Lifting Scheme on Graphs with Application to Image Representation

Moncef Hidane, Olivier Lézoray, Abderrahim Elmoataz

Normandie Univ., UNICAEN, ENSICAEN, GREYC UMR CNRS 6072, Caen, France

Abstract—We propose a new multiscale transform for scalar functions defined on the vertex set of a general undirected weighted graph. The transform is based on an adaption of the lifting scheme to graphs. One of the difficulties in applying directly the lifting scheme to graphs is the partitioning of the vertex set. We follow a recent greedy approach and extend it to a multilevel transform. We carefully examine each step of the algorithm, in particular its effect on the underlying basis. We finally investigate the use of the proposed transform to image representation by computing M-term nonlinear approximation errors. We provide a comparison with standard orthogonal and biorthogonal wavelet transforms.

I. INTRODUCTION

The continuous development in data sensing, gathering and simulation has led to an ever growing volume and variety of collected data sets. Mining these data sets, including denoising, clustering and regression is a major challenge. Formally, a data collection corresponds to a set $V = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, where each $\boldsymbol{x}_i$ belongs to $\mathbb{R}^p$, $p \geq 1$. The present paper is concerned with the *representation*, through multiscale transforms, of functions defined on the vertex set of a general weighted graph, referred to as *graph-signals* [1]. We will focus in particular on the case of digital images which we view as graph-signals.

In the context of image processing and analysis, the weighted similarity graph model has become prevalent as it allows to explicitly model self-recursions usually present in natural images. Viewing an image as a function on an similarity graph leads to so-called *nonlocal* graph-based methods [2], [3]. Multiscale transforms allow to capture the local correlations present in the underlying signals and yield highly compressible representations [4]. The effectiveness of this approach motivates the introduction of similar approaches for signals defined on general weighted graphs.

Classical wavelet transforms are obtained by computing, through fast algorithms, inner products with a family of signals obtained by dilating and translating a single mother wavelet [4]. The direct adaption of this approach to graph-signals poses obvious problems relating to the definition of intrinsic notions of translation and dilation on a graph. These difficulties have led to a series of recent works [5]–[8], each leading to a new multiscale transform.

We propose in this paper a new multiscale transform for signals defined on the vertex set of a general graph. The approach we propose uses the lifting scheme [9] and is inspired by the greedy variant proposed in [8]. We note that lifting-based transforms on graphs have also been propsosed in [10]

and [11]. We extend the approach of [8] to a multiple-level transform and examine its compression capability for digital images.

II. LIFTING SCHEME

A. Lifting Scheme for 1D Signals

The lifting scheme [9] allows to factorize orthogonal and biorthogonal wavelet transforms into elementary steps [12]. Let us quickly illustrate the principle for a one-dimensional signal $\boldsymbol{x} = (x_1, \ldots, x_N)$. To apply the forward lifting transform to $\boldsymbol{x}$, one first needs to *partition* the index set $\{1, \ldots, N\}$ into two sets I and J. Here, in the one-dimensional case, we take I to be the set of odd indices and J to be the set of even indices. Let us denote $\boldsymbol{x}^I = (x_i, i \in I)$, and $\boldsymbol{x}^J = (x_j, j \in J)$. A single lifting step consists of two operations: prediction and update. In the first operation, one computes an *approximation* $\tilde{\boldsymbol{x}}^I = P(\boldsymbol{x}^J)$ of the subsignal $\boldsymbol{x}^I$, and then evaluates the *prediction error*: $(\boldsymbol{x}^I)^* = \boldsymbol{x}^I - \tilde{\boldsymbol{x}}^I$. In the sequel, we will call $(\boldsymbol{x}^I)^*$ the *detail signal* as it captures the *spatial* correlation that might be present in $\boldsymbol{x}$.

