Oversampled Graph Laplacian Matrix for Graph Filter Banks

Akie Sakiyama, *Student Member, IEEE*, and Yuichi Tanaka, *Member, IEEE*

***Abstract*—We describe a method of oversampling signals defined on a weighted graph by using an oversampled graph Laplacian matrix. The conventional method of using critically sampled graph filter banks has to decompose the original graph into bipartite subgraphs, and a transform has to be performed on each subgraph because of the spectral folding phenomenon caused by downsampling of graph signals. Therefore, the conventional method cannot always utilize all edges of the original graph in a single stage transformation. Our method is based on oversampling of the underlying graph itself, and it can append nodes and edges to the graph somewhat arbitrarily. We use this approach to make one oversampled bipartite graph that includes all edges of the original non-bipartite graph. We apply the oversampled graph with the critically sampled graph filter bank or the oversampled one for decomposing graph signals and show the performances on some experiments.**

***Index Terms*—Graph filter banks, graph oversampling, graph signal processing, graph wavelets, multiresolution.**

I. Introduction

GRAPHS are data structures that can represent complex relationships among data and can be used in many fields of engineering and science. A graph consists of nodes and edges, and each edge is usually assigned a weight determined by the similarity and connectivity of the nodes. A recent development is that of graph signal processing, in which a sample is placed on each node of a graph and the processing takes into account the structure of the samples [1]–[14]. Whereas signals of regular signal processing have very simple structures, those of graph signal processing are allowed to have complex irregular structures.

Multiresolution analysis is an efficient way of analyzing, processing, and compressing signals [15]. Wavelet transforms for graph signals can be used to make multiresolution analysis [5], [13], [16], [17]. In particular, the use of two-channel critically sampled wavelet filter banks on graphs has been proposed [2], [3]. An important topic in graph signal processing is downsampling and upsampling. Similar to the aliasing effect of regular signal processing, the spectrum folding phenomenon affects downsampling of graph signals. In order to deal with this challenge, studies on critically sampled filter banks have focused on bipartite graphs and determined the conditions under which perfect reconstruction is possible.

The graph filter banks with downsampling operations, such as critically sampled graph filter banks and oversampled ones [16], [17], can only be applied to bipartite graphs. For arbitrary non-bipartite graphs, the original graph must be decomposed into bipartite subgraphs. Each subgraph has all of the nodes and some of the edges of the original graph, and their union yields back the original graph. The filter banks are applied to each of these subgraphs, and this leads to a multidimensional transform. The subgraph only has some of edges of the original graph, and the single stage transform usually does not involve many of the original edges.

The concept of graph oversampling using an oversampled graph Laplacian matrix was proposed in our recent paper [16]. By using such a matrix, we can append nodes and edges to the graph somewhat arbitrarily. Furthermore, graph signals can also be freely chosen. However, in that paper, we only presented a simple example of oversampling for a bipartite graph. An inappropriate choice of the oversampled graph Laplacian matrix and graph signals will degrade performance relative to that of the original graph. Additionally, theoretical connections to graph theory were not studied.

In this paper, we present an effective method of applying graph oversampling to non-bipartite graphs that avoids inappropriate choices. In particular, we describe a method that converts an arbitrary K-colorable graph into one bipartite graph containing all edges of the original graph and compare the oversampled bipartite graph with critically sampled ones. Moreover, we describe an image processing with graph oversampling. This formulation enables us to decompose images in an edge-preserving manner. The redundant multiresolution transform can be implemented with critically sampled graph filter banks or oversampled ones on the oversampled graph. We validated the performance of graph oversampling through an experiment on non-linear approximation of images and denoising of graph signals.

The rest of this paper is organized as follows. In Section II, we describe the notation used in this paper, the two-channel critically sampled wavelet filter bank on graphs [2], [3], and the oversampled one [16], [17]. Section III introduces the method of oversampling graph Laplacian matrices and input signals. Section IV 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

Manuscript received April 12, 2014; revised July 29, 2014; accepted October 09, 2014. Date of publication October 28, 2014; date of current version November 12, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiao-Ping Zhang. This work was supported in part by MEXT Tenure-Track Promotion Program and MIC SCOPE Grant Number 142103007. MATLAB code examples are available at http://tanaka.msp-lab.org/software.

The authors are with the Graduate School of BASE, Tokyo University of Agriculture and Technology, Koganei, Tokyo, 184-8588 Japan (e-mail: sakiyama@msp-lab.org; ytnk@cc.tuat.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2365761

proposed oversampled graph and the bipartite double cover of a graph in graph theory [18]–[20]. Section V shows examples of graph oversampling for images and a ring graph, and compares oversampled bipartite graphs with the critically sampled ones. Section VI describes experimental results on graph signal decomposition. Section VII concludes the paper.

II. Preliminaries

A. Graph Signals

A graph $\mathcal{G}$ is represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_0, v_1, \ldots, v_{N-1}\}$ and $\mathcal{E}$ denote sets of nodes and edges, respectively. The graph signal is defined as $\boldsymbol{f} \in \mathbb{R}^N$. 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 (m, n)-th element of the adjacency matrix $\mathbf{A}$ is defined as follows:

$$\mathbf{A}(m,n) = \begin{cases} w_{mn} & \text{if nodes } m \text{ and } n \text{ are connected,} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

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 $d_{mm} = \sum_n a_{mn}$. The unnormalized 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], and we will use $\boldsymbol{\mathcal{L}}$ in this paper. The eigenvalues of $\boldsymbol{\mathcal{L}}$ are λ_i and ordered as: $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \ldots < \lambda_{N-1} \leq 2$ without loss of generality.[1] The eigenvector $\boldsymbol{u}_{\lambda_i}$ corresponds to λ_i and satisfies $\boldsymbol{\mathcal{L}}\boldsymbol{u}_{\lambda_i} = \lambda_i \boldsymbol{u}_{\lambda_i}$. The eigenvectors $\mathbf{U} = [\boldsymbol{u}_{\lambda_0}, \ldots, \boldsymbol{u}_{\lambda_{N-1}}]$ satisfy

$$\mathbf{U}\mathbf{U}^T = \mathbf{I}_N, \tag{2}$$

where $\cdot^T$ is the transpose of a matrix or a vector and $\mathbf{I}_N$ is an $N \times N$ identity matrix. The entire spectrum of $\mathcal{G}$ is defined by $\sigma(\mathcal{G}) := \{\lambda_0, \ldots, \lambda_{N-1}\}$. The graph Fourier transform is defined as follows [5], [21]:

$$\bar{f}(\lambda_i) = \langle \boldsymbol{u}_{\lambda_i}, \boldsymbol{f} \rangle = \sum_{n=0}^{N-1} u^*_{\lambda_i}(n) f(n). \tag{3}$$

where $\cdot^*$ is the complex conjugate. The projection matrix for the eigenspace V_{λ_i} is

$$\mathbf{P}_{\lambda_i} = \sum_{\lambda = \lambda_i} \boldsymbol{u}_\lambda \boldsymbol{u}_\lambda^T. \tag{4}$$

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

$$\mathbf{P}_{\lambda_i}\mathbf{P}_{\lambda_j} = \delta(\lambda_i - \lambda_j)\mathbf{P}_{\lambda_i}, \tag{5}$$

where $\delta(\lambda)$ is the Kronecker delta function. Let $h(\lambda_i)$ be the spectral kernel of filter $\mathbf{H}$. The spectral domain filter can be written as

$$\mathbf{H} = \sum_{\lambda_i \in \sigma(\mathcal{G})} h(\lambda_i)\mathbf{P}_{\lambda_i}. \tag{6}$$

The spectral domain filtering of graph signals can be simply denoted as $\mathbf{H}\boldsymbol{f}$.

B. Two-Channel Graph Wavelet Filter Banks

A bipartite graph, whose nodes can be decomposed into two disjoint sets L and H such that every link connects a node in L to one in H, can be represented as $\mathcal{G} = \{H, L, \mathcal{E}\}$. The downsampling function β_H of a bipartite graph is defined as

$$\beta_H(m) = \begin{cases} +1 & \text{if } m \in H, \\ -1 & \text{if } m \in L. \end{cases} \tag{7}$$

The diagonal downsampling matrix is $\mathbf{J}_H = \operatorname{diag}\{\beta_H(n)\}$ and satisfies $\mathbf{J} = \mathbf{J}_H = -\mathbf{J}_L$. The downsampling-then-upsampling operation is defined as [3]

$$\begin{aligned} \mathbf{D}_{du,L} &= \frac{1}{2}(\mathbf{I}_N + \mathbf{J}_L), \\ \mathbf{D}_{du,H} &= \frac{1}{2}(\mathbf{I}_N + \mathbf{J}_H). \end{aligned} \tag{8}$$

where $\mathbf{I}_N$ is an $N \times N$ identity matrix.

