JOINT FILTERING OF GRAPH AND GRAPH-SIGNALS

Nicolas Tremblay(1,2), *Pierre Borgnat*(3)

[1] INRIA Rennes - Bretagne Atlantique, Beaulieu Campus, Rennes, France
[2] Institute of Electrical Engineering, EPFL, Lausanne, Switzerland
[3] CNRS, Ecole Normale Supérieure de Lyon, LPENSL, France
nicolas.tremblay@inria.fr, pierre.borgnat@ens-lyon.fr

ABSTRACT

Joint filtering of signals indexed on a graph consists in filtering not only the signal, but also the graph by an appropriate downsampling. Existing methods for filtering and downsampling graph signals approximate graphs as sums of bipartite graphs or use nodal domains of the Laplacian. Here, a different method is introduced, and is based on the partitioning in meaningful subgraphs of the graph itself, e.g. network's communities; this partition may be interpreted as a coarsening of the graph and may also be tailored to be aware of the signal structure. A method is proposed to create filterbanks that compute, for graph signals, an approximation and several details using the partition to downsample the graph. This means that we jointly filter the graph and the graph signal; it leads to the design of a new subgraph-based filterbank for graph signals. This design is tested on simple examples for compression and denoising.

Index Terms— graph signal processing, graph filtering, filterbanks, communities

1. INTRODUCTION

Graph signal processing is an emerging field meant to study signals defined on graphs by importing and adapting tools from classical signal processing to the graph context [1]. Among major tools in signal processing, filterbanks and particularly wavelet filterbanks [2] are important as they allow to decompose a signal in components of various frequencies and provide powerful tools, for instance for denoising and compression.

An element makes the adaptation of filterbanks to graph signal not straightforward: the graph itself is not the same from one situation to another, and the structure of the graph has to be taken into account to define filterbanks for graph signals. In the present work, we propose to follow the idea of jointly filtering the graph signal and the graph: the graph will be appropriately downsampled on a partition of the graph into connected subgraphs and this constitues a coarsening of the graph, while the graph signal is filtered on these connected subgraph.

We first present background elements required to the present work and discuss relevant works in Section 2. Then, we define in Section 3 the proposed filterbanks based on downsampling in connected subgraphs. Some examples are given in Section 4 and we conclude in Section 5.

This work was partly funded by the European Research Council, PLEASE project (ERC-StG-2011-277906); and the ANR-14-CE27-0001 GRAPHSIP grant.

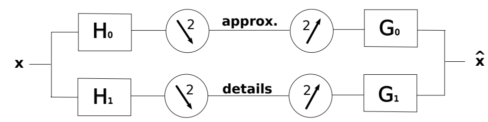


Fig. 1. Classical two channel filterbank.

2. BACKGROUND AND RELATED WORKS

2.1. The Graph Fourier Transform

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ be a undirected weighted graph with $\mathcal{V}$ the set of nodes, $\mathcal{E}$ the set of edges, and $\mathbf{A}$ the weighted adjacency matrix such that $\mathbf{A}_{ij} = \mathbf{A}_{ji} \geq 0$ is the weight of the edge between nodes i and j. Note N the total number of nodes.

Let us define the graph's Laplacian matrix $\boldsymbol{\mathcal{L}} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \mathbf{d}_i = \sum_j \mathbf{A}_{ij}$ the strength of node i. $\boldsymbol{\mathcal{L}}$ is real symmetric, therefore diagonalizable: its spectrum is composed of $(\lambda_l)_{l=1\dots N}$ its set of eigenvalues that we sort: $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_N$, and of $\mathbf{Q}$ the matrix of its normalized eigenvectors: $\boldsymbol{Q} = (\boldsymbol{q}_1|\boldsymbol{q}_2|\dots|\boldsymbol{q}_N)$.

A defining analogy for graph Fourier transform is to consider these eigenvalues to play the role of "frequencies" [1], and the corresponding eigenvectors the role of Fourier modes. This means that the graph Fourier transform of a signal defined on the nodes of a graph is the decomposition of this graph signal onto the basis formed by these eigenvectors. Hence, the graph Fourier transform of a signal $\boldsymbol{x}$ reads as $\hat{\boldsymbol{x}} = \boldsymbol{Q}^\top \boldsymbol{x}$.

