Acknowledgments

It is a pleasure to thank those who made this dissertation possible: my teachers, collaborators, family, and friends. First and foremost, I would like to thank my advisor, Professor Yuichi Tanaka for his expertise, continuous guidance, and understanding throughout the course of my studies at the Tokyo University of Agriculture and Technology (TUAT). This dissertation would not have been possible without his help and support. It has been a pleasure and privilege to work with Professor Yuichi Tanaka.

I would also like to thank Professor Takafumi Saito, Professor Toshihisa Tanaka, Professor Hiroshi Ishida, and Professor Izumi Nishidate for serving as my dissertation committee members. Their insightful comments have been a great help in improving this dissertation.

I also thank my collaborator in this research: Kana Watanabe, for her help which is shaped parts of this dissertation. Furthermore, I owe special thanks to Professor Antonio Ortega from the University of Southern California, Professor Toshihisa Tanaka, Professor David B. H. Tay from the Deakin University and Saho Yagyu, with whom I collaborated on different projects, for giving me a wonderful opportunity to collaborate with you.

I am grateful to colleagues in MSP Lab who taught me many things through various interactions and discussions. Especially, thanks go to Dr. Masaki Onuki for all the wonderful discussions and collaborations.

I greatly appreciate the financial support provided by the Japan Society for the Promotion of Science (JSPS): Research Fellow (DC1) grant.

Finally, and most importantly, I would also like to thank my father, my mother, my brother, my sister and my beloved dog for their constant support and unconditional love.

Abstract

This dissertation addresses the problem on designing spectral graph wavelets and filter banks in order to efficiently analyze and process graph signals.

Spectral graph wavelets and filter banks are one of the fundamental tools for signal processing on graphs. As in the case in traditional signal processing, they can capture features of graph signals by dividing the input signal into some different frequency bands. They are effective to represent signals sparsely since they have basis localized in the vertex domain unlike the graph Fourier transform. Furthermore, they are expected to be useful for many practical applications such as signal denoising, segmentation and compression. Therefore, designs of the efficient spectral graph wavelets and filter banks are an important problem for graph signal processing. Although there exist several conventional transforms, they have several disadvantages: 1) they have strong limitations on filter designs or redundancies, 2) decimated transforms are applicable only to the signal defined on a bipartite graph, and 3) their filter characteristics in graph frequency domain are not enough, i.e., they does not have all desired properties for spectral graph wavelets and filter banks such as tight frame, filters defined by smooth functions, and no DC leakage. To solve these problems, this dissertation proposes three effective approaches about spectral graph wavelets and filter banks: i) M-channel oversampled spectral graph filter bank which have low redundancies and high flexibility in filter design, ii) oversampled graph Laplacian matrix which enables us to apply arbitrary graph signals into decimated

transforms, and iii) the method of constructing spectral graph wavelets and filter banks that having all desired properties. This dissertation is organized as follows.

Chapter 1 describes the background, related works and objective of this dissertation.

The notations and preliminaries on graph signal processing are stated in Chapter 2.

Chapter 3 introduces the overall scheme of the graph spectral wavelets and filter banks and shows popular conventional approaches.

In Chapter 4, M-channel oversampled spectral graph filter banks are proposed. Since they are decimated transform with M filters, they have lower redundancies and have more flexibility in their design than the conventional spectral graph filter banks. The perfect reconstruction conditions of the oversampled spectral graph filter banks are shown. Some design examples indicate that the proposed filter banks have good stopband attenuation.

In Chapter 5, the oversampling method for graph signals is presented. It appends the nodes and edges to the original graph before performing graph filter banks. The effective oversampling method is also proposed that can make one bipartite graph that includes all edges of the original non-bipartite graph. It enables us to apply the non-bipartite graphs into any decimated transforms. Furthermore, the theoretical relationship between the proposed oversampling method and a covering method in graph theory is clarified.

Chapter 6 shows the method related to the effective spectral graph filter design. It is proven that any real-valued linear phase finite impulse response filter banks for regular signals can be reused as the filter banks for graph signals. The spectral graph filter bank has the same filter characteristic as the corresponding classical one and inherits the original properties, such as tight frame, smoothness and no DC leakage. Since the filters are defined by a sum of sinusoidal waves, they produce low approximation errors even if we use a lower-order shifted Chebyshev polynomial approximation for acceleration, and the upper bound of the error can be calculated rigorously. It is easily to design the spectral graph filter banks from the regular signal processing counterparts, such as Cohen-Daubechies-Feauveau wavelets, the discrete cosine transform, and the lapped orthogonal transform.

Chapter 7 concludes this dissertation and describes the future works.

The numerical performances of the proposed methods are evaluated by the experiments on denoising and non-linear approximation in each chapter. The proposed methods outperform the conventional decimated and critically sampled spectral graph wavelets and filter banks in all experiments.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the development of computer technology, and the cost reduction and miniaturization of devices, there have been considerable interests for analyzing or processing irregular and high-dimensional data in many fields, including intelligent infrastructure, physical infrastructure network like sensor networks, co-authorship studies, and neural networks. Graph signal processing has been developed to respond to these demands [1].

Graph is a generic data structure that can represent complex relationships among data and can be used in many fields of engineering and science. It consists of nodes and edges, and each edge is usually assigned a weight determined by the similarity of the nodes, e.g., physical or feature space distance between nodes in the network. In graph signal processing, a sample is placed on each node of a graph. Graph signal processing can explicitly consider the structure of the signal, unlike regular signal processing[1].

Graph signal processing is a relatively new field that has been extensively studied since around 2011. It has been a hot topic in signal and information processing for both theoretical and practical reasons [1–10]. From a theoretical viewpoint, it is related to signal processing, information theory [11], (spectral) graph theory [12], and computational harmonic analysis [13, 14]. Moreover, from a practical viewpoint, it has been used on an extensive amount of data with irregular structures, e.g., sensor and brain networks [15–17], traffic [18], learning [19–25], and images [26–29].

[1] Regular signal processing indicates the traditional digital signal processing.

The aim of graph signal processing is to achieve effective analysis and processing of graph signals as in the case of regular signal processing, that is useful to analyze regular-structured data, such as audio, image, video, and radar. Graph signal processing is also expected to be useful in many practical applications including compression, restoration, transmission and interpolation of data. To achieve the aim, the fundamental tools used in regular signal processing, including Fourier transform, filtering, and sampling, should be extended to graph signals.

Among them, wavelets and filter banks are one of the most important techniques. They consist of filters and sampling operators, and divide the input signal into several different frequency bands. The transforms are localized both in time/spatial and frequency domains, whereas Fourier transform has global basis in time/spatial domain. Therefore, wavelets and filter banks can provide a compact representation of a signal if the primary information of the signal is localized in time/spatial domain. They are effective for analyzing and processing data and useful for many applications such as denoising, segmentation, deconvolution and compression.

Wavelets and filter banks would be also effective for analyzing and processing graph signals and expected to be key techniques for applying graph signal processing into real problems on irregular and high-dimensional data, such as compression and denoising. However, researches on transforms for graph signals are in progress and there exist many issues to be solved. The main issues come from the fact that they should consider the data structures during filtering and sampling operations.

1.1 Related Works

The desired properties of graph wavelets and filter banks are listed as follows:

1. *Bandpass and highpass filters should have no DC leakage.* DC component is important information of a graph signal, and should not be affected during transform in many applications. Therefore, it is desirable that the lowpass filter passes all the DC energy.

2. *Transform should form tight frames (for undecimated transform) or orthogonal expansions (for decimated transform).* The filter set forming tight frame or transform with orthogonal expansion can avoid the calculation of the pseudo inverse of the analysis transform for reconstruction.

3. *Filters should be parametrized with smooth or polynomial functions.* The filters defined by polynomial functions can avoid the full eigendecomposition of the matrix and have localization in graph vertex domain. To obtain these benefits, the Chebyshev polynomial approximation [30] is often applied for filters defined by non-polynomial functions. If the filters are defined by smooth functions, they show small errors even when we use a lower-order Chebyshev polynomial approximation.

4. *Transform should be applicable to any graphs and variation operators.* The transform should be available to any graph signals regardless of their structure and should not modify the original relationships between the samples. There exist several variation operators, that are the basis of all tools for graph signal processing as the matrix representation of graph. They include combinatorial/normalized graph Laplacian and adjacency matrix. They should be selectable according to the applications.

There exist several conventional spectral graph wavelets and filter banks [13,16,31–38]. They are summarized in Table 1.1 with their properties. They are classified into *undecimated* and *decimated*.

Table 1.1: Conventional Perfect Reconstruct Spectral Graph Wavelet and Filter Banks with These Properties, Acronyms, And Corresponding Sections. The following properties are evaluated. UD: Undecimated, CS: Critically Sampling, DC: No-DC leakage, OE: Tight Frame or Orthogonal Expansion, SD: All filter kernels are Defined by Smooth or Polynomial Function, GS: Applicable to Any Graph Signals. Acronyms indicate SP: Cubic Spline, TM: Tight-Meyer, WH: Warped-Hann, UH: Uniform-Hann, QMF: Quadrature Mirror Filter, And SF: Spectral Factorization.

Method	Type	DC	OE	SD	GS	Acronym	Section
SGWT [13]	UD	✓			✓	UD-SP	III-A-1
Tight-Meyer [16]		✓	✓		✓	UD-TM	
Uniform-Hann [31]			✓	✓	✓	UD-UH	III-A-2
Log-Warped-Hann [31]		✓	✓		✓	UD-WH	
GraphQMF [32]	CS	✓	✓			CS-QMF	III-B-1
GraphBior [33]		✓	✓	✓		CS-SF	

The undecimated transforms have high flexibility in their design, i.e., they can use any filters with perfect reconstruction as long as they form a frame. However, they have high redundancy; the redundancy ρ is $\rho = M$, where M is the number of filters, and require a large amount of computations and memory. Spectral graph wavelet transform (SGWT) [13] firstly provided a framework of a graph wavelet with spectral graph filters, and the kernels can properly decompose the input signal into each frequency subband. However, it does not form tight frame and does not defined by smooth functions. Tight-Meyer [16] extends the kernels of SGWT to form a tight frame. However, the kernels are also not defined by smooth functions. Uniform-Hann and log-warped-Hann (octave-Hann) have been proposed in [31]. Uniform-Hann is defined by smooth functions and forms tight frame, but it produces DC leakage. Log-warped-Hann forms tight frame and does not produce DC leakage, but the kernels are not smooth functions.

The decimated transforms have a downsampling operation, and their redundancy is $\rho < M$. The conventional decimated transforms have strong limitations on their filter designs because of the critically sampled structure ($\rho = 1$). The downsampling of graph signal requires very careful manipulations. Unlike sampling for regular signals,

Table 1.2: Proposed Methods with These Acronyms And Corresponding Sections: Undecimated (UD), Critically Sampled (CS), Oversampled (OS), Spectral Factorization (SF), Frequency Conversion (FC), And Paraunitary (PU).

Method	Type	DC	OE	SD	GS	Acronym	Section
OSGFB	OS	✓		✓	–	OS-SF	Chap. 5
OS w/ OSGLM		properties depend on combined filters			✓	filter name + (OSGLM)	Chap. 6
CS w/ OSGLM							
FC: OSPRFB [39]		✓	✓	✓	–	OS-PU-FC	V-B-3
FC: CDF 9/7 [40,41]	CS	✓	✓	✓	–	UD-9/7-FC	V-A-1
FC: CDF 5/3 [40,41]		✓	✓	✓	–	UD-5/3-FC	
FC: CDF 4/4 [40,41]		✓	✓	✓	–	CS-4/4-FC	
FC: DCT [42–44]	UD	✓	✓	✓	✓	UD-DCT-FC	V-A-2
FC: LOT [45,46]		✓	✓	✓	✓	UD-LOT-FC	

the sampling points of graph signal are not determined uniquely, and the relationship between the frequency of the original signal and that of the signal after downsampling-then-upsampling is not clarified, except for the case of bipartite graphs. Only when the underlying graph is bipartite, the downsampling-then-upsampling operation causes spectral folding phenomenon that is similar to aliasing effect. Therefore, the conventional spectral graph transforms with decimations can only be applicable to signal defined on the bipartite graph. The famous spectral graph transforms with decimations are graphQMF [32] that is orthogonal transform but not localized in vertex domain, and graphBior [33] that is biorthogonal transform and localized both in vertex and graph spectral domains.

1.2 Contributions

The objective of this dissertation is to propose effective design methods of filter banks and wavelets for graph signals. The contributions of the proposed transforms are summarized as follows:

1. They can control the tradeoff between the redundancies and limitations on the filter design.

2. The decimated version of the transforms can also be applied to graph signals defined on any non-bipartite graphs.

3. They have all desirable properties listed above, i.e., no DC leakage, tight frame, smooth functions and applicability for signals on any graphs.

The transform satisfying above properties is realized by using three methods: two kinds of the oversampled transforms for graph signals and a novel and effective filter design method.

One of the oversampled transform is the M-channel spectral graph filter banks which have M filters and sampling operations. They would be useful for graph signals, since oversampled filter banks for regular signals have more freedom in their design and it has been shown that they outperform critically sampled systems in several applications [39, 47–51]. In fact, they have higher flexibility on their design than the conventional decimated spectral graph filter banks and have lower redundancies ($\rho = M/2$) than the undecimated transforms. We show the design examples of the filter set which satisfy the perfect reconstruction condition and have good stopband attenuation.

Another oversampled transform expands the signal on graphs as well as the underlying graph. It uses an oversampled graph Laplacian matrix (OSGLM), which adds the nodes and edges to the original graph. We consider the effective oversampling method that can construct one bipartite graph including all edges in the original non-bipartite graph. The decimated transform with the OSGLM can be applied to any graph signals even if the original graph is non-bipartite. Furthermore, we clarify the theoretical relationships between the proposed oversampling method and the graph covering method proposed in the context of graph theory.

Finally, we design the spectral graph filters defined by a sum of sinusoidal waves. The advantages of these filters are (a) they have low approximation errors even if a lower-order shifted Chebyshev polynomial approximation is used, (b) the upper bound of the error after the pth order Chebyshev polynomial approximation can be calculated rigorously without complex calculations, and (c) their parameters can be efficiently obtained from any real-valued linear phase finite impulse response filter banks in regular signal processing. The proposed filter bank has the same filter characteristics as the corresponding classical filter bank in the frequency domain and inherits the original properties, such as tight frame and no DC leakage. Furthermore, their approximation orders can be determined from the desired approximation accuracy. We show the graph filters based on the famous filters in regular signal processing: discrete cosine transform (DCT) [42–44, 52], lapped orthogonal transform (LOT) [45, 53, 54] and Cohen-Daubechies-Feauveau (CDF) wavelets [40,41]. They satisfy the desired properties listed above: no-DC leakage, tight-frame, smooth function, and perfect reconstruction.

The proposed methods are summarized in Table 1.2. The OSGLM is preprocessing and is used by combining with arbitrary decimated transforms. Therefore, their properties on DC, OE and SD depend on the filters used. The proposed decimated transforms with OSGLM satisfy GS and those without OSGLM do not satisfy GS.

1.3 Outline

Chapter 2 shows the notations and preliminaries on graph signal processing. The conventional spectral graph wavelets and filter banks are described in Chapters 3. Chapter 4 and 5 introduce the oversampled spectral graph filter banks: M-channel spectral graph filter banks and oversampling method for graph signals. Chapter 6 constructs the filter banks based on the perfect reconstruction linear phase FIR filter banks in regular signal processing. Finally, Chapter 7 concludes this dissertation.

Chapter 2

Notation and Preliminaries on Graph Signal Processing

This chapter shows the definitions and preliminaries which are useful for understanding the spectral graph wavelets and filter banks.

2.1 Graph

A graph $\mathcal{G}$ is represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_0, v_1, \dots, v_{N-1}\}$ and $\mathcal{E}$ denote sets of nodes and edges, respectively. We will only consider a finite undirected graph with no loops or multiple links. The number of nodes is $N = |\mathcal{V}|$, unless otherwise specified. The proposed graph spectral transforms are defined by using graph Laplacian matrix which is calculated from the graph adjacency matrix and graph degree matrix. The (m, n)-th element of the adjacency matrix $\mathbf{A}$ is defined as follows:

$$A(m,n) = \begin{cases} w_{mn} & \text{if nodes } m \text{ and } n \text{ are connected,} \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where w_{mn} denotes the weight of the edge between m and n. The degree matrix $\mathbf{D}$ is a diagonal matrix, and its m-th diagonal element is

$$D(m,m) = \sum_n w_{mn}. \tag{2.2}$$

The combinatorial graph Laplacian matrix (GLM) is defined as $\mathbf{L} := \mathbf{D} - \mathbf{A}$, and the symmetric normalized GLM is $\boldsymbol{\mathcal{L}} := \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. The symmetric normalized GLM has the property that its eigenvalues are within the interval $[0, 2]$. For spectral analysis of graph signals, we can use the eigenpairs of arbitrary variation matrix, such as adjacency matrix, graph Laplacian matrix and random walk matrix. Although we will mainly use $\boldsymbol{\mathcal{L}}$ in this dissertation, $\mathbf{L}$ can also be used for undecimated spectral graph filter banks in the same way. The eigenvalues of $\boldsymbol{\mathcal{L}}$ are λ_i and ordered as: $\lambda_{\text{min}} = 0 = \lambda_0 < \lambda_1 \leq \lambda_2 \ldots < \lambda_{N-1} = \lambda_{\text{max}} \leq 2$ without loss of generality[1]. The eigenvector $\boldsymbol{u}_i$ corresponds to λ_i and satisfies $\boldsymbol{\mathcal{L}}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$. The eigenvectors $\mathbf{U} = [\boldsymbol{u}_{\lambda_0}, \ldots, \boldsymbol{u}_{\lambda_{N-1}}]$ satisfy

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}_N, \tag{2.3}$$

where $\cdot^T$ is the transpose of a matrix or a vector and $\mathbf{I}_N$ is an $N \times N$ identity matrix.

2.2 Graph Signal and Spectrum

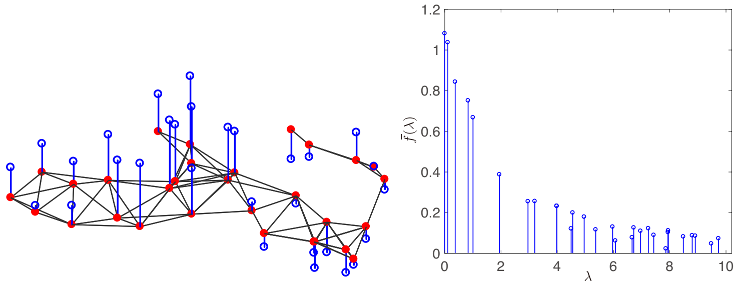
Graph signal has samples on each node and is defined as $\boldsymbol{f} \in \mathbb{R}^N$. This section introduces the Fourier transform for graph signals. The traditional Fourier transform for regular signals is defined as the expansion of a function $f(t)$ in terms of the complex exponentials:

$$F(\omega) := \langle f, e^{j\omega t} \rangle = \int_{\mathbb{R}} f(t) e^{-j\omega t} dt, \tag{2.4}$$

where $\omega = 2\pi\xi$ and ξ indicates frequency. In this equation, $e^{j\omega t}$ is the eigenfunction of one-dimensional Laplace operator:

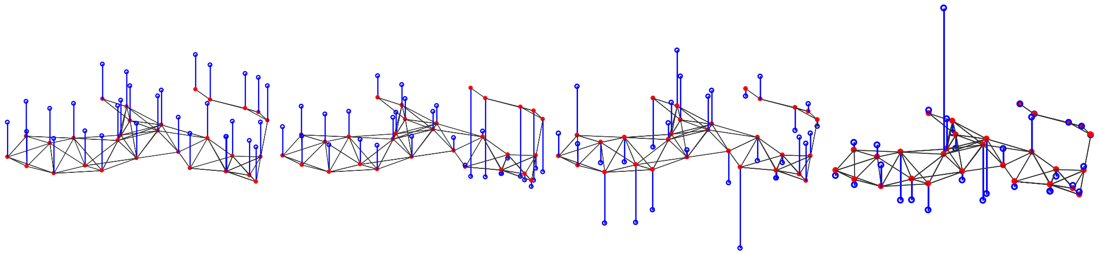
$$-\frac{\partial^2}{\partial t^2} e^{j\omega t} = \omega^2 e^{j\omega t}. \tag{2.5}$$

[1] The eigenvalue λ_1 will be nonzero only if the graph is connected. $\lambda_{N-1} = 2$ only for bipartite graphs.



(a) (b)

Figure 2.1: Graph signal and its spectrum. (a) signal on random network graph (vertex domain). The black lines, red circles, and blue lines indicate the edges, nodes, and samples, respectively. (b) spectrum of (a) (graph spectral domain).



(a) $i = 0$ (b) $i = 1$ (c) $i = 9$ (d) $i = 29$

Figure 2.2: ith eigenvector placed on the corresponding graph shown in Fig. 2.1.

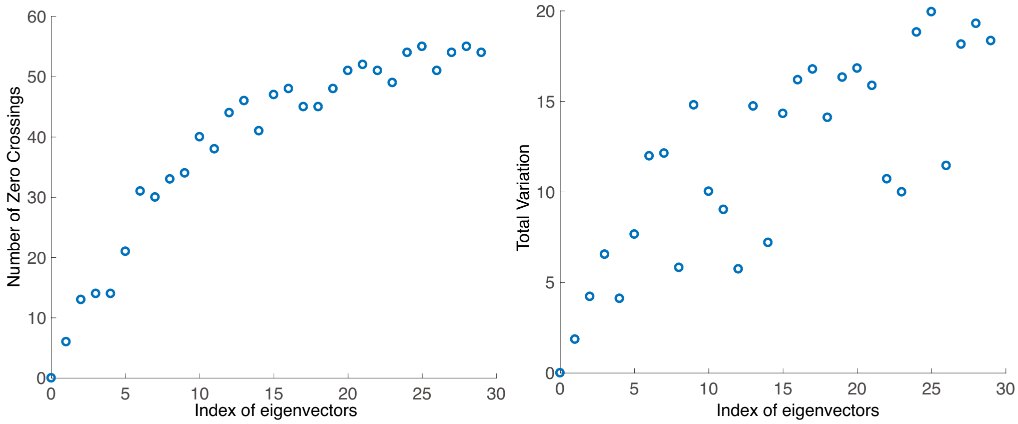
Based on the above definitions, the graph Fourier transform is defined as the expansion of a function $\boldsymbol{f}$ in terms of the eigenvectors of the graph variation operators [12,13]:

$$\overline{f}(i) = \langle \boldsymbol{u}_i, \boldsymbol{f} \rangle = \sum_{n=0}^{N-1} u_i^*(n) f(n). \tag{2.6}$$

where $\cdot^*$ is the complex conjugate. The inverse graph Fourier transform is defined as

$$f(i) = \langle \boldsymbol{u}_i^*, \overline{\boldsymbol{f}} \rangle = \sum_{n=0}^{N-1} u_i(n) \overline{f}(n). \tag{2.7}$$

An example of graph Fourier transform and its Fourier basis (eigenvectors) are shown in Figs. 2.1 and 2.2. It can be seen as the eigenvector associated with large eigenvalue



(a)

(b)

Figure 2.3: Number of zero crossing and total variation of eigenvectors corresponding graph shown in Fig. 2.1. (a) number of zero crossing. (b) total variation.

corresponds to high frequency and oscillates rapidly. In contrast, that associated with small eigenvalue, i.e., close to 0, corresponds to low frequency. This fact is evaluated by the number of zero crossing and total variation of each eigenvectors. The set of zero crossing and total variations for $\boldsymbol{u}$ are defined as

$$\mathcal{Z}_{\mathcal{G}}(\boldsymbol{u}) = \left\{ e = (i,j) \in \mathcal{E} : u(i)u(j) \right\}, \tag{2.8}$$

and

$$\mathrm{TV}(\boldsymbol{u}) = \sum_{i \in \mathcal{V}} \left(\sum_{j \in \mathcal{V}} w_{i,j} \left(u(j) - u(i) \right)^2 \right)^{\frac{1}{2}}, \tag{2.9}$$

respectively. The numbers of zero crossing and total variations of eigenvectors are shown in Fig. 2.3.

The entire spectrum of $\mathcal{G}$ is defined by $\sigma(\mathcal{G}) := \{\lambda_0, \ldots, \lambda_{N-1}\}$. The projection matrix for the eigenspace V_λ is

$$\mathbf{P}_\lambda = \sum_{\lambda_i = \lambda} \boldsymbol{u}_i \boldsymbol{u}_i^T. \tag{2.10}$$

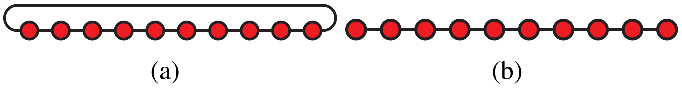


Figure 2.4: (a) ring graph. (b) path graph.

If λ_i and λ_j are different eigenvalues, $\mathbf{P}_{\lambda_i}$ and $\mathbf{P}_{\lambda_j}$ are orthogonal; that is,

$$\mathbf{P}_{\lambda_i}\mathbf{P}_{\lambda_j} = \delta(\lambda_i - \lambda_j)\mathbf{P}_{\lambda_i}, \tag{2.11}$$

where $\delta(\lambda)$ is the Kronecker delta function.

2.2.1 Connection With Traditional Signal Processing

Some graphs have the same basis as the famous transform in traditional signal processing. Ring graph shown in Fig. 2.4 (a) has following combinatorial graph Laplacian matrix:

$$\mathbf{L}_{\text{ring}} = \begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix} \tag{2.12}$$

Its eigenvectors are coincide with the basis of discrete Fourier transform (DFT)

$$\boldsymbol{u}_k = [1, w^k, w^{2k}, \ldots, w^{(N-1)k}]^T, \quad k = 0, \ldots, N-1, \tag{2.13}$$

where $w = e^{j\frac{2\pi}{N}}$.

The another graph is path graph shown in Fig. 2.4 (b). The combinatorial graph Laplacian is

$$\mathbf{L}_{\text{path}} = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}. \tag{2.14}$$

Its eigenvectors are coincide with the basis of discrete cosine transform:

$$u_k(n) = \cos\left(\left(n + \frac{1}{2}\right)\frac{k\pi}{N}\right). \tag{2.15}$$

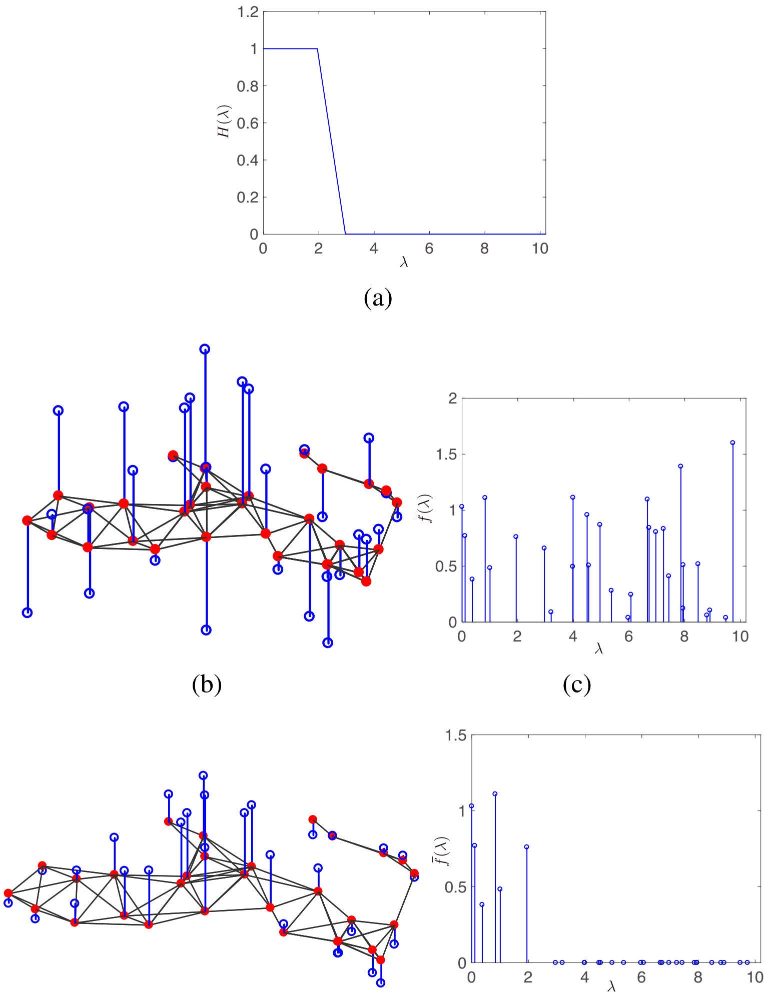
2.2.2 Spectral Graph Filtering and Chebyshev Polynomial Approximation

The spectral domain filtering for graph signals can be written as

$$\begin{aligned} \mathbf{H}\boldsymbol{f} &= \mathbf{U}H(\boldsymbol{\Lambda})\mathbf{U}^T\boldsymbol{f} \\ &= \sum_{\lambda\in\sigma(\mathcal{G})} H(\lambda)\mathbf{P}_\lambda\boldsymbol{f}, \end{aligned} \tag{2.16}$$

where $H(\cdot)$ is a spectral filter kernel. The example of the kernel of lowpass filter and, input and filtered signals are shown in Fig. 2.5. The lowpass filter is defined as

$$H(\lambda_i) = \begin{cases} 1 & \text{if } i < 7 \\ 0 & \text{otherwise.} \end{cases} \tag{2.17}$$



(a)

(b) (c)

(d) (e)

Figure 2.5: Filtering example. (a) filter kernel, (b) input signal, (c) graph spectrum of (b), (d) output signal, (e) graph spectrum of (d).

A filter kernel defined by polynomial function has some advantages. One is that it can avoid calculating the exact filter responses through a full eigendecomposition of a given Laplacian matrix $\boldsymbol{\mathcal{L}}$ (or $\mathbf{L}$). Spectral filtering with filters defined by p-th order polynomial function only needs $\mathcal{O}(p|\mathcal{E}|)$, whereas those with filters defined by non-polynomial function needs $\mathcal{O}(N^3)$ computational complexity. The another advantage is the graph filters defined by polynomial are localized in the vertex domain, because if the graph spectral filter is a p degree polynomial, i.e., $H(\lambda_i) = \alpha_p \lambda_i^p$ it is p-hop localized[2] in the vertex domain:

$$
\begin{aligned}
f_{\text{out}}(k) &= \sum_{i=0}^{N-1} \overline{f}_{\text{in}}(\lambda_i) H(\lambda_i) u_i(k) \\
&= \sum_{l=0}^{N-1} f_{\text{in}}(l) \sum_{p=0}^{K} \alpha_p \sum_{i=0}^{N-1} \lambda_i^p u_i^*(l) u_i(k) \\
&= \sum_{l=0}^{N-1} f_{\text{in}}(l) \sum_{p=0}^{K} \alpha_p (\boldsymbol{\mathcal{L}}^p)_{kl}.
\end{aligned}
\tag{2.18}
$$

The pth order shifted Chebyshev polynomial is often used to approximate $H(\lambda)$ as $\widetilde{H}_p(\lambda) = \alpha_p \lambda^p$ [1, 13][3] for graph spectral filtering. The signal filtered by $\widetilde{H}_p(\lambda)$ is represented as

$$
\boldsymbol{f}_{\text{out}} = \widetilde{H}_p(\boldsymbol{\mathcal{L}}) \boldsymbol{f}_{\text{in}} = \left\{ \frac{1}{2} c_0 + \sum_{i=1}^{p} c_i \overline{T}_i(\boldsymbol{\mathcal{L}}) \right\} \boldsymbol{f}_{\text{in}}, \tag{2.19}
$$

where $\overline{T}_0(\boldsymbol{\mathcal{L}}) = 1, \overline{T}_1(\boldsymbol{\mathcal{L}}) = 2(\boldsymbol{\mathcal{L}} - 1)/\lambda_{\text{max}}, \overline{T}_i(\boldsymbol{\mathcal{L}}) = 4(\boldsymbol{\mathcal{L}} - 1)\overline{T}_{i-1}(\boldsymbol{\mathcal{L}})/\lambda_{\text{max}} - \overline{T}_{i-2}(\boldsymbol{\mathcal{L}})$, and

$$
c_i = \frac{2}{S} \sum_{m=1}^{S} \cos\left(\frac{\pi i \left(m - \frac{1}{2}\right)}{S} \right) H_k \left(\frac{\lambda_{\text{max}}}{2} \left(\cos\left(\frac{\pi \left(m - \frac{1}{2}\right)}{S} \right) + 1 \right) \right) \tag{2.20}
$$

for $i = 0, \ldots, p$, where S is the number of sampling points used to compute the Chebyshev coefficients and is usually set to $S = p + 1$.

[2] If the diameter $d_{\mathcal{G}}$ of graph $\mathcal{G}$ is $d_{\mathcal{G}} \leq p$, the graph spectral filter with a p degree polynomial becomes a global operation.

[3] Although designing graph spectral wavelet and filter banks requires λ_{max}, λ_{max} can be easily estimated by using the power method [55] or the Arnoldi algorithm [34].