$\mathbf{J}$ and $\mathbf{P}_{\lambda_i}$ are related as follows [3] (spectral folding phenomenon):

$$\mathbf{J}\mathbf{P}_{\lambda_i} = \mathbf{P}_{2-\lambda_i}\mathbf{J}. \tag{9}$$

The nodes in H store the output of the highpass channel, whereas the nodes in L store the output of the lowpass channel. Critically sampled graph filter banks decompose $\boldsymbol{f}$ into $|L|$ lowpass coefficients and $|H|$ highpass coefficients, where $|L| + |H| = N$, as illustrated in Fig. 1. The overall transfer function of graph-QMF [3] and graphBior [2] can be written as

$$\begin{aligned} \mathbf{T} &= \frac{1}{2}\mathbf{G}_0(\mathbf{I} - \mathbf{J})\mathbf{H}_0 + \frac{1}{2}\mathbf{G}_1(\mathbf{I} + \mathbf{J})\mathbf{H}_1 \\ &= \frac{1}{2}(\mathbf{G}_0\mathbf{H}_0 + \mathbf{G}_1\mathbf{H}_1) + \frac{1}{2}(\mathbf{G}_1\mathbf{J}\mathbf{H}_1 - \mathbf{G}_0\mathbf{J}\mathbf{H}_0). \end{aligned} \tag{10}$$

The spectral folding term $\mathbf{G}_1\mathbf{J}\mathbf{H}_1 - \mathbf{G}_0\mathbf{J}\mathbf{H}_0$, arising from downsampling and upsampling, must be zero. In addition, $\mathbf{T} = \mathbf{I}_N$ should be satisfied for perfect reconstruction. Hence, the perfect reconstruction condition of graph-QMF [3] and graphBior [2] can be expressed as

$$\begin{aligned} g_0(\lambda)h_0(\lambda) + g_1(\lambda)h_1(\lambda) &= 2, \\ g_0(\lambda)h_0(2-\lambda) - g_1(\lambda)h_1(2-\lambda) &= 0. \end{aligned} \tag{11}$$

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 [22].

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

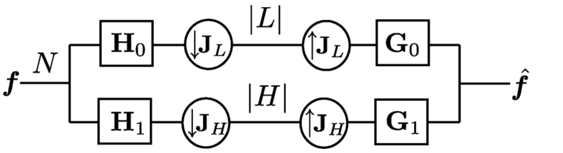


Fig. 1. Critically sampled two-channel graph filter bank.

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 [3], [23] 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.

C. M-Channel Oversampled Graph Filter Banks

The authors proposed M-channel oversampled graph filter banks [16], [17], where M is even and $M/2$ filters keep $|L|$ signals and other ones keep $|H|$ signals. Similar to the critically sampled case, the perfect reconstruction condition of the M-channel oversampled graph filter bank can be represented as

$$\sum_{k=0}^{M-1} g_k(\lambda)h_k(\lambda) = 2, \tag{12}$$

$$\sum_{k=0}^{M/2-1} g_k(\lambda)h_k(2-\lambda) - g_{k+M/2}(\lambda)h_{k+M/2}(2-\lambda) = 0 \tag{13}$$

for any λ. The latter equation is satisfied by choosing the filters $g_k(\lambda) = h_{k+M/2}(2-\lambda)$ and $g_{k+M/2}(\lambda) = h_k(2-\lambda)$. Furthermore, the product filters are defined as $p_k(\lambda) = g_k(\lambda)h_k(\lambda)$; accordingly, (12) can be written as

$$\sum_{k=0}^{M/2-1} p_k(\lambda) + p_k(2-\lambda) = 2. \tag{14}$$

In order to obtain the product filters satisfying this condition, the critically sampled product filter is factorized into lowpass and bandpass filters. Unlike critically sampled ones, M-channel oversampled graph filter banks can have arbitrary filters and still satisfy the perfect reconstruction condition.

III. Oversampled Graph Laplacian Matrix

In this section, we describe the structure of oversampled GLMs and oversampled graph signals [16]. Fig. 2 shows an example of the transform using graph oversampling with an M-channel oversampled graph filter bank. Before applying the graph filter bank, the original bipartite graph $\mathcal{G} = \{L, H, \mathcal{E}\}$ is expanded into an oversampled bipartite graph $\widetilde{\mathcal{G}} = \{\widetilde{L}, \widetilde{H}, \widetilde{\mathcal{E}}\}$ where $\widetilde{L}$ and $\widetilde{H}$ include L and 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 $\mathbf{L}_0$ 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

$$\widetilde{\boldsymbol{\mathcal{L}}} = \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{L}}\widetilde{\mathbf{D}}^{-1/2} \tag{15}$$

where

$$\widetilde{\mathbf{L}} = \widetilde{\mathbf{D}} - \widetilde{\mathbf{A}} \tag{16}$$

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_{01} \\ \mathbf{A}_{01}^T & \mathbf{0}_{N_1-N_0} \end{bmatrix}, \tag{17}$$

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. 2 are defined as

$$\mathbf{H}_k = \sum_{\lambda_i \in \sigma(\widetilde{\mathcal{G}})} h_k(\lambda_i)\widetilde{\mathbf{P}}_{\lambda_i},$$

$$\mathbf{G}_k = \sum_{\lambda_i \in \sigma(\widetilde{\mathcal{G}})} g_k(\lambda_i)\widetilde{\mathbf{P}}_{\lambda_i}, \tag{18}$$

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

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

$$\widetilde{\mathbf{J}} = \begin{bmatrix} \mathbf{J}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_1 \end{bmatrix}, \tag{19}$$

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

$$\widetilde{\boldsymbol{f}} = \begin{bmatrix} \boldsymbol{f}_0 \\ \boldsymbol{f}_1 \end{bmatrix}, \tag{20}$$

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 $\widehat{\boldsymbol{f}}'$.

IV. Effective Graph Expansion Methods

As described in Section III, 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.

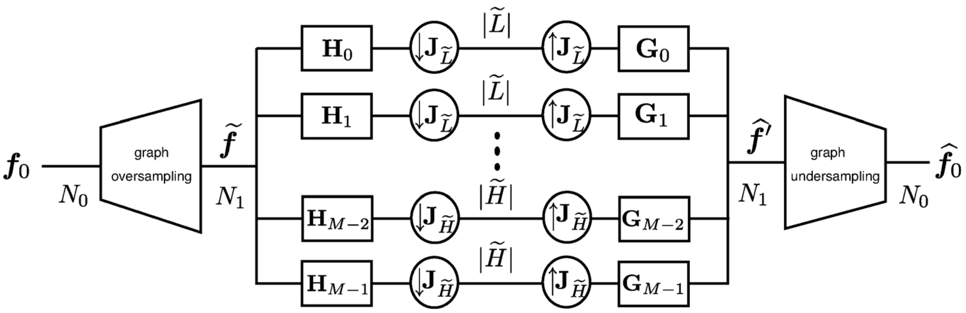

$\boldsymbol{f}_0$
N_0
graph oversampling
$\widetilde{\boldsymbol{f}}$
N_1
$\mathbf{H}_0$
$\mathbf{H}_1$
$\mathbf{H}_{M-2}$
$\mathbf{H}_{M-1}$
$\downarrow\mathbf{J}_{\widetilde{L}}$
$\downarrow\mathbf{J}_{\widetilde{H}}$
$|\widetilde{L}|$
$|\widetilde{H}|$
$\uparrow\mathbf{J}_{\widetilde{L}}$
$\uparrow\mathbf{J}_{\widetilde{H}}$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

Fig. 2. Graph oversampling followed by M-channel oversampled graph filter bank.