2.2. Filterbanks

A classical filterbank is an array of filters meant to separate a signal into several components, each in a specific band of frequency. A classical scheme is the two-channel multirate system as in Fig. 1, where the two filtered channels (one low-pass and one high-pass) are followed by a decimation operator $(\downarrow \mathbf{2})$, keeping only one every two samples, to downsample the output.

Let us recall briefly some elements of wavelet filterbanks [2]. Let us consider a discrete signal $\boldsymbol{x}$ of size N. The first channel gives an approximation $\boldsymbol{x}_1$ of size $N/2$:

$$\boldsymbol{x}_1 = (\downarrow \mathbf{2})\boldsymbol{C}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{C}$ is a smoothing operator associated to the scaling function; for the Haar filterbank, it is the sliding average operator. The second channel gives the detail $\boldsymbol{x}_2$ of size $N/2$:

$$\boldsymbol{x}_2 = (\downarrow \mathbf{2})\boldsymbol{D}\boldsymbol{x}, \tag{2}$$

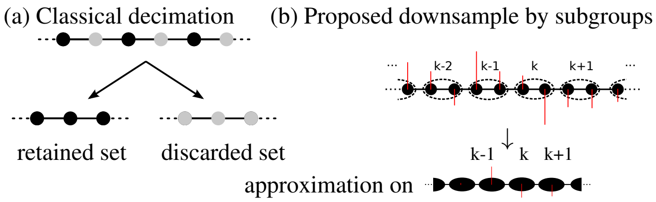


Fig. 2. Two approaches to downsampling for filterbanks: (a) classical decimation where one node over two is kept and the rest (in grey) is discarded; (b) proposed downsampling by subgraphs; here, the nodes are partitioned in subgraphs of 2 nodes and the approximation is on the "super-nodes" k that stand for each subgraph.

where $\boldsymbol{D}$ is the convolution by the wavelet; for Haar filterbank, this is the sliding difference operator.

Wavelet filterbanks can be made both orthonormal (i.e., exactly inversible with its transpose as inverse) and critically sampled (after filtering, the number if coefficients is N, equal to the number of initial coefficients). Decimation $(\downarrow \mathbf{2})$ has here a central role as it allows to keep the number of coefficients constant even though one obtains separate low-frequency approximations and high-frequency details. Decimation in this case follows a "one every two samples paradigm", as illustrated on Fig. 2 (a).

For graph signals, filterbanks following the same "one every two nodes paradigm" (as nodes take the role of samples) have already been defined in the literature. Narang and Ortega [3, 4] propose to consider signals defined on bipartite graphs, as one can then downsample the graph by naturally keeping one of the two partite sets, i.e., one every two nodes when following the edges in the graph. Then they generalize that to any graph by an approximate decomposition in bipartite graphs. Other graph filterbanks use bipartite graphs, such as Sakiyama and Tanaka's oversampling method [5], or Nguyen and Do's maximum spanning tree method [6]. Another way has been proposed Shuman et al. [7]. They use the polarity (sign) of the eigenvector associated to the maximum eigenvalue of the Laplacian, the one associated to the "highest frequency" of a signal defined on the graph. Again, this leads to keeping "one every two nodes" for the next level (e.g., the nodes with positive polarity).

In the present work, we will propose a different downsampling paradigm to define a filterbank for graph signals: the downsampled support of a graph signal will be defined via the grouping of nodes in connected subgraphs of the original graph.

2.3. Partitioning a graph in connected subgraphs

Techniques to partition a graph into subgraphs that are connected are numerous in the literature, e.g., see [8], [9], or even [10] that partitions graph according to its nodal domains (as used in [7] for the one with largest frequency).