The overall computational complexity in applying an spectral graph filter with a Chebyshev polynomial approximation is $\mathcal{O}(p(|\mathcal{E}| + N))$. $\mathcal{O}(p|\mathcal{E}|)$ is the cost of computing the Chebyshev polynomials $T_k(\boldsymbol{\mathcal{L}})\boldsymbol{f}$, and $\mathcal{O}(pN)$ is the cost of computing the coefficients for each scale.

Chapter 3

Wavelets and Filter Banks in Regular and Graph Signal Processing

This section describes the wavelets and filter banks for regular and graph signals. The wavelets and filter bank are sets of filters, sometimes linked by sampling operators. The basis are localized both in time/spatial/vertex domain and frequency domain. We show the overall scheme of the transforms and the typical filters for regular signals (Section 3.1) and for graph signals (Section 3.2).

3.1 Wavelets and Filter Banks for Regular Signals

Firstly, we introduce the wavelets and filter banks used in regular signal processing. The overall scheme of an M-channel filter bank with downsampling factor of P, where P is an integer, is shown in Fig. 3.1. When $M = P$, it is called a *critically sampled* filter bank. If $M > P$ then it is called an *oversampled* filter bank. $H_i(z)$ and $G_i(z)$ are arbitrary analysis and synthesis filters in z domain. $\downarrow P$ and $\uparrow P$ are the downsampling and upsampling operators with sampling factor of P. The downsampled signal $\boldsymbol{f}_\mathrm{d} \in \mathbb{R}^{N/P}$ with downsampling factor P can be represented as

$$f_\mathrm{d}(i) = f(Pi). \tag{3.1}$$

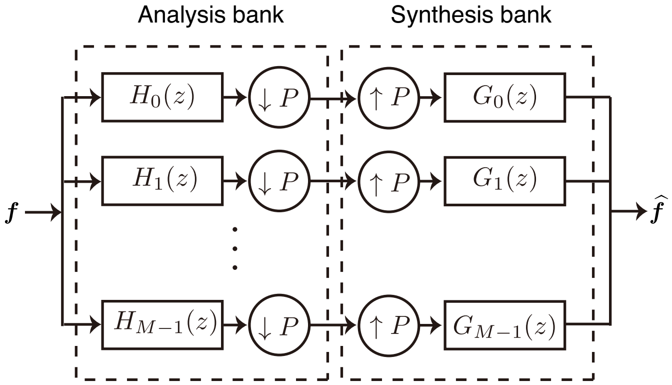


Figure 3.1: M-channel filter banks for regular signals with downsampling factor P.

The upsampling from $\boldsymbol{f} \in \mathbb{R}^{N/P}$ to $\boldsymbol{f}_{\mathrm{u}} \in \mathbb{R}^{N}$ is

$$f_{\mathrm{u}}(i) = \begin{cases} f(i/P) & \text{for mod}(i, P) = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

The most of the filter banks in regular signal processing uses $P = M > 2$. We show the famous transforms with $P = M > 2$, where the most of the filter banks in regular signal processing and with $P = 2$ which is corresponds to the critically sampled spectral graph wavelets (if $P = M$) and the proposed

3.1.1 Filter Banks With $P = M > 2$

The time domain matrix $\boldsymbol{\Psi}$ of an M-channel FIR filter bank is represented as [56]

$$\boldsymbol{\Psi} = \begin{bmatrix} \cdots & \cdots & \cdots & & & \\ & \boldsymbol{\Psi}_{D-1} & \cdots & \boldsymbol{\Psi}_0 & & \\ & & \boldsymbol{\Psi}_{D-1} & \cdots & \boldsymbol{\Psi}_0 & \\ & & & \cdots & \cdots & \cdots \end{bmatrix}. \tag{3.3}$$

where each block $\mathbf{\Psi}_d$, $0 \leq d \leq D-1$, is the $M \times P$ matrix

$$\mathbf{\Psi}_d = \begin{bmatrix} h_0(dP) & h_0(dP+1) & \dots & h_0(dP+(P-1)) \\ h_1(dP) & h_1(dP+1) & \dots & h_1(dP+(P-1)) \\ \vdots & & \ddots & \vdots \\ h_{M-1}(dP) & h_{M-1}(dP+1) & \dots & h_{M-1}(dP+(P-1)) \end{bmatrix} \tag{3.4}$$

where $\boldsymbol{h}_k$ is the k-th basis function. $\mathbf{\Psi}$ is a block Toeplitz matrix and has non-zero D blocks in each row. The polyphase matrix of the FIR filter bank is represented as

$$\mathbf{\Psi}_p(z) = \mathbf{\Psi}_0 + \mathbf{\Psi}_1 z^{-1} + \dots \mathbf{\Psi}_{D-1} z^{-(D-1)}. \tag{3.5}$$

In order for the filter bank to be orthogonal ($\mathbf{\Psi}_p(z)^T \mathbf{\Psi}_p(z) = \mathbf{I}_M$), $\mathbf{\Psi}$ should satisfy the following condition:

$$\sum_{i=0}^{D-1-l} \mathbf{\Psi}_i^T \mathbf{\Psi}_{i+l} = \delta(l)\mathbf{I}_M. \tag{3.6}$$

Discrete Cosine Transform

An example for the case of $D = 1$ is the DCT [42–44]. The (k, m)-th element of its $M \times M$ transform matrix $\mathbf{C}_{\text{DCT}}$ is represented as

$$C_{\text{DCT}}(k, m) = s_{\text{DCT}} \cos\left(\frac{(2m+1)\pi}{2M} k\right), \tag{3.7}$$

where

$$s_{\text{DCT}} = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{3.8}$$

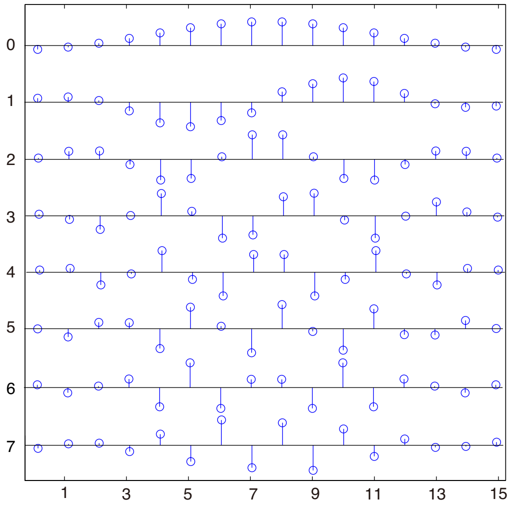


Figure 3.2: The basis function of the LOT with $M = 8$.

Lapped Orthogonal Transform

LOT is a linear phase FIR filter bank with $D = 2$. Its $M \times 2M$ transform matrix is defined as follows [45, 46]:

$$\mathbf{C}_{\text{LOT}} = \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{C}^e_{\text{DCT}} - \mathbf{C}^o_{\text{DCT}} & \mathbf{J}_{M/2}(\mathbf{C}^e_{\text{DCT}} - \mathbf{C}^o_{\text{DCT}}) \\ \mathbf{C}^e_{\text{DCT}} - \mathbf{C}^o_{\text{DCT}} & -\mathbf{J}_{M/2}(\mathbf{C}^e_{\text{DCT}} - \mathbf{C}^o_{\text{DCT}}) \end{bmatrix}, \tag{3.9}$$

where $\mathbf{C}^e_{\text{DCT}}$ and $\mathbf{C}^o_{\text{DCT}}$ are $M/2 \times M$ matrices of the symmetric and antisymmetric basis functions in the DCT. $\mathbf{J}_{M/2}$ is the $M/2 \times M/2$ counter-identify matrix. $\mathbf{V}$ and $\mathbf{W}$ are freely chosen orthogonal matrices. The basis functions with $M = 8$ are shown in Fig. 3.2.

Although this dissertation considers the above-mentioned M-channel linear phase FIR filter banks with filter lengths $L = DM$, those with $L = D_k M + \eta$, $k = 0, 1, \ldots, M-1$, where η is an arbitrary integer, have the same paraunitary constraints

[56] and can be applied to the proposed method. The detailed conditions of the M-channel linear phase perfect reconstruction filter bank with $L = D_k M + \eta$ are described in [57].

3.1.2 Filter Banks With $P = 2$

Two-Channel Wavelet Transform

In the z-domain, the perfect reconstruction condition of the two-channel discrete wavelet transform (DWT) can be expressed as [58]

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2z^{-l} \tag{3.10}$$

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0, \tag{3.11}$$

where z^{-l} is delay, $l = (L_0 + L_1 - 2)/2$, $H_i(z) = \sum_{m=0}^{L_i - 1} h_i(m) z^{-m}$, and L_0 and L_1 are the filter lengths of the lowpass filter $H_0(z)$ and the highpass filter $H_1(z)$, respectively. The filter lengths of the synthesis-side filters $G_0(z)$ and $G_1(z)$ are L_1 and L_0, respectively.

Theorem 1. *[59, Proposition 3.3], [58, Theorem 4.3] In a two-channel biorthogonal linear phase wavelet transform, the filter lengths are all odd or all even. The analysis filters $H_0(z)$ and $H_1(z)$ should be*

a) Both symmetric of odd length, differing by an odd multiple of 2.

b) One symmetric, the other antisymmetric[1], of even length, and equal to each other or differing by an even multiple of 2.

[1] Generally, lowpass filters are symmetric, and highpass filters are antisymmetric.

M-Channel Oversampled Filter Banks

For the sake of simplicity, this dissertation deals with oversampled filter banks whose filter lengths are $L = PK$, where P is a downsampling factor and K is an arbitrary integer, and the even-indexed filters are symmetric and the odd-indexed filters are anti-symmetric. The perfect reconstruction condition of the oversampled filter banks with a downsampling factor of $P = 2$ is expressed as [39, 60]

$$\sum_{k=0}^{M-1} G_k(z)H_k(z) = 2z^{-l} \tag{3.12}$$

$$\sum_{k=0}^{M-1} G_k(z)H_k(-z) = 0. \tag{3.13}$$

3.2 Spectral Graph Wavelets and Filter Banks

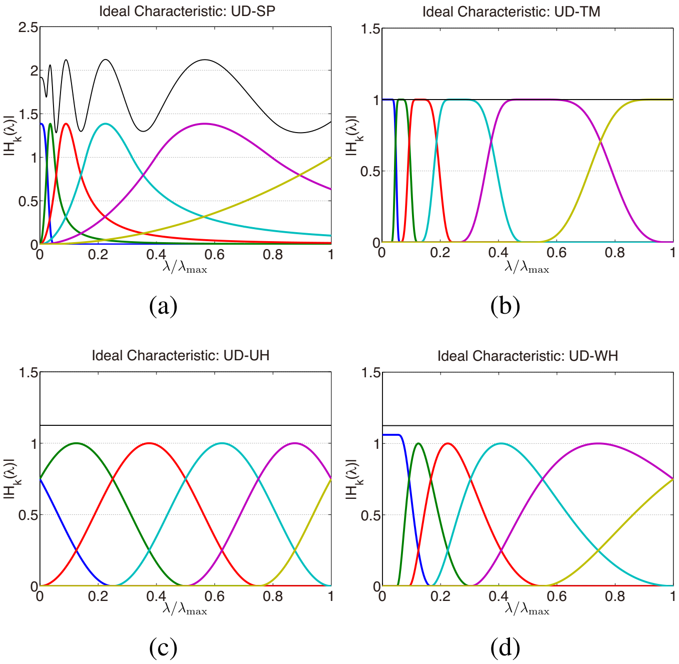
This section describes the existing approaches to design wavelets and filter banks in the graph spectral domain. The undecimated transforms are introduced in Section 3.1 and critically sampled transforms are in Section 3.2. The properties and acronyms of the conventional spectral graph wavelets and filter banks are summarized in Table 1.1.

3.2.1 Undecimated Transform

The undecimated transform divide the input signal into M different frequency subbands by using M spectral filters. Let us denote $\boldsymbol{\psi}_{k,n}$ as the spectral graph wavelet function in the vertex domain, at vertex n:

$$\psi_{k,n}(m) = \sum_{i=0}^{N-1} H_k(\lambda_i)u_i^*(n)u_i(m), \tag{3.14}$$

where $H_k(\lambda)$ is the spectral filter response of the kth filter in the graph spectral domain.



(a) (b)

(c) (d)

Figure 3.3: Ideal filter characteristics of spectral graph filter banks for $M = 6$ (black lines indicate $F(\lambda)$): (a) UD-SP [13]. (b) UD-TM [16]. (c) UD-UH ($\gamma = \lambda_{\max}$, $R = 3$, $K = 1$ and $\alpha_0 = \alpha_1 = 1/2$) [31]. (d) UD-WH [31].

The family of vectors $\mathcal{F} := \{\boldsymbol{\psi}_{k,n}\}_{k=0,\ldots,M-1,\ n=0,\ldots,N-1}$ is a frame of $\mathbb{R}^N$, if there exist two constants $B \geq A > 0$ such that

$$A\|\boldsymbol{f}\|_2^2 \leq \sum_{n=0}^{N-1}\sum_{k=0}^{M-1} |\langle \boldsymbol{f}, \boldsymbol{\psi}_{k,n}\rangle|^2 \leq B\|\boldsymbol{f}\|_2^2 \tag{3.15}$$

for all $\boldsymbol{f} \in \mathbb{R}^N$. If $A = B$, it is called a tight frame. In particular, the case of $A = B = 1$ is the Parseval frame. It conserves energy and has a simple reconstruction scheme; i.e., the analysis operator is also used as the synthesis operator [16].

Theorem 2. *([13, Theorem 5.6] [31, Lemma 1]) If $F(\lambda) := \sum_{k=0}^{M-1} |H_k(\lambda)|^2 > 0$ for all $\lambda \in \sigma(\boldsymbol{\mathcal{L}})$, $\mathcal{F} = \{\boldsymbol{\psi}_{k,n}\}_{k=0,\dots,M-1,\ n=0,\dots,N-1}$ is a frame with bounds,*

$$\begin{aligned} A &= \min_{\lambda \in \sigma(\boldsymbol{\mathcal{L}})} F(\lambda), \\ B &= \max_{\lambda \in \sigma(\boldsymbol{\mathcal{L}})} F(\lambda). \end{aligned} \tag{3.16}$$

If $F(\lambda)$ is constant for all $\lambda \in \sigma(\boldsymbol{\mathcal{L}})$, $\mathcal{F}$ is a tight frame.

Spectral Graph Wavelet Transforms

Graph wavelet kernels of the spectral graph wavelet transform (SGWT) have been designed on the basis of a cubic spline, so that the kernel $H_{\text{spline}}(\lambda)$ and its derivative $H'_{\text{spline}}(\lambda)$ are continuous [13]:

$$H_{\text{spline}}(\lambda) = \begin{cases} \lambda_a^{-\alpha}\lambda^{\alpha} & \text{for } \lambda < \lambda_a \\ s(\lambda) & \text{for } \lambda_a \le \lambda \le \lambda_b \\ \lambda_b^{\beta}\lambda^{-\beta} & \text{for } \lambda > \lambda_b \end{cases} \tag{3.17}$$

where α and β are integer parameters, λ_a and λ_b are parameters for the transition regions, $s(\lambda)$ is a cubic polynomial that satisfies $s(\lambda_a) = s(\lambda_b) = 1$, $s'(\lambda_a) = \alpha/\lambda_a$ and $s'(\lambda_b) = -\beta/\lambda_b$. The wavelet kernels are defined as $H_{k,\text{SGWT}} = H_{\text{spline}}(t_k\lambda),\ k = 1, \dots, M - 1$, where scales $\{t_k\}$ are logarithmically equispaced between the minimum scale $t_1 = \lambda_b/\lambda_{\text{min}}$ and the maximum scale $t_{M-1} = \lambda_b/\lambda_{\text{max}}$, $\lambda_{\text{min}} = \lambda_{\text{max}}/K$, and K is a design parameter. The scaling kernel is defined as $H_{0,\text{SGWT}}(\lambda) = \mu e^{-(\frac{\lambda}{0.6\lambda_{\text{min}}})^4}$, where μ is set so that $H_{0,\text{SGWT}}(0)$ has the maximum value of the wavelet kernels. Their graph frequency characteristics with $\alpha = \beta = 2$, $\lambda_a = 1$, $\lambda_b = 2$, $s(x) = -5 + 11x - 6x^2 + x^3$, and $K = 20$ are shown in Fig. 3.3(a). As can be seen from the figure, the SGWT is not tight, i.e., $G(\lambda)$ is not constant, so the pseudoinverse should be calculated for the synthesis transform.

Tight Mayer

To make the wavelet frames tight, several spectral graph wavelet kernels have been derived from regular wavelet kernels by transforming the variables [16]. As an example, Meyer-like kernels are defined as:

$$\begin{aligned} H_{\text{Meyer}}(\lambda) &= \begin{cases} \sin\left(\frac{\pi}{2}\nu\left(\frac{\lambda}{a}-1\right)\right) & \forall\lambda \in]a,2a] \\ \cos\left(\frac{\pi}{2}\nu\left(\frac{\lambda}{2a}-1\right)\right) & \forall\lambda \in]2a,4a] \\ 0 & \text{elsewhere} \end{cases} \\ H_{0,\text{Meyer}}(\lambda) &= \begin{cases} 1 & \forall\lambda \in [0,a] \\ \cos\left(\frac{\pi}{2}\nu\left(\frac{\lambda}{a}-1\right)\right) & \forall\lambda \in]a,2a] \\ 0 & \text{elsewhere} \end{cases} \\ \nu(x) &= x^4\left(35-84x+70x^2-20x^3\right) \end{aligned} \tag{3.18}$$

where $a \in \mathbb{R}^+$. The wavelet kernels are represented as $H_{k,\text{Meyer}} = H_{\text{Meyer}}(t_k\lambda)$ with the wavelet scaling $t_k = \frac{a}{\lambda_{\text{max}}}2^k,\ k = 1,\dots,M-1$. The graph frequency characteristics of Meyer-like kernels (tight-Meyer) with $a = 2/3$ are shown in Fig. 3.3(b). The kernels clearly form a tight frame.

Uniform/Octave Hann

Another tight graph wavelet frame was presented by Shuman et al. [31]. The kernel is defined as

$$H^{\text{Uniform}}_{\text{Half-cosine}}(\lambda) = \sum_{l=0}^{K} \alpha_l \cos\left(2\pi l \left(R'\lambda + \frac{1}{2}\right)\right) \mathbb{1}_{-\frac{1}{R'} \leq \lambda < 0}, \tag{3.19}$$

where $K < R/2$, $2 < R \leq M$, $R' = \frac{M+1-R}{R\gamma}$, γ is an upper bound on the spectrum, and R controls the overlap of the shifted kernels. The real sequence a_l satisfies $\sum_{l=0}^{K}(-1)^l \alpha_l = 0$. The k-th filter for an M-channel graph filter bank is represented on the basis of the kernel $H^{\text{Uniform}}_{\text{Half-cosine}}(\lambda)$, as

$$H^{\text{Uniform}}_{k,\text{Half-cosine}}(\lambda) = H^{\text{Uniform}}_{\text{Half-cosine}}\left(\lambda - \frac{1}{RR'}k\right). \tag{3.20}$$

The kernels form a tight frame:

$$G(\lambda) = R\alpha_0^2 + \frac{R}{2}\sum_{l=1}^{K} \alpha_l^2. \tag{3.21}$$

Figure 3.3(c) shows the shifted and scaled Hann kernel (uniform-Hann) $H^{\text{Uniform}}_{k,\text{Hann}}(\lambda)$ with $\gamma = \lambda_{\text{max}}$, which is obtained from the half-cosine kernel $H^{\text{Uniform}}_{k,\text{Half-cosine}}(\lambda)$ with $R = 3$, $K = 1$ and $\alpha_0 = \alpha_1 = 1/2$. It can be seen that the filter bank has a uniform band and that the DC component of the spectrum; i.e., the filter response at $\lambda = 0$, is shared by two filters.

A method of converting the UD-UH into octave-band spectral graph wavelets was also proposed in [31]. $M - 1$ wavelet kernels are constructed by warping the uniform filter kernels with a logarithmic warping function. As a result, the wavelet kernels and the scaling kernel are defined as

$$H_{k,\text{Hann}}^{\text{Octave}}(\lambda) = H_{k-1,\text{Hann}}^{\text{Uniform}}\left(\log(\lambda)\right),\ k = 1, \dots, M-1, \tag{3.22}$$

$$H_{0,\text{Hann}}^{\text{Octave}}(\lambda) = \sqrt{R\alpha_0^2 + \frac{R}{2}\sum_{l=1}^{K}\alpha_l^2 - \sum_{k=1}^{M-1}|H_{k,\text{Hann}}^{\text{Octave}}(\lambda)|^2}. \tag{3.23}$$

where $\gamma = \log(\lambda_{\text{max}})$ and the frame bounds are the same as those of the uniform kernels. The graph frequency characteristics (UD-WH) are shown in Fig. 3.3(d). We can see that the kernels still form a tight frame, but the scaling function will be different from a sinusoidal wave.

Furthermore, in [31], the spectrum-adapted filters are obtained using warping filters with a cumulative spectral density function. The warped filters are defined as $H_{k,\text{Hann}}^{\text{Warped}}(\lambda) = H_{k,\text{Hann}}^{\text{Uniform}}(\widetilde{\omega}(\lambda))$, where $\gamma = \widetilde{\omega}(\lambda_{\text{max}})$ and $\widetilde{\omega}(\lambda)$ is the spectral density-based warping function.

3.2.2 Decimated Transform

The conventional transforms with decimation have scheme shown in Fig. 3.4, where $\mathbf{H}_i$, and $\mathbf{G}_i$ for $i = 0, 1$ are analysis and synthesis filters, and $\downarrow$ S and $\uparrow$ S are down- and upsampling operations, respectively. The conventional transforms can only be applicable to the signals defined bipartite graph. A bipartite graph is two-colorable graph where nodes can be decomposed into two disjoint sets $\mathcal{L}$ and $\mathcal{H}$ such that every link connects a node in $\mathcal{L}$ to one in $\mathcal{H}$ and is represented as $\mathcal{G} = \{\mathcal{H}, \mathcal{L}, \mathcal{E}\}$.

The downsampling and upsampling operations are defined as follows. We consider the downsampling by remaining the samples on $\mathcal{L}$. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\} = \{\mathcal{H}, \mathcal{L}, \mathcal{E}\}$ and $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ be the original bipartite graph and the reduced-size graph, respectively, where $\mathcal{V}_1 = \mathcal{L}$ for downsampling operation keeping the samples on $\mathcal{L}$ and removing the samples on $\mathcal{H}$. The downsampling operator of a bipartite graph from $\boldsymbol{f} \in \mathbb{R}^{|\mathcal{V}|}$ to $\boldsymbol{f}_d \in \mathbb{R}^{|\mathcal{L}|}$ is defined as

$$f_d(n) = f(n') \text{ if } v_{n'} \in \mathcal{L} \text{ corresponds to } v_{1,n} \in \mathcal{V}_1. \tag{3.24}$$

It can be rewritten by using the matrix form as:

$$\boldsymbol{f}_d = \mathbf{S}_{d,\mathcal{L}}\boldsymbol{f}, \tag{3.25}$$

where $\mathbf{S}_{d,\mathcal{L}} = [\mathbf{I}_N]_{\mathcal{V}\mathcal{L}}$ and $[\mathbf{I}_N]_{\mathcal{V}\mathcal{L}}$ is the restriction matrix of $\mathbf{I}$, which is obtained from extracting columns indexed by $\mathcal{L}$ from $\mathbf{I}$. The upsampling operator from $\boldsymbol{f}_d \in \mathbb{R}^{|\mathcal{L}|}$ to $\boldsymbol{f}_u \in \mathbb{R}^{|\mathcal{V}|}$ is

$$f_u(n) = \begin{cases} f_d(n') & \text{if } v_{n'} \in \mathcal{V}_1 \text{ corresponds to } v_n \in \mathcal{L} \\ 0 & \text{otherwise.} \end{cases} \tag{3.26}$$

Its matrix form is

$$\boldsymbol{f}_u = \mathbf{S}_{u,\mathcal{L}}\boldsymbol{f}_d, \tag{3.27}$$

where $\mathbf{S}_{u,\mathcal{L}} = \mathbf{S}_{d,\mathcal{L}}^T$.

The downsampling-then-upsampling operation is defined as [32]

$$\begin{aligned} \mathbf{D}_{du,\mathcal{L}} &= \mathbf{S}_{u,\mathcal{L}}\mathbf{S}_{d,\mathcal{L}} = \frac{1}{2}(\mathbf{I}_N + \mathbf{S}_{\mathcal{L}}), \\ \mathbf{D}_{du,\mathcal{H}} &= \mathbf{S}_{u,\mathcal{H}}\mathbf{S}_{d,\mathcal{H}} = \frac{1}{2}(\mathbf{I}_N + \mathbf{S}_{\mathcal{H}}), \end{aligned} \tag{3.28}$$

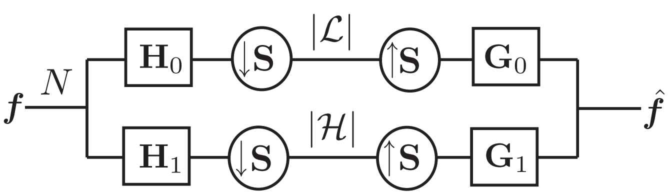


Figure 3.4: Critically sampled spectral graph filter bank.

where $\mathbf{S}_{\mathcal{L}}$ is the diagonal sampling matrix of a bipartite graph defined as

$$S_{\mathcal{L}}(m,m) = \begin{cases} +1 & \text{if } m \in \mathcal{H}, \\ -1 & \text{if } m \in \mathcal{L}, \end{cases} \tag{3.29}$$

and $\mathbf{S} := \mathbf{S}_{\mathcal{H}} = -\mathbf{S}_{\mathcal{L}}$.

Proposition 1. *(Downsampling Phenomenon of Bipartite Graph [32, Proposition 1]) The eigenspace projection matrix $\mathbf{P}_{\lambda_i}$ and the sampling matrix $\mathbf{S}$ are related as follows:*

$$\mathbf{S}\mathbf{P}_{\lambda_i} = \mathbf{P}_{2-\lambda_i}\mathbf{S}. \tag{3.30}$$

The nodes in $\mathcal{H}$ store the output of the highpass channel, whereas the nodes in $\mathcal{L}$ store the output of the lowpass channel. Critically sampled graph filter banks decompose $\boldsymbol{f}$ into $|\mathcal{L}|$ lowpass coefficients and $|\mathcal{H}|$ highpass coefficients, where $|\mathcal{L}| + |\mathcal{H}| = N$, as illustrated in Fig. 3.4. The overall transfer function of graph-QMF [32] and graphBior [33] can be written as

$$\begin{aligned} \mathbf{T} &= \frac{1}{2}\mathbf{G}_0(\mathbf{I}-\mathbf{S})\mathbf{H}_0 + \frac{1}{2}\mathbf{G}_1(\mathbf{I}+\mathbf{S})\mathbf{H}_1 \\ &= \frac{1}{2}(\mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1) + \frac{1}{2}(\mathbf{G}_1\mathbf{S}\mathbf{H}_1 - \mathbf{G}_0\mathbf{S}\mathbf{H}_0). \end{aligned} \tag{3.31}$$

The spectral folding term $\mathbf{G}_1\mathbf{S}\mathbf{H}_1 - \mathbf{G}_0\mathbf{S}\mathbf{H}_0$, arising from downsampling and upsampling, must be zero. From proposition 1, It can be rewritten as

$$\begin{aligned} \mathbf{G}_1\mathbf{S}\mathbf{H}_1 - \mathbf{G}_0\mathbf{S}\mathbf{H}_0 &= \sum_{\lambda} G_1(\lambda)\mathbf{P}_\lambda \mathbf{S} H_1(\lambda)\mathbf{P}_\lambda - G_0(\lambda)\mathbf{P}_\lambda \mathbf{S} H_0(\lambda)\mathbf{P}_\lambda \\ &= \left(\sum_{\lambda} G_1(\lambda)H_1(2-\lambda) - G_0(\lambda)H_0(2-\lambda)\right)\mathbf{P}_\lambda \mathbf{P}_{2-\lambda}\mathbf{S}. \end{aligned} \tag{3.32}$$

In addition, $\mathbf{T} = \mathbf{I}_N$ should be satisfied for perfect reconstruction, i.e.,

$$\mathbf{G}_1\mathbf{H}_1 - \mathbf{G}_0\mathbf{H}_0 = \left(\sum_{\lambda} G_1(\lambda)H_1(\lambda) - G_0(\lambda)H_0(\lambda)\right)\mathbf{P}_\lambda = 0. \tag{3.33}$$

should be satisfied. Hence, the perfect reconstruction condition of graph-QMF [32] and graphBior [33] can be expressed as

$$G_0(\lambda)H_0(\lambda) + G_1(\lambda)H_1(\lambda) = 2, \tag{3.34}$$

$$G_0(\lambda)H_0(2-\lambda) - G_1(\lambda)H_1(2-\lambda) = 0. \tag{3.35}$$

Additionally, the orthogonal transform, graph-QMF, has an orthogonality condition $H_0^2(\lambda) + H_0^2(2-\lambda) = c^2$. Therefore, filters are chosen in a way that satisfies $H_1(\lambda) = H_0(2-\lambda)$, $H_0(\lambda) = G_0(\lambda)$ and $H_1(\lambda) = G_1(\lambda)$. Unfortunately, filters that satisfy these conditions are not compact supports. That is, if graph-QMF were forced to be a compact support, it would suffer from a loss of orthogonality and a reconstruction error. On the other hand, graphBior relaxes the orthogonal condition of graph-QMF and has a perfect reconstruction condition and compact support because it uses a design method similar to Cohen-Daubechies-Feauveau's construction for regular signals [40].

The critically sampled filter bank is designed for bipartite graphs. When it is applied to an arbitrary graph, the original graph should be decomposed into an edge-disjoint collection of $K_{\mathcal{G}}$ bipartite subgraphs by using a bipartite subgraph approximation such as Harary's algorithm [32, 61] and the transform performed on each subgraph. Each subgraph has the same node set as the original graph, and their union yields back the original graph. This decomposition leads to a separable *multidimensional* graph wavelet filter bank.

Chapter 4

M-channel Oversampled Spectral Graph Filter Banks

In this chapter, we describe the M-channel filter banks for graph signals which have M filters both in the analysis and synthesis sides. They allow us to design filters with arbitrary parameters, i.e., perfect reconstruction is possible even if we use an *arbitrary* lowpass filter, unlike the conventional critically sampled graph filter banks. Furthermore, they show that they outperform critically sampled systems in several applications similar to the case in regular signal processing.

The overall scheme and perfect reconstruction conditions are described in Section 4.1. Section 4.2 designs filters having good stopband attenuation. They satisfy the perfect reconstruction conditions and are based on the spectral factorizations. The design examples and their performances on signal analysis and denoising are shown in Section 4.3.

4.1 Perfect Reconstruction Condition

The details of the perfect reconstruction condition are discussed in this section. For clearer understanding, we present the case of $M = 4$ first, and extend it to any value of M afterwards.

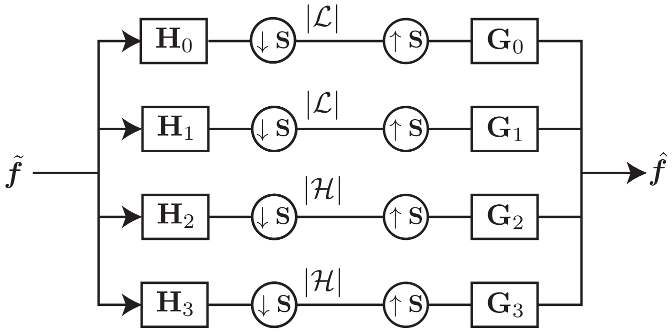


Figure 4.1: Oversampled graph filter bank.

