A GRAPH LAPLACIAN MATRIX LEARNING METHOD FOR FAST IMPLEMENTATION OF GRAPH FOURIER TRANSFORM

Keng-Shih Lu and Antonio Ortega

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

ABSTRACT

In this paper, we propose an efficient graph learning approach for fast graph Fourier transform. We consider a maximum likelihood problem with additional constraints based on a matrix factorization of the graph Laplacian matrix, such that its eigenmatrix is a product of a block diagonal matrix and a butterfly-like matrix. We show that a special case of this problem reduces to a learning problem with constraints enforcing certain graph symmetries. Then, we provide an efficient approximation approach for the general problem without enforcing any symmetry constraint. We use this approach to design a fast nonseparable transform for intra predictive residual blocks in video compression. The resulting transform achieves a better rate-distortion performance than the 2D DCT and the hybrid DCT/ADST transform.

Index Terms— Graph Fourier transform, graph learning, fast algorithm, bisymmetric matrix

1. INTRODUCTION

Graph signal processing [1, 2] is a research field that studies signals supported on graphs, and extends classical signal processing techniques to provide flexibility or convenience. This research field has found applications in certain areas such as computer graphics, sensor networks [3], and transform coding for image and video compression [4, 5]. In particular, in transform coding, the coding efficiency of the Discrete Cosine Transform (DCT) can be justified [6] when the block pixels are modeled by a particular Gaussian Markov Random Field (GMRF) [7, 8]. Since the precision matrix (inverse covariance matrix) of a GMRF is also a generalized graph Laplacian matrix, each GMRF has a one to one correspondence to a graph. As DCT is shown to be the Graph Fourier Transform (GFT) of a uniform line graph [9], it can provide the optimal decorrelation when the blocks are modeled by the GMRF with precision matrix corresponding to the Laplacian of a uniform line graph. Similarly, the Asymmetric Discrete Sine Transform (ADST) can be viewed as a GFT, and the associated graph corresponds to a GMRF that is suitable for modeling intra residual blocks [5].

Data-driven graph Laplacian matrix learning [10, 11] is an important problem because for real-world signals the underlying graphs are not always available. In [12], a maximum-a-posteriori (MAP) problem with a Gaussian prior is considered, with a global smoothness constraint imposed to minimize the variations of the signals on the learn graphs. A more widely used formulation is a Gaussian maximum likelihood (ML) estimation with an l_1 regularization term that promotes graph sparsity [13]. This regularized problem can be solved using the graphical Lasso algorithm [14].

Fast implementations for GFTs are of great importance in graph signal processing. A GFT is given by the eigenmatrix of a Laplacian matrix, which requires a high computational cost if the number of nodes is large. In [15], a series of Givens rotations are used to approximate the KLT by a more efficient nonseparable transform. In [16, 17], the authors consider a sparse matrix factorization problem of the Laplacian matrix, whose solution gives rise to a GFT with a smaller number of multiplications. In our recent works [18, 19], we study particular graph symmetries that lead to fast implementations of GFTs using a butterfly structure and its variations. In this paper, we base on the results in [19], to learn graphs with symmetry properties such that they have fast GFTs. Then, we consider a more general block diagonalization problem for obtaining fast GFTs without any symmetry assumption. Our block diagonalization problem can also be viewed as a sparse matrix factorization problem in a general sense, but our approach differs from those previous works in two aspects. Firstly, rather than approximating a given GFT, our objective is to learn from data a graph that has a fast GFT. Secondly, the previous works either greedily solve Givens rotations or rely on a proximal algorithm for a nonconvex matrix factorization problem; however, we approach the problem by refining a Kronecker product approximation, whose optimal solution is based on a rank-1 approximation.

The rest of this paper is organized as follows. In Section 2 we introduce graph signal processing concepts and a lemma relevant to this work. In Section 3 we state a block diagonalization problem, whose solution provides a fast implementation. In Section 4 we formulate the graph learning problem with constraints for fast GFTs, and present a proposed method. In Section 5 we show experimental results on intra predictive residual blocks. Finally, we draw some conclusions in Section 6.

