IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 22, 2021, accepted July 13, 2021, date of publication July 16, 2021, date of current version July 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097720

Image Compression Based on a Partially Rotated Discrete Cosine Transform With a Principal Orientation

GIHWAN LEE, (Graduate Student Member, IEEE), AND YOONSIK CHOE, (Senior Member, IEEE)
Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Yoonsik Choe (yschoe@yonsei.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant through the Korea government (MSIT) under Grant 2021-0-00022, and in part by the AI Model Optimization and Lightweight Technology Development for Edge Computing Environment.

ABSTRACT Image transforms are necessary for image and video compression. Analytic transforms are powerful in compacting natural signals for wider exploitation. Various methods have been introduced to represent such data as a small number of bases, and several of these methods use machine learning, usually based on sparse coding, to outperform analytic transforms. They show sufficient data compaction abilities. However, these methods focus only on data compaction and reconstruction performance, without considering computational issues during implementation. We introduce a new framework for a more efficient transform based on a two-dimensional discrete cosine transform (DCT) and its characteristics. We aimed to improve the data compaction ability of transforms to levels better or similar to that of the DCT and other data-driven transforms, with fast and efficient implementation. We focused on the properties of the DCT, including horizontal and vertical directional information, and approximated its direction using the transform. Our framework was designed by rotating some of the DCT bases to fit this direction. As expected, our framework achieves a transform design with minimized computation for efficient implementation. It does not require an iterative algorithm or brute-force methods to find the best transform matrix or other parameters, thereby making it much faster than other methods. Our framework is 10 times faster than the steerable DCT (SDCT) and twice as fast as the eight-level SDCT with minimum performance reduction. Experimental validation with various images indicates that the proposed method sufficiently approaches the performance of the other transforms despite faster implementation.

INDEX TERMS Directional discrete cosine transform, discrete cosine transform, image transformation, image compression, sparse coding transform.

I. INTRODUCTION

Sparsity is an important issue in image processing, which involves expressing the input image using only a small amount of information, while minimizing the loss of original information as much as possible. Analytic transforms, such as the discrete cosine transform (DCT) and Karhunen-Loeve transform (KLT), represent natural signals more sparsely. By using an appropriate transform, many natural signals can be represented by a small number of large transform coefficients and a large number of extremely small coefficients. This has made image transform a long-standing staple in various image compression applications [1], [9]–[11]. The KLT is known as an optimal transform in Gaussian processes. In other words, if the input signal follows a Gaussian process, it exhibits optimal data compaction performance. However, it requires high memory resources to calculate the eigenvectors of the covariance matrix, and the algorithm is slow; therefore, instead of the KLT, the DCT is widely utilized in various fields because of its good performance, which is close to that of the KLT [8], and faster implementation algorithm.

Sparse coding is a technique used to design an appropriate dictionary, based on the input signal, and express this signal with the smallest number of dictionary atoms. This method is

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Sanchez.

similar to existing transforms in that it represents the input signal with minimal basis [2], [3]. In sparse coding-based transform, general characteristics can be learned from data beyond restricted conventional constraints. For example, the KLT is optimal for Gaussian processes, and the DCT exhibits good performance for horizontally and vertically directed patches, whereas sparse coding-based transform involves direct learning from the input data and the generation of an optimal transform matrix based on the input data. Therefore, many studies have proposed transforms based on sparse coding, which outperform existing analytic transforms [4], [5], [15]–[17]. The object of sparse coding has considerable commonalities with that of the transform; thus, transform techniques using sparse coding show better performance than existing analytic transforms, such as the DCT and the KLT.

Sparse coding techniques are based on an overcomplete dictionary [4], [5], [17]. Generally, in an overcomplete dictionary, the number of columns is considerably greater than the number of rows. Therefore, the dictionary has redundant properties and can make the input signal sparser. Owing to this characteristic, the data compression can be improved. However, for sparse coding based on an overcomplete dictionary, finding an appropriate dictionary is generally a nondeterministic polynomial time-hard problem. Therefore, iterative optimization methods, such as the alternating direction method of multipliers (ADMM) [20] and augmented Lagrange multipliers (ALM) [21], and brute-force algorithms, such as basis pursuit [18] and orthogonal matching pursuit [19], are simultaneously used to estimate the approximate value. This inevitably requires considerable time and memory resources for learning. In addition, the overcomplete dictionary is not a square matrix and does not have to be orthogonal. Thus, it does not have the same basis as the transform and has no inverse transform. Because of these problems, this method cannot replace the existing analytic transforms regardless of its capabilities.

To overcome this problem and to design transforms that are more similar to existing transforms, studies on orthogonal dictionaries have been conducted [4]–[6]. The sparse orthonormal transform(SOT) is a transform based on orthogonal dictionary sparse coding [4], [5]. Sezer *et al.* developed this transform with the same characteristics as existing analytic transforms, such as the DCT and KLT, based on orthogonal sparse coding. They theoretically proved that the transform matrix of SOTs is the same as that of the KLT, which is known to be the most optimal, in Gaussian processes and superior to that of the KLT in non-Gaussian processes [5]. In addition, by using an orthonormal dictionary, the computation speed of each iteration is faster than that of overcomplete dictionary-based methods. However, SOTs cannot replace analytic transforms. They also require optimization algorithms, which need thousands of iterations to update the sparse coefficients and dictionary. Therefore, it is difficult to apply this method in real-world applications because of the problems faced by sparse coding techniques.

As mentioned above, the existing sparse coding-based transforms are superior to the existing transforms in terms of compacting data but require a large amount of computation and memory resources. This problem has prevented sparse coding-based transforms from being utilized in real-world applications, despite their good performance. In [5], the authors mathematically proved that the SOT outperforms KLT in general cases. The foundation of our works is to approximate the transform better than the KLT. However, because the transform is based on the sparse coding, its computational burden is high. In [16] and [36], a method of dictionary learning, involving the product of a base dictionary and another matrix, was proposed. Based on this method, the SOT was obtained by expressing the DCT matrix and a rotation matrix [25]. Although the method was developed to express the transform matrix in a compact manner, the computational requirements are increased because the rotation matrix is used like another dictionary. In addition, to compact the rotation matrix, L0 constraints must be added, which prevents the use of the fast optimization used in [5]. From this existing publication, we assume that if the rotation matrix is approximated, an approximated transform better than the KLT can be generated without iterative computations. However, the general n-dimensional rotation matrix has n(n-1)/2 angles, and thus we need to restrict the number of angles to be computed in a reasonable manner. Methods to restrict the rotation matrix have been presented previously [22]–[24]. Fracastoro *et al.* proposed a directional linear transform that is not based on the sparse coding method and called it steerable DCT (SDCT) [22]–[24]. The authors also configured the transform into a DCT matrix and a rotation matrix, but unlike Hou *et al.* [25], the DCT rotation was limited to pairs of bases with the same complexity. When we visualized the basis in [22], we observed that the direction of bases is also rotated in the same angle as that of the rotation matrix. Our work follows the methods presented in [22]–[24] to limit the number of basis pairs to be rotated, thereby not computing all cases. More details regarding the angle computation are discussed in Section III. This study developed a framework to approximate an SOT that is better than the KLT by approximating the rotation matrix.

The DCT has a fast implementation method, and this it can be calculated with fewer operations and provides a basis for enabling suboptimal sparse representation. However, the basis images of the DCT provide optimal compaction for patches with dominant edges in the horizontal and vertical directions and the energy compaction capability is much lower in the case of patches with the same orientation in different directions [37]. In other words, if the DCT basis image can be adjusted according to the direction of the edge of each patch, a transform that can replace the existing transforms with only a few additional computations can be created. As mentioned earlier, the DCT basis already contain information about the horizontal and vertical edges of each image patch (Fig. 1). Using this, we can infer the direction of the edge with minimal computation [30]–[32].

Therefore, if the direction of the dominant edge is calculated and a rotation matrix that rotates the bases to the calculated angle is achieved, a new transform basis can be generated by multiplying the matrix by the DCT matrix. In addition, the framework for making this transform can be applied in real-world application because it only adds a small computational burden to the DCT, which has a fast implementation algorithm.

Our contributions can be summarized as follows:

- Instead of iterative optimizations or other complex mathematical algorithms that require many computations and high memory usage, we only compute the principal angle with pre-computed DCT coefficients. We design a simple but improved transform with higher energy compaction and a few additional parameters. Through this transform, images can be represented by fewer bases.
- We use and modify the concept of the rotation matrix in the SDCT to generate a faster framework than the SDCT and the SOT with slightly lower performance at high compression ratio.
- Because our transform fits the basis to the dominant edge direction, our visual experiments shows fewer block artifacts than the DCT.
- Through experiments, we verify that our transform has better performance than previous analytic methods, with fast implementation and sufficiently similar performance to recently developed methods.

The remainder of this paper is organized as follows: Section II describes other recently proposed transforms with good performance. Section III explains our proposed method and the preliminaries used. In Section IV, we present our experimental results and our analyses. Section V provides the conclusions of our paper.

The main objective of this work is to design a new framework to create a fast and efficient transform as an improvement and substitute for the DCT and KLT. This work only focuses on the transform itself, but with appropriate coding design, our work could have real image and video codec applications, such as JPEG or HEVC.

