FAST IMPLEMENTATION FOR SYMMETRIC NON-SEPARABLE TRANSFORMS BASED ON GRIDS

Keng-Shih Lu and Antonio Ortega

Department of Electrical Engineering, University of Southern California, Los Angeles, USA

ABSTRACT

When a line graph is symmetric, the associated graph Fourier transform has a fast implementation. In this paper, we extend this idea to the 2D non-separable case, where the graph of interest is a square-shaped grid. We investigate a number of symmetry types for 2D grids. Then, for each type of symmetry we derive a block-diagonalization form of the graph Laplacian matrix, based on which fast implementations with reduced number of multiplications can be obtained. We show that for moderate block sizes, certain types of grid symmetry enable us to design non-separable block transforms that have computational complexities comparable to those of separable ones.

Index Terms— Non-separable transforms, bisymmetric matrix, fast implementation, graph Fourier transform

1. INTRODUCTION

In transform coding schemes [1], a linear transformation is applied to each $N \times N$ image/video block, to obtain N^2 transform coefficients. The Karhunen-Loeve Transform (KLT) is a well-known transform that achieves an optimal energy compaction, but it is data-driven and non-separable, limiting its practical use. Instead, the separable discrete cosine transform (DCT) [2] is widely used for video coding because of its fast implementations and good energy compaction. Another separable transform, the asymmetric discrete sine transform (ADST) [3] is used for encoding intra residual video blocks. While separable transforms are adopted because they have lower computational complexity, non-separable designs offer additional flexibility in shaping the corresponding basis functions and thus can potentially provide a higher compression gain, in particular for blocks that include diagonal structures.

In graph signal processing [4, 5], inter-sample relation is characterized by edge weights in a graph. The corresponding graph Fourier transform (GFT) can be defined using the associated graph Laplacian matrix. Mathematical models or heuristics can be used to design a graph whose GFT can match the properties of a given data set. In particular, 2D grid graphs have been proposed to model image data, leading to non-separable transforms for pixel blocks, derived from the corresponding GFTs. Examples of such graph based transforms (GBTs) include [6, 7, 8], which target compression of images containing various types of sharp discontinuities (edges). In addition, indeed, both the DCT and the ADST can be interpreted as the GFTs of line graphs [2, 3] (i.e., grid graphs in 1D).

Computational complexity of GFTs based on non-separable 2D grids can be significant because it can lead to interest in the development of fast algorithms for such transforms. In [9], a rotation-based speedup technique was proposed, which computes a number of rotation angles in a stage-by-stage manner to derive fast implementations. In [10], the Flexible Approximate MUlti-layer Sparse Transforms (FAμST) was proposed to solve a sparse factorization problem that gives an approximate GFT with fast implementation. Both approaches apply to general GFTs for 2D grids, but they have certain limitations. In particular, in [10], the search for jointly optimal angles becomes impractical when the transform size is large. In FAμST, the fast GFTs are not exact, and the acquisition of such transforms relies on a proximal algorithm in [11].

In our recent work [12], we showed that the GFT for a line graph has a fast implementation if the graph is symmetric. In this paper, our goal is to obtain fast implementations based on grid symmetries by extending the idea in [12] from 1D line graphs to 2D non-separable grids. We investigate several types of grid symmetry: UD/LR-symmetry, centrosymmetry, diagonal and anti-diagonal symmetries, and cases with multiple symmetries. For each symmetry type, we exploit the corresponding properties of the Laplacian matrix, then derive speedup techniques using node reordering and Kronecker product. As opposed to previous methods [9, 10], we focus on graphs with specific symmetry properties, and analytically derive factorization forms that lead to speedups. We show that in those cases, exact GFTs with analytic representations in terms of the corresponding Laplacian matrices can always be obtained without searching parameters or using other solvers.