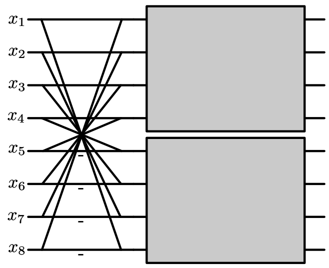

Fig. 1. A butterfly structure for fast implementation of a linear transform whose basis functions are either symmetric or skew-symmetric.

2. PRELIMINARIES

In graph signal processing, for a given graph, samples of the signal are associated to the nodes, and edges describe the inter-sample relations. If we denote a graph by $\mathcal{G}(\mathcal{V} = \{1, \dots, N\}, \mathcal{E})$, and the graph signal with N points by $\mathbf{x}$, then the (combinatorial) Laplacian matrix of $\mathcal{G}$ is defined as $\mathbf{L}_{\text{comb}} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ and $\mathbf{W}$ are the degree and the weighted adjacency matrix of $\mathcal{G}$, respectively. The generalized Laplacian matrix is defined by $\mathbf{L} = \mathbf{D}_{\text{SL}} + \mathbf{L}_{\text{comb}}$, where $\mathbf{D}_{\text{SL}}$ is a diagonal matrix containing the self-loop weights. The Graph Fourier transform (GFT) is defined using the eigenmatrix of the Laplacian matrix, $\mathbf{U}$, so that the i-th GFT coefficient of graph signal $\mathbf{x}$ is the i-th element of $\mathbf{U}^\top \mathbf{x}$.

In order to introduce an important block diagonalization form (2), we denote

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

where $\mathbf{I}_N$ is the $N \times N$ identity matrix. In (1), $\mathbf{J}_N$ is the order reversal permutation matrix, and $\mathbf{B}_N$ is a permuted and scaled version of the so-called butterfly (see Fig. 1), which leads to computational speedup, since the two following blocks operate on $N/2$ dimensional inputs.

Lemma 1. Let N be even. If an $N \times N$ matrix $\mathbf{L}$ is bisymmetric (symmetric around both diagonals), then it can be block diagonalized by $\mathbf{B}_N$ [20]:

$$\mathbf{L} = \mathbf{B}_N^\top \begin{pmatrix} \mathbf{A} + \mathbf{J}_{N/2}\mathbf{C} & \mathbf{O} \\ \mathbf{O} & \mathbf{A} - \mathbf{J}_{N/2}\mathbf{C} \end{pmatrix} \mathbf{B}_N. \qquad (2)$$

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

3. BLOCK DIAGONALIZATION OF CERTAIN LAPLACIAN MATRICES

Based on (2), when a graph with an even number of nodes N has certain symmetry properties, we are allowed to use

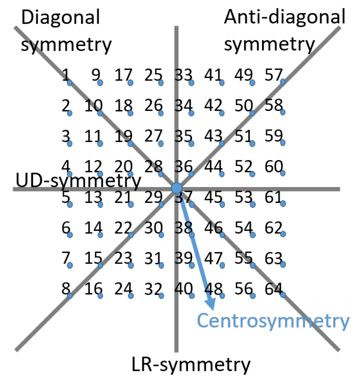

Fig. 2. The corresponding axis/point of symmetry for each grid symmetry type introduced in [19]. The numbers indicates that column-first node ordering in the 8×8 case.

butterfly-based fast implementations for the associated GFT [18, 19]. As a generalization of (2), we would like to write $\mathbf{L}$ as $\mathbf{L} = \mathbf{H}\mathbf{R}\mathbf{H}^\top$ with criteria C1 and C2:

C1 $\mathbf{H}$ is orthogonal, and each of its columns is a constant multiple of a vector, on which the projection can be implemented efficiently. That is, $\mathbf{H} = \mathbf{G}_\mathbf{H}\mathbf{D}_\mathbf{H}$, where $\mathbf{D}_\mathbf{H}$ is diagonal, $\mathbf{G}_\mathbf{H}$ has orthogonal columns, and $\mathbf{G}_\mathbf{H}^\top \mathbf{x}$ has fast implementation. Here are two specific criteria that allow fast implementations:

C1-A All entries in $\mathbf{G}_\mathbf{H}$ are 0, 1, or -1.