II. RELATED WORKS

For the past few decades, there have been many attempts to make data-driven transforms achieve better performance than analytic transforms. They exploit various machine-learning methodologies to compact the input data. In this section, we introduce the recent works based on two different approaches.

A. SPARSE CODING-BASED TRANSFORMS

In [4] and [5], SOTs were designed via an orthogonal sparse coding methodology. Sezer *et al.*[4], [5] formulated a transform with an orthonormal matrix and an L_0 norm constraint to the transform coefficients.

$$\min_{G,c}\left\{\|\text{x-Gc}\|^2+\lambda\|c\|_0\right\} \tag{1}$$

where c is the sparse transform coefficient and G is the SOT. To find the optimal G, they used iterative optimization of G and c with the orthogonal Procrustes problem, as it provides a faster solution than other overcomplete matrix-based methods.

These existing publications indicate that SOTs are the principled extensions of KLTs because their transforms are theoretically reduced to KLTs in Gaussian processes, via the following proposition.

Proposition 1 [5] (SOT vs. KLT in Gaussian Processes): Suppose the signals of interest are obtained as realizations of a zero-mean Gaussian process. Then, the SOT is equal to the KLT.

They also showed that SOTs outperformed the KLT in non-Gaussian processes.

Hou *et al.* factorized the SOT matrix to the rotation matrix and DCT basis matrix [25]. They attempted to create a sparse transform matrix for memory efficiency by saving only the sparse rotation matrix, because the DCT matrix can be calculated via mathematical formulation. The proposed approach relies on the results in [16] and [26]. To make compact bases, an L_0 norm constraint is added to the rotation matrix.

$$\begin{aligned}&\min_{R,c}\left\{\|\text{x-TRc}\|^2+\alpha\|c\|_0+\beta\|R\|_0\right\}\\&\text{s.t.}\quad R^TR=RR^T=I,\quad det(R)>0,\end{aligned} \tag{2}$$

where T is the DCT matrix and R is the rotation matrix.

This produces a transform that has a large number of zeros and similar performance to that of the original SOTs in experiments. However, the data compaction and convergence speed remain unaffected. The convergence speed remains relatively slow because they could not exploit a fast orthogonal sparse coding scheme owing to the L_0-constraint for the compact bases. They used the augmented Lagrange multiplier method to make the rotation matrix sparse, resulting in slower convergence.

Rusu *et al.* introduced two orthonormal transform methods based on Householder reflectors [13] and Givens factorization [12]. In [13], the transform was created from the products of a few Householder reflectors. In [12], Rusu and Thompson built a transform using the products of Givens rotations and expanded it into a non-orthogonal transform. Both transforms were designed for fast implementation and low computational complexity. In both factorization methods, the number of parameters to be calculated was reduced.

B. DIRECTIONAL DISCRETE COSINE TRANSFORMS

There have been many attempts to instill directional information into the DCT. The two-dimensional DCT (2D-DCT) consists of two separate 1-D DCTs applied along the vertical and horizontal directions, respectively. Therefore, this DCT is optimal to represent images that have dominant horizontal or vertical edges rather than dominant edges in other directions. Zheng and Fu referred to directional prediction modes in H.264 and developed a directional DCT [27]. In [27], the authors did not apply the 1-D DCT along the

horizontal or vertical direction. Instead, the transform was performed along one of several predefined directions. They defined eight directional modes. Utilizing this development, Drémeau *et al.* [28] extended and exploited this directional DCT to create a set of bases using bintree segmentation and rate-distortion optimization.

Fracastoro *et al.* exploited directional information in the DCT in a different way. They proposed the SDCT [22], [23] based on the graph Fourier transform [29] framework. The graph Fourier transform is defined by the eigenvectors of graph Laplacians. In image processing, an image can be regarded as a graph because each pixel is seen as a node of the graph, and the adjacency of pixels can be described as the edges. When an image is considered as a four-way-connected square grid graph, the graph transform, with all weights equal to one, has the same formulation with the 2D-DCT. The eigenvalues of this Laplacian are larger than one, and thus, the 2D-DCT is not a unique basis. The two bases highlighted in the red box in Fig. 1 show this visually. They have the same eigenvalues with a multiplicity of 2. Fig. 1 shows that the two eigenvectors are orthogonal but related, because they identify the same frequency, but in different directions.

Using this property, the SDCT rotates the eigenspaces spanned by the corresponding eigenvectors as follows:

$$\begin{bmatrix} v^{(k,l)'} \\ v^{(l,k)'} \end{bmatrix} = \begin{bmatrix} \cos\theta_{k,l} & \sin\theta_{k,l} \\ -\sin\theta_{k,l} & \cos\theta_{k,l} \end{bmatrix} \begin{bmatrix} v^{(k,l)} \\ v^{(l,k)} \end{bmatrix} \tag{3}$$

where $0^\circ \leq \theta_{k,l} \leq 90^\circ$, $v^{(k,l)'}$ and $v^{(l,k)'}$ are the bases of the SDCT.

Masera *et al.* generalized the SDCT to any 2D separable transform [24]. A 2D separable transform can be defined by the Kronecker product of a 1D transform. Then, Masera *et al.* substituted $v^{(k,l)}$ and $v^{(l,k)}$ into (3), based on 2D separable transforms. By replacing all the pairs with the rotated ones, they designed a new directional transform. Similar to the previous SDCT, the diagonal directional bases, shown in the diagonal images in Fig. 1, are not rotated.

In this section, the recent transforms that provide better performance than DCT were introduced. However, they have some drawbacks that need to be addressed for their real-world application. In the next section, we propose a framework for designing a novel transform by utilizing the DCT coefficients and their bases, with a few additional computations.

III. PROPOSED METHOD

A. PROBLEM DEFINITION

In the previous section, we introduced the development of several transforms. These efforts achieved better performance than analytic transforms, such as DCT and KLT, but have not been adopted in real-world applications because they require an additional optimization algorithm to determine their optimal basis and parameters. SOTs [4], [5] formulate their problem using (1). They require two unknown variables, G and α; thus, an iterative optimization algorithm is required to find the optimal transform. In addition, the algorithm is sensitive to the regularization parameter, lambda; therefore, it must exploit other algorithms to invoke additional computations to find the best value.

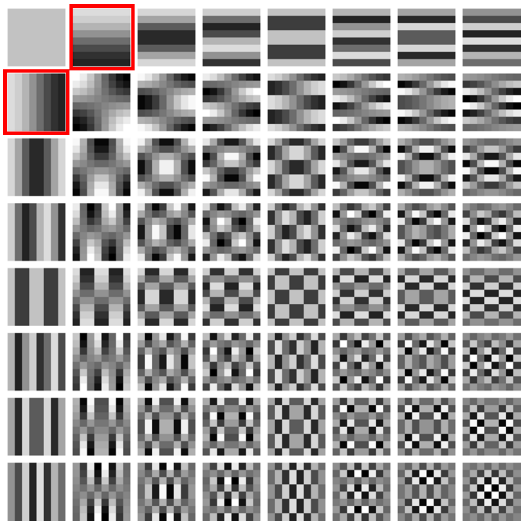


FIGURE 1. Bases of two-dimensional discrete cosine transform. Each basis is linearly independent. The bases in the red boxes have the same frequent basis.

The transform matrix of the SDCT [22], [23] is formulated as follows:

$$V(\theta) = VR(\theta) \tag{4}$$

where θ is the set of angles in (3). It also requires finding the optimal θ to minimize the reconstruction error and depends on brute-force methods and minimization algorithms in a rate-distortion (RD)-optimized manner.

These methods generate good transforms that fit the input data well in terms of data compaction, because of their good optimization algorithms. However, to compute the basis, they have drawbacks in terms of computational cost and time.

In this section, we propose a novel framework to generate a transform to achieve good data compaction close to that of existing transforms, without heavy computations. We call it the "Partially rotated DCT"(PRDCT). To design the PRDCT, we exploit the characteristics of the DCT and generate insertion angles for the rotation matrix. Before testing the proposed PRDCT, we present the details of the methods used.

B. PRELIMINARIES

1) EDGE ORIENTATION IN THE DCT DOMAIN

As the DCT is popular in image compression applications, it is important to analyze and extract information of images in the compressed domain for fast implementation [30]–[32], [40]. In particular, Shen and Sethi [31] designed an edge model in the DCT domain, based on the characteristics of the DCT. The bases of the DCT represent the horizontal and vertical directions or the diagonal directions made by their combinations. The two directional bases have the same edge complexity, according to their order. This

is well illustrated in the bases shown in Fig. 1. The bases in the red box show the same complex edge information in different directions. Shen and Sethi [31] directly extracted low-level features, such as edge orientation, edge offset, and edge strength, from DCT compressed images. Kim and Han [32] proposed a verification method of discontinuity based on the DCT properties and a technique to evaluate it in a compressed domain [32].

