

# Graph-Based Construction and Assessment of Motion-Adaptive Transforms

Du Liu and Markus Flierl  
School of Electrical Engineering  
KTH Royal Institute of Technology  
Stockholm, Sweden  
{dul, mflierl}@kth.se

**Abstract**—In this paper, we propose two algorithms to construct motion-adaptive transforms that are based on vertex-weighted graphs. The graphs are constructed by motion vector information. The weights of the vertices are given by scale factors that are used to accommodate proper concentration of energy in transforms. The vertex-weighted graph defines a one dimensional linear subspace. Thus, our transform basis is subspace constrained. We propose two algorithms. The first is based on the Gram-Schmidt orthonormalization of the discrete cosine transform (DCT) basis. The second combines the rotation of the DCT basis and the Gram-Schmidt orthonormalization. We assess both algorithms in terms of energy compaction. Moreover, we compare to prior work on graph-based rotation of the DCT basis and on so-called motion-compensated orthogonal transforms (MCOT). In our experiments, both algorithms outperform MCOT in terms of energy compaction. However, their performance is similar to that of graph-based rotation of the DCT basis.

## I. INTRODUCTION

For efficient video coding, our goal is to improve conventional frame-by-frame coding by developing motion-adaptive representations that are suitable for coding at least two frames jointly. The motion between frames forces us to deviate from conventional wavelets. Wavelet type constructions for functions defined on more general manifolds are feasible [1]. However, due to occlusions in the pictures, local homeomorphisms may not always exist. On the other hand, we are able to define wavelets on graphs [2]. For example, lifting transforms on graphs [3] may be constructed based on spatial and temporal connections due to block motion. In [2], the wavelets are based on the spectral decomposition of the discrete Laplacian matrix of the graph [4]. One drawback is that the wavelet is overcomplete since there is no downsampling as in the classic discrete wavelet transform.

The so-called motion-compensated lifting wavelet permits a reversible filter structure, but struggles with unconnected, connected, and multi-connected pixels when performing the update step. Hence, [5] and [6] propose modified update operators to address this shortcoming. Instead of using the lifting wavelet, the class of motion-compensated orthogonal transforms (MCOT) is designed for successive pictures in a video sequence and maintains strict orthogonality for any motion field [7], [8]. Not requiring the concept of lifting, these transforms compact the signal energy efficiently by using the concept of scale factors. Further, sub-pel accurate motion and

variable block sizes can be used with MCOT for efficient video coding [9].

Beyond wavelets, [10] considers an edge-adaptive graph-based transform, where it is constructed by the eigenvectors of the Laplacian matrix of the graph. This method requires additional information on the edge map.

In [11], we designed a class of motion-adaptive transforms based on vertex-weighted graphs. The vertex-weighted graph is constructed from the motion information only. Since the transform basis is derived from the eigen-decomposition of a subspace representation of the covariance matrix, the transform achieves optimal energy compaction given the subspace constraint. The advantage is that the constraint is defined by the vertex-weighted graph only. One disadvantage of the method is that many basis vectors are signal dependent. Thus, they need to be signaled to the decoder.

We continued to design a transform that takes the same vertex-weighted graph into consideration, but the basis vectors are not signal dependent [12]. The transform can be obtained simply by rotating the discrete cosine transform (DCT) basis. No extra information needs to be transmitted to recover the basis. However, since there are many ways to determine the rotation of the DCT basis, the high dimensions of the transform matrix are not unique.

In this paper, we focus on the high dimensions of the transform matrix. Given the same vertex-weighted graph, we propose two additional methods to construct the motion-adaptive transform with the utility of the DCT basis. We also discuss the relation between different constructed transform matrices.

The paper is organized as follows: Section II defines the vertex-weighted graph. Section III proposes two algorithms for constructing the transform matrix and discusses the relation between them. Section IV presents the experimental results.

## II. SUBSPACE CONSTRAINT

We consider in [11] and [12] a subspace-constrained transform that is based on vertex-weighted graphs. The graph is constructed by motion vector information. The vertex weights are given by the scale factors, which are defined according to [7]. As the vertex-weighted graph determines the transform in a one-dimensional linear subspace, the transform is subspace

constrained. In the following, we explain the scale factors and the subspace constraint.