2. PRELIMINARIES

In graph signal processing, nodes of the graph are associated to samples of the signal to be processed, and each edge describes the inter-sample relation of the two corresponding nodes. Given a signal $\mathbf{x}$ with N samples and N is even, we denote the graph by $\mathcal{G}(\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \dots, N\}$ and $\mathcal{E}$ represent the sets of nodes and edges. The generalized Laplacian matrix of graph $\mathcal{G}$ is defined as $\mathbf{L} = \mathbf{S} + \mathbf{D} - \mathbf{W}$, where $\mathbf{S}$, $\mathbf{D}$ and $\mathbf{W}$ denote the self-loop, degree, and adjacency matrices of $\mathcal{G}$, respectively. The GFT is defined using the eigendecomposition of the Laplacian matrix: given a graph signal $\mathbf{x}$, its GFT coefficients are $\mathbf{U}^T\mathbf{x}$, where $\mathbf{U}$ is the eigenmatrix of the associated graph Laplacian matrix. We refer to $\mathbf{U}$ as the GFT matrix.

We will frequently use the two following matrices:

$$\mathbf{J}_N = \begin{pmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{pmatrix}, \quad \mathbf{K}_N = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{I}_{N/2} & -\mathbf{J}_{N/2} \\ \mathbf{I}_{N/2} & \mathbf{J}_{N/2} \end{pmatrix}, \qquad (1)$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix and $\mathbf{J}_N$ is the $N \times N$ order-reversal permutation matrix. The sizes of $\mathbf{I}$, $\mathbf{J}$, and $\mathbf{K}$ are indicated by their subscripts. When $\mathbf{L}$ is an $N^2 \times N^2$ matrix, we denote its $N \times N$ block-partition representation by $\mathbf{L} = (\mathbf{L}_{i,j})_{i,j=1,\dots,N}$, where $\mathbf{L}_{i,j}$ is the (i, j)-th $N \times N$ subblock of $\mathbf{L}$.

This work was supported in part by NSF under grant CCF-1410009.

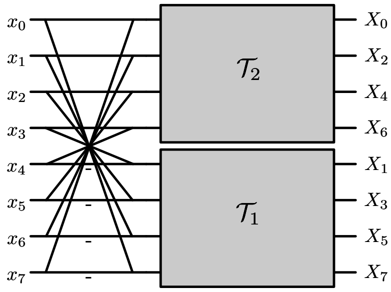

Fig. 1: GFT implementation when $\mathbf{L}$ is bisymmetric and 8×8. Components $\mathcal{T}_1$ and $\mathcal{T}_2$ are 4×4 transforms whose basis functions are the columns of $\mathbf{J}_{N/2}\mathbf{E}_1/\sqrt{2}$ and $\mathbf{E}_2/\sqrt{2}$, where $\mathbf{E}_1$ and $\mathbf{E}_2$ are eigenmatrices of $\mathbf{A} \mp \mathbf{J}_{N/2}\mathbf{C}$, as in Theorem 2.

Definition 1 (Symmetries of vectors and matrices)**.** An $N \times 1$ vector $\mathbf{v}$ is called *symmetric* if $v_i = v_{N-i+1}$ for $i = 1, \dots, N$, and *skew-symmetric* if $v_i = -v_{N-i+1}$ for $i = 1, \dots, n$. An $N \times N$ matrix $\mathbf{Q}$ is *centrosymmetric* (symmetric about the center) if $q_{i,j} = q_{N-i+1,N-j+1}$. If $\mathbf{Q}$ is symmetric and centrosymmetric, then $q_{i,j} = q_{j,i} = q_{N-i+1,N-j+1} = q_{N-j+1,N-i+1}$. In this case, $\mathbf{Q}$ is called *bisymmetric*.

Theorem 2 ([13, 14])**.** Let N be even. An $N \times N$ matrix $\mathbf{L}$ has a set of N linearly independent eigenvectors that are either symmetric or skew-symmetric if and only if $\mathbf{L}$ is centrosymmetric. In particular, if $\mathbf{L}$ is bisymmetric, it can be block-diagonalized by $\mathbf{K}_N$:

$$\mathbf{L} = \mathbf{K}_N^T \begin{pmatrix} \mathbf{A} - \mathbf{J}_{N/2}\mathbf{C} & \mathbf{O} \\ \mathbf{O} & \mathbf{A} + \mathbf{J}_{N/2}\mathbf{C} \end{pmatrix} \mathbf{K}_N, \tag{2}$$

where $\mathbf{A}$ and $\mathbf{C}$ are top-left and bottom-left $N/2 \times N/2$ block elements of $\mathbf{L}$: $A_{i,j} = L_{i,j}$, $C_{i,j} = L_{N/2+i,j}$, $i, j = 1, \dots, N/2$.

