THE GENERALIZED HAAR-WALSH TRANSFORM

Jeff Irion and Naoki Saito

University of California, Davis
Department of Mathematics
One Shields Avenue, Davis, CA 95616

ABSTRACT

We introduce a novel multiscale transform for signals on graphs which is a generalization of the classical Haar and Walsh-Hadamard Transforms. Using a recursive partitioning of the graph and successive averaging and differencing operations, our transform generates an overcomplete dictionary of orthonormal bases. We describe how to adapt the classical best-basis search algorithm to this setting, and show results from preliminary denoising experiments.

Index Terms— Fiedler vectors, spectral graph partitioning, multiscale basis dictionaries, wavelets on graphs

1. INTRODUCTION

A current area of focus in the field of signal processing is the extension of classical techniques on regular domains to the setting of graphs. Examples of such signals on graphs include recordings on sensor networks and flow measurements on networks. In particular, much effort has been devoted to developing wavelet transforms for these signals on graphs, but a key challenge in doing so is that we no longer have tools such as translation and dilation at our disposal. Furthermore, the notion of frequency is not well-defined on graphs, in general. In our previous work [1], we circumvented these difficulties by developing a multiscale transform rather than a true generalization of wavelets; in fact, our transform can be viewed as a generalization of the hierarchical block discrete cosine transforms. The Hierarchical Graph Laplacian Eigen Transform (HGLET), as we named it, utilizes a recursive partitioning of the graph to produce an overcomplete basis dictionary.

In this article, we use the general framework of the HGLET to develop a novel multiscale redundant transform which we call the *Generalized Haar-Walsh Transform (GHWT)*; this is a generalization of the classical Haar Transform and Walsh-Hadamard Transform. Furthermore, we show that for this transform there exists an analog of frequency (or sequency). We also take the best-basis algorithm [2], originally developed for wavelet packets on classical signals, and generalize it to our HGLET/GHWT dictionaries.

This research was partially supported by ONR grant N00014-12-1-0177 and was conducted with Government support under contract FA9550-11-C-0028 and awarded by the Department of Defense, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

2. PRELIMINARIES

Let G be an undirected connected graph. Associated with G we have a set of nodes $V(G)$, and we define $N := |V(G)|$. We also have a set of edges $E(G)$ and edge weights $W(G) = (W_{ij}) \in \mathbb{R}^{N \times N}$. As G is undirected, $W(G)$ is symmetric.

The foundation upon which the GHWT is constructed is a recursive partitioning of the graph, and we now introduce the necessary notation. This partitioning will have multiple levels, and we use $j = 0$ to denote the coarsest level (where the only region, or subgraph, is the entire graph) and $j = j_{\max}$ to denote the finest level (where each region contains a single node). We use K^j to denote the number of regions on level j and we use k to refer to these regions, where $0 \le k < K^j$. We use G_k^j to denote the subgraph formed by restricting to the nodes in region k on level j and to the edges between them, and we define $N_k^j := |V(G_k^j)|$.

We partition a subgraph G_k^j into two subgraphs on level $j+1$ as follows. First, we define the *Laplacian* matrix of a graph G as $L(G) := D(G) - W(G)$, where $D(G)$ is the diagonal degree matrix with entries $d_i = \sum_j W_{ij}$. Alternatively, we may use the *random-walk normalized Laplacian* $L_{\text{rw}}(G) := D(G)^{-1} L(G)$. For both L and L_{rw}, it is known that the smallest eigenvalue is $\lambda_0 = 0$ and that the corresponding eigenvector is $\boldsymbol{\phi}_0 = \mathbf{1}_{V(G)}/\sqrt{N}$; that is, the normalized global indicator vector [3]. For a connected graph, all other eigenvalues are strictly positive. We compute the first nonconstant eigenvector $\boldsymbol{\phi}_1$, also known as the *Fiedler vector*, and partition the subgraph according to the signs of its entries. Justification of such bipartitioning comes from that fact that it yields an approximate minimizer of the bipartitioning criterion called the RatioCut (or the Normalized Cut) when L (or L_{rw}, respectively) is used [3]. When applied to G_k^j, where $N_k^j > 1$, this partitioning method generates two subgraphs, $G_{k'}^{j+1}$ and $G_{k'+1}^{j+1}$ (as the overall partitioning of the graph is not required to be balanced, we have that $k \le k' \le 2k$). We repeat this process until the graph is fully partitioned, meaning that all partitions at the finest level contain a single node. Note that the GHWT

does not require the partitioning to be balanced. Moreover, it accepts any other recursive graph bipartitioning techniques.