C1-B $\mathbf{G}_\mathbf{H}$ is sparse.

C2 $\mathbf{R}$ is block diagonal, and as sparse as possible, i.e., having many smaller blocks is preferable.

Let the eigenmatrix of $\mathbf{R}$ be $\mathbf{U}_\mathbf{R}$, then the GFT of signal $\mathbf{x}$ is $\mathbf{U}_\mathbf{R}^\top \mathbf{D}_\mathbf{H}^\top \mathbf{G}_\mathbf{H}^\top \mathbf{x}$. Note that multiplying by $\mathbf{G}_\mathbf{H}^\top$ is efficient, and $\mathbf{U}_\mathbf{R}^\top \mathbf{D}_\mathbf{H}^\top$ has the same block diagonal structure as $\mathbf{R}$ does. Thus, the sparsity of $\mathbf{R}$ enables reduction in the computation of this GFT. A special case based on (2) can be written as:

Corollary 2 ([18])**.** If $\mathcal{G}$ is a symmetric line graph, then its Laplacian matrix is bisymmetric. From Lemma 1, we know that the choice of $\mathbf{H}_{\text{line}} = \mathbf{B}_N^\top$ satisfies C1-A and gives an $\mathbf{R}$ having two nonzero diagonal block elements with size $N/2 \times N/2$.

In our recent work [19] we have introduced a number of grid symmetry types, as shown in Fig. 2. For a grid with each symmetry, or with multiple symmetries, we have presented (in Table 1 of [19]) an associated $\mathbf{H}$ that satisfies C1-A and leads to a corresponding block diagonal structure of $\mathbf{R}$. For example, if $\mathcal{G}$ is a grid graph with $N = n \times n$ nodes and is left-right symmetric, we have $\mathbf{H}_{\text{LR}} = \mathbf{B}_n^\top \otimes \mathbf{I}_n$, and $\mathbf{R}$ has two nonzero diagonal block elements with size $n^2/2 \times n^2/2$. Note that for grid graphs, we use column-first node ordering, as indicated by the numbers in Fig. 2.

4. GRAPH LEARNING FOR FAST GFT

Let $\mathbf{S} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i\mathbf{x}_i^\top$ be the correlation or covariance matrix obtained from the data $\mathbf{x}_i$'s. The Gaussian Maximum Likelihood (ML) estimation problem for the graph Laplacian precision matrix $\mathbf{L}$ is formulated as

$$\underset{\mathbf{L}\succeq 0,\ L_{i,j}\le 0,\ i\ne j}{\text{minimize}} \quad -\log\det(\mathbf{L}) + \text{trace}(\mathbf{LS}). \tag{3}$$

This problem can be solved using efficient iterative approaches in [10, 11]. Here, we would like to impose C1 and C2 on $\mathbf{L}$ such that its GFT has a fast implementation. Writing $\mathbf{L} = \mathbf{HRH}^\top$ with C1 and C2, we have $\det(\mathbf{L}) = \det(\mathbf{R})$ and $\text{trace}(\mathbf{LS}) = \text{trace}(\mathbf{HRH}^\top\mathbf{S}) = \text{trace}(\mathbf{R}(\mathbf{H}^\top\mathbf{SH}))$. The problem (3) can be rewritten as

$$\begin{aligned}
\underset{\mathbf{H},\mathbf{R}}{\text{minimize}} \quad & -\log\det(\mathbf{R}) + \text{trace}(\mathbf{R}(\mathbf{H}^\top\mathbf{SH}))\\
\text{subject to} \quad & \mathbf{R}\succeq 0,\ \left(\mathbf{HRH}^\top\right)_{i,j}\le 0 \text{ for } i\ne j\\
& \mathbf{H} \text{ satisfies C1, } \mathbf{R} \text{ satisfies C2.}
\end{aligned} \tag{4}$$