4.1.1 Four-Channel Case

Consider the four-channel graph filter bank shown in Fig. 4.1. After filtering with $\mathbf{H}_k$, the zeroth and first channels pass $|\mathcal{L}|$ signals, whereas the second and third ones keep $|\mathcal{H}|$ signals. $\widehat{\boldsymbol{f}_k}$ is represented as

$$\widehat{\boldsymbol{f}_k} = \begin{cases} \frac{1}{2}\mathbf{G}_k(\mathbf{I}-\mathbf{S})\mathbf{H}_k\boldsymbol{f} & k=0,1 \\ \frac{1}{2}\mathbf{G}_k(\mathbf{I}+\mathbf{S})\mathbf{H}_k\boldsymbol{f} & k=2,3. \end{cases} \tag{4.1}$$

where

$$\mathbf{G}_k = \sum_{\lambda_i \in \sigma(\boldsymbol{\mathcal{L}})} G_k(\lambda_i)\mathbf{P}_{\lambda_i}$$
$$\mathbf{H}_k = \sum_{\lambda_i \in \sigma(\boldsymbol{\mathcal{L}})} H_k(\lambda_i)\mathbf{P}_{\lambda_i}.$$

Therefore, the overall transfer function $\mathbf{T}$ is

$$\begin{aligned}\mathbf{T} =& \frac{1}{2}\sum_{\lambda_i}\sum_{k=0}^{3} G_k(\lambda_i)H_k(\lambda_i)\mathbf{P}_{\lambda_i} \\ &+ \frac{1}{2}\sum_{\lambda_i,\gamma_j}\{G_2(\lambda_i)H_2(\gamma_j) + G_3(\lambda_i)H_3(\gamma_j) \\ &- G_0(\lambda_i)H_0(\gamma_j) - G_1(\lambda_i)H_1(\gamma_j)\}\mathbf{P}_{\lambda_i}\mathbf{S}\mathbf{P}_{\gamma_j}.\end{aligned} \tag{4.2}$$

From Proposition 1, $\mathbf{P}_{\lambda_i}\mathbf{S}\mathbf{P}_{\gamma_j} = \mathbf{P}_{\lambda_i}\mathbf{P}_{2-\gamma_j}\mathbf{S}$ and the orthogonality of $\mathbf{P}_{\lambda_i}$, we get

$$\begin{aligned}\mathbf{T} =& \frac{1}{2}\sum_{\lambda_i}\sum_{k=0}^{3} G_k(\lambda_i)H_k(\lambda_i)\mathbf{P}_{\lambda_i} \\ &+ \frac{1}{2}\sum_{\lambda_i}\{G_2(\lambda_i)H_2(2-\lambda_i) + G_3(\lambda_i)H_3(2-\lambda_i) \\ &- G_0(\lambda_i)H_0(2-\lambda_i) - G_1(\lambda_i)H_1(2-\lambda_i)\}\mathbf{P}_{\lambda_i}\mathbf{S}.\end{aligned} \tag{4.3}$$

As a result, the perfect reconstruction condition becomes

$$\sum_{k=0}^{3} G_k(\lambda)H_k(\lambda) = 2 \tag{4.4}$$

and

$$G_2(\lambda)H_2(2-\lambda) + G_3(\lambda)H_3(2-\lambda) - G_0(\lambda)H_0(2-\lambda) - G_1(\lambda)H_1(2-\lambda) = 0 \tag{4.5}$$

for any λ. (4.5) is satisfied if we use the constraints $G_0(\lambda) = H_2(2-\lambda)$, $G_1(\lambda) = H_3(2-\lambda)$, $G_2(\lambda) = H_0(2-\lambda)$, and $G_3(\lambda) = H_1(2-\lambda)$ (similar to what is done in [33]). Accordingly, (4.4) becomes

$$G_0(\lambda)H_0(\lambda) + G_0(2-\lambda)H_0(2-\lambda) + G_1(\lambda)H_1(\lambda) + G_1(2-\lambda)H_1(2-\lambda) = 2. \tag{4.6}$$

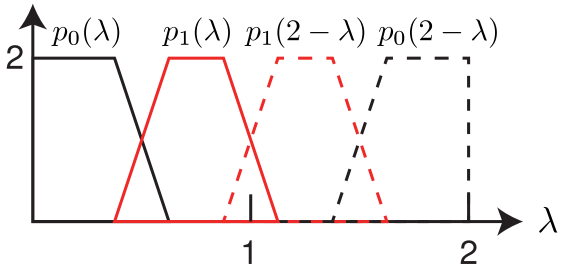


Figure 4.2: Four-channel product filter example.

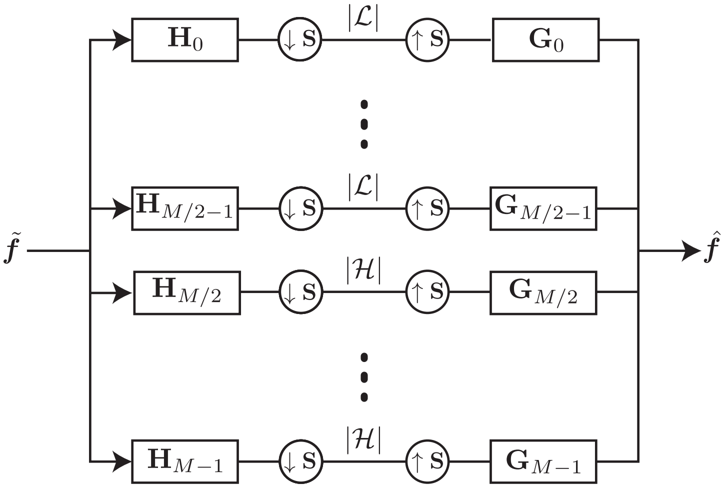


Figure 4.3: M-channel oversampled graph filter bank.

Let us define a product filter as $p_k(\lambda) = G_k(\lambda)H_k(\lambda)$. Finally, (4.6) can be rewritten as

$$p_0(\lambda) + p_0(2-\lambda) + p_1(\lambda) + p_1(2-\lambda) = 2. \tag{4.7}$$

By using this perfect reconstruction condition, we can select *four-channel* product filters instead of two-channel systems of the critically sampled graph filter bank. The situation is shown in Fig. 4.2.

4.1.2 General M-Channel Case

Let us assume that an oversampled graph filter bank has M channels, where M is even. Additionally, we assume that $M/2$ filters keep $|\mathcal{L}|$ signals and the other ones keep $|\mathcal{H}|$ signals, as shown in Fig. 4.3. Similar to (4.3), the transfer function can be calculated as

$$\begin{aligned}\mathbf{T} =& \frac{1}{2}\sum_{\lambda_i}\sum_{k=0}^{M-1} G_k(\lambda_i)H_k(\lambda_i)\mathbf{P}_{\lambda_i} \\ &+ \frac{1}{2}\sum_{\lambda_i}\sum_{k=0}^{M/2-1}\{-G_k(\lambda_i)H_k(2-\lambda_i) + G_{k+M/2}(\lambda_i)H_{k+M/2}(2-\lambda_i)\}\mathbf{P}_{\lambda_i}\mathbf{S}.\end{aligned} \tag{4.8}$$

The perfect reconstruction condition is

$$\sum_{k=0}^{M-1} G_k(\lambda)H_k(\lambda) = 2 \tag{4.9}$$

$$\sum_{k=0}^{M/2-1} -G_k(\lambda)H_k(2-\lambda) + G_{k+M/2}(\lambda)H_{k+M/2}(2-\lambda) = 0 \tag{4.10}$$

for any λ. The latter equation is valid if we choose $G_k(\lambda) = H_{k+M/2}(2-\lambda)$ and $G_{k+M/2}(\lambda) = H_k(2-\lambda)$. Accordingly, (4.9) becomes

$$\sum_{k=0}^{M/2-1} G_k(\lambda)H_k(\lambda) + G_k(2-\lambda)H_k(2-\lambda) = 2. \tag{4.11}$$

As a result, the product filter $p_k(\lambda)$ must satisfy the following condition:

$$\sum_{k=0}^{M/2-1} p_k(\lambda) + p_k(2-\lambda) = 2. \tag{4.12}$$

4.1.3 Redundancy of M-channel Spectral Graph Filter Banks

Let $m < M$ be the number of filters which keep $|\mathcal{L}|$ signals after downsampling. The overall redundancy ρ of the M-channel spectral graph filter banks can be expressed as

$$\begin{aligned}\rho &= \frac{m|\mathcal{L}| + (M-m)|\mathcal{H}|}{N} \\ &= \frac{m|\mathcal{L}| + (M-m)(N-|\mathcal{L}|)}{N},\end{aligned} \tag{4.13}$$

where $|\mathcal{H}| = N - |\mathcal{L}|$. For example, when $M = 4$, $m = 2$, i.e., using a four-channel filter bank, $\rho = 2$.

4.2 Design of M-Channel Oversampled Graph Filter Bank

First, we will consider the case of $M = 4$. Let us define $q(\lambda) = p_0(\lambda) + p_1(\lambda)$. (4.7) can be rewritten as

$$q(\lambda) + q(2-\lambda) = 2. \tag{4.14}$$

This equation is the same as that of a two-channel biorthogonal graph filter bank [33]. Therefore, the design problem boils down to separating the critically sampled product filter $q(\lambda)$ into lowpass and bandpass (Fig. 4.2) filters $p_0(\lambda)$ and $p_1(\lambda)$ such that the *sum* of filters is $q(\lambda)$.

Let us assume that a lowpass product filter $p_0(\lambda)$ is arbitrarily chosen so that $H_0(\lambda)$ and $G_0(\lambda)$ are "good" lowpass filters. By changing the variable of $\lambda = 1 + l$ [33], $p_0(1 + l)$ can be expressed as

$$p_0(1+l) = (1+l)^K \left(\sum_{m=0}^{J_0} \alpha_m l^m \right), \tag{4.15}$$

where α_m is an arbitrary parameter.

Following [33], $q(1+l)$ has the degree $2K-1$ and its even degree must be zero from the halfband condition.[1] Hence, $q(1+l)$ is represented as

$$q(1+l) = (1+l)^K \left(1 + \sum_{m=1}^{K-1} r_m l^m\right) = 1 + \sum_{n=0}^{K-1} c_{2n+1} l^{2n+1}, \tag{4.16}$$

which is, of course, the same as that in [33] and has a unique solution satisfying (4.14). Finally, the remaining product filter $p_1(\lambda)$ can be defined as follows:

$$p_1(1+l) = q(1+l) - p_0(1+l) = (1+l)^K \left(\sum_{m=0}^{J_1} \beta_m l^m\right). \tag{4.17}$$

Example: $K = 2$ zeros at $l = -1$

This is the same example as in [33]. We assume $J_0 = 1$. As in (4.15) and (4.17), $p_0(1+l)$ and $p_1(1+l)$ are

$$p_0(1+l) = (1+l)^2 \left(\alpha_0 + \alpha_1 l\right) \tag{4.18}$$

$$p_1(1+l) = (1+l)^2 \left(\beta_0 + \beta_1 l\right), \tag{4.19}$$

where α_0 and α_1 are arbitrarily chosen parameters. Then, the sum of the product filter $q(1+l)$ is defined as

$$q(1+l) = \frac{1}{2}(1+l)^2(2-l), \tag{4.20}$$

[1] Although *Proposition 1* in [33] restricts $q(\lambda)$ to being a *product* of two kernels, it is nonetheless applicable to the *sum* of two kernels assumed in this dissertation.

which is an odd-order polynomial and it is the same product filter as that in [33]. To guarantee the perfect reconstruction, β_0 and β_1 must be

$$\begin{aligned} \beta_0 &= 1 - \alpha_0 \\ \beta_1 &= -\left(\alpha_1 + \frac{1}{2}\right). \end{aligned} \tag{4.21}$$

That is, we can add free parameters (α_0 and α_1) to design a halfband filter, and this will lead to better filter characteristics.

A similar derivation is possible for general M-channel graph filter banks. In that case, the parameters for $(M-2)/2$ product filters can be freely chosen, and the last product filter can be designed so that the entire product filter $q(\lambda)$ is a maximally flat halfband filter.

Remark 1: The above design method yields perfect reconstruction graph filter banks. Unfortunately, the filter selection similar to CS-QMF [32] cannot obtain the perfect reconstruction filter set with the real-valued exact polynomial filters even for this oversampled case. This is easily confirmed by examining the transfer function in (4.8). If the filter bank is chosen similar to CS-QMF, the condition $G_k(\lambda) = H_k(\lambda)$ must be satisfied. Therefore, the perfect reconstruction condition (4.12) becomes

$$\sum_{k=0}^{M/2-1} H_k^2(\lambda) + H_k^2(2-\lambda) = 2. \tag{4.22}$$

Here, let us focus on the highest degree of $\sum_{k=0}^{M/2-1} H_k^2(\lambda)$. Indeed, each $H_k^2(\lambda)$ has an even highest degree. However, as previously mentioned, $\sum_{k=0}^{M/2-1} H_k^2(\lambda)$ must be an odd degree polynomial. Hence, at least one $H_k^2(\lambda)$ has a complex coefficient to cancel the highest degree. This means real-valued oversampled graph filter banks with the filter selection similar to CS-QMF cannot be designed.

Remark 2: Let us define $\mathbf{T}_a$ to be the matrix form of the analysis transform. Riesz bounds, which give the lower and upper bounds A and B such that $A||\boldsymbol{f}||^2 \leq ||\mathbf{T}_a\boldsymbol{f}||^2 \leq B||\boldsymbol{f}||^2$, of the analysis filter bank can be calculated similarly to what is shown in [13, 33], as

$$\begin{aligned}\mathbf{T}_a^T\mathbf{T}_a =& \frac{1}{2}\sum_{\lambda_i}\sum_{k=0}^{M-1} H_k^2(\lambda_i)\mathbf{P}_{\lambda_i} \\ &+\frac{1}{2}\sum_{\lambda_i}\sum_{k=0}^{M/2-1}\{-H_k(\lambda_i)H_k(2-\lambda_i)+H_{k+M/2}(\lambda_i)H_{k+M/2}(2-\lambda_i)\}\mathbf{P}_{\lambda_i}\mathbf{S}.\end{aligned} \tag{4.23}$$

Practically speaking, the second term in (4.23) can be ignored since the first term is much larger than the second one. That is, the Riesz bounds can be approximated as

$$\begin{aligned}A &\sim \inf_{\lambda}\frac{1}{2}\sum_{k=0}^{M-1} H_k^2(\lambda) \\ B &\sim \sup_{\lambda}\frac{1}{2}\sum_{k=0}^{M-1} H_k^2(\lambda).\end{aligned} \tag{4.24}$$

This result is the similar to what is shown in [33].

4.3 Design Examples and Experimental Results

In this section, we show the design methodology of M-channel oversampled graph filter banks and a few design examples.

4.3.1 Design Methodology

As mentioned above, we can use arbitrary parameters to design filters. In what follows, we will use a sequential design method to obtain good filter banks:

1. Design $H_k(1-l)$ and $G_k(1-l)$ ($k = 0, \dots, M/2-2$) with k_0 and k_1 zeros (where $K = k_0 + k_1$ in (4.15)–(4.16)) at $l = 1$ ($\lambda = 2$). They are represented as follows:

   $$\begin{aligned} H_k(1-l) &= (1-l)^{k_0} \sum_{m=0}^{J_k^{(h)}} s_{h,k,m} l^m \\ G_k(1-l) &= (1-l)^{k_1} \sum_{m=0}^{J_k^{(g)}} s_{g,k,m} l^m \end{aligned} \tag{4.25}$$

   where $s_{h,k,m}$ and $s_{g,k,m}$ are filter coefficients. i.e., the product filter $p_k(1-l) = G_k(1-l)H_k(1-l)$ can be represented as

   $$p_k(1-l) = (1-l)^K \left(\sum_{m=0}^{J_k^{(h)}+J_k^{(g)}} \alpha_{k,m} l^m \right). \tag{4.26}$$

   The numbers of arbitrary parameters in $H_k(1-l)$ and $G_k(1-l)$ are $J_k^{(h)}$ and $J_k^{(g)}$, respectively.

   The filters are optimized by using the cost function of the stopband attenuation shown below:

   $$C(H_k) = w_0 \int_{l \in \omega_p} (\sqrt{2} - H_k(1-l))^2 dl + w_1 \int_{l \in \omega_s} H_k^2(1-l) dl, \tag{4.27}$$

   where w_0 and w_1 are weights and ω_p and ω_s are defined as the passband and stopband ($-1 \leq \omega_p, \omega_s \leq 1$), respectively.

2. Calculate the two-channel halfband filter pair $q(1-l) = q(\lambda)$ and $q(1+l) = q(2-\lambda)$ with K zeros at $l = 1$ so that the pair satisfies (4.14).

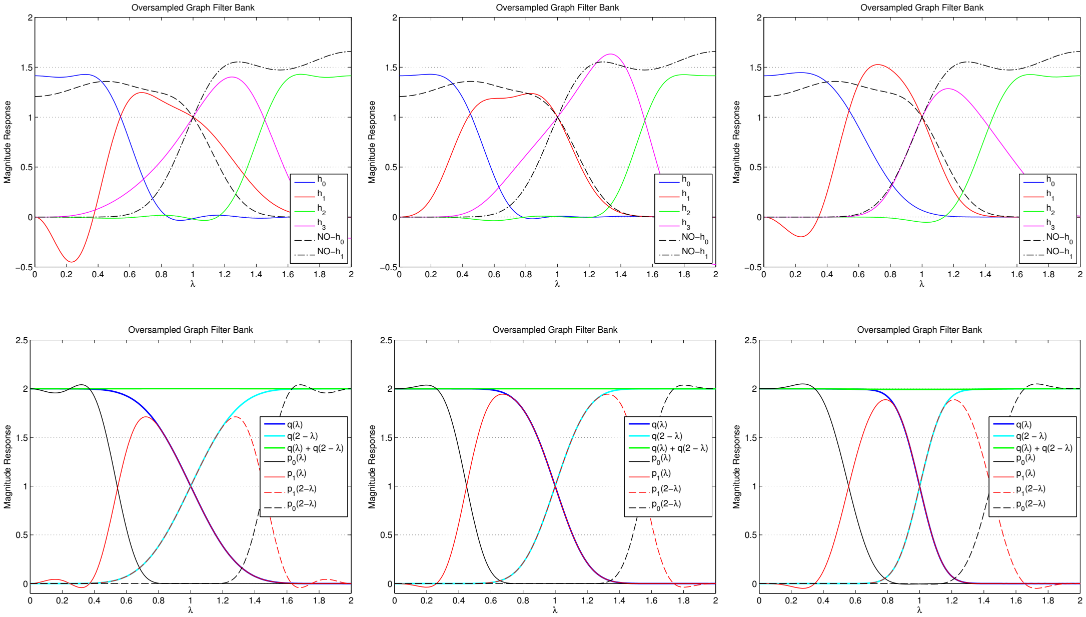


Figure 4.4: Four-channel oversampled graph filter banks. From left to right: $(k_0, k_1) = (2, 2)$, $(4, 4)$, and $(8, 8)$. Top row: analysis filter bank. Black lines indicate graphBior(6, 6) [33]. Bottom row: halfband filters.

3. Calculate the bandpass product filter[2]

$$p_{\frac{M}{2}-1}(1-l) = q(1-l) - \sum_{k=0}^{M/2-2} p_k(1-l) = (1-l)^K \tilde{p}_{\frac{M}{2}-1}(1-l). \qquad (4.28)$$

4. Factorize $p_{\frac{M}{2}-1}(1-l)$ into two bandpass filters $H_{\frac{M}{2}-1}(1-l)$ and $G_{\frac{M}{2}-1}(1-l)$. Test all combinations of roots as long as both bandpass filters have real-valued coefficients, and select the best combination, i.e., the filters minimizing $C(H_{\frac{M}{2}-1}) + C(G_{\frac{M}{2}-1})$.

[2] There always exists $q(1-l)$ which satisfies the perfect reconstruction condition (4.14) [33]. Therefore, $p_{\frac{M}{2}-1}(1-l)$ also has a unique solution with real coefficients as long as all of the arbitrary design filters have real coefficients. Additionally, since the perfect reconstruction condition is only imposed on $p_{\frac{M}{2}-1}(1-l)$, $J_k^{(h)}$ and $J_k^{(g)}$ in (4.25) can be set arbitrarily regardless of K.

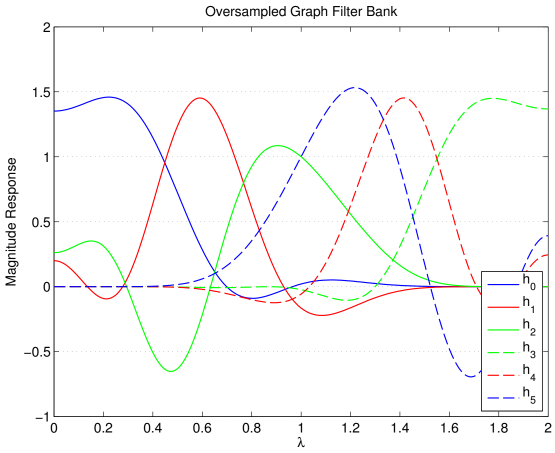


Figure 4.5: Six-channel oversampled graph filter bank: analysis bank.

Fig. 4.4 shows an example of oversampled graph filter banks. The arbitrary lowpass filters $H_0(\lambda)$ and $G_0(\lambda)$ are designed to have degree 10 and 11, respectively. We used $-1 \leq \omega_p \leq -0.84$ and $-0.75 \leq \omega_s \leq 1$. For comparison, the frequency responses of the critically sampled CS-SF [33] are also plotted. They have 13-taps for the lowpass filter and 12-taps for the highpass filter. It is clear that our oversampled lowpass filter has a sharper transition band and a more uniform response in the passband than the critically sampled graph filter banks have. In the following experiments, we use the oversampled filter bank with $(k_0, k_1) = (4, 4)$ zeros.

Additionally, Fig. 4.5 presents a six-channel oversampled graph filter bank with $(k_0, k_1) = (4, 4)$ zeros and $\{J_h, J_g\} = \{7, 8\}$. The filter lengths are 11 or 12 taps. For lowpass filters, we used $-1 \leq \omega_p \leq -0.93$ and $-0.83 \leq \omega_s \leq 1$. For bandpass ones, $-0.83 \leq \omega_p \leq -0.75$ and $-1 \leq \omega_s \leq -0.95 \wedge -0.65 \leq \omega_s \leq 1$ are used. It is clear that our sequential design methodology can be used for the general M-channel case and the frequency responses of the filters are well localized.

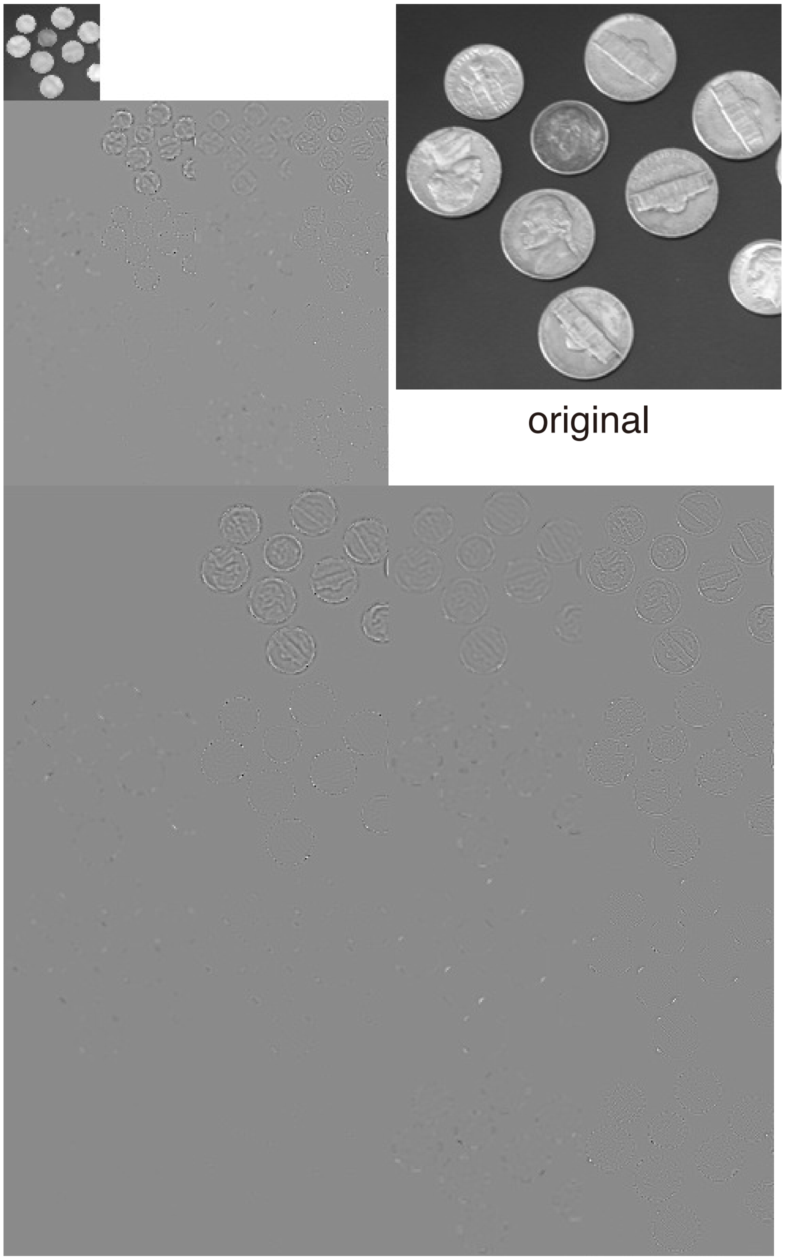


Figure 4.6: Multiresolution *Coins* image after three-level decomposition using the oversampled graph filter bank. The original image on the same scale is shown at the top right. The values of the transformed coefficients are scaled to be in the range $[0, 1]$ for the sake of visualization.

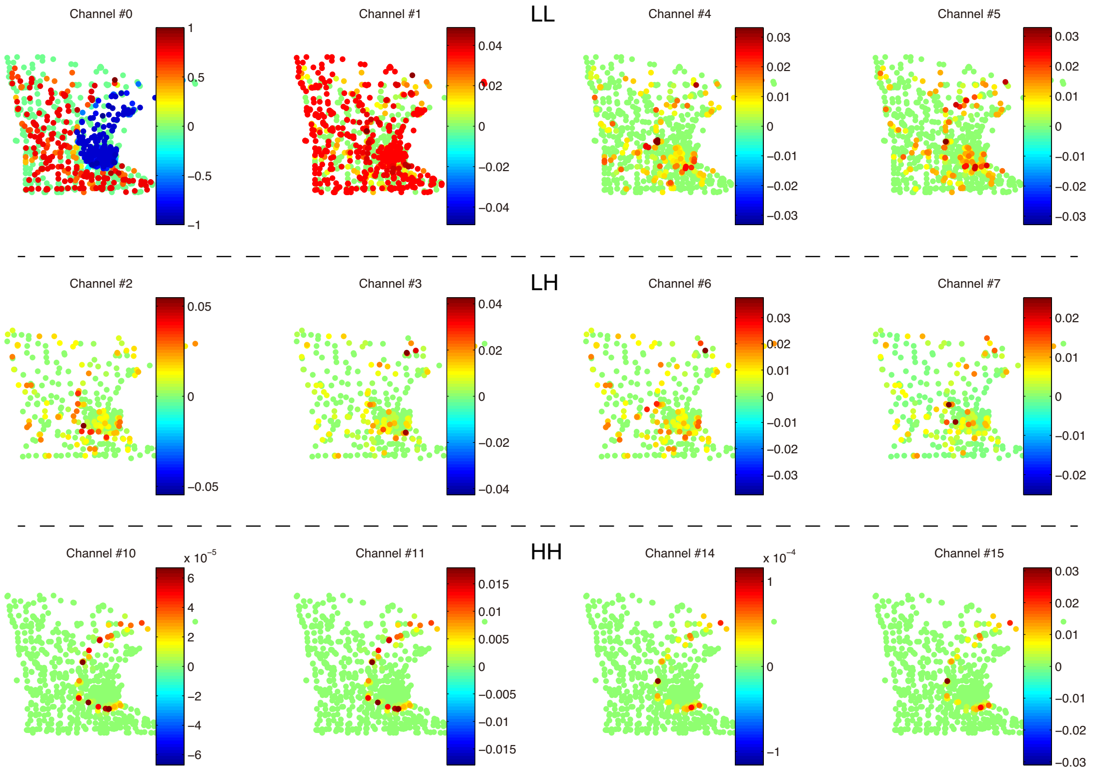


Figure 4.7: Graphs decomposed by the proposed oversampled graph filter bank. (Colors are adjusted according to each channel for the sake of visualization.) Original signal is shown in Fig. 4.9(a). We use a two-dimensional four-channel filter bank leading to $4^2 = 16$ channels. Note that the graph is three-colorable: therefore, channels 8, 9, 12, and 13 (corresponding to the HL channel for the critically sampled filter banks) are empty.

4.3.2 Graph Signal Decomposition

The signal on graph is decomposed by the proposed M-channel graph filter bank. Since the filters designed in this dissertation are exact polynomials, the filtered signal can be efficiently computed by using Chebychev polynomials [13]. Therefore, an explicit computation of the entire set of eigenvalues and eigenvectors of $\boldsymbol{\mathcal{L}}$ is not required.

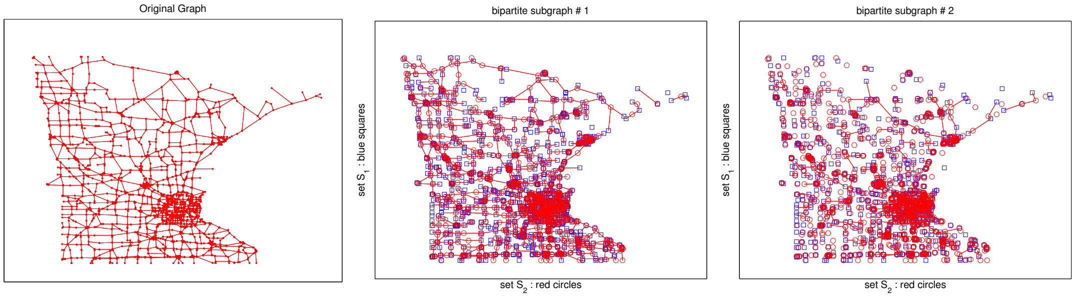


Figure 4.8: Structure of the *Minnesota Traffic Graph*. It was reproduced from the MATLAB code of Narang and Ortega [33], and Harary's algorithm [32, 61] was used to yield two bipartite subgraphs. From left to right: Original graph, bipartite graph #1, and bipartite graph #2.

Figs. 4.6 and 4.7 show the decomposed results of the graph signals using the proposed oversampled graph filter bank. For the *Coins* image shown in Fig. 4.6, the eight-connected image graph is decomposed into a rectangular bipartite subgraph and a diagonal bipartite subgraph. Its further details can be found in [32, 33]. The structure of the *Minnesota Traffic Graph*, whose decomposition is presented in Fig. 4.7, is shown in Fig. 4.8. It is three-colorable: thus, the HL channel is empty and not shown. It is clear that the decomposed graph signals are well localized and different channels extract different signal characteristics.

4.3.3 Denoising of Graph Signal

Here, we show the potential ability of using oversampled graph filter banks to remove additive white Gaussian noise from graph signals of the *Minnesota Traffic Graph*. The oversampled graph filter bank is compared with graphBior(6, 6) (CS-SF) [33], the spectral graph wavelet transform (UD-SP) with three scales [13] and a regular one-dimensional wavelet *sym8*, which can be found in the Wavelet Toolbox in MATLAB. For the regular wavelet transform, the input signal $\boldsymbol{f}$ is treated as a vector, and one-level and five-level dyadic decompositions are performed. Only one level transform is used

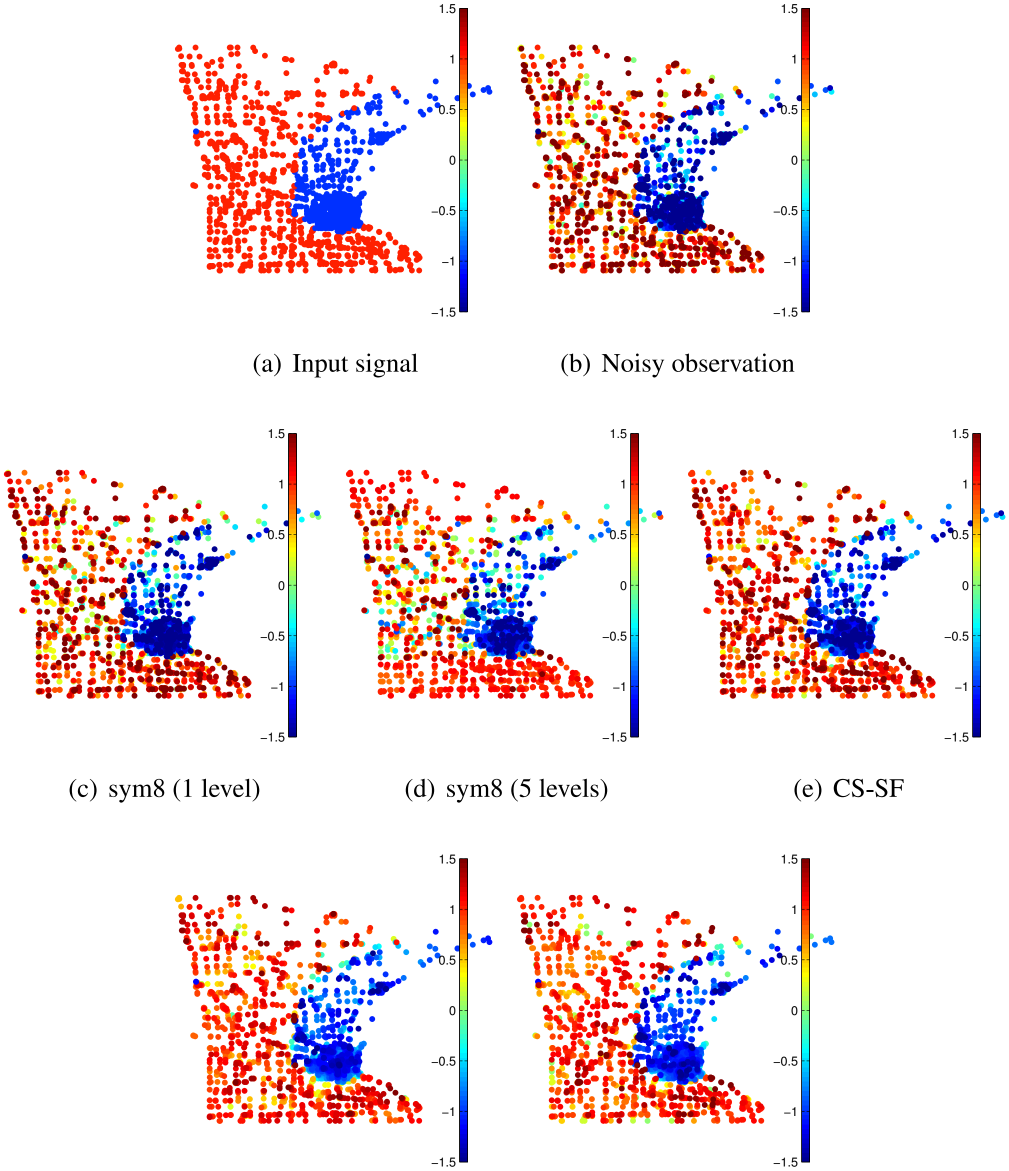
Table 4.1: Denoised Results of Minnesota Traffic Graph (Average of Ten Executions): SNR (dB)

	σ	noisy	sym8	sym8	CS-SF	UD-SP	OS-SF
			(1 level)	(5 levels)			
Example 1	1/32	30.04	30.02	30.06	31.25	33.24	**34.64**
	1/16	24.08	24.24	24.03	25.66	27.73	**28.77**
	1/8	18.07	18.59	17.96	20.16	**22.22**	21.82
	1/4	12.06	12.00	11.02	14.25	15.19	**15.36**
	1/2	6.05	6.26	5.74	8.51	10.29	**10.31**
	1	0.00	1.60	3.05	2.77	**8.82**	4.18
Example 2	1/32	29.45	28.69	28.24	29.13	29.17	**30.30**
	1/16	23.44	22.81	22.00	23.45	23.08	**24.54**
	1/8	17.41	16.64	15.31	17.91	17.27	**18.09**
	1/4	11.44	9.99	8.52	11.90	11.59	**11.99**
	1/2	5.40	4.40	3.31	6.49	**7.35**	7.32
	1	-0.64	0.27	0.94	1.54	**4.72**	3.22
Redundancy	–		1.00	1.00	1.00	4.00	4.00

for the graph filter banks. All methods retain the lowest-frequency subband and the remaining high-frequency subbands are hard-thresholded with $T = 3\sigma$, where σ is the standard deviation of noise.