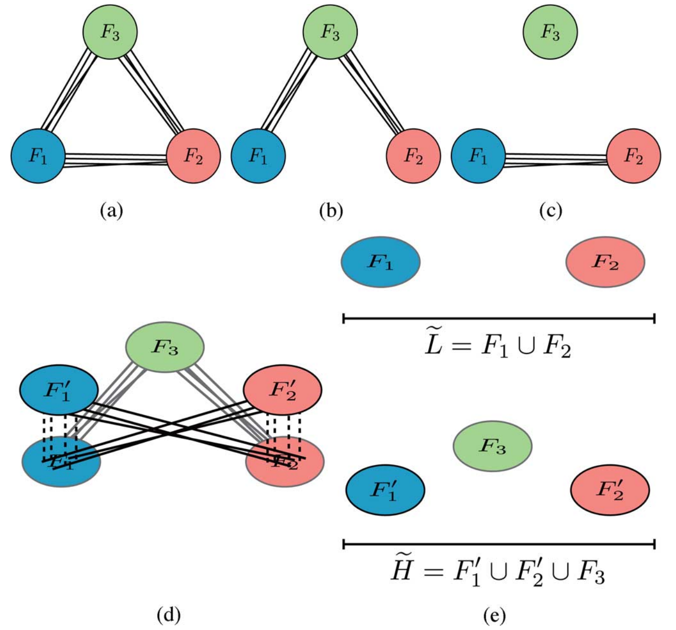

F_3
F_1
F_2
(a)
(b)
(c)
F_1'
F_2'
$\widetilde{L} = F_1 \cup F_2$
$\widetilde{H} = F_1' \cup F_2' \cup F_3$
(d)
(e)

Fig. 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 $\widetilde{L}$ and $\widetilde{H}$ of the oversampled bipartite graph.

A. 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. 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. 3(b)), and $\mathcal{B}_2$ that contains edges between F_1 and F_2 (Fig. 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 graph that contains all edges and nodes in the original graph (Fig. 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 $\widetilde{L} = F_1 \cup F_2$ and $\widetilde{H} = F_1' \cup F_2' \cup F_3$, as shown in Fig. 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

$$\begin{aligned}\rho &= \frac{M\left\{(|F_1| + |F_2|) + (|F_1'| + |F_2'| + |F_3|)\right\}}{2N} \\ &= \frac{M\left(N + |F_1| + |F_2| - |I|\right)}{2N}\end{aligned} \tag{21}$$

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

B. 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 $F_1, F_2, \ldots, F_K$. Fig. 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* $\mathcal{G}_b = \{L_b, H_b, \mathcal{E}_b\}$ is made from the original graph. $L_b = \{F_1, F_2, \ldots, F_l\}$ and $H_b = \{F_{l+1}, F_{l+2}, \ldots, F_K\}$, 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 L_b and H_b (Figs. 4(b) and 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 $\overline{\mathcal{G}}(L_b)$ and a $(K - l)$-colorable graph $\overline{\mathcal{G}}(H_b)$ (Figs. 4(c) and 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 $\widetilde{H}$, it can be connected freely with the nodes in $\{F_2, \ldots, F_l\}$ since they belong to $\widetilde{L}$. The edges between F_1' and $\{F_2, \ldots, F_l\}$ are appended in accordance with the edge information of $\overline{\mathcal{G}}(L_b)$. 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 $\overline{\mathcal{G}}(L_b)$.

6) Similar operations to Steps 3 to 5 can also be applied to the sets in H_b. As a result, the sets $F'_{l+1}, \dots, F'_K$ and the edges in $\overline{\mathcal{G}}(H_b)$ are appended to the foundation bipartite graph.

Consequently, the sets $F'_1, \dots, F'_K$ 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 $\widetilde{\mathcal{G}} = \{\widetilde{L}, \widetilde{H}, \widetilde{\mathcal{E}}\}$ containing all edges of the original graph is generated as shown in Figs. 4(d) and 4(g), where $\widetilde{L}$ and $\widetilde{H}$ respectively include L_b and H_b. Note that $\widetilde{L}$ and $\widetilde{H}$ of the oversampled graph become

$$\widetilde{L} = F_1 \cup \dots F_l \cup F'_{l+1} \cup \dots F'_K, \tag{22}$$

$$\widetilde{H} = F'_1 \cup \dots F'_l \cup F_{l+1} \cup \dots F_K. \tag{23}$$

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

$$|\widetilde{L}| = \sum_{i=1}^{l} |F_i| + \sum_{i=l+1}^{K} |F'_i| = N - |I_{H_b}|, \tag{24}$$

$$|\widetilde{H}| = \sum_{i=l+1}^{K} |F_i| + \sum_{i=1}^{l} |F'_i| = N - |I_{L_b}|, \tag{25}$$

where $|I_{L_b}|$ and $|I_{H_b}|$ are the number of isolated nodes in $\overline{\mathcal{G}}(L_b)$ and $\overline{\mathcal{G}}(H_b)$, respectively, and satisfy

$$\sum_{i=1}^{l} |F'_i| = \sum_{i=1}^{l} |F_i| - |I_{L_b}|, \tag{26}$$

$$\sum_{i=l+1}^{K} |F'_i| = \sum_{i=l+1}^{K} |F_i| - |I_{H_b}|. \tag{27}$$

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

$$\rho = \frac{M\left((N - |I_{L_b}|) + (N - |I_{H_b}|)\right)}{2N} = M - \frac{M\left(|I_{L_b}| + |I_{H_b}|\right)}{2N}. \tag{28}$$

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 $l = 1$, L_b is equal to F_1 and $\overline{\mathcal{G}}(L_b)$ has no edges as shown in Figs. 4(e) and 4(f). Therefore, we do not need to append nodes just above L_b, and the oversampled bipartite graph becomes

$$\widetilde{L} = F_1 \cup F'_2 \cup \dots F'_K, \tag{29}$$

$$\widetilde{H} = F_2 \cup \dots F_K. \tag{30}$$

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

$$\widetilde{L} = F_1 \cup \dots F_{K-1}, \tag{31}$$

$$\widetilde{H} = F'_1 \cup \dots F'_{K-1} \cup F_K. \tag{32}$$

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

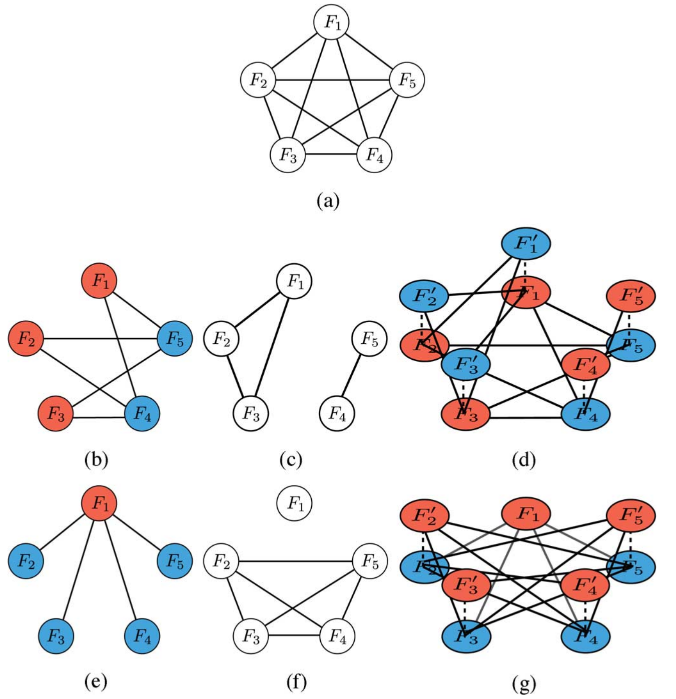


Fig. 4. Examples of oversampled bipartite graphs for a five-colorable graph. The circles filled with red and blue represent sets $\widetilde{L}$ and $\widetilde{H}$ sets, respectively. (a) Original graph. (b) Foundation bipartite graph with $l = 3$. (c) $\overline{\mathcal{G}}$ with $l = 3$. (d) Oversampled bipartite graph with $l = 3$. The dashed lines indicate the vertical edges. (e) Foundation bipartite graph with $l = 1$. (f) $\overline{\mathcal{G}}$ with $l = 1$. (g) Oversampled bipartite graph with $l = 1$.

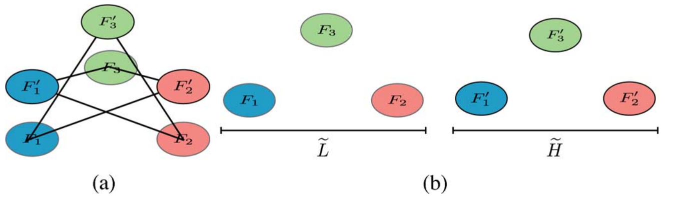


Fig. 5. (a) Bipartite double cover of a three-colorable graph. (b) Its set of $\widetilde{L}$ and $\widetilde{H}$.

