# Compression-Dependent Transform-Domain Downward Conversion for Block-Based Image Coding

Shuyuan Zhu, *Member, IEEE*, Zhiying He, Xiandong Meng, *Student Member, IEEE*, Jiantao Zhou, *Member, IEEE*, and Bing Zeng, *Fellow, IEEE*

***Abstract*— Transform-domain downward conversion (TDDC) for image coding is usually implemented by discarding some high-frequency components from each transformed block. As a result, a block of fewer coefficients is formed, and a lower compression cost is achieved due to the coding of only a few low-frequency coefficients. In this paper, we focus on the design of a new TDDC-based coding method by using our proposed interpolation-compression directed filtering (ICDF) and error-compensated scalar quantization (ECSQ), leading to the compression-dependent TDDC (CDTDDC)-based coding. More specifically, ICDF is first used to convert each 16 × 16 macro-block into an 8 × 8 coefficient block. Then, this coefficient block is compressed with ECSQ, resulting in a smaller compression distortion for those pixels that locate at some specific positions of a macro-block. We select these positions according to the 4:1 uniform sub-sampling lattice and use the pixels locating at them to reconstruct the whole macro-block through an interpolation. The proposed CDTDDC-based coding can be applied to compress both grayscale and color images. More importantly, when it is used in the color image compression, it offers not only a new solution to reduce the data-size of chrominance components but also a higher compression efficiency. Experimental results demonstrate that applying our proposed CDTDDC-based coding to compress still images can achieve a significant quality gain over the existing compression methods.**

***Index Terms*— Image coding, sub-sampling, interpolation, distortion, quantization.**

Manuscript received May 31, 2017; revised October 6, 2017 and December 19, 2017; accepted February 6, 2018. Date of publication February 14, 2018; date of current version March 12, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61672134 and Grant 61720106004, in part by the National Key Basic Research Program of China under Grant 2015CB351804, in part by the Fundamental Research Funds for Central Universities of China under Grant ZYGX2016J038, in part by the 111 Projects under Grant B17008, in part by the Macau Science and Technology Development Fund under Grant FDCT/022/2017/A1, and in part by the Research Committee, University of Macau, under Grant MYRG2016-00137-FST. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wen Gao. *(Corresponding authors: Jiantao Zhou; Bing Zeng.)*

S. Zhu and Z. He are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

X. Meng is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong.

J. Zhou is with the Faculty of Science and Technology, University of Macau, Macau 999078, China.

B. Zeng is with the Institute of Image Processing, Hong Kong University of Science and Technology, Hong Kong.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2806281

## I. Introduction

THE popular JPEG coding scheme for still images [1] is built upon the block-based framework, which leads to a rather low complexity. Such a low complexity also makes JPEG used more frequently than JPEG2000 [2] for the compression of image signals, although the latter one achieves a higher compression efficiency. In JPEG, the source image is partitioned into non-overlapped blocks and each block is first transformed by the discrete cosine transform (DCT). Then, the transform coefficients are quantized by a uniform quantization and further compressed through an entropy coding. All these operations aim at removing the spatial redundancy within the raw image data, and a number of modifications on them have been developed to improve the coding performance over the past few years [3]–[8]. Meanwhile, some advanced techniques, such as image inpainting [9], [10], graph-based transform [11], [12], object extraction [13], [14], just-noticeable difference (JND) based perceptual optimization [15], and compressive sensing [16], have been applied to image compression, and significant improvements for the coding performance have been achieved.

In image coding, removing the spatial redundancy from the raw image data is always one of the top-priority considerations to reach a high compression. One straightforward way to achieve this goal is to perform the spatial sub-sampling on the source image to reduce its resolution, which accordingly produces a low-resolution (LR) image and forms the pixel-domain downward conversion (PDDC). Then, all compression operations are performed on the LR image to conduct the encoding. After the decoding is completed, an image interpolation is normally performed on the decoded LR image to reconstruct a full-resolution image. The decimation of the source image makes a lower bit-cost, which may offer a potential improvement on the compression efficiency. The coding efficiency of such a PDDC-based compression for grayscale images has been verified in [17], where the LR image is generated by down-sampling the source image according to a uniform 4:1 sub-sampling lattice and the LR image is compressed by JPEG. This method has been effectively improved through introducing more sub-sampling modes to implement the spatial down-sampling for the compression [18].

