MAT 280: Harmonic Analysis on Graphs & Networks
Lecture 20: Wavelets on Graphs V
The Generalized Haar-Walsh Transform (GHWT) and Its Extension eGHWT

Naoki Saito

Department of Mathematics
University of California, Davis

December 5, 2019

Outline

Acknowledgment

- Jeff Irion (formerly at UC Davis, currently at Neato Robotics, Inc.)
- Yiqun Shao (UC Davis)
- The Science News dataset provided by Jeff Solka (George Mason Univ.) via Raphy Coifman (Yale) and Matan Gavish (Hebrew Univ.)
- NSF Grants: DMS-1418779, IIS-1631329, DMS-1912747, CCF-1934568
- ONR Grants: N00014-16-1-2255

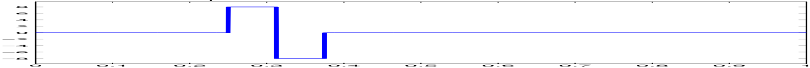
- This lecture is based on the following papers:

  - J. Irion & N. Saito: "The generalized Haar-Walsh transform," *Proc. 2014 IEEE Workshop on Statistical Signal Processing*, pp.472–475, 2014.
  - J. Irion & N. Saito: "Applied and computational harmonic analysis on graphs and networks," in *Wavelets and Sparsity XVI, Proc. SPIE 9597*, Paper # 95971F, 2015.
  - Y. Shao & N. Saito: "The extended Generalized Haar-Walsh Transform and applications," in *Wavelets and Sparsity XVIII, Proc. SPIE 11138*, Paper #111380C, 2019.

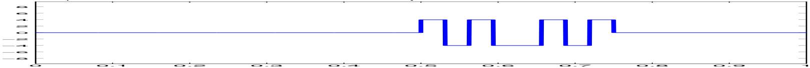
Outline

Motivations: Wavelets and Wavelet Packets

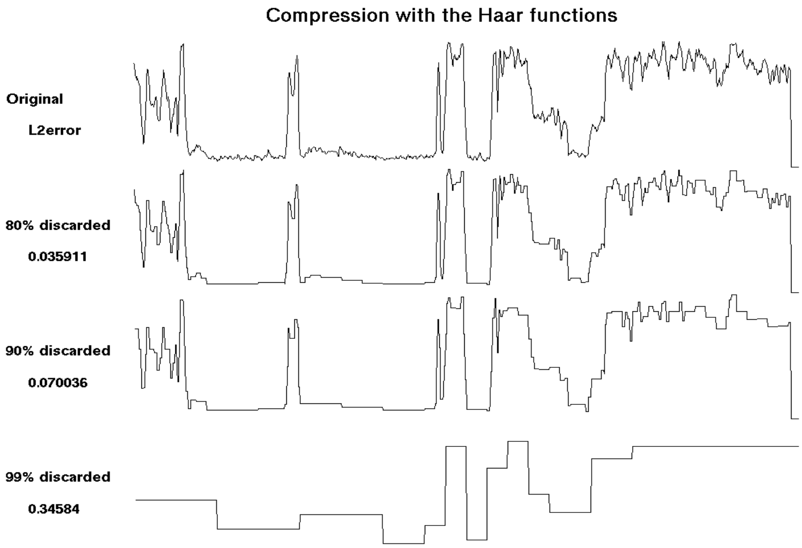
- For usual signals and images, *wavelet transforms* and their generalization *wavelet packet transforms* have a proven track record of success, e.g., JPEG *2000* Image Compression Standard.
- The *Haar* wavelet transform is the simplest among them; it decomposes a given signal into *translations* and *dilations* of a difference of *blocky* functions.



- The *Walsh* transform decomposes a given signal into more oscillatory global square waves.
- The *Haar-Walsh* wavelet packet transform decomposes a given signal into all sorts of local, global, and/or oscillatory blocky functions (hence, it is a *redundant* transform).



Motivations: Wavelets and Wavelet Packets . . .



Motivations: Lifting the Haar-Walsh Wavelet Packets to Graphs

- Want to lift the Haar-Walsh wavelet packet transform to the graph setting
- The Haar-Walsh wavelet packet transform is the most amenable to graphs and networks among all the wavelets and wavelet packets family due to its operational simplicity (straightforward *sum and difference computation*)

Outline

Generalized Haar-Walsh Transform (GHWT)

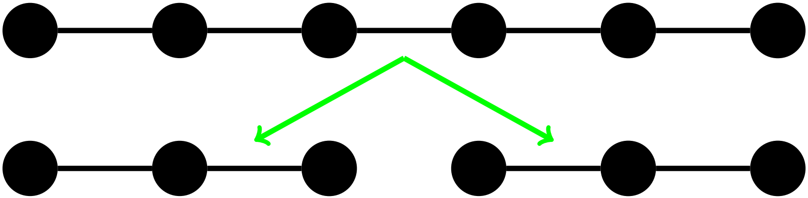
The *GHWT* is a true generalization of the classical Haar-Walsh Wavelet Packet Transform, and generates a *dictionary* (i.e., a redundant set) of basis vectors that are *piecewise-constant* on their support.

The algorithm using the Fiedler vectors can be summarized as follows although any other graph partitioning algorithm can be used:

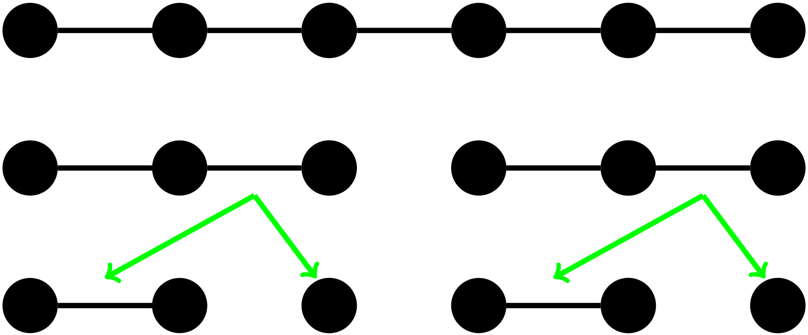
1. Generate a full recursive bipartitioning of the graph using *Fiedler vectors* $\boldsymbol{\phi}^j_{k,1}$ of $L_{\mathrm{rw}}(G^j_k)$, where $k = 0, \ldots, K^j - 1$ indicates a region, $j = 0, \ldots, j_{\mathrm{max}}$ indicates a level (or scale), $V = V^0_0 = V^1_0 \cup V^1_1 = \cdots$
2. Generate an orthonormal basis for level j_{max} (the finest level) $\Rightarrow$ ***scaling vectors*** on the single-node regions
3. Using the basis for level j_{max}, generate an orthonormal basis for level $j_{\mathrm{max}} - 1 \Rightarrow$ ***scaling*** and ***Haar*** vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j - 1 \Rightarrow$ ***scaling***, ***Haar***, and ***Walsh*** vectors