C. 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 [18]–[20]. $\widetilde{\mathcal{G}}_{BDC}$ has $2N$ nodes and $2|\mathcal{E}|$ edges. An example of the bipartite double cover of a three-colorable graph (Fig. 3(a)) is shown in Fig. 5. The set of foundation nodes $\{F_1, \dots, F_K\}$ is contained in $\widetilde{L}$ and the set of additional nodes $\{F'_1, \dots, F'_K\}$ is in $\widetilde{H}$. 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

$$\widetilde{\mathbf{A}}_{BDC} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \tag{33}$$

and its normalized GLM is

$$\widetilde{\boldsymbol{\mathcal{L}}}_{BDC} = \begin{bmatrix} \mathbf{I} & -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \\ -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} & \mathbf{I} \end{bmatrix}. \tag{34}$$

If $\boldsymbol{u}_{\lambda_i}$ is an eigenvector of $\boldsymbol{\mathcal{L}}$ with the eigenvalue λ_i, one can immediately see that $\widetilde{\boldsymbol{u}}_{\widetilde{\lambda}_i} = \frac{1}{\sqrt{2}}[\boldsymbol{u}_{\lambda_i}^T\ \boldsymbol{u}_{\lambda_i}^T]^T$ and $\widetilde{\boldsymbol{u}}_{\widetilde{\lambda}_i} = \frac{1}{\sqrt{2}}[\boldsymbol{u}_{\lambda_i}^T\ -\boldsymbol{u}_{\lambda_i}^T]^T$ 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

$$\widetilde{\boldsymbol{u}}_{\lambda_i}^T\widetilde{\boldsymbol{f}} = \frac{1}{\sqrt{2}}\begin{bmatrix}\boldsymbol{u}_{\lambda_i}^T & \boldsymbol{u}_{\lambda_i}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{f}_0\\ \boldsymbol{f}_0\end{bmatrix} = \sqrt{2}\boldsymbol{u}_{\lambda_i}^T\boldsymbol{f}_0 = \sqrt{2}\overline{f}_0(\lambda_i), \quad (35)$$

and that associated with $2-\lambda_i$ is

$$\widetilde{\boldsymbol{u}}_{2-\lambda_i}^T\widetilde{\boldsymbol{f}} = \frac{1}{\sqrt{2}}\begin{bmatrix}\boldsymbol{u}_{\lambda_i}^T & -\boldsymbol{u}_{\lambda_i}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{f}_0\\ \boldsymbol{f}_0\end{bmatrix} = 0. \quad (36)$$

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

$$\widetilde{\mathbf{A}} = \begin{bmatrix}\mathbf{A}_f & \mathbf{A}_r\\ \mathbf{A}_r & \mathbf{0}\end{bmatrix} \quad (37)$$

and

$$\widetilde{\mathbf{D}} = \begin{bmatrix}\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_r\end{bmatrix}, \quad (38)$$

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

$$\widetilde{\boldsymbol{\mathcal{L}}} = \begin{bmatrix}\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_f\mathbf{D}^{-\frac{1}{2}} & -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}}\\ -\mathbf{D}_r^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}^{-\frac{1}{2}} & \mathbf{I}\end{bmatrix}. \quad (39)$$

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

$$\begin{aligned}\widetilde{\boldsymbol{\mathcal{L}}}\begin{bmatrix}\boldsymbol{u}_{\lambda_i}\\ \boldsymbol{u}_{\lambda_i}\end{bmatrix} &= \begin{bmatrix}\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_f\mathbf{D}^{-\frac{1}{2}} & -\mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}}\\ -\mathbf{D}_r^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}^{-\frac{1}{2}} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_{\lambda_i}\\ \boldsymbol{u}_{\lambda_i}\end{bmatrix}\\ &= \lambda_i\begin{bmatrix}\boldsymbol{u}_{\lambda_i}\\ \boldsymbol{u}_{\lambda_i}\end{bmatrix}. \quad (40)\end{aligned}$$

The constraint can be simplified as

$$\left(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_f\mathbf{D}^{-\frac{1}{2}} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}}\right)\boldsymbol{u}_{\lambda_i} = \lambda_i\boldsymbol{u}_{\lambda_i}. \quad (41)$$

On the other hand, the original GLM satisfies

$$\begin{aligned}\lambda_i\boldsymbol{u}_{\lambda_i} &= \boldsymbol{\mathcal{L}}\boldsymbol{u}_{\lambda_i}\\ &= \left(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)\boldsymbol{u}_{\lambda_i}\\ &= \left(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}_f + \mathbf{A}_r)\mathbf{D}^{-\frac{1}{2}}\right)\boldsymbol{u}_{\lambda_i}. \quad (42)\end{aligned}$$

Comparing (41) and (42), $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}^{-\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}_r\mathbf{D}_r^{-\frac{1}{2}}$ has to be satisfied. As a result, $\widetilde{\boldsymbol{\mathcal{L}}}$ has the eigenvector $\widetilde{\boldsymbol{u}}_{\lambda_i} = \frac{1}{\sqrt{2}}[\boldsymbol{u}_{\lambda_i}^T\ \boldsymbol{u}_{\lambda_i}^T]^T$ 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 $2N$.

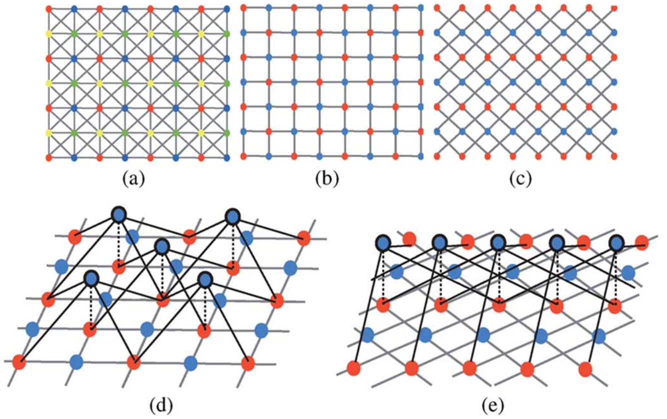


Fig. 6. (a) Image graph. (b) Rectangular bipartite subgraph. (c) Diagonal bipartite subgraph. (d) Oversampled rectangular bipartite graph. (e) Oversampled diagonal bipartite graph. The appended nodes are black circles filled with blue, and the appended edges are black lines.

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.

V. 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.

A. Image Graphs

Images can be viewed as graph signals by connecting each pixel with its eight neighboring ones, as shown in Fig. 6(a) [3]. Since this graph is four-colorable, it can be decomposed into rectangular (Fig. 6(b)) and diagonal (Fig. 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 [2], [3]. 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 oversampled graph bipartite. The rectangular oversampled image graph is illustrated in Fig. 6(d). The blue nodes are appended just above

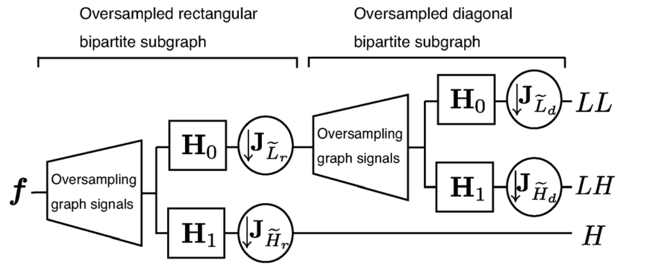

Oversampled rectangular bipartite subgraph
Oversampled diagonal bipartite subgraph
$\boldsymbol{f}$
Oversampling graph signals
$\mathbf{H}_0$
$\downarrow\mathbf{J}_{\widetilde{L}_r}$
$\mathbf{H}_1$
$\downarrow\mathbf{J}_{\widetilde{H}_r}$
Oversampling graph signals
$\mathbf{H}_0$
$\downarrow\mathbf{J}_{\widetilde{L}_d}$
LL
$\mathbf{H}_1$
$\downarrow\mathbf{J}_{\widetilde{H}_d}$
LH
H

Fig. 7. One-level decomposition of images. $\mathbf{J}_{\widetilde{L}_r}/\mathbf{J}_{\widetilde{H}_r}$ and $\mathbf{J}_{\widetilde{L}_d}/\mathbf{J}_{\widetilde{H}_d}$ denote the downsampling operations of the rectangular graph and the diagonal graph, respectively.

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 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. 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. 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.

