LEARNING SEPARABLE TRANSFORMS BY INVERSE COVARIANCE ESTIMATION

Eduardo Pavez, Antonio Ortega* and Debargha Mukherjee†*

* University of Southern California, Los Angeles, CA, USA
† Google, Inc., Mountain View, CA, USA

ABSTRACT

Orthogonal transforms are one of the most important components of a video encoder system. They are applied to residual block images obtained as the difference between a target and its prediction. In this paper we propose a framework to design separable transforms from prediction residual statistics. We model the data as a 2D Gaussian Markov random field and approximate its inverse covariance by a matrix with a separable structure, thus explicitly constructing a separable orthonormal matrix that approximates the KLT. Our designed transforms can adapt to prediction residual statistics, have low complexity (compared to non separable transforms), require selecting few parameters and outperform hybrid DCT/ADST separable transform for intra coding of AV1 residuals.

Index Terms— separable transform, transform learning, directional transform, precision matrix, video coding

1. INTRODUCTION

Orthonormal transforms are fundamental to video compression. They are applied to rectangular residual image blocks obtained as the difference between a target (for coding) image block and its prediction from a neighboring one, either from the same frame (intra) or from a previous frame (inter). The transformed coefficients are then uniformly quantized and entropy coded [1].

Coding of prediction residuals can be improved by adapting the transform to different statistical characteristics [2]. For a family of signals with a known probability distribution, the optimal transform for energy compaction and decorrelation of coefficients are the eigenvectors of their covariance matrix, which is called the Karhunen Loeve Transform (KLT). The KLT is not practical because it does not, in general, have an efficient implementation, and requires estimating a large number of parameters. A more practical approach is taken in video encoders, where there is a set of fixed transforms, which are applied to a prediction mode according to some pre-specified rule or rate-distortion criteria, for example in High Efficiency Video Coding standard (H.265/HEVC) [3] and Google's VP10/AV1 [4] various DCT/DST based transforms are used for intra coding.

Compression efficiency has to be leveraged against low computational complexity, which can be achieved by using one dimensional transforms with fast implementations. A transform for two dimensional signals is constructed by applying the one dimensional transforms to rows and columns of a residual image. This method produces what is called *separable transform*. The most used one dimensional transforms to construct two dimensional transforms are the Discrete Cosine Transform (DCT-2)[5] and Discrete Sine Transform (DST-4, and DST-7). They correspond to KLTs of some simple Gaussian processes that model prediction residuals [6, 2].

In this paper we propose a framework to learn two dimensional separable transforms from data. We model the residuals image blocks with a Gaussian Markov Random Field (GMRF) distribution, whose inverse covariance matrix is diagonalized by a separable transform. We design the separable transform using maximum likelihood estimation under this model. We show that our formulation corresponds to a bi-convex optimization problem, which can be solved using an alternating minimization method.

We also propose the family of parametric Discrete Trigonometric Transforms (p-DTT) as a natural data adaptive extension of the DCT and DST. Each p-DTT is generated as the eigenvectors of the Generalized Laplacian [7] matrix of a line graph with unit edge weights and real valued self loop weights in the first and last vertices. The p-DTTs are used as row and column transforms of the designed separable transform. The solution of our transform learning method with p-DTT constraints leads us to data adaptive transforms with very few parameters (4 in total, 2 for each p-DTT). Although we only design p-DTTs, our framework can be applied to design transforms with other characteristics, such as the fast transforms on line graphs from [8]. We apply our methodology to design transforms for each intra prediction mode of the AV1 video encoder. The resulting transforms are separable and directional, and show promising coding gains of intra prediction residuals.