We introduce the four metrics that Shen and Sethi [31] suggested for edge orientation, with coefficients based on the 8 x 8 block DCT, as follows:

$$\tan\theta = \left(\sum_{v=1}^{7} C_{0v}\right) / \left(\sum_{u=1}^{7} C_{u0}\right) \tag{5}$$

$$\tan\theta = \left(\sum_{v=1}^{7} H_{0v}\right) / \left(\sum_{u=1}^{7} H_{u0}\right) \tag{6}$$

$$\tan\theta = C_{01}/C_{10} \tag{7}$$

$$\tan\theta = H_{01}/H_{10} \tag{8}$$

where C_{0v} and C_{u0} are the DCT coefficients corresponding to the first row and column bases in Fig. 1, respectively, and H_{0v} and H_{u0} are the coefficients from the compressed stream, via Huffman decoding.

This previous study heuristically demonstrated that these metrics provide a reasonable estimation of edge orientation compared with the Sobel edge operator, and the metric is not affected by edge strength. Kim and Han [32] also showed via experiments that the metric in (1) was precise enough to estimate directions with few errors.

C. PARTIALLY ROTATED DISCRETE COSINE TRANSFORM

In Section II, some recent works on transforms were introduced. We highlighted that they do not only show improvements over analytic transform, but also have major drawbacks that prevent their fast and simple implementations.

To avoid this problem, it is inevitable that iterative optimization techniques and brute-force methods must be eliminated. In particular, iterative methods often require hundreds or thousands of iterations, taking up a large amount of time. To reduce the time, we focused on utilizing the DCT and its coefficients. Our work is based on the understanding that if the 2D-DCT contains sufficient directional information, we can compute reasonable values using which the DCT bases should be appropriately rotated. To this end, we propose a novel framework to create a transform that does not require any constraints, such as predefined sparsity (the number of retained coefficients) or excessive computation. We acknowledge that such constraints and computations make powerful transforms, but they are unsuitable for real-world applications.

Our work starts with the ideas of compact bases-SOT [25] and SDCT [22], [23], in which a transform matrix is derived by multiplication between a basis matrix and rotation matrix. Before introducing our proposed method, we show that the framework, in which the transform matrix is made from the DCT matrix and rotation matrix, is reasonable.

Lemma 1: Given a DCT matrix in $\mathbb{R}^{n\times n}$, there exists a rotation matrix R that satisfies the orthonormal transform matrix H = DR equivalent to or better than the KLT, and which has the same compaction properties with the SOT matrix in Proposition 1.

Proof: We denote a matrix G in $\mathbb{R}^{n\times n}$ as an SOT matrix. We prove this lemma through Proposition 1 by showing that there is a matrix H with the same performance as G. Following the rotation criterion in [38], we assume that R satisfies the condition $RR^T = 1$ and $det(R) = 1$.

Considering an orthonormal matrix A:

$$A = \mathrm{D}^{-1}\mathrm{G}$$

Then

$$det(A) = det(\mathrm{D}^{-1})det(\mathrm{G}) = det(\mathrm{D})det(\mathrm{G}).$$

i) $det(\mathrm{G}) = det(\mathrm{D})$

Because G and D are both orthogonal, $det(\mathrm{G}) = det(\mathrm{D}) = 1$ or $det(\mathrm{G}) = det(\mathrm{D}) = -1$. In both cases, $det(A) = 1$ and $AA^T = 1$ because A is orthonormal. Using the rotation criterion in [38], we can set the rotation matrix R = A. Then we can define

$$\mathrm{H} = \mathrm{D}R = \mathrm{D}\mathrm{D}^{-1}\mathrm{G} = \mathrm{G}.$$

ii) $det(\mathrm{G}) \neq det(\mathrm{D})$

In this case, $det(A) = -1$. Let B be the diagonal matrix with diagonal elements $diag(B) = [-1, 1, \ldots, 1]$. Because $det(A) = -1$ and $det(B) = det(B) = -1$, we can set a rotation matrix $R = AB$. Then

$$\mathrm{H} = \mathrm{D}R = \mathrm{D}AB = \mathrm{G}B.$$

Let h_i and g_i denote column vectors of H and G. For image patch X, $X = \sum_i c_i h_i$ and $X = \sum_i d_i g_i$ where c_i and d_i are scalar. For simplicity, we assume $c_1 > c_2 > \ldots > c_n$ and $d_1 > d_2 > \ldots > d_n$. Then, the best m-term approximation of X is $\widehat{X}_H = \sum_{i\leq m} c_i h_i$ and $\widehat{X}_G = \sum_{i\leq m} d_i g_i$ and each approximation error is $\left\|X - \widehat{X}_H\right\|_F = \sum_{i>m} c_i h_i$ and $\left\|X - \widehat{X}_G\right\|_F = \sum_{i>m} d_i g_i$. Because H = G$B$, $h_i = g_i$ and $c_i = d_i$ for $i > 1$, these two errors are mathematically identical. Thus, the two transform matrices have the same data compaction capabilities.

Using this lemma, we theoretically verify the framework to be defined by a DCT and a rotation matrix, such as the compact bases-SOT [25]. The DCT can be implemented using a fast algorithm, if we design the rotation matrix efficiently, the transform can be generated quickly. However, generating a general n-dimensional rotation matrix requires complex optimizations, as in [25] or the calculation of $\frac{n(n-1)}{2}$ Givens rotations with each angle [12], [39]. However, this is computationally equivalent to other optimization-based transforms [4]–[6]. Therefore, simple implementation of rotation matrix is important for a fast and efficient transform. By approximating the rotation matrix using a small number of

angles, we can efficiently generate a transform matrix similar to KLT in terms of energy compaction.

We use Fig. 2 to estimate the angles that best describe the characteristics of the image patches. Fig. 2 shows the results of rotating the bases by a particular angle using (3), which also represents the directional edges. The DCT is known for its good performance with respect to horizontal and vertical information, based on the basis images in Fig. 1. In [37], even in special cases where the image was very smooth and had discontinuities that existed only vertically or horizontally, the DCT was shown to represent the data slightly more compactly than the KLT. Based on these results, we designed a new framework to create transform matrix by computing the rotation matrix using the dominant direction of each patch.

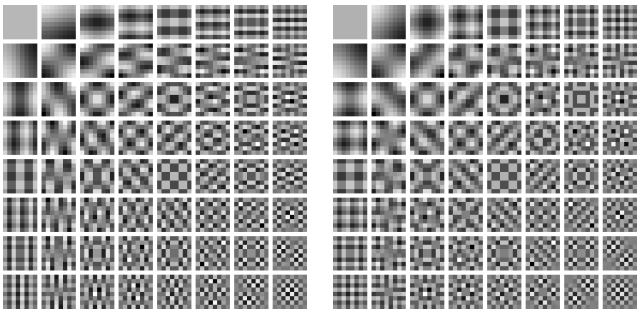


FIGURE 2. Basis images of (a) the SDCT with the same angle 20° for each pair and (b) 45° for each pair.

We designed a new transform to solve the problem by simply inserting two steps into the 2D-separable DCT procedures; approximating the direction and rotating the bases. First, we estimated the main orientation, based on the characteristics of the 2D-DCT. As mentioned above and shown in Fig. 1, the bases of the 2D-DCT contain directional information along the vertical and horizontal directions. For some metrics in the 2D-DCT, we exploit the methodologies presented in [32], which were explained in Section III-B, (5) to (8).

We defined new equations to estimate the angles, based on Equations (5) to (8). We divided the equations into two cases, based on the ratio between the energy of the low-frequency components and the DC component. The ratio was formulated as follows:

$$E_{low} = \frac{\sqrt{\sum_{(u,v)\in[0,1]} C_{uv}^2}}{\sqrt{\sum_{(u,v)\in[0,n-1]} C_{uv}^2}} \tag{9}$$

Eq. (9) indicates the distance between the edge and the center of the patch. Fig. 3 shows patches with different E_{low} values. Fig. 3 (a) and (b) show E_{low} values lower than 0.90. These figures present edges that are further away from the center. In contrast, the edges in Fig. 3 (c) and (d) with high E_{low} values are located near the center of the patch.

The range of the angle is restricted from 0° to 90° because each basis already has a 90°-rotated pair. Because the sign of the DCT coefficients indicates the direction of the edge from center(left or right) with the corresponding basis, we constrained the range of direction using the signs of the first two AC coefficients. We formulated the direction as follows and set the threshold th of E_{low} to 0.90, in the experiments:

$$\theta = \begin{cases} \tan^{-1}\left(\left|\frac{C_{01}}{C_{10}}\right|\right) \\ \quad \text{where} \quad C_{01}C_{10} \geq 0 \text{ and } E_{low} > \text{th} \\ \tan^{-1}\left(\frac{\sqrt{\sum_{v=1}^{n}(C_{0v})^2}}{\sqrt{\sum_{u=1}^{n}(C_{u0})^2}}\right) \\ \quad \text{where} \quad C_{01}C_{10} \geq 0 \text{ and } E_{low} < \text{th} \\ 90^\circ - \tan^{-1}\left(\left|\frac{C_{01}}{C_{10}}\right|\right) \\ \quad \text{where} \quad C_{01}C_{10} < 0 \text{ and } E_{low} > \text{th} \\ 90^\circ - \tan^{-1}\left(\frac{\sqrt{\sum_{v=1}^{n}(C_{0v})^2}}{\sqrt{\sum_{u=1}^{n}(C_{u0})^2}}\right) \\ \quad \text{where} \quad C_{01}C_{10} < 0 \text{ and } E_{low} < \text{th} \end{cases} \tag{10}$$

Then, we found the best set of pairs for each patch to rotate it by a particular angle. We rotated the vector spaces spanned by pairs with specific frequencies and complexities, instead of all pairs. For our pre-computed angles, we classified the basis vectors into five groups based on their complexity and direction. We attempted to find the best pairs for two sizes of blocks, 8×8 and 4×4. A visual description of our separation is shown in Fig. 4. Based on [30], [31], we assume that the calculated angle reflects the dominant orientation but does not include the complicated details inside the patches. In other words, in cases where only the low-frequency components are rotated, they fit more compactly than in cases where all the bases are rotated. The details and comparative experiments are explained in Section IV-B and Fig. 9. We heuristically verified our assumption and found the detailed set to be rotated using Figs. 8 and 9. In case of 8×8 blocks, the set in which the pairs colored blue and yellow in Fig. 4 are rotated is the best, but in case of 4×4 blocks, rotating all bases showed the best performance.