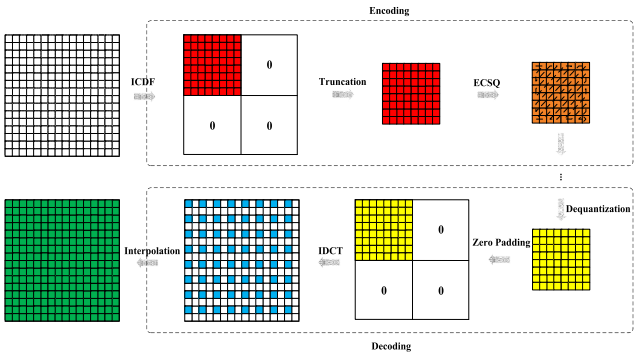



Fig. 1. Framework of CDTDDC-based coding.

Although the PDDC-based coding offers a lower bit-cost, its compression efficiency is often limited by image interpolation. To solve this problem, the adaptive interpolation-directed sub-sampling is proposed to get a better interpolation [19]. It has been noticed that a so-called collaborative adaptive down-sampling and up-conversion (CADU) [20] is developed to preserve the edge information in the down-sized image and it has been used in the PDDC-based coding. Also, the interpolation-dependent image down-sampling (IDID) [21] is designed to build up a block-based image compression. Meanwhile, the super-resolution directed down-sampling (SRDDS) [22] that optimizes the sub-sampling via minimizing the mean square error (MSE) between the source image and the up-converted image has also been used successfully to implement a block-based image compression. To make the sub-sampling based coding more efficient, a rate distortion directed decimation filter is proposed in [23], where the filtering parameters are designed by minimizing both the interpolation distortion and the quantization distortion. Recently, another spatial sub-sampling based compression is proposed in [24], where the sub-sampling of image blocks is determined adaptively by a JND-based threshold. Although the spatial down-sampling based coding offers considerable benefits to image compression [25], [26], how to effectively reduce both the interpolation distortion and the quantization distortion is still a challenge problem in this coding scheme.

So far, most of the advanced PDDC-based methods have been developed for the compression of grayscale images. When these methods are adopted in color image compression after the RGB-to-YCbCr conversion, they are only applied to the chrominance components for an overall bit-cost saving. The most popular PDDC-based color image coding is the 4:2:0 chroma format coding in which the chrominance components are directly sub-sampled along both the horizonal and vertical directions by a factor of 2. Based on such a sub-sampling, the spatial redundancy of the chrominance component is removed effectively and a higher overall compression efficiency is achieved accordingly. This coding format has been used most frequently in the compression of color images and also improves the compression efficiency very significantly.

The image downward conversion may also be implemented in the transform domain by discarding some high-frequency components from the transformed block [27], which produces a small-sized coefficient block and yields the transform domain downward conversion (TDDC). However, when the reconstruction is carried out, a serious quality degradation often happens due to the lack of necessary high-frequency information. This also limits the application of TDDC in practical image coding.

In this work, we focus on the design of a new TDDC-based coding to compress still images, including both grayscale and color images. To achieve this goal, we first propose an interpolation-compression directed filtering (ICDF) to convert each $16 \times 16$ macro-block into an $8 \times 8$ coefficient block. After ICDF is carried out, only the $8 \times 8$ sub-block locating at the top-left corner of the transformed macro-block receives meaningful transform coefficients, while all other positions receive zeros. As a result, this $8 \times 8$ small-sized coefficient block is reserved for the further compression and the other coefficients are all removed, achieving the downward conversion in the transform domain. Then, the small-sized coefficient block is compressed with our proposed error-compensated scalar quantization (ECSQ) to produce a high-quality compression for those pixels locating at some specific positions of the macro-block. In this work, these positions are defined according to the 4:1 uniform sub-sampling lattice, and the pixels locating at these positions are used to interpolate the whole macro-block after compression.