In the second operation, one computes a coarse approximation of the subsignal $\boldsymbol{x}$ by adding to $\boldsymbol{x}^J$ a linear transformation $U : \mathbb{R}^{|I|} \rightarrow \mathbb{R}^{|J|}$ of $(\boldsymbol{x}^I)^*$. We then get the subsignal $(\boldsymbol{x}^J)^* = \boldsymbol{x}^J + U\left((\boldsymbol{x}^I)^*\right)$. Depending on the precise choice of the operator U, this second operation aims at producing a coarse approximation $(\boldsymbol{x}^J)^*$ whose first k moments agree with those of $\boldsymbol{x}$.

Globally, a single-step lifting transform corresponds to a bijective application $T : \boldsymbol{x} \in \mathbb{R}^N \rightarrow (\boldsymbol{x})^* \in \mathbb{R}^N$ whose inverse can be obtained immediately:

$$\boldsymbol{x}^J = (\boldsymbol{x}^J)^* - U\left((\boldsymbol{x}^I)^*\right), \quad \boldsymbol{x}^I = (\boldsymbol{x}^I)^* + \tilde{\boldsymbol{x}}^I. \quad (1)$$

A multiple-step transform can be obtained by recursively applying the same principle to the successive detail signals.

B. Lifting on Graphs

1) Notations: We begin by briefly fixing the definitions and notations we use. A weighted graph is a triple $G = (V(G), E(G), w)$, where $V(G)$ is a vertex set, $E(G) \subset V(G) \times V(G)$ the set of edges and $w : V(G) \rightarrow]0, +\infty[$ is a weight function. We restrict ourselves in all the rest of this paper to undirected graphs with no self loops. For $i, j \in V(G)$, we write $i \sim j$ if there is an edge joining i and j. If $V(G)$ is fixed and has N vertices, G is characterized by its weighted adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ given by $w_{i,j} = w(i, j)$ if

$(i, j) \in E$ and $w_{i,j} = 0$ otherwise. We denote the set of scalar signals on $V(G)$ by $\ell_2(V(G))$. The elements of $\ell_2(V(G))$ are written in bold. Finally the vector $\mathbf{1}$ denotes the vector whose entries are all equal to one.

2) Algorithm adaption: Consider a graph-signal $\boldsymbol{f} \in \ell_2(V(G))$. In order to apply the lifting scheme to $\boldsymbol{f}$ we need to specify three components: the partitioning of the vertices and the prediction and update operators. We begin with the choice of the operators and assume that we have found two subsets I and J such that $V(G) = I \cup J$, and $I \cap J = \emptyset$.

For the prediction step, we restrict ourselves to linear operators. In practice, we build the weighted adjacency matrix from the input data such that the weight associated with two data instances represents a *similarity measure* between them. In other other words, the graphs we consider here are similarity graphs. It is thus natural to use those weights as prediction coefficients. Applying a prediction step in this manner amounts to multiplying by the following prediction matrix $\boldsymbol{P}$ verifying

$$(\forall j \in J)\,(\forall l \in V(G)) \quad p_{j,l} = \delta_{j,l} = \begin{cases} 1 & \text{if } j = l, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

and

$$(\forall i \in I)\,(\forall l \in V(G)) \quad p_{i,l} = \begin{cases} \frac{w_{i,l}}{\sum_{\substack{j \in J \\ j \sim i}} w_{i,j}} & \text{if } l \in J \text{ and } l \sim i \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Evaluating the prediction error thus amounts to multiplying by $\boldsymbol{I} - \boldsymbol{P}$.

In the same manner, the update step can be written with an *update matrix* $\boldsymbol{U} \in \mathbb{R}^{N \times N}$ verifying

$$(\forall i \in I)\,(\forall l \in V(G)) \quad u_{i,l} = \delta_{i,l}, \quad (4)$$

and

$$(\forall j \in J)\,(\forall l \in V(G)) \quad u_{j,l} \neq 0 \implies l \sim j \text{ and } l \in I. \quad (5)$$

We will come back to the precise choice the matrix $\boldsymbol{U}$ in the next section.