GHWT on P_6

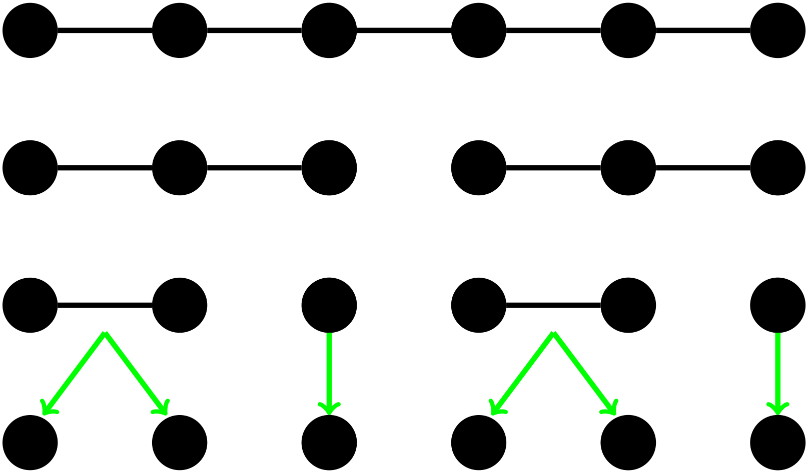
GHWT on P_6



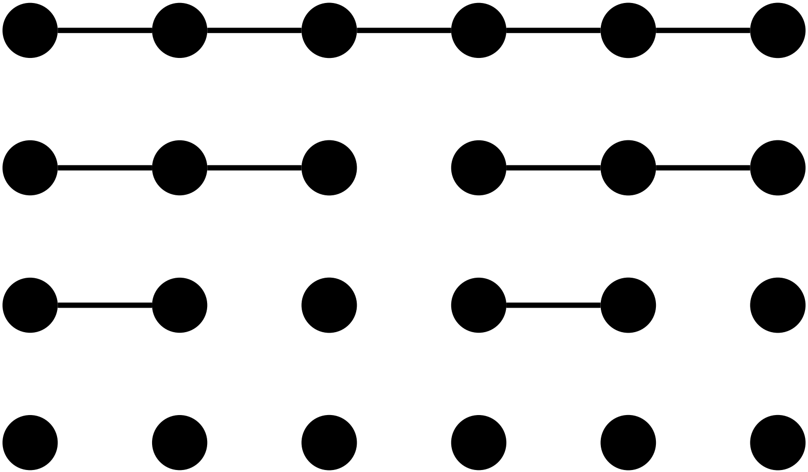
GHWT on P_6



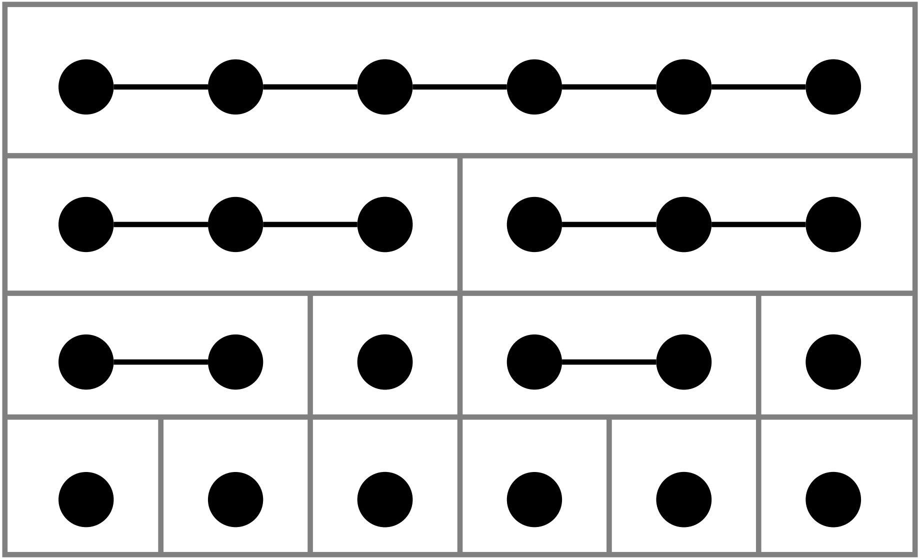
GHWT on P_6



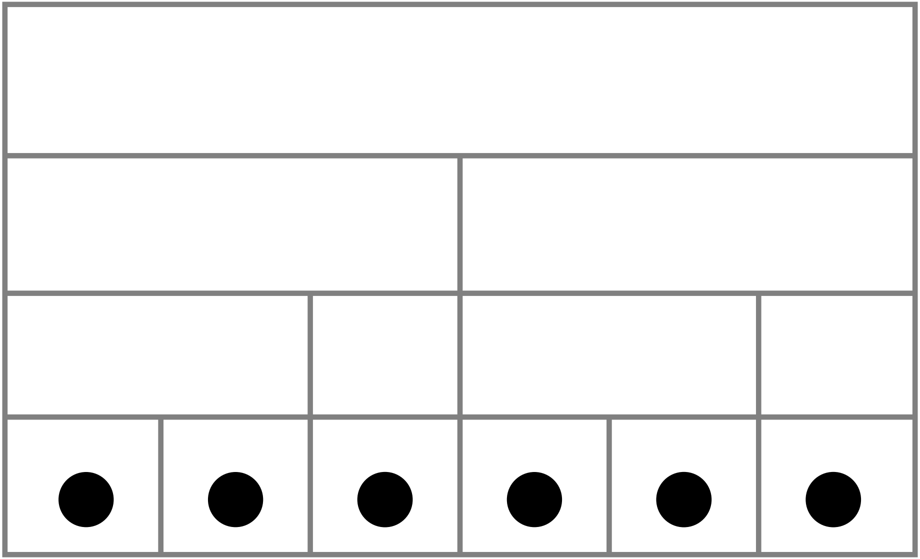
GHWT on P_6



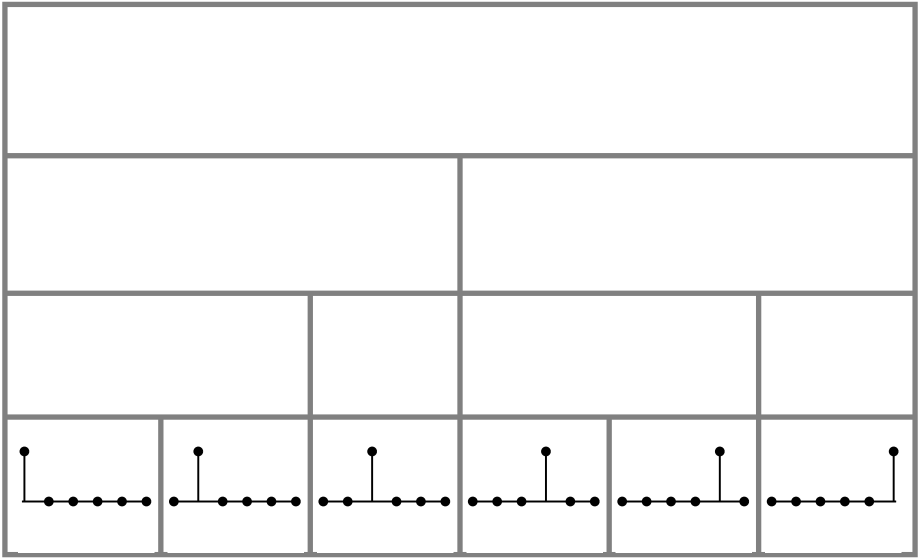
GHWT on P_6



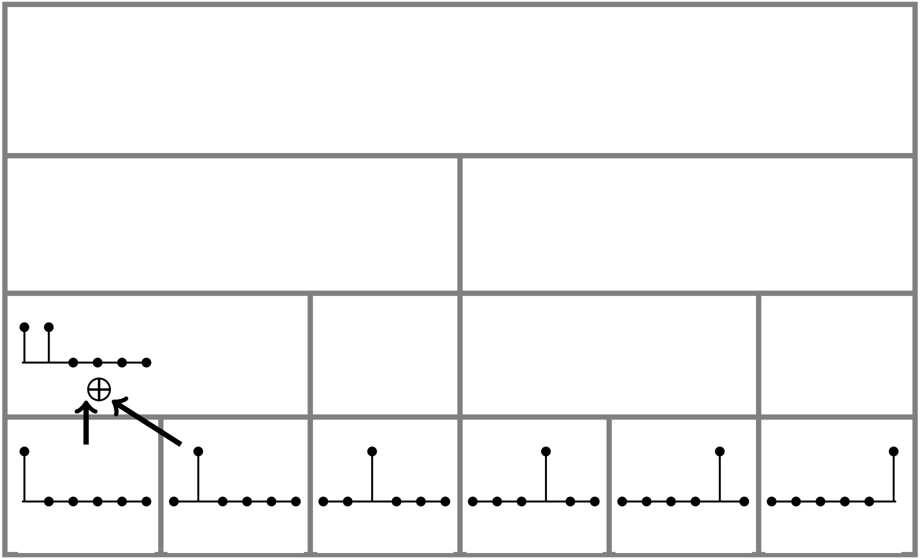
GHWT on P_6



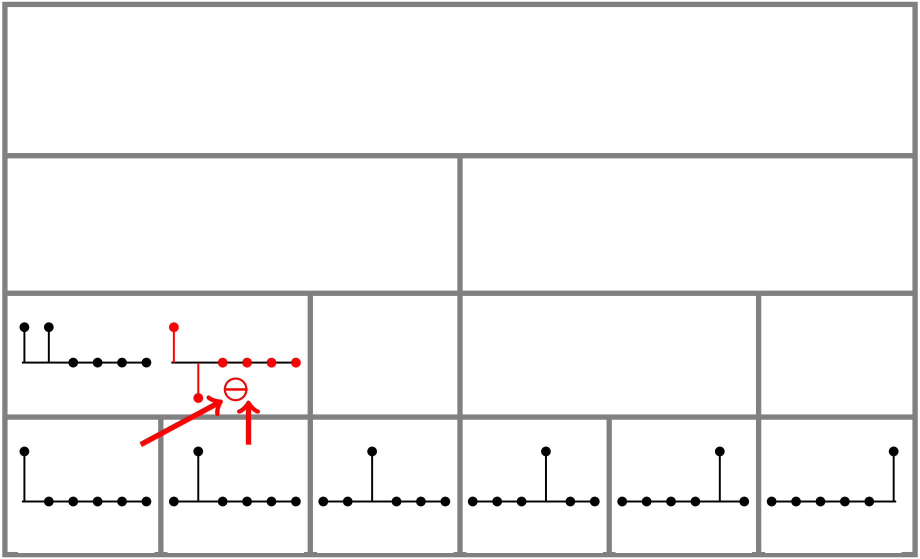
GHWT on P_6



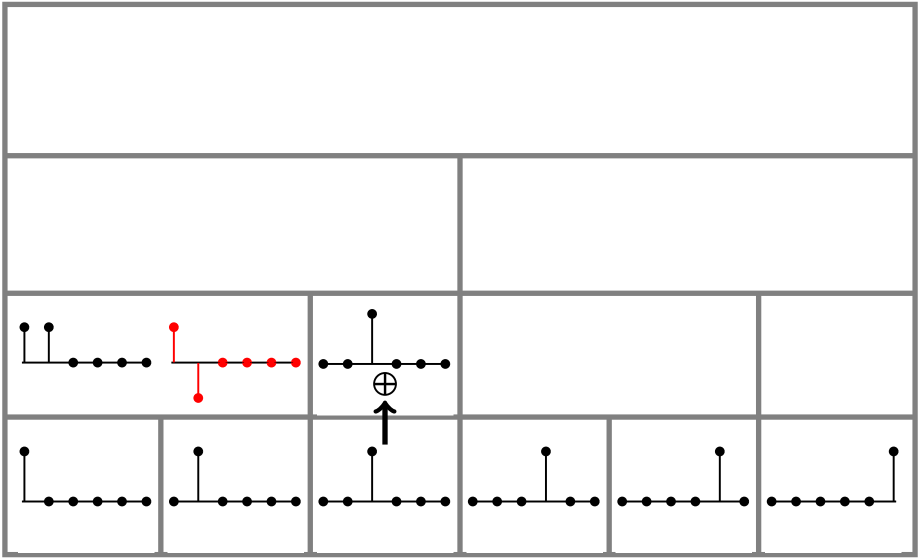
GHWT on P_6



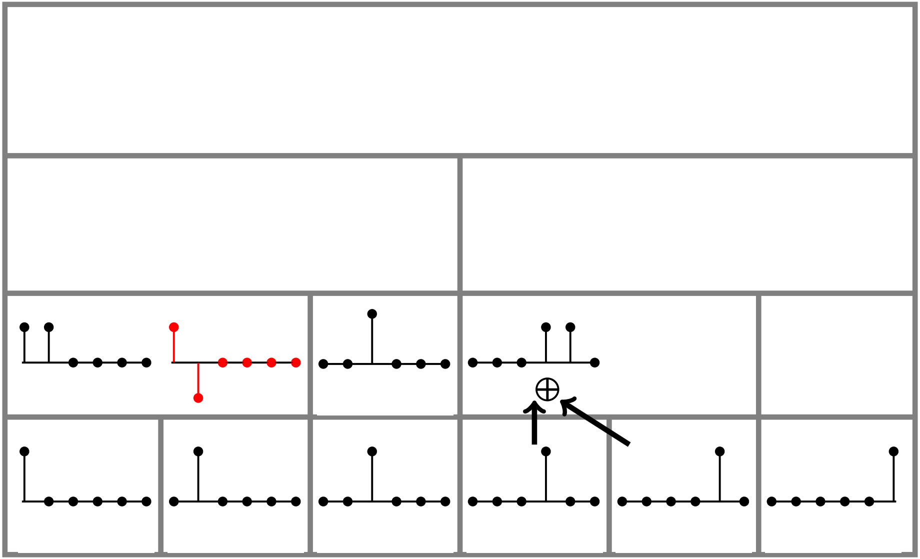
GHWT on P_6



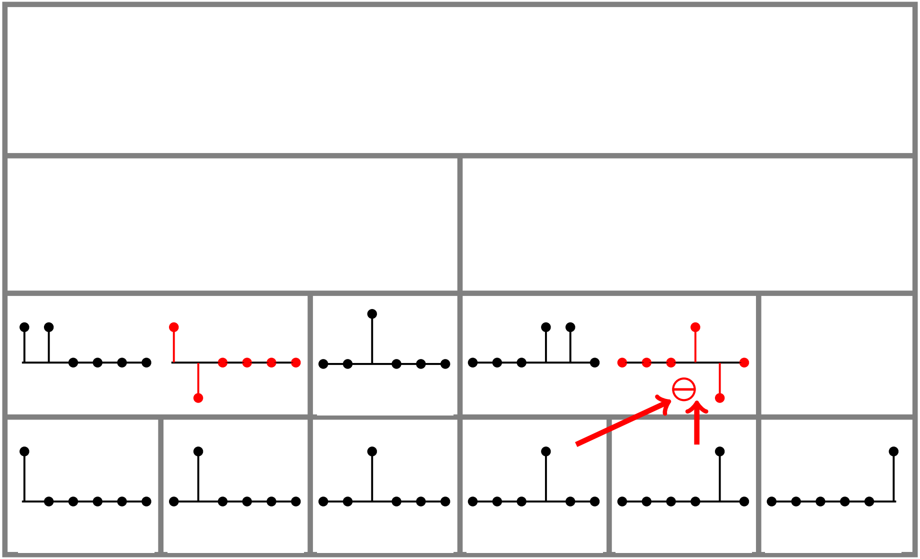
GHWT on P_6



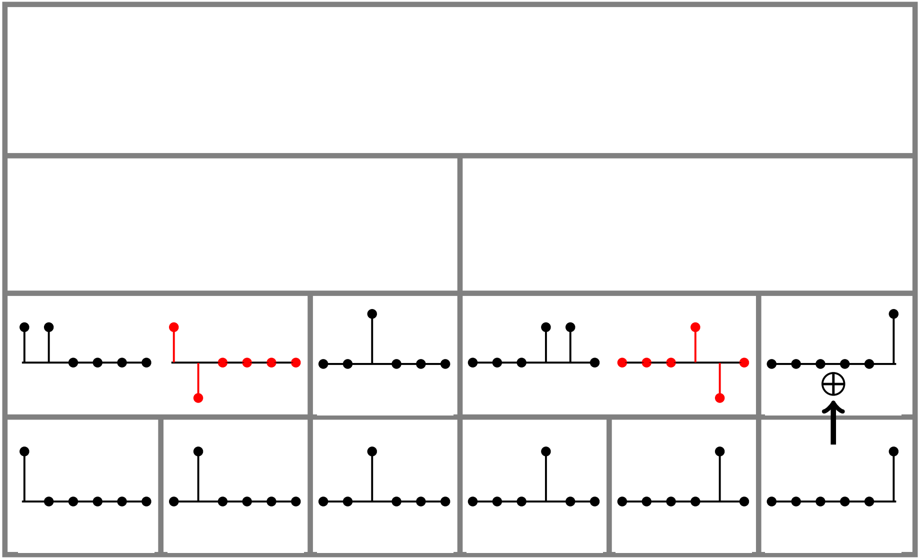
GHWT on P_6



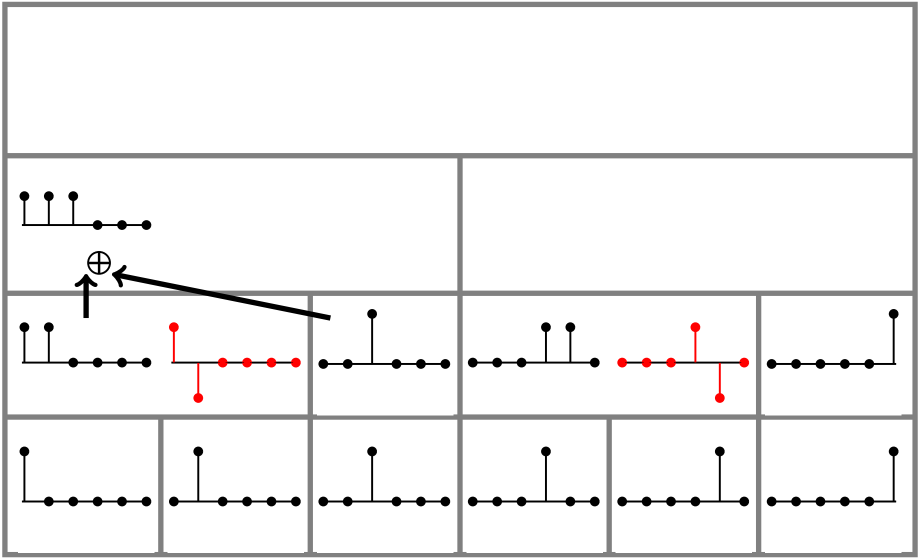
GHWT on P_6



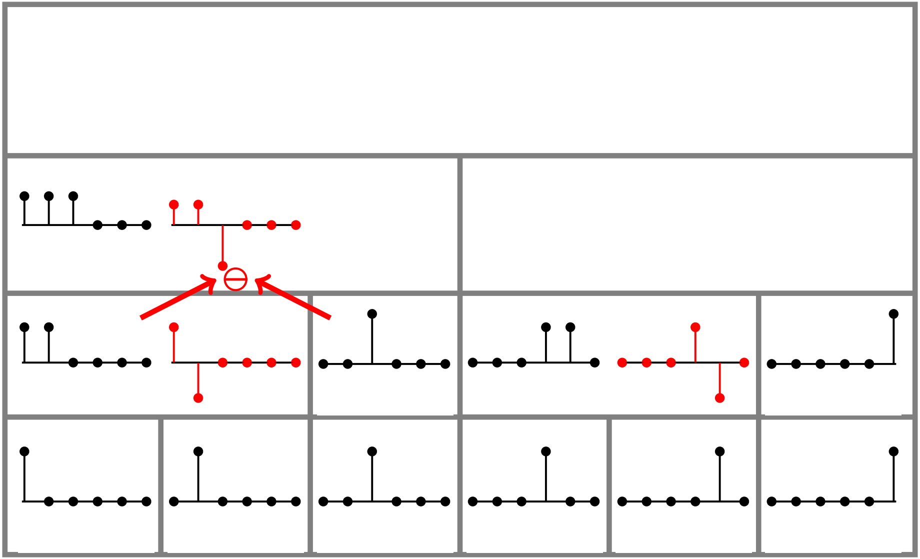
GHWT on P_6



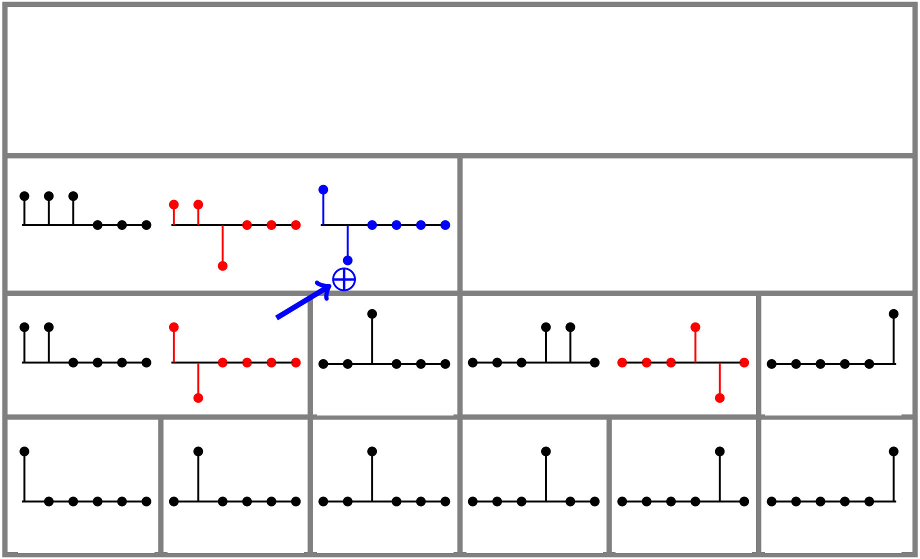
GHWT on P_6



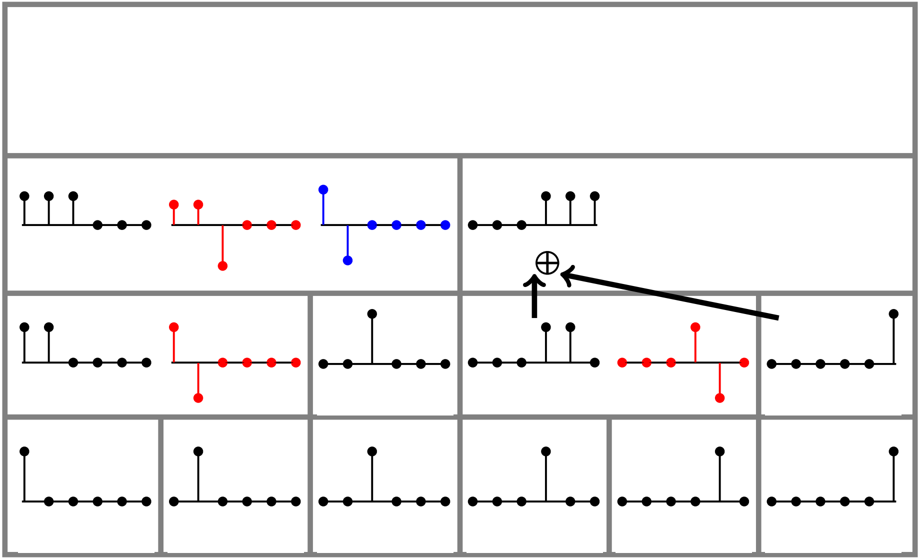
GHWT on P_6



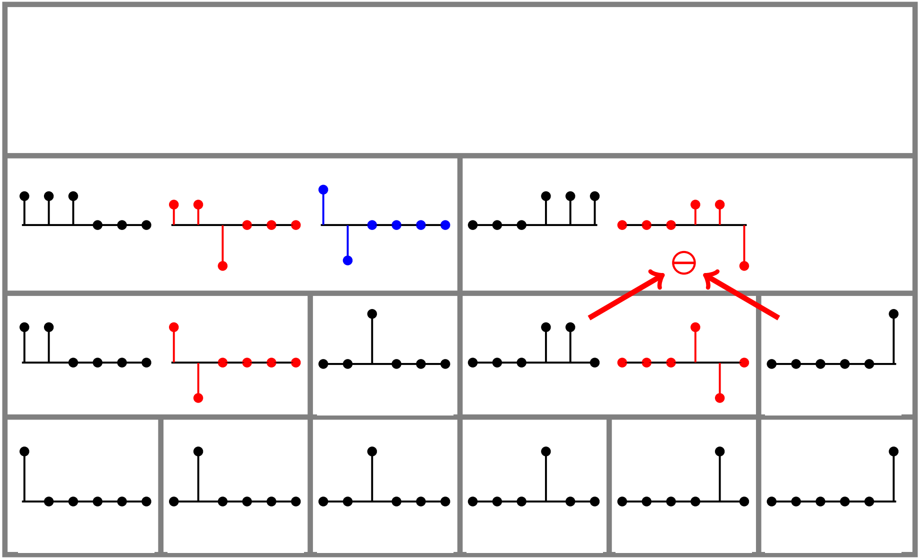