Design of separable tranforms also has been considered before in [9], where the row and column transform are designed independently using training data obtained from intra prediction residuals. The method estimates each transform using Singular Value Decomposition. A closer method to ours is [10], where the rows and columns of prediction residuals are modeled with a one dimensional GMRF represented by a line graph. Then for each family of residuals (e.g. an intra prediction mode) one row and one column transform is designed by estimating the inverse covariance of the row and column GMRFs. Then a separable transform is constructed by combining the one dimensional transforms. The main difference with our learning approach, is that we model the image residuals as a two dimensional GMRF, and we estimate a separable transform that approximately diagonalizes its precision matrix. We can view the method from [10] as learning the row and column transforms independently, while our method learns them jointly. A mode adaptive transform for intra coding called directional DCT was proposed in [11], which can be described as a standard DCT transform applied along groups of pixels scanned following certain directions. That transforms requires an edge detector algorithm to decide the direction to be applied. Another similar approach idea is the Steerable 2D DCT [12], which uses the fact that 2D DCT correspond to eigenvectors of certain regular graphs, and exploits multiplicity of some eigenspaces to adapt to orientations.

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

This paper is organized as follows, in Section 2 we introduce the separable transform framework, our GMRF model, and the transform learning algorithm. In Section 3 we introduce the parametric Discrete Trigonometric Transforms. We show comparison between our transforms and DCT/DST hybrid transforms for intra coding of AV1 residuals in Section 4. Finally we conclude and propose future extensions of this work in Section 5.

2. 2D SEPARABLE GMRF, MODELS AND ALGORITHMS

In this section we introduce our proposed separable GMRF model. Then we discuss the maximum likelihood estimation of the separable inverse covariance matrix. We end with a proposed algorithmic solution and discuss some implementation details.

2.1. Separable 2D GMRF model

Consider arbitrary orthonormal $n \times n$ transforms $\mathbf{U}_c, \mathbf{U}_r$, where r and c stand for row and column wise application. For a given $n \times n$ image block $\mathbf{X}$, its transform coefficients are given by $\hat{\mathbf{X}} = \mathbf{U}_c^T \mathbf{X} \mathbf{U}_r$, which using properties of the Kronecker product $\otimes$ can we written in vector form as

$$\text{vect}\,(\hat{\mathbf{X}}) = (\mathbf{U}_r \otimes \mathbf{U}_c)^T \,\text{vect}\,(\mathbf{X}). \quad (1)$$

The separable 2D transform is by the matrix $\mathbf{U} = \mathbf{U}_r \otimes \mathbf{U}_c$. We assume the vectorized residual $\mathbf{x} = \text{vect}\,(\mathbf{X})$ is distributed as a zero mean Gaussian process with $n^2 \times n^2$ dimensional covariance matrix $\mathbf{\Sigma}$ and precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$. Its pdf is given by

$$p(\mathbf{x}) = \frac{\sqrt{\det(\mathbf{\Theta})}}{(2\pi)^{n^2/2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T \mathbf{\Theta} \mathbf{x}\right\}.$$

Ideally, we would like the separable transform $\mathbf{U}$ to be the eigenvectors of the covariance matrix, i.e., the KLT. By imposing that the inverse covariance matrix $\mathbf{\Theta}$ is diagonalized by the separable transform $\mathbf{U}$, the precision matrix $\mathbf{\Theta}$ can be factorized as follows

$$\begin{aligned}\mathbf{\Theta} &= (\mathbf{U}_r \otimes \mathbf{U}_c)(\mathbf{\Lambda}_r \otimes \mathbf{\Lambda}_c)(\mathbf{U}_r \otimes \mathbf{U}_c)^T \\ &= (\mathbf{U}_r \mathbf{\Lambda}_r \mathbf{U}_r^T) \otimes (\mathbf{U}_c \mathbf{\Lambda}_c \mathbf{U}_c^T) \\ &= \mathbf{M}_r \otimes \mathbf{M}_c. \end{aligned} \quad (2)$$

Where $\mathbf{\Theta} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and the eigenvalues satisfy $\mathbf{\Lambda} = \mathbf{\Lambda}_r \otimes \mathbf{\Lambda}_c$ for some diagonal $n \times n$ matrices $\mathbf{\Lambda}_r, \mathbf{\Lambda}_c$. Then $\mathbf{M}_r = \mathbf{U}_r \mathbf{\Lambda}_r \mathbf{U}_r^T$ and $\mathbf{M}_c = \mathbf{U}_c \mathbf{\Lambda}_c \mathbf{U}_c^T$ correspond to square $n \times n$ matrices whose eigenvectors are the row and column transforms respectively.