The method of rotating a space formed by two row vectors can be defined by the Givens rotation matrix. Eq. (3) explains the rotation for two vectors. It can also be described in another n^2-dimensional form:

$$R_{ij} = \begin{bmatrix} I_{i-1} & & & & \\ & \cos\theta & & \sin\theta & \\ & & I_{j-i-1} & & \\ & -\sin\theta & & \cos\theta & \\ & & & & I_{n^2-j} \end{bmatrix} \in \mathbb{R}^{n^2\times n^2} \tag{11}$$

where i and j are indexes of the row vectors and n is the size of the blocks.

Let Ω be a set of pairs to be rotated, and i and j be the indexes of the vectors in Ω. The total rotation matrix can then

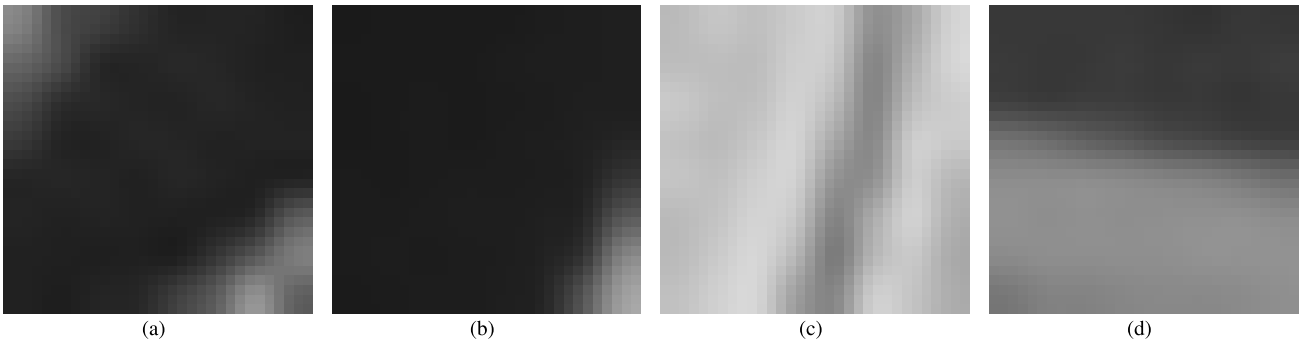


FIGURE 3. Examples of patches with different E_{low} in equation (9). They have different edge patterns by E_{low}. E_{low} = (a) 0.8813 (b) 0.8974 (c) 0.9926 (d) 0.9917.

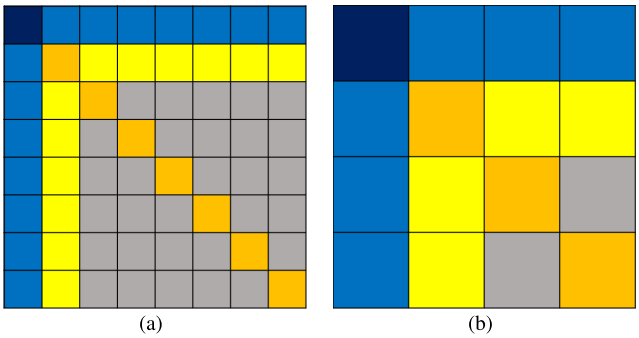


FIGURE 4. Classification of 2D-DCT basis. (a) 8 × 8 block DCT and (b) 4 × 4 block DCT. We classify the basis into 5 groups. The navy block indicates the DC coefficient. The diagonal terms are shown in orange. The blue blocks indicates bases with the lowest frequency. The yellow blocks represent second-leveled complicated basis. The other bases are shown in gray because we do not compare them indivisually.

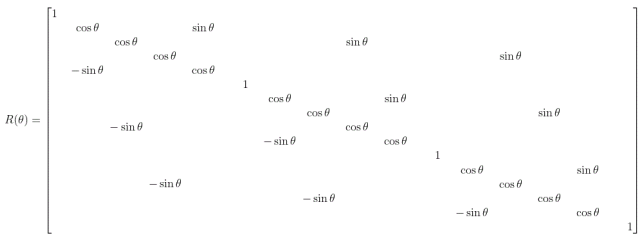


FIGURE 5. Rotation matrix for the 4 × 4 DCT matrix. If θ is known, the matrix can be made directly without any computations.

be computed as follows:

$$R(\theta) = \prod_{(i,j)\in\Omega} R_{ij} \tag{12}$$

Fig. 5 shows $R(\theta)$ for 4 × 4 blocks. The figure shows that $R(\theta)$ is only easy to generate if we know the angle because it can be parameterized by only one variable θ.

Then, the final transform matrix is simple to implement. We designed the matrix by multiplying the rotation matrix and the 2D-DCT matrix.

$$\mathbf{H} = R(\theta)\mathbf{T} \tag{13}$$

where $\mathbf{H}$ is our proposed transform matrix, $\mathbf{T}$ is the 2D-DCT matrix and θ is from (10).

The basis images of $\mathbf{H}$ are visualized in Fig. 6. For more efficient implementations, two methods to reduce computational complexity are optional. First, the coefficient calculation can be worked in a separable way in the same manner as in [24]. Non-separable transforms require more arithmetic operations, such as addition and multiplication. Therefore, separation sharply decreases complexity. Second, the computation of the rotation matrix can be replaced by predefined forms, as shown in Fig. 5. Only if an index table for the location of θ is available can the matrix be generated directly without any computation. Our algorithm is summarized in Algorithm 1 and Fig. 6.

Algorithm 1: Implementation of Partially Rotated Discrete Cosine Transform

Input: Input image X, Block size n
Output: Transform matrix $\mathbf{H}$
1. Convert image X to $n \times n$ patches x;
2. Set $\mathbf{T}$ to $n \times n$ DCT matrix;
3. Compute the DCT coefficients, $C = \mathbf{T}x\mathbf{T}^T$;
4. Compute E_{low} using (9);
5. Compute the angle θ using (10);
6. Compute the rotation matrix $R(\theta)$ using (11) and (12);
7. Apply the rotation matrix $R(\theta)$;

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

We experimentally tested with our methods using the four images in Fig. 7. To create equivalent environments for comparison, we resized the images to 256 × 256 pixels before applying the transforms. To measure the performances of different transforms, we focused on energy compaction by comparing image quality with the number of used coefficients. We compared our method with the KLT, SOTs in [4], SDCT in [24], and DDCT in [27]. To make an equivalent comparison, we attempted to find the optimal values and basis for each transform. SOTs are dependent on the λ in

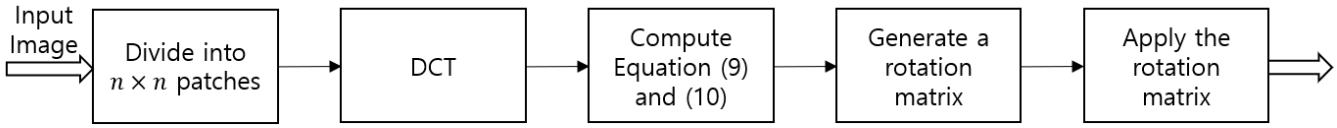


FIGURE 6. Block diagram of Algorithm 1.

