

Steerable Discrete Cosine Transform

Original

Steerable Discrete Cosine Transform / Fracastoro, Giulia; Magli, Enrico. - ELETTRONICO. - (2015), pp. 1-6. (Intervento presentato al convegno 2015 IEEE International Workshop on Multimedia Signal Processing nel 2015) [10.1109/MMSP.2015.7340827].

Availability:

This version is available at: 11583/2638898 since: 2016-04-01T17:18:48Z

Publisher:

IEEE

Published

DOI:10.1109/MMSP.2015.7340827

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Steerable Discrete Cosine Transform

Giulia Fracastoro ^{#1}, Enrico Magli ^{#2}

[#] *Department of Electronics and Telecommunications, Politecnico di Torino, Italy*

¹giulia.fracastoro@polito.it

²enrico.magli@polito.it

Abstract—Block-based separable transforms tend to be inefficient when blocks contain arbitrarily shaped discontinuities. For this reason, transforms incorporating directional information are an appealing alternative. In this paper, we propose a new approach to this problem, designing a new transform that can be steered in any chosen direction and that is defined in a rigorous mathematical way. This new steerable DCT allows to rotate in a flexible way pairs of basis vectors, enabling precise matching of directionality in each image block, and thereby achieving improved coding efficiency. We tested the proposed transform on several images and the results show that it provides a significant performance gain compared to the DCT. Moreover, the mathematical framework on which the steerable DCT is based allows to generalize the transform to more complex steering patterns than a single pure rotation.

I. INTRODUCTION

The 2D Discrete Cosine Transform (DCT) is by far the most widespread transform used for block-based image and video compression [1]. The 2D DCT is implemented by two separable 1D transforms along the vertical and horizontal directions. Thus, the conventional DCT can efficiently represent images in which horizontal or vertical edges are dominating. Instead, one of the main drawbacks of the DCT is that it becomes inefficient when a block contains arbitrarily shaped discontinuities.

To overcome this problem, various solutions have been proposed in the past and different approaches have been explored. Several researchers have proposed to modify the implementation of the separable 2D DCT in order to incorporate directional information into the transform [2]–[4]. The Directional DCT (DDCT) presented in [2] is the first attempt in this sense. The authors developed a new separable transform in which the first 1D DCT may choose to follow a direction other than the vertical or horizontal one, then the coefficients produced by all directional transforms in the first step are rearranged so that the second transform can be applied to the coefficients that are best aligned with each other. However, this method faces several issues [5]: 1D DCTs of various lengths are needed, some of them are very short and their lengths are not always a power of 2. Moreover, the second DCT may not always be applied to coefficients of similar AC frequencies.

In the specific case of intra-frame video coding, another approach to design a directional transform has been investigated. In [6], the authors proposed to derive mode-dependent directional transforms (MDDT) from Karhunen-Loeve transform (KLT) using prediction residuals from training video data. Using this approach, various follow-up works have presented several variations and enhancements for the MDDT exploiting

the use of symmetry to reduce the number of transform matrices needed [7]–[9]. However, the main problem of this methods is that training sets must be processed to obtain transforms that are optimal for a given mode.

Recently, a new approach in image and video coding is emerging. Any image can be viewed as a graph, where each pixel is a node of the graph and the edges describe the connectivity relations among the pixels, e.g. in terms of similarity [10]. This graph representation allows one to design an edge-aware transform in an elegant and effective way. Block-based method using graph transform have been proposed in [11], [12], but they all reported unsatisfactory results on natural images that are not piece-wise smooth. For the specific case of texture images, a new class of transforms called graph template transforms was proposed to approximate the KLT by exploiting a priori information known about signals represented by a graph-template [13]. However, one of the main drawbacks of graph-based compression techniques lies in the cost required to represent and encode the graph, which may outweigh the coding gain provided by the edge adaptive transform. For this reason, recently some graph-based compression methods that require a small overhead have been presented [14], [15], however, even if their performances are competitive compared to the DCT, the construction of the transform matrix has a high computational cost.

In this paper, we present a new framework for directional transforms. Starting from the graph transform of a grid graph, we design a new transform, the Steerable DCT (SDCT), that can be obtained simply by rotating the 2D DCT basis. In this way, the proposed transform can be oriented in any possible direction and can achieve better compression performance than the DCT. The SDCT is defined in a rigorous mathematical way and can be easily computed starting from the 2D DCT. In addition, we show that the proposed framework is very general and it is open to many further extensions that could lead to significant improvements.