Two signal examples, shown in Figs. 4.9(a) and 4.10(a) were tested. Both structures, i.e., the original GLM, are the same, but the values on the nodes are different. Table 4.1 summarizes the denoising performances together with the redundancies of the transforms. As expected, the graph filter banks perform much better than the regular wavelet transform. Furthermore, our oversampled graph filter bank outperforms CS-SF by 1–3 dB in SNR. The UD-SP performs better for the strong noise case $\sigma = 1$, whereas the proposed oversampled graph filter banks are better than the UD-SP for the other σ.

The both OS-SF and UD-SP have four filters: however, the frequency partitions are different. The UD-SP belongs to a class of *nonuniform* graph filter banks, whereas our filter bank is a *uniform* one. Indeed, the performance depends on the signal characteristics. The graph Fourier spectra of both examples are shown in Fig. 4.11. Obviously, the



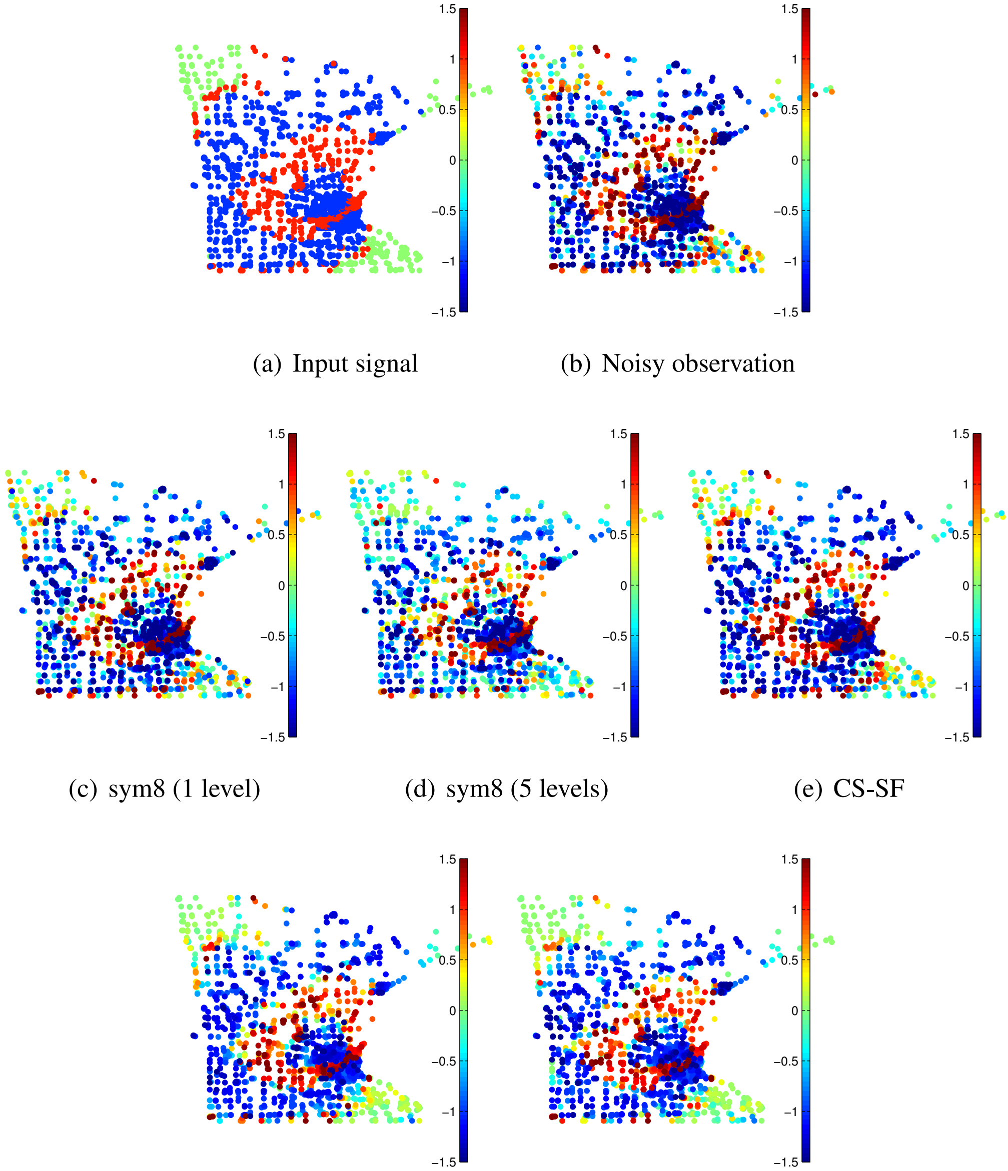
(a) Input signal (b) Noisy observation

(c) sym8 (1 level) (d) sym8 (5 levels) (e) CS-SF

(f) UD-SP (g) OS-SF

Figure 4.9: Denoising results of Example 1.

spectrum of Example 1 is more concentrated at the low λ values than that of Example 2. This characteristic is responsible for the good denoising performance of the UD-SP with $\sigma = 1$ in Example 1.



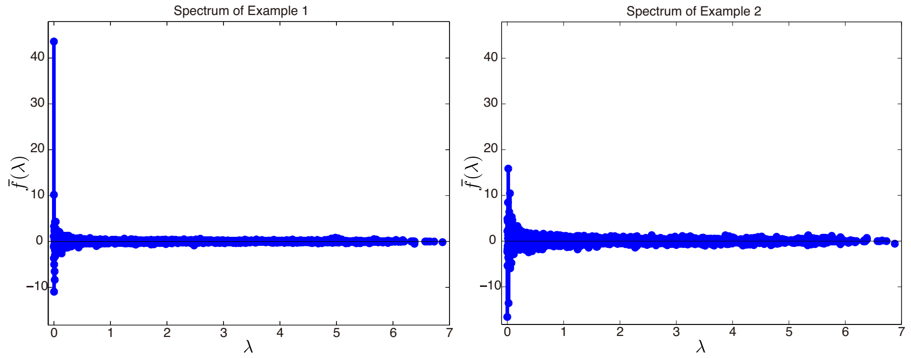
(a) Input signal (b) Noisy observation

(c) sym8 (1 level) (d) sym8 (5 levels) (e) CS-SF

(f) UD-SP (g) OS-SF

Figure 4.10: Denoising results of Example 2.

The denoised signals of Example 1 and 2 for $\sigma = 1/2$ are shown in Figs. 4.9 and 4.10, respectively. Since the regular wavelet transform does not consider the structure of signals explicitly, the signals are over-smoothened across the boundary of the center and surrounding areas; many blue points appear in the surrounding area. In contrast,



(a) Example 1

(b) Example 2

Figure 4.11: Graph Fourier spectra of *Minnesota Traffic Graph*. Since the experiment uses the unnormalized graph Laplacian matrix, the maximum value of λ is not restricted to be 2. We utilized the code by Shuman et al. in [31].

graph filter banks preserve the solid boundary. It is clear that the proposed filter bank performs better than the critically sampled one. It is well-known in signal/image processing circles that the oversampled filter banks are better than the critically sampled ones for signal analysis. The experiment showed this to be the case for graph signal processing.

4.4 Summary

We presented a method of designing M-channel oversampled filter banks for graph signals. It satisfies the perfect reconstruction condition and allows us to use arbitrary parameters, unlike critically sampled graph filter banks. Furthermore, it was shown to outperform other transforms, including regular wavelet transforms, in a graph signal denoising experiment.

Chapter 5

Oversampled Graph Laplacian Matrix

This chapter introduces the concept of graph oversampling using an oversampled graph Laplacian matrix (OSGLM) and shows the effective ways of graph oversampling. The oversampling method is the completely different from the M-channel oversampled spectral graph filter banks shown in Chapter 4.

For the graph oversampling, we can append nodes and edges to the graph somewhat arbitrarily and freely choose additional graph signals. This chapter presents the effective method of applying graph oversampling to non-bipartite graphs, which can solve the problem on the conventional decimated transforms such as critically sampled graph filter banks and oversampled ones, i.e., the problem that the conventional decimated transforms can only be applied into bipartite graphs. Our method can converts an arbitrary K-colorable graph into one bipartite graph containing all edges of the original graph. Hence, the decimated transform with the proposed oversampling can treat any graph signals without removing the edges in the original graphs.

The concept of graph oversampling is shown in Section 5.1. Section 5.2 describes the way of oversampling arbitrary graphs that makes one oversampled bipartite graph from a K-colorable graph. We also clarify the theoretical relationship between the proposed oversampled graph and the bipartite double cover of a graph in graph theory [62–64]. Section 5.3 shows examples of graph oversampling for images and a ring graph, and compares oversampled bipartite graphs with the critically sampled ones. We validated the performance of graph oversampling through an experiment on non-linear approximation of images and denoising of graph signals in Section 5.4.

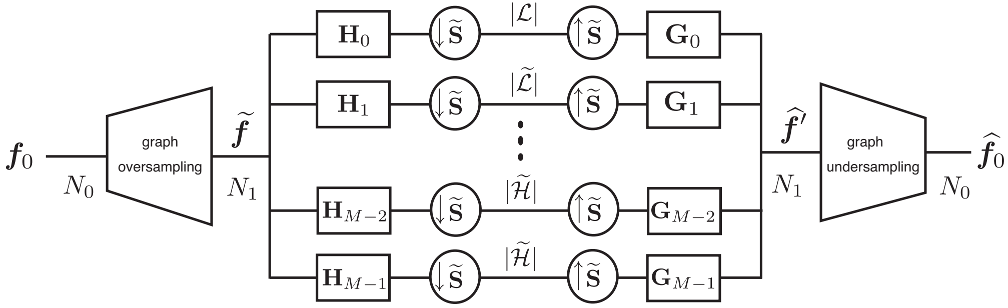

$\boldsymbol{f}_0$
N_0
graph oversampling
$\widetilde{\boldsymbol{f}}$
N_1
$\mathbf{H}_0$
$\mathbf{H}_1$
$\mathbf{H}_{M-2}$
$\mathbf{H}_{M-1}$
$\downarrow\widetilde{\mathbf{S}}$
$|\widetilde{\mathcal{L}}|$
$|\widetilde{\mathcal{L}}|$
$|\widetilde{\mathcal{H}}|$
$|\widetilde{\mathcal{H}}|$
$\uparrow\widetilde{\mathbf{S}}$
$\mathbf{G}_0$
$\mathbf{G}_1$
$\mathbf{G}_{M-2}$
$\mathbf{G}_{M-1}$
$\widehat{\boldsymbol{f}}'$
N_1
graph undersampling
$\widehat{\boldsymbol{f}}_0$
N_0

Figure 5.1: Graph oversampling followed by M-channel oversampled graph filter bank.

5.1 Concept of Graph Oversampling

In this section, we describe the concept of oversampling method for graph signals. Figure 5.1 shows an example of the transform using graph oversampling with an M-channel oversampled graph filter bank. The graph and graph signals are oversampled before applying graph filter banks by using the oversampled graph Laplacian matrix. Firstly, we describe the detail of oversampling of underlying graph and signal.

5.1.1 Oversampled Graph Laplacian Matrix

Before applying the graph filter bank, the original bipartite graph $\mathcal{G} = \{\mathcal{L}, \mathcal{H}, \mathcal{E}\}$ is expanded into an oversampled bipartite graph $\widetilde{\mathcal{G}} = \{\widetilde{\mathcal{L}}, \widetilde{\mathcal{H}}, \widetilde{\mathcal{E}}\}$ where $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ include $\mathcal{L}$ and $\mathcal{H}$, respectively. The spectral domain filtering is then performed based on the oversampled GLM.

Let us denote the original GLM of a bipartite graph by $\boldsymbol{\mathcal{L}}_0$ and its adjacency matrix by $\mathbf{A}_0$, and let us set their sizes as $N_0 \times N_0$. The normalized oversampled GLM $\widetilde{\boldsymbol{\mathcal{L}}}$ is an $N_1 \times N_1$ matrix $(N_1 > N_0)$, and $N_1 - N_0$ is the number of the additional nodes. It is represented as

$$\widetilde{\boldsymbol{\mathcal{L}}} = \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{L}}\widetilde{\mathbf{D}}^{-1/2} \tag{5.1}$$

where

$$\widetilde{\mathbf{L}} = \widetilde{\mathbf{D}} - \widetilde{\mathbf{A}} \tag{5.2}$$

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_{01} \\ \mathbf{A}_{01}^T & \mathbf{0}_{N_1 - N_0} \end{bmatrix}, \tag{5.3}$$

in which $\widetilde{\mathbf{A}}$ is the oversampled adjacency matrix, and $\widetilde{\mathbf{D}}$ is the degree matrix that normalizes the new GLM. Additionally, $\mathbf{A}_{01}$ contains information on the connection between the original nodes and appended ones. Note that nodes are appended so that $\widetilde{\boldsymbol{\mathcal{L}}}$ is still a bipartite graph. Filters in the spectral domain in Fig. 5.1 are defined as

$$\begin{aligned} \mathbf{H}_k &= \sum_{\lambda_i \in \sigma(\widetilde{\mathcal{G}})} H_k(\lambda_i) \widetilde{\mathbf{P}}_{\lambda_i}, \\ \mathbf{G}_k &= \sum_{\lambda_i \in \sigma(\widetilde{\mathcal{G}})} G_k(\lambda_i) \widetilde{\mathbf{P}}_{\lambda_i}, \end{aligned} \tag{5.4}$$

where $\widetilde{\mathbf{P}}_{\lambda_i}$ is the projection matrix of the oversampled graph.

The downsampling matrix $\widetilde{\mathbf{S}} = -\mathbf{S}_{\widetilde{\mathcal{L}}} = \mathbf{S}_{\widetilde{\mathcal{H}}}$ of the oversampled graph is determined by $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$. It can be represented as follows:

$$\widetilde{\mathbf{S}} = \begin{bmatrix} \mathbf{S}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_1 \end{bmatrix}, \tag{5.5}$$

where $\mathbf{S}_0$ and $\mathbf{S}_1$ are the downsampling matrices of the original and additional nodes, respectively. The oversampled signal $\widetilde{\boldsymbol{f}}$ is written as

$$\widetilde{\boldsymbol{f}} = \begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{f}_1 \end{bmatrix}, \tag{5.6}$$

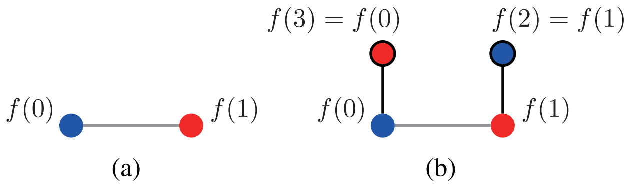


Figure 5.2: Toy example of graph oversampling. (a) Scenario 1: Two-node-graph. (b) Scenario 2: Oversampled two-node-graph. The black lines are appended edges.

where $\boldsymbol{f}_1$ is the signal for the additional nodes and its length is $N_1 - N_0$. Since the perfect reconstruction condition of graph filter banks does not depend on the graph oversampling as long as the oversampled graph is bipartite, the output signal $\widehat{\boldsymbol{f}}'$ is equal to the input signal $\widetilde{\boldsymbol{f}}$ regardless of the additional signal value $\boldsymbol{f}_1$. Naturally, $\widehat{\boldsymbol{f}}_0$ can be obtained from $\widehat{\boldsymbol{f}}'$.

5.1.2 Relationship With Undecimated Transforms

We would like to mention that *the output signal after filtering with the oversampled normalized GLM is completely different from that obtained with undecimated graph filtering*. To introduce the fact, we assume two following scenarios:

Scenario 1: Filtering using the *original* graph *without* downsampling.

Scenario 2: Filtering using the *oversampled* graph *with* downsampling.

Here, a toy example is considered: a two-node-graph illustrated in Fig. 5.2(a). All weights on edges are 1 and the original graph signal is $\boldsymbol{f}_0 = [f(0)\ f(1)]^T$. We use this graph for the scenario 1. For the scenario 2, the oversampled graph shown in Fig. 5.2(b) is used. Obviously, the oversampled graph is still a bipartite graph. The oversampled graph signal in this example is defined as the duplicated copy of $\boldsymbol{f}_0$, i.e., $\widetilde{\boldsymbol{f}} = [f(0)\ f(1)\ f(1)\ f(0)]^T$.

The original GLM is represented as

$$\mathbf{L}_0 = \boldsymbol{\mathcal{L}}_0 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{5.7}$$

Its eigenvalues and eigenvectors are clearly $\{\lambda_0, \lambda_1\} = \{0, 2\}$ and

$$\mathbf{U}_0 = \begin{bmatrix} \boldsymbol{u}_0 \; \boldsymbol{u}_1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The oversampled GLM is

$$\widetilde{\mathbf{L}} = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}, \tag{5.8}$$

and its normalized version is

$$\widetilde{\boldsymbol{\mathcal{L}}} = \begin{bmatrix} 1 & -1/2 & 0 & -1/\sqrt{2} \\ -1/2 & 1 & -1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 1 & 0 \\ -1/\sqrt{2} & 0 & 0 & 1 \end{bmatrix}. \tag{5.9}$$

Its eigenvalues are $\{\widetilde{\lambda}_0, \widetilde{\lambda}_1, \widetilde{\lambda}_2, \widetilde{\lambda}_3\} = \{0, 1/2, 3/2, 2\}$ and

$$\widetilde{\mathbf{U}} = \frac{1}{\sqrt{6}} \begin{bmatrix} \sqrt{2} & 1 & -1 & \sqrt{2} \\ \sqrt{2} & -1 & -1 & -\sqrt{2} \\ 1 & -\sqrt{2} & \sqrt{2} & 1 \\ 1 & \sqrt{2} & \sqrt{2} & -1 \end{bmatrix}.$$

For the scenario 1, the output signal after filtering with $H_0(\lambda)$ can be represented as

$$\begin{aligned} \boldsymbol{f}_0' &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} H_0(0) & \\ & H_0(2) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} H_0(0) & H_0(2) \\ H_0(0) & -H_0(2) \end{bmatrix} \begin{bmatrix} f(0) + f(1) \\ f(0) - f(1) \end{bmatrix}. \end{aligned} \tag{5.10}$$

As a result, it captures the signal characteristics at $\lambda_i = 0, 2$ (the minimum and maximum eigenvalues).

Whereas for the scenario 2, we can observe the different filter response. The filtered and downsampled signal for the scenario 2 can be represented as follows:

$$\begin{aligned}
\widetilde{\boldsymbol{f}}' =& \frac{1}{6}\begin{bmatrix}\sqrt{2} & 1 & -1 & \sqrt{2}\\ 1 & -\sqrt{2} & \sqrt{2} & 1\end{bmatrix}\begin{bmatrix}H_0(0) & & & \\ & H_0(\frac{1}{2}) & & \\ & & H_0(\frac{3}{2}) & \\ & & & H_0(2)\end{bmatrix}\\
&\times\begin{bmatrix}\sqrt{2} & \sqrt{2} & 1 & 1\\ 1 & -1 & -\sqrt{2} & \sqrt{2}\\ -1 & -1 & \sqrt{2} & \sqrt{2}\\ \sqrt{2} & -\sqrt{2} & 1 & -1\end{bmatrix}\begin{bmatrix}f(0)\\ f(1)\\ f(1)\\ f(0)\end{bmatrix}\\
=& \begin{bmatrix}\sqrt{2}c_0H_0(0) - c_1H_0(\frac{3}{2}) & c_0H_0(\frac{1}{2}) + \sqrt{2}c_1H_0(2)\\ c_0H_0(0) + \sqrt{2}c_1H_0(\frac{3}{2}) & -\sqrt{2}c_0H_0(\frac{1}{2}) + c_1H_0(2)\end{bmatrix}\begin{bmatrix}f(0)+f(1)\\ f(0)-f(1)\end{bmatrix},
\end{aligned} \tag{5.11}$$

where $c_0 = \frac{1}{6}(\sqrt{2}+1)$ and $c_1 = \frac{1}{6}(\sqrt{2}-1)$. In this example, the extra eigenvalues $\widetilde{\lambda}_i = 1/2, 3/2$ are used for the signal analysis. Roughly speaking, finer characteristics of the graph signal can be captured by oversampling the graph signal and the underlying graph.

5.1.3 Redundancy of Transform With Oversampled Graph Laplacian Matrix

Let $m < M$ be the number of filters which keep $|\mathcal{L}|$ signals after downsampling. The overall redundancy ρ of the M-channel spectral graph filter banks with oversampled GLM can be expressed as

$$\rho = \frac{m|\widetilde{\mathcal{L}}| + (M-m)(N_1 - |\widetilde{\mathcal{L}}|)}{N_0}, \tag{5.12}$$

where $|\widetilde{\mathcal{H}}| = N_1 - |\widetilde{\mathcal{L}}|$.

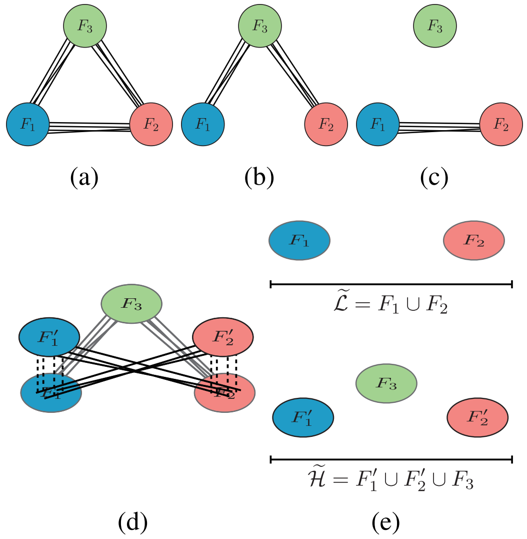
5.2 Effective Graph Expansion Methods

As described in Section 5.1.1, the appended nodes of the oversampled GLM can be arbitrarily connected to the nodes as long as the oversampled graph is bipartite. The additional signal value $\boldsymbol{f}_1$ can also be freely chosen. However, an inappropriate choice of graph oversampling causes a performance loss. In this section, we describe an efficient way to construct oversampled graphs that avoids such losses. Since the oversampled graph has to be a bipartite graph, we first decompose the original graph into bipartite subgraphs. We take one bipartite subgraph and append nodes and edges in the other bipartite subgraphs to it. In this way, *one oversampled bipartite graph containing all the edges of the bipartite subgraphs* is obtained.

5.2.1 Three-Colorable Graphs

First, we describe a way to convert a three-colorable graph into one oversampled bipartite graph containing all edges of the original graph. We assign three colors to nodes such that adjacent nodes have different colors and distinguish these nodes as F_1, F_2 and F_3, respectively. The three-colorable graph (Fig. 5.3(a)) can be decomposed into two bipartite subgraphs: $\mathcal{B}_1$ that contains edges linking $F_1 \cup F_2$ and F_3 (Fig. 5.3(b)), and $\mathcal{B}_2$ that contains edges between F_1 and F_2 (Fig. 5.3(c)). Hence, the edges in $\mathcal{B}_2$ only have connections on one side of the subsets (F_1 and F_2) of $\mathcal{B}_1$.

To make the oversampled graph, nodes are appended just above each node in F_1 and F_2 of $\mathcal{B}_1$. The additional node sets are represented as F_1' and F_2', respectively. Each appended node has the same value as the corresponding node. By adding the edges between F_1 and F_2 to the graph, we can convert the original graph into one bipartite


F_3
F_1
F_2
F_1'
F_2'
$\widetilde{\mathcal{L}} = F_1 \cup F_2$
$\widetilde{\mathcal{H}} = F_1' \cup F_2' \cup F_3$

(a) (b) (c)

(d) (e)

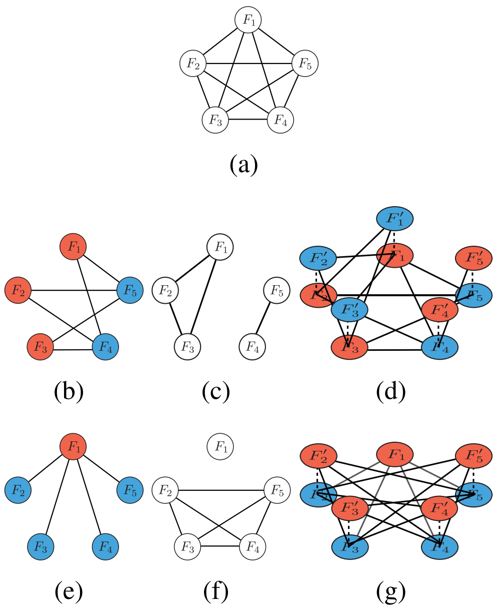
Figure 5.3: Bipartite oversampled graph construction for three-colorable graphs. (a) three-colorable graph whose node sets are F_1, F_2 and F_3. (b) bipartite subgraph $\mathcal{B}_1$. (c) bipartite subgraph $\mathcal{B}_2$. (d) oversampled bipartite graph. The gray lines are edges contained in $\mathcal{B}_1$, and the dashed and solid black lines are vertical edges and additional edges according to the edge information of original graph, respectively. (e) sets $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ of the oversampled bipartite graph.

graph that contains all edges and nodes in the original graph (Fig. 5.3(d)). Since each appended node and its corresponding node have the same value, they can be connected by a vertical edge. The oversampled graph has node sets $\widetilde{\mathcal{L}} = F_1 \cup F_2$ and $\widetilde{\mathcal{H}} = F_1' \cup F_2' \cup F_3$, as shown in Fig. 5.3(e).

If some of the nodes in F_1 and F_2 only have connections to F_3, they are isolated in $\mathcal{B}_2$. Hence, there is no need to append these nodes to the oversampled graph. The redundancy after a transformation by using the M-channel graph filter bank and oversampling of a three-colorable graph can be computed as

$$\begin{aligned}\rho =& \frac{M\{(|F_1| + |F_2|) + (|F_1'| + |F_2'| + |F_3|)\}}{2N} \\ =& \frac{M(N + |F_1| + |F_2| - |\mathcal{I}|)}{2N}\end{aligned} \tag{5.13}$$

where $|\mathcal{I}|$ is the number of isolated nodes and satisfies $|F_1'| + |F_2'| = |F_1| + |F_2| - |\mathcal{I}|$.



(a)

(b) (c) (d)

(e) (f) (g)

Figure 5.4: Examples of oversampled bipartite graphs for a five-colorable graph. The circles filled with red and blue represent sets $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ sets, respectively. (a) original graph. (b) foundation bipartite graph with $l = 3$. (c) oversampled graph with $l = 3$. (d) oversampled bipartite graph with $l = 3$. The dashed lines indicate the vertical edges. (e) foundation bipartite graph with $l = 1$. (f) oversampled graph with $l = 1$. (g) oversampled bipartite graph with $l = 1$.

5.2.2 K-Colorable Graphs

For K-colorable graphs where $K \geq 4$, the method described above can be extended to make one bipartite graph including all of the edges of the original graph. We assume that the nodes of the original graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ are assigned colors and divided into K sets $F_1, F_2, \ldots, F_K$. Figure 5.4 shows two examples of the oversampled bipartite graphs for a five-colorable graph. The oversampled bipartite graph is generated according to the following steps:

1. The *foundation bipartite graph* $\mathcal{G}_b = \{\mathcal{L}_b, \mathcal{H}_b, \mathcal{E}_b\}$ is made from the original graph. $\mathcal{L}_b = \{F_1, F_2, \ldots, F_l\}$ and $\mathcal{H}_b = \{F_{l+1}, F_{l+2}, \ldots, F_K\}$, where l is an arbitrary integer value satisfying $1 \leq l \leq K$. $\mathcal{E}_b$ is defined as the edge set containing all edges between $\mathcal{L}_b$ and $\mathcal{H}_b$ (Figs. 5.4(b) and 5.4(e)).

2. The remaining graph $\overline{\mathcal{G}} = \{\mathcal{V}, \mathcal{E} \setminus \mathcal{E}_b\}$ is computed. $\overline{\mathcal{G}}$ has two disjoint graphs: an l-colorable graph $\overline{\mathcal{G}}(\mathcal{L}_b)$ and a $(K-l)$-colorable graph $\overline{\mathcal{G}}(\mathcal{H}_b)$ (Figs. 5.4(c) and 5.4(f)).

3. We place appended nodes F_1' directly above each node in F_1 of the foundation bipartite graph. The nodes in F_1' have the same values as those in F_1.

4. By letting F_1' be in $\widetilde{\mathcal{H}}$, it can be connected freely with the nodes in $\{F_2, \ldots, F_l\}$ since they belong to $\widetilde{\mathcal{L}}$. The edges between F_1' and $\{F_2, \ldots, F_l\}$ are appended in accordance with the edge information of $\overline{\mathcal{G}}(\mathcal{L}_b)$. By using the above operation, all nodes can connect with F_1 or F_1' while keeping the graph bipartite.

5. Steps 3 to 4 are repeated for $F_2, \ldots, F_l$ to yield oversampled sets $F_2', \ldots, F_l'$ and appended new edges in $\overline{\mathcal{G}}(\mathcal{L}_b)$.

6. Similar operations to Steps 3 to 5 can also be applied to the sets in $\mathcal{H}_b$. As a result, the sets $F_{l+1}', \ldots, F_K'$ and the edges in $\overline{\mathcal{G}}(\mathcal{H}_b)$ are appended to the foundation bipartite graph.