Coupling ICDF and ECSQ together, we build up the compression-dependent TDDC (CDTDDC) for the compression of image signals and the framework of this coding scheme is shown in Fig. 1. Moreover, when the proposed CDTDDC-based coding is adopted in the compression of grayscale images, it will work competitively with the JPEG baseline coding as two coding modes for each macro-block. On the other hand, when it is used to compress color images, it is only performed on two chrominance components after the RGB-to-YCbCr conversion. In this way, it offers not only a new solution to reduce the data-size of color images but also a high compression efficiency.

It is important to point out that our proposed method is quite different from the traditional PDDC-based method. Firstly, we reduce the data-size of the source image in the transform domain rather than in the pixel domain. Secondly, we design the ICDF algorithm to preserve the high-frequency information for the source image in the transform domain, making it more efficient than the adaptive spatial down-sampling [20]–[22] adopted in the PDDC-based method. Thirdly, in our proposed CDTDDC-based coding, the ICDF algorithm is designed not only to make a high-quality reconstruction but also to guarantee a low bit-cost for the compression. Meanwhile, the ECSQ algorithm is proposed to reduce the compression distortion and to improve the final reconstruction quality. In the traditional PDDC-based compression, the adaptive down-sampling is only designed for a a high-quality reconstruction, where the optimization of the compression-cost is ignored.

The rest of this paper is organized as follows. We first present a brief review on the TDDC-based coding and the interpolation-directed image down-sampling in Section II, where both techniques are related to our proposed method.

Then, we describe how to implement our CDTDDC-based coding in Section III in which both ICDF and ECSQ are introduced and the effectiveness of them are verified. Experimental results are presented in Section IV and conclusions are finally drawn in Section V.

## II. Background

### A. TDDC-Based Coding

Let us use $\mathbf{b}$ to denote an $N \times N$ image block, extracted from either a grayscale image or the chrominance component of an RGB-to-YCbCr converted image. After the 2-D DCT is performed on it, we get the transformed block $\mathbf{B}$, where $\mathbf{B} = \mathbf{C}\mathbf{b}\mathbf{C}^T$ and $\mathbf{C}$ is the DCT matrix. Moreover, we can represent $\mathbf{B}$ as

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{00} & \mathbf{B}_{01} \\ \mathbf{B}_{10} & \mathbf{B}_{11} \end{bmatrix} \tag{1}$$

where $\mathbf{B}_{i,j}$ $(i, j = 0, 1)$ is the $N/2 \times N/2$ sub-block of $\mathbf{B}$.

If $\mathbf{B}_{01}$, $\mathbf{B}_{10}$ and $\mathbf{B}_{11}$ are removed from $\mathbf{B}$ so that only $\mathbf{B}_{00}$ is reserved, a small-sized coefficient block $\downarrow \mathbf{B} = \mathbf{B}_{00}$ is obtained, achieving the transform domain downward conversion. Assuming that $\Re$ is composed by all positions of the top-left corner of $\mathbf{B}$ and $\bar{\Re}$ is composed by the other positions of $\mathbf{B}$, removing $\mathbf{B}_{01}$, $\mathbf{B}_{10}$ and $\mathbf{B}_{11}$ from $\mathbf{B}$ may also be represented by filling $\bar{\Re}$ with zeros in the practical compression, i.e.,

$$\tilde{\mathbf{B}} = \begin{bmatrix} \downarrow \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{2}$$

After this downward conversion, $\downarrow \mathbf{B}$ is encoded by performing the quantization and entropy coding on it. When one needs to decode the compressed block, the quantized $\downarrow \mathbf{B}$ should be first padded with zeros as Eq. (2) implies. After such a zero-padding, the 2-D inverse DCT (IDCT) is performed on the padded coefficient block to get the decoded block. This leads to the TDDC-based image coding.