3. GENERALIZED HAAR-WALSH TRANSFORM

We now introduce our Generalized Haar-Walsh Transform. As with the HGLET, the two main steps in this transform are to (1) recursively partition the graph and (2) generate a full orthonormal basis on each level of the graph partitioning.

Our notation for the basis vectors is $\boldsymbol{\psi}_{k,l}^{j}$, where $j \in [0, j_{\max}]$ denotes the level, $k \in [0, K^j)$ denotes the index of the region on level j, and $l \in [0, 2^{j_{\max}-j})$ denotes the *tag* of the basis vector. A basis vector's tag is an integer which, when expressed in binary, specifies the sequence of low-frequency (averaging) and high-frequency (differencing) operations that were used to generate it. Within a given region k on level j, the tags are never duplicated, and thus they serve as unique identifiers for the basis vectors within the region. We refer to basis vectors with tag $l = 0$ as *scaling vectors*, those with tag $l = 1$ as *Haar-like vectors*, and those with tag $l \geq 2$ as *Walsh-like vectors*.

The GHWT algorithm can be summarized as follows:

Algorithm 1 (GHWT)

Step 1: Generate a full recursive partitioning of the graph, as described in Section 2. This yields a set of regions G_k^j, with $0 \leq j \leq j_{\max}$ and $0 \leq k < K^j$.

Step 2: Generate an orthonormal basis $\{\boldsymbol{\psi}_{k,0}^{j_{\max}}\}_{0 \leq k < N}$ on level $j_{\max}$. Since each region contains a single node, we simply have $\boldsymbol{\psi}_{k,0}^{j_{\max}} = \mathbf{1}_{V(G_k^{j_{\max}})} \in \mathbb{R}^N$.

Step 3: **For** $j = j_{\max}, \ldots, 1$, use the orthonormal basis on level j to generate an orthonormal basis on level $j-1$ as follows. **For** $k = 0, \ldots, K^{j-1} - 1$:

Step 3a: Compute the scaling vector on G_k^{j-1} as

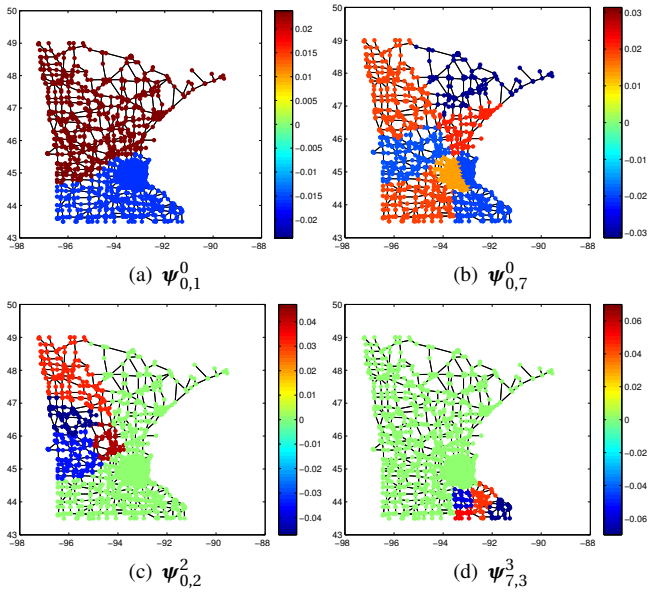
$$\boldsymbol{\psi}_{k,0}^{j-1} := \mathbf{1}_{V(G_k^{j-1})} \Big/ \sqrt{N_k^{j-1}}. \tag{1}$$

Step 3b: **If** $N_k^{j-1} > 1$, **then** compute the Haar-like vector on G_k^{j-1} as

$$\boldsymbol{\psi}_{k,1}^{j-1} := \frac{N_{k'+1}^{j}\sqrt{N_{k'}^{j}}\boldsymbol{\psi}_{k',0}^{j} - N_{k'}^{j}\sqrt{N_{k'+1}^{j}}\boldsymbol{\psi}_{k'+1,0}^{j}}{\sqrt{N_{k'}^{j}\left(N_{k'+1}^{j}\right)^2 + N_{k'+1}^{j}\left(N_{k'}^{j}\right)^2}}, \tag{2}$$

where G_k^{j-1} is split into $G_{k'}^{j}$ and $G_{k'+1}^{j}$.