B. Ring Graph

All ring graphs with an odd number of nodes are three-colorable as shown in Fig. 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. 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. 8(c) and 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 $\widetilde{L}$ and $\widetilde{H}$ are shown in Figs. 8(b) and 8(e), respectively. As shown in Fig. 8(e), $\widetilde{L}$ has nodes in F_1 and F_2 and $\widetilde{H}$ has those in F_1', F_2' and F_3. Therefore, the number of nodes in $\widetilde{L}$ and $\widetilde{H}$ 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.

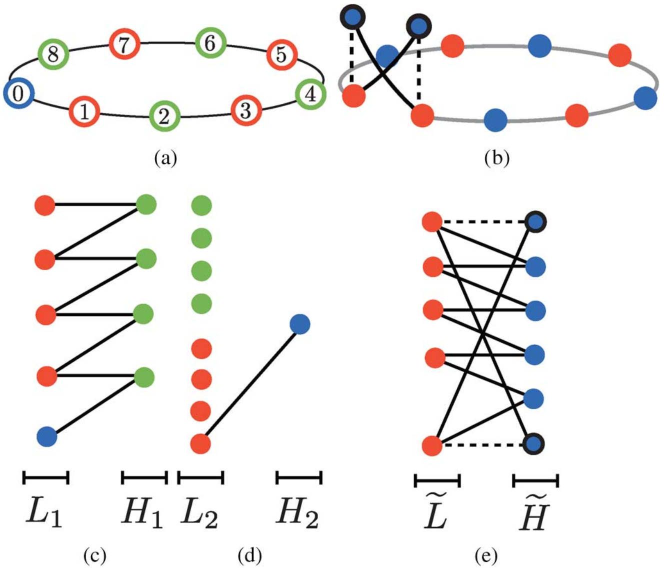


Fig. 8. (a) Ring graph $(n = 4)$. (b) Oversampled bipartite graph. (c) Sets L_1 and H_1 of the bipartite subgraph $\mathcal{B}_1$. (d) Sets L_2 and H_2 of the bipartite subgraph $\mathcal{B}_2$. (e) Sets $\widetilde{L}$ and $\widetilde{H}$ of the oversampled bipartite graph.

VI. Experimental Results

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

A. 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 [22], the Laplacian pyramid for regular signals [24], the critically sampled graphBior filter bank [2], and the Laplacian pyramid for graph signals [4]. The graph-based methods used the same filters (*graphBior(5,5)*). The Laplacian pyramid for regular signals used 9/7 filters and a reconstruction scheme using the pseudo inverse [25]. The graphBior used an edge-aware image graph [26]. The edge-aware image graphs were made as follows. The links around the edges of the images are classified into *regular* or *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. 6(a)–(c)). For example, the edge-aware image graphs of Fig. 9(a) are shown in Figs. 9(b) and 9(c). The graph Laplacian pyramid used the same graph and downsampling operation as graphBior for the lowpass channel. The proposed method used the oversampled edge-aware image graph [27]. In this case, on the basis of the edge-aware image bipartite graph, the

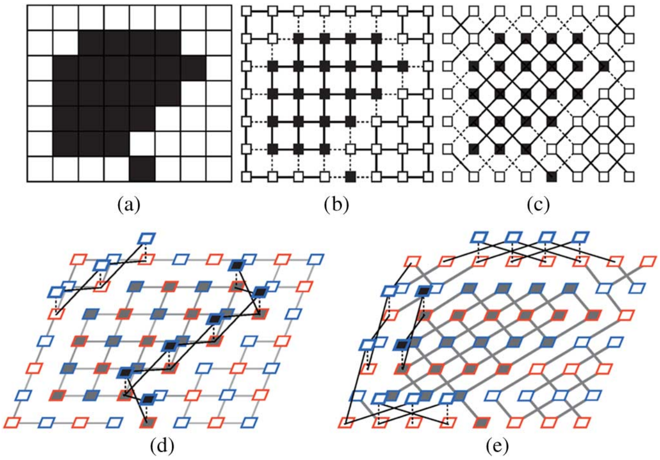


Fig. 9. (a) Original image. (b) Edge-aware rectangular bipartite graph. The solid and dashed lines are regular and less-reliable links, respectively. (c) Edge-aware diagonal bipartite graph. (d) Oversampled edge-aware rectangular bipartite graph. The black lines are additional edges. The dashed black 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.

TABLE I
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 [22]	24.57	31.76	35.83	43.10	52.87	60.05
Laplacian pyramid [24]	24.57	30.49	33.71	38.27	45.74	55.29
graphBior [3]	31.66	42.39	45.98	50.69	56.10	61.37
graph Laplacian pyramid [4]	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
graphBior	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
graphBior	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
graphBior	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

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. 9(a) are shown in Figs. 9(d) and 9(e). The critically sampled graph filter banks are applied to these graphs using the method described in Section V-A.

Table I 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 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 graphBior, in most cases. It can be seen that the vertical edges provide significant gains.

Fig. 10 shows images reconstructed from all lowpass coefficients and 3% of the highpass coefficients, and Fig. 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 in the reconstructed image of the proposed method are greatly suppressed compared with other graph-based transforms.

B. 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 IV. These original graphs can be decomposed into two bipartite graphs by using Harary's algorithm [3], [23], as shown in Figs. 12 and 13, respectively. The original signals are shown in Figs. 12(b) and 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 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 II. In order to verify the performance of the proposed oversampled graphs, we applied the graphBior filter bank to each graph.

1) 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 III summarizes the 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.

2) 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 graphBior filter bank using the oversampled graph, the lowpass signal was further decomposed on the basis of a downsampled graph consisting of vertices in the set $\widetilde{L}$ and edges in the original graph. Table IV shows the SNR values of the reconstructed signals from all lowpass coefficients and some fraction of the detail coefficients. As a benchmark, we applied graphBior with critically sampled GLM