Corollary 3 ([12])**.** Let $\mathbf{L}$ be a bisymmetric graph Laplacian matrix, then it can be derived from (2) that the corresponding GFT can be realized using a butterfly operation followed by two parallel $N/2 \times N/2$ matrix transforms on $\mathbf{x}$, as shown in Fig. 1. This reduces the number of multiplications by half as compared to a general $N \times N$ matrix product.

3. GFTS WITH SYMMETRIC 2D GRIDS

Symmetric structures of grids are useful for several reasons. Firstly, as we have explored in [12] and stated in Corollary 3, structural symmetry of 1D line graphs (edges and self loops) is favorable because it allows an acceleration of GFT using a butterfly stage. Here, we are interested in how symmetries of 2D grids can be exploited to provide speedups similar to those in the 1D case. Secondly, it can be observed that symmetric grids occur in real-world data. For example, Fig. 2 shows graphs estimated from inter-pixel correlation of 8×8 residual blocks under two scenarios of predictive coding in HEVC [15]. The graphs are learned using a recently proposed graph Laplacian matrix estimation algorithm [16]. It can be shown that the GFT corresponding to one such data-driven graph is an approximation to the KLT for the data and thus can lead to efficient compression. We observe that the weights in Fig. 2(a) are nearly up-down and left-right symmetric, and those in Fig. 2(b) are nearly symmetric around both diagonals. Fig. 2(c) shows a few blocks among those used to generate Fig. 2(b). We observe that those data from the *BasketballDrill* video, have many diagonal-oriented edges. This yields

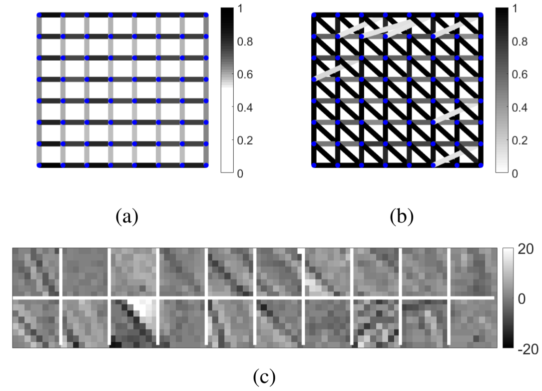

Fig. 2: Estimated inter-pixel correlations in residues of (a) 8×8 blocks that are intra-predicted by planar mode in *ParkScene* video, and (b) 8×8 inter-predicted blocks in *BasketballDrill* video. (c) Randomly chosen sample blocks from those generating (b).

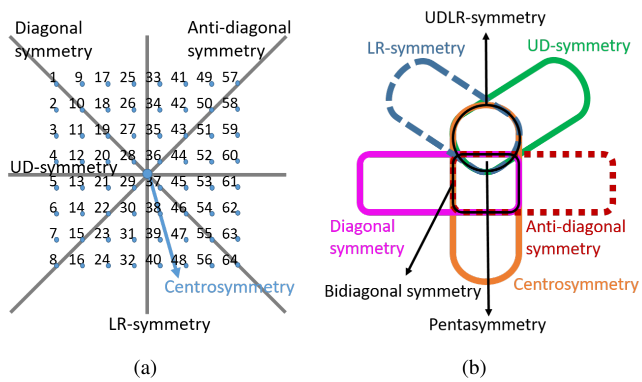

Fig. 3: (a) The axis/point of symmetry for each grid symmetry type. (b) Relationship among types of grid symmetries.

the edge patterns in Fig. 2(b), where the graph weights are larger (higher correlation) in the direction top-left to bottom-right.

Definition 4 (Symmetries of 2D grids)**.** For a grid $\mathcal{G}$, the notion of symmetry can be defined in multiple ways. We give those symmetries the following names: a) UD-symmetry (up-down), b) LR-symmetry (left-right), c) centrosymmetry, d) diagonal symmetry, e) anti-diagonal symmetry[1]. Those symmetries arise if all the edges and self loops of $\mathcal{G}$ are symmetric about the following structures: a) central horizontal axis, b) central vertical axis, c) center of the grid, d) the northwest-to-southeast diagonal, and e) the northeast-to-southwest diagonal. $\mathcal{G}$ is called UDLR-symmetric if it is UD- and LR- symmetric, bidiagonal symmetric if it is diagonally and anti-diagonally symmetric, and pentasymmetric if it satisfies all of a) to e). For each symmetry type, the axis/point of symmetry is shown in Fig. 3(a), and the degree of freedom is listed in Table 1. The interrelation among those symmetries can be visualized in Fig. 3(b).