Designing a separable transform that matches the KLT of a 2D GMRF process is equivalent to finding matrices $\mathbf{M}_r, \mathbf{M}_c$. We assume $(\mathbf{M}_r, \mathbf{M}_c) \in \mathcal{M} \times \mathcal{M}$, where $\mathcal{M} \subset \mathbb{S}_n^+$ is a convex set.

2.2. Separable inverse covariance estimation

Consider a set of $n \times n$ residual image blocks $\{\mathbf{X}_i\}_{i=1}^N$, assume they are independent, and their vectorized versions $\mathbf{x}_i = \text{vect}(\mathbf{X}_i)$ follow a GMRF distribution with inverse covariance matrix $\mathbf{\Theta}$. Then the negative log-likelihood $-\log(p(\mathbf{x}_1, \cdots, \mathbf{x}_N))$ is

$$\begin{aligned}-\sum_{i=1}^N \log(p(\mathbf{x}_i)) &= -N \log\left(\frac{\sqrt{\det(\mathbf{\Theta})}}{(2\pi)^{n^2/2}}\right) + \frac{1}{2}\sum_{i=1}^N \mathbf{x}_i^T \mathbf{\Theta} \mathbf{x}_i \\ &= -\frac{N}{2}\log\det(\mathbf{\Theta}) + \frac{N}{2}\text{tr}(\mathbf{\Theta}\mathbf{S}) + const. \end{aligned} \quad (3)$$

And $\mathbf{S} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T$ is the maximum likelihood estimator of the covariance matrix. By incorporating the separable structure in $\mathbf{\Theta}$ the maximum likelihood estimation of the separable precision matrix corresponds to the following optimization problem

$$\min_{(\mathbf{M}_r, \mathbf{M}_c) \in \mathcal{M} \times \mathcal{M}} -\log\det(\mathbf{M}_r \otimes \mathbf{M}_c) + \text{tr}((\mathbf{M}_r \otimes \mathbf{M}_c)\mathbf{S}). \quad (4)$$

The Kronecker product factorization of the inverse covariance matrix makes (4) a type of non convex optimization problem called bi-convex [13].

Proposition 1. *The optimizaton problem in (4) is bi-convex .*

Proof. We have to show, that for a fixed $\mathbf{M}_c$, the function is convex in $\mathbf{M}_r$, and viceversa. Using basic properties of Kronecker products, the $\log\det$ term can be decomposed as

$$\log\det(\mathbf{M}_r \otimes \mathbf{M}_c) = n\log\det(\mathbf{M}_r) + n\log\det(\mathbf{M}_c),$$

which is a sum of concave functions. The trace term is a bi-linear function, and since we consider $(\mathbf{M}_r, \mathbf{M}_c) \in \mathcal{M} \times \mathcal{M}$, the problem is bi-convex. ☐

Bi-convex problems can be solved by alternating minimization [13]. We show our transform learning method using that approach in Algorithm 1. Since each step inside the loop of algorithm 1 is a small dimensional convex problem, it can be implemented using general convex optimization solvers.

Algorithm 1 Separable transform algorithm

Require: $\{\mathbf{X}_i\}_{i=1}^N$, $\mathbf{M}_c^{(0)}$.
1: Estimate $\mathbf{S}$.
2: **while** not converged **do**
3: $\quad \mathbf{M}_r^{(t+1)} \leftarrow \arg\min_{\mathbf{M} \in \mathcal{M}} -\log\det(\mathbf{M} \otimes \mathbf{M}_c^{(t)}) + \text{tr}((\mathbf{M} \otimes \mathbf{M}_c^{(t)})\mathbf{S})$ (row transform update)
4: $\quad \mathbf{M}_c^{(t+1)} \leftarrow \arg\min_{\mathbf{M} \in \mathcal{M}} -\log\det(\mathbf{M}_r^{(t+1)} \otimes \mathbf{M}) + \text{tr}((\mathbf{M}_r^{(t+1)} \otimes \mathbf{M})\mathbf{S})$ (column transform update)
5: $\quad t \leftarrow t + 1$
6: **end while**

3. PARAMETRIC DISCRETE TRIGONOMETRIC TRANSFORMS

In previous sections we described a general framework for learning separable 2D transforms. In this section we propose a constraint set $\mathcal{M}$, which leads to one dimensional transforms called parametric Discrete Trigonometric Transforms (p-DTTs).