GHWT on P_6



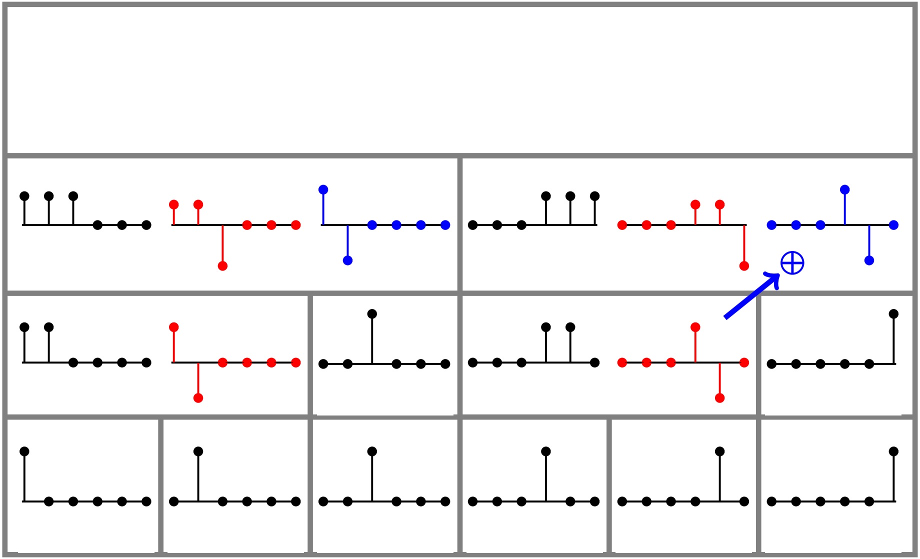
GHWT on P_6



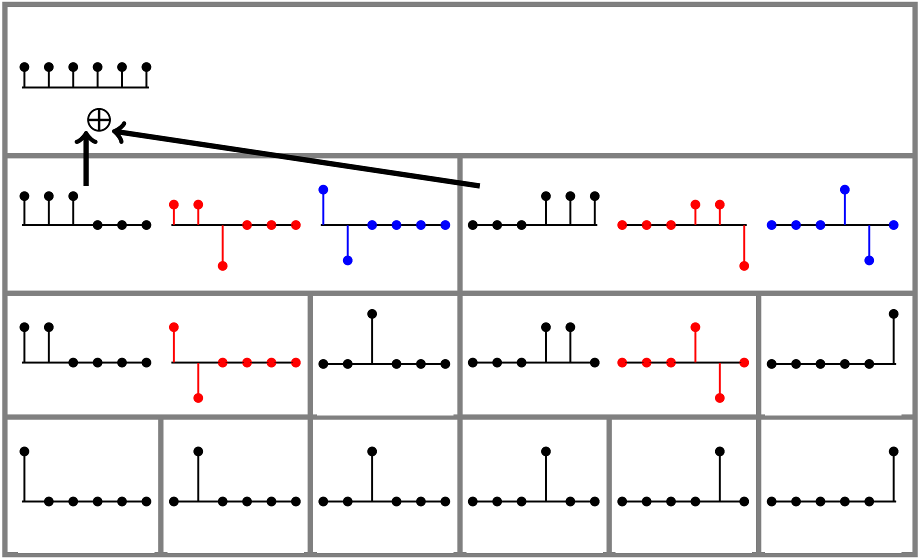
GHWT on P_6



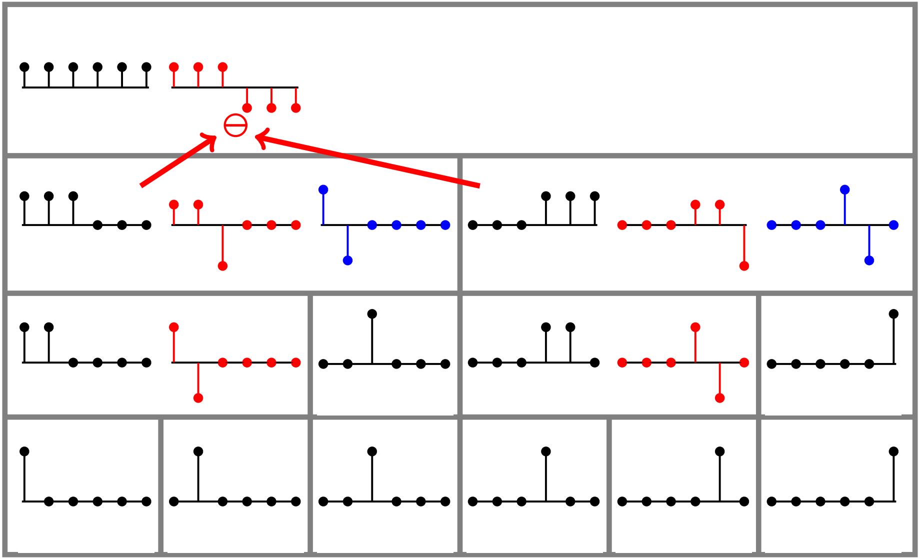
GHWT on P_6



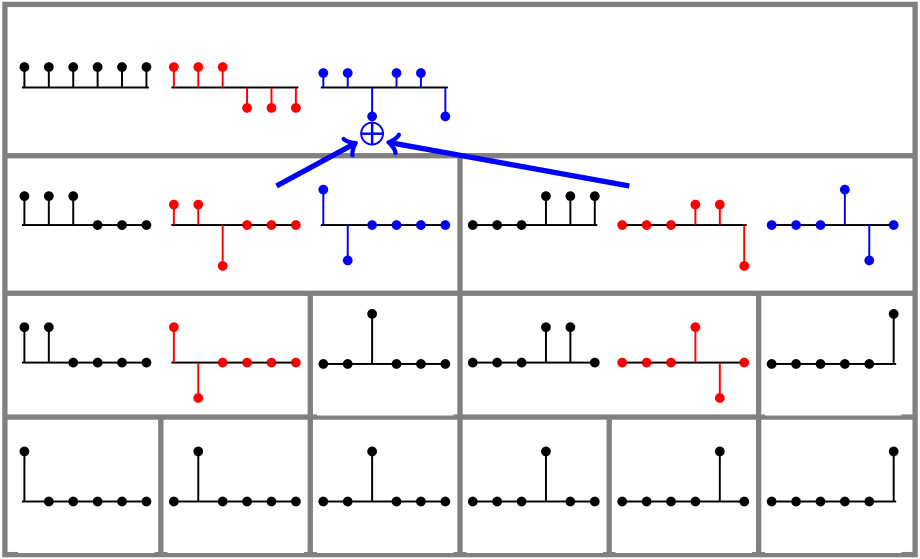
GHWT on P_6



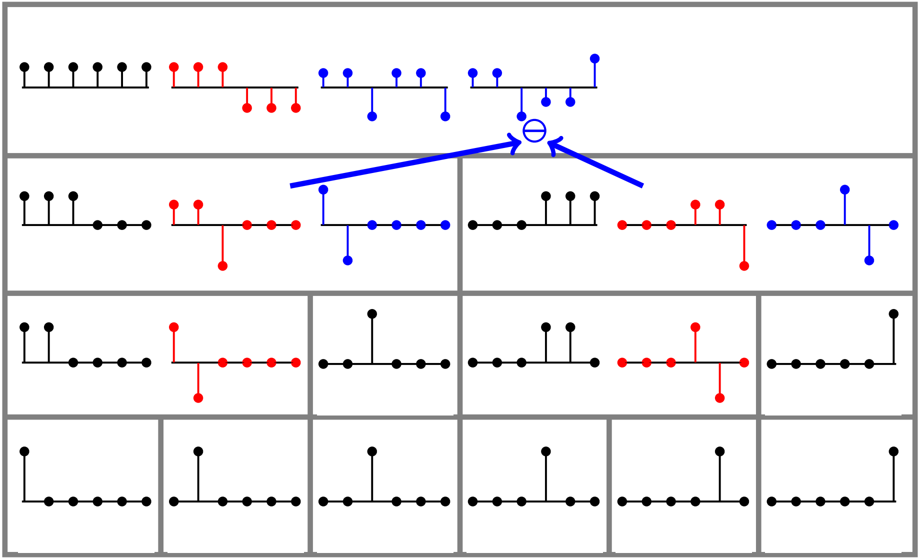
GHWT on P_6



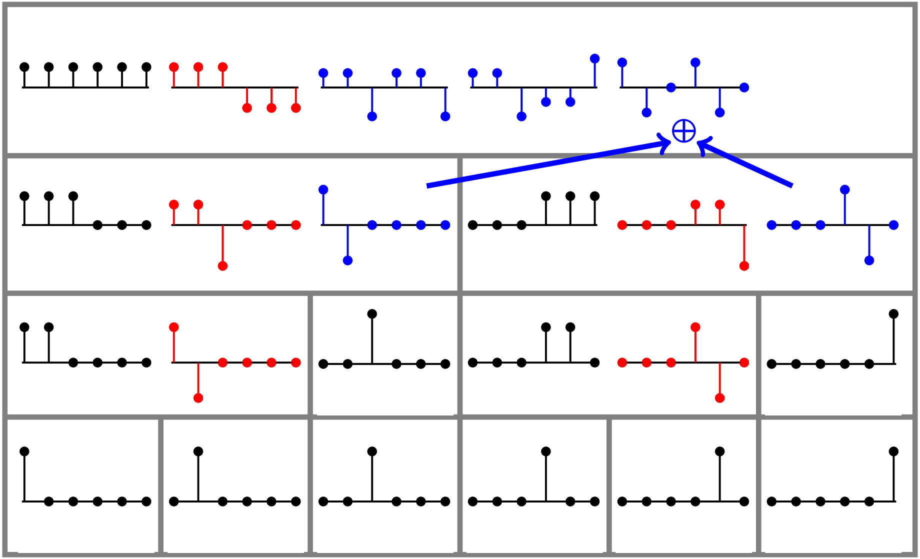
GHWT on P_6



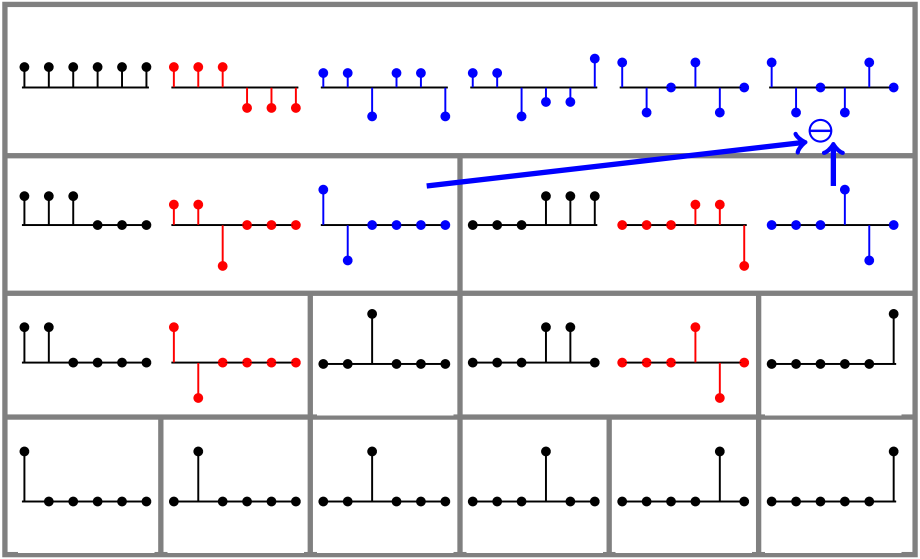
GHWT on P_6



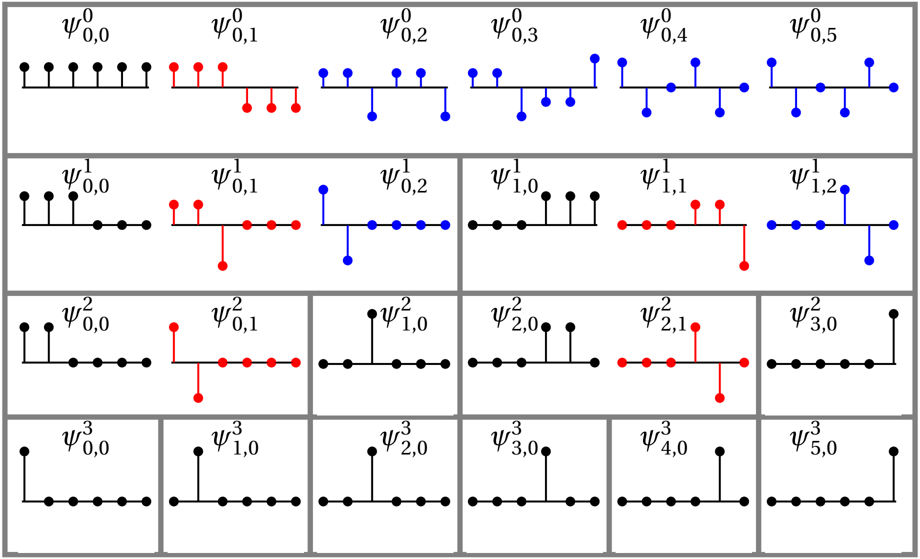
GHWT on P_6



GHWT on P_6



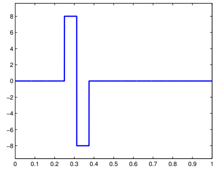
GHWT on P_6


$\psi^0_{0,0}$
$\psi^0_{0,1}$
$\psi^0_{0,2}$
$\psi^0_{0,3}$
$\psi^0_{0,4}$
$\psi^0_{0,5}$
$\psi^1_{0,0}$
$\psi^1_{0,1}$
$\psi^1_{0,2}$
$\psi^1_{1,0}$
$\psi^1_{1,1}$
$\psi^1_{1,2}$
$\psi^2_{0,0}$
$\psi^2_{0,1}$
$\psi^2_{1,0}$
$\psi^2_{2,0}$
$\psi^2_{2,1}$
$\psi^2_{3,0}$
$\psi^3_{0,0}$
$\psi^3_{1,0}$
$\psi^3_{2,0}$
$\psi^3_{3,0}$
$\psi^3_{4,0}$
$\psi^3_{5,0}$