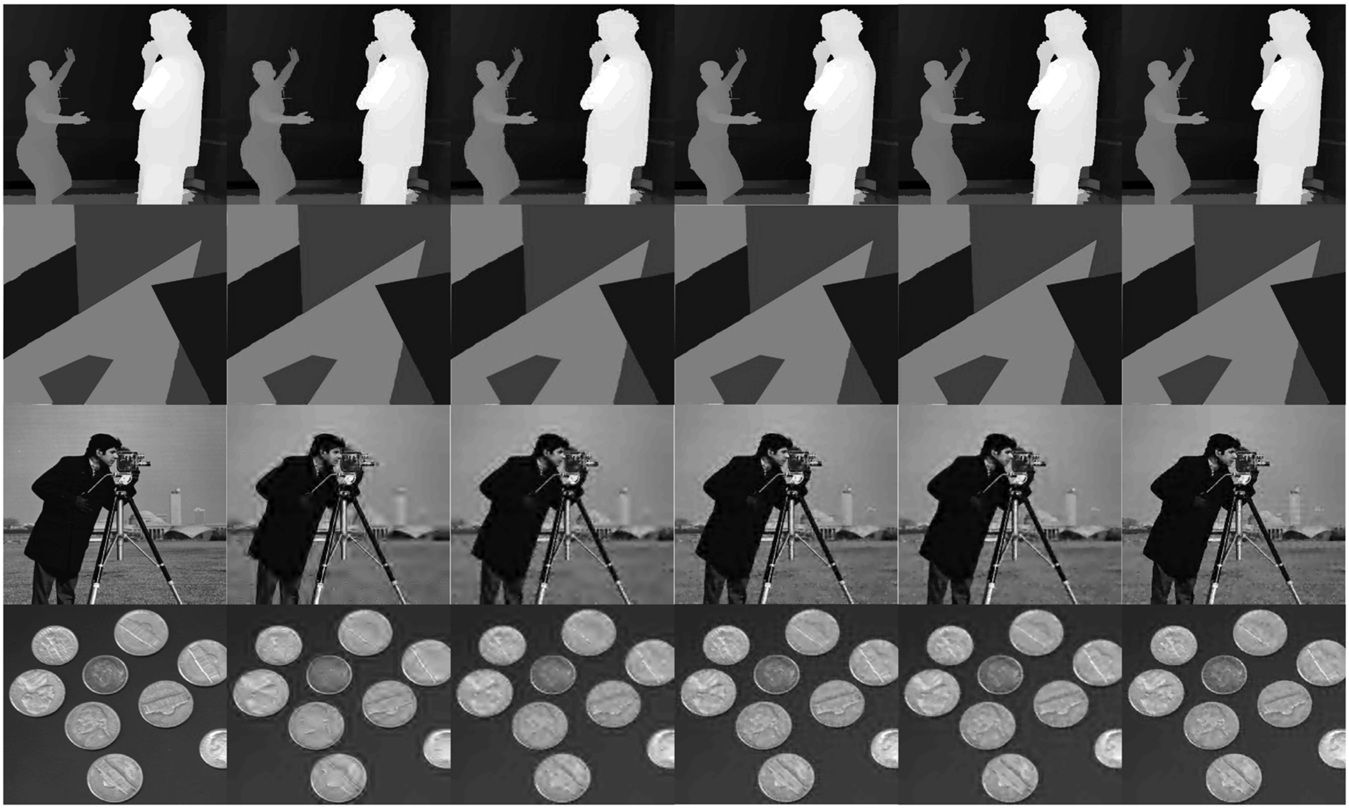
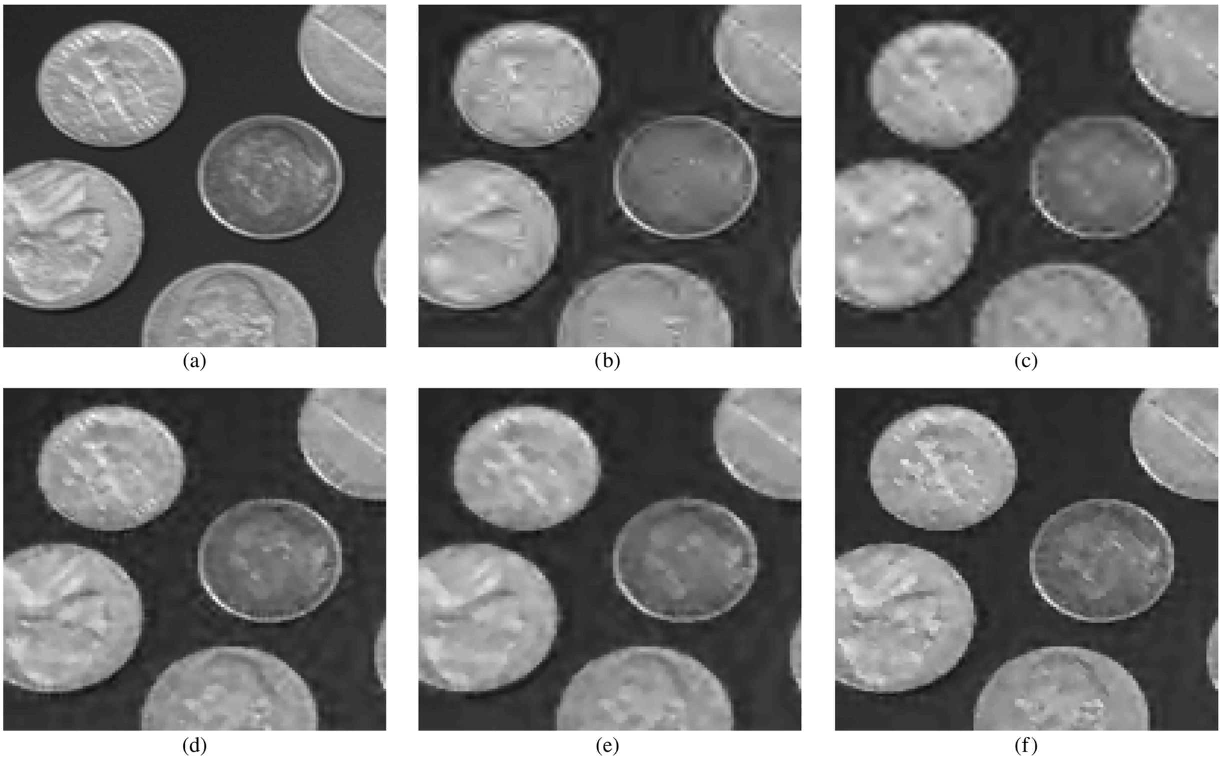


Fig. 10. Images reconstructed from all lowpass coefficients and 3% of the highpass coefficients after a three-level decomposition. From left to right: original image, CDF 9/7 wavelet, the Laplacian pyramid for regular signals, graphBior filter bank, the Laplacian pyramids for graph signals, and the proposed method with vertical edges. From top to bottom: *Ballet*, *Synthetic*, *Cameraman*, and *Coins*.

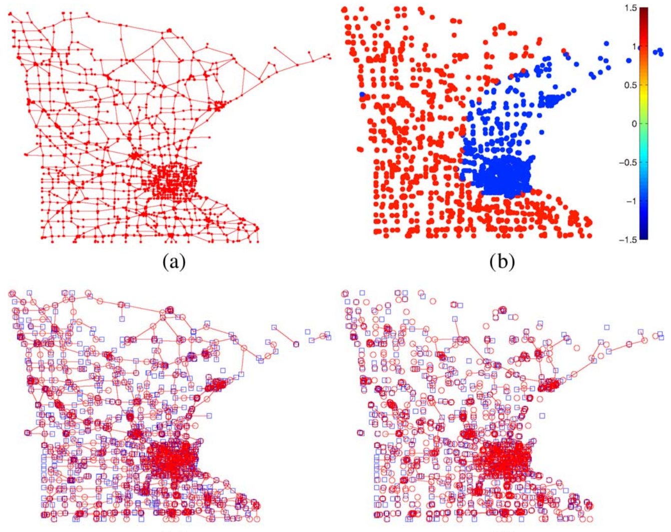


(a) (b) (c) (d) (e) (f)

Fig. 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: 29.64 dB). (c) Laplacian pyramid for regular signals (PSNR: 29.13 dB). (d) graphBior (PSNR: 31.75 dB). (e) Laplacian pyramid for graph signals (PSNR: 30.75 dB) (f) Proposed method with vertical edges (PSNR: 32.10 dB).

(CSGLM). The number of lowpass coefficients was the same for all methods.

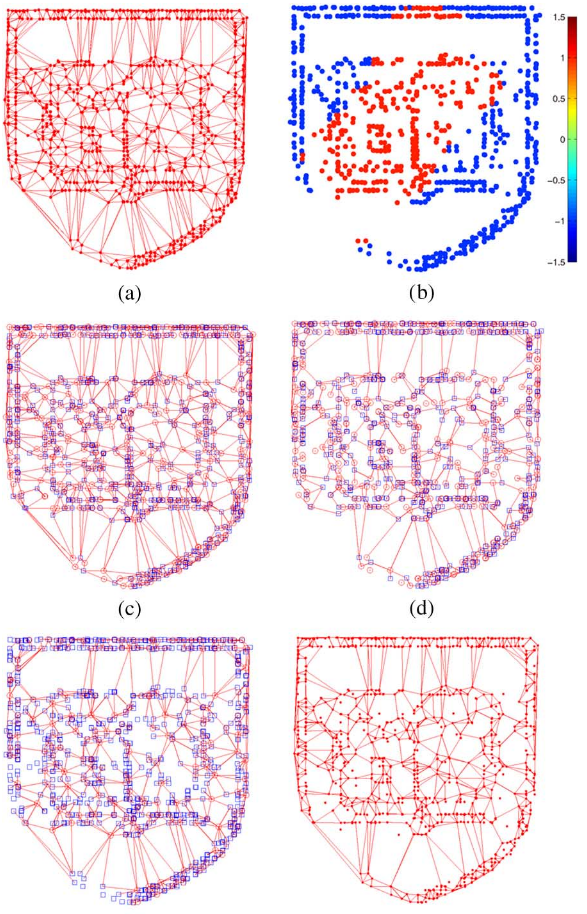
It can be seen that the proposed method performed better when the reconstructed signal was approximated from only



(a) (b)

(c) (d)

Fig. 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 [3]. (c) Bipartite subgraph #1. The blue squares and red circles indicate sets L and H, respectively. (d) Bipartite subgraph #2.



(a) (b)

(c) (d)

(e) (f)

Fig. 13. (a) Original graph of the *Yale Coat of Arms*. It was reproduced from the course website by Spielman [28]. 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 [5]. (c) Bipartite subgraph #1. The blue squares and red circles indicate sets L and H, respectively. (d) Bipartite subgraph #2. (e) Foundation graph of $\widetilde{\mathcal{G}}^{3}_{YC}$. (f) Remaining graph of $\widetilde{\mathcal{G}}^{3}_{YC}$.

TABLE II
NOTATION OF OVERSAMPLED BIPARTITE GRAPHS

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

TABLE III
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	–

TABLE IV
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

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 the oversampled graph with $l = K - 1$ without vertical edges ($\widetilde{\mathcal{G}}^{K-1}$) in the following experiments on arbitrary graphs.