A single-step lifting transform is then represented by the matrix $\boldsymbol{T} \in \mathbb{R}^{N \times N}$ given by $\boldsymbol{T} = (\boldsymbol{I} + \boldsymbol{U})\,(\boldsymbol{I} - \boldsymbol{P})$. Let us denote $\boldsymbol{g} = \boldsymbol{T}\boldsymbol{f}$. The components $(\boldsymbol{g}_i)_{i \in I}$ correspond to *detail* or *wavelet* coefficients while the components $(\boldsymbol{g}_j)_{j \in J}$ correspond to *scale* coefficients.

More generally, many prediction/update steps can be performed, leading to the following single-step transform $\boldsymbol{T} = \prod_{m=1}^{k} (\boldsymbol{I} + \boldsymbol{U}_m)\,(\boldsymbol{I} - \boldsymbol{P}_m)$. If we define a (weighted) adjacency relation on J, one can iterate this transform on the detail signal.

3) Difficulties: We identify at this point a first difficulty concerning the application of the lifting algorithm on general graphs. This difficulty concerns the partitioning step, which in turn covers two aspects. The first aspect concerns the efficient use the edges of the graph. Indeed, let us note that a single lifting step does not use edges whose endpoints both belong to I or J. In the sequel, we will say that such edges are *contentious*.

A first partitioning criterion can be obtained by seeking a partitioning that minimizes the sum of the weights of contentious edges. This criterion seeks a maximal spanning bipartite subgraph and is thus equivalent to the maximal cut problem. Unfortunately, this combinatorial is NP-hard.

In regard to the previous discussion, we adopt a greedy approach inspired by [8] where a series of *elementary lifting steps* are performed. We elaborate on this approach in the next section.

III. PROPOSED ALGORITHM

The algorithm we propose replaces the global partitioning step described in the previous section by a greedy approach. Its is summarized below:

1: Input: A weighted graph G on $\{1, \ldots N\}$, a signal $\boldsymbol{f} \in \ell_2(V(G))$ and an integer $m \geq 1$.
2: Initialization: $\mathcal{S}_0 := V(G), \mathcal{D}_0 := \emptyset, c_{0,i} := f_i, \forall i \in \mathcal{S}_0$.
3: **for all** $r \in \{1, \ldots, m\}$ **do**
4:     Choose $i_r \in \mathcal{S}_{r-1}$.
5:     $\mathcal{S}_r := \mathcal{S}_{r-1} \backslash \{i_r\}$.
6:     $\mathcal{D}_r := \mathcal{D}_{r-1} \bigcup \{i_r\}$.
7:     Prediction:

$$d_r := c_{r-1,i_r} - \sum_{\substack{k \sim i_r \\ k \in \mathcal{S}_r}} \alpha_{r,k} c_{r-1,k}. \quad (6)$$

8:     Update:

$$(\forall k \in \mathcal{S}_r) \quad c_{r,k} = \begin{cases} c_{r-1,k} + \beta_{r,k} d_r & \text{if } k \sim i_r \\ c_{r-1,k} & \text{otherwise.} \end{cases} \quad (7)$$

9: **end for**

We call an iteration of the main loop an *elementary lifting step*. At the rth elementary lifting step, we select a vertex i_r and compute a unique detail coefficient d_r along with scale coefficients $c_{r,k}$. The detail coefficient d_r corresponds to a prediction error as explained in the previous section. Insights into the precise choice of the coefficients $\beta_{r,k}$ in (7) can be gained when examining the effect of each elementary lifting on the underlying basis. We start by examining each component of the proposed algorithm.

