# Lifting Based Wavelet Transforms on Graphs

Sunil K. Narang and Antonio Ortega
Department of Electrical Engineering
University of Southern California (USC)
Los Angeles, California
E-mail: kumarsun@usc.edu, antonio.ortega@sipi.usc.edu

***Abstract*—We present a novel method to implement lifting based wavelet transforms on general graphs. The detail and approximation coefficients computed from this graph transform can be interpreted similarly to their counterparts in standard signal processing process. Our approach is based on partitioning all nodes in the graph into two sets, containing "even" and "odd" nodes, respectively. Then, as in standard lifting, nodes of one parity are used to predict/update those of the other. We discuss the even-odd assignment problem on the graph and provide a solution that is well suited to construct the transform. As an example we discuss how our transform could be used in a denoising application.**

## I. Introduction

In this paper our goal is to analyze graph structured data using local invertible transforms. Graphs, particularly labeled graphs arise naturally in data-mining, biology [1], network analysis [2],[3],[4] and social studies [5]. We are concerned with graphs ranging from planar acyclic graphs such as trees to more general multi-dimensional non-acyclic graphs such as social networks, Internet etc.

Transform techniques for graph analysis can be broadly be divided into a) global methods, e.g., those using concepts of graph spectral theory, and b) local methods, which exploit correlations in a local neighborhood. Global methods are often based on the Laplacian matrix, whose eigenvalues and eigenvectors contain global information about the shape of the graph. Major applications of global methods include, graph partitioning [5], simplification and graph based feature extraction [6],[7]. A comprehensive discussion of global methods can be found in [8] and [9]. While global methods are widely used, they are highly sensitive to changes in graph structures. For example the Laplacian matrix may have very different eigenvalues and eigenvectors, even when the corresponding graphs have similar structure. In addition to uncovering mostly global information, global methods do not scale well as the graph size increases, e.g., the time required to perform the eigenvalue decomposition can be significant. Therefore there is a need for practical scalable algorithms that can capture both local and global patterns in the graph data and are robust to small changes in the graph structures. Local methods on the other hand exploit local similarities in a graph. They are distributed and locally generated resulting in less computational complexity. Unlike global methods, changes in a local region of the graph only affect the coefficients in that region and do not alter the overall results. In spite of these advantages, very limited work has been reported in this area.

While wavelet-based techniques would seem well suited to provide efficient local analysis, a major obstacle to their application to graphs is that these, unlike images, are not regularly structured. For example, discrete wavelet transforms use local filtering operations followed by downsampling. In a graph, locality can be defined, e.g., by considering the one-hop neighborhood of a node (the set of nodes directly connected to it), but there is no obvious way to downsample in a regular manner, since these neighborhoods vary in size and orientation. Recently there have been proposals to create wavelet transforms for data on graphs. Crovella and Kolaczyk [2] proposed wavelet-like basis functions $\psi_{jk}$ for graphs which are localized w.r.t. a range of location/scale indices, but their transform is not invertible in general. Wang and Ramchandran [4] have proposed graph dependent basis functions for sensor network graphs. These basis functions are locally supported but their dual basis are not. Hence this method cannot be called a locally invertible transform. Shen and Ortega [3] have applied wavelet lifting transforms on spanning trees for a wireless sensor network application, where invertibility is guaranteed for any tree, as long as nodes in the tree are partitioned into two sets (even and odd nodes) and the transform is structured by modifying even nodes based on odd nodes (and vice versa). The starting point for our work is the observation that the idea in [3] can be extended to arbitrary graphs, no longer constrained to be planar and acyclic, as long as suitable even/odd assignment algorithms on the graph can be identified. In Section II we define these novel lifting transforms. Our experiments in Section III provide promising preliminary results using these transforms on a simple denoising task.

## II. Lifting Transform on Graph

Wavelet transforms have been widely used as a signal processing tool for a sparse representation of signals. Wavelet based transforms split the sample space into an approximation and a detail subspace. The approximation subspace contains a smoother version of the original signal and the details of the signal are contained in the detail subspace. Crovella and Kolacyzk [2], apply a discretized wavelet like transform on graphs for anomaly detection. Regions around each node are segmented into disks such that a k-hop disk contains nodes which are exactly k-hop distant from the root node. The wavelets centered at each node are assigned positive weights for even-hop disks and negative weights for odd-hop disks. This transform can be applied to general graphs, but is non-invertible in general, making this transform unsuitable for certain applications such as compression and denoising. Additionally these wavelet transforms use average data of all the nodes situated on the same disk around the root node. For large dataset the k-hop disk size can grow rapidly with increasing k and resulting in loss of locality of the transform.