Our objective is to find butterfly-based speedup techniques by exploiting grid symmetries as follows. With an $N \times N$ grid of a

[1]For matrices, the (main) diagonal and the anti-diagonal refer to the northwest-to-southeast and the northeast-to-southwest diagonals. Here, we extend those terms to the grid case.

given symmetry, we would like to express its Laplacian matrix $\mathbf{L}$ in the form $\mathbf{L} = \mathbf{H}\mathbf{R}\mathbf{H}^T$ (as an extension of (2)) such that

1. $\mathbf{H}$ is orthogonal, and each column of $\mathbf{H}$ is a constant multiple of a vector whose entries are 0, 1, or -1. That is, $\mathbf{H} = \mathbf{G}_\mathbf{H}\mathbf{D}_\mathbf{H}$ with diagonal $\mathbf{D}_\mathbf{H}$, and $\mathbf{G}_\mathbf{H} \in \{0, 1, -1\}^{N\times N}$ has orthogonal columns. One example of such matrix is diag($\mathbf{I}_2$, $\mathbf{K}_6$).
2. $\mathbf{R}$ is a block-diagonal matrix (as the matrix in the middle of (2)), which we would like to be as close as possible to diagonal (i.e., having many smaller blocks is preferable).

With such factorization, each GFT matrix can be represented by $\mathbf{G}_\mathbf{H}\mathbf{D}_\mathbf{H}\mathbf{U}_\mathbf{R}$, where $\mathbf{U}_\mathbf{R}$ is the eigenmatrix of $\mathbf{R}$. Note that multiplying $\mathbf{G}_\mathbf{H}$ does not require any multiplications, and $\mathbf{D}_\mathbf{H}\mathbf{U}_\mathbf{R}$ has the same block diagonal structure as $\mathbf{R}$ does; thus, the number of multiplications is reduced. The closer $\mathbf{R}$ is to diagonal, the more sparse $\mathbf{D}_\mathbf{H}\mathbf{U}_\mathbf{R}$ is, and the fewer multiplications are required. In what follows, we focus on how block-diagonal forms are obtained and what the butterfly-related matrices ($\mathbf{H}$'s) are. We leave out the forms of $\mathbf{R}$'s since they require lengthy notations, and are trivial to derive given the associated $\mathbf{H}$'s.

In a grid, there are also multiple ways to determine the order of nodes, i.e., the one-to-one mapping between 2D coordinates $(1,1)$, $(1,2)$, ..., (N,N) and vertex indices v_1,..., v_{N^2}. In this paper, we use a fixed (column-first) vertex ordering, shown by the numbers in Fig. 3(a), for Laplacian matrix $\mathbf{L}$. Note that we will be choosing symmetry-specific permutations to reorder the nodes such that good symmetry properties of Laplacian matrices emerge. Such reordering is allowed because it does not change the connectivity of the graph, and the associated permutation matrix can be absorbed into $\mathbf{H}$.

3.1. Block-diagonalization of Laplacian matrices

A centrosymmtric grid has a bisymmetric Laplacian matrix. This arises from the fact that for each i, node i and node $N^2 + 1 - i$ are centrosymmetric nodes with respect to each other. The associated bisymmetric $\mathbf{L}$ can be block-diagonalized by $\mathbf{K}_{N^2}$.