The paper is organized as follows. Section 2 presents some theoretical results that will be needed in the following sections of the paper. We then introduce the construction method for the Steerable DCT basis in Section 3. In Section 4, the results of our experimental tests are presented. A final discussion on the proposed method is conducted in Section 5.

II. GRAPH FOURIER TRANSFORM AND DCT

We start our analysis by briefly reviewing the graph Fourier transform and highlighting the link between this transform and

the DCT.

A. Laplacian matrix and graph Fourier transform

A graph can be denoted as $G = (V, E)$, where V is the set of vertices (or nodes) with $|V| = N$ and $E \subset V \times V$ is the set of edges. It is possible to represent a graph by its adjacency matrix $A \in \mathbb{R}^{N \times N}$, where $A_{ij} = 1$ if there is an edge between node i and j , otherwise $A_{ij} = 0$. The graph Laplacian is defined as $L = D - A$, where D is a diagonal matrix whose i th diagonal element d_i is equal to the number of edges incident to node i .

We can define a signal f on the vertices of the graph and it can be represented as a vector $\mathbf{f} \in \mathbb{R}^N$, where the i th component of \mathbf{f} represents the signal value at the i th vertex in V .

In the graph domain, it is possible to define an equivalent of the Fourier transform, i.e. the graph Fourier transform [10]. The graph Fourier transform $\hat{\mathbf{f}}$ of any signal $\mathbf{f} \in \mathbb{R}^N$ is defined as

$$\hat{\mathbf{f}} = U\mathbf{f},$$

where U is the matrix whose rows are the eigenvectors of the graph Laplacian L . The inverse graph Fourier transform is then given by

$$\mathbf{f} = U^T \hat{\mathbf{f}}.$$

B. Product graph and its Laplacian

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the product of G_1 and G_2 is the graph $G = G_1 \times G_2$ whose vertex set is the Cartesian product $V_1 \times V_2$. Suppose $v_1, v_2 \in V_1$ and $u_1, u_2 \in V_2$. Then (v_1, u_1) and (v_2, u_2) are adjacent in G if and only if one of the following conditions are satisfied [16]:

- $v_1 = v_2$ and $\{u_1, u_2\} \in E_2$;
- $\{v_1, v_2\} \in E_1$ and $u_1 = u_2$.

Theorem [16], [17]. *Let G_1 and G_2 be graphs on n_1 and n_2 vertices, respectively. Then the eigenvalues of $L(G_1 \times G_2)$ are all possible sums of $\lambda_i(G_1) + \lambda_j(G_2)$, with $0 \leq i \leq n_1 - 1$ and $0 \leq j \leq n_2 - 1$. Moreover, if $\mathbf{v}^{(i)}$ is an eigenvector of G_1 corresponding to $\lambda_i(G_1)$, $\mathbf{v}^{(j)}$ an eigenvector of G_2 corresponding to $\lambda_j(G_2)$ and \otimes is the Kronecker product, then $\mathbf{v}^{(i)} \otimes \mathbf{v}^{(j)}$ is an eigenvector of G corresponding to $\lambda_i(G_1) + \lambda_j(G_2)$.*

C. Grid graph and 2D DCT

A path graph P_n is a particularly simple graph consisting of n vertices, whose structure is shown in Fig 1(a). It is known that the eigenvectors of the Laplacian matrix L of the path graph P_n are identical to the basis vectors of the discrete cosine transform (type II) [18]

$$v_j^{(k)} = \cos\left(\frac{\pi k}{n} \left(j + \frac{1}{2}\right)\right), \quad j = 0, 1, \dots, n-1 \quad (1)$$

$$\lambda_k = 4 \sin^2\left(\frac{\pi k}{2n}\right), \quad (2)$$

for $k = 0, 1, \dots, n-1$, where $\mathbf{v}^{(k)} = (v_0^{(k)}, v_1^{(k)}, \dots, v_{n-1}^{(k)})^T$ is the eigenvector of L corresponding to λ_k . Given that the