Step 3c: **If** $N_k^{j-1} > 2$, **then** compute the Walsh-like vectors on G_k^{j-1}. **For** $l = 1, \ldots, 2^{j_{\max}-j} - 1$:



(a) $\boldsymbol{\psi}_{0,1}^{0}$ (b) $\boldsymbol{\psi}_{0,7}^{0}$ (c) $\boldsymbol{\psi}_{0,2}^{2}$ (d) $\boldsymbol{\psi}_{7,3}^{3}$

Fig. 1. GHWT basis vectors on the Minnesota road network ($N = 2636$), where the random-walk normalized Laplacian L_{rw} was used for recursive partitioning. The edge weights of the graph were the inverse of the physical distances between the corresponding nodes.

Case 1: **If** neither subregion has a basis vector with tag l, **then** do nothing.

Case 2: **If** (without loss of generality) only subregion $G_{k'}^{j}$ has a basis vector with tag l, **then** set

$$\boldsymbol{\psi}_{k,2l}^{j-1} := \boldsymbol{\psi}_{k',l}^{j}. \tag{3}$$

Case 3: **If** both subregions have a basis vector with tag l, **then** compute

$$\boldsymbol{\psi}_{k,2l}^{j-1} := \left(\boldsymbol{\psi}_{k',l}^{j} + \boldsymbol{\psi}_{k'+1,l}^{j}\right) \Big/ \sqrt{2} \tag{4}$$

$$\boldsymbol{\psi}_{k,2l+1}^{j-1} := \left(\boldsymbol{\psi}_{k',l}^{j} - \boldsymbol{\psi}_{k'+1,l}^{j}\right) \Big/ \sqrt{2}. \tag{5}$$

The result is an overcomplete dictionary of orthonormal bases, each of which we view as an orthogonal matrix in $\mathbb{R}^{N \times N}$. The overall cost of generating these bases is $O(N^2)$, which is due to forming a dense $N \times N$ matrix via simple arithmetic operations. If we simply wish to expand a signal on the graph (i.e., $f \in \mathbb{R}^N$) in terms of these bases, the cost is $O(N \log N)$. This is done by expanding the signal in terms of the basis on the level $j_{\max}$ (which merely amounts to reordering the signal), and then performing the operations in (1)-(5) on the coefficients rather than the basis vectors. Fig. 1 shows some GHWT basis vectors on the MN road network.

At this point we make several observations about the GHWT. First, note that the GHWT basis vectors on each level are localized; i.e., their support does not extend beyond

the region to which they correspond. This is due to the way in which the basis vectors are defined on the finest level (where each region contains a single node) in Step 2, and to how the basis vectors on regions containing multiple nodes are formed by taking linear combinations of basis vectors on their two subregions in Step 3. Therefore, the bases corresponding to subgraphs G_k^j and $G_{k'}^{j'}$ will be disjoint if $V(G_k^j) \cap V(G_{k'}^{j'}) = \emptyset$. Furthermore, if region G_k^j is divided into subregions $G_{k'}^{j+1}$ and $G_{k'+1}^{j+1}$, then the basis vectors corresponding to G_k^j will span the same space as the union of those corresponding to $G_{k'}^{j+1}$ and $G_{k'+1}^{j+1}$. By making use of these properties, we can select an orthonormal basis containing basis vectors corresponding to multiple levels of the graph partitioning.

To demonstrate these points, we use the simple example of P_6; that is, the unweighted path graph of length six. We group the basis vectors by region and arrange them as in Fig. 2. This illustrates both the manner in which the graph is recursively partitioned and how the span of each block of basis vectors includes the span of all blocks of basis vectors directly beneath it. We refer to this ordering of basis vectors as the *coarse-to-fine dictionary*.

In addition to grouping basis vectors by their corresponding region, we can also group them by their tag, l, and we call this the *fine-to-coarse dictionary*. From Step 3, note that the basis vectors on level j with tag l are used to generate the basis vectors on level $j-1$ with tags $2l$ and $2l+1$. Therefore, the vectors $\{\boldsymbol{\psi}_{k,l}^j\}_k$ span the same space as the vectors $\{\boldsymbol{\psi}_{k,2l}^{j-1}\}_k \cup \{\boldsymbol{\psi}_{k,2l+1}^{j-1}\}_k$. Exploiting this relationship affords us more options for selecting a basis.