Based on the above discussions, it is found that 3/4 components of $\mathbf{B}$ are dropped and only a small-sized coefficient block remains. Clearly, this leads to a considerable bit-cost saving for the compression of $\mathbf{b}$. However, such a TDDC-based coding often causes a serious quality degradation over all pixels due to losing some necessary high-frequency information. Although the advanced transform domain up-conversion techniques [28]–[30] have been proposed to estimate these lost information, the reconstruction quality is still very limited.

### B. Interpolation-Directed Image Down-Sampling

Image interpolation refers to composing a new pixel by using a weighted average of some neighbouring pixels that are all known. The weighting/interpolation coefficients may be adaptively determined according to the content of the image [31]–[33] or assigned with some fixed values such as the bicubic, bilinear and Lanczos [34] interpolations.

In this work, we use $\mathbf{b}_L$ to denote the available LR image block of size $N/2 \times N/2$, which is obtained by sub-sampling the source block $\mathbf{b}$ along both horizontal and vertical directions by a factor of 2. Also, we use $\mathbf{b}_I$ to denote the $N \times N$ image block obtained by performing an interpolation on $\mathbf{b}_L$. Assuming that $\Pi$ is composed by all interpolated positions of $\mathbf{b}_I$, the interpolation for any position $(m, n) \in \Pi$ may be represented as

$$\mathbf{b}_I(m, n) = \mathbf{w} \cdot \mathbf{v}_{\Omega(m,n)}, \quad \forall (]m, n) \in \Pi \tag{3}$$

where $\mathbf{v}_{\Omega(m,n)}$ is a column-vector composed by $K$ nearest available pixels around $(m, n)$ in $\mathbf{b}_I$ and $\mathbf{w}$ denotes the weighting vector including $K$ interpolation coefficients.

In the 1-D representation for image interpolation, let us use $\mathbf{x}_L$ to stand for a column-vector obtained by concatenating all columns of $\mathbf{b}_L$. Then, the interpolation to get a column-vector $\mathbf{x}_I$ for the 2-D interpolated block $\mathbf{b}_I$ may be represented as

$$\mathbf{x}_I = \mathbf{H}\mathbf{x}_L \tag{4}$$

where $\mathbf{H}$ is the interpolation matrix which consists of the corresponding interpolation coefficients used in Eq. (3), and these coefficients are interlaced with zeros to form $\mathbf{H}$. In our work, we compose $\mathbf{H}$ exactly following the way used in [21].

In the PDDC-based coding, the compression efficiency is always limited by the interpolation. To make the interpolation more efficient, the IDID-based pre-filtering is applied to the source image block to get the optimal $\mathbf{x}_L$ for interpolation. In general, the optimal $\mathbf{x}_L$ is determined by solving the minimum mean square error (MMSE) related problem as

$$\tilde{\mathbf{x}}_L = \arg\min_{\mathbf{x}_L} \|\mathbf{x} - \mathbf{H}\mathbf{x}_L\|_2^2. \tag{5}$$

where $\mathbf{x}$ is a column-vector generated by concatenating all columns of $\mathbf{b}$. As a result, we may get the interpolation-optimized $\mathbf{x}_L$ as

$$\tilde{\mathbf{x}}_L = (\mathbf{H}^T\mathbf{H})^{-1}(\mathbf{H}^T\mathbf{x}). \tag{6}$$

In the practical application, $\tilde{\mathbf{x}}_L$ should be converted back into the 2-D block for the further processing. The simulation results in [21] has demonstrated that adopting IDID in the PDDC-based coding to build up the IDID-based coding improves the compression efficiency for the low bit-rate grayscale image coding.