Let  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  be a vector representing  $n$  connected pixels. Energy compaction changes the magnitude of the pixel intensities, while the model of ideal motion implies constant intensity for connected pixels. In order to combine energy compaction and ideal motion, we use the concept of scale factors from [7] and represent the original pixel values by  $x'_1, x'_2, \dots, x'_n$ . Then, let  $c_1, c_2, \dots, c_n$  be the scale factors of the pixels  $x_1, x_2, \dots, x_n$ , respectively, i.e.,  $x_k = c_k x'_k$ ,  $k = 1, 2, \dots, n$ .

To illustrate concept of the scale factors, let us consider the following example. Suppose a reference pixel  $x'_1$  with intensity 100 is connected to two pixels  $x'_2$  and  $x'_3$  with the same intensity. In the first step of the transform, the energy of  $x'_2$  is shifted to  $x'_1$ . Then,  $x'_2$  becomes an energy-removed highband coefficient. The value of the reference pixel becomes  $x_1 = \sqrt{2} \cdot 100 = c_1 x'_1$ , where  $c_1 = \sqrt{2}$  is the scale factor and  $x_1$  captures the energy of  $x'_1$  and  $x'_2$ . In the second transform step, the energy of  $x'_3$  is shifted to  $x'_1$  as well. As a result,  $x_1 = \sqrt{3} \cdot 100 = c_1 x'_1$ , where  $c_1$  is updated to  $\sqrt{3}$ .  $x_1$  now captures the total energy of  $x'_1, x'_2$ , and  $x'_3$ , where the compaction of the energy is represented by  $c_1$ .

Similar to [7], we assume that the  $n$  pixels are connected by ideal motion such that  $x'_1 = x'_2 = \dots = x'_n$ . The weights of the vertices are then the scale factors and the vector of scale factors is  $\mathbf{c} = [c_1, c_2, \dots, c_n]^T$ . Then, an  $n \times n$  orthonormal transform matrix  $T = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n]$  compacts all the energy into the first coefficient,

$$\mathbf{y} = T^T \mathbf{x} = T^T \begin{bmatrix} c_1 x'_1 \\ c_2 x'_2 \\ \vdots \\ c_n x'_n \end{bmatrix} = T^T \begin{bmatrix} c_1 x'_1 \\ c_2 x'_1 \\ \vdots \\ c_n x'_1 \end{bmatrix} = \begin{bmatrix} \sqrt{\sum_{k=1}^n c_k^2} \cdot x'_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (1)$$

where  $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$  is the output of the transform. Equation (1) can be rewritten as

$$\mathbf{t}_1^T \mathbf{c} = \sqrt{\mathbf{c}^T \mathbf{c}}, \quad (2)$$

$$\mathbf{t}_k^T \mathbf{c} = 0, \quad k = 2, 3, \dots, n. \quad (3)$$

That is,  $\mathbf{t}_1$  captures the whole energy of the signal and  $\mathbf{t}_2, \dots, \mathbf{t}_n$  capture no energy. Considering an  $n$ -dimensional orthogonal space defined by  $n$  orthonormal vectors  $\mathbf{t}_1$  to  $\mathbf{t}_n$ , the above two equations (2) and (3) are projections of  $\mathbf{c}$  onto this  $n$ -dimensional orthogonal space. The first vector  $\mathbf{t}_1$  should be collinear with  $\mathbf{c}$  such that the projection of  $\mathbf{c}$  onto  $\mathbf{t}_1$  contains the total energy. Since the space is orthogonal, no energy is left to be projected on the remaining  $(n-1)$ -dimensional subspace. Thus,  $\mathbf{t}_1$  is determined by  $\mathbf{c}$ , which gives our subspace constraint

$$\mathbf{t}_1 = \frac{\mathbf{c}}{\|\mathbf{c}\|_2} = \frac{\mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{c}}}. \quad (4)$$

If  $\mathbf{x}$  is not due to ideal motion, i.e.,  $x_1, x_2, \dots, x_n$  are affected by noise, we will not obtain perfect energy compaction into one coefficient. However, we keep the subspace constraint  $\mathbf{c}$  as it reflects the vertex-weighted graph.