A. Vertices Selection

We propose one criterion for the selection of detail vertices. It consists in simply choosing at each step the vertex with maximum degree among the remaining scale vertices:

$$i_r = \arg\max_{i \in \mathcal{S}_{r-1}} \sum_{j \in \mathcal{S}_{r-1}} w_{i,j}. \quad (8)$$

Intuitively, the prediction error should be small when computed at a vertex whose centrality, here measured by its degree, is high. This choice thus favors the sparsity of the overall transform.

B. Prediction

As in the previous section, we restrict ourselves to *linear* prediction operators whose coefficients are related to edges weights:

$$(\forall k \in \mathcal{S}_r, k \sim i_r) \quad \alpha_{r,k} = \frac{w_{i_r,k}}{\sum_{\substack{l \sim i_r \\ l \in \mathcal{S}_r}} w_{i_r,l}}. \tag{9}$$

This choice is motivated by the fact that the graphs we are dealing with are similarity graphs. The choice of the normalization in (9) seeks to obtain zero detail coefficient for locally constant signals.

We insist on the fact that, although linear with respect to a fixed graph, the prediction operator can be globally nonlinear if the weighted adjacency relation is built from the data at hand.

C. Effect on the Underlying Basis

The transform that we have just described is linear and bijective. It thus corresponds to a *change of basis*. Let $T_r : \mathbb{R}^N \to \mathbb{R}^N$ denote the transform corresponding to the application of r elementary lifting step:

$$(\forall \boldsymbol{f} \in \mathbb{R}^N) \quad (T_r \boldsymbol{f})_l = \begin{cases} c_{r,l} & \text{if } l \in \mathcal{S}_r, \\ d_l & \text{if } l \in \mathcal{D}_r. \end{cases} \tag{10}$$

Let $(\boldsymbol{\phi}_{0,i})_{1 \leq i \leq N}$ denote the canonical basis of $\mathbb{R}^N$ given by $\boldsymbol{\phi}_{0,i}(j) = \delta_{i,j}$. At each step $r \in \{1, \ldots, m\}$, the components of $T_r \boldsymbol{f}$ in the canonical basis correspond to the components of $\boldsymbol{f}$ in a *new multiscale basis* denoted $\mathrm{MSB}_r := \{\boldsymbol{\phi}_{r,i}, \boldsymbol{\psi}_{r,j}\}_{i \in \mathcal{D}_r, j \in \mathcal{D}_r}$:

$$\boldsymbol{f} = \sum_{i \in \mathcal{S}_r} c_{r,i} \boldsymbol{\phi}_{r,i} + \sum_{i \in \mathcal{D}_r} d_i \boldsymbol{\psi}_{r,i}. \tag{11}$$

Due to the linearity of T_r, the basis MSB_r is obtained simply by applying T_r^{-1} to $(\boldsymbol{\phi}_{0,i})_{1 \leq i \leq N}$:

$$\begin{cases} (\forall i \in \mathcal{S}_r) & \boldsymbol{\phi}_{r,i} = T_r^{-1}(\boldsymbol{\phi}_{0,i}), \\ (\forall j \in \mathcal{D}_r) & \boldsymbol{\psi}_{r,j} = T_r^{-1}(\boldsymbol{\phi}_{0,i}). \end{cases} \tag{12}$$

It is then possible to relate the elements of two successive multiscale basis obtained in this way. The relations are the following

$$(\forall i \in \mathcal{S}_r) \quad \boldsymbol{\phi}_{r,i} = \begin{cases} \boldsymbol{\phi}_{r-1,i} + \alpha_{r,i} \boldsymbol{\phi}_{r-1,i_r} & \text{if } i \sim i_r \\ \boldsymbol{\phi}_{r-1,i} & \text{otherwise.} \end{cases} \tag{13}$$

and

$$(\forall i \in \mathcal{D}_r) \quad \boldsymbol{\psi}_{r,i} = \begin{cases} \boldsymbol{\phi}_{r-1,i_r} - \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} \boldsymbol{\phi}_{r,i} & \text{if } i = i_r, \\ \boldsymbol{\psi}_{r-1,i} & \text{otherwise.} \end{cases} \tag{14}$$