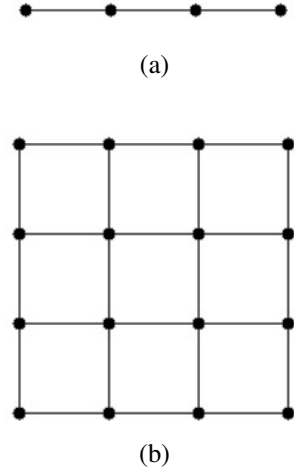


Fig. 1. Two graph models: (a) the path graph P_4 , (b) the square grid graph $P_4 \times P_4$.

multiplicity of the eigenvalues in (2) is always 1, the 1D DCT basis is the unique eigenbasis for the Laplacian of a path graph, therefore the graph transform for a signal represented by a path graph is equivalent to the 1D DCT transform.

It is easy to see that a grid graph as the one shown in Fig. 1(b) is the product of two path graphs. If the two path graphs have the same number of vertices n , their product graph is a square grid graph. It has been proved that the basis vectors of the 2D DCT form an eigenbasis of the Laplacian matrix of the square grid graph [19].

III. STEERABLE DCT

Starting from the theoretical results presented in the previous section, we build a new transform that can be oriented in any direction.

A. Preliminaries

Using the theorem presented in the previous section and equations (1) and (2), we can compute the eigenvalues and the corresponding eigenvectors of the Laplacian of the square grid graph

$$\lambda_{k,l} = \lambda_k + \lambda_l = 4 \sin^2\left(\frac{\pi k}{2n}\right) + 4 \sin^2\left(\frac{\pi l}{2n}\right), \quad (3)$$

$$\mathbf{v}^{(k,l)} = \mathbf{v}^{(k)} \otimes \mathbf{v}^{(l)}, \quad (4)$$

for $0 \leq k, l \leq n-1$, where $\mathbf{v}^{(k)}$ is the eigenvector of the path graph corresponding to λ_k and $\mathbf{v}^{(l)}$ is the eigenvector corresponding to λ_l . From (3), it is evident that some repeated eigenvalues are present, indeed $\lambda_{k,l} = \lambda_{l,k}$ if $k \neq l$. Moreover, through straightforward computations, it is possible to prove that the eigenvalue $\lambda = 4$ has multiplicity $n-1$ and corresponds to all eigenvalues $\lambda_{k,n-k}$ with $1 \leq k \leq n-1$. Therefore, in the spectrum of L there are only $n-1$ eigenvalues with algebraic multiplicity 1 (i.e. $\lambda_{k,k}$ with $k \neq n/2$), then all the others but $\lambda_{k,n-k}$ have multiplicity 2. It is important to highlight that even if $\lambda_{k,l} = \lambda_{l,k}$ when $k \neq l$, we still have

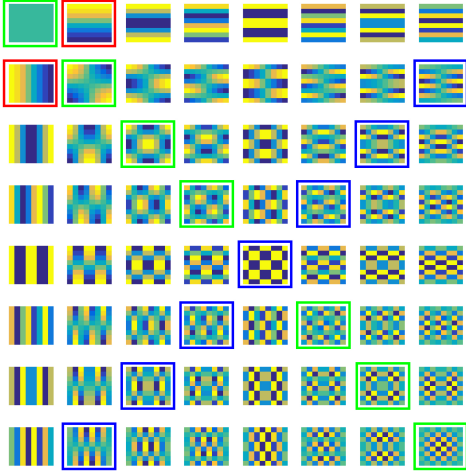


Fig. 2. 2D DCT basis vectors represented in matrix form (with $n = 8$): the corresponding two eigenvectors of an eigenvalue with multiplicity 2 are highlighted in red, the $n - 1$ eigenvectors corresponding to $\lambda = 4$ are highlighted in blue and the $n - 1$ eigenvectors corresponding to the eigenvalues with multiplicity 1 are highlighted in green.

that $\mathbf{v}^{(k,l)}$ and $\mathbf{v}^{(l,k)}$ are linearly independent, because the Kronecker product is not commutative. Therefore, the dimension of the eigenspaces corresponding to these eigenvalues is bigger than one. This means that the 2D DCT is not the unique eigenbasis for the Laplacian of a square grid graph.

In Figure 2 the 2D DCT basis with $n = 8$ is represented in matrix form; as an example, we have highlighted in red the corresponding two eigenvectors of an eigenvalue with multiplicity 2, we can see that they are clearly related to each other, since one is the 90° rotation of the other.