In practice, an efficient interpolation would produce a high-quality output for IDID. However, most efficient interpolation methods (such as [31]–[33]) are designed according to some statistical properties of the image signal. When these methods are adopted in IDID, the interpolation coefficients of $\mathbf{H}$ need to be determined adaptively based on the input signal. In practical image coding, the statistical properties of the image signal often change very drastically before and after compression. If the adaptive interpolation is adopted in the IDID-based coding, such a change will produce different interpolation matrices for encoder and decoder, which would limit the coding efficiency of the IDID-based compression. Transmitting the interpolation matrix from encoder side to decoder side may solve this problem, but it needs lots of additional bits to represent every interpolation matrix for each image block, which would also degrade the compression efficiency. On the other hand, using the adaptive interpolation will introduce much more additional computations, not only at the encoder side but also at the decoder side. To guarantee a high coding efficiency as well as a low complexity, the interpolation

with fixed coefficients is preferred for the IDID-based image coding.

The design of IDID aims at a high-efficient interpolation by preserving enough prior information of the source image. However, after such a down-sampling, the pixels within a down-sized block are not so closely correlated as before. Therefore, performing the transform coding on the down-sized block cannot guarantee a high compression efficiency, often resulting in a higher bit-cost. To solve this problem, we propose a new down-sampling method to reduce the image data-size in the transform domain rather than in the pixel domain. The most important advantage of our proposed down-sampling is that it not only transfers the high-frequency information from the source image to the down-sized image but also guarantees a lower bit-cost for the compression.

## III. Proposed Method

In this work, we propose a new TDDC-based coding to compress each $16 \times 16$ macro-block. In general, the transform domain downward conversion is implemented by using an advanced pre-filtering and the reconstruction of a compressed macro-block is achieved based on the image interpolation. The pre-filtering used in our work not only facilitates an efficient interpolation at the decoder side but also produces a specific transformed macro-block (as Eq. (2) represents) at the encoder side, which leads to the transform domain downward conversion. To this end, an interpolation-compression directed filtering (ICDF) is proposed and performed on each macro-block before compression. Then, an error-compensated scalar quantization (ECSQ) is designed to reduce the distortion occurring on some specific pixels of the compressed macro-block, where these pixels will be collected to implement the interpolation for the reconstruction of a macro-block.

### A. Interpolation-Compression Directed Filtering

We present our analysis with the 1-D representation for the corresponding 2-D image block in this subsection. Firstly, we compose a transform matrix $\Psi$ by performing the Kronecker product, denoted as $\otimes$, on the DCT matrix $\mathbf{C}$ as $\Psi = \mathbf{C} \otimes \mathbf{C}$. Then, we concatenate all columns of $\mathbf{B}$ to form a coefficient vector $\mathbf{X}$. Based on $\Psi$ and $\mathbf{x}$, $\mathbf{X}$ may be obtained by $\mathbf{X} = \Psi\mathbf{x}$, which implies that the 2-D transform has been achieved via the 1-D operation. Finally, the inverse transform will be implemented as

$$\mathbf{x} = \Psi^{-1}\mathbf{X}. \tag{7}$$

In this work, the compressed pixels locating at some pre-determined positions (denoted as $\Omega$) of an $N \times N$ macro-block will be used to reconstruct a decompressed macro-block via image interpolation. Specifically, these positions are defined according to the 4:1 uniform sub-sampling lattice. Regardless of the compression, let $\mathbf{x}_\Omega$ be composed by $K = N^2/4$ pixels locating at $\Omega$ in $\mathbf{b}$. Typically, these pixels are picked out from $\mathbf{b}$ along the vertical direction. If we compose a $K \times N^2$ matrix $\mathbf{D}$ by using $K$ rows of $\Psi^{-1}$ corresponding to $K$ selected positions belonging to $\Omega$, we may get