D. Update

Equation (14) allows to clarify the role of the update step. Let us denote $I_{r,i} = \langle \boldsymbol{\phi}_{r,i}, \mathbf{1} \rangle$, where $\langle ., . \rangle$ denotes the standard inner product on $\mathbb{R}^N$. Notice that $I_{0,i} = 1$ for all $i \in \{1, \ldots, N\}$. As for traditional wavelet transforms, we seek wavelet signals with a vanishing moment. In our context, this requirement is equivalent to $I_{r-1,i_r} = \sum_{\substack{i \sim i_r \\ i \in \mathcal{S}_r}} \beta_{r,i} I_{r,i}$ for all $r \in \{1, \ldots, m\}$. We choose, as in [8], the minimum Euclidean norm solution, given by $\beta_{r,i} = \frac{I_{r,i} I_{r-1,i_r}}{\sum_{\substack{j \sim i_r \\ j \in \mathcal{S}_r}} I_{r,j}^2}$.

E. Multilevel Lifting Transform

The number of elementary lifting steps is given by the integers m. First, this parameter depends on the graph topology. Indeed, each elementary lifting operation cancels the edges incident to the selected detail vertex. Thus depending on the graph topology, it may happen that a certain number of elementary lifting steps invalidate all the edges of the graph. In this case, no further lifting can be performed and we say that we have performed a single *cycle*.

Let us remark at this point that the strategy of computing detail coefficients at vertices with largest degree may lead to cycles with small m. An alternative strategy can be obtained by dividing by the number of incident edges: $i_r = \arg\max_{i \in \mathcal{S}_{r-1}} \frac{\sum_{j \in \mathcal{S}_{r-1}} w_{i,j}}{\mathrm{Card}(\{j \sim i, j \in \mathcal{S}_{r-1}\})}$. We only experimented in the sequel with the strategy described in (8).

In order to proceed with other cycles, one needs to define a sensible adjacency relation on the remaining scale coefficients $\mathcal{S}_m$. We propose to infer this relation by using powers of $\boldsymbol{W}$. Thus, two vertices $i, j \in \mathcal{S}_m$ will be connected by a new edge if and only if $\boldsymbol{W}_{i,j}^2 \neq 0$. Let us observe at this point that if $\boldsymbol{W}$ is binary, then $\boldsymbol{W}_{i,j}^n$ gives the number of paths of length n connecting i to j.

Proceeding with this approach allows to iterate many lifting cycles. The algorithm can be stopped either when there remains a single scale coefficient or by fixing, *a priori*, the ratio between the overall number of vertices and the number of detail coefficients.

We now describe the application of the proposed algorithm to the approximation of digital images, viewed as signals on weighted graphs.

IV. EXPERIMENTS

We illustrate in this section the capability of the proposed transform to represent digital images viewed as graph-signals. Let $\boldsymbol{f} \in \mathbb{R}^N$ be the vector obtained by concatenating the columns of grayscale image matrix. We associate with each pixel of the image a vertex of the graph. We consider three kind of adjacency relations: unweighted 4-adjacency grid graphs, 8-adjacency grid graphs, 25 nearest-neighbors graphs with respect to the photometric distance $d(i, j) = |f_i - f_j|$. We have assigned Gaussian weights to the edges of the second and third types of graphs: $w_{i,j} = e^{-|f_i - f_j|^2/2\sigma^2}$.