4.1. Solving (4) with a Particular Symmetry Type

If the objective is to obtain a graph with a given symmetry type, then, as given in [19], $\mathbf{H}$ is specified and the sizes of block diagonal elements of $\mathbf{R}$ are known. We denote $\mathbf{R}$ =diag$(\mathbf{R}_1,\ldots,\mathbf{R}_M)$, where element $\mathbf{R}_i$ has size $k_i\times k_i$ and $\sum_{i=1}^{M}k_i = N$. We also write $\mathbf{H}$ as $\mathbf{H} = (\mathbf{H}_1,\ldots,\mathbf{H}_M)$, where $\mathbf{H}_i$, with size $N\times k_i$, is a vertical slice of $\mathbf{H}$ containing k_i columns. Thus, $\mathbf{HRH}^\top = \sum_{l=1}^{M}\mathbf{H}_l\mathbf{R}_l\mathbf{H}_l^\top$, and

$$\left(\mathbf{HRH}^\top\right)_{i,j} = \mathbf{e}_i^\top\left(\sum_{l=1}^{M}\mathbf{H}_l\mathbf{R}_l\mathbf{H}_l^\top\right)\mathbf{e}_j = \sum_{l=1}^{M}\mathbf{h}(i,l)\mathbf{R}_l\mathbf{h}(j,l)^\top,$$

where $\mathbf{h}(i,l)$ denotes the i-th row of $\mathbf{H}_l$. We define $\mathbf{\Theta}$ as

$$\mathbf{\Theta} := \mathbf{H}^\top\mathbf{SH} = \begin{pmatrix}\mathbf{\Theta}_{1,1} & \mathbf{\Theta}_{1,2} & \ldots & \mathbf{\Theta}_{1,M}\\ \mathbf{\Theta}_{2,1} & \mathbf{\Theta}_{2,2} & \ldots & \mathbf{\Theta}_{2,M}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{\Theta}_{M,1} & \mathbf{\Theta}_{M,2} & \ldots & \mathbf{\Theta}_{M,M}\end{pmatrix},$$

where $\mathbf{\Theta}_{i,j}$ has size $k_i\times k_j$, then (4) can be reduced to a convex problem:

$$\begin{aligned}
\underset{\mathbf{R}_i\in\mathbb{R}^{k_i\times k_i}}{\text{minimize}} \quad & \sum_{l=1}^{M}\left[-\log\det(\mathbf{R}_l) + \text{trace}(\mathbf{R}_l\mathbf{\Theta}_{l,l})\right]\\
\text{subject to} \quad & \mathbf{R}_i\succeq 0,\ \sum_{l=0}^{M}\mathbf{h}(i,l)\mathbf{R}_l\mathbf{h}(j,l)^\top\le 0,\ i\ne j.
\end{aligned} \tag{5}$$

The number of variables is reduced from $N^2 = (\sum_{l=1}^{M}k_l)^2$ to $\sum_{l=1}^{M}k_l^2$. Since $\mathbf{H}$ is typically sparse, the constraints in (5) are simplified in many practical cases. Note that if $\mathbf{\Theta}_{i,j}$ is zero for all $i\ne j$, the solution of (5) is the same as that of (3). Indeed, when the specified $\mathbf{H}$ makes $\mathbf{\Theta}$ closer to block diagonal, the resulting Laplacian matrix is closer to that of (5).