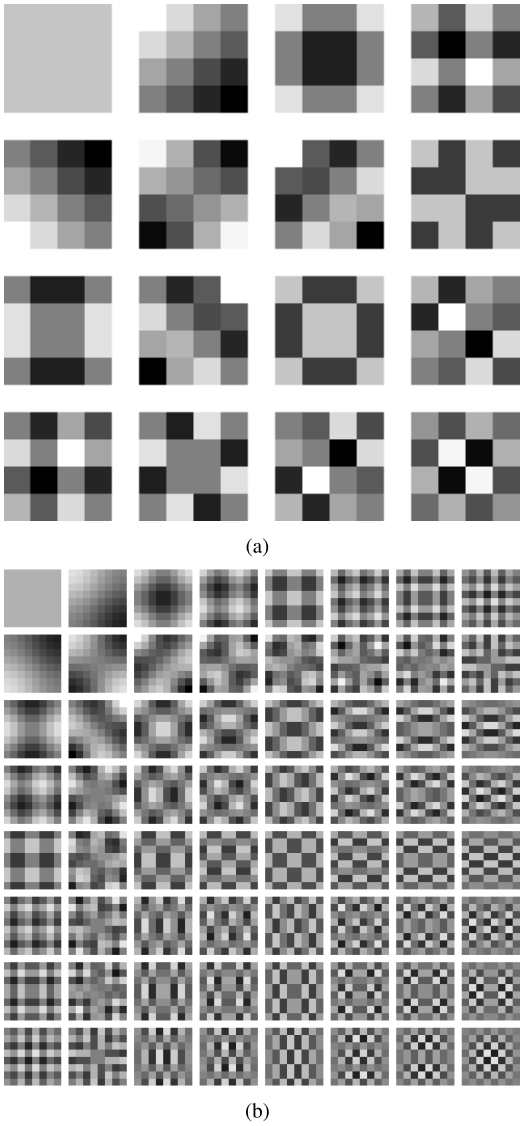


FIGURE 7. Basis of our transform for (a) 4 × 4 and (b) 8 × 8 blocks with $\theta = 45°$.

their formulation and the optimal λ varies with the number of bases used. λ is set to be optimal for each sparsity. In subsection B-2, we attempted to find the best λ with a step size of 0.01. In subsection B-3, the step size of λ was set to 0.1. Then, given the parameter, we implemented the optimization for SOTs with sparse coding formulations. Likewise, SDCT has a different optimal θ for each number of retained coefficients. Therefore, we attempted to find the best value from 0 to 90 for each number of retained coefficients. Furthermore, the DDCT in [27] was designed using eight intra prediction modes of H.264. For achieving the best performance, we found the best mode for each patch using an exhaustive method.

In the proposed method, quantization is not required in the searching step, because we calculate the angle using only (10). However, for efficient transmission in real compression applications, we sufficiently utilize the quantization in the next encoding scheme to reduce the side information.

Before performing the experiments, we identified the best sets of pairs to be rotated using (9). In subsection B-1, we present the best basis set. Then, using the set, we compare our transform with other transforms.

B. EXPERIMENTAL RESULTS

1) PERFORMANCES FOR DIFFERENT SETS OF ROTATED VECTORS

To identify the best basis pairs to be rotated, we set four groups: (1) only the first column and row, except the DC coefficients (blue in Fig. 4 and "Rotating 1" in Figs. 9 and 10); (2) only the second column and row except the diagonal coefficients (yellow in Fig. 4 and "Rotating 2" in Figs. 9 and 10); (3) combined first and second columns and rows, except the DC and diagonal coefficients (blue and yellow in Fig. 4 and "Rotating 1&2" in Fig. 9 and 10); (4) all the pairs, except the DC and diagonal coefficients (blue, yellow and gray in Fig. 4 and "Rotating all" in Figs. 9 and 10).

Fig. 9 shows the 4 × 4 comparisons. From these observations, rotating all pairs produces the best performances. Rotating only two columns and rows shows similar, but slightly lower, performances. This is because there is only one component in "Rotating all" not contained in "Rotating 1&2" and a few high-frequency components in small patches.

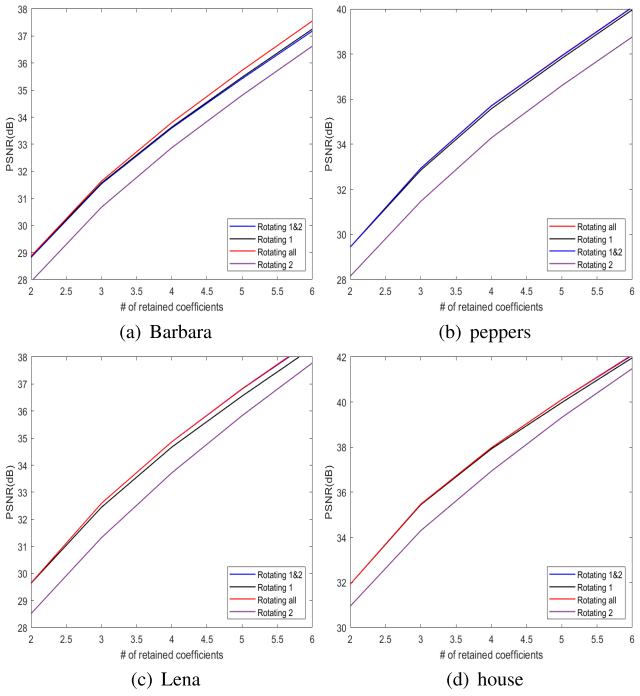
Fig. 10 shows the 8 × 8 results. The result differ from the 4 × 4 results. The results show that, at lower number of bases, "Rotating 1&2" produces the best performances. At a higher numbers of bases, there are some cases in which "Rotating 1" has similar peak signal-to-noise ratios (PSNRs), depending on the images. Generally, we chose "Rotating 1&2" as the best option for our transform. At all levels, "Rotating all" was lower than the two former groups, unlike in the 4 × 4 block experiment. This is reasonable because, for larger blocks, a greater amount of information is present, including more complicated directional information.

2) QUALITATIVE COMPARISONS

In this subsection, we compare the DCT, KLT, SOT [4], SDCT [24], DDCT [27], and our method in terms of energy

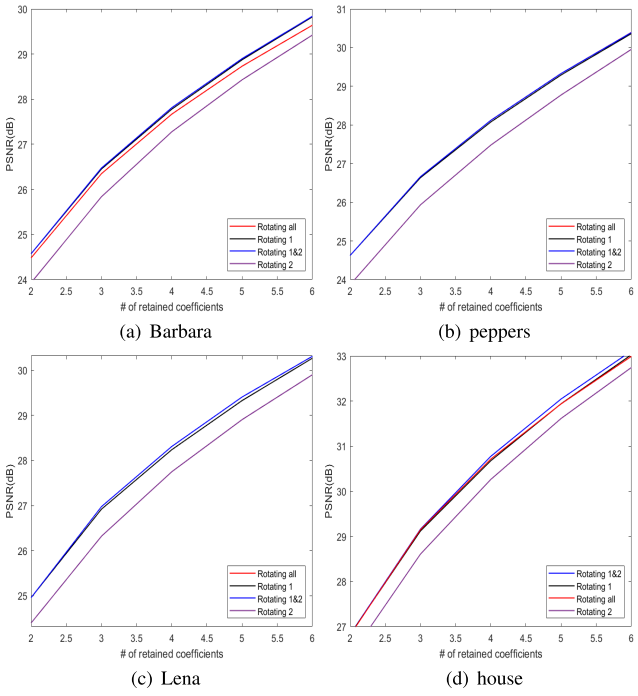
(a) Barbara (b) peppers (c) Lena (d) house

FIGURE 8. Test images.



(a) Barbara (b) peppers (c) Lena (d) house

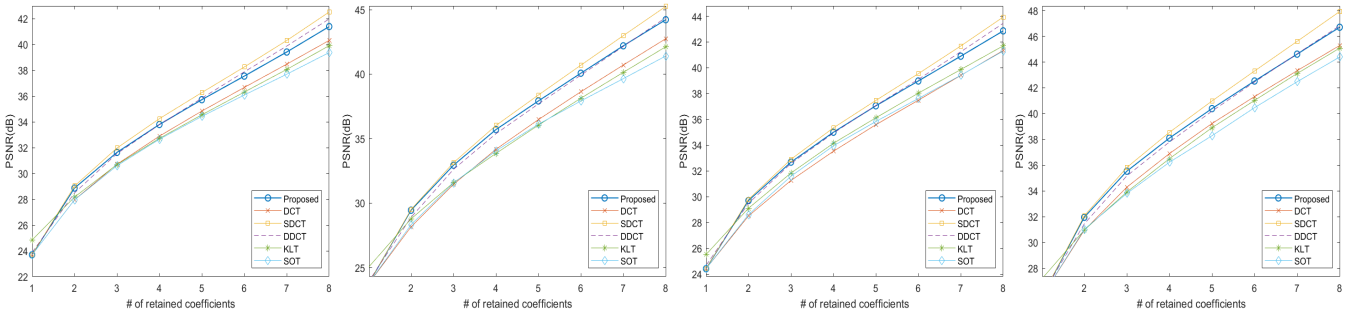
FIGURE 9. PSNR(dB) versus the number of retained coefficient with different sets to be rotated for 4 × 4 blocks.