Again, we use P_6 as an example. Reversing the order of the levels in Fig. 2 and grouping the basis vectors by tag, we obtain Fig. 3. As with the coarse-to-fine dictionary, the span of each block of basis vectors in the fine-to-coarse dictionary includes the span of all blocks of basis vectors directly beneath it. However, notice that the structure of the groups/blocks differs between the coarse-to-fine and fine-to-coarse dictionaries.

Not only do the tags allow us to regroup the basis vectors, thereby providing more choosable bases, they also impart upon the basis vectors an approximate notion of frequency. From (1), we see that the scaling vectors ($l = 0$) are constant on their support. From (2), we see that the Haar-like vectors ($l = 1$) assume exactly two distinct values on their support. And from (3)-(5), we see that the tags of the Walsh-like vectors ($l \geq 2$) specify the sequence of average and difference operations by which they were generated. Generally speaking, larger l values indicate more oscillation, with exceptions occurring when imbalances in the partitioning necessitate the use of (3), as opposed to (4) and (5).

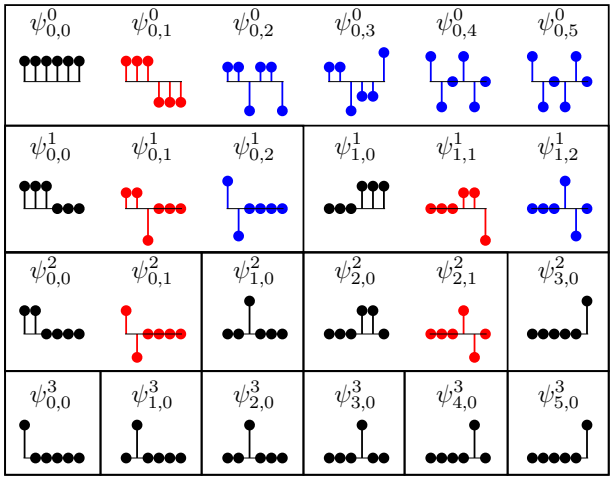


Fig. 2. The coarse-to-fine dictionary for P_6. Scaling, Haar-like, and Walsh-like vectors are in black, red, and blue, respectively. The basis vectors are grouped by region, and thus the structure of the blocks indicates the manner in which the graph is recursively partitioned.

4. BEST-BASIS ALGORITHM

With the GHWT basis vectors arranged in both the coarse-to-fine and fine-to-coarse dictionaries, the best-basis algorithm is a straightforward generalization of [2]. As one of the objectives is to achieve efficient approximation of a given signal on the graph, we begin by specifying a cost functional $\mathscr{J}$ which favors sparse signals. We first perform a search among the coarse-to-fine dictionary. We initialize the best basis as the union of the bases on the bottom level ($j = j_{\max}$) and we proceed upwards one level at a time, using $\mathscr{J}$ to compare the cost of a block of basis vectors to the cost of its descendant basis vectors in the current best basis and updating the best basis when necessary. Upon completion, this search yields the coarse-to-fine best basis. We then search among the fine-to-coarse dictionary, starting at the bottom level ($j = 0$) and proceeding upwards until we have obtained the fine-to-coarse best basis. The final step is to compare the costs of the coarse-to-fine and fine-to-coarse best bases, and the result is the overall best basis for that signal on the graph. By construction, this basis is orthonormal. Furthermore, we can use this same search procedure to find the best basis from among the HGLET dictionary, although in that context there is only one dictionary from which to search. That is, there is no fine-to-coarse HGLET dictionary because the basis vectors on a given level can be grouped only by region, not by tag.

5. DENOISING EXPERIMENTS

Here we illustrate the results of a simple denoising experiment. As our original signal, we used a mutilated Gaussian

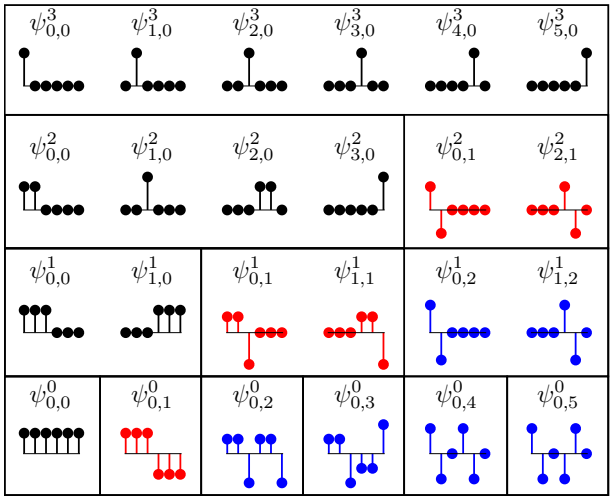


Fig. 3. The fine-to-coarse dictionary for P_6, where basis vectors are grouped by tag.