### III. CONSTRUCTION OF TRANSFORM MATRICES

#### A. Graph-Based Rotation (GBR)

In [12], we considered the well-known discrete Fourier transform (DFT) and circulant matrices that can be diagonalized by DFT. Let  $L_0$  be a covariance matrix with symmetric circulant structure and  $\Lambda_0$  the eigenvalue matrix of  $L_0$ . Based on the DFT matrix  $F$  and  $L_0$ , we want to find a covariance matrix  $L_1$  that can be diagonalized by  $T$ . Let  $\Lambda_1$  be the eigenvalue matrix of  $L_1$ . There exists a rotation matrix  $R$  that rotates the first basis vector  $\mathbf{f}_1 = \frac{1}{\sqrt{N}}[1, \dots, 1]^T$  to  $\mathbf{t}_1$  to meet the subspace constraint, i.e.,  $\mathbf{t}_1 = R\mathbf{f}_1$ . We keep the same rotation for higher dimensional basis vectors, thus,

$$T = RF. \quad (5)$$

Since the energy is preserved after the rotation, the sum of the eigenvalues for  $L_0$  and  $L_1$  are equal. For simplicity, we let  $\Lambda_1 = \Lambda_0$ . Then, the relation between  $L_0$  and  $L_1$  is given by

$$L_1 = RL_0R^T. \quad (6)$$

Fig. 1 depicts a commutative diagram for  $L_0, L_1, F$ , and  $T$ . The problem of rotating  $F$  to  $T$  is related to the mapping from  $L_0$  to  $L_1$ . The link is given by the eigenvalue decomposition / Karhunen Loeve Transform (KLT).

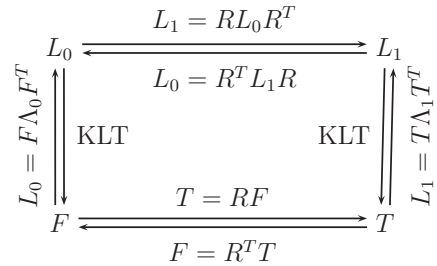


Fig. 1. Commutative diagram for covariance matrices and transforms.

We discussed in [12] that if there exists a  $T$  that satisfies the subspace constraint  $\mathbf{t}_1$ , we can find a corresponding  $L_1$  that this  $L_1$  in return gives  $T$  as an eigenvector matrix with the first basis vector  $\mathbf{t}_1$ . However, since  $T$  is not unique in high dimensions, it cannot be determined directly from the commutative diagram.

Based on the discussion, we proposed a practical algorithm, namely graph-based rotation (GBR) that rotates  $F$  to  $T$ . We rotated  $F$  only in the plane spanned by  $\mathbf{f}_1$  and  $\mathbf{t}_1$ , i.e., the projection part of  $F$  onto this plane will be rotated only. With that,  $\mathbf{f}_1$  will be rotated to  $\mathbf{t}_1$  to meet the subspace constraint. The orthogonal part of  $F$  is left untouched, which becomes the orthogonal part of  $T$ . Then,  $T$  is obtained by adding the orthogonal part and the rotated projection part together. If the weights in a graph are all equal, then  $\mathbf{t}_1 = \mathbf{f}_1$ , there is no need to rotate  $\mathbf{f}_1$  to  $\mathbf{t}_1$ . In this case  $T$  is the same as  $F$ .

#### B. Construction Based on Gram-Schmidt Orthonormalization

Since  $T$  is not unique in high dimensions, we propose here two additional algorithms to construct  $T$  based on  $F$ .

One is the Gram-Schmidt orthonormalization based on the DFT basis, the other is a combination of Gram-Schmidt orthonormalization and graph-based rotation.