Definition 1 (p-DTT)**.** *We say the orthogonal matrix* $\mathbf{U}$ *is a p-DTT if it satisfies* $\mathbf{M}(\mathbf{a}) = \mathbf{U}\mathbf{\Omega}\mathbf{U}^T$ *where*

$$\mathbf{M}(\mathbf{a}) = \begin{bmatrix} a_1 & -a_2 & & & \\ -a_2 & a_3 & -a_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & a_3 & \\ & & & -a_2 & a_4 \end{bmatrix},$$

$\mathbf{\Omega} = \text{diag}(\omega_1, \cdots, \omega_n)$ *is a matrix of eigenvalues sorted in non decreasing order, and* $\mathbf{a} = [a_1, a_2, a_3, a_4] \geq \mathbf{0}$.

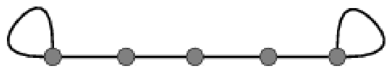


Fig. 1: Line graph with self loops

	$\beta = 0$	$\beta = 1$	$\beta = 2$
$\alpha = 0$	DCT-2	DCT-8	DCT-4
$\alpha = 1$	DST-7	DST-1	DST-5
$\alpha = 2$	DST-4	DST-6	DST-2

Table 1: Self loop weights that produce known DTTs

We can always write $\mathbf{M}(\mathbf{a}) = \gamma\mathbf{I} + \delta\mathbf{L}(\alpha, \beta)$ where

$$\mathbf{L}(\alpha, \beta) = \text{diag}(\alpha, 0, \cdots, 0, \beta) + \mathbf{L}_{DCT} \tag{5}$$

$$= \begin{bmatrix} \alpha & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & \beta \end{bmatrix} + \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 2 & \\ & & & -1 & 1 \end{bmatrix},$$

and α, β are called vertex weights and represented self loops in a graph. The matrix $\mathbf{M}(\mathbf{a})$ and $\mathbf{L}(\alpha, \beta)$ have the same eigenvectors[1], then all p-DTTs can be described by those matrices. The matrix $\mathbf{L}_{DCT}$ corresponds to the combinatorial Laplacian of a line graph with unit weights, and $\mathbf{L}(\alpha, \beta)$ is the generalized Laplacian [7] of a line graph with unit weights and self loops in first and last vertex as shown in figure 1. The p-DTTs include the Discrete Trigonometric Transforms from [14] and [15], for example for $\alpha = \beta = 0$ we obtain the DCT-2. Other popular transforms used in video coding are the DST-4 or ADST $\alpha = 2, \beta = 0$, and the DST-7 with $\alpha = 1, \beta = 0$. We show in table 1 all the DTTs that can be represented with an undirected graph. The sign and sparsity pattern of the p-DTTs produce GMRFs where pixels are connected to their horizontal and vertical neighbors with positive weights, and to their diagonal neighbors with negative weights.

Since the scaling factors γ and δ are inversely proportional to the energy level of the residual images, for optimization purposes we will consider matrices in the form $\mathbf{M}(\mathbf{a})$, but for analysis of the resulting transforms and in terms of number of parameters, we only need to know (α, β).