B. Transform definition

Since the 2D DCT is not the unique eigenbasis for the Laplacian of a square grid graph, we want to find other possible eigenbases of the Laplacian and choose as transform matrix the one that better fits the properties of the specific image block that we have to compress.

Given an eigenvalue $\lambda_{k,l}$ of L whose corresponding eigenspace has dimension two and the two vectors of the 2D DCT $\mathbf{v}^{(k,l)}$ and $\mathbf{v}^{(l,k)}$ that are the eigenvectors of L corresponding to $\lambda_{k,l}$, we can write any other possible basis of the eigenspace corresponding to $\lambda_{k,l}$ as the result of a rotation of the eigenvectors $\mathbf{v}^{(k,l)}$ and $\mathbf{v}^{(l,k)}$

$$\begin{bmatrix} \mathbf{v}^{(k,l)'} \\ \mathbf{v}^{(l,k)'} \end{bmatrix} = \begin{bmatrix} \cos \theta_{k,l} & \sin \theta_{k,l} \\ -\sin \theta_{k,l} & \cos \theta_{k,l} \end{bmatrix} \begin{bmatrix} \mathbf{v}^{(k,l)} \\ \mathbf{v}^{(l,k)} \end{bmatrix}, \quad (5)$$

where $0^\circ \leq \theta_{k,l} \leq 90^\circ$.

For every eigenvalue $\lambda_{k,l}$ with multiplicity 2, we can rotate the corresponding eigenvectors as shown in (5); the $n - 1$ eigenvectors corresponding to $\lambda = 4$ are rotated in pairs $\mathbf{v}^{(k,n-k)}$ and $\mathbf{v}^{(n-k,k)}$, if n is even the eigenvector $\mathbf{v}^{(\frac{n}{2}, \frac{n}{2})}$ is not rotated. In the 2D DCT matrix, the pairs of basis vectors $\mathbf{v}^{(k,l)}$ and $\mathbf{v}^{(l,k)}$ are replaced with the rotated ones $\mathbf{v}^{(k,l)'}$ and $\mathbf{v}^{(l,k)'}$ obtaining a new transform matrix that can be defined

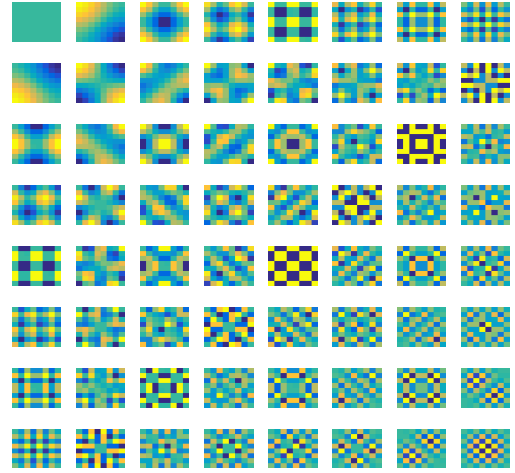


Fig. 3. Steerable DCT with $\theta = 45^\circ$.

only by the rotation angles used, that we have to transmit to the decoder.

In principle, we can rotate each eigenspace with a different angle, choosing the best one for every pair of DCT basis vectors. However, in this case the cost required to transmit all the rotation angles may be too high. In this paper, we limit our study to the case where we use the same rotation angle for every eigenspace, transmitting, therefore, only one angle per block. The case of a rotation with a different angle for each pair of basis vectors is left as future work.

Rotating all the eigenspaces by the same angle, we obtain a transform matrix that is still the graph transform of a square grid graph, but its orientation is different from that of the DCT. In Figure 3, we show the basis vectors obtained rotating by 45° degree each pair of eigenvectors. As can be seen, the diagonal elements $\mathbf{v}^{(k,k)}$ are the same as the DCT ones because the corresponding eigenvalues have multiplicity one, instead all the others are rotated by 45° .

The aim of using a transform matrix whose vector basis has a different orientation from the horizontal/vertical one is to obtain a more compact signal representation by unbalancing the transform coefficients. For each pair of rotated eigenvectors, the total energy of the corresponding transform coefficients remains unchanged, but it is possible to sparsify the signal representation in each eigenspace. In the optimal case, the rotation compacts all the energy of the pair in one of the two coefficients. If every couple of vectors is rotated by the same angle, in the majority of cases we will not achieve a complete unbalancing that zeros out one of the two coefficients, but we will still obtain an improvement in the compression performance, as we will show later in the paper.