1) *Gram-Schmidt Orthonormalization (GSO)*: Let the projection of vector  $\mathbf{f}_1$  onto vector  $\mathbf{t}_1$  denote as  $proj(\mathbf{f}_1, \mathbf{t}_1) \cdot \mathbf{t}_1$ . Given  $\mathbf{f}_1$ , we find a vector that is orthogonal to  $\mathbf{t}_1$ , i.e., subtract  $\mathbf{f}_1$  the projected part. Thus,  $\mathbf{e}_2 = \mathbf{f}_1 - proj(\mathbf{f}_1, \mathbf{t}_1) \cdot \mathbf{t}_1$ . The second basis vector  $\mathbf{t}_2 = \frac{\mathbf{e}_2}{\|\mathbf{e}_2\|_2}$  is then the normalized  $\mathbf{e}_2$ . Similarly, for high dimensional vectors in  $T$ , we have

$$\mathbf{e}_k = \mathbf{f}_{k-1} - \sum_{i=1}^{k-1} proj(\mathbf{f}_{k-1}, \mathbf{t}_i) \cdot \mathbf{t}_i, \quad (7)$$

$$\mathbf{t}_k = \frac{\mathbf{e}_k}{\|\mathbf{e}_k\|_2}, \text{ for } k = 2, 3, \dots, n, \quad (8)$$

where  $\mathbf{t}_k$  and  $\mathbf{f}_k$  are the  $k$ -th basis vectors in  $T$  and  $F$ , respectively. Equation (7) implies that  $\mathbf{e}_k$  is obtained by subtracting all the projected parts of  $\mathbf{f}_{k-1}$  onto  $\mathbf{t}_1, \dots, \mathbf{t}_{k-1}$ . It is guaranteed to be orthogonal to all the previous basis vectors  $\mathbf{t}_1$  to  $\mathbf{t}_{k-1}$ . The basis vector  $\mathbf{t}_k$  is then normalized in (8). Thus,  $T$  is an orthonormal matrix. We see that  $T$  is obtained by decomposing the basis vectors in  $F$  step by step.

2) *Combination of GBR and GSO (GBR+GSO)*: First, we use the GBR to find a complete  $n$ -dimensional basis  $B$ . At the same time, we use the GSO to find another complete basis  $G$ . Clearly, both  $B$  and  $G$  have the same first basis vector  $\mathbf{t}_1$ . The difference is in the high dimensional basis vectors. Second, we use the GBR again to rotate  $B$  to meet the second basis vector in  $G$ . Since the second basis vectors of  $B$  and  $G$  are both orthogonal to  $\mathbf{t}_1$ , the rotation does not affect the common vector  $\mathbf{t}_1$ .

In other words, we rotate the DFT matrix  $F$  twice. In the first rotation,  $F$  is rotated to  $B$  to meet  $\mathbf{t}_1$ . In the second rotation,  $B$  is rotated to meet the second basis vector in  $G$ . As a result, the first two basis vectors of this algorithm are the same as the GSO, while the rest of the basis vectors are based on the GBR.

The two algorithms give different high dimensional basis vectors in  $T$ . Similarly to the GBR, if  $\mathbf{t}_1 = \mathbf{f}_1$ , there is no need to decompose or rotate  $F$ . In this case for both algorithms,  $T$  is the same as  $F$ .

### C. Discussion

From the rotation  $T = RF$ , we have

$$L_0 = F\Lambda_0F^T = R^T T\Lambda_0T^T R. \quad (9)$$

Then  $L_1$  can be expressed as

$$L_1 = RL_0R^T = T\Lambda_0T^T. \quad (10)$$

The above (10) states that  $L_1$  is diagonalized by  $T$  with an eigenvalue matrix  $\Lambda_0$ . As  $\Lambda_1$  is defined as the eigenvalue matrix of  $L_1$ , it gives

$$\Lambda_1 = \Lambda_0. \quad (11)$$

Therefore,  $\Lambda_1$  preserves the structure of  $\Lambda_0$ . For example, assuming the covariance matrix  $L_0$  has a symmetric circulant

structure, its eigenvalues in  $\Lambda_0 = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n]$  satisfy the following structure [13]

$$\lambda_j = \lambda_{n+j-2}, \text{ for } j = 2, 3, \dots, n. \quad (12)$$

Thus, the eigenvalues in  $\Lambda_1$  also satisfy the property in (12). The structure of eigenvalues of the underlying covariance matrix is maintained before and after the rotation.