In the following, we seek a design of graph partitioning that will typically transform the original signal in a sparser one after analysis. For that, we decide to look for partitions that separate the graph into groups of nodes more connected to themselves than with the rest of the graph – these subgraphs are also known as communities [11]. As there are many manners to partition graphs in communities (they are reviewed in [11] and a graph processing approach can be found in [12]), we limit here the discussion to one method: **the greedy Louvain method** [13] **for maximizing the modularity** [14], a well-known objective function that measures the quality of a partition in communities.

Consider an arbitrary graph $\mathcal{G}$ and an arbitrary partition of this graph in K connected subgraphs $\{\mathcal{G}^k\}_{k\in\{1,\dots,K\}}$. Write N_k the number of nodes in subgraph $\mathcal{G}^k$ of label k. Let us first define the matrix $\boldsymbol{S} \in \mathbb{R}^{N\times K}$, a practical way (for linear algebra calculus) to encode the connected subgraph structure:

$$\boldsymbol{S} = \left(\mathbb{1}_{\text{Csub 1}}|\mathbb{1}_{\text{Csub 2}}|\cdots|\mathbb{1}_{\text{Csub K}}\right)\,. \tag{3}$$

where $\mathbb{1}_{\text{Csub k}}$ is subgraph k's indicator function, i.e.:

$$\begin{aligned}\mathbb{1}_{\text{Csub k}}(i) &= 1 \text{ if } i \in \mathcal{G}^k \\ &= 0 \text{ if not.}\end{aligned} \tag{4}$$

For a partition described by $\boldsymbol{S}$, the modularity is defined as

$$Q(\boldsymbol{S}) = \frac{1}{2m}\text{Trace}\left(\boldsymbol{S}^T(\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^T}{2m})\boldsymbol{S}\right) \tag{5}$$

where $\mathbf{d}$ is the vector of strength (or degrees for unweighted graphs) with $d_i = \sum_j A_{ij}$, and $2m = \sum_i d_i$.

For maximizing (approximately) Q, the Louvain algorithm repeats two steps iteratively. Initially, each node is supposed to be in its own community. (1) Choose a node and move it to its adjacent group that increases Q the most; continue with another node until no individual move of node can increase Q. Then (2) Merge the nodes that are in the same community to obtain a new network, where the new nodes are the communities of (1). After that, go back to (1) and iterate. The procedure makes the existing groups grow in size by absorption of other nodes/groups. The procedure stops when step (1) becomes unable to increase Q.

In the following, we will change an aspect of the algorithm: we stop the iterations once the desired sizes of the subgraphs is reached.

3. FILTERBANKS BY DOWNSAMPLING ON CONNECTED SUBGRAPHS

We let go of the "one every two nodes" sampling paradigm, and focus on a novel approach to downsampling that is based on a partition of the graph in many connected subgraphs: the downsample is made by coarsening the structure of the graph thanks to the definition of "supernodes", each one of them representing a connected subgraph of the partition. Also, in the present work we do not define separately the downsampling operators $(\downarrow \mathbf{2})$ and the filter operators $\boldsymbol{C}$ and $\boldsymbol{D}$ as in eq. (1) and (2) for graph signals. Instead, we define directly an operator $\boldsymbol{L}$ having the global effect of low-pass and downsampling, and other ones $\boldsymbol{B}$ that combine band-pass and downsampling. For classical two-channel filterbanks, $\boldsymbol{L} = (\downarrow \mathbf{2})\boldsymbol{C}$ and $\boldsymbol{B} = (\downarrow \mathbf{2})\boldsymbol{D}$.

3.1. Proposed method

Definition of required local operators. The first ingredient is to consider a partition of the graph $\mathcal{G}$ into K subgraphs $\mathcal{G}^k$ which are connected. Each $\mathcal{G}^k$ is composed of N_k nodes, that we note $\{v_{\sigma^k(1)}, v_{\sigma^k(2)}, \dots, v_{\sigma^k(N_k)}\}$. Consider the sampling operator $\boldsymbol{C}_k^\top$, that takes only the values from $\mathcal{G}^k$ on $\mathcal{G}$; its transpose is expressed as a matrix of size $N \times N_k$:

$$\boldsymbol{C}_k = \left(\boldsymbol{\delta}_{\sigma^k(1)}|\boldsymbol{\delta}_{\sigma^k(2)}|\cdots|\boldsymbol{\delta}_{\sigma^k(N_k)}\right), \tag{6}$$

where $\boldsymbol{\delta}_{\sigma^k(i)}(j) = 1$ if $\sigma^k(i) = j$, and zero otherwise.