Then, we consider UD- and LR- symmetries. We note that the entry $(\mathbf{L}_{i,j})_{kl}$ is associated to the edge between the k-th node of column i and the l-th node of column j. If $\mathbf{L}$ is LR-symmetric, the weight of this edge must be identical to that between the k-th node of column $N + 1 - i$ and the l-th node of column $N + 1 - j$, yielding $(\mathbf{L}_{i,j})_{kl} = (\mathbf{L}_{N+1-i,N+1-j})_{kl}$. As a result, we have $\mathbf{L}_{i,j} = \mathbf{L}_{N+1-i,N+1-j}$ for all i, j, which is a subblock version of matrix centrosymmetry. We can block-diagonalize $\mathbf{L}$ by using the block version of (2), with the $\mathbf{K}$ matrix replaced by $\mathbf{H}_{\text{LR}} = \mathbf{K}_N^T \otimes \mathbf{I}_N$. The UD-symmetric case can be regarded as a flipped (permuted) version of the LR-symmetric case. Therefore, $\mathbf{H}_{\text{UD}}$ can be determined by a row-first reordering $\mathbf{P}_{\text{UD}}$ (shown in Fig. 4(a)) followed by $\mathbf{H}_{\text{LR}}$. When $\mathbf{L}$ is associated to a UDLR-symmetric grid, the finest block-diagonalization can be obtained by cascading $\mathbf{H}_{\text{LR}}$ and $\mathbf{H}_{\text{UD}}$ followed by a node permutation:

$$\mathbf{P}_\delta = (\pi_{i,j})_{i,j=1,\dots,2N} \otimes \mathbf{I}_{N/2}, \quad \pi_{i,j} = \begin{cases} \delta_{i,2j-1}, & 1 \le j \le N \\ \delta_{i,2(j-N)}, & N < j \le 2N \end{cases}$$

Unlike the above symmetry types, a diagonally or anti-diagonally symmetric grid in general does not have a bisymmetric Laplacian matrix; in fact, such matrix in general cannot be made bisymmetric by any node reordering. However, diagonal and anti-diagonal symmetries can still be exploited using our proposed reordering technique as illustrated in Fig. 4(b)(c): we scan the elements on the axis of symmetry followed by the other two separated regions using a zigzag path. For the diagonal symmetric case, in the resulting

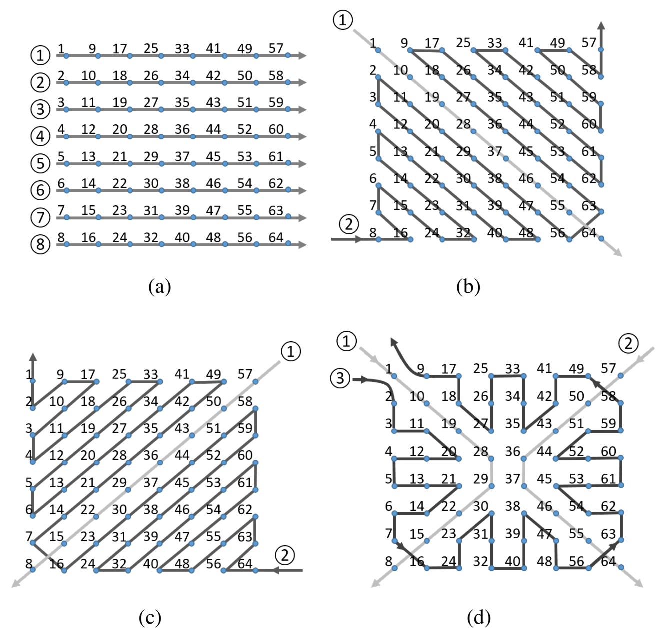

Fig. 4: Vertex reordering by horizontal or zigzag scanning for types of grid symmetries. (a) UD-symmetry. (b) Diagonal symmetry. (c) Anti-diagonal symmetry. (d) Bidiagonal symmetry. The new order is determined by the order of nodes met by the indicated paths, ordered by the circled numbers. These figures characterize the permutation matrices $\mathbf{P}_{\text{UD}}$, $\mathbf{P}_{\text{DS}}$, $\mathbf{P}_{\text{AS}}$, and $\mathbf{P}_{\text{BS}}$, respectively. For example, based on (d), $\mathbf{P}_{\text{BS}}$ is obtained by: `I64=eye(64); P_BS=I64(:,[1,10,19,28,29,22,15,8,57,50,...]);` in MATLAB implementation. Note that (a), (b), and (c) can be generalized to any size N, but (d) is only valid for even N.