## IV. EXPERIMENTAL RESULTS

Since the DFT matrix is complex, we use the discrete cosine transform (DCT) matrix instead in our experiments. We evaluate the energy compaction for the QCIF sequences *Foreman*, *Bus*, *City* and *Mobile*. We compare the motion-compensated orthogonal transform (MCOT) [7], the graph-based rotation (GBR) [12], the Gram-Schmidt Orthonormalization (GSO), the combination of GBR and GSO (GBR+GSO), and the subspace-constrained transform (SCT) [11] for given graphs.

The MCOT is based on hierarchical decomposition. The SCT is derived from the eigen-decomposition of a subspace representation of the covariance matrix. Since the SCT achieves optimal energy compaction given the subspace constraint, it is given as a reference for the optimal highband energy. We concluded in [11] and [12] that  $\mathbf{t}_1$  approximates the energy compaction of the KLT well, since it compacts lowband energy close to that of the KLT. It confirms the relevance of the subspace constraint. Thus, we do not include the energy compaction of the KLT here.

The basis vectors for MCOT, GBR, GSO, and GBR+GSO are only graph dependent. However, the basis vectors  $\mathbf{t}_k$ ,  $k = 2, 3, \dots, n$ , for the SCT are signal dependent and require covariance estimation. For a practical coding scheme, they need to be signaled to the decoder. At this point, we look at the energy compaction only. Practical coding schemes will be considered in future work.

The covariance matrix for the SCT is estimated from samples that can provide the same vertex-weighted graphs. The vertex-weighted graphs are considered to be the same if they are formed by the same motion vectors and have the same vertex weights. In the experiments, we use graphs that are supported by at least thirty samples and that contain less than nine connections for a reliable estimation. For the transforms that are not signal dependent, we use only these reliable samples for all the transforms to ensure a fair comparison.

Since the GSO and the GBR+GSO share the same  $\mathbf{t}_2$ , the second coefficients of the transform output are equal. To show that the GSO and the GBR+GSO give the same highband energy in the first highband, we consider the graphs that have only one coefficient in the second frame, since for each instance of the graph, the transform determines the co-located subband coefficients. The output coefficients  $y_1, \dots, y_n$  will be at the same positions as  $x_1, \dots, x_n$ , respectively.

The transforms are performed on a hierarchical decomposition of a group of pictures (GOP). In the experiments, we choose a GOP size of sixteen. Each GOP is divided into four graph sets, where each set contains four successive frames. Each set outputs one temporal lowband and three

temporal highbands. A GOP of sixteen frames requires two decomposition levels. Since our graph-based transform does not require a fixed graph structure, it is possible to choose a different GOP size, rearrange the number of graph sets, and alter the number of decomposition levels. The graphs are defined by  $16 \times 16$  block motion with a search range of  $\pm 32$ .

Tables I to IV show the relative energy in the temporal subband frames on the second decomposition level for *Foreman*, *Bus*, *City*, and *Mobile*, respectively. All the transforms share the same basis vector  $t_1$ , thus they give the same lowband energy. For higher dimensional basis vectors  $t_k$ ,  $k > 1$ , we see that the second largest energy component of GBR is greater than the others. Therefore, among the proposed constructed transforms, GBR achieves similar or slightly better coding performance. Comparing GSO and GBR+GSO, they provide the same first highband energy, as they share the same basis vectors  $t_2$ . However, the third largest energy component of GSO is larger than GBR+GSO, which implies that GSO is better than GBR+GSO. Involving the GBR in the algorithm does not give advantage in energy compaction.

TABLE I

RELATIVE ENERGY IN THE TEMPORAL SUBBAND FRAMES ON THE SECOND DECOMPOSITION LEVEL FOR *Foreman*.

	Low	High 1	High 2	High 3
MCOT	98.771%	0.253%	0.573%	0.403%
GBR	98.771%	0.671%	0.282%	0.276%
GSO	98.771%	0.662%	0.289%	0.278%
GBR+GSO	98.771%	0.662%	0.285%	0.282%
SCT	98.771%	0.932%	0.224%	0.073%

TABLE II

RELATIVE ENERGY IN THE TEMPORAL SUBBAND FRAMES ON THE SECOND DECOMPOSITION LEVEL FOR *Bus*.