C. Choice of the rotation angle

To choose the best rotation angle, we use an exhaustive method, finding, among a finite set of N possible angles between 0° and 90° , the one that optimizes a predetermined objective function J (e.g a measure of the sparsity of the transform coefficients, more details will be given in the

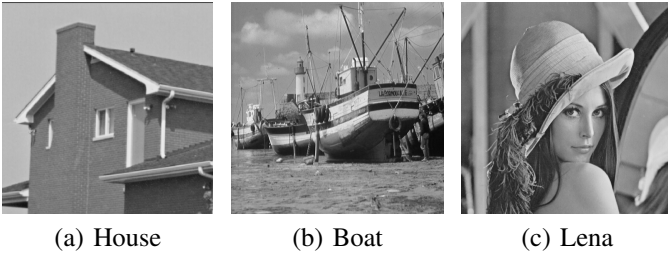


Fig. 4. Original images.

following section). Algorithm 1 describes the method used. The implementation of a more efficient method for choosing

Algorithm 1 Steerable DCT. θ : set of N angles between 0° and 90° , J : objective function, J_{opt} : optimal value of the objective function, V : 2D DCT matrix.

```

Set  $J_{opt}$  to 0;
Set the optimal angle to 0;
for  $i=1$  to  $N$  do
  Rotate each pair of vectors in  $V$  by  $\theta_i$ ;
  Build a new transform matrix  $V'$  with the rotated vectors;
  Compute  $J(\theta_i)$ : the value of the objective function  $J$  using  $V'$ ;
  if  $J(\theta_i) > J_{opt}$  then
    Set the optimal angle to  $\theta_i$ ;
     $J_{opt} = J(\theta_i)$ ;
  end if
end for

```

the optimal angle, e.g a hierarchical search, will be evaluated in future.

IV. EXPERIMENTAL RESULTS

For the purpose of experimentation, first we subdivide the image into blocks; then, in each block we apply the SDCT, using only one rotation angle for all the eigenspaces. To evaluate the performance of the proposed SDCT, we use the M term non-linear approximation, where we keep the M largest coefficients and set the others to zero:

$$I_{rec} = \sum_{i=1}^M c_i \mathbf{v}_i,$$

where I_{rec} is the reconstructed image, $\{c_i\}_{1 \leq i \leq M}$ are the M transform coefficients with largest magnitude and \mathbf{v}_i are the corresponding basis vectors. To find the best rotation angle, we choose, for each M , the one that maximizes the energy in the M largest coefficients ($J = \sum_{i=1}^M c_i^2$). Then, we compute the PSNR of each image and we compare the performance of the proposed SDCT with the standard DCT. We have tested this method on several grayscale images, three of them are shown in Figure 4.

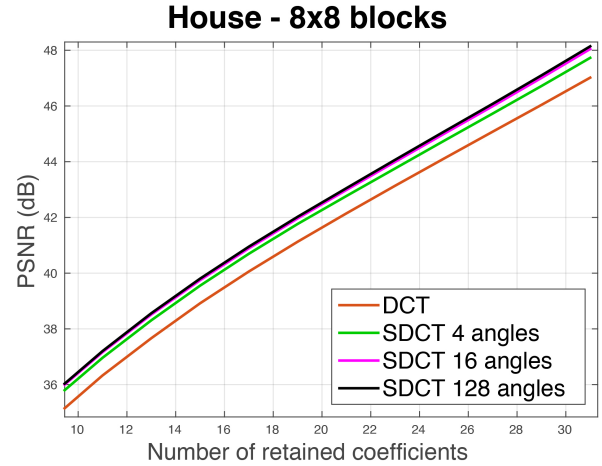


Fig. 5. M -term non-linear approximation using different angle quantizations.

The results presented are preliminary and they are meant to demonstrate the potentiality of the proposed transform, but further work is needed to develop image coding applications. In particular, the non-linear approximation used does not take into account the overhead bits needed to transmit the rotation angle, therefore the comparison with the DCT is not completely fair, even if the required overhead for the SDCT is very low.