Our goal is to achieve invertibility with a local transform. Shen and Ortega [3] design a unidirectional 2D lifting transform along arbitrary trees in a wireless sensor network application. Given a tree graph, the authors split the nodes into even and odd nodes based on their minimum hopping distance from the root node (see the tree defined by solid lines in Figure 1 as an example). A lifting transform is then applied locally on the tree using these assignments. Since trees are acyclic planar graphs, the even-odd assignment of nodes is well-defined and no pair of directly connected nodes is assigned identical (even/odd) parity. To apply this idea to arbitrary graphs (in general cyclic and non-planar) would require selecting an even-odd assignment on these graphs. Referring again to Figure 1 if we now consider a graph that includes both solid and dashed lines (planar but cyclic) it can be seen that nodes that are neighbors in the graph are no longer guaranteed to have opposite parity (e.g., 4 is even and connected to 3 and 5 which are both even as well).

Since a lifting-based transform uses information from even (resp. odd) nodes in order to predict (update) an odd (resp. even) node, having neighboring nodes with same parity means that some local information cannot be used (e.g., we cannot use information in all neighbors to predict information in a given node).

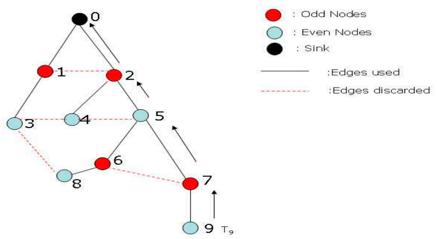



Fig. 1. Even Odd Assignment in routing trees designed in [3].The dashed lines show the edges not used by the transform though they are within radio-range

Thus in order to apply a lifting-based transform to an arbitrary graph we would like to split the nodes $\mathcal{V}$ in the graph into even and odd sets. Using the notations of [3], starting from $j = 1$, at each scale $j$ of the transform the set of nodes $\mathcal{U}_{j-1}$ are first split into a set of 'even' nodes $\mathcal{U}_j$ and 'odd' nodes $\mathcal{P}_j$ . The coefficients in odd nodes $\mathcal{P}_j$ denoted $d_{j,m}(m \in \mathcal{P}_j)$ are then predicted from the coefficients in $\mathcal{U}_j$ denoted $s_{j-1,n}(n \in \mathcal{U}_j)$ by applying prediction step of lifting. The coefficients in even set of nodes $\mathcal{U}_j$ are then updated to $s_{j,n}$ using $d_{j,m}$. For next level of transform set $\mathcal{U}_j$ is again split into the sets $\mathcal{U}_{j+1}$ and $\mathcal{P}_{j+1}$ and similar steps of lifting are applied. Thus after any such $k$ decompositions the number of coefficients equals number of original samples $s_{0,n}(n \in \mathcal{U}_0 \equiv \mathcal{V})$ and completely describe them, making the entire process reversible.

While any split will guarantee invertibility, we seek techniques to split (i.e., to label or color) the graph that minimize the number of conflicts (i.e., the percentage of direct neighbors in the graph that have same parity). This is then a bipartite subgraph problem, where the goal is to split a graph into two clusters (even/odd) so as to minimize the number of removed edges (only edges connecting nodes within a cluster are removed). The problem is NP-hard in general and for a completely random even-odd assignment of nodes, the probability of an edge having same parity on both its ends is roughly $50\%$. Hence with a random assignment almost $50\%$ edges are not utilized in the transform. In the next section we describe a greedy method to approximate good even-odd split.

### A. Even-Odd splitting of Graph

Assume an algorithm assigns a label (even/odd) to each vertex of a graph $G = (V, E)$ of size $N$ with adjacency matrix $\mathbf{Adj}$ such that there are $m$ odd labels and $N - m = l$ even labels. If we rearrange the vector $\mathbf{v}$ of vertices to gather even and odd vertices at one place and rearrange the adjacency matrix accordingly, we have