C. 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. 14(a). The comparison is between the critically sampled bipartite graph and the oversampled bipartite graph. The lowpass filtered signals are shown in Figs. 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.

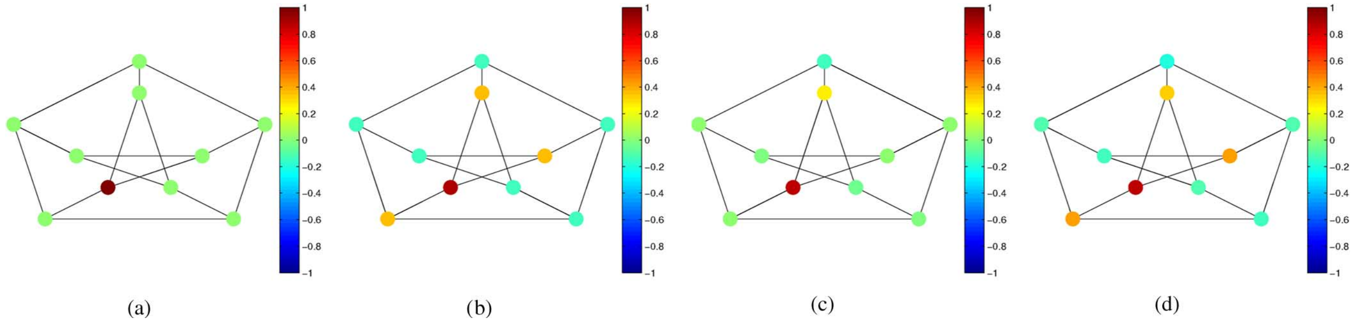


Fig. 14. Signal spread. (a) Input signal. (b) Lowpass filtered signal using the (non-bipartite) original graph. (c) Lowpass filtered signal using bipartite subgraph. (d) Lowpass filtered signal using oversampled bipartite graph.

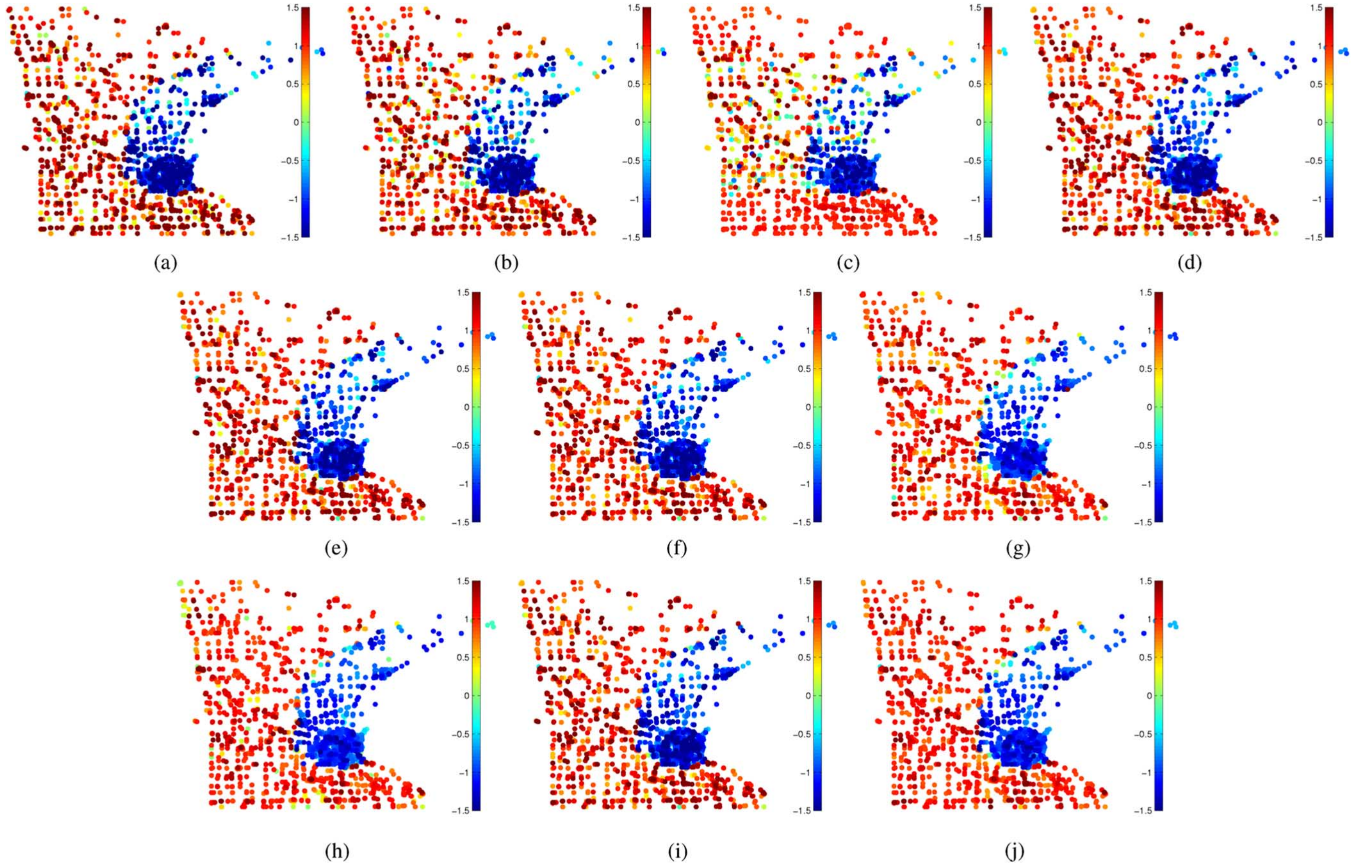
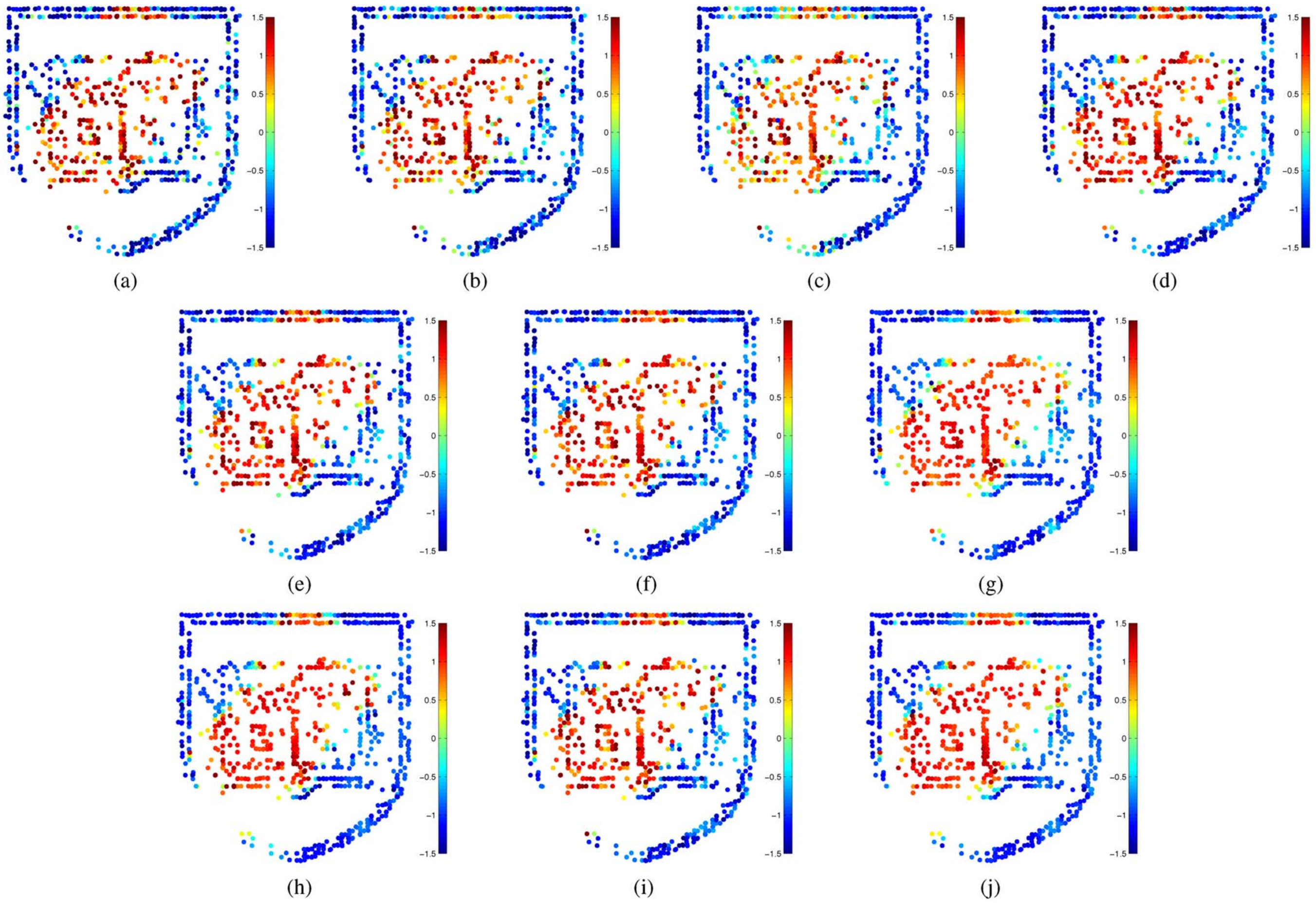