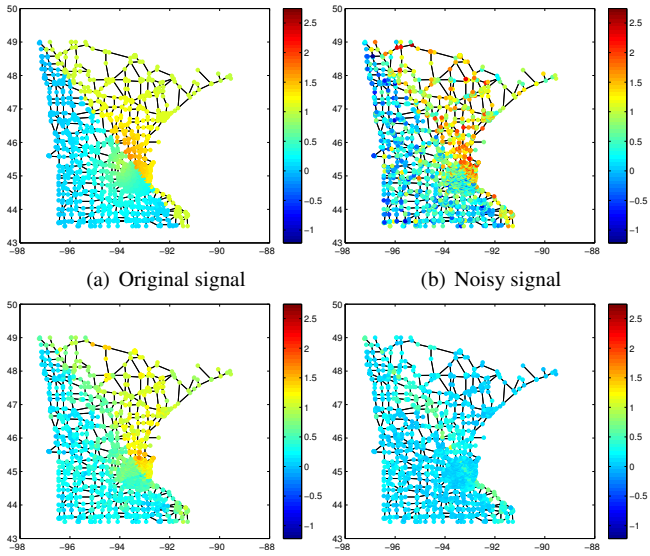


Fig. 4. Signals from our denoising experiment.

function on the MN road network (Fig. 4a). We added white Gaussian noise to the signal so that the SNR became 5.00 dB (Fig. 4b). We ran both the HGLET and GHWT on the noisy signal, using L_{rw} where the edge weights are inverse of the physical distances between nodes. We then ran the best-basis search algorithm on the transformed signals and used soft-thresholding to denoise them. (Note: all scaling coefficients were kept intact.) To be more precise, we sorted the magnitude of the best-basis coefficients in the non-increasing order; used these sorted coefficient magnitudes as the threshold value for soft-thresholding; and searched the threshold value that yielded the highest SNR. The highest SNR of the HGLET best basis was 6.77 dB (48.9% of the coefficients kept) whereas that of the GHWT best basis was 11.56 dB (11.0% of the coefficients were kept; see Fig. 4c,d). It is interesting to note that the GHWT best basis came from the fine-to-coarse dictionary; recall that the HGLET does not have a fine-to-coarse dictionary. This suggests that for the purpose of denoising, grouping the basis vectors by 'frequency' (as in the fine-to-coarse dictionary) seems more effective than grouping them by location (as in the coarse-to-fine and HGLET dictionaries).

6. DISCUSSION

The purpose of this article has been to introduce our novel multiscale transform, the GHWT. While the framework of the GHWT is similar to that of our HGLET [1], the resulting bases are fundamentally different: the HGLET yields basis vectors that are smooth on their support, whereas the GHWT produces basis vectors that are piecewise constant on their support. Furthermore, by construction the GHWT basis vectors are equipped with a notion of 'frequency' which is reflected in their tags, l. In contrast, the HGLET basis vectors are formulated by extending Laplacian eigenvectors on subgraphs to the entire graph and therefore, as shown in [4], we cannot make definitive statements about their 'frequency.' Another key difference is that the GHWT provides not only the coarse-to-fine dictionary but also the fine-to-coarse dictionary for a given graph.

It is worth mentioning that a number of researchers have developed generalizations of the Haar Transform to this graph setting; see [1] and the references therein. The GHWT is more general in that each of these Haar-like transforms generate a single basis, whereas the GHWT generates a *dictionary* of bases, with the generalized Haar basis as one choosable option. And by using the best-basis algorithm, we are guaranteed to obtain a basis that (according to the specified cost functional $\mathscr{J}$) is at least as good as the generalized Haar basis for the task at hand.

7. REFERENCES

[1] J. Irion and N. Saito, "Hierarchical graph Laplacian eigen transforms," *JSIAM Letters*, 2014, to appear.

[2] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 713–718, 1992.

[3] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[4] Y. Nakatsukasa, N. Saito, and E. Woei, "Mysteries around the graph Laplacian eigenvalue 4," *Linear Algebra Appl.*, vol. 438, no. 8, pp. 3231–3246, 2013.