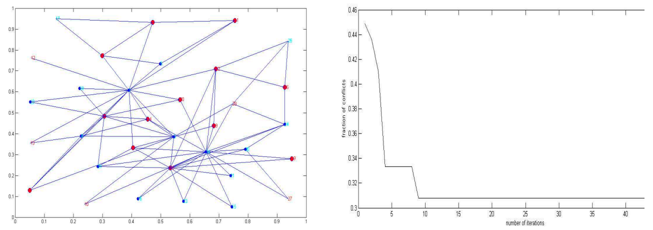
$$\widetilde{\mathbf{v}} = \begin{pmatrix} \mathbf{v_{odd}} \\ \mathbf{v_{even}} \end{pmatrix} \qquad \widetilde{\mathbf{Adj}} = \begin{pmatrix} \mathbf{F_{m\times m}} & \mathbf{J_{m\times l}} \\ \mathbf{K_{l\times m}} & \mathbf{L_{l\times l}} \end{pmatrix} \tag{1}$$

where $\mathbf{v_{odd}}$ is a $m \times 1$ array and $\mathbf{v_{even}}$ is a $l \times 1$ array. The submatrix $\mathbf{F}$ of $\widetilde{\mathbf{Adj}}$ is adjacency matrix of a subgraph containing odd nodes only. Similarly $\mathbf{L}$ is a submatrix of a subgraph having even nodes only. These matrices contain edges which have conflicts since they connect nodes of same parity. The block matrices $\mathbf{J}$ and $\mathbf{K}$ contain edges which do not have conflicts. A lifting transform based on this even-odd assignment utilizes only the $\mathbf{J}$ and $\mathbf{K}$ matrices of adjacency matrix. $\mathbf{F}$ and $\mathbf{L}$ matrices are considered non-existent. So any quality criteria for the even-odd assignment should be based on minimizing edge information present in matrices $\mathbf{F}$ and $\mathbf{L}$. We propose one such criterion of minimizing the row sum of adjacency matrices $\mathbf{F}$ and $\mathbf{L}$.

For this purpose we use an algorithm called conservative fixed probability colorer (CFP) given in [10]. The CFP colorer algorithm solves the corresponding problem of 2 colors graph coloring problem (2-GCP) so as to minimize the conflicts. This algorithm is based on a simple greedy local heuristics and gives competitive results as compared to other k-GCP algorithms [10]. The algorithm is iterative and at each iteration few randomly chosen nodes are activated. Each activated node counts the number of conflicting edges with its neighbors and changes its parity based on the conflict. In a more conservative approach, the parity change happens sequentially in each iteration. Formally the algorithm is presented in algorithm 1. Figure 2(a) shows a sample even-odd assignment of Karate Data [11] and Figure 2(b) shows the reduction of conflicts with each iteration. The x-axis in Figure 2(b) is number of iterations. The value on y-axis is the fraction of conflicting edges. The convergence of solution has been discussed in [10]. If the solution converges, it ensures in probability that there are no nodes having more than $50\%$ neighbors of same parity. The algorithm can also be extended to weighted edge graphs.

**Algorithm 1** Even-Odd Assignment Algorithm

Randomly assign initial label to each node
**for** k = (1:1:max_iter) **do**
  Activate each node randomly with a fixed uniform probability.
  For each activated node choose a parity that minimizes its conflict with neighboring nodes
  Inform the neighboring nodes, if the parity is changed.
**end for**



(a) Even-Odd assignment on Graph (b) convergence of CFP algorithm

Fig. 2. Even-Odd assignment on Zachary Karate Data [11] using even-odd algorithm 1.

*B. Graph Transform design*

Once we have a disjoint set of even-odd assignment of nodes in the graph, we can perform a lifting wavelet transform, as given in Equation (2).

$$\begin{aligned} \mathbf{D_1^1} &= \mathbf{X_{odd}} - \mathbf{J_P} \times \mathbf{X_{even}} \\ \mathbf{S_1^1} &= \mathbf{X_{even}} + \mathbf{K_U} \times \mathbf{D_1^1} \end{aligned} \tag{2}$$

where matrix $\mathbf{J_p}$ is prediction matrix computed from matrix $\mathbf{J}$ of Equation 1 by multiplying each row with prediction weights. Similarly $\mathbf{K_u}$ is update matrix, computed from matrix $\mathbf{K}$ by multiplying each row with update weights. This transform is invertible and the original values can be recovered by following inverse lifting steps given in Equation (3)

$$\begin{aligned} \mathbf{X_{even}} &= \mathbf{S_1^1} - \mathbf{K_U} \times \mathbf{D_1^1} \\ \mathbf{X_{odd}} &= \mathbf{D_1^1} + \mathbf{J_P} \times \mathbf{X_{even}} \end{aligned} \tag{3}$$

The prediction and update weights depend on the type of application we choose. For the denoising example in the result section, we use prediction and update weights similar to the ones designed in lifting transform given in [3]. The prediction weight for row $i$ of matrix $\mathbf{J}$ is $p_i = (\sum_{j=1}^{m} \mathbf{J}(i,j)+1)^{-1}$ and the update weight for row $i$ of matrix $\mathbf{K}$ is $u_i = (2(\sum_{j=1}^{l} \mathbf{K}(i,j) + 1))^{-1}$. Thus, in our implementation of lifting, first the data on odd nodes are subtracted from a weighted sum of even parity neighboring nodes to obtain detail coefficients.The even nodes then update their data by adding a weighted sum of detail values obtained in the previous step from their odd parity neighboring nodes. This gives us a critically sampled invertible transform. In some application when we want over-sampled transforms on the graph, we swap the parity of even and odd nodes. In this case, each node has one detail coefficient and one update coefficient value. Original data values do not have to be stored. The block diagram of an oversampled lifting transform is given in Figure 3.