From that, the adjacency matrix $\boldsymbol{A}$ may be written as the sum of two adjacency matrices: $\boldsymbol{A} = \boldsymbol{A_i} + \boldsymbol{A_e}$ where $\boldsymbol{A_i}$ is the intra-subgraph adjacency matrix, i.e. sampled by subgraphs:

$$\boldsymbol{A_i} = \sum_{k=1}^{K} \boldsymbol{C}_k \boldsymbol{C}_k^\top \boldsymbol{A} \boldsymbol{C}_k \boldsymbol{C}_k^\top . \tag{7}$$

It keeps only the edges within each subgraph. The second one, $\boldsymbol{A_e}$ will only keep the edges connecting subgraphs together.

For each subgraph $\mathcal{G}^k$, we note $\boldsymbol{A}_{\boldsymbol{i}}^k$ the reduction of $\boldsymbol{A_i}$ to $\mathcal{G}^k$:

$$\boldsymbol{A}_{\boldsymbol{i}}^k = \boldsymbol{C}_k^\top \boldsymbol{A_i} \boldsymbol{C}_k, \tag{8}$$

and $\boldsymbol{\mathcal{L}}_{\boldsymbol{i}}^k$ the local Laplacian associated to $\boldsymbol{A}_{\boldsymbol{i}}^k$. It is diagonalisable, of size N_k, and

$$\boldsymbol{\mathcal{L}}_{\boldsymbol{i}}^k = \boldsymbol{Q}^k \boldsymbol{\Lambda}^k \boldsymbol{Q}^{k\top}, \tag{9}$$

with $\boldsymbol{\Lambda}^k = \operatorname{diag}(\lambda_1^k, \lambda_2^k, \ldots, \lambda_{N_k}^k)$, the diagonal matrix of sorted eigenvalues (λ_1^k being the smallest), and $\boldsymbol{Q}^k$ the orthonormal basis of local Fourier modes:

$$\boldsymbol{Q}^k = \left(\boldsymbol{q}_1^k | \boldsymbol{q}_2^k | \ldots | \boldsymbol{q}_{N_k}^k \right). \tag{10}$$

Lastly, for each eigenvector $\boldsymbol{q}_i^k$ of size N_k defined on the local subgraph $\mathcal{G}^k$, we note $\bar{\boldsymbol{q}}_i^k$ its zero-padded extension to the whole global graph:

$$\forall k \in \{1, K\} \quad \forall i \in \{1, N_k\} \qquad \bar{\boldsymbol{q}}_i^k = \boldsymbol{C}_k \boldsymbol{q}_i^k. \tag{11}$$

Definition of the different analysis channels. We propose to define the equivalent of $\boldsymbol{L}$ (for the approximation) as a low-pass filtering inside each subgraph, i.e. on $\boldsymbol{A_i}$ only, followed by a downsample of each group in one super-node only. For that, the simple solution studied here is to make an average of the signals on each subgroup.

Given that the first eigenvector of a Laplacian is simply constant over all the nodes, it turns out that the local average can be expressed using the local Laplacian's associated to the $\boldsymbol{A}_{\boldsymbol{i}}^k$, as being the projection of the signal over:

$$\boldsymbol{Q}_1 = \left(\bar{\boldsymbol{q}}_1^1 | \bar{\boldsymbol{q}}_1^2 | \cdots | \bar{\boldsymbol{q}}_1^K \right). \tag{12}$$