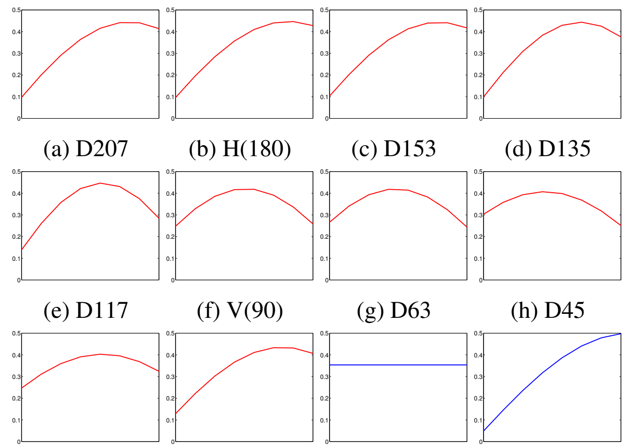
4. EXPERIMENTS

In this section we design separable transforms, and apply them to transform coding of intra prediction residuals collected from the AV1[2] codec. AV1 supports 10 intra prediction modes, a DC mode, a True Motion (TM) mode and 8 directional prediction modes. The prediction units can be rectangular of sizes $i \times j$ with $i, j \in \{4, 8, 16, 32, 64\}$, and the transform is applied to square blocks of size $n \times n$ where $n \leq \min(i, j)$. For each prediction block there is a fixed transform size, for instance a prediction block might have size 8×16, and will be encoded using a transform on blocks of size 4×4. We construct a dataset of intra prediction residuals from 48 SD videos from the derf[3] dataset, we consider all videos and use resolution at most CIF. We randomly pick 40 of those videos to form a training set for transforms learning, and the rest of the videos is used as test set for coding experiments. The training and testing data are organized according to transform block sizes 4×4 , 8×8 and 16×16 and prediction mode.

[1]shift by identity and multiplication by a constant only change the eigenvalues.

[2]We use the nextgenv2 branch of the webm/libvpx project.

[3]https://media.xiph.org/video/derf/



(a) D207 (b) H(180) (c) D153 (d) D135 (e) D117 (f) V(90) (g) D63 (h) D45 (i) DC (j) TM (k) DCT (l) ADST

Fig. 2: First basis functions of Row transforms of size 8 for different intra prediction modes.

4.1. Transform Learning

For each prediction mode and block size, we compute an empirical covariance matrix and learn a separable transform, where each one dimensional transform is a p-DTT. In Figure 2 we show the first basis functions of the row transforms obtained using our method. We can observe that for the DC mode the basis function is flatter, resembling the first eigenvector of the DCT-2, while for the horizontal prediction mode (H), the basis function vanishes on the left side and increases on the right side resembling the ADST. All other basis functions have a *valley* shape, most notably for the vertical prediction mode (V) where the basis function is most symmetric. When there is horizontal prediction the residual error is smaller in the leftmost pixels, thus a transform with a basis function that increases away from the boundary can represent better that type of residual signal. The self loop weights in the first and last vertices force the first basis function to have smaller magnitude near the image boundary, and as we can see from Figure 2 this behavior is more pronounced for the closer to horizontal prediction modes.

4.2. Coding of residuals

To evaluate the new separable transforms we perform transform coding of prediction residuals of our test set which consists of videos *galleon_422_cif, mobile_calendar_422_cif, football_422_cif, container_cif, washdc_422_cif, highway_cif, crew_cif, husky_cif* . Since each 1D transform corresponds to eigenvectors of a generalized Laplacian matrix, they are graph transforms [16]. We sort the eigenvectors according to their graph frequencies (i.e., eigenvalues of the generalized Laplacian matrix) in increasing order of magnitude, hence, when applying the 2D transform to a square residual image block, the low frequency coefficients will be concentrated in the top-left corner of the transformed block. The transformed coefficients are uniformly quantized with step sizes in $\{18, 22, 26, \cdots, 54\}$, then sorted in zig-zag scanning order and entropy coded using the