permuted matrix $\mathbf{P}_{\text{DS}}^T\mathbf{L}\mathbf{P}_{\text{DS}}$, the first N rows/columns correspond to diagonal nodes in the grid, and the rest correspond to other nodes. With this ordering, the diagonal symmetry leads to symmetry properties in block elements of $\mathbf{P}_{\text{DS}}^T\mathbf{L}\mathbf{P}_{\text{DS}}$:

$$\mathbf{P}_{\text{DS}}^T\mathbf{L}\mathbf{P}_{\text{DS}} = \begin{pmatrix} \mathbf{A} & \mathbf{B} & \mathbf{B}\mathbf{J}_m \\ \mathbf{B}^T & \mathbf{C} & \mathbf{J}_m\mathbf{F}\mathbf{J}_m \\ \mathbf{J}_m\mathbf{B}^T & \mathbf{F} & \mathbf{J}_m\mathbf{C}\mathbf{J}_m \end{pmatrix}, \tag{3}$$

where $m = (N^2 - N)/2$, $\mathbf{A}$ has size $N \times N$, and $\mathbf{C}$ has size $m \times m$. Then, we choose a matrix, diag$(\mathbf{I}_N, \mathbf{K}_{N^2-N}^T)$, to deduce a favorable similar matrix to $\mathbf{L}$:

$$\text{diag}(\mathbf{I}_N, \mathbf{K}_{N^2-N}^T)^T \mathbf{P}_{\text{DS}}^T\mathbf{L}\mathbf{P}_{\text{DS}}\, \text{diag}(\mathbf{I}_N, \mathbf{K}_{N^2-N}^T)$$

$$= \begin{pmatrix} \mathbf{A} & \mathbf{O} & \sqrt{2}\mathbf{B} \\ \mathbf{O} & \mathbf{C} - \mathbf{J}_m\mathbf{F} & \mathbf{O} \\ \sqrt{2}\mathbf{B}^T & \mathbf{O} & \mathbf{C} + \mathbf{J}_m\mathbf{F} \end{pmatrix} \tag{4}$$

$$= \mathbf{P}_\alpha \begin{pmatrix} \mathbf{C} - \mathbf{J}_m\mathbf{F} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{A} & \sqrt{2}\mathbf{B} \\ \mathbf{O} & \sqrt{2}\mathbf{B}^T & \mathbf{C} + \mathbf{J}_m\mathbf{F} \end{pmatrix} \mathbf{P}_\alpha^T, \tag{5}$$

where $\mathbf{P}_\alpha$ is a properly chosen permutation matrix according to (4). Based on (5), we obtain the form of $\mathbf{H}_{\text{DS}}$ as shown in Table 1. The middle matrix in (5) is the desired block-diagonal matrix $\mathbf{R}$, whose nonzero block elements have lengths $(N^2 - N)/2$ and $(N^2 + N)/2$. This matrix is only slightly less sparse than that of the bisymmetric

Table 1: Types of grid symmetries and their speedup techniques for different types of non-separable transforms on $N \times N$ grids. N is assumed to be even. Self loops are counted into the degrees of freedom. The grids are non-separable if not specified. The numbers of multiplications are evaluated by counting the number of nonzero elements in $\mathbf{R}$'s.