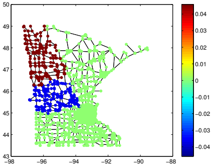
Basis Vector & Coefficient Notation

GHWT basis vectors and coefficients are written as $\boldsymbol{\psi}^j_{k,\ell}$ and $c^j_{k,\ell}$, respectively, where j and k correspond to level and region and ℓ is the **tag**.

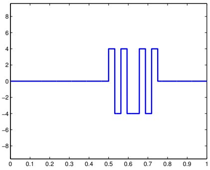
- $\ell = 0 \Rightarrow$ **scaling coefficient/basis vector**
- $\ell = 1 \Rightarrow$ **Haar coefficient/basis vector**
- $\ell \geq 2 \Rightarrow$ **Walsh coefficient/basis vector**



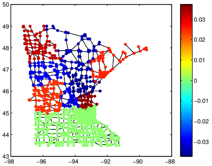
(a) Haar function on $\mathbb{R}$



(b) Haar vector $\boldsymbol{\psi}^2_{0,1}$



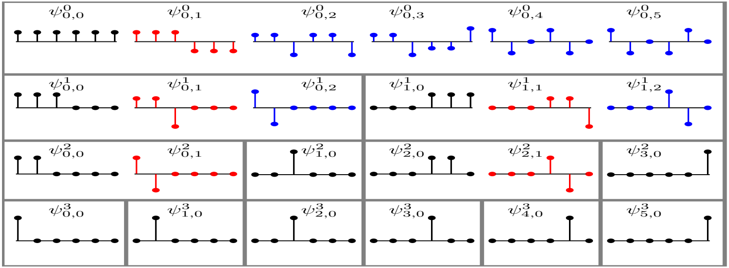
(c) Haar-Walsh wavelet packet on $\mathbb{R}$



(d) Walsh vector $\boldsymbol{\psi}^1_{0,5}$

Remarks

- For an unweighted path graph of dyadic length, this yields *exactly* a dictionary of the conventional Haar-Walsh wavelet packets.