	DC	TM	D207	H	D153	D135	D117	V	D63	D45
4×4	0.1356 -1.6721	0.5175 -6.7930	0.2399 -3.1381	0.1162 -1.6428	0.3783 -5.4830	0.4480 -6.3712	0.4054 -6.3086	0.1808 -2.7140	0.2805 -4.0909	0.1571 -2.0844
8×8	-0.0090 0.1189	0.3046 -4.2149	0.0857 -1.0866	0.0669 -0.9680	0.1892 -2.5098	0.1632 -2.2399	0.2490 -3.9131	0.0399 -0.6592	0.0417 -0.5866	0.1166 -1.4594
16×16	-0.0891 1.5292	0.1358 -2.4965	-0.0538 0.8696	0.0209 -0.3198	-0.0170 0.3530	0.2033 -2.8678	0.3562 -6.4577	-0.0783 1.2928	-0.0825 1.2355	-0.1041 1.6319

Table 2: Bjontegaard gains of p-DTT coding versus best per mode DCT/ADST/I hybrid transform. Each box contains BD-PSNR (top) and BD-bitrate % reduction (bottom).

	DC	TM	D207	H	D153	D135	D117	V	D63	D45
4×4	0.0136 -0.1710	0.0659 -0.9249	0.0038 -0.0406	0.0027 -0.0376	0.0130 -0.1955	0.1192 -1.7407	-0.0073 0.1190	0.0055 -0.0757	0.0033 -0.0508	0.0121 -0.1581
8×8	0.0060 -0.0873	0.0991 -1.4436	0.1054 -1.3392	0.0742 -1.0787	0.0389 -0.5332	0.1358 -1.8439	0.0363 -0.5391	0.0068 -0.1254	0.0632 -0.8896	0.1197 -1.5001
16×16	-0.0777 1.3339	0.0598 -1.1166	0.2751 -4.2836	0.1367 -2.2352	0.1796 -2.8069	0.3850 -5.6491	0.1163 -2.1413	0.2637 -4.2086	0.4242 -6.4251	0.0856 -1.3422

Table 3: Bjontegaard gains of p-DTT coding versus GBST. Each box contains BD-PSNR (top) and BD-bitrate % reduction (bottom).

algorithm from [17], which adapts to the distribution of the transformed coefficients. We consider transform blocks of size 4×4, 8×8 and 16×16 for all intra prediction modes and compare against all hybrid 2D transforms constructed from row-column transforms in $\{$DCT-2, ADST, I$\}$ (except identity transform in rows and columns, which correspnds to transform skip), where I stands for identity transform, i.e. no application of row/column transform. In table 2 we show the coding gains measured with BD-PSNR gains BD-bitrate reduction [18]. For transform block sizes 4×4 we observe significant gains (positive BD-PSNR and negative BD-bitrate) for all prediction modes. More notable in the TM mode there is a gain of 0.5 dB while for the directional prediction modes in between horizontal and vertical directions (D153, D135 and D117), the gains are in the range $0.38 - 0.45$ dB. The smaller gain on the D45 mode is likely due to the smaller probability of choosing that prediction direction, resulting in much less training data to accurately capture its statistics. In the case of DC, V and H modes, the DCT/ADST hybrid transforms are known to perform very well, thus the gains of our transforms are smaller. For transform block sizes 8×8 we observe significant gains in all modes except the DC prediction mode, and again good gains around 0.2 dB are obtained for D153, D135 and D117 modes. For block size 16×16 the behavior is less conclusive, although we again observe good improvements in the D135, D117 and TM modes. When there are gains, they are more significant in directional prediction modes that are not horizontal or vertical. In these cases the hybrid DCT/ADST separable transforms are not well suited for diagonal prediction directions, thus the coding gain in adapting the p-DTT weights to the data is higher. As for the poor performance in 16 block sizes, it is most likely due to the higher signal dimension and lower amount of data used for training and testing. This occurs because the videos we used to construct our dataset have low resolution (CIF), thus the transform block sizes chosen by the encoder are small most of the time. We leave more in depth experiments with different data sets created using videos at varying resolutions for future work.