	Low	High 1	High 2	High 3
MCOT	92.742%	2.392%	3.543%	1.323%
GBR	92.742%	4.763%	1.315%	1.180%
GSO	92.742%	4.316%	2.007%	0.935%
GBR+GSO	92.742%	4.316%	1.730%	1.212%
SCT	92.742%	5.976%	1.059%	0.223%

TABLE III

RELATIVE ENERGY IN THE TEMPORAL SUBBAND FRAMES ON THE SECOND DECOMPOSITION LEVEL FOR *City*.

	Low	High 1	High 2	High 3
MCOT	90.648%	2.742%	4.265%	2.345%
GBR	90.648%	5.018%	2.130%	2.204%
GSO	90.648%	4.803%	2.345%	2.204%
GBR+GSO	90.648%	4.803%	2.286%	2.263%
SCT	90.648%	7.087%	1.762%	0.503%

## V. CONCLUSION

This paper considers motion-adaptive transforms based on vertex-weighted graphs and their practical construction using the DCT basis. The vertex-weighted graph determines the

TABLE IV

RELATIVE ENERGY IN THE TEMPORAL SUBBAND FRAMES ON THE SECOND DECOMPOSITION LEVEL FOR *Mobile*.

	Low	High 1	High 2	High 3
MCOT	92.850%	1.280%	4.498%	1.372%
GBR	92.850%	5.258%	1.363%	0.529%
GSO	92.850%	5.253%	1.366%	0.531%
GBR+GSO	92.850%	5.253%	1.362%	0.535%
SCT	92.850%	5.551%	1.268%	0.331%

first basis vector of the linear transform and leads to a subspace constraint. We propose two algorithms to construct the transform practically. One is based on the Gram-Schmidt orthonormalization of the DCT basis, the other is a combination of the rotation of the DCT basis and the Gram-Schmidt orthonormalization. A discussion of the relations in the commutative diagram shows that any rotation from  $F$  to  $T$  maintains the structure of eigenvalues of the underlying covariance matrices. The two methods introduced in this paper, together with the graph-based rotation [12], provide three examples to construct practical transforms that approximate the SCT.

## ACKNOWLEDGMENT

This work has been supported in part by the Swedish Research Council under the grant 2011-5841.

## REFERENCES

- [1] P. Schröder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," in *Proc. of the 22nd Conference on Computer Graphics and Interactive Techniques*, 1995, pp. 161–172.
- [2] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [3] E. Martínez-Enríquez, F. Díaz-de-María, and A. Ortega, "Video encoder based on lifting transforms on graphs," in *Proc. of the IEEE International Conference on Image Processing*, Sep. 2011, pp. 3509–3512.
- [4] F. R. K. Chung, *Spectral Graph Theory*, ser. CMBS Regional Conference Series in Mathematics, No. 92. American Mathematical Society, 1997.
- [5] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar, "Improved update operators for lifting-based motion-compensated temporal filtering," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 146–149, 2005.
- [6] B. Girod and S. Han, "Optimum update for motion-compensated lifting," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 150–153, 2005.
- [7] M. Flierl and B. Girod, "A motion-compensated orthogonal transform with energy-concentration constraint," in *Proc. of the IEEE International Workshop on Multimedia Signal Processing*, Oct. 2006, pp. 391–394.
- [8] —, "Half-pel accurate motion-compensated orthogonal video transforms," in *Proc. of the IEEE Data Compression Conference*, Mar. 2007, pp. 13–22.
- [9] D. Liu and M. Flierl, "Video coding with adaptive motion-compensated orthogonal transforms," in *Proc. of the Picture Coding Symposium*, May 2012, pp. 293–296.
- [10] W.-S. Kim, S. K. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2012, pp. 813–816.
- [11] D. Liu and M. Flierl, "Motion-adaptive transforms based on vertex-weighted graphs," in *Proc. of the IEEE Data Compression Conference*, Mar. 2013, pp. 181–190.
- [12] —, "Graph-based rotation of the DCT basis for motion-adaptive transforms," in *Proc. of the IEEE International Conference on Image Processing*, Sep. 2013.
- [13] O. Rojo and H. Rojo, "Some results on symmetric circulant matrices and on symmetric centrosymmetric matrices," *Linear Algebra and Its Applications*, vol. 392, pp. 211–233, 2004.