Fig. 15. Denoising results of *Minnesota Traffic Graph*. (a) Noisy observation; (b) sym8 (1 level); (c) sym8 (5 level); (d) graphBior with CSGLM; (e) GLP; (f) graphBior with BDC; (g) SGWT; (h) OSGFB with CSGLM; (i) graphBior with OSGLM; (j) OSGFB with OSGLM.

D. 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 critically sampled graphBior filter bank (abbreviated as graphBior with OSGLM) [2] or the four-channel oversampled graph filter bank (abbreviated as OSGFB with OSGLM) [16] on oversampled graphs.

We compared the above two methods with the regular one-dimensional wavelet *sym8* with one-level and five-level decompositions, graphBior(6, 6) (graphBior with CSGLM) [2], the Laplacian pyramid for graph signals (GLP) [4], the spectral graph wavelet transform (SGWT) with three scales [5], graphBior with the bipartite double cover (graphBior with BDC), and the four-channel oversampled graph filter bank with the bipartite graph decomposition (OSGFB with CSGLM) [16], [17]. 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 graphBior for the lowpass channel. All of the graph-based methods performed one-level transforms. That is, graphBior, 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 lowest frequency subband was kept, and the other high-frequency subbands were hard-thresholded with the threshold $T = 3\sigma$.



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

(e) (f) (g)

(h) (i) (j)

Fig. 16. Denoised results of the *Yale Coat of Arms*. (a) Noisy observation; (b) sym8 (1 level); (c) sym8 (5 level); (d) graphBior with CSGLM; (e) GLP; (f) graphBior with BDC; (g) SGWT; (h) OSGFB with CSGLM; (i) graphBior with OSGLM; (j) OSGFB with OSGLM.

TABLE V
DENOISED RESULTS OF MINNESOTA TRAFFIC GRAPH: SNR (dB)

σ	noisy	sym8 (1 level)	sym8 (5 levels)	graphBior (CSGLM)	GLP	SGWT	OSGFB (CSGLM)	graphBior (BDC)	graphBior (OSGLM)	OSGFB (OSGLM)
1/32	30.15	30.17	30.22	31.44	31.39	33.35	34.75	32.54	32.46	**35.08**
1/16	24.08	24.25	24.07	25.61	25.68	27.76	28.78	26.75	26.76	**29.34**
1/8	18.06	18.65	17.99	19.97	20.02	22.08	21.84	20.81	20.88	**23.17**
1/4	12.02	11.94	11.07	14.19	14.24	15.05	15.26	14.79	14.94	**17.63**
1/2	5.99	6.23	5.76	8.50	8.51	10.33	10.29	8.92	9.00	**12.31**
1	-0.02	1.59	3.13	2.63	2.61	**8.82**	4.24	3.03	3.11	7.04
Redundancy	-	1.00	1.00	1.00	2.05	4.00	4.00	2.00	1.37	2.74

TABLE VI
DENOISED RESULTS OF YALE COAT OF ARMS: SNR (dB)

σ	noisy	sym8 (1 level)	sym8 (5 levels)	graphBior (CSGLM)	GLP	SGWT	OSGFB (CSGLM)	graphBior (BDC)	graphBior (OSGLM)	OSGFB (OSGLM)
1/32	30.15	29.64	29.41	29.85	30.10	29.40	30.11	31.55	31.33	**31.77**
1/16	24.10	23.67	23.33	24.24	24.71	23.70	24.66	25.43	25.32	**26.80**
1/8	18.08	18.02	17.04	18.75	19.21	18.24	19.05	20.01	19.90	**21.22**
1/4	12.01	11.36	10.19	13.16	13.58	12.95	13.98	14.39	14.29	**15.21**
1/2	6.04	6.19	4.98	8.66	8.80	8.85	10.25	9.23	9.23	**10.49**
1	0.00	1.82	2.21	3.94	4.00	6.47	7.45	3.88	4.11	**7.48**
Redundancy	-	1.00	1.00	1.00	1.79	4.00	4.00	2.00	1.78	3.56

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

Figs. 15 and 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 OSGFB with OSGLM performed better than the other transforms.

VII. Conclusion

This paper 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.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper, especially their suggestion on the bipartite double cover.

References

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

[2] S. K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Trans. Signal Process.* vol. 61, no. 19, pp. 4673–4685, 2013.

[3] 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.

[4] D. I. Shuman, M. J. Faraji, and P. Vandergheynst, "A multiscale pyramid transform for graph signals," 2014, arXiv preprint arXiv:1308.4942 [Online]. Available: http://www.macalester.edu/~dshuman1/publications.html

[5] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.* vol. 30, no. 2, pp. 129–150, 2011 [Online]. Available: http://wiki.epfl.ch/sgwt

[6] 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*, 2010, pp. 367–374.

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

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

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

[10] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 53–94, 2006.

[11] 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.

[12] 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.

[13] D. I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vandergheynst, "Spectrum-adapted tight graph wavelet and vertex-frequency frames," 2013, arXiv preprint arXiv:1311.0897 [Online]. Available: http://documents.epfl.ch/users/s/sh/shuman/www/publications.html

[14] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Appl. Comput. Harmon. Anal.*, 2013, submitted for publication.

[15] S. G. Mallat, *A Wavelet Tour of Signal Processing*. New York, NY, USA: Academic, 1999.

[16] Y. Tanaka and A. Sakiyama, "M-channel oversampled graph filter banks," *IEEE Trans. Signal Process.*, vol. 62, pp. 3578–3590, 2014.

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

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

[19] E. Sampathkumar, "On tensor product graphs," *J. Austral. Math. Soc.*, vol. Series A, pp. 268–273, 1975.

[20] A. L. Dulmage and N. S. Mendelsohn, "Coverings of bipartite graphs," *Can. J. Math.*, pp. 517–534, 1958.

[21] F. R. K. Chung, *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. Providence, RI, USA: Amer. Math. Soc., 1997.

[22] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, 1992.

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

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

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

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

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

[28] D. Spielman, Spectral Graph Theory, 2012 [Online]. Available: http://www.cs.yale.edu/homes/spielman/561/

Akie Sakiyama (S'14) received the B.E. degree in computer and information sciences from Tokyo University of Agriculture and Technology, Tokyo, Japan, in 2014. She is currently a Ph.D. student with Graduate School of BASE, Tokyo University of Agriculture and Technology. Her current research interests are graph signal processing and high-dimensional data analysis.

Yuichi Tanaka (S'06–M'07) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2003, 2005, and 2007, respectively. He was a Postdoctoral Scholar at Keio University, Yokohama, Japan, from 2007 to 2008 supported by the Japan Society for the Promotion of Science (JSPS). From 2006 to 2008, he was also a visiting scholar at the University of California, San Diego (Video Processing Group supervised by Prof. T. Q. Nguyen). From 2008 to 2012, he was an Assistant Professor in the Department of Information Science, Utsunomiya University, Tochigi, Japan. Since 2012, he has been an Associate Professor in Graduate School of BASE, Tokyo University of Agriculture and Technology, Koganei, Japan. He is a recipient of Yasujiro Niwa Outstanding Paper Award in 2010, and also TELECOM System Technology Award in 2011. Since 2013, he has been an Associate Editor of *IEICE Transactions on Fundamentals*. His current research interests are in the field of multidimensional signal processing which includes: graph signal processing, image and video coding with computer vision techniques, distributed video coding, objective quality metric, and effective time-frequency transform design.