- Recursive Partitioning (RP) via Fiedler vectors costs $O(N^2)$ in general.
- Given a recursive partitioning with $O(\log N)$ levels, the computational cost of expanding an input data into the GHWT is $O(N \log N)$.
- We can select an orthonormal basis for the entire graph by taking the union of orthonormal bases on disjoint regions.



Remarks . . .

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar**, or **Walsh**).

Remarks . . .

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar**, or **Walsh**).

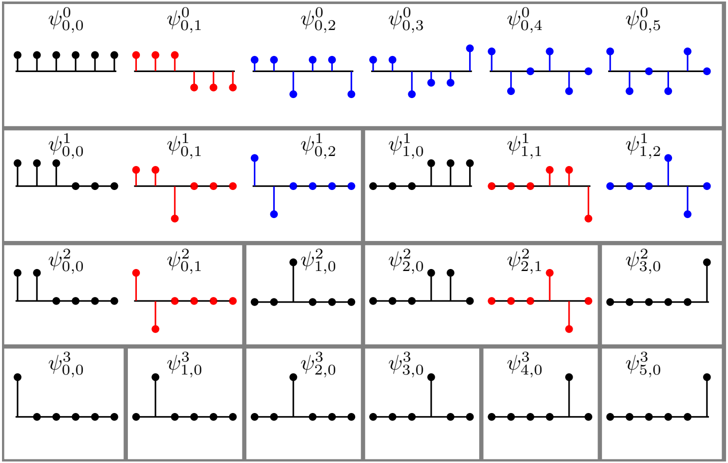


Figure: Default dictionary; i.e., *coarse-to-fine*

Remarks . . .

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar**, or **Walsh**).

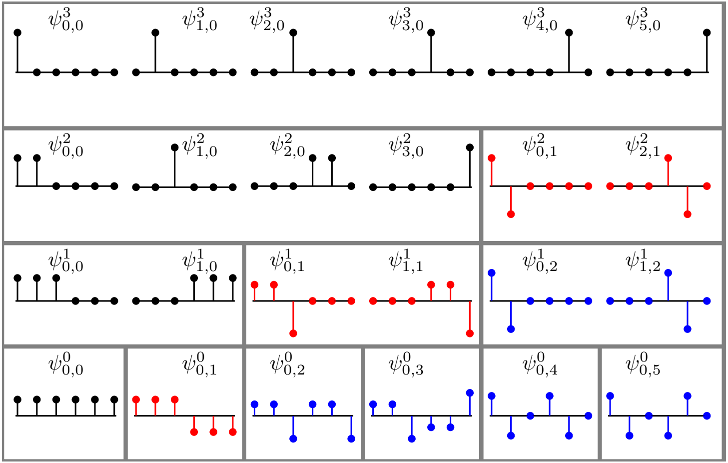


Figure: Reordered & regrouped dictionary; i.e., *fine-to-coarse*

Outline

Best-Basis Algorithms for GHWT

- Coifman and Wickerhauser (1992) developed the *best-basis* algorithm as a means of selecting the basis from a dictionary of wavelet packets that is "best" for approximation/compression.
- We generalize this approach, developing and implementing an algorithm for selecting the basis from the GHWT dictionary in the *bottom-up* manner that is "best" for approximation and compression.
- We require an appropriate cost functional $\mathscr{J}$, e.g.,

$$\mathscr{J}\left(\boldsymbol{c}_k^j\right) = \left\|\boldsymbol{c}_k^j\right\|_p := \left(\sum_{\ell=0}^{N_k^j-1} \left|c_{k,\ell}^j\right|^p\right)^{1/p} \qquad 0 < p \leq 1,$$

  for seeking the *sparsest* representation of the input graph signal.
- For other tasks, e.g., classification and regression, see the work of N.S. on *Local Discriminant Basis*, *Local Regression Basis*, *Least Statistically-Dependent Basis*, ..., all of which use different cost functionals and can also be used in the graph setting.