4.3. Comparison with GBST

The Graph Based Separable Transforms (GBST) [10] are constructed using row and column transforms corresponding to eigenvectors of Generalized Graph Laplacian (GGL) matrices of line graphs. The row and column GGLs are estimated independently using the graph learning method from [19]. The GBST are associated to graphs with self loops in all vertices and arbitrary edge weights, our p-DTTs have equal edge weights and we allow self loop weights only in the first and last vertices. Therefore the GBST requires estimating $2(2n-1)$ parameters for each prediction mode, while our separable transform only require estimating 8 parameters (4 for each p-DTT). To learn the GBST we construct row and column covariance matrices from the AV1 intra prediction residuals of our training set. We show the BD-PSNR and BD-bitrate reduction of our method with respect to the GBST in Table 3. The proposed transforms outperform the GBST in for all block sizes and prediction modes except for 16×16 DC mode and 4×4 D117 mode. For smaller block sizes, the advantage of our method is only significant for the D135 mode (0.1 dB gain), while for the rest the gains are insignificant. For block size of 16×16, our method achieves significant gains for directional prediction modes. Even though our method estimates a separable transform with less number of parameters, its coding performance is at least as good as the GBST.

5. CONCLUSION

In this paper we proposed a novel method for learning separable transforms using a inverse covariance estimation formulation. We also proposed the family of parametric Discrete Trigonometric Transforms (p-DTTs) as a data adaptive generalization of the DCT and DST. By using our learning method with p-DTT constraints on prediction residuals collected from the AV1 video coder we design transforms that: 1) are separable, 2) have few parameters, and 3) outperform hybrid DCT/ADST/I transforms in compression of intra prediction residuals. In particular, for diagonal intra prediction modes where hybrid DCT/ADST separable transforms are not well suited, our transforms show higher coding gains.

6. ACKNOWLEDGMENTS

We thank Urvang Joshi and Hui Su for providing code to collect intra prediction residuals.

7. REFERENCES

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

[2] J. Han, A. Saxena, and K. Rose, "Towards jointly optimal spatial prediction and adaptive transform in video/image coding," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 726–729.

[3] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the hevc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, 2012.

[4] D. Mukherjee, H. Su, J. Bankoski, A. Converse, J. Han, Z. Liu, and Y. Xu, "An overview of new video coding tools under consideration for vp10: the successor to vp9," in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2015, pp. 95 991E–95 991E.

[5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[6] R. Clarke, "Relation between the karhunen loeve and cosine transforms," in *IEE Proceedings F-Communications, Radar and Signal Processing*, vol. 128, no. 6. IET, 1981, pp. 359–360.

[7] T. Biyikoglu, J. Leydold, and P. F. Stadler, "Laplacian eigenvectors of graphs," *Lecture notes in mathematics*, vol. 1915, 2007.

[8] K.-S. Lu and A. Ortega, "Symmetric line graph transforms for inter predictive video coding," in *2016 Picture Coding Symposium (PCS)*, 2016, pp. –.

[9] Y. Ye and M. Karczewicz, "Improved h. 264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 2116–2119.

[10] H. E. Egilmez, Y. H. Chao, A. Ortega, B. Lee, and S. Yea, "Gbst: Separable transforms based on line graphs for predictive video coding," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 2375–2379.

[11] B. Zeng and J. Fu, "Directional discrete cosine transforms a new framework for image coding," *IEEE transactions on circuits and systems for video technology*, vol. 18, no. 3, pp. 305–313, 2008.

[12] G. Fracastoro and E. Magli, "Steerable discrete cosine transform," in *Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on*. IEEE, 2015, pp. 1–6.

[13] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.

[14] M. Püschel and J. M. Moura, "The algebraic approach to the discrete cosine and sine transforms and their fast algorithms," *SIAM Journal on Computing*, vol. 32, no. 5, pp. 1280–1316, 2003.

[15] G. Strang, "The discrete cosine transform," *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.

[16] E. Pavez and A. Ortega, "Generalized laplacian precision matrix estimation for graph signal processing." ICASSP, 2016.

[17] A. Said and W. A. Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," in *Information Theory. 1997. Proceedings., 1997 IEEE International Symposium on*. IEEE, 1997, p. 61.

[18] G. Bjontegaard, "Calcuation of average psnr differences between rd-curves," *Doc. VCEG-M33 ITU-T Q6/16*, 2001.

[19] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," *arXiv preprint arXiv:1611.05181*, 2016.