Algorithm 1 Proposed Approximation Method for (4)

Require: $\mathbf{S}\in\mathbb{R}^{N\times N}$, a chosen factorization $N = N_1N_2$
Ensure: $(\mathbf{H},\mathbf{R})$ as a solution of (4)
1: Evaluate $\mathbf{S}_1$ and $\mathbf{S}_2$ such that $\mathbf{S}\approx\mathbf{S}_2\otimes\mathbf{S}_1$
2: Evaluate the eigenmatrix $\mathbf{E}_2$ of $\mathbf{S}_2$
3: $\mathbf{H}\leftarrow\mathbf{E}_2\otimes\mathbf{I}_{N_1}$
4: Solve (5) for $\mathbf{R}_i$, given $\mathbf{H}$ and $k_i = N_1$ for each i
5: $\mathbf{R}\leftarrow\text{diag}(\mathbf{R}_1,\ldots,\mathbf{R}_{N_2})$

In fact, whether or not the symmetry type is specified, the above always applies as long as $\mathbf{H}$ is given and the block diagonal structure of $\mathbf{R}$ is specified. Our approximation method below will be based on this fact.

4.2. Proposed Approximation Method for (4)

When $\mathbf{H}$ is not given, (4) is hard to solve. We propose an efficient approximation method as follows. We factor N as $N = N_1N_2$ and approximate $\mathbf{S}$ by the Kronecker product of two matrices of sizes $N_2\times N_2$ and $N_1\times N_1$:

$$\mathbf{S}\approx\mathbf{S}_2\otimes\mathbf{S}_1. \tag{6}$$

The optimal approximation in terms of Frobenius norm is given by the optimal rank-1 approximation of $\tilde{\mathbf{S}}$, which has size $N_2^2\times N_1^2$ and its entries are those of $\mathbf{S}$ with relocations [21, 22]. Thus, $\mathbf{S}_2$ and $\mathbf{S}_1$ can be obtained by solving the first left and right singular vectors of $\tilde{\mathbf{S}}$ followed by vector reshaping. We denote the eigendecomposition of $\mathbf{S}_2$ as $\mathbf{S}_2 = \mathbf{E}_2\mathbf{D}_2\mathbf{E}_2^\top$, then

$$\begin{aligned}
\mathbf{S} &\approx \left(\mathbf{E}_2\mathbf{D}_2\mathbf{E}_2^\top\right)\otimes\left(\mathbf{I}_{N_1}\mathbf{S}_1\mathbf{I}_{N_1}\right)\\
&= \left(\mathbf{E}_2\otimes\mathbf{I}_{N_1}\right)\left(\mathbf{D}_2\otimes\mathbf{S}_1\right)\left(\mathbf{E}_2\otimes\mathbf{I}_{N_1}\right)^\top.
\end{aligned} \tag{7}$$

If N_2 is small, $\mathbf{E}_2\otimes\mathbf{I}_{N_1}$ is sparse; thus, it is a legitimate matrix satisfying C1-B and approximately block-diagonalizes $\mathbf{S}$ to $\mathbf{D}_2\otimes\mathbf{S}_1$. Therefore, we pick $\mathbf{E}_2\otimes\mathbf{I}_{N_1}$ as the $\mathbf{H}$ matrix, and use the block structure of $\mathbf{D}_2\otimes\mathbf{S}_1$, to solve (5) and obtain $\mathbf{R}$ which has $M = N_2$ nonzero $N_1\times N_1$ diagonal blocks.

We summarize this proposed method in Algorithm 1. For the choice of N_1 and N_2, we note that it is preferable to choose similar values as N_1 and N_2. With the resulting $\mathbf{H}$ and $\mathbf{R}$ matrices, the number of multiplications required for the fast GFT is $N(N_1+N_2)$, which is smaller when N_1 and N_2 are closer. In addition, when N_1 and N_2 are closer, the degree of freedom in $\mathbf{S}_1\otimes\mathbf{S}_2$ is larger, giving a potentially better approximation in (6).

When the data are pixel blocks with size $N_1\times N_2$, the eigenmatrices $\mathbf{E}_1$ and $\mathbf{E}_2$ of $\mathbf{S}_1$ and $\mathbf{S}_2$ characterize the column and row transforms in a separable scheme. Therefore, the GFT obtained by the proposed method can be regarded as a transform modified from a separable transform. As the matrix $\mathbf{D}_2\otimes\mathbf{S}_1$ in (7) is modified by the solver of (5), this Kronecker product is modified to a general block diagonal

Table 1. Bit rate reduction of separable KLT (Sep-KLT), fast GFT, and hybrid DCT/ADST (Hybrid) as compared to the 2D separable DCT, tested on residual blocks with intra mode-2 (M2) and mode-16 (M16). Positive values mean compression gain, measured in the Bjontegaard metric (percentage of bit rate reduction). BDrill is meant to be *BasketballDrill*.