Outline

Motivation of developing eGHWT

- We want to complete the lifting of the Haar-Walsh wavelet packets to the graph setting:

Regular Lattice	$\subset$ Graphs	# Choosable Bases	Costs
H-W Wavelet Packets[1]	$\subset$ GHWT[2]	$> (1.5)^N$	$O(N \log N)$
$\cap$	$\cap$	$\wedge$	$\|\|$
Adaptive H-W Tilings[3]	$\subset$ eGHWT[4]	$> 0.618 \cdot (1.84)^N$	$O(N \log N)$

- The difference between these two could be huge: for $N = 1024$, eGHWT searches 10^{270} possible bases whereas GHWT does 10^{181} bases.

[1] Coifman-Meyer (1989)
[2] Irion-Saito (2014)
[3] Thiele-Villemoes (1996)
[4] Saito-Shao (2019)

Outline

Time-Frequency Adapted Haar-Walsh Tilings

- Thiele and Villemoes (1996) proposed an $O(N \log N)$ algorithm to search the best basis among much larger collection of orthonormal bases than the conventional best basis algorithm due to Coifman and Wickerhauser (1992) can search.
- The essence of this algorithm is that at each step of the recursive evaluation of the costs of subspaces, it compares the cost of the parent subspace with not only its two children subspaces partitioned in the "frequency" domain (like the wavelet packets), but also its two children subspaces partitioned in the "time" (or "space") domain (like the local cosines).
- Lindberg and Villemoes (2000) extended this algorithm for 2D signals and got quite good compression/approximation of various digital images.

The Thiele-Villemoes Algorithm (1D)

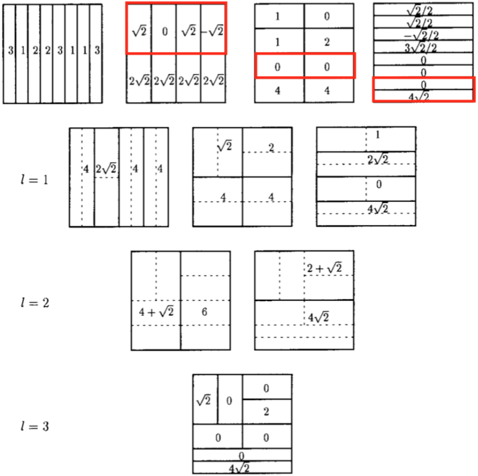


Figure: An example: 1D input vector $\boldsymbol{f} = [3, 1, 2, 2, 3, 1, 1, 3]^{\mathsf{T}} \in \mathbb{R}^8$ (from *Appl. Comput. Harmon. Anal.*, vol.3, p.96, 1996)

The Lindberg-Villemoes Algorithm (2D)

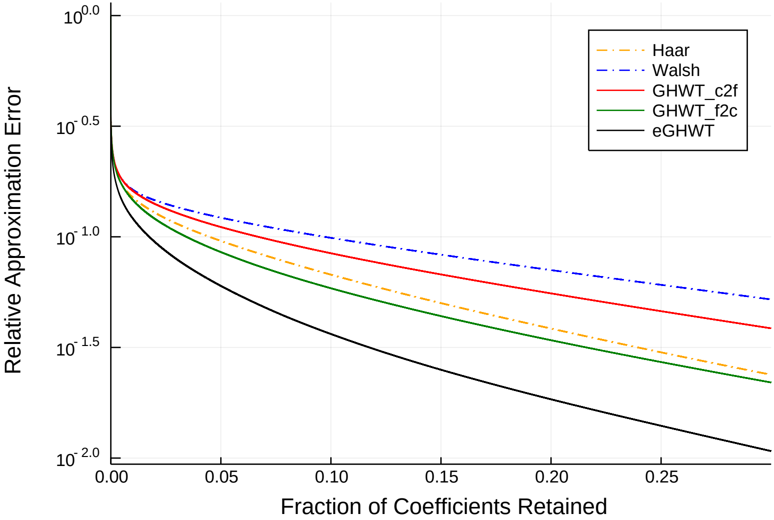


Figure: Relative ℓ^2 approximation error of the Barbara Image 512×512

(a) Haar

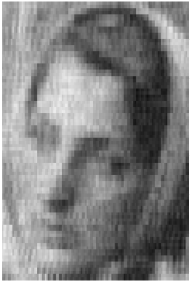
(b) GHWT c2f

(c) GHWT f2c

(d) eGHWT

Figure: Comparison of various bases: using only 3.125% of coefficients

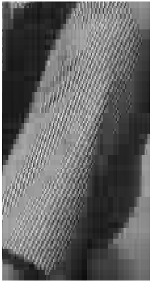
(a) Haar



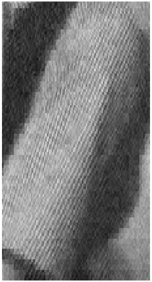
(b) GHWT c2f

(c) GHWT f2c

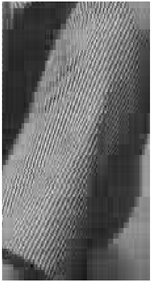
(d) eGHWT



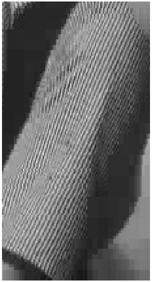
(a) Haar



(b) GHWT c2f



(c) GHWT f2c

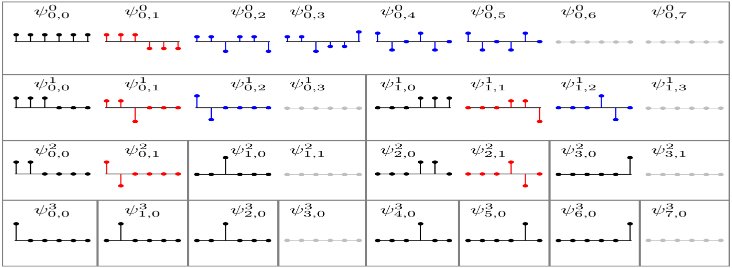


(d) eGHWT

Outline

The eGHWT best-basis algorithm

1. Given the c2f GHWT dictionary, add *fictitious leaves* to the bottom of the tree so that all the leaves are in pair
2. Proceed upward as the GHWT to make it a *completely balanced tree*
3. Adjust the tag l and region index k of each $\psi^j_{k,l}$ accordingly. 0 will be assigned as the expansion coefficients of an input graph signal relative to the newly added basis vectors due to those fictitious leaves:



4. *Apply the Thiele-Villemoes algorithm* on this modified binary tree.
5. *Restrict the support* of the best basis vectors selected from this balanced tree to the original nodes $\Longrightarrow$ *the best basis we want!*

Outline

Graph Signal Compression

- Vehicular traffic volume data over a 24 hour period at intersections in the road network of Toronto ($N = 2202$ nodes and $M = 4877$ edges) are used for comparing the performance of various graph bases
- Edge weights = the Euclidean distances between the nodes

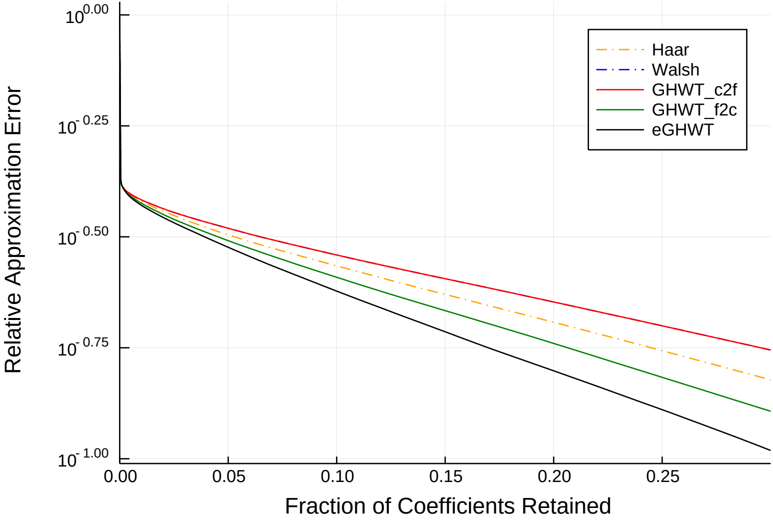
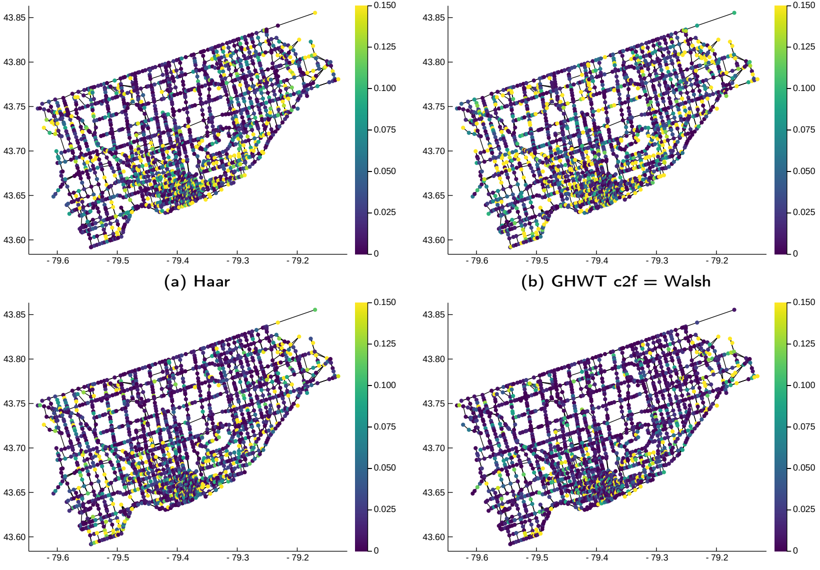