(a) Barbara (b) peppers (c) Lena (d) house

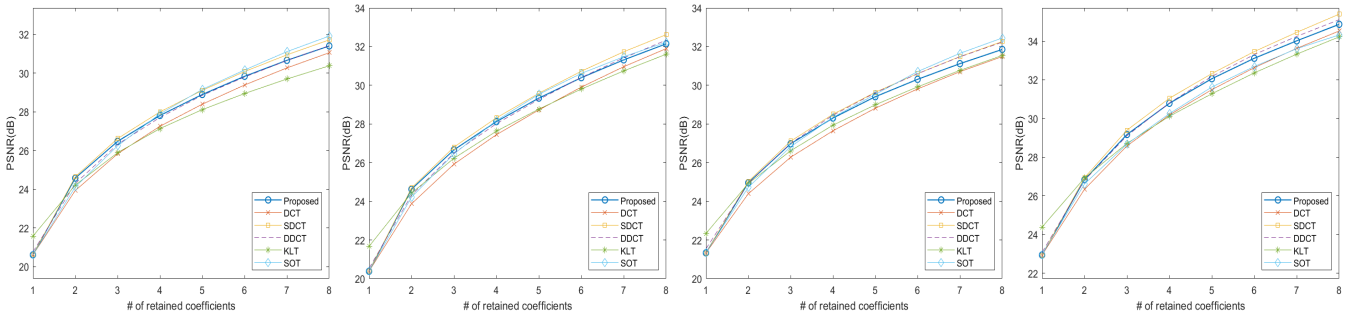
FIGURE 10. PSNR(dB) versus the number of retained coefficient with different sets to be rotated for 8 × 8 blocks.

compaction. For the SDCT, we compare all angles from 0° to 90° for each number of bases used. This is close to that of the SDCT, which is similar to the ground truth for our proposed method, when smaller numbers of coefficients are retained (small values in the x-axis in Figs. 11 and 12). This is because, in such cases, the high frequency components are not often used and the low frequency components are usually retained. The graphs in Figs. 11 and 12 show the objective quality, in PSNR (dB), for each number of retained bases. The results show that our methods are similar to the SDCT at a low number of retained coefficients. The KLT shows higher performance with one retained coefficient, but when a greater number of coefficients are used, it shows lower performance. We also identify that the proposed method has considerably better performance than the original DCT for small block sizes because the directional information is simple. The SOT is theoretically better for data compaction than other analytic methods, as shown in Fig. 12 (a), (b), and (c). This is expected because the transform is learned and trained directly from the input data. In Fig. 12 (d), however, the SOT shows poorer performance than the other transforms. We infer the result stems from the choice of λ in (1). When choosing this parameter, we limited the range from 0.01 to 0.9, with a step size of 0.01. The performance of the SOT is extremely sensitive to λ, with little variation. A poor choice gives worse results than any other transforms. Therefore, to enable an optimal transform via SOT, an additional algorithm is required to find the λ parameter. In Fig. 12 (b), our PRDCT shows relatively similar performance to that of the SOT, but the performance of PRDCT lower in Fig. 12 (c). This can be attributed to the



(a) Barbara (b) peppers (c) Lena (d) house

FIGURE 11. Objective quality comparison: PSNR(dB) versus the number of retained coefficients for 4 × 4 blocks;.



(a) Barbara (b) peppers (c) Lena (d) house

FIGURE 12. Objective quality comparison: PSNR(dB) versus the number of retained coefficients for 8 × 8 blocks;.

characteristics of images in Figs. 8 (b) and (c). The image of peppers is composed of patches with simple and clear directions but the image of Lena includes more complex patches in the hair and feathers that do not have a definite dominant direction. In [22], it was shown that SDCT and DDCT have similar performance in terms of the PSNR. In small sparsity cases, most of the PSNR results of our method are higher than that of the DDCT and other methods. Our method generally provides good performance in cases with small target sparsity and simple directional data.

Fig. 13 shows the structural similarity index measure (SSIM) comparison results. The PSNR is a good measurement of image quality but cannot always guarantee good quality. Therefore, we tested our methods using the SSIM index. The SSIM results show a similar tendency to the PSNR, but the performance is more similar to the SDCT (the semi-ground truth for small numbers of retained coefficients) than to the objective measurement. This means that our method accurately reflects the edge information, which usually indicates the main orientation of the block. For all cases, the PRDCT outperformed the DCT and sufficiently approximated the performance of the SDCT without complex optimization or computations.

As mentioned above, unlike the 2D-DCT, which shows the optimal bases for the horizontal and vertical sides, our method is designed by rotating the DCT basis in the major direction. To show that our method achieves good data compaction performance for diagonal patches, we visually compare various directional patches in Fig. 14. The images shown in Fig. 14 were reconstructed using four transform coefficients for 8 × 8 blocks. By comparing (c) and (e), we note that our method reconstructs the image with fewer block artifacts. Thus, the proposed method exhibits better performance in terms of SSIM than PSNR and for uncomplicated directional data, such as small patches or simple directional textures, than large or complex data.

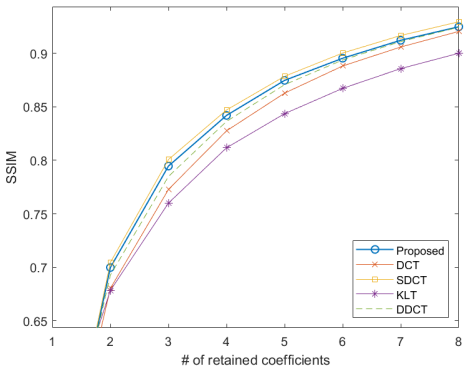
For a more subjective quality comparison, we present the mean opinion score (MOS) in Fig. 15. Experiments were conducted for two sparsity levels, 3 and 5, with 8 × 8 patches. In Fig. 15, we can observe that while our method and SDCT have similar mean values, the SOT presents low scores that are different from the PSNR results. Although the SOT can reconstruct images with small errors, it creates block artifacts that hinder the subjective quality. The proposed method achieved a higher score than DCT, and this result is consistent with Figs. 13 and 14.

3) PROCESSING TIME

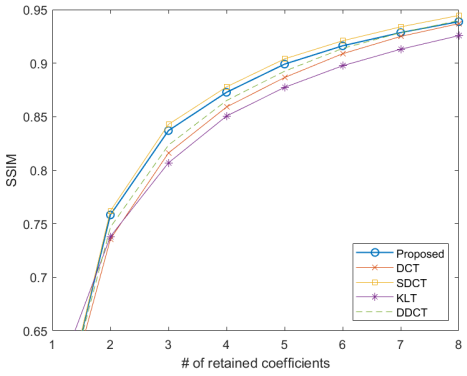
The main contribution of our algorithm is that our approach is robust to fast implementation. To verify this contribution, we compared the processing times of the implementations. For equivalent comparisons, all algorithms were

TABLE 1. Comparison of operation times among transforms for different images of 256 × 256. Our method is faster than the SDCT by a factor of 10 or more.

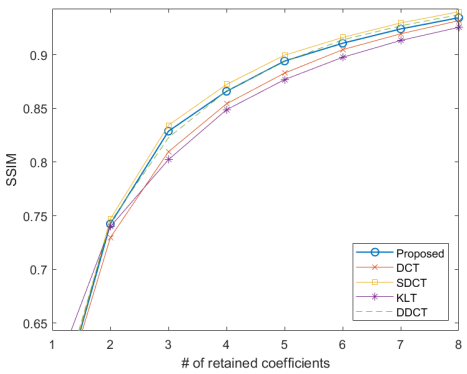
Time(s)		SDCT	SDCT-Q	SOT	Ours
Barbara	4×4	2.7650 ± 0.0469	0.4308 ± 0.0142	2.1238 ± 0.0937	$\mathbf{0.2259 \pm 0.0489}$
	8×8	2.6923 ± 0.1327	0.2783 ± 0.0493	4.1491 ± 0.1082	$\mathbf{0.0840 \pm 0.0104}$
peppers	4×4	2.7425 ± 0.0218	0.4212 ± 0.0211	4.0250 ± 0.1921	$\mathbf{0.2262 \pm 0.0128}$
	8×8	2.6247 ± 0.1246	0.3095 ± 0.0230	9.427 ± 0.2405	$\mathbf{0.0931 \pm 0.0128}$
Lena	4×4	2.7415 ± 0.0434	0.4255 ± 0.0174	2.1373 ± 0.1245	$\mathbf{0.2244 \pm 0.0285}$
	8×8	2.6152 ± 0.0947	0.3275 ± 0.0366	15.5226 ± 1.4745	$\mathbf{0.0850 \pm 0.0131}$
house	4×4	2.7781 ± 0.1203	0.4291 ± 0.0162	3.5321 ± 0.1631	$\mathbf{0.2244 \pm 0.0123}$
	8×8	2.6901 ± 0.1642	0.3220 ± 0.0190	16.6103 ± 0.7792	$\mathbf{0.0810 \pm 0.0067}$