Consequently, the sets $F_1', \ldots, F_K'$ and the edges corresponding to $\mathcal{E} \setminus \mathcal{E}_b$ are added to the foundation bipartite graph. Based on the above operations, an oversampled bipartite graph $\widetilde{\mathcal{G}} = \{\widetilde{\mathcal{L}}, \widetilde{\mathcal{H}}, \widetilde{\mathcal{E}}\}$ containing all edges of the original graph is generated as shown in Figs. 5.4(d) and 5.4(g), where $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ respectively include $\mathcal{L}_b$ and $\mathcal{H}_b$. Note that $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ of the oversampled graph become

$$\widetilde{\mathcal{L}} = F_1 \cup \ldots F_l \cup F_{l+1}' \cup \ldots F_K', \tag{5.14}$$

$$\widetilde{\mathcal{H}} = F_1' \cup \ldots F_l' \cup F_{l+1} \cup \ldots F_K. \tag{5.15}$$

Similar to the three-colorable case, vertical edges can be appended and isolated nodes in $\overline{\mathcal{G}}$ will be removed. As a result, the number of the nodes in these sets can be represented as

$$|\widetilde{\mathcal{L}}| = \sum_{i=1}^{l} |F_i| + \sum_{i=l+1}^{K} |F_i'| = N - |\mathcal{I}_{\mathcal{H}_b}|, \tag{5.16}$$

$$|\widetilde{\mathcal{H}}| = \sum_{i=l+1}^{K} |F_i| + \sum_{i=1}^{l} |F_i'| = N - |\mathcal{I}_{\mathcal{L}_b}|, \tag{5.17}$$

where $|\mathcal{I}_{\mathcal{L}_b}|$ and $|\mathcal{I}_{\mathcal{H}_b}|$ are the number of isolated nodes in $\overline{\mathcal{G}}(\mathcal{L}_b)$ and $\overline{\mathcal{G}}(\mathcal{H}_b)$, respectively, and satisfy

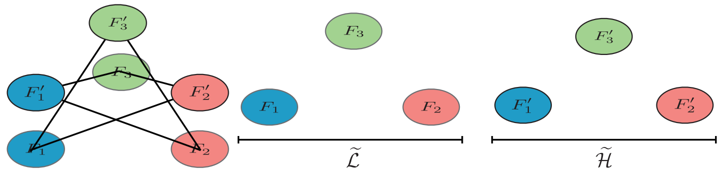
$$\sum_{i=1}^{l} |F_i'| = \sum_{i=1}^{l} |F_i| - |\mathcal{I}_{\mathcal{L}_b}|, \tag{5.18}$$

$$\sum_{i=l+1}^{K} |F_i'| = \sum_{i=l+1}^{K} |F_i| - |\mathcal{I}_{\mathcal{H}_b}|. \tag{5.19}$$

The redundancy after the transformation with the M-channel graph filter bank with graph oversampling can be calculated as

$$\begin{aligned} \rho &= \frac{M((N - |\mathcal{I}_{\mathcal{L}_b}|) + (N - |\mathcal{I}_{\mathcal{H}_b}|))}{2N} \\ &= M - \frac{M(|\mathcal{I}_{\mathcal{L}_b}| + |\mathcal{I}_{\mathcal{H}_b}|)}{2N}. \end{aligned} \tag{5.20}$$

According to the choice of l, there exists $\lfloor \frac{K}{2} \rfloor$ variations of the oversampled graph for K-colorable graphs. For the special case of $l = 1$, $\mathcal{L}_b$ is equal to F_1 and $\overline{\mathcal{G}}(\mathcal{L}_b)$ has



(a) (b)

Figure 5.5: (a) bipartite double cover of a three-colorable graph. (b) its set of $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$.

no edges as shown in Figs. 5.4(e) and 5.4(f). Therefore, we do not need to append nodes just above $\mathcal{L}_b$, and the oversampled bipartite graph becomes

$$\widetilde{\mathcal{L}} = F_1 \cup F_2' \cup \dots F_K', \tag{5.21}$$

$$\widetilde{\mathcal{H}} = F_2 \cup \dots F_K. \tag{5.22}$$

Similarly, when $l = K - 1$, the oversampled bipartite graph becomes

$$\widetilde{\mathcal{L}} = F_1 \cup \dots F_{K-1}, \tag{5.23}$$

$$\widetilde{\mathcal{H}} = F_1' \cup \dots F_{K-1}' \cup F_K. \tag{5.24}$$

The oversampling is done in the same way as described in Section 5.2.1 for three-colorable graphs.

5.2.3 Theoretical Relationship with Bipartite Double Cover

In graph theory, the bipartite double cover of a graph $\mathcal{G}$ is defined as the tensor product $\widetilde{\mathcal{G}}_{BDC} = \mathcal{G} \otimes K_2$, where K_2 is the complete graph of two vertices [62–64]. $\widetilde{\mathcal{G}}_{BDC}$ has $2N$ nodes and $2|\mathcal{E}|$ edges. An example of the bipartite double cover of a three-colorable graph (Fig. 5.3(a)) is shown in Fig. 5.5. The set of foundation nodes $\{F_1, \dots, F_K\}$ is contained in $\widetilde{\mathcal{L}}$ and the set of additional nodes $\{F_1', \dots, F_K'\}$ is in $\widetilde{\mathcal{H}}$. The bipartite double cover is equivalent to the proposed oversampled graph in the case of $l = K$ without vertical edges.

The adjacency matrix of $\widetilde{\mathcal{G}}_{BDC}$ can be represented as

$$\widetilde{\mathbf{A}}_{BDC} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \tag{5.25}$$

and its normalized GLM is

$$\widetilde{\boldsymbol{\mathcal{L}}}_{BDC} = \begin{bmatrix} \mathbf{I} & -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \\ -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} & \mathbf{I} \end{bmatrix}. \tag{5.26}$$

If $\boldsymbol{u}_i$ is an eigenvector of $\boldsymbol{\mathcal{L}}$ with the eigenvalue λ_i, one can immediately see that $\widetilde{\boldsymbol{u}}_{\widetilde{\lambda}_i} = \frac{1}{\sqrt{2}}[\boldsymbol{u}_i^T \; \boldsymbol{u}_i^T]^T$ and $\widetilde{\boldsymbol{u}}_{\widetilde{\lambda}_i} = \frac{1}{\sqrt{2}}[\boldsymbol{u}_i^T \; -\boldsymbol{u}_i^T]^T$ are eigenvectors of $\widetilde{\boldsymbol{\mathcal{L}}}_{BDC}$ with eigenvalues $\widetilde{\lambda}_i = \lambda_i$ and $2 - \lambda_i$, respectively. The graph Fourier coefficient of the oversampled graph signal $\widetilde{\boldsymbol{f}} = [\boldsymbol{f}_0^T \; \boldsymbol{f}_0^T]^T$ associated with λ_i is

$$\widetilde{\boldsymbol{u}}_{\lambda_i}^T \widetilde{\boldsymbol{f}} = \frac{1}{\sqrt{2}} \begin{bmatrix} \boldsymbol{u}_i^T & \boldsymbol{u}_i^T \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{f}_0 \end{bmatrix} = \sqrt{2}\boldsymbol{u}_i^T \boldsymbol{f}_0 = \sqrt{2}\overline{f}_0(\lambda_i), \tag{5.27}$$

and that associated with $2 - \lambda_i$ is

$$\widetilde{\boldsymbol{u}}_{2-\lambda_i}^T \widetilde{\boldsymbol{f}} = \frac{1}{\sqrt{2}} \begin{bmatrix} \boldsymbol{u}_i^T & -\boldsymbol{u}_i^T \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{f}_0 \end{bmatrix} = 0. \tag{5.28}$$

As a result, we can obtain only N nonzero graph Fourier coefficients which are equal to those with the original GLM. In other words, the graph Fourier spectrum using the bipartite double cover is the same as the one using the original graph.

On the other hand, let us define the adjacency matrix of the foundation bipartite graph $\mathcal{G}_b$ of the proposed method as $\mathbf{A}_f$ and that of the remaining graph $\overline{\mathcal{G}}$ as $\mathbf{A}_r$. For simplicity, we will consider the expansion method without vertical edges. The adjacency matrix and degree matrix of our approach become

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_f & \mathbf{A}_r \\ \mathbf{A}_r & \mathbf{0} \end{bmatrix} \tag{5.29}$$

and

$$\widetilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_r \end{bmatrix}, \tag{5.30}$$

where $\mathbf{A} = \mathbf{A}_f + \mathbf{A}_r$ and $\mathbf{D} = \mathbf{D}_f + \mathbf{D}_r$. Its normalized GLM is

$$\widetilde{\boldsymbol{\mathcal{L}}} = \begin{bmatrix} \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_f\mathbf{D}^{-\frac{1}{2}} & -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}} \\ -\mathbf{D}_r^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}^{-\frac{1}{2}} & \mathbf{I} \end{bmatrix}. \tag{5.31}$$

If we assume the oversampled graph has eigenvectors $\widetilde{\boldsymbol{u}}_{\lambda_i} = [\boldsymbol{u}_i^T \ \boldsymbol{u}_i^T]^T$, then it must satisfy

$$\begin{aligned} \widetilde{\boldsymbol{\mathcal{L}}} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_i \end{bmatrix} &= \begin{bmatrix} \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_f\mathbf{D}^{-\frac{1}{2}} & -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}} \\ -\mathbf{D}_r^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}^{-\frac{1}{2}} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_i \end{bmatrix} \\ &= \lambda_i \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_i \end{bmatrix}. \end{aligned} \tag{5.32}$$

The constraint can be simplified as

$$(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_f\mathbf{D}^{-\frac{1}{2}} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}})\boldsymbol{u}_i = \lambda_i\boldsymbol{u}_i. \tag{5.33}$$

On the other hand, the original GLM satisfies

$$\begin{aligned}\lambda_i \boldsymbol{u}_i &= \boldsymbol{\mathcal{L}}\boldsymbol{u}_i \\ &= (\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\boldsymbol{u}_i \\ &= (\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}_f + \mathbf{A}_r)\mathbf{D}^{-\frac{1}{2}})\boldsymbol{u}_i. \qquad (5.34)\end{aligned}$$

Comparing (5.33) and (5.34), $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}^{-\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}}$ has to be satisfied. As a result, $\widetilde{\boldsymbol{\mathcal{L}}}$ has the eigenvector $\widetilde{\boldsymbol{u}}_i = \frac{1}{\sqrt{2}}[\boldsymbol{u}_i^T\ \boldsymbol{u}_i^T]^T$ with $\widetilde{\lambda}_i = \lambda_i$ iff $\mathbf{D}_r = \mathbf{D}$, which is the case of the bipartite double cover. In other cases, the eigenvalues and eigenvectors of the oversampled graph are different from those of the original graph. Hence, we can obtain a different graph Fourier spectrum from that of the original graph by using our approach with $l < K$. Additionally, $\mathbf{A}_r$ has columns and rows whose elements are all zero when $l = K - 1$ or the remaining graph has isolated nodes. In this case, the size of $\widetilde{\boldsymbol{\mathcal{L}}}$ is less than $2N$.

In summary, the proposed oversampled way in the case of $l = K$ without vertical edges is a bipartite double cover, and its graph Fourier spectrum is the same as that of the original graph except for a trivial scaling. The proposed oversampled method with $l < K$ has different eigenvectors from those of the original graph, and its redundancy is less than that of the bipartite double cover.

5.3 Examples of Graph Oversampling

Here, we show examples of graph oversampling for images and arbitrary graphs. Furthermore, we compare the oversampled bipartite graph of *ring graph* with the critically sampled ones.

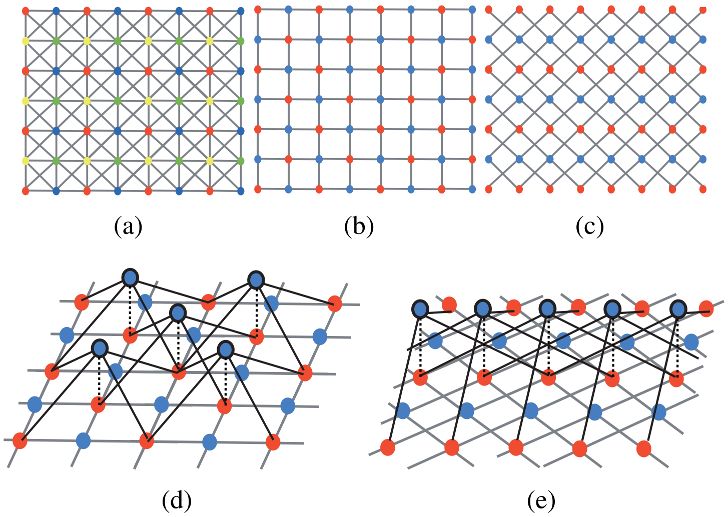


Figure 5.6: (a) image graph. (b) rectangular bipartite subgraph. (c) diagonal bipartite subgraph. (d) oversampled rectangular bipartite graph. (e) oversampled diagonal bipartite graph. The appended nodes are black circles filled with blue, and the appended edges are black lines.

5.3.1 Image Graphs

Images can be viewed as graph signals by connecting each pixel with its eight neighboring ones, as shown in Fig. 5.6(a) [32]. Since this graph is four-colorable, it can be decomposed into rectangular (Fig. 5.6(b)) and diagonal (Fig. 5.6(c)) bipartite subgraphs. If we use critically sampled graph filter banks on the image signal, the diagonal edges will be ignored in a single stage if only the rectangular bipartite graph is used. Moreover, horizontal and vertical edges will be ignored if only the diagonal graph is used. For the critically sampled graph filter banks, a multidimensional transform is applied to multiple bipartite subgraphs to resolve the problem [32, 33]. However, we cannot perform the transform that considers the rectangular and diagonal connections simultaneously.

The above problem can be partially solved by exploiting the oversampled GLM. That is, we append diagonal edges to the rectangular bipartite graph while keeping the

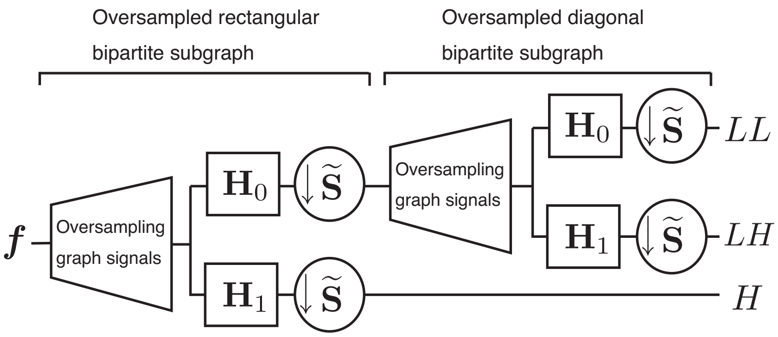


Figure 5.7: One-level decomposition of images. $\mathrm{S}_{\widetilde{\mathcal{L}}_r}/\mathrm{S}_{\widetilde{\mathcal{H}}_r}$ and $\mathrm{S}_{\widetilde{\mathcal{L}}_d}/\mathrm{S}_{\widetilde{\mathcal{H}}_d}$ denote the downsampling operations of the rectangular graph and the diagonal graph, respectively.

oversampled graph bipartite. The rectangular oversampled image graph is illustrated in Fig. 5.6(d). The blue nodes are appended just above the red nodes of the rectangular bipartite graph, and they have diagonal edges connecting them to the red nodes. The additional blue nodes have the same pixel values as the corresponding red nodes. In order for the number of lowpass coefficients to be equal to the critically sampled case, we append nodes only on the $\mathcal{H}$ side. After the downsampling of the critically sampled graph filter bank, the red nodes contain the lowpass component, and their corresponding nodes (additional blue nodes) and blue nodes in the original graph contain the highpass component. Hence, the transform becomes redundant in spite of the use of the critically-sampled transform. At this point, we can use the oversampled diagonal bipartite graph (Fig. 5.6(e)) as a second stage of decomposition. The overall transform including the critically sampled graph filter bank with the oversampled GLM for images is shown in Fig. 5.7. We can iterate this process on the LL subbands to realize a multilevel image decomposition. Thus, with the oversampled bipartite graph, we can transform images with the rectangular graphs *plus* diagonal connections as well as diagonal graphs *plus* horizontal and vertical connections in the single stage transform.

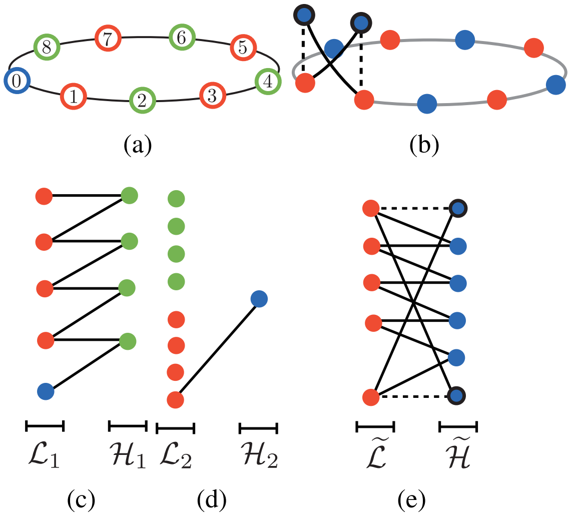


Figure 5.8: (a) ring graph ($n = 4$). (b) oversampled bipartite graph. (c) sets $\mathcal{L}_1$ and $\mathcal{H}_1$ of the bipartite subgraph $\mathcal{B}_1$. (d) sets $\mathcal{L}_2$ and $\mathcal{H}_2$ of the bipartite subgraph $\mathcal{B}_2$. (e) sets $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ of the oversampled bipartite graph.

5.3.2 Ring Graph

All ring graphs with an odd number of nodes are three-colorable as shown in Fig. 5.8(a). Let us assume that the original graph has $2n + 1$ nodes, and F_1, F_2, and F_3 are the red, blue, and green nodes in Fig. 5.8(a), respectively. F_1 and F_3 each have n nodes, and F_2 has 1 node.

The bipartite subgraphs $\mathcal{B}_1$ and $\mathcal{B}_2$ of this ring graph are shown in Figs. 5.8(c) and 5.8(d). The critically sampled graph filter bank must be applied to each bipartite graph. After the signal decomposition using the critically sampled graph filter bank, the LL and HL channels each have n nodes, whereas the LH channel has only one node (the HH channel is empty). Therefore, the number of coefficients in each channel is heavily biased. Because of the graph decomposition basis, the transform for $\mathcal{B}_1$ treats all edges except the one between v_0 and v_1, whereas the transform for $\mathcal{B}_2$ handles only the edge between v_0 and v_1. Although v_2 and v_{2n} are in the 3-hop neighborhood in the original ring graph, their relationship becomes very weak as a result of the bipartite decomposition. That is, v_2 and v_{2n} are in the $(2n-2)$-hop neighborhood of $\mathcal{B}_1$ and are not connected to $\mathcal{B}_2$.

The oversampled bipartite graph of the ring graph and its sets $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ are shown in Figs. 5.8(b) and 5.8(e), respectively. As shown in Fig. 5.8(e), $\widetilde{\mathcal{L}}$ has nodes in F_1 and F_2 and $\widetilde{\mathcal{H}}$ has those in F_1', F_2' and F_3. Therefore, the number of nodes in $\widetilde{\mathcal{L}}$ and $\widetilde{\mathcal{H}}$ are $n+1$ and $n+2$. Hence, the redundancy is only $(2n+3)/(2n+1)$. In the oversampled bipartite graph, all adjacent nodes are connected and all edges of the original graph can be considered in a single stage transform. Furthermore, if we append vertical edges, nodes v_2 and v_{2n} are in a 4-hop neighborhood and have a strong connection like that of the original graph.

5.4 Experimental Results

We performed experiments on images and arbitrary graphs to assess the performances of the oversampled GLM.

5.4.1 Image Processing

We performed non-linear approximation to introduce the potential ability of graph oversampling. The proposed methods are compared with the standard separable CDF 9/7 wavelet filter bank [40], the Laplacian pyramid for regular signals [65], the critically sampled CS-SF filter bank [33], and the Laplacian pyramid for graph signals [66]. The graph-based methods used the same filters (*CS-SF(5,5)*). The Laplacian pyramid for regular signals used 9/7 filters and a reconstruction scheme using the pseudo inverse [67]. The CS-SF used an edge-aware image graph [68]. The edge-aware image graphs were made as follows. The links around the edges of the images are classified into *regular* or

Table 5.1: Reconstruction of Images Using NLA: PSNR (dB)

Fraction of highpass coeffs.	0.00	0.01	0.02	0.04	0.08	0.16
Ballet						
9/7 filter [40]	24.57	31.76	35.83	43.10	52.87	60.05
Laplacian pyramid [65]	24.57	30.49	33.71	38.27	45.74	55.29
CS-SF [32]	31.66	42.39	45.98	50.69	56.10	61.37
graph Laplacian pyramid [66]	31.66	41.37	44.16	47.70	52.00	56.96
proposed (without vertical edges)	30.78	45.12	50.12	55.04	59.35	63.80
proposed (with vertical edges)	**33.01**	**49.97**	**53.51**	**57.02**	**60.35**	**64.58**
Synthetic						
9/7 filter	30.44	37.95	42.96	50.81	**66.73**	**110.10**
Laplacian pyramid	30.44	36.30	39.43	44.23	51.64	66.61
CS-SF	32.92	40.77	44.18	49.34	58.94	76.88
graph Laplacian pyramid	32.92	39.77	42.35	45.60	50.48	58.83
proposed (without vertical edges)	33.22	40.40	43.99	49.83	60.20	81.62
proposed (with vertical edges)	**34.29**	**43.36**	**46.99**	**52.53**	61.53	82.38
Cameraman						
9/7 filter	20.66	23.38	25.30	27.61	30.82	35.73
Laplacian pyramid	20.66	23.73	25.15	27.05	29.81	33.76
CS-SF	21.74	25.82	27.42	29.58	32.46	37.07
graph Laplacian pyramid	21.74	25.28	26.66	28.49	30.76	33.95
proposed (without vertical edges)	21.29	24.47	25.92	27.97	30.69	35.11
proposed (with vertical edges)	**21.75**	**26.26**	**27.78**	**29.81**	**32.72**	**37.12**
Coins						
9/7 filter	23.23	26.49	28.29	30.73	34.32	40.17
Laplacian pyramid	23.23	26.34	27.94	30.11	33.11	37.66
CS-SF	24.78	28.87	30.54	32.77	35.92	**40.88**
graph Laplacian pyramid	24.78	28.43	29.76	31.55	33.94	37.37
proposed (without vertical edges)	24.56	27.21	28.70	30.73	33.93	38.63
proposed (with vertical edges)	**25.15**	**29.11**	**30.75**	**33.09**	**36.22**	40.75

less-reliable links. They were determined by checking that the difference in pixel intensity between the edge pixels is more than or less than a certain threshold. The weights of less-reliable links are set to zero or reduced to a value lower than those of the regular links (Figs. 5.6(a)-(c)). For example, the edge-aware image graphs of Fig. 5.9(a) are shown in Figs. 5.9(b) and 5.9(c). The graph Laplacian pyramid used the same graph and downsampling operation as CS-SF for the lowpass channel. The proposed method

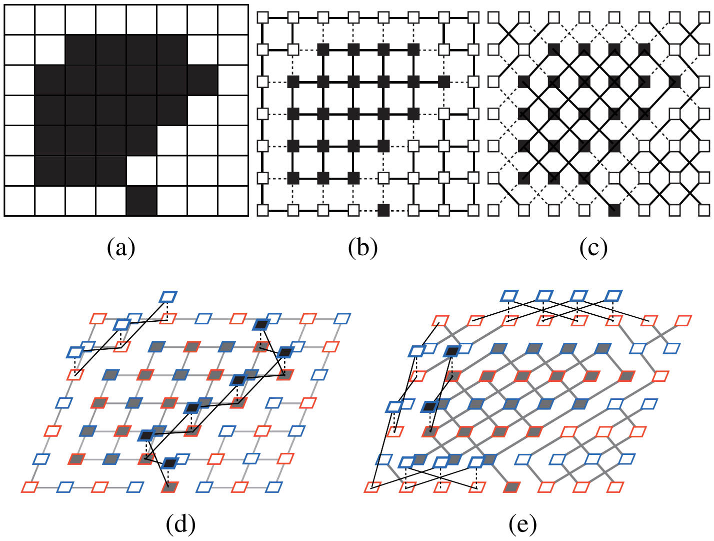


Figure 5.9: (a) original image. (b) edge-aware rectangular bipartite graph. The solid and dashed lines are regular and less-reliable links, respectively. (c) edge-aware diagonal bipartite graph. (d) oversampled edge-aware rectangular bipartite graph. The black lines are additional edges. The dashed black lines indicate vertical edges. The red nodes contain the lowpass component, and the blue nodes contain the highpass component after downsampling. (e) oversampled edge-aware diagonal bipartite graph.

used the oversampled edge-aware image graph [69]. In this case, on the basis of the edge-aware image bipartite graph, the nodes and the links are added along the edges: 1) diagonal direction regular links are added to the edge-aware rectangular graph and 2) rectangular direction regular links are added to the edge-aware diagonal graph. For instance, the oversampled graphs for the rectangular and diagonal bipartite graphs of Fig. 5.9(a) are shown in Figs. 5.9(d) and 5.9(e). The critically sampled graph filter banks are applied to these graphs using the method described in Section 5.3.1.

Table 5.1 lists the PSNRs of the reconstructed images, i.e., reconstructions from all lowpass coefficients and some fraction of the highpass coefficients after the three-level decomposition. Since the fraction of highpass coefficients is relative to the size of the original image, the number of the lowpass and highpass coefficients used for the reconstruction is the same for all methods. However, the ratio of the remaining highpass

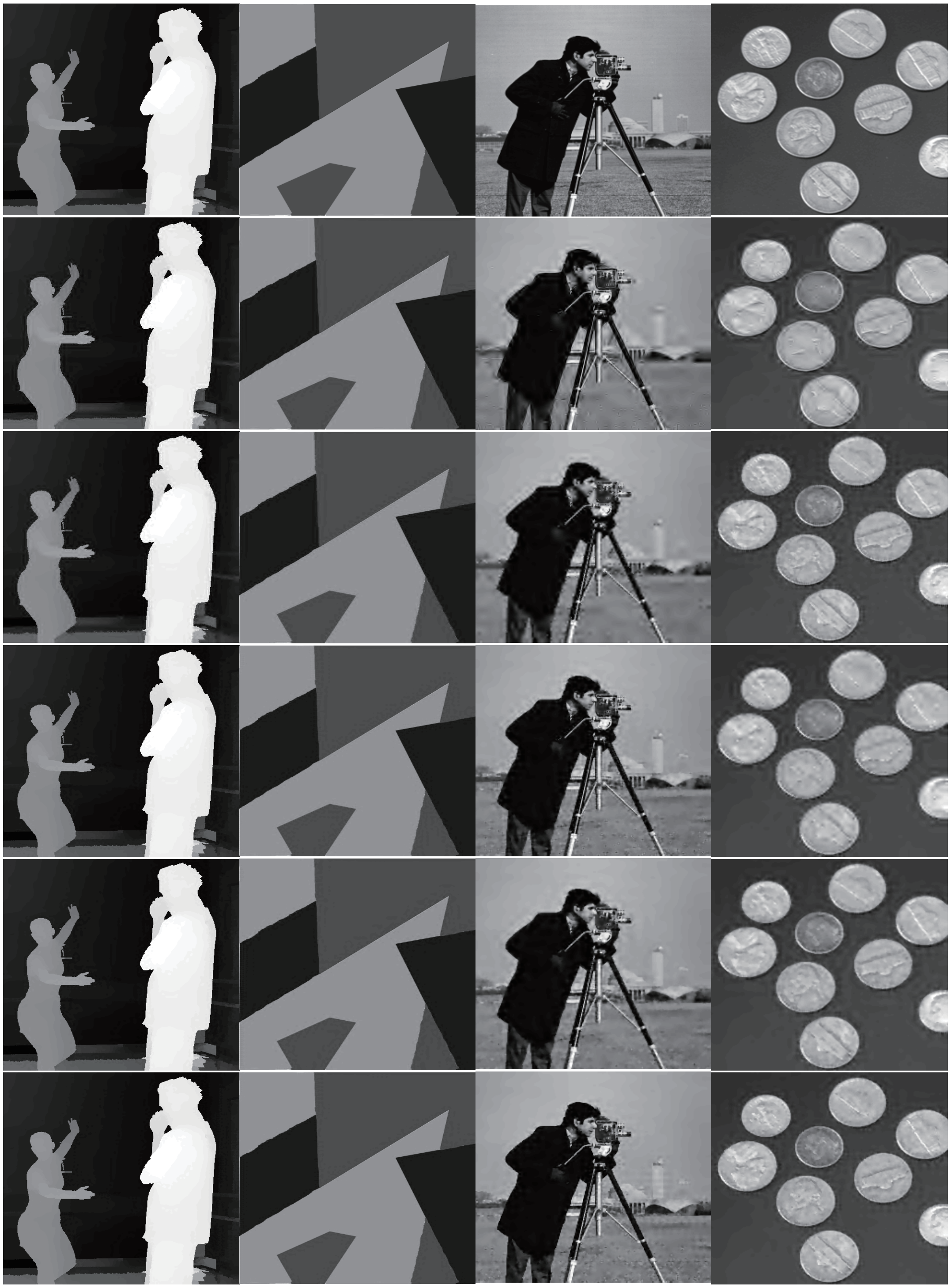


Figure 5.10: Images reconstructed from all lowpass coefficients and 3% of the highpass coefficients after a three-level decomposition. From top to bottom: original image, CDF 9/7 wavelet, the Laplacian pyramid for regular signals, CS-SF, the Laplacian pyramids for graph signals, and the proposed method with vertical edges. From top to bottom: *Ballet*, *Synthetic*, *Cameraman*, and *Coins*.

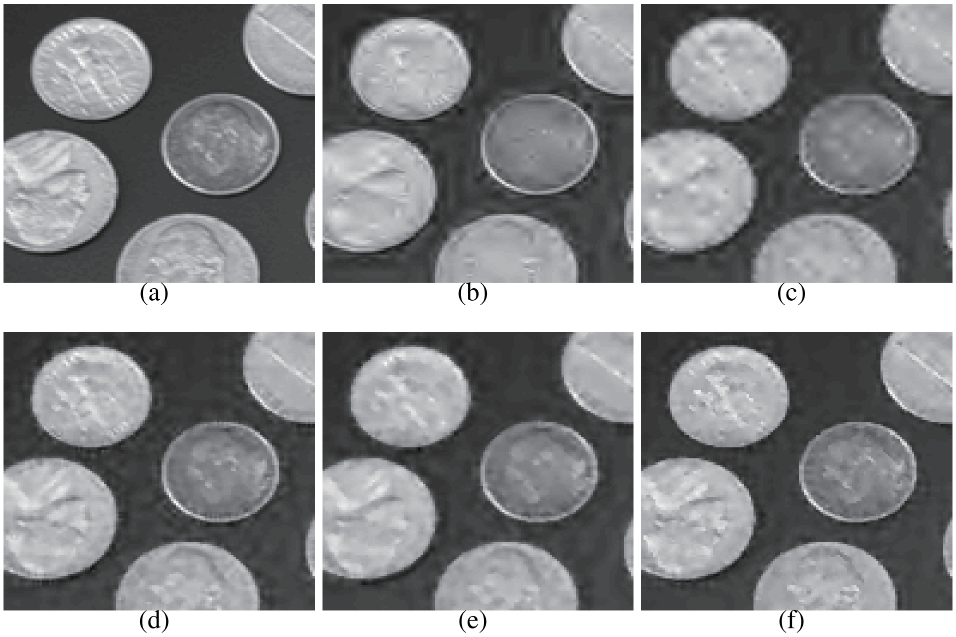


Figure 5.11: Zoomed in *Coin* image reconstructed from lowpass coefficients and 3% of the highpass coefficients after three-level decomposition. (a) original image. (b) CDF 9/7 wavelet (PSNR: 29.64dB). (c) Laplacian pyramid for regular signals (PSNR: 29.13dB). (d) CS-SF (PSNR: 31.75dB). (e) Laplacian pyramid for graph signals (PSNR: 30.75dB) (f) proposed method with vertical edges (PSNR: 32.10dB).

coefficients to the total number of highpass coefficients varies since the Laplacian pyramid and the proposed method are redundant transforms. In spite of this, the proposed method performed better than the other methods, including CS-SF, in most cases. It can be seen that the vertical edges provide significant gains.

Figure 5.10 shows images reconstructed from all lowpass coefficients and 3% of the highpass coefficients, and Fig. 5.11 shows zoomed-in *Coins* images. The standard CDF 9/7 and Laplacian pyramid for regular signals did not take into account the edge information, and as a result, the reconstructed images were blurred around the edges. Since the graph-based transforms consider the rectangular and/or diagonal edges, they preserve the edges well. We can see that blurring and ringing artifacts around the edges

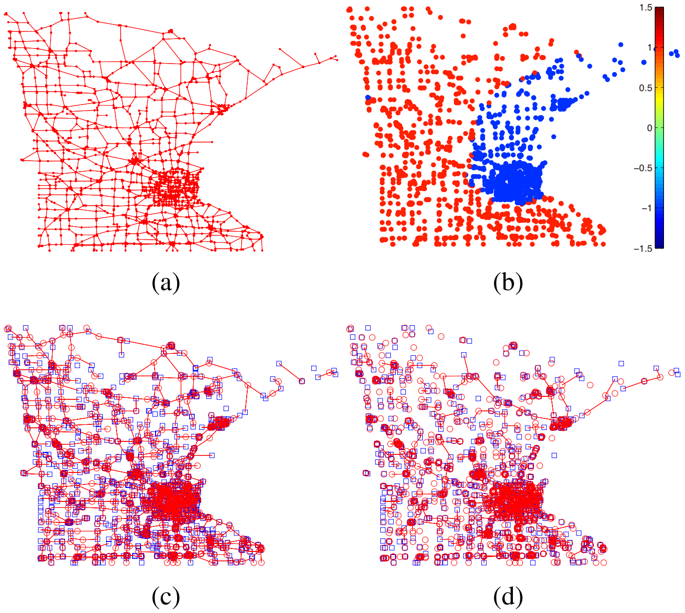


Figure 5.12: (a) original graph of the *Minnesota Traffic Graph*. (b) input signal. The original graph and input signal were reproduced from the MATLAB code of Narang and Ortega [32]. (c) bipartite subgraph #1. The blue squares and red circles indicate sets $\mathcal{L}$ and $\mathcal{H}$, respectively. (d) bipartite subgraph #2.

in the reconstructed image of the proposed method are greatly suppressed compared with other graph-based transforms.