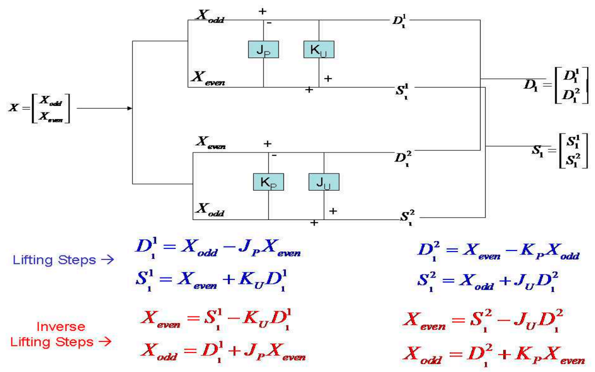



Fig. 3. Block diagram for an oversampled lifting transform and lifting transform equations.

## III. Experiments and Results

We address a simple graph denoising application to demonstrate the advantages of our invertible graph transform. Graph denoising mayb be applied as a preprocessing tool in analyzing real world graphs, e.g., protein interaction networks [12]. The toy graphs of our experiment are similarity graphs (see [9], Section 2.2) with $N$ uniformly sampled nodes from two partially overlapping Gaussian distributions. An edge $\{i, j\}$ between two vertices in the graph exists if the difference in the corresponding sample values is less than some threshold. An example graph with $N = 200$ sample values is shown in Figure 4(a). Figures 4(b)-(f) show Voronoi tessellations of the distribution field with $N = 1500$ sampled points as Voronoi sites. For Figure 4(b) the value of each sample is the mean of the distribution from which it is drawn. In Figure 4(c) sample values are the actual noisy values. The intensity of each cell reflects the value of corresponding sample in the cell rescaled to the range between $[0, 1]$. Figure 4 (d),(e),(f) are the Voronoi tessellations of denoised samples. This problem can be seen as a 2D version of denoising of a general M-dimensional discrete data. While our results are preliminary they demonstrate promising performance as compared to simple, single-step methods operating on the Laplacian matrix that have been proposed in the literature. Wavelet denoising is done by transforming noisy data into the wavelet domain, applying thresholding in the wavelet domain, and inverse transforming the denoised wavelet coefficients. In this work, the wavelet coefficients are prediction coefficients obtained by applying the proposed lifting based transform on the graph. For thresholding we apply universal threshold given by Donoho [13] $thr = \sqrt{2\log_2(N)}$ on the wavelet coefficients normalized to the noise level [14]. We compare our results to both short time and long time solutions of the diffusion heat equation ([8], [15]) on the graphs . The Voronoi tessellations of the field constructed from denoised values of the samples are drawn in Figure 4(c)-(f). The plots show that lifting transform based denoising results are closer to original distribution in Figure 4(b) than diffusion based methods. To quantitatively assess these results we use two

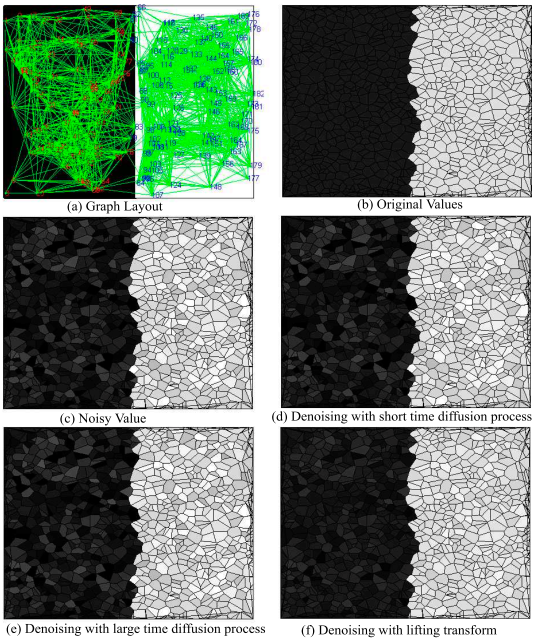



Fig. 4. (a)Similarity graph with 200 sampled points from the underlying distribution.The nodes in shaded region are $\mathcal{N}(\mu_1, \sigma^2)$ and the nodes in white region are $\mathcal{N}(\mu_2, \sigma^2)$ (b)-(f) Voronoi Plots