(a) Barbara



(b) peppers



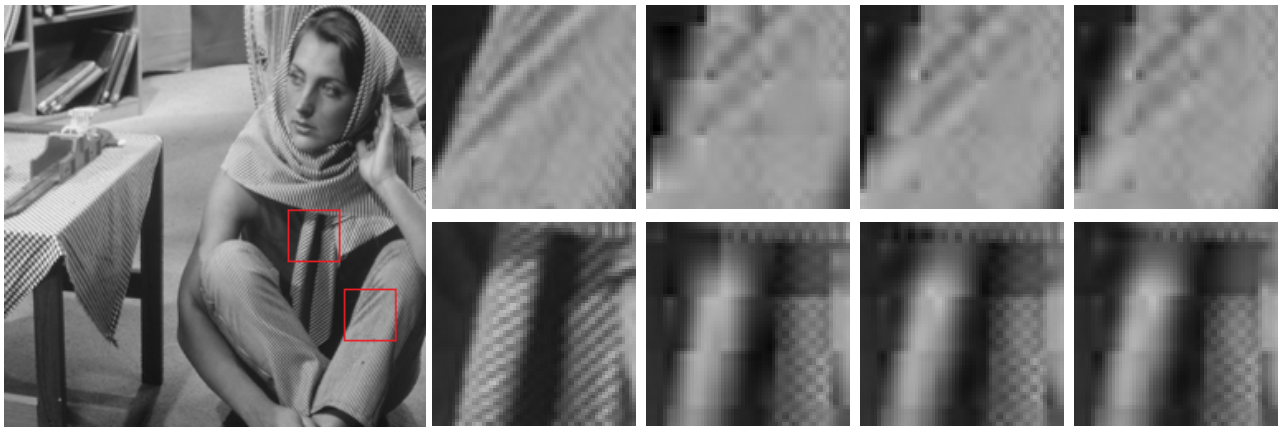
(c) Lena

FIGURE 13. Subject quality comparison: SSIM versus the number of retained coefficient for 8 × 8 blocks.

implemented using MATLAB R2019a in Windows 10 Education on the same computer, equipped with an Intel i7-9700 CPU and 32-GB RAM. Masera *et al.* [24] proposed angle quantization for their SDCT to improve the implementation efficiency. Therefore, we compared two versions of the SDCT. The first is the non-quantized SDCT ("SDCT" in Table 1), and the second is the "SDCT-Q,", which is an SDCT implemented using eight-level angle quantization. We performed 10 experimental trials for each image and block size using these approaches. The average values and errors are listed in Table 1. The bold text indicates the best implementation times. For all images and block sizes, the results show that our approach can achieve the fastest processing times. For 4×4 blocks, the SDCT requires about 2.6 to 2.7 s, but the proposed method requires only approximately 0.22 s. For 8 x 8 blocks, the SDCT requires similar times as that for 4 x 4 blocks, but the proposed method only requires approximately 0.08 s. The SOT implementation time varied widely depending on the image but was higher than our proposed method in all cases. In short, at minimum, our method consumes a tenth of the time required for SDCT and half of that needed for the eight-level quantized SDCT.

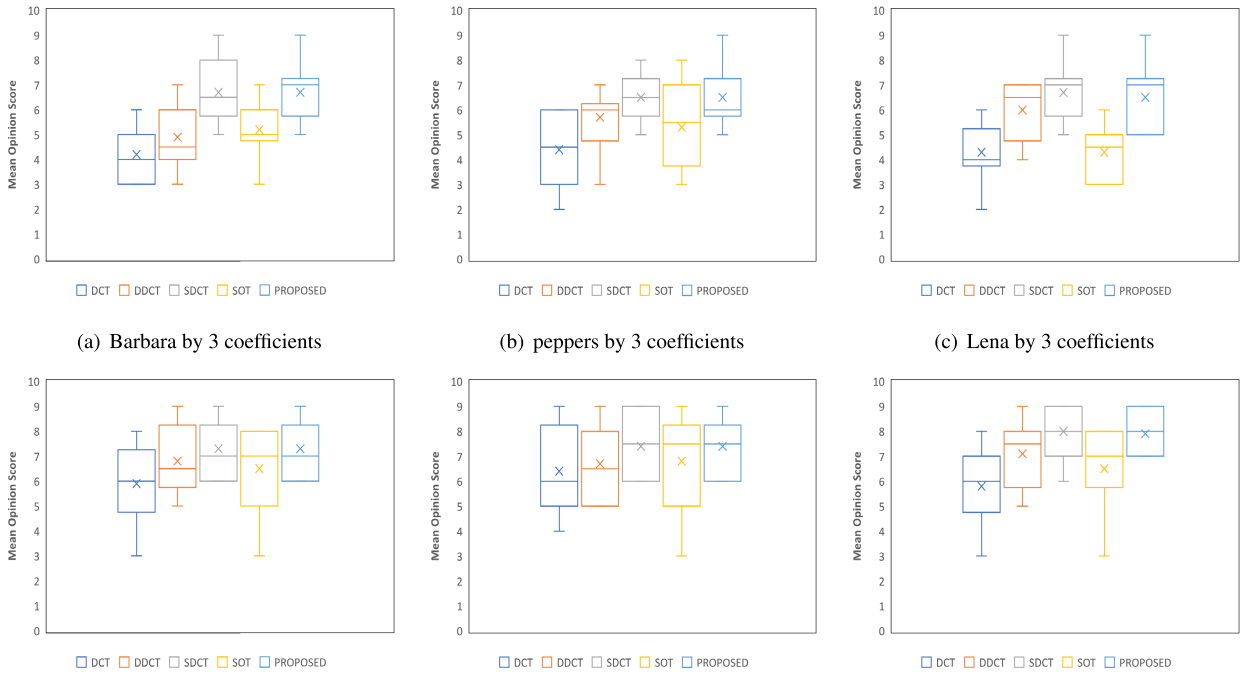
4) COMPUTATIONAL COMPLEXITY

To compare real complexity, we calculated the number of operations, i.e., multiplications and additions for a dataset $\mathbf{X} = (x_1, x_2, \ldots, x_N)$ with x_i of size $n \times n$. In this section, we do not consider vectorization operations. We refer to [41], [42], and [43] for KLT, DCT, and SOT, respectively. The number of operations required to calculate KLT matrix is $(n^4 + 3n^2)N + 9n^6$. The number of operations for SOT is $2n^4N + n^2sN + 21n^6$ for one iteration. The KLT and SOT are computed by non-separable methods, where transform matrices are applied to vectorized data. Two non-separable transforms are applied by the multiplication of a matrix of size $n^2 \times n^2$ and a vector of size $n^2 \times 1$ for N data. The number of operations is $(2n^4 - n^2)N$. Reference [42] presents a fast one-dimensional DCT algorithm. The number of operations of one-dimensional DCT is $2n \log n - n + 2$. Because we use a separable two-dimensional DCT, n one-dimensional DCTs are computed in two ways, i.e., the row and column transforms for N data. An SDCT algorithm consists of DCT computations and the application of the rotation matrix for each quantization level to the entire data. The number of operations for the DCT is equal to that for the conventional DCT, and the application of the rotation matrix requires n^3



(a) Barbara (b) original (c) DCT (d) SDCT (e) Proposed

FIGURE 14. Visual comparison for Barbara for 8 × 8 blocks with four transform coefficients. Our method and SDCT result in less block artifacts than block-based DCT. Furthermore, the proposed method is 10 times faster than the SDCT and twice as fast as than the quantized version.



(a) Barbara by 3 coefficients (b) peppers by 3 coefficients (c) Lena by 3 coefficients

(d) Barbara by 5 coefficients (e) peppers by 5 coefficients (f) Lena by 5 coefficients

FIGURE 15. Mean opinion score (MOS) for each method. (a)-(c) represent the results where images are reconstructed using three coefficients. (d)-(f) indicate the results obtained using images reconstructed by five coefficients.

TABLE 2. Number of operations of each methods. The operations are computed for $n \times n$ image patches. T indicates iterations for convergence of SOT and s is the target sparsity. Q indicates quantization level.

Method	Number of operations	Order of complexity
DCT[42]	$2n(n\log n - n + 2)N$	$O(n^2 N \log n)$
KLT	$3n^4N + 9n^6 + 2n^2N$	$O(Nn^4)$ or $O(n^6)$
SOT	$(2n^4N + n^2sN + 21n^6)T + (2n^4 - n^2)N$	$O(n^4NT)$ or $O(n^6T)$
SDCT	$(2n(n\log n - n + 2) + n^3)QN$	$O(n^3QN)$
Ours	$(4n^2\log n + n^3 + 8n + 1)N$	$O(n^3N)$

multiplications and $n^2(n-1)$ additions. The total number of operations is $(2n(n\log n - n + 2) + n^3)QN$.

For the proposed method, the operations are performed according to Algorithm 1. Eq. (9) requires $n^2 + 3$ multiplications, $n^2 + 1$ additions, and 1 division. Eq. (10) requires $2(2n-1)$ multiplications, $2(2n-2)$ additions, and 1 division for the worst case. Application of the rotation matrix requires n^3 multiplications and $n^2(n-1)$ additions.

The total number of operations is $(4n^2 \log n + n^3 + 8n + 1)N$. For simplicity, Table 2 indicates the largest order of operations. Compared to other methods, except for original DCT, the proposed method requires less operations.

V. CONCLUSION

In this paper, we propose a novel image transformation framework that can efficiently approximate SOTs using the characteristics of 2D-DCT coefficients and their basis functions. Existing transform methods produce good energy compaction but require a long time and considerable resources because of their iterative optimizations. These necessarily result in heavy computational burdens (such as in the SOT) or rely on prior information (e.g., the angle of the SDCTs and regularization parameters such as λ in(1)), which require additional computation. To minimize these burdens, our method does not require any prior information or high-computational iterative methods. By exploiting the properties of 2D-DCT, we can address the aforementioned issues with only a few additional computations and parameters. In addition, this method reduces the block artifacts that usually occur in block-based 2D transforms. Our experiments verify that our method performs well in objective and subjective quality comparisons. In addition, we show that our approach can be implemented faster than other recent approaches. In Table 1, we observe that our proposed method only requires a tenth of the time compared of the SDCT and half that of SDCT-Q. Our method requires about 0.08 s, while the SDCT requires 2.6-2.7 s and SDCT-Q requires about 0.3 s with a minimum reduction of reconstruction errors. This is achieved by eliminating time-consuming computations and approximating the direction and rotation matrix.