Then, for the details, we will make use of the other local graph Fourier modes (i.e., eigenvectors of the Laplacian). In the case of the Haar filterbanks, as in Fig. 2 (b), the details are in reality given by the projection on the grouping of the second Fourier modes of each subgraph of 2 nodes, as this second Fourier mode is $(1, -1)/\sqrt{2}$, i.e. the Haar wavelet. We propose, by analogy, to keep as a second channel the projection of the signal over:

$$\boldsymbol{Q}_2 = \left(\bar{\boldsymbol{q}}_2^1 | \bar{\boldsymbol{q}}_2^2 | \cdots | \bar{\boldsymbol{q}}_2^K \right), \tag{13}$$

where we omit in the list the eigenvectors that would be associated to subgraphs $\mathcal{G}^k$ that are singleton, because then $\bar{\boldsymbol{q}}_2^k$ does not exist. The size of $\boldsymbol{Q}_2$ is then $N \times M_k$ where M_k is the number of subgraphs $\mathcal{G}^k$ that have at least 2 nodes (i.e., it is K minus the number of singleton in the partition in $\mathcal{G}^k$'s).

We continue to define other detail channels by considering the third local graph Fourier modes, then the fourth, and so on. In each channel, the atom of decomposition is:

$$\boldsymbol{Q}_l = \left(\bar{\boldsymbol{q}}_l^{I_l(1)} | \bar{\boldsymbol{q}}_l^{I_l(2)} | \cdots | \bar{\boldsymbol{q}}_l^{I_l(|I_l|)} \right), \tag{14}$$

where I_l is the list of subgraph labels containing at least l nodes. This means that operator $\boldsymbol{Q}_l$ groups together all local Fourier modes

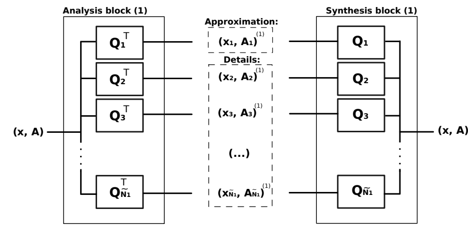
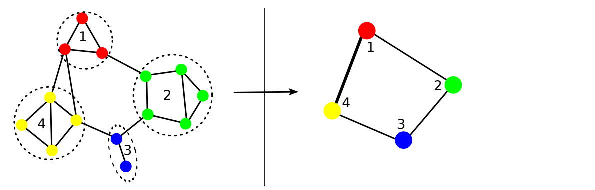


Fig. 3. Scheme of the proposed filterbanks on connected subgraphs. The number of channels $\tilde{N}_1$ is equal to the size of the largest subgraph in the used partition.



Original graph Coarsened graph of super-nodes

Fig. 4. Partition of a graph into communities.

associated to the l-th eigenvalue of all subgraphs containing at least l nodes. The number of channel $\tilde{N}_1$ is equal to the size of the largest subgraph in the partition in $\mathcal{G}^k$'s.

Analysis filterbank. Given a signal $\boldsymbol{x}$ defined on the graph, its decomposition through the proposed filterbanks is done by $\tilde{N}_1$ channels, as shown on Fig. 3:

$$\forall l \in \{1, ..., \tilde{N}_1\} \qquad \boldsymbol{x_l} = \boldsymbol{Q}_l^\top \boldsymbol{x}. \tag{15}$$

Each of them is defined on a graph whose adjacency matrix reads:

$$\boldsymbol{A}_l = \boldsymbol{S}_l^\top \boldsymbol{A}_e \boldsymbol{S}_l, \tag{16}$$

when using grouping operators

$$\boldsymbol{S}_l = \left(\mathbb{1}_{\text{Csub } I_l(1)} | \mathbb{1}_{\text{Csub } I_l(2)} | \ldots | \mathbb{1}_{\text{Csub } I_l(|I_l|)} \right), \tag{17}$$

meaning that $\boldsymbol{S}_l$ groups together indicator functions of subgraphs containing at least l nodes.

3.2. Properties

This filterbank is critically sampled, because it collects (after reorganization) the N_k coefficients associated to each local graph Fourier matrix (associated to $\mathcal{G}^k$). As a partition of $\mathcal{G}$, the subgraphs satisfy that $\sum_{k=1}^{K} N_k = N$, hence there are N total coefficients in the different channel outputs $\boldsymbol{x_l}$, the same as the N samples of the graph signal $\boldsymbol{x}$.