5.4.2 Experiments on Oversampled Graphs

The performance of the proposed oversampled graph was examined in two applications for arbitrary graphs. We made the oversampled bipartite graphs from two original graphs, a three-colorable *Minnesota Traffic Graph* $\mathcal{G}_{MN}$ and a four-colorable *Yale Coat of Arms* $\mathcal{G}_{YC}$, according to the description in Section 5.2. These original graphs can be decomposed into two bipartite graphs by using Harary's algorithm [32, 61], as shown in Figs. 5.12 and 5.13, respectively. The original signals are shown in Figs. 5.12(b) and 5.13(b). We tested a number of setups for the oversampled graphs, such as with/without vertical edges and different values of l of the foundation bipartite graph. Let us denote

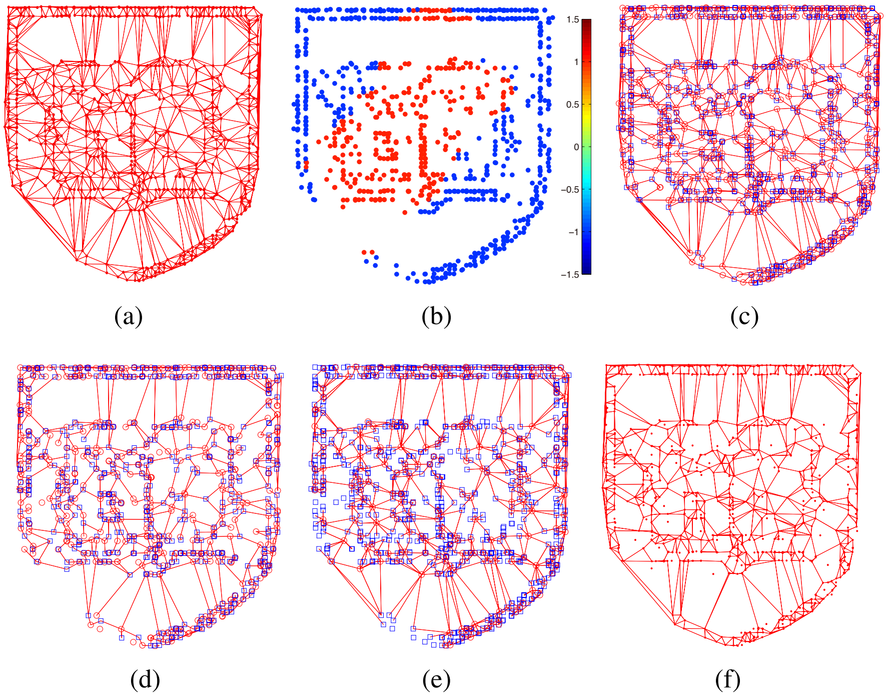


Figure 5.13: (a) original graph of the *Yale Coat of Arms*. It was reproduced from the course website by Spielman [70]. The four-colorable graph was made by removing the nodes assigned the fifth color from the original five-colorable graph. (b) input signal. It was created using SGWT toolbox [13]. (c) bipartite subgraph #1. The blue squares and red circles indicate sets $\mathcal{L}$ and $\mathcal{H}$, respectively. (d) bipartite subgraph #2. (e) foundation graph of $\widetilde{\mathcal{G}}^3_{YC}$. (f) remaining graph of $\widetilde{\mathcal{G}}^3_{YC}$.

an oversampled graph *without* vertical edges as $\widetilde{\mathcal{G}}^l$ and *with* vertical edges as $\widetilde{\mathcal{G}}^{l'}$. The notations of the tested oversampled graphs are summarized in Table 5.2. In order to verify the performance of the proposed oversampled graphs, we applied the CS-SF to each graph.

Denoising

First, we tried denoising the graph signals. The input signal was corrupted by additive white Gaussian noise. After a one-level decomposition, we retained the lowest-frequency subband and the remaining high-frequency subbands were hard-thresholded with $T = 3\sigma$, where σ is the standard deviation of the noise. Table 5.3 summarizes the

Table 5.2: Notation of Oversampled Bipartite Graphs

Graph		$\widetilde{\mathcal{L}}$	$\widetilde{\mathcal{H}}$	Vertical Edges
Minnesota	$\widetilde{\mathcal{G}}^{2'}_{MN}$	F_1, F_2	F_3, F'_1, F'_2	YES
Traffic Graph	$\widetilde{\mathcal{G}}^{2}_{MN}$			NO
	$\widetilde{\mathcal{G}}^{2'}_{YC}$	F_1, F_2, F'_3, F'_4	F_3, F_4, F'_1, F'_2	YES
Yale Coat	$\widetilde{\mathcal{G}}^{2}_{YC}$			NO
of Arms	$\widetilde{\mathcal{G}}^{3'}_{YC}$	F_1, F_2, F_3	F_4, F'_1, F'_2, F'_3	YES
	$\widetilde{\mathcal{G}}^{3}_{YC}$			NO

Table 5.3: Comparison of Oversampled Graphs (Denoising): SNR (dB)

σ	1/32	1/16	1/8	1/4	1/2	1	Redundancy
Minnesota Traffic Graph							
$\widetilde{\mathcal{G}}^{2'}_{MN}$	32.19	26.46	20.58	14.59	8.68	2.80	1.37
$\widetilde{\mathcal{G}}^{2}_{MN}$	**32.46**	**26.76**	**20.88**	**14.94**	**9.00**	**3.11**	1.37
noisy	30.15	24.08	18.06	12.02	5.99	-0.02	–
Yale Coat of Arms							
$\widetilde{\mathcal{G}}^{2'}_{YC}$	31.12	25.20	19.31	13.37	7.69	1.86	1.96
$\widetilde{\mathcal{G}}^{2}_{YC}$	31.14	25.21	19.32	13.36	7.68	1.87	1.96
$\widetilde{\mathcal{G}}^{3'}_{YC}$	31.13	**25.34**	19.45	13.96	8.59	3.14	1.78
$\widetilde{\mathcal{G}}^{3}_{YC}$	**31.33**	25.32	**19.90**	**14.29**	**9.23**	**4.11**	1.78
noisy	30.15	24.10	18.08	12.01	6.04	0.00	–

denoising results. For the *Minnesota Traffic Graph*, $\widetilde{\mathcal{G}}^{2}_{MN}$ performs better than $\widetilde{\mathcal{G}}^{2'}_{MN}$. For the *Yale Coat of Arms*, the SNR of $\widetilde{\mathcal{G}}^{2}_{YC}$ is the almost same as that of $\widetilde{\mathcal{G}}^{2'}_{YC}$ when $l = 2$. In contrast, for $l = 3$, $\widetilde{\mathcal{G}}^{3}_{YC}$ outperforms $\widetilde{\mathcal{G}}^{3'}_{YC}$. Furthermore, the oversampled graph with $l = 3$ provides better SNRs than that with $l = 2$ in spite of it having less redundancy.

Non-linear Approximation

Next, we considered the non-linear approximation of the signal on the *Minnesota Traffic Graph*. We used a two-level decomposition of the proposed methods, i.e., after applying the CS-SF using the oversampled graph, the lowpass signal was further decomposed on the basis of a downsampled graph consisting of vertices in the set $\widetilde{\mathcal{L}}$ and edges in

Table 5.4: Comparison of Oversampled Graphs (Non-linear Approximation): SNR (dB)

# of highpass coeff.	CSGLM	$\widetilde{\mathcal{G}}^{2'}_{MN}$	$\widetilde{\mathcal{G}}^{2}_{MN}$
0	18.49	19.06	**19.10**
30	**26.71**	25.03	26.12
60	32.18	33.04	**33.55**
90	35.02	36.45	**36.72**
120	37.77	39.81	**40.18**
150	40.59	**42.16**	42.02
180	**45.22**	44.76	44.74

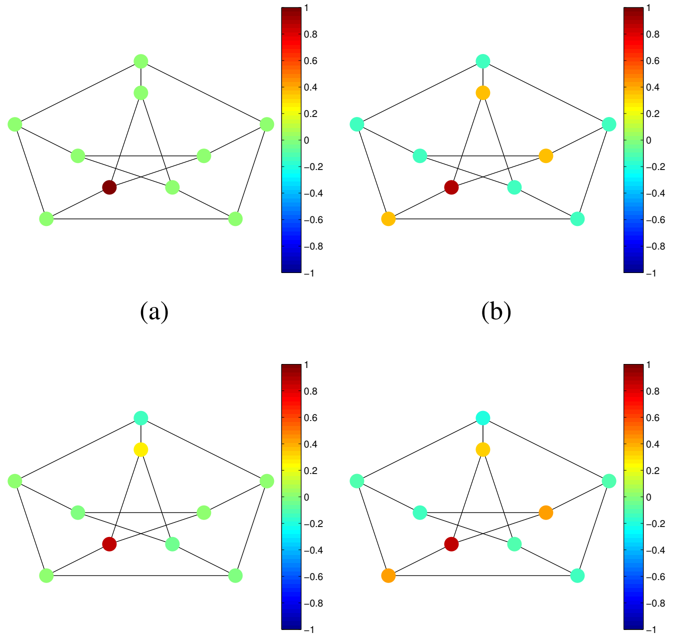
the original graph. Table 5.4 shows the SNR values of the reconstructed signals from all lowpass coefficients and some fraction of the detail coefficients. As a benchmark, we applied CS-SF with critically sampled GLM (CSGLM). The number of lowpass coefficients was the same for all methods.

It can be seen that the proposed method performed better when the reconstructed signal was approximated from only lowpass signals and from all lowpass coefficients with 60–150 highpass coefficients. Since the redundancy of the oversampled graphs is greater than the critically sampled bipartite graph, the critically sampled graph outperformed the oversampled graphs in the case of the reconstruction using > 180 detail coefficients. Additionally, $\widetilde{\mathcal{G}}^{2}_{MN}$ had a better SNR than $\widetilde{\mathcal{G}}^{2'}_{MN}$ in most cases.

For these results, we decide to use oversampled graph with $l = K - 1$ without vertical edges ($\widetilde{\mathcal{G}}^{K-1}$) in the following experiments on arbitrary graphs.

5.4.3 Signal Spread on Arbitrary Graphs

To demonstrate the advantage of the oversampled bipartite graph, we compared the signal spreads of a critically sampled bipartite graph and an oversampled one. The original graph in this case was the Petersen graph, and it was decomposed into the two bipartite subgraphs. The input signal is shown in Fig. 5.14(a). The comparison is between the critically sampled bipartite graph and the oversampled bipartite graph. The



(a)

(b)

(c)

(d)

Figure 5.14: Signal spread. (a) input signal. (b) lowpass filtered signal using the (non-bipartite) original graph. (c) lowpass filtered signal using bipartite subgraph. (d) lowpass filtered signal using oversampled bipartite graph.

lowpass filtered signals are shown in Figs. 5.14(b)–(d). As expected, the spread of the signal after using the oversampled bipartite graph is very similar to that of the original (non-bipartite) graph.

5.4.4 Denoising of Graph Signals

The detailed experiments of graph signal denoising are shown; signals are corrupted by white Gaussian noise. For the proposed method, we applied the CS-SF (abbreviated as CS-SF with OSGLM) [33] or the four-channel oversampled graph filter bank (abbreviated as OS-SF with OSGLM) [71] on oversampled graphs.

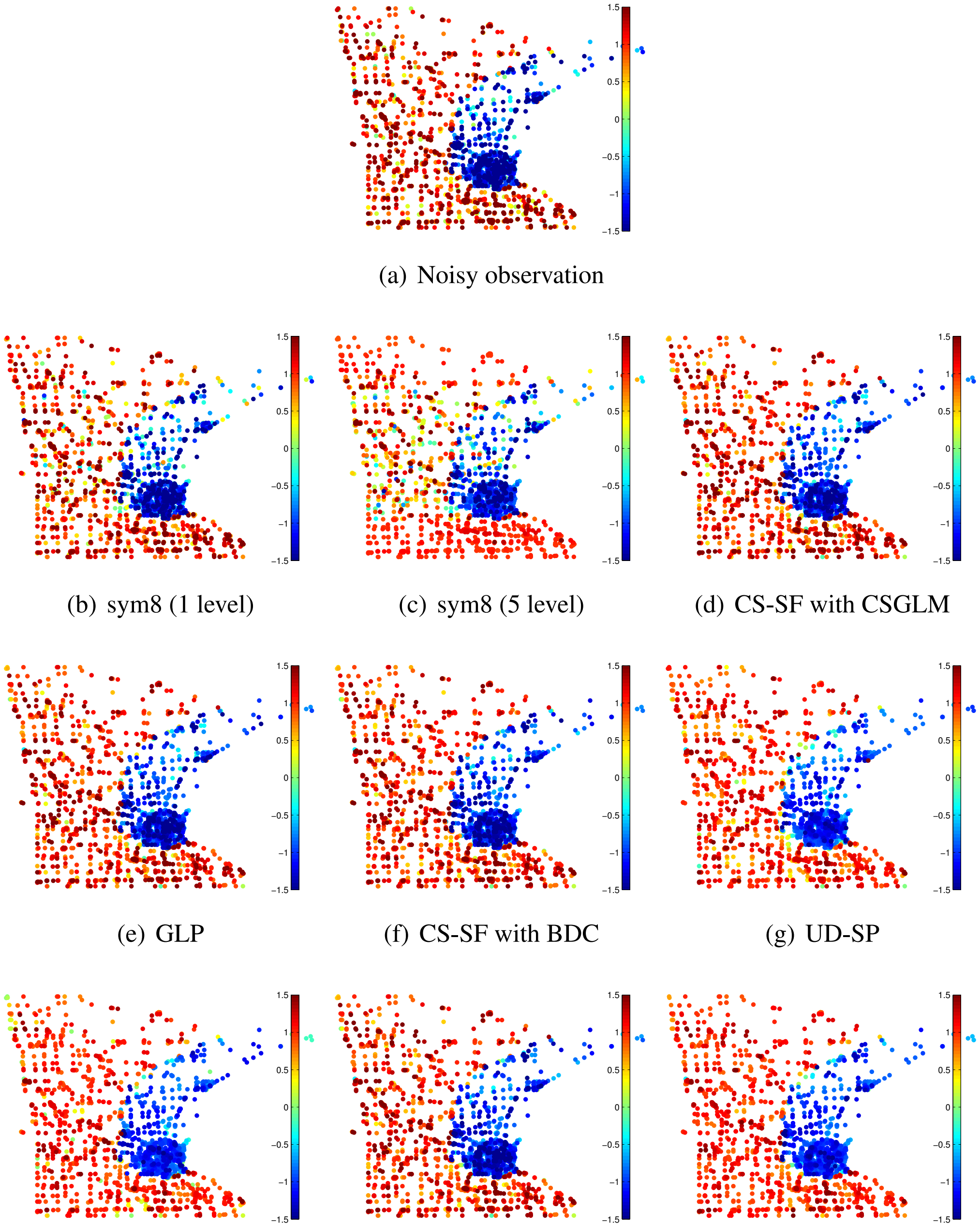
Table 5.5: Denoised Results of Minnesota Traffic Graph: SNR (dB)

σ	1/32	1/16	1/8	1/4	1/2	1	Redundancy
sym8 (1 level)	30.17	24.25	18.65	11.94	6.23	1.59	1
sym8 (5 level)	30.22	24.07	17.99	11.07	5.76	3.13	1
CS-SF (CSGLM)	31.44	25.61	19.97	14.19	8.50	2.63	1
GLP	31.39	25.68	20.02	14.24	8.51	2.61	2.05
UD-SP	33.35	27.76	22.08	15.05	10.33	**8.82**	4
OS-SF (CSGLM)	34.75	28.78	21.84	15.26	10.29	4.24	4
CS-SF (BDC)	32.54	26.75	20.81	14.79	8.92	3.03	2
CS-SF (OSGLM)	32.46	26.76	20.88	14.94	9.00	3.11	1.37
OS-SF (OSGLM)	**35.08**	**29.34**	**23.17**	**17.63**	**12.31**	7.04	2.74
noisy	30.15	24.08	18.06	12.02	5.99	-0.02	-

Table 5.6: Denoised Results of Yale Coat of Arms: SNR (dB)

σ	1/32	1/16	1/8	1/4	1/2	1	Redundancy
sym8 (1 level)	29.64	23.67	18.02	11.36	6.19	1.82	1
sym8 (5 level)	29.41	23.33	17.04	10.19	4.98	2.21	1
CS-SF (CSGLM)	29.85	24.24	18.75	13.16	8.66	3.94	1
GLP	30.10	24.71	19.21	13.58	8.80	4.00	1.79
UD-SP	29.40	23.70	18.24	12.95	8.85	6.47	4
OS-SF (CSGLM)	30.11	24.66	19.05	13.98	10.25	7.45	4
CS-SF (BDC)	31.55	25.43	20.01	14.39	9.23	3.88	2
CS-SF (OSGLM)	31.33	25.32	19.90	14.29	9.23	4.11	1.78
OS-SF (OSGLM)	31.77	26.80	21.22	15.21	10.49	7.48	3.56
noisy	30.15	24.10	18.08	12.01	6.04	0.00	-

We compared the above two methods with the regular one-dimensional wavelet *sym8* with one-level and five-level decompositions, graphBior(6, 6) (CS-SF with CS-GLM) [33], the Laplacian pyramid for graph signals (GLP) [66], the UD-SP with three scales [13], CS-SF with the bipartite double cover (CS-SF with BDC), and the four-channel oversampled graph filter bank with the bipartite graph decomposition (OS-SF with CSGLM) [71, 72]. Since *sym8* treated the signal as a vector, it did not take into account the structure of the signals. For a fair comparison, the graph Laplacian pyramid used the same bipartite graphs and downsampling operation as those of CS-SF for the lowpass channel. All of the graph-based methods performed one-level transforms. That



(a) Noisy observation

(b) sym8 (1 level)

(c) sym8 (5 level)

(d) CS-SF with CSGLM

(e) GLP

(f) CS-SF with BDC

(g) UD-SP

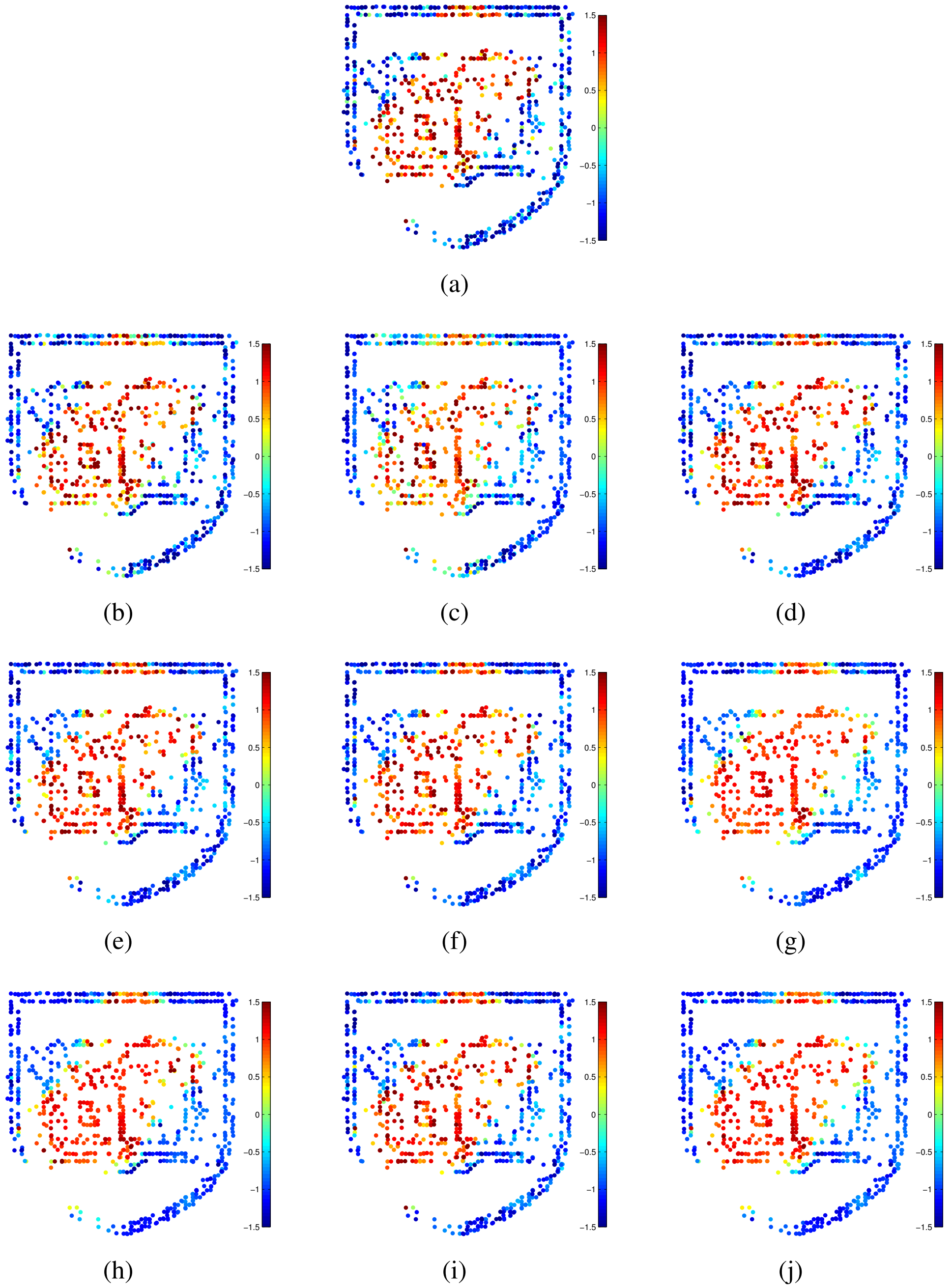
(h) OS-SF with CSGLM

(i) CS-SF with OSGLM

(j) OS-SF with OSGLM

Figure 5.15: Denoising results of *Minnesota Traffic Graph*.

is, CS-SF, the oversampled graph filter bank and the graph Laplacian pyramid performed two-dimensional transforms by using two subgraphs, whereas the proposed methods performed one-dimensional transforms by using the oversampled bipartite graph. The



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

Figure 5.16: Denoised results of the *Yale Coat of Arms*. (a) noisy observation. (b) sym8 (1 level). (c) sym8 (5 level). (d) CS-SF with CSGLM. (e) GLP. (f) CS-SF with BDC. (g) UD-SP. (h) OS-SF with CSGLM. (i) CS-SF with OSGLM. (j) OS-SF with OSGLM.

lowest frequency subband was kept, and the other high-frequency subbands were hard-thresholded with the threshold $T = 3\sigma$.

Tables 5.5 and 5.6 compare the SNRs after denoising. The graph-based transforms outperformed the regular wavelet transforms. OS-SF with OSGLM shows better performance than other graph-based transforms in most cases. It was especially superior to OS-SF with CSGLM and UD-SP on the *Minnesota Traffic Graph*, in spite of it having less redundancy. In comparison with the methods using graphBior filters, CS-SF with BDC and CS-SF with OSGLM have significantly better SNR. Moreover, CS-SF with OSGLM had similar levels of performance as CS-SF with BDC, especially for the strong noise case, despite that its redundancy is less than CS-SF with BDC.

Figures 5.15 and 5.16 show the denoised signals of the *Minnesota Traffic Graph* and the *Yale Coat of Arms* for $\sigma = 1/2$, respectively. Since the regular signal processing did not take into account the structure of the signals, the signals denoised by *sym8* were still noisy. We can see that OS-SF with OSGLM performed better than the other transforms.

5.5 Summary

This chapter presented a method of oversampling graph signals. The method appends nodes and edges to the original graph to construct an oversampled GLM. It is applicable to arbitrary K-colorable graphs, including image graphs. The graph oversampling can consider connections of many nodes while keeping the oversampled graph bipartite. We performed non-linear approximation for images and graph signal denoising experiments showing that our oversampling method outperforms the other transforms.

Chapter 6

Spectral Graph Filters With Low Approximation Error

This chapter aims to design the spectral graph filters having good characteristics. The desired graph filters satisfy following conditions: a) they do not produce DC leakage, b) they are defined by smooth function, c) they satisfy perfect reconstruction conditions, and d) they form a tight frame. However, the filter satisfying all of the conditions does not exist in graph signal processing. Furthermore, the existing wavelets and filter banks in the graph spectral domain do not consider the magnitude of the approximation errors, although they often uses Chebyshev polynomial approximation, and the approximation order for achieving a good approximation has not yet been studied. In addition, to the best of our knowledge, there are no studies on the appropriate approximation orders of each filter in the graph filter banks regardless of their deep connection with localization in the vertex domain. In regular signal processing, the basis functions of wavelets/filter banks can have different lengths according to the features of the signals passed by each filter. Slowly varying signals are well represented by long basis functions, whereas rapidly oscillating signals should be represented by short basis functions [73].

The proposed graph filters are obtained from the real-valued linear phase finite impulse response (FIR) filter banks of regular signal processing. They are defined as a sum of sinusoidal waves, and therefore, they can be easily approximated by lower-order polynomials because of their smoothness. The property yields both theoretical and practical advantages. From a theoretical perspective, the upper bound of the error after the

Chebyshev polynomial approximation can be rigorously calculated. We can determine the approximation order such that the error is less than a pre-determined tolerable value. From a practical perspective, when the proposed filter banks are approximated with the minimum approximation orders whose errors are acceptably small, the highpass filter is more localized in the vertex domain than the lowpass filter, which is similar to the case of regular signal processing.

In this approach, *all* real-valued linear phase filter banks for regular signals are completely reusable for graph signals, and their filter characteristics in the graph spectral domain are easily recognized from their frequency domain counterpart. The filters obtained by the proposed method satisfies the desired conditions listed above and shows better performances than the conventional filters.

Section 6.1 describes the derivation of the proposed spectral graph filter banks and the upper bound of the approximation errors. Section 6.2 shows several examples of designing spectral graph filter banks. For undecimated filter banks, we construct the octave-band and uniform-band spectral graph filter banks based on the undecimated CDF wavelets [40, 41], the DCT [42–44, 52], and the LOT [45, 53, 54]. They form tight frames and do not produce DC leakage. These uniform-band filter banks can be converted into spectrum-adapted filter banks by using the warping function described in [31]. We also show that decimated spectral graph filter banks derived from M-channel linear phase perfect reconstruction filter banks with a downsampling factor of 2 also satisfy the perfect reconstruction condition for graph signals in Section 6.2. The proposed method also enables us to design odd-channel oversampled spectral graph filter banks, whereas the method in Chapter 4 is only for the even-channel case. Section 6.3 presents experimental results on denoising and non-linear approximation applications.

Table 6.1: Four Types of Linear Phase FIR Filters

Type	Filter length	Symmetry
I	odd	symmetric: $h(n) = h(L-1-n)$
II	even	
III	odd	antisymmetric: $h(n) = -h(L-1-n)$
IV	even	

6.1 Spectral Graph Filters Defined by Sinusoidal Waves

Filters in the graph spectral domain should 1) be precisely defined over the interval $[0, \lambda_{\text{max}}]$, and 2) be real-valued functions. Here, we use the discrete-time Fourier transform of the linear phase FIR filters to construct graph filters that consist of sums of sinusoids and have these two properties. Furthermore, we show that the upper bound of the approximation error can be calculated.

6.1.1 Conversion from Frequency Domain to Graph Spectral Domain

Real-valued linear phase FIR filters are divided into four types depending on whether they are symmetric or antisymmetric and whether their filter lengths are even or odd [56]. These types are summarized in Table 6.1, where $h(n),\ n = 0, 1, \ldots, L-1$ is the nth element of a linear phase FIR filter and L is the filter length. In the frequency domain $\omega \in [0, \pi]$, the filter characteristics are formulated using the discrete-time Fourier transform (DTFT). The transform is represented as follows:

$$H(\omega) = \sum_{m=0}^{L-1} h(m) e^{-j\omega m}. \tag{6.1}$$

After elementary calculations, it is found that the filter characteristics of each type of FIR filter $H_{\text{type}}(\omega)$ can be represented in terms of cosine and sine series:

$$
\begin{aligned}
H_{\mathrm{I}}(\omega) =& e^{-j\frac{L-1}{2}\omega}\left(2\sum_{m=0}^{\frac{L-3}{2}} h_{\mathrm{I}}(m)\cos\left(\frac{b_m}{2}\omega\right)+h_{\mathrm{I}}\left(\frac{L-1}{2}\right)\right), && (6.2)\\
H_{\mathrm{II}}(\omega) =& 2e^{-j\frac{L-1}{2}\omega}\sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{II}}(m)\cos\left(\frac{b_m}{2}\omega\right), && (6.3)\\
H_{\mathrm{III}}(\omega) =& 2je^{-j\frac{L-1}{2}\omega}\sum_{m=0}^{\frac{L-3}{2}} h_{\mathrm{III}}(m)\sin\left(\frac{b_m}{2}\omega\right), && (6.4)\\
H_{\mathrm{IV}}(\omega) =& 2je^{-j\frac{L-1}{2}\omega}\sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{IV}}(m)\sin\left(\frac{b_m}{2}\omega\right), && (6.5)
\end{aligned}
$$

where $b_m = L-(2m+1)$. We assume that $h_{\mathrm{III}}(\frac{L-1}{2}) = 0$ for odd-length antisymmetric filters[1]. From the above equations, the modulated frequency characteristics $\widehat{H}_{\text{type}}(\omega) = e^{j\frac{L-1}{2}\omega}H_{\text{type}}(\omega)$ become real (for symmetric filters) or imaginary (for antisymmetric filters) functions that cover $\omega \in [0,\pi]$.

By transforming the variables $\omega \in [0,\pi] \to \lambda \in [0,\lambda_{\max}]$ and multiplying the modulated characteristics by $-j$ for types III and IV, all $\widehat{H}_{\text{type}}(\omega)$ become real-valued

[1] An M-channel linear phase FIR filter bank with odd-length filters does not produce DC leakage iff it satisfies $h(\frac{L-1}{2}) = 0$ for antisymmetric filters.

functions that cover the entire graph frequency range. As a result, the closed form of the spectral graph filters derived from each type of the linear phase FIR filters is

$$H_{\mathrm{I}}(\lambda) = 2\sum_{m=0}^{\frac{L-3}{2}} h_{\mathrm{I}}(m)\cos\left(\frac{b_m\pi}{2\lambda_{\text{max}}}\lambda\right) + h_{\mathrm{I}}\left(\frac{L-1}{2}\right), \tag{6.6}$$

$$H_{\mathrm{II}}(\lambda) = 2\sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{II}}(m)\cos\left(\frac{b_m\pi}{2\lambda_{\text{max}}}\lambda\right), \tag{6.7}$$

$$H_{\mathrm{III}}(\lambda) = 2\sum_{m=0}^{\frac{L-3}{2}} h_{\mathrm{III}}(m)\sin\left(\frac{b_m\pi}{2\lambda_{\text{max}}}\lambda\right), \tag{6.8}$$

$$H_{\mathrm{IV}}(\lambda) = 2\sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{IV}}(m)\sin\left(\frac{b_m\pi}{2\lambda_{\text{max}}}\lambda\right). \tag{6.9}$$

These filter characteristics in the graph spectral domain consist of a finite number of sinusoidal waves. To perform the filtering in the graph spectral domain without eigendecomposition of the graph Laplacian, the Chebyshev polynomial approximation [13] is often used as described in Section III-C. We present the upper bound of the approximation error in the next subsection.

6.1.2 Upper Bound of Approximation Error

The overall computational complexity in applying an M-channel spectral graph filter bank with a Chebyshev polynomial approximation is $\mathcal{O}(p_{\text{max}}|\mathcal{E}| + N\sum_{m=0}^{M-1} p_m)$, where $p_{\text{max}} = \max\{p_m | m = 0, \ldots, M-1\}$ and p_m is the approximation order of the mth filter $H_m(\lambda)$ [1, 13]. $\mathcal{O}(p_{\text{max}}|\mathcal{E}|)$ is the cost of computing the Chebyshev polynomials $T_k(\mathbf{L})\boldsymbol{f}$, and $\mathcal{O}(N\sum_{i=0}^{M-1} p_i)$ is the cost of computing the coefficients for each scale. Therefore, if the complexity is dominated by the cost of computing $T_k(\mathbf{L})\boldsymbol{f}$, the overall computational complexity depends only on the maximum approximation order p_{max}.

Here, let us derive the maximum error of the Chebyshev polynomial approximation as $E_{\max,p} = \max \left| H(\lambda) - \widetilde{H}_p(\lambda) \right|$ for any $\lambda \in \sigma(\mathbf{L})$, where $H(\lambda)$ is the original function, and $\widetilde{H}_p(\lambda)$ is the pth order approximated function of $H(\lambda)$. The following proposition gives an upper bound of the error due to the Chebyshev polynomial approximation of the proposed filters.

Proposition 2. *If a filter characteristic is derived from the linear phase FIR filter in the frequency domain in* (6.6)–(6.9)*, an upper bound of the error due to the Chebyshev polynomial approximation is represented as follows:*

$$\begin{aligned} E_{\max,p} &= \max |H(\lambda) - \widetilde{H}_p(\lambda)| \\ &\leq \frac{1}{2^{p-1}\,(p+1)!} \sum_{m=0}^{Q-1} |h(l)| \left(\frac{b_m \pi}{4} \right)^{p+1} \end{aligned} \tag{6.10}$$

for any $\lambda \in \sigma(\mathbf{L})$, where $Q = (L-1)/2$ for type I or III and $Q = L/2$ for type II or IV.