Grid Type & Degrees of Freedom	Proposed $\mathbf{H}$	Lengths of nonzero subblocks of $\mathbf{R}$	# Multiplications
No symmetry ($N^4/2+N^2/2$)	N/A	N^2	N^4
Centrosymmetry ($N^4/4+N^2/2$)	$\mathbf{K}_{N^2}^T$	$N^2/2$, $N^2/2$	$N^4/2$
UD-symmetry ($N^4/4+N^2/2$)	$\mathbf{P}_{\text{UD}}\left(\mathbf{K}_N^T \otimes \mathbf{I}_N\right)$	$N^2/2$, $N^2/2$	$N^4/2$
LR-symmetry ($N^4/4+N^2/2$)	$\mathbf{K}_N^T \otimes \mathbf{I}_N$	$N^2/2$, $N^2/2$	$N^4/2$
UDLR-symmetry ($N^4/8+N^2/2$)	$\left(\mathbf{K}_N^T \otimes \mathbf{I}_N\right)\mathbf{P}_{\text{UD}}\left(\mathbf{K}_N^T \otimes \mathbf{I}_N\right)\mathbf{P}_\delta$	$N^2/4$, $N^2/4$, $N^2/4$, $N^2/4$	$N^4/4$
Diagonal sym. ($N^4/4+3N^2/4$)	$\mathbf{P}_{\text{DS}}\ \text{diag}(\mathbf{I}_N,\ \mathbf{K}_{N^2-N}^T)\mathbf{P}_\alpha$	$(N^2-N)/2$, $(N^2+N)/2$	$N^2(N^2+1)/2$
Anti-diagonal sym. ($N^4/4+3N^2/4$)	$\mathbf{P}_{\text{AS}}\ \text{diag}(\mathbf{I}_N,\ \mathbf{K}_{N^2-N}^T)\mathbf{P}_\alpha$	$(N^2-N)/2$, $(N^2+N)/2$	$N^2(N^2+1)/2$
Bidiagonal sym. ($N^4/8+3N^2/8+3N/4$)	$\mathbf{P}_{\text{BS}}\ \text{diag}(\mathbf{K}_{2N}^T,\ \mathbf{K}_{N^2-2N}^T)$ $\cdot\ \text{diag}(\mathbf{I}_{2N},\ \mathbf{I}_2 \otimes \mathbf{K}_{N^2/2-N}^T)\mathbf{P}_\beta$	$(N^2-2N)/4$, $N^2/4$, $N^2/4$, $(N^2+2N)/4$	$N^2(N^2+2)/4$
Pentasymmetry ($N^4/16+N^2/2$)	see (6) below	$(N^2-2N)/8$, $(N^2-2N)/8$, $N^2/4$, $N^2/4$, $(N^2+2N)/8$, $(N^2+2N)/8$	$N^2(3N^2+4)/16$
Separable, no symmetry (N^2+N)	N/A	N	$2N^3$
Separable, UDLR-sym. ($N^2/2$)	$\mathbf{K}_N^T$ for both directions	$N/2$, $N/2$	N^3

$$\mathbf{H}_{\text{penta}} = \mathbf{P}_{\text{BS}}\ \text{diag}(\mathbf{K}_{2N}^T, \mathbf{K}_{N^2-2N}^T)\text{diag}(\mathbf{I}_{2N}, \mathbf{I}_2 \otimes \mathbf{K}_{N^2/2-N}^T)\text{diag}(\mathbf{I}_N, \mathbf{K}_N^T, \mathbf{K}_{N^2/4-N/2}^T, \mathbf{I}_{N^2/2-N}, \mathbf{K}_{N^2/4-N/2}^T)\mathbf{P}_\gamma \tag{6}$$

Table 2: Percentages of residual blocks that have high absolute values of correlation (>0.7) with their flipped versions. The blocks are chosen from selected intra-prediction modes in *Kimono1* video.

Mode	UD	LR	Centro	Diag.	Anti-diag.
HOR	9%	15%	10%	7%	10%
VER	13%	10%	12%	10%	12%
VER+8	10%	13%	19%	18%	19%

case, meaning that the computational saving is comparable to those of UD-, LR-, and centrosymmetric cases. For anti-diagonal symmetry, (3) and (5) apply with $\mathbf{P}_{\text{DS}}$ replaced by $\mathbf{P}_{\text{AS}}$.

For bidiagonally symmetric and pentasymmetric grids, we propose another zigzag scanning as shown in Fig. 4(d) to exploit the symmetry properties. The associated $\mathbf{H}$'s can be obtained based on similar techniques as in (5). Our results are summarized in Table 1, where the permutation matrices $\mathbf{P}_\beta$ and $\mathbf{P}_\gamma$ are determined similarly to the case of $\mathbf{P}_\alpha$. A MATLAB toolbox is available at [17], which provides block diagonalization results given a Laplacian matrix with known symmetry type as input. That each proposed $\mathbf{H}$ satisfies the desired conditions (i.e., $\mathbf{H} = \mathbf{G}_\mathbf{H}\mathbf{D}_\mathbf{H}$) has been verified numerically. Those analytic proofs not included here require lengthy notations, and are left for future work due to lack of space.