Also, because the local graph Fourier modes are bases (being eigenvectors of the local Laplacians), they are invertible and their inverse are their transpose. It means that the synthesis block is directly obtained by taking the transpose of the analysis filter and can be written as $\begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 & \ldots & \boldsymbol{Q}_{\tilde{N}_1} \end{bmatrix}^\top$. The perfect reconstruction comes from the fact that:

$$\begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 & \ldots & \boldsymbol{Q}_{\tilde{N}_1} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}_1^\top \\ \boldsymbol{Q}_2^\top \\ \ldots \\ \boldsymbol{Q}_{\tilde{N}_1}^\top \end{bmatrix} = \boldsymbol{I}_N, \tag{18}$$

(a) Original (b) Noisy SNR 12.12

(c) CoSub, SNR 16.81 (d) EdAwCoSub, SNR 23.27

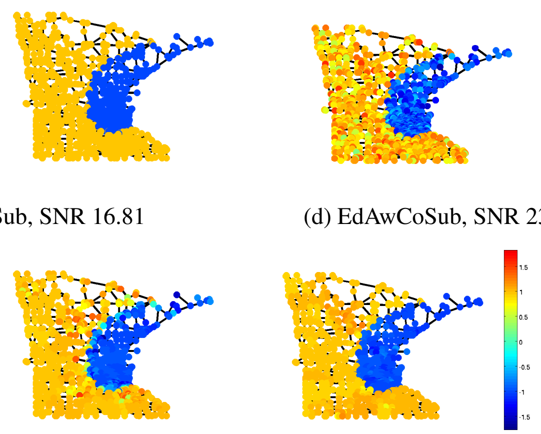


Fig. 5. Example of denoising on the Minnnesota traffic network. The resulting SNR is indicated above each graph.

where $\boldsymbol{I}_N$ is the identity matrix of size N.

If we keep $\{\boldsymbol{x}_l\}_{l\in\{1,\tilde{N}_1\}}$ and the graph structure and partition, we can perfectly recover $\boldsymbol{x}$. Indeed, from the subgraph structure, one can compute again the $\{\boldsymbol{Q}_l\}_{l\in\{1,\dots,\tilde{N}_1\}}$. Then, the original signal is exactly:

$$\tilde{\boldsymbol{x}} = \sum_{l=1}^{\tilde{N}_1} \boldsymbol{Q}_l \boldsymbol{x}_l = \sum_{l=1}^{\tilde{N}_1} \boldsymbol{Q}_l \boldsymbol{Q}_l^\top \boldsymbol{x} = \boldsymbol{x}. \tag{19}$$

We show in Fig. 3 a representation of the analysis and synthesis blocks of the proposed graph filterbanks.

3.3. Joint processing of signal and graph

If the partition in subgraphs is obtained via community detection, as proposed in Section 2.3, $\boldsymbol{A_i}$ will contain many edges while $\boldsymbol{A_e}$ may be comparatively sparse. This is illustrated in Fig. 4. Moreover, if the communities can also be selected such that they follow the structure of the graph signal, the details will also be relatively sparse.

To explore that, we define two different types of filterbanks:

- **CoSub**, short for Connected Subgraphs Filterbanks, where we directly apply the Louvain algorithm for maximizing modularity $Q(\boldsymbol{S})$ for the adjacency matrix $\boldsymbol{A}$, and the obtained partition $\boldsymbol{S}$ is used to write $\boldsymbol{A} = \boldsymbol{A_i} + \boldsymbol{A_e}$ as defined before;
- **EdAwCoSub**, short for Edge Aware Connected Subgraphs Filterbanks: first we consider a modified adjacency matrix that takes the signal $\boldsymbol{x}$ into account thanks to

$$\begin{aligned} \boldsymbol{A_x}(i,j) &= e^{-\frac{(\boldsymbol{x}(i)-\boldsymbol{x}(j))^2}{2\sigma_x^2}} && \text{if } \boldsymbol{A}(i,j) \neq 0 \\ &= 0 && \text{if } \boldsymbol{A}(i,j) = 0 \end{aligned} \tag{20}$$