Proof. For simplicity, we will prove the case of the type II filter. The upper bound of the approximation error of an arbitrary function $H\left(\xi\right)$ is represented as

$$E_{\max,p} \leq \frac{1}{2^p (p+1)!} \max \left| \frac{d^{(p+1)}}{d\xi^{(p+1)}} H\left(\xi\right) \right|, \tag{6.11}$$

for any $\xi \in [-1, 1]$ [74], [30, Theorem 2.4.6]. To calculate the upper bound of the approximation error of the filter defined in $\lambda \in [0,\ \lambda_{\max}]$, we shift the domain from $\lambda \in [0, \lambda_{\max}]$ to $\xi \in [-1, 1]$ by using the transformation $\lambda = \lambda_{\max}\left(\xi + 1\right)/2$. As a result, the upper bound of the approximation error of $H_{\text{II}}(\lambda)$ is represented as

$$E_{\max,p} \leq \frac{1}{2^p (p+1)!} \max \left| \frac{d^{(p+1)}}{d\xi^{(p+1)}} H_{\text{II}} \left(\frac{\lambda_{\max}}{2} (\xi + 1) \right) \right|, \tag{6.12}$$

for any $\xi \in [-1, 1]$. The shifted filter characteristic is represented as

$$H_{\mathrm{II}}\left(\lambda_{\max}(\xi+1)/2\right) = 2\sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{II}}(m)\cos\left(\frac{b_m\pi}{4}(\xi+1)\right). \tag{6.13}$$

It is differentiable $(p+1)$ times, and its $(p+1)$th derivative is

$$\frac{d^{(p+1)}}{d\xi^{(p+1)}} H_{\mathrm{II}}\left(\lambda_{\max}\left(\xi+1\right)/2\right) = \\ \begin{cases} 2\left(-1\right)^{\frac{p+2}{2}} \displaystyle\sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{II}}(m)\left(\frac{b_m\pi}{4}\right)^{p+1} \sin\left(\frac{b_m\pi}{4}\left(\xi+1\right)\right) & \text{for even } p, \\ 2\left(-1\right)^{\frac{p+1}{2}} \displaystyle\sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{II}}(m)\left(\frac{b_m\pi}{4}\right)^{p+1} \cos\left(\frac{b_m\pi}{4}\left(\xi+1\right)\right) & \text{for odd } p. \end{cases} \tag{6.14}$$

Therefore, the maximum absolute value of the $(p+1)$th derivative of the shifted filter characteristic is

$$\max\left|\frac{d^{(p+1)}}{d\xi^{(p+1)}} H_{\mathrm{II}}\left(\frac{\lambda_{\max}}{2}\left(\xi+1\right)\right)\right| \le 2\sum_{m=0}^{\frac{L}{2}-1} \left|h_{\mathrm{II}}(m)\right| \left(\frac{b_m\pi}{4}\right)^{p+1}. \tag{6.15}$$

Substituting (6.15) for (6.12), we obtain (6.10).

Although we only derived an upper bound of the approximation error in the case of a type II filter, the cases of the other types of real-valued linear phase FIR filter can be proved similarly. □

6.2 Graph Filter Banks Derived from Linear Phase FIR Filter Banks

In this section, we show the examples of the spectral graph wavelets and filter banks derived from linear phase perfect reconstruction filter banks in regular signal processing and prove that they satisfy the perfect reconstruction conditions for graph signals.

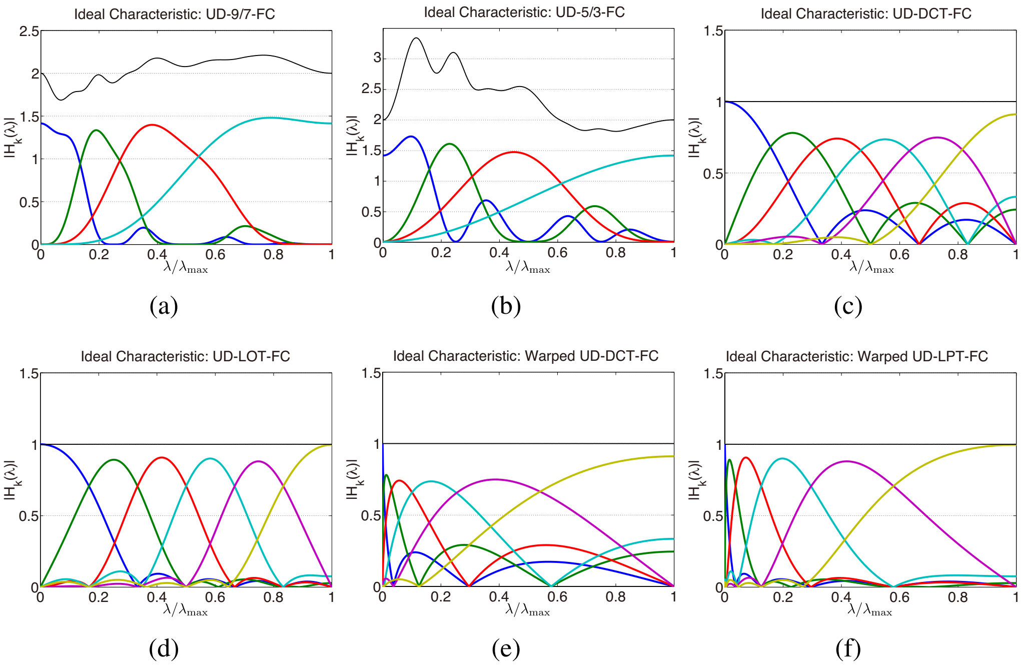
6.2.1 M-Channel Undecimated Graph Filter Banks

Octave-band Undecimated Spectral Graph Wavelets

The octave-band undecimated spectral graph wavelets can be constructed from the octave-band linear phase wavelets in regular signal processing which is obtained by removing the downsampling operations from the CDF filter banks [41]. We calculate the filter coefficients of the undecimated CDF 9/7 and 5/3 wavelets with $M = 4$ and convert them into filters in the graph spectral domain, as shown in Figs. 6.1(a) and 6.1(b). Clearly, the UD-9/7-FC and UD-5/3-FC are not tight-frames, i.e., $F(\lambda)$ is not constant due to the filter characteristic in the frequency domain.

Uniform-band Undecimated Spectral Graph Filter Banks

The DCT and LOT are used as examples of uniform-band spectral graph filter banks. Whereas these filter banks are the decimated filter banks with a decimation factor of M in regular signal processing, we use them as undecimated filter banks by removing downsampling operations as in the octave-band ones. This is because the filter banks with a decimation factor of 2 for regular signals are the only ones that guarantee perfect reconstruction for graph signals, as described Section V-C.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.1: Ideal filter characteristics of the proposed undecimated spectral graph filter banks (black lines indicate $F(\lambda)$). (a) UD-9/7-FC ($M = 4$). (b) UD-5/3-FC ($M = 4$). (c) UD-DCT-FC ($M = 6$). Each filter coefficient of the proposed filter banks is divided by $\sqrt{M}$ so that $F(\lambda) = 1$. (d) UD-LOT-FC ($M = 6$). (e) warped UD-DCT-FC with warping function $\widetilde{\omega}(\lambda) = \lambda^{\frac{1}{3}}$. (f) warped UD-LOT-FC.

Filters in the DCT for even M are type II for even-indexed and type IV for odd-indexed filters. Therefore, the kth filter of the UD-DCT-FC can be represented using (6.7) and (6.9):

$$H_{k,\text{DCT}}(\lambda) = \begin{cases} 2s_{\text{DCT}} \displaystyle\sum_{m=0}^{\frac{M}{2}-1} \cos\left(\frac{(2m+1)\pi}{2M}k\right) \cos\left(\frac{b_m \pi}{2\lambda_{\max}}\lambda\right) & \text{for even } k, \\ 2s_{\text{DCT}} \displaystyle\sum_{m=0}^{\frac{M}{2}-1} \cos\left(\frac{(2m+1)\pi}{2M}k\right) \sin\left(\frac{b_m \pi}{2\lambda_{\max}}\lambda\right) & \text{for odd } k. \end{cases} \tag{6.16}$$

where $L = M$ and s_{DCT} is as in (3.8). The transform inherits the advantages of the DCT: it is composed of a set of smooth functions and does not produce DC leakage. In the same way, the UD-LOT-FC with $L = 2M$ can be obtained by substituting $h_{k,\text{LOT}}(m) =$

$C_{\text{LOT}}(k,m),\ k = 0,1,\ldots,M-1,\ m = 0,1,\ldots,2M-1$ into (6.7) (for even k) and (6.9) (for odd k). The ideal filter responses of UD-DCT-FC and UD-LOT-FC are shown in Figs. 6.1(c) and 6.1(d), respectively. Clearly, they are equivalent to their regular signal processing counterparts.

Frame Bounds

Let us calculate the frame bounds of the undecimated spectral graph filter banks derived from the linear phase FIR filter banks. Here, we define $\mathbf{T}_a = [H_0(\lambda)\ H_1(\lambda)\ \ldots H_{M-1}(\lambda)]^T$ as an analysis transform of the spectral graph filter banks. Then,

$$\mathbf{T}_a^{\dagger}(\lambda)\mathbf{T}_a(\lambda) = \sum_{k=0}^{M-1} |H_k(\lambda)|^2 = F(\lambda). \tag{6.17}$$

Proposition 3. *If a linear phase FIR filter bank is paraunitary, the converted undecimated filter bank for graph signals satisfies*

$$F(\lambda) = M \tag{6.18}$$

for all $\lambda \in \sigma(\mathbf{L})$; i.e., the spectral graph filter bank is a tight one with frame bounds $A = B = M$.

Proof.

Here, let us consider the graph spectral filter bank derived from a linear phase FIR filter bank whose filter length $L = DM$. Their $M \times DM$ basis matrix is $\boldsymbol{\Psi}_b = [\boldsymbol{\Psi}_0\ \boldsymbol{\Psi}_1 \ldots \boldsymbol{\Psi}_{D-1}] = [\boldsymbol{h}_0^T\ \boldsymbol{h}_1^T\ \ldots\ \boldsymbol{h}_{M-1}^T]^T$, where $\boldsymbol{h}_k = [h_k(0)\ h_k(1)\ \ldots\ h_k(L-1)]$ is the kth basis function in the time domain of regular signal processing and $\boldsymbol{\Psi}_d$, $d = 0,\ldots,D-1$ are $M \times M$ matrices. The analysis transform matrix is represented as

$$\mathbf{T}_a(\lambda) = e^{j\frac{DM-1}{2}\widehat{\lambda}}\mathbf{K}\boldsymbol{\Psi}_b\boldsymbol{\Phi}_{\text{DTFT}}(\widehat{\lambda}), \tag{6.19}$$

where $\widehat{\lambda} = \frac{\pi}{\lambda_{\text{max}}}\lambda$, $\mathbf{\Phi}_{\text{DTFT}}(\widehat{\lambda}) = [e^{-j\widehat{\lambda}\cdot 0} \ \ldots \ e^{-j\widehat{\lambda}\cdot(L-1)}]^T$, and $\mathbf{K}$ is a diagonal matrix whose (n, n)-th element is

$$K(n,n) = \begin{cases} 1 & \text{if } \boldsymbol{h}_n \text{ is a symmetric filter,} \\ -j & \text{if } \boldsymbol{h}_n \text{ is an antisymmetric filter.} \end{cases} \tag{6.20}$$

From the paraunitary property, the basis satisfy the following conditions [58, Theorem 9.5]:

$$\sum_{d=0}^{D-1} \mathbf{\Psi}_d^T \mathbf{\Psi}_d = \mathbf{I}. \tag{6.21}$$

Then $\mathbf{T}_a^\dagger(\lambda)\mathbf{T}_a(\lambda)$ can be expanded as

$$\begin{aligned} F(\lambda) =& \mathbf{T}_a^\dagger(\lambda)\mathbf{T}_a(\lambda) \\ =& \mathbf{\Phi}_{\text{DTFT}}^\dagger(\widehat{\lambda})\mathbf{\Psi}_b^T\mathbf{K}^\dagger\mathbf{K}\mathbf{\Psi}_b\mathbf{\Phi}_{\text{DTFT}}(\widehat{\lambda}) \\ =& \mathbf{\Phi}_{\text{DTFT}}^\dagger(\widehat{\lambda})\left(\sum_{i=0}^{D-1}\mathbf{\Psi}_i^T\mathbf{\Psi}_i\right)\mathbf{\Phi}_{\text{DTFT}}(\widehat{\lambda}) \\ =& \mathbf{\Phi}_{\text{DTFT}}^\dagger(\widehat{\lambda})\mathbf{\Phi}_{\text{DTFT}}(\widehat{\lambda}) \\ =& M, \end{aligned} \tag{6.22}$$

i.e., $F(\lambda) = M$ is satisfied regardless of the value of λ. Therefore, the frame bounds of M-channel spectral graph filter banks derived from linear phase FIR paraunitary filter banks are always $A = B = M$. □

We can obtain a Parseval frame by dividing each filter coefficient by $\sqrt{M}$.

Warped Filter Banks

The undecimated spectral graph filter banks derived from the linear phase FIR filter banks can be warped with the warping function $\widetilde{\omega}(\lambda)$, similarly to [31]:

$$H_k^{\text{Warped}}(\lambda) = H_k(\widetilde{\omega}(\lambda)),\ k = 0, 1, \dots, M-1. \tag{6.23}$$

The warped UD-DCT-FC and UD-LOT-FC with the warping function[2] $\widetilde{\omega}(\lambda) = \lambda^{\frac{1}{3}}$ are shown in Figs. 6.1(e) and 6.1(f), respectively.

The spectrum-adapted spectral graph filter banks can also be constructed with the warping function generated from the cumulative spectral density function [31]. The frame bound of the warped filter bank is also M as long as $\widetilde{\omega}(0) = 0$, $\widetilde{\omega}(\lambda_{\max}) = \lambda_{\max}$, since the warping function is a monotonically increasing function and the original linear phase FIR filter bank is paraunitary. This is because $F(\lambda)$ is always constant, as mentioned above.

6.2.2 M-Channel Graph Filter Banks with Downsampling Factor of Two

Perfect Reconstruction Conditions

Many critically sampled and oversampled spectral graph filter banks so far are only applicable to bipartite graphs. The downsampling corresponds to decimation with a factor of 2 for regular signals. For example, downsampling signals on a path graph corresponds to the operation of deleting every other sample. The decimated spectral graph filter banks use a normalized graph Laplacian matrix $\boldsymbol{\mathcal{L}}$, and therefore, $\lambda_{\text{max}} = 2$.

[2] Although $\log(x)$ is used for the warping function in [31], we use a function satisfying $\widetilde{\omega}(0) = 0$ in order to avoid $\log(0) = -\infty$.

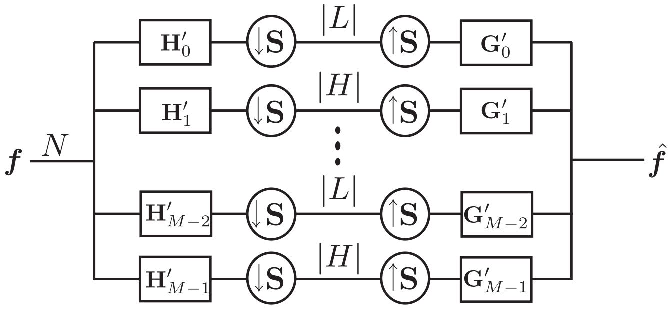


Figure 6.2: M-channel decimated spectral graph filter bank with filters obtained from linear phase FIR filters.

Let us define the filters in the analysis and synthesis decimated graph filter banks as $\{H'_0(\lambda), H'_1(\lambda), \ldots, H'_{M-1}(\lambda)\}$ and $\{G'_0(\lambda), G'_1(\lambda), \ldots, G'_{M-1}(\lambda)\}$, respectively, as shown in Fig. 6.2. $H'_k(\lambda)$ and $G'_k(\lambda)$ are yielded by substituting the filter coefficients into (6.6)–(6.9) according to the type of filter in Table 6.1 with one slight modification. We should multiply -1 with a synthesis-side graph filter that corresponds to an anti-symmetric filter for regular signals to ensure perfect reconstruction. As a result, $H'_k(\lambda)$ and $G'_k(\lambda)$ are represented as

$$H'_k(\lambda) = H_k(\lambda) \tag{6.24}$$

for all types, and

$$G'_k(\lambda) = \begin{cases} G_k(\lambda) & \text{if } g_k(m) \text{ is type I or II,} \\ -G_k(\lambda) & \text{if } g_k(m) \text{ is type III or IV,} \end{cases} \tag{6.25}$$

where $H_k(\lambda)$ and $G_k(\lambda)$ are obtained by substituting the impulse responses $h_k(m)$ and $g_k(m)$ into (6.6)–(6.9).

In the proposed M-channel decimated spectral graph filter bank ($M \geq 2$) shown in Fig. 6.2, the even-indexed subbands retain the nodes in L, whereas in [71, 72], the first $M/2$ subbands retain those nodes. The proposed filter bank has the same condition as

(4.9). The condition for cancelling the spectral folding phenomenon (4.10) is modified as follows:

$$\begin{cases} \displaystyle\sum_{k=0}^{M/2-1} -G_{2k}(\lambda)H_{2k}(2-\lambda) + G_{2k+1}(\lambda)H_{2k+1}(2-\lambda) = 0 \quad \text{for even } M, \\ \displaystyle\sum_{k=0}^{(M-3)/2} -G_{2k}(\lambda)H_{2k}(2-\lambda) + G_{2k+1}(\lambda)H_{2k+1}(2-\lambda) \\ \qquad\qquad - G_{M-1}(\lambda)H_{M-1}(2-\lambda) = 0 \quad \text{for odd } M. \end{cases} \tag{6.26}$$

The following proposition guarantees perfect reconstruction of these filter banks.

Proposition 4. *The critically sampled and oversampled spectral graph filter banks for bipartite graphs, which are derived from a linear phase perfect reconstruction filter bank with a downsampling factor of* 2*, from* (6.6)–(6.9)*,* (6.24) *and* (6.25)*, satisfy the perfect reconstruction conditions in* (3.34) *and* (3.35) *(for critically sampled spectral graph filter banks), or* (4.9) *and* (6.26) *(for oversampled spectral graph filter banks).*

Proof. Let us define $\breve{H}(z) = H(-z)$. Since $-z = -e^{j\omega} = e^{j(\omega+\pi)}$, $\breve{H}(z)$ in the frequency domain $\omega \in [0, \pi]$ can be represented as $\breve{H}(\omega) = H(\omega+\pi)$. From (6.3), type II of $\breve{H}_{\mathrm{II}}(\omega)$ becomes

$$\begin{aligned} \breve{H}_{\mathrm{II}}(\omega) &= H_{\mathrm{II}}(\omega+\pi) \\ &= 2e^{-j\frac{L-1}{2}(\omega+\pi)} \sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{II}}(m)\cos\left(\frac{b_m}{2}(\omega+\pi)\right) \\ &= 2j(-1)^{\frac{L}{2}} e^{-j\frac{L-1}{2}\omega} \sum_{m=0}^{\frac{L}{2}-1} h_{\mathrm{II}}(m)\cos\left(\frac{b_m}{2}(\omega+\pi)\right). \end{aligned} \tag{6.27}$$

Table 6.2: Four Types of Flipped Filters

Type	Modulating function	$\breve{H}_{\text{type}}(\lambda)$
I	$e^{j\frac{L-1}{2}\omega}$	$(-1)^{\frac{L-1}{2}} H_{\text{I}}(2-\lambda)$
II	$-je^{j\frac{L-1}{2}\omega}$	$(-1)^{\frac{L-2}{2}} H_{\text{II}}(2-\lambda)$
III	$-je^{j\frac{L-1}{2}\omega}$	$(-1)^{\frac{L-3}{2}} H_{\text{III}}(2-\lambda)$
IV	$e^{j\frac{L-1}{2}\omega}$	$(-1)^{\frac{L-2}{2}} H_{\text{IV}}(2-\lambda)$

By multiplying the modulated characteristics by $-je^{j\frac{L-1}{2}\omega}$ and transforming the variables $\omega \in [0, \pi] \to \lambda \in [0, 2]$, we obtain

$$\begin{aligned} \breve{H}_{\text{II}}(\lambda) &= (-1)^{\frac{L}{2}} \left(2 \sum_{m=0}^{\frac{L}{2}-1} h_{\text{II}}(m) \cos\left(\frac{b_m}{2} \left(\frac{\pi\lambda}{2} + \pi \right) \right) \right) \\ &= (-1)^{\frac{L-2}{2}} \left(2 \sum_{m=0}^{\frac{L}{2}-1} h_{\text{II}}(m) \cos\left(\frac{b_m \pi}{4} (2-\lambda) \right) \right) \\ &= (-1)^{\frac{L-2}{2}} H_{\text{II}}(2-\lambda). \end{aligned} \tag{6.28}$$

Table 6.2 lists each type of $\breve{H}_{\text{type}}(\lambda)$; these were derived using a similar approach to that for type II, with a modulating function multiplied so as to obtain a real-valued function (6.28) from (6.27).

Case 1: Critically sampled spectral graph filter banks with odd-length filters. By multiplying both sides by $e^{j\frac{L_0+L_1-2}{2}\omega}$, (3.10) in the frequency domain $\omega \in [0, \pi]$ can be rewritten as[3]

$$\widehat{G}_0(\omega)\widehat{H}_0(\omega) + \widehat{G}_1(\omega)\widehat{H}_1(\omega) = 2. \tag{6.29}$$

Accordingly, the graph filters derived by transforming the variable $\omega \in [0, \pi] \to \lambda \in [0, 2]$ and using (6.24) and (6.25) have the following relationship:

$$G_0'(\lambda)H_0'(\lambda) + G_1'(\lambda)H_1'(\lambda) = 2. \tag{6.30}$$

[3] From Theorem 1, both synthesis-side filters are symmetric in this case.

Similarly, from (3.11), we obtain

$$\begin{aligned}
&G_0(\lambda)\breve{H}_0(\lambda) + G_1(\lambda)\breve{H}_1(\lambda) = 0, \\
&\xrightarrow[\text{Table 6.2, (6.24), (6.25)}]{} (-1)^{\frac{L_0-1}{2}} G_0'(\lambda)H_0'(2-\lambda) + (-1)^{\frac{L_1-1}{2}} G_1'(\lambda)H_1'(2-\lambda) = 0.
\end{aligned} \tag{6.31}$$

From Theorem 1, the possible forms of the filter lengths are $\{L_0 = 4n_0 + 1, L_1 = 4n_1 + 3\}$, or $\{L_0 = 4n_0 + 3, L_1 = 4n_1 + 1\}$, where $n_0, n_1 \in \mathbb{R}$. In both cases, (6.31) is equal to (3.35).

Case 2: Critically sampled spectral graph filter banks with even-length filters. In this case, the highpass filter $G_1(z)$ on the synthesis side is type IV. Similar to Case 1, (3.10) can be modulated as

$$\begin{aligned}
&\widehat{G}_0(\omega)\widehat{H}_0(\omega) - (-j\widehat{G}_1(\omega))(-j\widehat{H}_1(\omega)) = 2, \\
&\xrightarrow[\omega\to\lambda]{} G_0(\lambda)H_0(\lambda) - G_1(\lambda)H_1(\lambda) = 2, \\
&\xrightarrow[\text{(6.24), (6.25)}]{} G_0'(\lambda)H_0'(\lambda) + G_1'(\lambda)H_1'(\lambda) = 2.
\end{aligned} \tag{6.32}$$

(3.11) is rewritten as

$$\begin{aligned}
&\widehat{G}_0(\omega)(-j\widehat{\breve{H}}_0(\omega)) + (-j\widehat{G}_1(\omega))\widehat{\breve{H}}_1(\omega) = 0, \\
&\xrightarrow[\omega\to\lambda]{} G_0(\lambda)\breve{H}_0(\lambda) + G_1(\lambda)\breve{H}_1(\lambda) = 0, \\
&\xrightarrow[\text{Table 6.2, (6.24), (6.25)}]{} (-1)^{\frac{L_0-2}{2}} G_0'(\lambda)H_0'(2-\lambda) + (-1)^{\frac{L_1}{2}} G_1'(\lambda)H_1'(2-\lambda) = 0.
\end{aligned} \tag{6.33}$$

The filter lengths should be $\{L_0 = 4n_0, L_1 = 4n_1\}$, or $\{L_0 = 4n_0 + 2, L_1 = 4n_1 + 2\}$, from Theorem 1. Hence, the critically sampled spectral graph filter banks with even-length filters satisfy (3.35).

Case 3: M-channel oversampled spectral graph filter banks. Since the odd-indexed synthesis-side filters $G_{2k}(\omega)$ are type IV filters, (3.12) with $l = L-1$ and even M is modulated as

$$\begin{aligned} &\sum_{k=0}^{M/2-1} \widehat{G}_{2k}(\omega)\widehat{H}_{2k}(\omega) - (-j\widehat{G}_{2k+1}(\omega))(-j\widehat{H}_{2k+1}(\omega)) = 2, \\ &\xrightarrow[\omega\to\lambda]{} \sum_{k=0}^{M/2-1} G_{2k}(\lambda)H_{2k}(\lambda) - G_{2k+1}(\lambda)H_{2k}(\lambda) = 2. \\ &\xrightarrow[\text{(6.24), (6.25)}]{} \sum_{k=0}^{M/2-1} G'_{2k}(\lambda)H'_{2k}(\lambda) + G'_{2k+1}(\lambda)H'_{2k}(\lambda) = 2. \end{aligned} \tag{6.34}$$

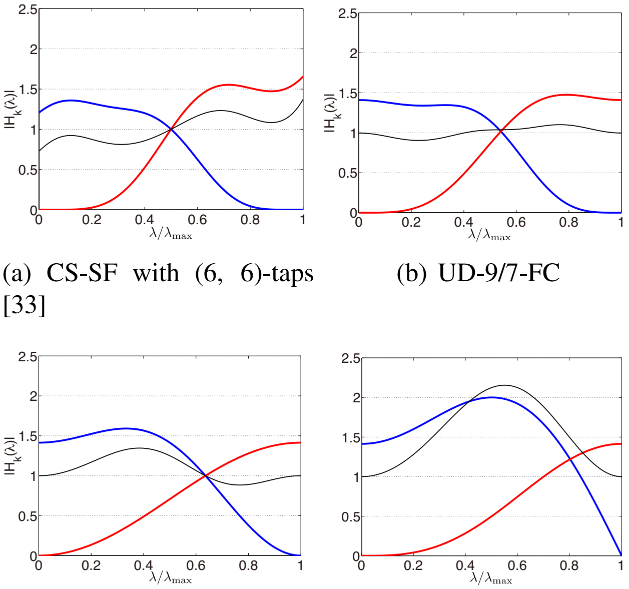
(3.13) is rewritten as

$$\begin{aligned} &\sum_{k=0}^{M/2-1} \widehat{G}_{2k}(\omega)(-j\widehat{\breve{H}}_{2k}(\omega)) + (-j\widehat{G}_{2k+1}(\omega))\widehat{\breve{H}}_{2k+1}(\omega) = 0, \\ &\xrightarrow[\omega\to\lambda]{} \sum_{k=0}^{M/2-1} G_{2k}(\lambda)\breve{H}_{2k}(\lambda) + G_{2k+1}(\lambda)\breve{H}_{2k+1}(\lambda) = 0, \\ &\xrightarrow[\text{Table 6.2, (6.24), (6.25)}]{} \\ &\quad (-1)^{\frac{L-2}{2}} \left(\sum_{k=0}^{M/2-1} -G'_{2k}(\lambda)H'_{2k}(2-\lambda) + G'_{2k+1}(\lambda)H'_{2k+1}(2-\lambda) \right) = 0. \end{aligned} \tag{6.35}$$

(6.34) and (6.35) indicate that the oversampled spectral graph filter bank with even M satisfies the perfect reconstruction conditions in (4.9) and (6.26). The case for odd M can be proven in the same way. □

Critically Sampled Spectral Graph Wavelets

Figure 6.3 shows examples of the CDF-based critically sampled spectral graph wavelet transforms. They were obtained by substituting the filter coefficients $h_k(m),\ m =$



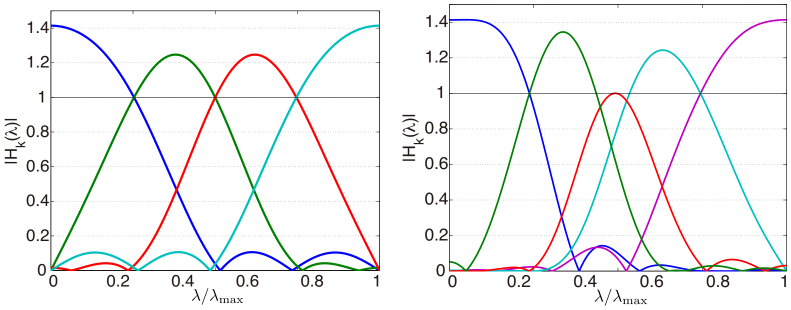
(a) CS-SF with (6, 6)-taps [33]

(b) UD-9/7-FC

(c) UD-5/3-FC

(d) UD-4/4-FC

Figure 6.3: Analysis filters of critically sampled spectral graph wavelet transforms. The black line indicates $\frac{1}{2}(H_0'^2(\lambda) + H_1'^2(\lambda))$. The filter characteristic of the regular CDF 9/7 DWT is in [75, Fig. 5.3–4].



(a) 4-channel OS-PU-FC

(b) 5-channel OS-PU-FC

Figure 6.4: Analysis filters of oversampled spectral graph filter banks. The black line indicates $\frac{1}{2}\sum_k H_k'^2(\lambda)$.

$0, 1, 2, \ldots, L_k - 1$ for $k = 0, 1$ of the CDF-DWTs [40] into (6.6) and (6.8) (for the CS-9/7-FC and the CS-5/3-FC) and (6.7) (for the CS-4/4-FC). The frequency response of a CS-SF with (6, 6)-taps [33] is also shown for comparison. It can be seen that the filter characteristics of the CS-9/7-FC, CS-5/3-FC and CS-4/4-FC are equivalent to those of CDF wavelets for regular signals.

Oversampled Spectral Graph Filter Banks

In the oversampled filter banks in regular signal processing, when M is even, the odd-indexed and even-indexed filters are type IV and type II, respectively. Whereas if M is odd, the odd-indexed and even-indexed filters are type III and type I, respectively. Figure 6.4 shows examples of the OS-PU-FC constructed from the oversampled linear phase paraunitary filter banks in [39] with $M = 4, 5$.

6.3 Experimental Results

6.3.1 Comparison of Filter Performance and Approximation Errors

Comparison of Uniform Filter Banks

Here, we compare the performances of uniform-band undecimated graph filter banks: UD-UH [31], UD-DCT-FC, and UD-LOT-FC.

Figure 6.5 plots the approximation error E between the ideal and approximated filter responses against the approximation order:

$$E = \frac{1}{M} \sum_{k=0}^{M-1} \left(\int_0^{\lambda_{\max}} |H_k(\lambda) - \widetilde{H}_k(\lambda)|^2 d\lambda \right). \tag{6.36}$$

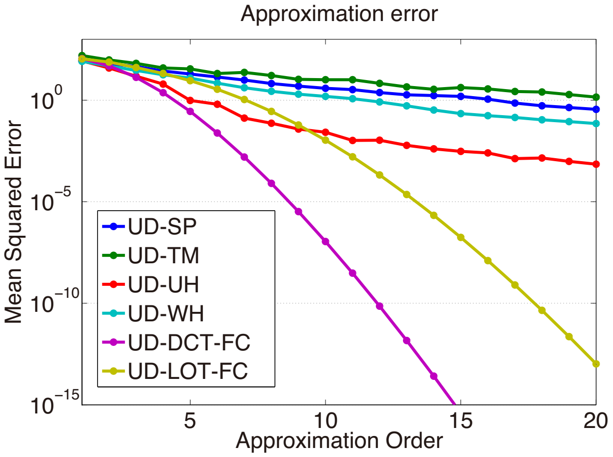
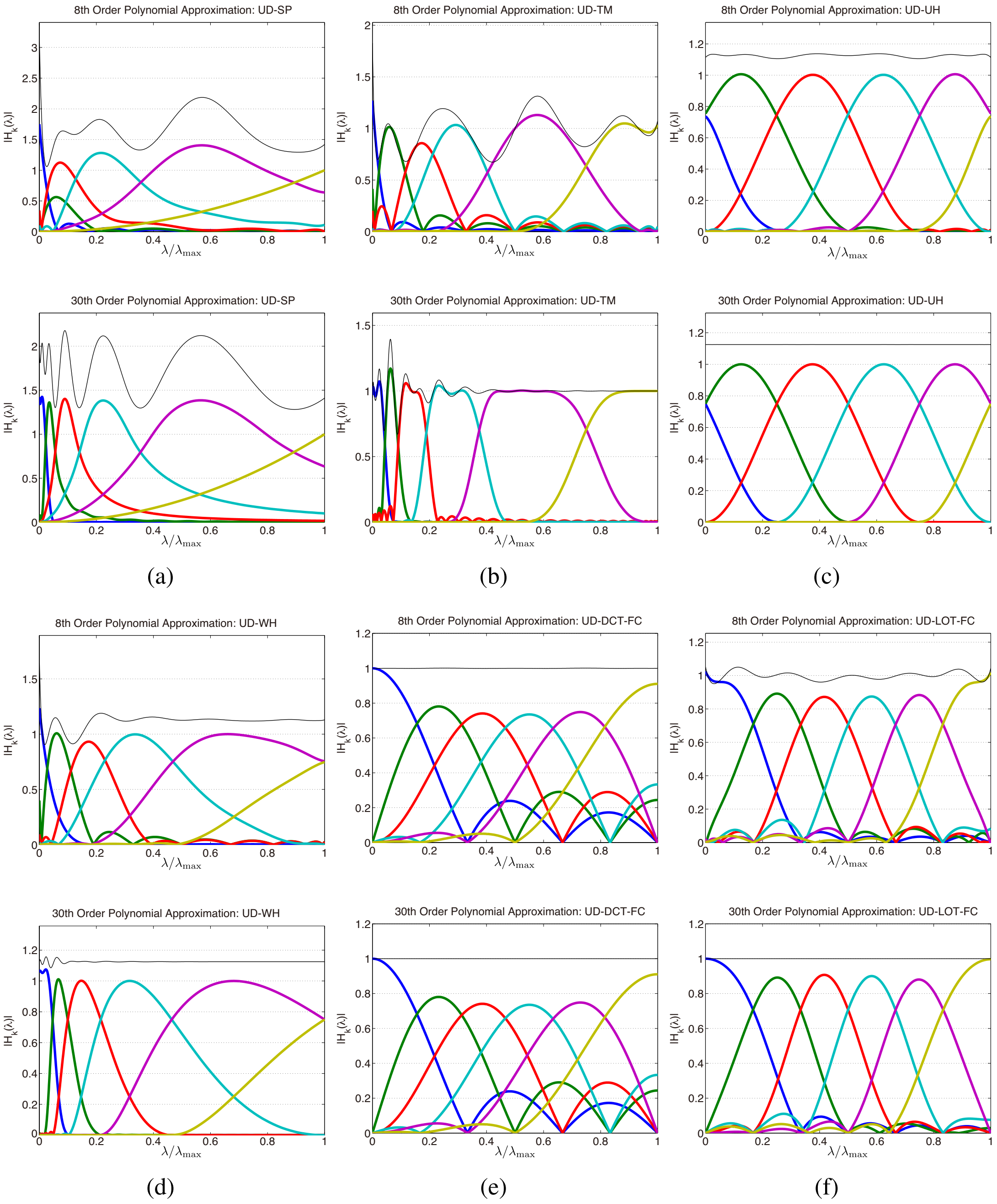


Figure 6.5: Mean squared approximation errors against approximation order. All filter banks were designed for $M = 6$. Note that the vertical axis is a logarithmic scale.