Figure: Relative ℓ^2 approximation error of the Toronto traffic data



(a) Haar

(b) GHWT c2f = Walsh

(c) GHWT f2c

(d) eGHWT

Figure: Pointwise relative ℓ^2-error using 25% of coefficients

Textured Image Compression

- Applying eGHWT to a regular 2D image lead to basis vectors whose support has better adapted to the structure of the input image (not necessarily rectangular shapes)
- To do so, build a graph from a regular 2D image where each node represents a pixel and each edge weight reflects local similarity around the two pixels associated with that edge

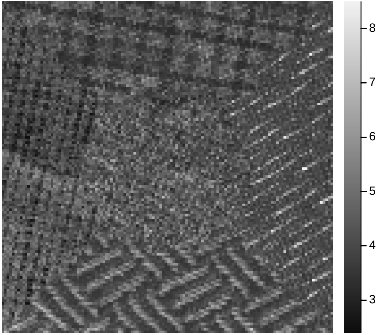


Figure: A composite textured image 128×128

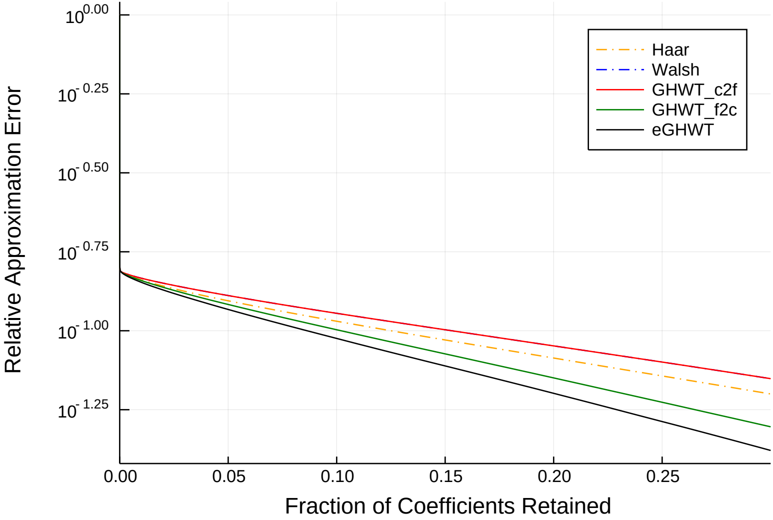


Figure: Relative ℓ^2 approximation error of the composite texture image

Figure: Top 9 Haar basis vectors from GHWT f2c

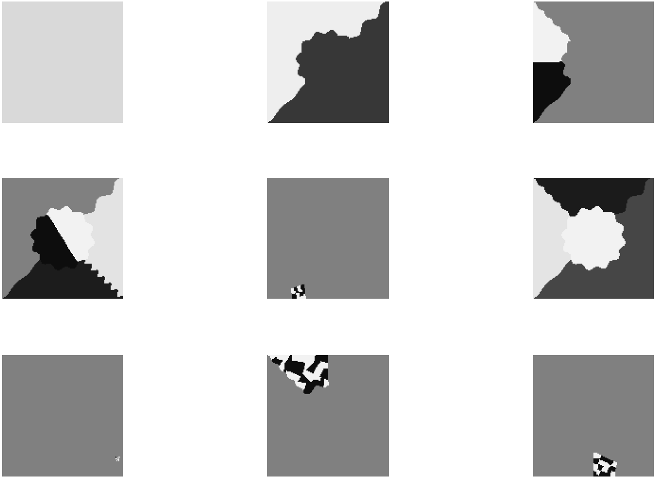


Figure: Top 9 eGHWT best basis vectors

Term-Document Matrix Analysis

Dataset: the Science News database (1153×1042)

- Rows → preselected words
- Columns → articles from 8 fields: Anthropology; Astronomy; Behavioral Sciences; Earth Sciences; Life Sciences; Math & CS; Medicine; Physics
- a_{ij} → the relative frequency of word i appears in article j ⇒ all column sums are 1
- eGHWT best basis vectors for rows analyze meaningful groupings of words while those for columns do the same for documents

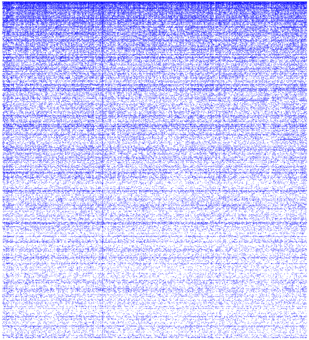


Figure: Science News database (original order)

Term-Document Matrix Analysis

Dataset: the Science News database (1153×1042)

- Rows → preselected words
- Columns → articles from 8 fields: Anthropology; Astronomy; Behavioral Sciences; Earth Sciences; Life Sciences; Math & CS; Medicine; Physics
- a_{ij} → the relative frequency of word i appears in article j ⇒ all column sums are 1
- eGHWT best basis vectors for rows analyze meaningful groupings of words while those for columns do the same for documents

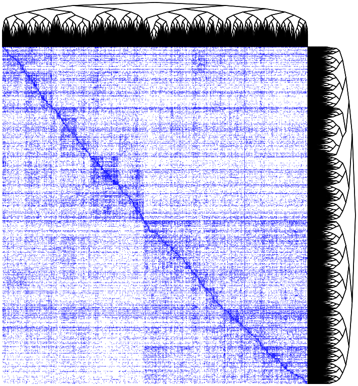


Figure: Science News database (reordered rows and columns)

Outline

Summary

- The eGHWT best-basis algorithm searches over an immense number of orthonormal bases, much more than the conventional GHWT best-basis algorithm does.
- When selected using an appropriate cost functional, the eGHWT best basis outperforms the graph Haar/Walsh bases, the conventional GHWT best basis.
- Graph signal compression and denoising demonstrate an advantage of a *data-adaptive basis dictionary* from which one can select the most suitable basis for one's task at hand!
- Combining the spectral co-clustering and GHWT leads to a powerful tool to analyze *matrix data*, e.g., term-document matrices, microarray data, etc.

Future Plan

- Should develop true *quadtree*-based 2D dictionaries instead of the tensor product of binary-tree-based 1D dictionaries.
- Should explore different cost functionals than the sparsity $\Longrightarrow$ Local Regression Basis (LRB) of Saito and Coifman.
- Should explore the *Group Lasso* idea in classification using the whole GHWT dictionary coefficients to examine/select the coordinates that are *simultaneously useful for all the classes*.
- What to do if your input data is of *tensor* form, i.e., $A = (a_{ijk}) \in \mathbb{R}^{I \times J \times K}$? $\Longrightarrow$ a *tripartite* graph (a.k.a. 3-uniform *hypergraph*)!

Outline

References

The following articles (and the other related ones) are available at
http://www.math.ucdavis.edu/~saito/publications/

- N. Saito & Y. Shao: "The extended Generalized Haar-Walsh Transform and applications," in *Wavelets and Sparsity XVIII, Proc. SPIE 11138*, Paper #111380C, 2019.
- J. Irion & N. Saito: "Efficient approximation and denoising of graph signals using the multiscale basis dictionaries,", *IEEE Trans. Signal Inform. Process. Netw.*, vol. 3, no. 3, pp. 607–616, 2017.
- J. Irion & N. Saito: "Learning sparsity and structure of matrices with multiscale graph basis dictionaries," in *Proc. 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2016.
- J. Irion & N. Saito: "Applied and computational harmonic analysis on graphs and networks," in *Wavelets and Sparsity XVI, Proc. SPIE 9597*, Paper # 95971F, 2015.
- J. Irion & N. Saito: "The generalized Haar-Walsh transform," *Proc. 2014 IEEE Workshop on Statistical Signal Processing*, pp. 488–491, 2014.

We are planning to disseminate our eGHWT/GHWT codes written in *Julia* in the near future, so stay tuned!

Thank you very much for taking this course!