4. DISCUSSION

Table 1 shows the $\mathbf{H}$'s and numbers of multiplications associated to speedup techniques for various symmetries. In particular, UDLR- and bidiagonal symmetries enable us to use two butterfly stages, reducing the number of multiplications from N^4 to (approximately) $N^4/4$. Note that when $N \leq 8$, this number is smaller or comparable to $2N^3$, the number of multiplications required by general separable transforms. If the grid is pentasymmetric, the non-separable transform can be accelerated further. In addition, non-separable transforms generally have many more degrees of freedom than separable ones. Thus, using a non-separable symmetric graph may be more efficient than using separable graphs, even if the data does not always match exactly the symmetries we introduce here.

Non-separable transform with fast implementations can be applied based on data-driven or heuristic approaches. As discussed earlier, particular real-world data are associated to grids that are nearly symmetric, based on which computationally efficient GFTs can be derived. In addition, symmetry of grids is related to symmetry of blocks, so it is possible to design useful GFTs heuristically based on block symmetries. In Table 2 we show the percentages of nearly symmetric residual blocks from a number of intra-prediction modes. The residual blocks are extracted from *Kimono1* video using HEVC test model HM-16.9. The degrees of symmetry are measured by the absolute values of correlation between each block and its flipped versions (using corresponding axes/points of symmetry). Note that 36% of all the blocks considered in this experiment exhibited at least one kind of symmetry. This significant percentage for which block symmetries occur indicates that some 2D GFTs with symmetric grids are potentially useful. Future work will focus on how to identify such subsets of blocks, especially those with diagonal/anti-diagonal symmetry types, where we expect to gain more compression rates from non-separable transforms as compared to separable ones.

5. CONCLUSIONS

In this paper, we have explored various types of symmetry for 2D grids and discussed how block-diagonalization of the associated Laplacian matrices can be achieved. In particular, we have shown how diagonal symmetries can be exploited by using diagonal-first vertex reordering. We have summarized speedup techniques by showing relevant matrices $\mathbf{H}$'s and their resulting numbers of multiplications in Table 1, in which the reduced numbers of multiplications can also be compared with separable cases. Finally, we have discussed potential schemes for applications based on data-driven procedure or heuristics.

6. REFERENCES

[1] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 9–21, Sep 2001.

[2] G. Strang, "The Discrete Cosine Transform," *SIAM Rev.*, vol. 41, no. 1, pp. 135–147, 1999.

[3] J. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1874–1884, Apr 2012.

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

[6] G. Shen, W.-S. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," *Picture Coding Symposium (PCS)*, pp. 566–569, 2010.

[7] W. Hu, G. Cheung, , and A. Ortega, "Intra-prediction and generalized graph Fourier transform for image coding," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1913–1917, Nov 2015.

[8] H. E. Egilmez, A. Said, Y. H. Chao, and A. Ortega, "Graph-based transforms for inter predicted video coding," *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3992–3996, Sep. 2015.

[9] H. Chen, S. Zhu, and B. Zeng, "Design of non-separable transforms for directional 2-D sources," pp. 3697–3700, Sept 2011.

[10] L. Le Magoarou and R. Gribonval, "Are there approximate fast fourier transforms on graphs?," pp. 4811–4815, March 2016.

[11] L. Le Magoarou and R. Gribonval, "Flexible Multi-layer Sparse Approximations of Matrices and Applications," *IEEE Journal of Selected Topics in Signal Processing*, Jun 2016.

[12] K.-S. Lu and A. Ortega, "Symmetric Line Graph Transforms for Inter Predictive Video Coding," *to appear, Picture Coding Syposium (PCS)*, 2016.

[13] A. L. Andrew, "Eigenvectors of certain matrices," *Linear Algebra and its Applications*, , no. 7, pp. 151–162, 1973.

[14] A. Cantoni and P. Butler, "Eigenvalues and eigenvectors of symmetric centrosymmetric matrices," *Linear Algebra and its Applications*, vol. 13, no. 3, pp. 175–288, 1976.

[15] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[16] E. Pavez and A. Ortega, "Generalized Laplacian Precision Matrix Estimation for Graph Signal Processing," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6350–6354, 2016.

[17] K.-S. Lu and A. Ortega, "Toolbox: fast implementation of GFTs based on grid symmetries," 2017, [Online]. Available: https://github.com/STAC-USC/symmetric_grid.