Table 6.3: Total Approximation Errors E of Undecimated Graph Filter Banks

All filters use the same approximation order		
p	59	34
UD-SP	9.70×10^{-4}	3.05×10^{-2}
UD-TM	3.00×10^{-5}	3.09×10^{-2}
UD-WH	1.75×10^{-5}	2.31×10^{-4}
Each filter uses the different approximation order		
p	{59, 51, 25, 12}	{34, 26, 14, 7}
UD-9/7-FC	4.88×10^{-10}	–
UD-5/3-FC	–	6.29×10^{-11}

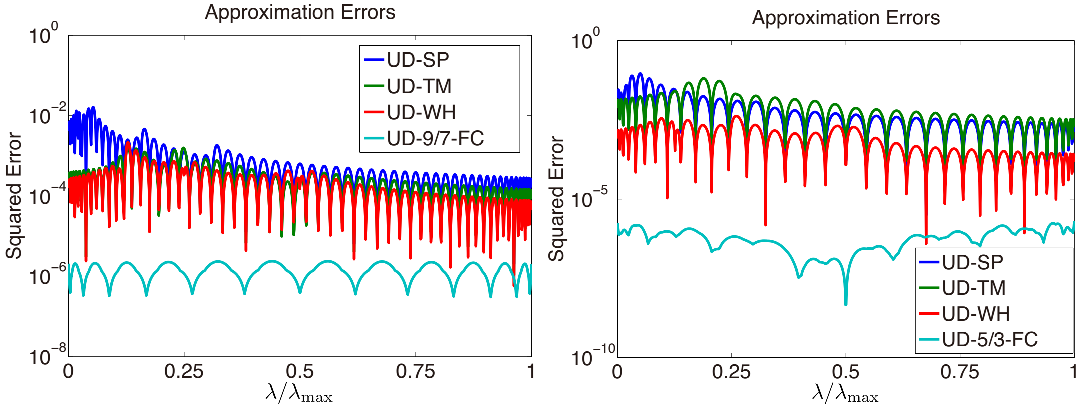
For reference, we also plot the approximation error of the UD-SP [13], UD-TM [16] and UD-WH [31]. As can be seen, the uniform-band graph filter banks, i.e., the UD-DCT-FC, the UD-LOT-FC, and the UD-UH, converged quickly as the approximation order increased. The approximation with $p = 8$–10 was close enough to reproduce the ideal filter response for UD-DCT-FC and UD-LOT-FC, whereas the other graph wavelets, except the one with the uniform band, required higher order approximations. Even when $p = 20$, there were still small differences from the ideal response. This was due to



(a) (b) (c)

(d) (e) (f)

Figure 6.6: Filter characteristics of graph filter banks with Chebyshev polynomial approximation for $M = 6$. Black lines indicate $\sum_k |\widetilde{H}_k(\lambda)|^2$. Top row: 8th order approximation. Bottom row: 30th order approximation. (a) UD-SP [13]. (b) UD-TM [16]. (c) UD-UH [31]. (d) UD-WH [31]. (e) UD-DCT-FC. (f) UD-LOT-FC.



(a)

(b)

Figure 6.7: Approximation errors. Note that the vertical axis is a logarithmic scale. (a) comparison with UD-9/7-FC. (b) comparison with UD-5/3-FC.

the different definitions of the ideal kernel. The existing transforms are *stitched* kernels[4] ; different functions are used for the passband, stopband and/or transition bands. In contrast, the UD-DCT-FC and UD-LOT-FC covered the entire graph frequency range with one series of sinusoidal waves, as shown in (6.6)–(6.9).

Figure 6.6 shows the approximated filter responses and $\sum_k |\widetilde{H}_k(\lambda)|^2$. Each filter was approximated with $p = 8$ or 30. As expected, the UD-DCT-FC had an ideal response even with $p = 8$, whereas the UD-LOT-FC and UD-UH produced small errors. Unfortunately, the other graph wavelets were not tight enough even after the 30th order approximation. The tight graph wavelets had large errors, especially in their scaling functions.

Approximation Errors of Octave-band Filter Banks

Let us examine the approximation errors of the UD-9/7-FC and UD-5/3-FC. Figs. 6.1(a) and 6.1(b) show their ideal characteristics. The approximation order of each filter is

[4] They are defined with piecewise functions. For the UD-SP kernel, the second derivative will be discontinuous, which is leading to the slow decay of error for the Chebyshev polynomial approximation.

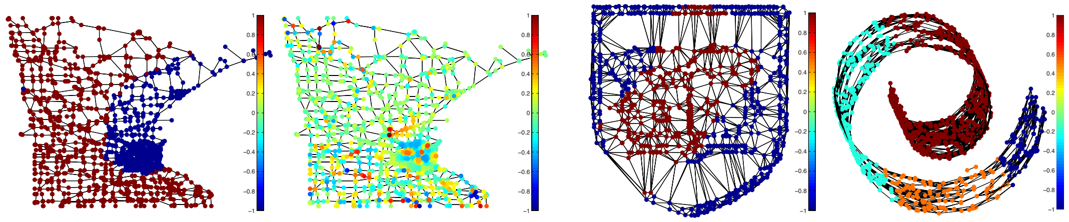
determined on the basis of Proposition 1 such that the approximation error does not exceed the pre-determined tolerable value $T = 1.00 \times 10^{-5}$. The approximation order is estimated as follows:

1. Initialize p to 1.
2. Calculate the estimated maximum error $E_{\text{max},p}$ in (6.10).
3. If $E_{\text{max},p} \leq T$, set p. Else, return to Step 2 with $p \leftarrow p + 1$.

This process costs less than directly calculating the error between the ideal filter and the pth approximated filters, since the direct calculation computes the Chebyshev coefficients (2.20) and approximation errors for all $i = 0, \dots, p$ and $k = 0, \dots, M - 1$ [13].

The filter lengths in the time domain and the estimated approximation orders of filters $H_k(\lambda)$, $k = 0, 1, 2, 3$ are $L = \{57,\ 57,\ 25,\ 9\}$ and $p = \{59,\ 51,\ 25,\ 12\}$ for UD-9/7-FC, and $L = \{29,\ 29,\ 13,\ 5\}$ and $p = \{34,\ 26,\ 14,\ 7\}$ for UD-5/3-FC. Figure 6.7 shows the approximation errors $\sum_{k=0}^{M-1} \left| H_k(\lambda) - \widetilde{H}_k(\lambda) \right|^2$. For comparison, Figs. 6.7(a) and 6.7(b) also show the errors of the UD-SP [13], UD-TM [16], and UD-WH [31] with $M = 4$ with approximation orders $p = 59$ and $p = 34$ for all filters, respectively. Table 6.3 shows the total approximation errors E in (6.36). The proposed filter banks have significantly smaller total approximation errors compared with the conventional approaches even when the maximum approximation orders are the same.

It is worth noting that, in our spectral graph filter banks, the lowpass filter needs a large approximation order, whereas the highpass one requires a relatively small order. This fact leads to the conclusion that the highpass graph filters in our method are more localized than the even-order approximations. This is a generally required characteristic for wavelet transforms.



(a) (b) (c) (d)

Figure 6.8: Original graph signals. (a) *Minnesota Traffic Graph* (Example 1). The signal was reproduced from the MATLAB code by Narang and Ortega [32]. (b) *Minnesota Traffic Graph* (Example 2). The signal is obtained by dividing nodes into three clusters and calculating the signal so that each cluster contains the graph spectrum $[0.06, 0.08], [0.3, 0.5], [3.2, 3.7]$, respectively, as in [31]. (c) *Yale Coat of Arms*. The signals of Figs. (c) and (d) were created by using SGWT toolbox [13]. (d) *Swiss Roll*.

6.3.2 Denoising

To make a numerical comparison of the undecimated transforms, we performed a graph signal denoising experiment. The input signal was corrupted by additive white Gaussian noise. For the proposed methods, we applied the UD-DCT-FC, the UD-9/7-FC, the spectrum-adapted UD-DCT-FC, and the spectrum-adapted UD-9/7-FC. These were compared with the UD-UH [31], the spectrum-adapted UD-UH [31], and the UD-SP [13]. We set $M = 4$ for all methods. All filters used the same filter length $p = 12$ for a pure comparison of filter banks. After decomposition of the corrupted input signal, the lowest frequency subband was kept, and the other high-frequency subbands were hard-thresholded with the threshold $T = 3\sigma$, where σ is the standard deviation of noise[5]. Figure 6.8 shows the tested signals, and Table 6.4 examines the denoising performance. Although UD-SP performed well for $\sigma = 1$ in a few cases, the proposed methods outperformed the existing methods for $\sigma \leq 1/2$.

[5] The noise variance of each subband was estimated by a Monte-Carlo method in order to estimate the energy of each subband. It repeatedly generates random white noise (graph) signals and averages their variance after the transformation [76].

Table 6.4: Denoised Results (Average of Ten Executions): SNR (dB). The Abbreviation SA Means Spectrum-Adapted.

	σ	1/32	1/16	1/8	1/4	1/2	1
Minnesota Traffic Graph (Example 1)	UD-UH	32.09	26.23	19.93	7.30	5.26	5.05
	SA UD-UH	33.13	27.43	21.33	7.75	5.45	5.29
	UD-SP	33.21	27.62	22.11	15.07	10.13	**9.01**
	UD-DCT-FC	34.43	28.21	21.50	15.74	10.75	5.34
	UD-9/7-FC	34.08	28.42	**22.77**	16.90	11.40	6.33
	SA UD-DCT-FC	**34.68**	**29.76**	22.50	16.44	12.30	7.46
	SA UD-9/7-FC	33.70	26.92	22.38	**17.85**	**12.94**	8.56
	noisy	30.09	24.05	18.08	12.05	6.01	0.12
Yale Coat of Arms	UD-UH	30.86	25.04	19.25	6.97	4.45	4.14
	SA UD-UH	31.32	25.41	19.93	7.75	4.77	4.39
	UD-SP	29.16	23.46	18.29	13.02	9.07	6.57
	UD-DCT-FC	**32.12**	**26.61**	20.32	13.99	10.27	5.67
	UD-9/7-FC	31.58	26.04	20.84	15.05	10.65	6.52
	SA UD-DCT-FC	30.99	25.50	18.99	13.62	10.61	6.70
	SA UD-9/7-FC	31.45	25.89	**21.14**	**15.20**	**10.74**	**7.10**
	noisy	30.04	23.97	18.24	11.99	6.02	0.05
Swiss Roll	UD-UH	29.57	21.51	15.97	7.03	5.03	4.73
	SA UD-UH	30.76	22.34	16.91	7.86	5.28	4.95
	UD-SP	29.32	24.46	19.57	15.33	11.79	**8.24**
	UD-DCT-FC	**32.96**	**26.36**	20.70	15.97	10.79	5.06
	UD-9/7-FC	31.41	26.19	**21.07**	**16.59**	**12.16**	7.26
	SA UD-DCT-FC	31.47	25.24	20.22	15.99	11.38	5.89
	SA UD-9/7-FC	30.12	25.31	20.60	16.14	12.04	7.27
	noisy	28.35	22.39	16.27	10.32	4.32	-1.69
	σ	1/128	1/64	1/32	1/16	1/8	1/4
Minnesota Traffic Graph (Example 2)	UD-UH	28.00	21.13	14.24	8.44	4.58	2.86
	SA UD-UH	27.91	21.94	15.50	9.26	4.78	2.82
	UD-SP	27.85	21.64	15.91	11.35	7.22	3.61
	UD-DCT-FC	28.19	21.58	15.14	10.16	6.20	2.11
	UD-9/7-FC	**30.72**	**24.90**	**19.09**	**13.33**	**8.24**	3.36
	SA UD-DCT-FC	27.92	21.89	15.46	10.66	7.18	3.69
	SA UD-9/7-FC	29.69	23.95	18.21	13.00	8.20	**4.15**
	noisy	28.37	22.41	16.38	10.29	4.34	-1.69

Table 6.5: Results on Non-linear Approximation: SNR(dB) for *Minnesota Traffic Graph* and PSNR(dB) for *Coins* Image

Fraction of highpass coeffs.		0.00	0.01	0.02	0.04	0.08	0.16
Minnesota Traffic Graph: Example 1	CS-QMF	17.79	21.51	24.58	27.83	31.36	33.90
	CS-SF	16.67	20.45	22.80	26.61	33.61	57.57
	CS-5/3-FC	**18.75**	**22.78**	**27.46**	**39.71**	**69.74**	**124.71**
	CS-9/7-FC	18.17	22.25	26.54	31.60	37.09	58.82
Minnesota Traffic Graph: Example 2	CS-QMF	4.28	**5.52**	**6.43**	**8.02**	10.49	14.87
	CS-SF	4.11	5.22	6.02	7.47	9.93	14.16
	CS-5/3-FC	**4.46**	5.32	6.27	7.66	10.37	**15.42**
	CS-9/7-FC	4.33	5.42	6.41	7.96	**10.56**	15.10
Coins	CDF 9/7 DWT	20.38	23.78	25.65	28.25	32.01	38.08
	CS-QMF	22.28	25.70	26.77	27.92	29.06	29.93
	CS-SF	18.73	26.33	28.34	30.89	34.34	39.39
	CS-5/3-FC	**23.76**	**27.91**	**29.67**	31.84	35.06	39.68
	CS-9/7-FC	22.61	27.79	29.59	**31.96**	**35.30**	**40.16**

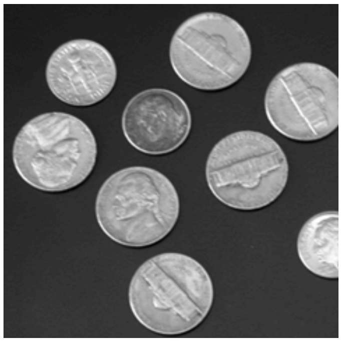


Figure 6.9: *Coins* image.

6.3.3 Non-linear Approximation

In the non-linear approximation, we compared the CS-9/7-FC and CS-5/3-FC with CDF 9/7 DWT for regular signals, CS-QMF [32], and CS-SF with (5,5)-taps [33]. The original signals of the *Minnesota Traffic Graph* (Examples 1 and 2) and the *Coins* image are shown in Figs. 6.8(a), 6.8(b) and 6.9. All graph-based transforms had a filter length of $p = 10$, for a fair comparison of filter performance, and used edge-aware image graphs for the *Coins* image [68]. Note that the CDF 9/7 for regular signals is not an edge-aware

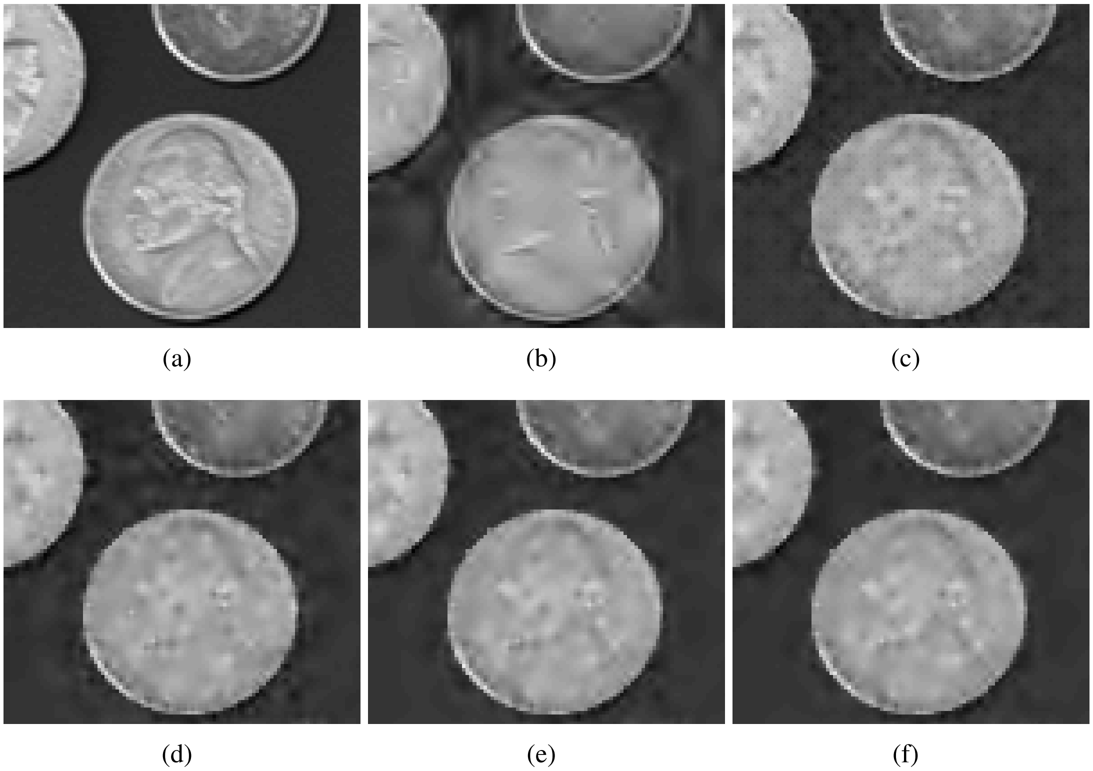


Figure 6.10: Zoomed in *Coins* images reconstructed from all lowpass coefficients and 3% of highpass coefficients. (a) original image. (b) CDF 9/7 DWT (27.16dB). (c) CS-QMF (27.92dB). (d) CS-SF (29.79dB). (e) CS-9/7-FC (30.88dB). (f) CS-5/3-FC (30.85dB).

transform. After a four-level (for the *Coins* image) or one-level decomposition[6] (for the *Minnesota Traffic Graph*), the input signal was reconstructed from all lowpass coefficients and some fraction of the highpass coefficients. Table 6.5 shows PSNR and SNR together with the fraction of highpass coefficients. The table shows that the proposed critically sampled spectral graph wavelet transforms outperformed the other methods on all signals. Since the CS-5/3-FCs can be exactly approximated at low order, it is very localized in the graph vertex domain. Therefore, if the original signal is piecewise-smooth,

[6] Since the edge-aware image graphs and the *Minnesota Traffic Graph* are four-colorable and three-colorable graphs, respectively, they can be decomposed into two bipartite subgraphs before applying the two-dimensional critically sampled spectral graph wavelet transforms [32, 61].

such as the signal of the *Minnesota Traffic Graph* (Example 1) in Fig. 6.8(a), it can be almost perfectly recovered from a small fraction of highpass coefficients. Although the CDF 9/7 DWT for regular signals and CS-9/7-FC have the same filter characteristics, the performance of the CS-9/7-FC is significantly better than that of the CDF 9/7 DWT for regular signals since the graph-based transform can decompose images while considering the image edges. Figure 6.10 shows the reconstructed *Coins* images. We can see that the CS-9/7-FC and CS-5/3-FC suppress ringing artifacts compared to the other methods.

6.4 Summary

We proposed a method of constructing wavelets and filter banks in the graph spectral domain. They are defined as a sum of sinusoidal waves. Therefore, they are very smooth and the upper bound of their approximation errors can be calculated in a closed form. We also described the method of constructing these spectral graph filter banks from the linear phase perfect reconstruction filter banks in regular signal processing. We showed examples of such M-channel undecimated and decimated perfect reconstruction spectral graph filter banks. The proposed undecimated spectral graph filter banks performed very well in the denoising experiments. The critically sampled spectral graph filter banks derived from the CDF-DWTs were also shown to outperform the existing critically sampled spectral graph filter banks in non-linear approximation experiments.

Chapter 7

Conclusion and Future Work

This dissertation addressed the design problems on spectral graph wavelets and filter banks. They are one of the most important fundamental tools in signal processing on graphs and useful for analyzing and processing the signals on graphs. The drawbacks of the conventional methods are 1) they have high redundancies or low flexibilities on their design, 2) the decimated transforms can be applied only to the bipartite graph, and 3) no transform exists that satisfies no DC leakage, orthogonal transform, defined by smooth function and applicable to any graphs.

To solve these problems, the following methods were proposed.

1. *M-channel spectral graph filter banks (Chapter 4).* The spectral graph filter banks with M filters and sampling operations were proposed. They are more flexible on the redundancies and filter design than the conventional critically sampled or undecimated filter banks. We showed the perfect reconstruction conditions, design methods of the filter sets based on spectral factorizations and design examples of the filter banks.

2. *Oversampled graph Laplacian matrix (Chapter 5).* The oversampled graph Laplacian matrix that can freely append the nodes and edges into the original graph has been proposed. The graph signals are also expanded simultaneously along with the oversampling of the graphs. We showed the oversampling method is effective when we use decimated spectral graph filter banks. It can make one bipartite graph that includes all edges in the original non-bipartite graph and it enables us to

apply the decimated transform into any graphs. We also clarified the relationships between the proposed method and the graph covering method.

3. *Filters converted from those in regular signal processing (Chapter 6).* We proposed the filters calculated from the linear phase FIR filters in regular signal processing. They are desired by smooth functions and their error caused by the polynomial approximation can be rigorously estimated since they are defined by the sum of sinusoidal waves. We can reuse any filters in time/spatial domain for graph signals since the spectral graph filters obtained by the proposed method inherit the characteristics of corresponding filters in regular signal processing. We showed the graph filters based on DCT, LOT, CDF wavelets, and oversampled filter banks for regular signals.

Each of them or their combination is applied to the non-linear approximation and denoising. They showed better results than the conventional wavelets for regular signals and spectral graph wavelets and filter banks for graph signals.

There are several questions that we would like to consider in future. The decimated transform proposed in Chapter 4 should use normalized graph Laplacian and undirected graphs. The decimated spectral filter banks with perfect recovery for directed graphs and another variation operators such as combinatorial graph Laplacian would be useful.

In Chapter 5, we considered the oversampling method for decimated transforms. To explore the capability of the oversampling method should be considered and theoretically clarified.

For filter design, we will consider the filters well behaved for directed graphs that have the different characteristics on eigen-pairs from that of undirected graphs. The directed graphs have complex valued eigenvalues and eigenvectors, and therefore, we cannot utilize the theories of the undirected graphs, straightforwardly, such as Chebyshev polynomial approximation. Our future work includes construction of the filtering methods for directed graphs with high performance and low computational complexity. It would be useful for utilizing the spectral graph wavelets and filter banks into real world problems by applying them to other applications such as anomaly detection and segmentation, and to find more effective implementations for large graphs.

Bibliography

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

[2] W. Wang and K. Ramchandran, "Random multiresolution representations for arbitrary sensor network graphs," in *Proc. ICASSP'06*, 2006, pp. IV–161–IV–164.

[3] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53–94, 2006.

[4] C. Zhang and D. Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 106–109, 2012.

[5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[6] ——, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, 2014.

[7] ——, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80–90, 2014.

[8] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal denoising on graphs via graph filtering," in *Proc. GlobalSIP*, 2014, pp. 872–876.

[9] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *DCOSS'11*, 2011, pp. 1–8.

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

[11] A. Agaskar and Y. M. Lu, “A spectral graph uncertainty principle,” *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4338–4356, 2013.

[12] F. R. K. Chung, *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, 1997.

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

[14] D. I. Shuman, B. Ricaud, and P. Vandergheynst, “Vertex-frequency analysis on graphs,” *submitted to Applied and Computational Harmonic Analysis*, 2013.

[15] G. Shen and A. Ortega, “Transform-based distributed data gathering,” *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3802–3815, 2010.

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

[17] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, “Efficient sensor position selection using graph signal sampling theory,” in *ICASSP’16*, 2016, pp. 6225–6229.

[18] M. Crovella and E. Kolaczyk, “Graph wavelets for spatial traffic analysis,” in *Proc. INFOCOM’03*, vol. 3, 2003, pp. 1848–1857.

[19] M. Gavish, B. Nadler, and R. R. Coifman, “Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning,” in *Proc. ICML’10*, 2010, pp. 367–374.

[20] S. Chen, F. Cerda, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovačević, “Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring,” *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2879–2893, 2014.

[21] D. Thanou, D. I. Shuman, and P. Frossard, “Learning parametric dictionaries for signals on graphs,” *IEEE Trans. Signal Process.*, vol. 62, no. 15, pp. 3849–3862, Aug. 2014.

[22] R. Rustamov and L. Guibas, “Wavelets on graphs via deep learning,” in *Proc. NIPS’13*, 2013, pp. 998–1006.

[23] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” *ArXiv e-prints: arXiv:1406.7842*, 2014. [Online]. Available: http://arxiv.org/abs/1406.7842

[24] A. Gadde, A. Anis, and A. Ortega, "Active semi-supervised learning using sampling theory for graph signals," in *Proc. KDD'14*, 2014, pp. 492–501.

[25] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, "Compressive spectral clustering," in *Proc. ICML'16*, 2016, pp. 1002–1011.

[26] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.

[27] A. Kheradmand and P. Milanfar, "A general framework for kernel similarity-based image denoising," in *Proc. GlobalSIP'13*, 2013, pp. 415–418.

[28] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *Proc. MMSP'13*, 2013.

[29] Y. Iizuka and Y. Tanaka, "Depth map denoising using collaborative graph wavelet shrinkage on connected image patches," in *Proc. ICIP'14*, 2014, pp. 828–832.

[30] G. M. Phillips, *Interpolation and Approximation by Polynomials*. Springer, 2013.

[31] D. I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vandergheynst, "Spectrum-adapted tight graph wavelet and vertex-frequency frames," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4223–4235, Aug. 2015.

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

[33] ——, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4673–4685, 2013. [Online]. Available: http://biron.usc.edu/wiki/index.php/Graph_Filterbanks

[34] N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints: arXiv:1408.5781*, Aug. 2014. [Online]. Available: http://arxiv.org/abs/1408.5781

[35] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, 2015.

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

[37] H. Q. Nguyen and M. N. Do, "Downsampling of signals on graphs via maximum spanning trees," *IEEE Trans. Signal Process.*, vol. 63, pp. 182–191, Jan. 2015.

[38] V. N. Ekambaram, G. Fanti, B. Ayazifar, and K. Ramchandran, “Critically-sampled perfect-reconstruction spline-wavelet filterbanks for graph signals,” in *Global-SIP’13*, Dec. 2013, pp. 475–478.

[39] L. Gan and K.-K. Ma, “Oversampled linear-phase perfect reconstruction filter-banks: theory, lattice structure and parameterization,” *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 744–759, 2003.

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

[41] J.-L. Starck, F. Murtagh, and J. M. Fadili, *Sparse image and signal processing: wavelets, curvelets, morphological diversity*. Cambridge University Press, 2010.

[42] K. R. Rao and P. Yip, *Discrete cosine transform: algorithms, advantages, applications*. CA: Academic Press, 1990.

[43] G. Strang, “The discrete cosine transform,” *SIAM Rev.*, vol. 41, no. 1, pp. 135–147, 1999.

[44] V. Britanak, P. C. Yip, and K. R. Rao, *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Academic Press, 2010.

[45] H. S. Malvar and D. H. Staelin, “The LOT: transform coding without blocking effects,” *IEEE Trans. Signal Process.*, vol. 37, no. 4, pp. 553–559, 1989.

[46] H. S. Malvar, *Signal processing with lapped transforms*. Norwood, MA: Artech House, 1992.

[47] Z. Cvetković and M. Vetterli, “Oversampled filter banks,” *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1245–1255, May 1998.

[48] P. L. Dragotti, J. Kovacevic, and V. K. Goyal, “Quantized oversampled filter banks with erasures,” in *Proc. IEEE Data Compression Conf.*, 2001, pp. 173–182.

[49] F. Labeau, L. Vandendorpe, and B. Macq, “Structures, factorizations, and design criteria for oversampled paraunitary filterbanks yielding linear-phase filters,” *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 3062–3071, 2000.

[50] T. Tanaka and Y. Yamashita, “The generalized lapped pseudo-biorthogonal transform: oversampled linear-phase perfect reconstruction filterbanks with lattice structures,” *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 434–446, 2004.

[51] Y. Tanaka, M. Ikehara, and T. Q. Nguyen, “Higher-order feasible building blocks for lattice structure of oversampled linear-phase perfect reconstruction filter banks,” *Signal Processing*, vol. 89, no. 9, pp. 1694–1703, 2009.

[52] J. M. F. Moura and M. G. S. Bruno, "DCT/DST and Gauss-Markov fields: conditions for equivalence," *IEEE Trans. Signal Process.*, vol. 46, no. 9, pp. 2571–2574, Sep. 1998.

[53] R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao, "The GenLOT: generalized linear-phase lapped orthogonal transform," *IEEE Trans. Signal Process.*, vol. 44, no. 3, pp. 497–507, 1996.

[54] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 6, pp. 1557–1571, 2003.

[55] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins Univ. Press, 1996.

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

[57] T. D. Tran and T. Q. Nguyen, "On M-channel linear phase FIR filter banks and application in image compression," *IEEE Trans. Signal Process.*, vol. 45, no. 9, pp. 2175–2187, 1997.

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

[59] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Process.*, vol. 40, no. 9, pp. 2207–2232, 1992.

[60] T. D. Tran, R. L. de Queiroz, and T. Q. Nguyen, "Linear phase perfect reconstruction filter bank: lattice structure, design, and application in image coding," *IEEE Trans. Signal Process.*, vol. 48, no. 1, pp. 133–147, 2000.

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

[62] R. A, F. Harary, and Z. Miller, "Bigraphs versus digraphs via matrices," *Journal of Graph Theory*, vol. 4, pp. 51–73, 1980.

[63] E. Sampathkumar, "On tensor product graphs," *Journal of the Australian Mathematical Society*, vol. Series A, pp. 268–273, 1975.

[64] A. L. Dulmage and N. S. Mendelsohn, "Coverings of bipartite graphs," *Canadian Journal of Mathematics*, pp. 517–534, 1958.

[65] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.

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

[67] M. N. Do and M. Vetterli, “Framing pyramids,” *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2329–2342, 2003.

[68] S. K. Narang, Y. H. Chao, and A. Ortega, “Graph-wavelet filterbanks for edge-aware image processing,” in *Proc. IEEE SSP Workshop.*, pp. 141–144, Aug. 2012.

[69] A. Sakiyama and Y. Tanaka, “Edge-aware image graph expansion methods for oversampled graph Laplacian matrix,” in *Proc. ICIP’14*, 2014, pp. 2958–2962.

[70] D. Spielman, “Spectral graph theory,” 2012. [Online]. Available: http://www.cs.yale.edu/homes/spielman/561/

[71] Y. Tanaka and A. Sakiyama, “M-channel oversampled perfect reconstruction filter banks for graph signals,” in *Proc. ICASSP’14*, 2014, pp. 2623–2627.

[72] ——, “M-channel oversampled graph filter banks,” *IEEE Trans. Signal Process.*, vol. 62, pp. 3578–3590, 2014.

[73] T. D. Tran, M. Ikehara, and T. Q. Nguyen, “Linear phase paraunitary filter bank with filters of different lengths and its application in image compression,” *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2730–2744, 1999.

[74] M. Vetterli, J. Kovaćevič, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge University Press, 2014.

[75] J. W. Woods, *Multidimensional Signal, Image, and Video Processing and Coding*. Academic Press, 2011.

[76] D. D.-Y. Po and M. N. Do, “Directional multiscale modeling of images using the contourlet transform,” *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1610–1620, 2006.