A. Angle quantization

To obtain a finite set of angles, we perform an uniform quantization of the angles between 0° and 90° . Since the SDCT needs to transmit as side information the rotation angle that we used for the transform, it is important to have a small number of possible angles. We have evaluated the performance of the SDCT as a function of the number of available angles. The results obtained using 8×8 blocks are shown in Figure 5. From the results, we can see that increasing the number of angles improves the performance of the SDCT, but after a while this improvement becomes not significant, for example the performance using 16 angles and 128 angles is nearly the same. For the tests presented in the following sections, we decided to use quantization onto 16 possible angles.

B. Effect of block size

The proposed SDCT can be applied to square blocks of any size. However, in our experiments we have noticed that the gain of the SDCT depends on the block size used. In Figure 6 we show the performance curves obtained with different block sizes. On average, we have a coding gain of 1.5 dB with 4×4 blocks, 0.7 dB with 8×8 blocks and 0.25 dB with 16×16 blocks. We can clearly see that the effectiveness of the SDCT increase when we use small blocks, because in larger blocks it is more difficult to find just one predominant direction.

A detail of the significant improvement obtained by the SDCT can be visually appreciated in Figure 7 and 8. As can be seen, the SDCT provides much better visual quality, minimizing artifacts along the edges thanks to the alignment of the transform to the edge direction.

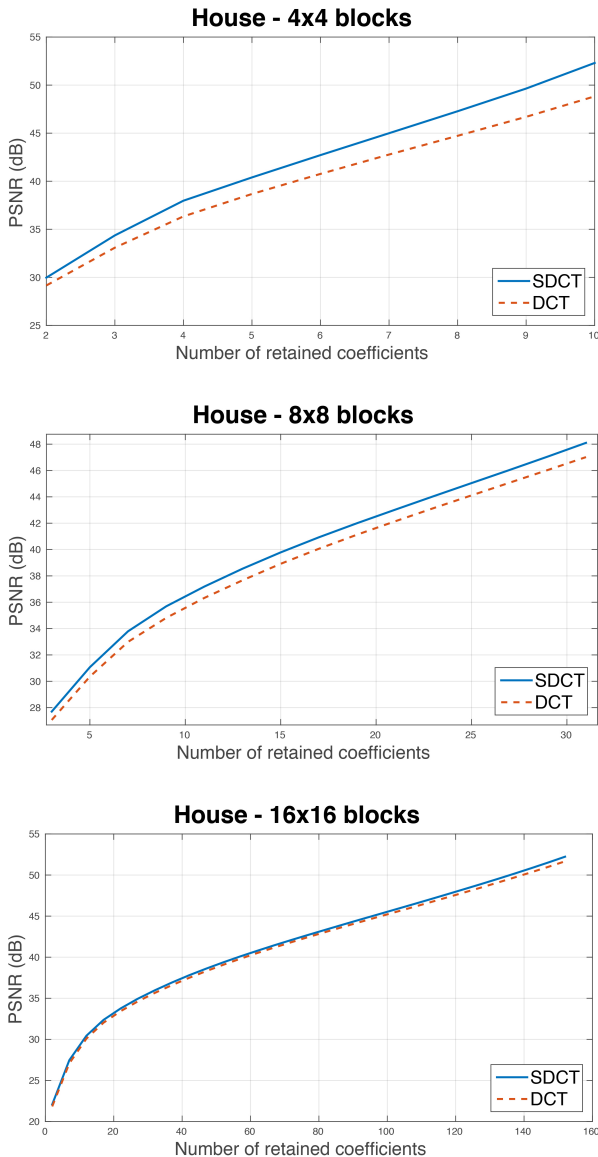


Fig. 6. M -term non-linear approximation using different block sizes.