Once a graph is constructed from an image, we consider the grayscale values as a signal on this graph and apply our

Fig. 1. Reconstruction of "Barbara" image from 1024 (top row) and 256 (bottom row) coefficients. Columns from left to right: Haar, Daubechies 4, CDF(2.2), lifting on unweighted 4-adjacency graph, lifting on 8-adjacency graph, lifting on 25 nearest-neighbors graph with Gaussian weights.
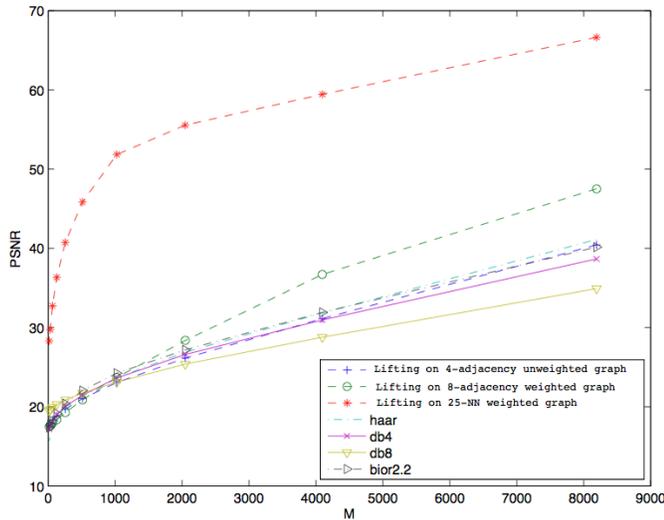

Fig. 2. M-term nonlinear approximation errors for "Barbara". Transforms : lifting on graphs (4 et 8 adjacency, nearest neighbors), Haar, Daubechies 4, Daubechies 8, CDF(2,2).

algorithm to that signal. In order to evaluate the approximation property of the algorithm, we compute M-term nonlinear approximation errors for different values of M For all the experiments, the multilevel transform is iterated until there remains only one scale coefficient.

Figure 2 shows the approximation error on 128x128 standard "Barbara" image for different values of M. In that same figure we also show the results obtained with the Haar, Daubechies 4, Daubechies 8 and CDF(2,2) transforms [4]. We can see that transforms based on 8-adjacency and nearest neighbors graphs achieve very low approximation errors. This is confirmed in Figure 1 which shows the approximations obtained with $M = 1024$ et $M = 256$ terms.

V. CONCLUSION

We have proposed an adaption of the lifting scheme to signals defined on the vertex set of a general weighted graph. We followed a greedy approach suggested in [8] and proposed

to extend it to a multilevel transform. We have also carefully examined each step of the proposed algorithm, in particular its effect on the underlying basis. Finally, we have investigated the use of the proposed transform to image approximation by computing M-term nonlinear approximation errors.

The transform we proposed achieves very high approximation accuracy at the expense of using an adapted weighted graph data structure. A first line of investigation concerns the trade off between the sparsity of the adjacency matrix of the underlying graph and the approximation capability of our transform. Another line of investigation concerns the use of our transform as a prior for solving inverse problems involving graph-signals.

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] A. Elmoataz, O. Lézoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing," *IEEE T. Image Process.*, vol. 17, no. 7, pp. 1047–1060, jul 2008.

[3] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model Sim*, vol. 7, no. 3, pp. 1005–1028, 2008.

[4] S. Mallat, *A Wavelet Tour of Signal Processing, the Sparse way*. Academic Press, 2009.

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

[6] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied Comput. Harmon. A.*, vol. 30, no. 2, pp. 129–150, 2011.

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

[8] M. Jansen, G. Nason, and B. Silverman, "Multiscale methods for data on graphs and irregular multidimensional situations," *J. Roy. Stat. Soc. B*, vol. 71, no. 1, pp. 97–125, 2009.

[9] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.

[10] G. Shen, "Lifting transforms on graphs: Theory and applications," Ph.D. dissertation, UNIVERSITY OF SOUTHERN CALIFORNIA, 2010.

[11] S. K. Narang and A. Ortega, "Lifting based wavelet transforms on graphs," in *Proceedings: APSIPA ASC 2009*, 2009, pp. 441–444.

[12] P. Schröder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," in *ACM SIGGRAPH*, 1995, pp. 161–172.