	Sep-KLT		fast GFT		Hybrid	
	M2	M16	M2	M16	M2	M16
BDrill	11.9	8.6	14.5	10.2	3.4	2.5
BQMall	9.7	6.3	12.9	8.2	0.7	1.3
Crew	17.5	14.8	20.7	16.2	3.2	1.7
Harbour	26.6	24.9	29.7	23.0	1.7	3.2
Ice	28.4	11.8	35.5	16.8	3.7	1.7
Soccer	14.6	12.7	15.6	11.2	3.9	1.5
Average	13.2	10.4	20.2	11.8	2.8	2.1

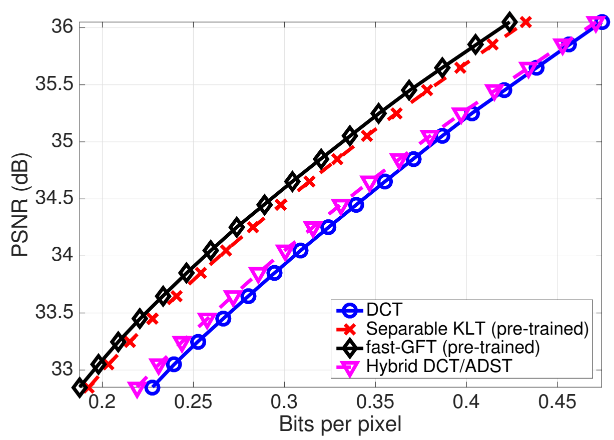

Fig. 3. Rate-distortion performance of separable KLT, fast GFT, and hybrid DCT/ADST. The testing blocks in this figure are mode-2 intra residual blocks.

matrix with a given structure, allowing more freedom. The fast GFT can be realized by a common row transform applied to all rows followed by different column transforms applied to different columns[1]. The number of multiplications is the same as a separable transform. The number of required coefficients is $NN_1 + N_2^2$, which is between the numbers in the separable case ($N_1^2 + N_2^2$) and the nonseparable case (N^2).

5. EXPERIMENTAL RESULTS

Here we apply the proposed learning method to transform coding. We extract intra-predictive residual blocks from test video sequences *BasketballDrill, BQMall, City, Crew, Harbour, Ice* and *Soccer*, using HEVC test model HM-16.9. We train and test the transforms on 8×8 luma residual blocks with intra prediction mode-2 (HOR+8) and mode-16 (HOR-6), which have nearly diagonal directions of prediction. We train a fast GFT for mode-2 (resp. mode-16) based on Algorithm 1, using mode-2 (resp. mode-16) blocks in the *City* sequence as the training set. The convex problem (4) is solved using CVX [23] toolbox. Then, we test the results on mode-2 (resp. mode-16) blocks in the other 6 sequences, not including the one for training.

To compare the results, we consider other transforms with comparable or lower computational costs: the 2D separable DCT, separable KLT, and the mode-dependent hybrid DCT/ADST transform. In the hybrid scheme, we apply a mode-dependent combination of DCT and ADST, as suggested in [24], for the row and column transform. Each KLT (for mode-2 and mode-16, respectively), is pre-trained from the same training set as the fast GFT. The transform coefficients are uniformly quantized using various quantization steps that yield PSNR levels in the range of 29-38 dB. The quantized coefficients are encoded using the AGP codec [25], which is an entropy coding method adaptive to the distribution of transform coefficients, so it provides a fair comparison for the rate-distortion (RD) performances.