This study focused on image compression and defined an appropriate angle and basis set for rotation. In the DCT, a large amount of information tends to be concentrated on a basis with low complexity at the front. Therefore, by using this basis, the direction of the dominant edge can be estimated and only basis sets with low complexity rotated. If this method is applied to other applications in the future, there is potential to create a new transform basis using an appropriate angle estimation method and basis set.

Our work has limitations in that it only focuses on a high level of sparsity, i.e., cases in which only a small number of coefficients are retained. In such cases, the energy compaction of our method is better than that of the original DCT. However, our transforms contain pairs that do not rotate (the gray blocks in Fig. 4 (a)), which means that our transform only performs at similar or slightly higher levels than the original DCT in case with a large number of retained coefficients. This is not a problem typically because the goal of the transform requires only a small number of bases, but this could be an issue in other applications. In future works, we will attempt to determine the value of angles larger than one, to better describe other levels of sparsity and contain more complex directional information.

REFERENCES

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

[2] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[3] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 3736–3745, Dec. 2006.

[4] O. G. Sezer, O. Harmanci, and O. G. Guleryuz, "Sparse orthonormal transforms for image compression," in *Proc. 15th IEEE Int. Conf. Image Process.*, San Diego, CA, USA, Oct. 2008, pp. 149–152.

[5] O. G. Sezer, O. G. Guleryuz, and Y. Altunbasak, "Approximation and compression with sparse orthonormal transforms," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2328–2343, Aug. 2015.

[6] C. Bao, J.-F. Cai, and H. Ji, "Fast sparsity-based orthogonal dictionary learning for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 3384–3391, doi: 10.1109/ICCV.2013.420.

[7] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974, doi: 10.1109/T-C.1974.223784.

[8] A. K. Jain, "A sinusoidal family of unitary transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 4, pp. 356–365, Oct. 1979, doi: 10.1109/TPAMI.1979.4766944.

[9] M. Vetterli, "Wavelets, approximation, and compression," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 59–73, Sep. 2001, doi: 10.1109/79.952805.

[10] W. Hardle, G. Kerkyacharian, D. Picard, and A. Tsybakov, *Wavelets, Approximation, and Statistical Applications* (Lecture Notes in Statistics), vol. 129. New York, NY, USA: Springer, 1998.

[11] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, 2nd ed. Boston, MA, USA: Kluwer, 2001.

[12] C. Rusu and J. Thompson, "Learning fast sparsifying transforms," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4367–4378, Aug. 2017, doi: 10.1109/TSP.2017.2712120.

[13] C. Rusu, N. González-Prelcic, and R. W. Heath, Jr., "Fast orthonormal sparsifying transforms based on householder reflectors," *IEEE Trans. Signal Process.*, vol. 64, no. 24, pp. 6589–6599, Dec. 2016, doi: 10.1109/TSP.2016.2612168.

[14] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 41, no. 7, pp. 909–996, Oct. 1988.

[15] H. Schütze, E. Barth, and T. Martinetz, "Learning efficient data representations with orthogonal sparse coding," *IEEE Trans. Comput. Imag.*, vol. 2, no. 3, pp. 177–189, Sep. 2016, doi: 10.1109/TCI.2016.2557065.

[16] S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for images," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4598–4612, Dec. 2013, doi: 10.1109/TIP.2013.2274384.

[17] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Trans. Signal Process.*, vol. 61, no. 5, pp. 1072–1086, Mar. 2013, doi: 10.1109/TSP.2012.2226449.

[18] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.

[19] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[21] J. Yang and Y. Zhang, "Alternating direction algorithms for l_1-problems in compressive sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.

[22] G. Fracastoro and E. Magli, "Steerable discrete cosine transform," in *Proc. IEEE 17th Int. Workshop Multimedia Signal Process. (MMSP)*, Xiamen, China, Oct. 2015, pp. 1–6, doi: 10.1109/MMSP.2015.7340827.

[23] G. Fracastoro, S. M. Fosson, and E. Magli, "Steerable discrete cosine transform," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 303–314, Jan. 2017, doi: 10.1109/TIP.2016.2623489.

[24] M. Masera, G. Fracastoro, M. Martina, and E. Magli, "A novel framework for designing directional linear transforms with application to video compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 1812–1816, doi: 10.1109/ICASSP.2019.8683527.
[25] J. Hou, H. Liu, L.-P. Chau, Y. He, and J. Chen, "Sparsifying orthogonal transforms with compact bases for data compression," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Jeju, South Korea, Dec. 2016, pp. 1–4, doi: 10.1109/APSIPA.2016.7820905.
[26] S. Ravishankar and Y. Bresler, "Doubly sparse transform learning with convergence guarantees," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 5262–5266, doi: 10.1109/ICASSP.2014.6854607.
[27] B. Zeng and J. Fu, "Directional discrete cosine transforms—A new framework for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 305–313, Mar. 2008, doi: 10.1109/TCSVT.2008.918455.
[28] A. Dremeau, C. Herzet, C. Guillemot, and J.-J. Fuchs, "Sparse optimization with directional DCT bases for image compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, USA, Mar. 2010, pp. 1290–1293, doi: 10.1109/ICASSP.2010.5495421.
[29] D. I. Shuman, S. K. 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," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[30] Z. Qian, W. Wang, and T. Qiao, "An edge detection method in DCT domain," *Procedia Eng.*, vol. 29, pp. 344–348, Jan. 2012, doi: 10.1016/j.proeng.2011.12.720.
[31] B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," *Proc. SPIE*, vol. 2670, pp. 404–414, Mar. 1997, doi: 10.1117/12.234779.
[32] T. Y. Kim and J. H. Han, "Model-based discontinuity evaluation in the DCT domain," *Signal Process.*, vol. 81, pp. 871–882, Apr. 2001, doi: 10.1016/S0165-1684(00)00261-9.
[33] R. Merris, "Laplacian matrices of graphs: A survey," *Linear Algebra Appl.*, vol. 197, pp. 143–176, Jan./Feb. 1994.
[34] C. Zhang and D. Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 106–109, Jan. 2013.
[35] M. S. Nixon and A. S. Aguado, *Feature Extraction & Image Processing for Computer Vision*, 3rd ed. New York, NY, USA: Academic, 2012, pp. 37–82, doi: 10.1016/B978-0-12-396549-3.00002-1.
[36] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553–1564, Mar. 2010, doi: 10.1109/TSP.2009.2036477.
[37] E. Pavez, H. E. Egilmez, Y. Wang, and A. Ortega, "GTT: Graph template transforms with applications to image coding," in *Proc. Picture Coding Symp. (PCS)*, Cairns, QLD, Australia, May 2015, pp. 199–203, doi: 10.1109/PCS.2015.7170075.
[38] J. Stillwell, *Naive Lie Theory*. New York, NY, USA: Springer-Verlag, 2008.
[39] O. I. Zhelezov, "N-dimensional rotation matrix generation algorithm," *Amer. J. Comput. Appl. Math.* vol. 7, no. 2, pp. 51–57, 2017.
[40] B. La, M. Eom, and Y. Choe, "Fast mode decision for intra prediction in H.264/AVC encoder," in *Proc. IEEE Int. Conf. Image Process.*, San Antonio, TX, USA, Sep./Oct. 2007, pp. V-321–V-324, doi: 10.1109/ICIP.2007.4379830.
[41] I. Blanes and J. Serra-Sagristà, "Clustered reversible-KLT for progressive lossy-to-lossless 3D image coding," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Mar. 2009, pp. 233–242, doi: 10.1109/DCC.2009.7.
[42] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-32, no. 6, pp. 1243–1245, Dec. 1984.
[43] C. Bao, J. Cai, and H. Ji, "Fast sparsity-based orthogonal dictionary learning for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3384–3391, doi: 10.1109/ICCV.2013.420.

GIHWAN LEE (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from Yonsei University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering. His research interests include sparse coding, image transformation, and neural networks.

YOONSIK CHOE (Senior Member, IEEE) received the B.S. degree in electrical engineering from Yonsei University, Seoul, South Korea, in 1979, the M.S.E.E. degree in systems engineering from Case Western Reserve University, Cleveland, OH, USA, in 1984, the M.S. degree in electrical engineering from Pennsylvania State University, State College, PA, USA, in 1987, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1990. From 1990 to 1993, he was the Principal Research Staff with the Industrial Electronics Research Center, Hyundai Electronics Company Ltd. Since 1993, he has been with the Department of Electrical and Electronic Engineering, Yonsei University, where he is currently a Professor. His research interests include video coding, video communication, statistical signal processing, and digital image processing.