quality metrics: peak signal to noise ratio (PSNR) and standard deviation(STD) of samples. Results are in Figure 6 and 5. As can be seen in Figure 6, PSNR achieved in lifting is higher than for diffusion based methods, with better results achieved with the oversampled approach. Note that gains from oversampling are only significant for relatively sparse graphs. In Figure 5 we can observe reduction STD with respect to the original signal and, here too, we observe STD of oversampled transform to be lower than STD in critically sampled case.

## IV. Conclusions and Future Work

We propose a lifting based wavelet transform which can be applied to arbitrary graphs. We define a very new way of applying signal processing tools on graph based data with this transform. In future, we would want to improve our even-odd assignment algorithms, and extend the idea to a multi-level lifting transform.

## References

[1] M. Weber and S. Kube, "Robust perron cluster analysis for various applications in computational life science," in *CompLife*, 2005, pp. 57–66.

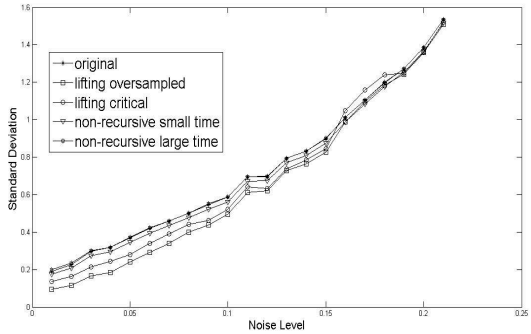



Fig. 5. STD of the original and denoised samples

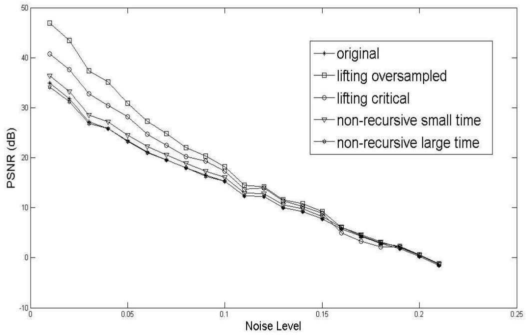



Fig. 6. PSNR of the original and denoised samples

[2] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *INFOCOM 2003*, Mar 2003, vol. 3, pp. 1848–1857.
[3] G. Shen and A. Ortega, "Optimized distributed 2D transforms for irregularly sampled sensor network grids using wavelet lifting," in *ICASSP'08*, April 2008, pp. 2513–2516.
[4] W. Wang and K. Ramchandran, "Random multiresolution representations for arbitrary sensor network graphs," in *ICASSP*, May 2006, vol. 4, pp. IV–IV.
[5] M. Girvan and M. E. Newman, "Community structure in social and biological networks.," *Proc Natl Acad Sci U S A*, vol. 99, no. 12, pp. 7821–7826, June 2002.
[6] R.I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *Proc. ICML*, 2002, pp. 315–322.
[7] H. Qiu and E. R. Hancock, "Graph simplification and matching using commute times," *Pattern Recogn.*, vol. 40, no. 10, pp. 2874–2889, 2007.
[8] Fan R. K. Chung, *Spectral Graph Theory (CBMS Regional Conf. Series in Math., No. 92)*, American Mathematical Society, February 1997.
[9] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
[10] S. Fitzpatrick and L. Meertens, "An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs," in *In Proc. SAGA'01*, 2001, pp. 49–64.
[11] W.W. Zachary, "An information flow model for conflict and fission in small groups," *Journ. of Anthropological Research*, vol. 33, pp. 452–473, 1977.
[12] G. Pandey, M. Steinbach, R. Gupta, T. Garg, and V. Kumar, "Association analysis-based transformations for protein interaction networks: a function prediction case study," in *KDD '07*. 2007, pp. 540–549, ACM.
[13] D.L. Donoho, "De-noising by soft-thresholding," *Information Theory, IEEE Trans. on*, vol. 41, no. 3, pp. 613–627, May 1995.
[14] I. M. Johnstone and B. W. Silverman, "Wavelet threshold estimators for data with correlated noise," *Royal Statistical Society: Series B (Statistical Methodology)*, vol. 59, pp. 319–351, 1997.
[15] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recogn.*, vol. 41, no. 11, 2008.