where $\sigma_x = \text{std}(\{|\boldsymbol{x}(i) - \boldsymbol{x}(j)|\}_{i\sim j})$ ($i \sim j$ means i neighbor to j in $\boldsymbol{A}$). Then the Louvain algorithm is applied on $\boldsymbol{A_x}$. Finally, the obtained partition $\boldsymbol{S}$ enables us to decompose the original adjacency matrix in $\boldsymbol{A} = \boldsymbol{A_i} + \boldsymbol{A_e}$.

(a) Original (b) EdAwGrBior SNR 28.3

(c) CoSub, SNR 26.56 (d) EdAwCoSub, SNR 30.30

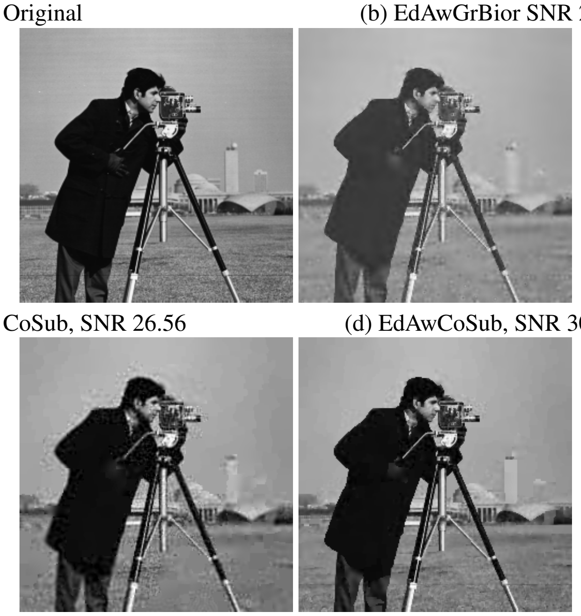


Fig. 6. Example of compression on an image, keeping aroung 3 % of the coefficients. The resulting SNR is indicated above each graph. The proposed method EdAwCoSub with edge awareness obtained here a better SNR.

With some awareness of the graph signal, we expect the second method to perform better if the graph and the graph signal can be filtered jointly in a consistent manner (i.e., the communities found by the Louvain algorithm from the structure of the graph are also meaningful for the signal).

3.4. Cascade

It is possible to cascade the filterbank as usual. For each channel, one iterates the same analysis scheme, thereby obtaining successive approximations and details of the original signal at different scales of analysis. Often, we will cascade the construction only on the approximation signal at each level of the cascade. Remember that the number of channels is adaptative and changes down the cascade: it is equal to the maximal number of nodes in the largest subgraph in the considered partition at this level.

4. EXAMPLES

Among the classical use of filterbanks, one finds denoising and compression. We illustrate the proposed construction on two specific examples of these applications.

First, we look at an example in compression on the Minnesota road graph, as shown in Fig. 5. A piece-wise constant graph signal (that has only two possible values: +1 and -1), is corrupted with an additive Gaussian noise of standard deviation σ (equal to 1/4 here). We attempt to restore the original image by doing an analysis with one level of filterbank, and we reconstruct a signal from all low-pass coefficients and hard thresholded high-pass coefficients that have absolute value higher than a threshold $T = 3\sigma$. These preliminary results show that the method performs well, and that the output of EdAwCoSub –the version of the algorithm that takes into account

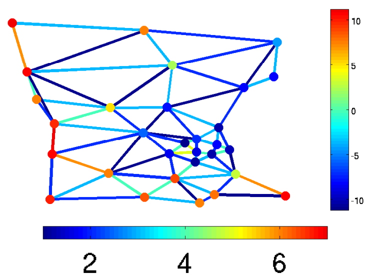


Fig. 7. Joint filtering of graph and graph signal of Fig. 5 (a): the graph is coarsened by the proposed method while the signal is filtered though the proposed filterbank.

the signal– is better. This is because the detected communities are better adapted to the signal.