Table 1 shows the coding gain in BD-rate (percentages of bit rate reduction) over 2D DCT. The RD plot for mode-2 blocks is shown in Fig. 3. The resulting fast GFT achieves an average of 20.2% bit rate gain over DCT for mode-2 blocks, and 11.8% for mode-16 blocks. It also outperforms the hybrid transform for each sequence/mode. In most cases, the fast GFT can also provide a higher coding gain than the pre-trained separable KLT. A probable reason is that those constraints forcing the transform to be a GFT provide regularizations and prevent overfitting, while a separable KLT learned and tested on different sets may suffer from overfitting.

6. CONCLUSIONS

In this paper, we have introduced a graph Laplacian learning problem with constraints for fast implementations. We have established criteria C1 and C2, based on which a graph Laplacian matrix is associated to a fast GFT. We have imposed constraints corresponding to those criteria on a maximum likelihood estimation problem, to learn fast GFTs. We have shown that given some conditions related to graph symmetries, the ML problem reduced to a convex problem. Then, we have proposed an approximation method, which solves a Kronecker product factorization followed by the reduced convex problem. We have applied our method to transform coding with intra predictive residual blocks. The experimental results show that for particular intra modes, the pre-trained GFT outperforms 2D DCT and the hybrid DCT/ADST in terms of RD performance.

[1] Note that when the input covariance matrix has row-first node ordering, the same method will yield an approximate transform with a common column transform and different row transforms.

7. REFERENCES

[1] 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.

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

[3] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1085–1089.

[4] 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.

[5] 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.

[6] C. Zhang and D. Florencio, "Analyzing the Optimality of Predictive Transform Coding Using Graph-Based Models," *IEEE Signal Processing Letters*, vol. 20, pp. 106–109, Jan. 2013.

[7] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, vol. 104 of *Monographs on Statistics and Applied Probability*, Chapman & Hall, London, 2005.

[8] C. Zhang, D. Florencio, and P. A. Chou, "Graph Signal Processing - A Probabilistic Framework," Tech. Rep., April 2015.

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

[10] 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.

[11] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph Learning From Data under Structural and Laplacian Constraints," *submitted*.

[12] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian Matrix in Smooth Graph Signal Representations," *IEEE Trans. Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[13] B. M. Lake and J. B. Tenenbaum, "Discovering structure by learning sparse graph," in *Proceedings of the 33rd Annual Cognitive Science Conference*, 2010.

[14] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2007.

[15] H. Chen and B. Zeng, "Design of Low-complexity, Non-separable 2-D Transforms Based on Butterfly Structures," in *2012 IEEE International Symposium on Circuits and Systems*, May 2012, pp. 2921–2924.

[16] L. L. Magoarou and R. Gribonval, "Are there approximate fast fourier transforms on graphs?," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4811–4815, March 2016.

[17] L. L. Magoarou, R. Gribonval, and N. Tremblay, "Approximate Fast Fourier Transforms on Graphs via Multilayer Sparse Approximations," *submitted*.

[18] K.-S. Lu and A. Ortega, "Symmetric Line Graph Transforms for Inter Predictive Video Coding," in *Picture Coding Syposium (PCS)*. IEEE, 2016, pp. 1–5.

[19] K.-S. Lu and A. Ortega, "Fast Implementation for Symmetric Non-separable Transforms Based on Grids," *to appear, International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[20] 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.

[21] N. P. Pitsianis, *The Kronecker Product in Approximation and Fast Transform Generation*, Ph.D. thesis, Ithaca, NY, USA, 1997, UMI Order No. GAX97-16143.

[22] C. F. Van Loan and N. Pitsianis, *Approximation with Kronecker Products*, pp. 293–314, Springer Netherlands, Dordrecht, 1993.

[23] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," `http://cvxr.com/cvx`, Mar. 2014.

[24] A. Saxena and F. C. Fernandes, "DCT/DST-Based Transform Coding for Intra Prediction in Image/Video Coding," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3974–3981, Oct 2013.

[25] A. Said and W. A. Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," in *Proceedings of IEEE International Symposium on Information Theory*, Jun 1997, pp. 61–73.