C. Comparison with DDCT

In the case of 8×8 blocks, we have also made a comparison with the DDCT presented in [2]. In Figure 9, the results obtained with some test images are shown. We can see that the performances of the two methods are very similar, even if the proposed SDCT is slightly better with an average coding gain of approximately 0.1 dB. Even if the performances of the two transforms are similar, the SDCT presents a more general framework for image coding that can be further extended using more than one rotation angle. In order to show the potential of the proposed framework, we have ordered the pairs of vectors using the zigzag ordering and divided them in four subbands of equal size, then we have used a different rotation angle for each subband. The results obtained are plotted in Figure 9. We can see that the improvement over the SDCT and the DDCT

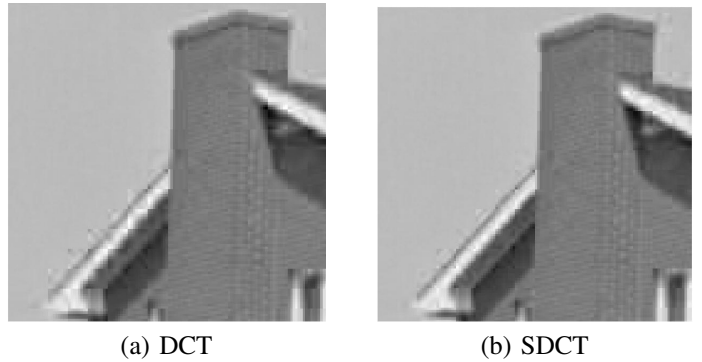


Fig. 7. Visual comparison between the conventional DCT and the proposed SDCT: a detail of the House image reconstructed from $M = 6$ coefficients using 8×8 blocks.

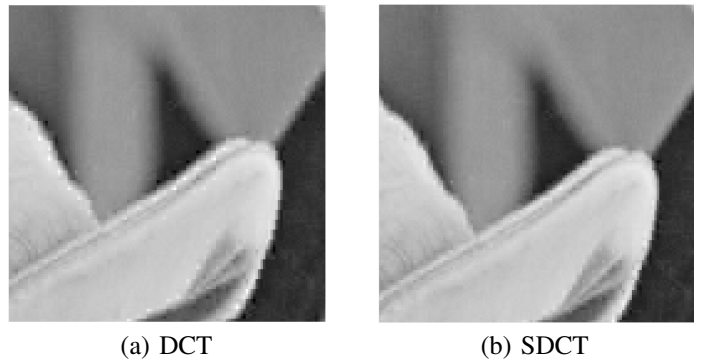


Fig. 8. Visual comparison between the conventional DCT and the proposed SDCT: a detail of the Lena image reconstructed from $M = 3$ coefficients using 4×4 blocks.

is significant, with an average quality gain of 0.45 dB over the SDCT and of 1.15 dB over the standard DCT.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduce a completely new framework for block-based directional DCT. The proposed transform has an elegant mathematical definition: exploiting the properties of the graph transform of a grid graph, we obtain a new transform basis that can be steered in any chosen direction. This new transform is strongly related to the 2D DCT and it can be easily computed starting from there. The proposed method provides a significant improvement over the DCT and it slightly outperforms the DDCT.

Furthermore, it is very important to underline that, differently from other directional transforms, the proposed mathematical framework is highly general. The SDCT can be applied to blocks of any size, it can be oriented in any direction and we can use more complex steering patterns than a single pure rotation. In addition, future work might also include the use of the proposed SDCT for intra prediction coding. In this case, we intend to implement the proposed SDCT into HEVC so that the rotation angle could be derived from the intra prediction