The second application is the use of the graph signal processing approach for classical images, as one can easily consider an image as a graph signal over the rectangular grid. We consider the square benchmark image *cameraman*, of size 256×256 ($N = 65536$). We use the proposed filterbank with only 1 level of cascade, where the communities found contain $K = 546$ subgraphs for CoSub. We keep the approximation signal of size 546 and 3.71% of the high-pass coefficients in order to have exactly a number 2959 of non-zero coefficients before reconstruction – which is the number of non-zero coefficients when keeping the approximation obtained after three levels of a classical filterbank, and 3% of its high-pass coefficients. For EdAwCoSub, we do the same and retain the same number of coefficients; finally, we compare the method to the Graph Bior filterbank [4] with Nonzero DC and including edge-awareness of [15] (EdAwGrBior), for the same compression rate.

Comparing the restored images after these compressions with the two proposed filterbanks (CoSub and EdAwCoSub), and with EdAwGrBior, the result is that, given awareness of the graph signal, our new proposed scheme is on par with (here slightly better than) other graph-based filterbanks.

A last example illustrates the potentialities of the proposed filterbanks for joint filtering of graph and graph signal. On Fig. 7, the Minnesota traffic graph and a function on it as on Fig. 5 (a) are processed through the filterbank and we show only the resulting low-pass filtered signal on the resulting filtered (or coarsened) adjacent matrix. The result can really be seen as a step towards joint filtering of signals and graphs.

5. CONCLUSION

We have proposed a new scheme to obtain a multi rate, orthogonal and critically sampled filterbank for graph signals. It relies on the partition of the underlying graph in connected subgraphs. This partitioning may be based on communities in the graph, though it can be changed at will. It may also comprehend some awareness of the signal over the graph. The filterbank's structure is based on the local Fourier modes. However a perspective of the work would be to use more general filters instead of these modes; this could for instance help in reducing the number of analysis channels that can be high if some subgraphs are large.

6. REFERENCES

[1] D. Shuman, S. 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," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] G. Strang and T. Nguyen, *Wavelets and filter banks*. SIAM, 1996.

[3] S. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *Signal Processing, IEEE Transactions on*, vol. 60, no. 6, pp. 2786–2799, June 2012.

[4] ——, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *Signal Processing, IEEE Transactions on*, vol. 61, no. 19, pp. 4673–4685, Oct 2013.

[5] A. Sakiyama and Y. Tanaka, "Oversampled graph Laplacian matrix for graph filter banks," *Signal Processing, IEEE Transactions on*, vol. 62, no. 24, pp. 6425–6437, Dec 2014.

[6] H. Nguyen and M. Do, "Downsampling of signals on graphs via maximum spanning trees," *Signal Processing, IEEE Transactions on*, vol. 63, no. 1, pp. 182–191, Jan 2015.

[7] D. I. Shuman, M. J. Faraji, and P. Vandergheynst, "A framework for multiscale transforms on graphs," *CoRR*, vol. abs/1308.4942, 2013. [Online]. Available: http://arxiv.org/abs/1308.4942

[8] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998. [Online]. Available: http://dx.doi.org/10.1137/S1064827595287997

[9] S.-H. Teng, "Coarsening, sampling, and smoothing: Elements of the multilevel method," in *Algorithms for Parallel Processing*, ser. The IMA Volumes in Mathematics and its Applications. Springer New York, 1999, vol. 105, pp. 247–276.

[10] R. Band, I. Oren, and U. Smilansky, "Nodal domains on graphs - how to count them and why?" *Analysis on Graphs and its applications Proc. Symp. Pure Math.*, 2008.

[11] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[12] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *Signal Processing, IEEE Transactions on*, vol. 62, no. 20, pp. 5227–5239, Oct 2014.

[13] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[14] M. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, no. 23, p. 8577, 2006.

[15] S. Narang, Y. H. Chao, and A. Ortega, "Graph-wavelet filterbanks for edge-aware image processing," in *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, Aug 2012, pp. 141–144.