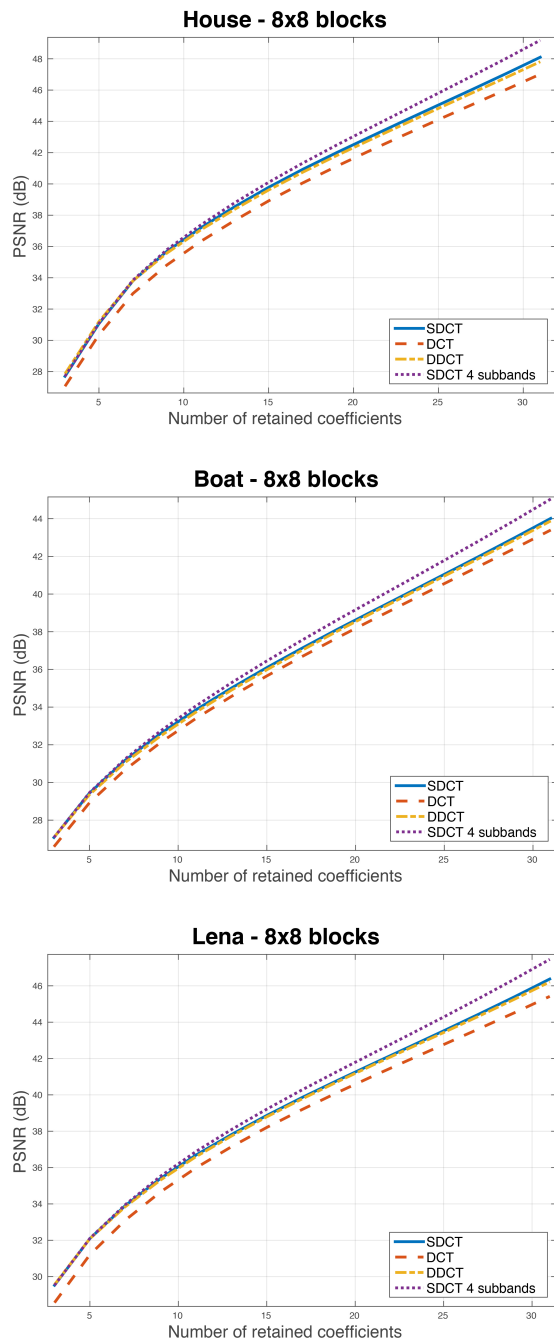


Fig. 9. Performance comparison between DCT, SDCT and DDCT.

mode chosen for the block, avoiding the use of overhead bits to represent the rotation angle.

ACKNOWLEDGEMENT

This work has been supported by Sisvel Technology.

REFERENCES

[1] K. Sayood, *Introduction to data compression*. Newnes, 2012.
 [2] B. Zeng and J. Fu, "Directional discrete cosine transforms - a new framework for image coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 305–313, 2008.

[3] C. Chang and B. Girod, "Direction-adaptive partitioned block transform for image coding," in *Image Processing (ICIP), 2008 15th IEEE International Conference on*, 2008, pp. 145–148.
 [4] R. Cohen, S. Klomp, A. Vetro, and H. Sun, "Direction-adaptive transforms for coding prediction residuals," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 185–188.
 [5] C. Yeo, Y. Tan, Z. Li, and S. Rahardja, "Mode-dependent transforms for coding directional intra prediction residuals," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 4, pp. 545–554, 2012.
 [6] Y. Ye and M. Karczewicz, "Improved H.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," in *Image Processing (ICIP), 2008 15th IEEE International Conference on*, 2008, pp. 2116–2119.
 [7] M. Budagavi and M. Zhou, "Orthogonal MDDT and mode dependent DCT," *ITU-T Q*, vol. 6, 2010.
 [8] H. Yang, J. Zhou, and H. Yu, "Simplified MDDT (SMDDT) for intra prediction residual," *Doc. JCTVC-B039, MPEG-H/JCT-VC*, 2010.
 [9] A. Tanizawa, J. Yamaguchi, T. Shiodera, T. Chujoh, and T. Yamakage, "Improvement of intra coding by bidirectional intra prediction and 1 dimensional directional unified transform," *Doc. JCTVC-B042, MPEG-H/JCT-VC*, 2010.
 [10] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83–98, 2013.
 [11] G. Shen, W. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *Picture Coding Symposium (PCS), 2010*. IEEE, 2010, pp. 2808–2811.
 [12] W. Kim, S. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 813–816.
 [13] E. Pavez, H. Egilmez, Y. Wang, and A. Ortega, "GTT: Graph Template Transforms with applications to image coding," in *Picture Coding Symposium (PCS), 2015*, 2015.
 [14] G. Fracastoro and E. Magli, "Predictive graph construction for image compression," in *Image Processing (ICIP), 2015 IEEE International Conference on*, 2015, to be published.
 [15] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven graph transform for image compression," in *Image Processing (ICIP), 2015 IEEE International Conference on*, 2015, to be published.
 [16] R. Merris, "Laplacian matrices of graphs: a survey," *Linear algebra and its applications*, vol. 197, pp. 143–176, 1994.
 [17] —, "Laplacian graph eigenvectors," *Linear algebra and its applications*, vol. 278, no. 1, pp. 221–236, 1998.
 [18] G. Strang, "The discrete cosine transform," *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.
 [19] C. Zhang and D. Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models," *Signal Processing Letters, IEEE*, vol. 20, no. 1, pp. 106–109, 2013.