$$\mathbf{x}_\Omega = \mathbf{D}\mathbf{X}. \tag{8}$$

Moreover, if the positions of $\mathbf{X}$'s elements are changed, some necessary column-swaps must be performed on $\mathbf{D}$ to guarantee the correct output for $\mathbf{x}_\Omega$. More specifically, let us use $\mathbf{X}_\Re$ to represent a column-vector composed by $K$ coefficients locating at $\Re$ in $\mathbf{B}$ and use $\mathbf{X}_{\bar{\Re}}$ to denote another column-vector composed by the other $(N^2 - K)$ coefficients locating at $\bar{\Re}$. Then, we combine $\mathbf{X}_\Re$ and $\mathbf{X}_{\bar{\Re}}$ together to construct $\mathbf{X}$ as $\mathbf{X} = [\mathbf{X}_\Re^T \quad \mathbf{X}_{\bar{\Re}}^T]^T$. Meanwhile, we use $K$ corresponding columns of $\mathbf{D}$ to compose a matrix $\mathbf{D}_\Re$ and use the other $(N^2 - K)$ columns of it to compose another matrix $\mathbf{D}_{\bar{\Re}}$. Coupling $\mathbf{D}_\Re$ and $\mathbf{D}_{\bar{\Re}}$ together, we get $\mathbf{D} = [\mathbf{D}_\Re \ \mathbf{D}_{\bar{\Re}}]$ which makes

$$\mathbf{x}_\Omega = [\mathbf{D}_\Re \ \mathbf{D}_{\bar{\Re}}] \begin{bmatrix} \mathbf{X}_\Re \\ \mathbf{X}_{\bar{\Re}} \end{bmatrix}. \tag{9}$$

Based on Eqs. (4) and (8), the interpolation by using $\mathbf{x}_\Omega$ may be represented by the coefficient vector as

$$\mathbf{x}_I = \mathbf{H}\mathbf{D}\mathbf{X}. \tag{10}$$

According to the IDID-based solution, an interpolation-optimized $\mathbf{X}$ may be determined as

$$\tilde{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathbf{x} - \mathbf{H}\mathbf{D}\mathbf{X}\|_2^2 \tag{11}$$

Therefore, the optimal $\mathbf{X}$ is calculated as

$$\tilde{\mathbf{X}} = \big((\mathbf{H}\mathbf{D})^T(\mathbf{H}\mathbf{D})\big)^{-1}\big((\mathbf{H}\mathbf{D})^T\mathbf{x}\big). \tag{12}$$

In this work, we aim at an optimal $\mathbf{X}$ which not only produces an efficient interpolation for a better reconstruction, but also effectively controls the compression cost, i.e., the bit-rate, for practical coding. However, the IDID-based solution only targets an optimal interpolation, but ignores the cost for the compression.

It has been noticed that a lower compression cost is highly-related to the sparsity of the transformed image signal in image compression. In other words, a very sparse transformed image signal often leads to a much easier compression. DCT has shown an excellent energy-compaction capability [35], [36], which makes the image signal sparse enough in the transform domain, as proved by the Laplacian distribution of transform coefficients [37]–[39]. Therefore, it has been adopted in all of the block-based image and video codecs [1], [40], [41] to help compress image and video signals. On the other hand, the sparsity of the transformed image signal is usually evaluated by the $l_0$-norm.

To achieve a better interpolation as well as a lower compression cost, the $l_0$-norm of the transformed block is involved in the interpolation-directed pre-filtering, leading to our proposed ICDF algorithm. Moreover, after such a filtering, the optimal interpolation-compression directed transform coefficients are obtained and they are determined as

$$\tilde{\mathbf{X}} = \arg\min_{\mathbf{X}} \big(\|\mathbf{x} - \mathbf{H}\mathbf{D}\mathbf{X}\|_2^2 + \lambda\|\mathbf{X}\|_0\big). \tag{13}$$

The minimization of $l_0$-norm is an *NP*-hard problem and thus the $l_1$-norm is often used as an alternative solution in practical applications. As a result, the optimal transform coefficients may be obtained as

$$\tilde{\mathbf{X}} = \arg\min_{\mathbf{X}} \big(\|\mathbf{x} - \mathbf{H}\mathbf{D}\mathbf{X}\|_2^2 + \lambda\|\mathbf{X}\|_1\big). \tag{14}$$

**Algorithm 1**

**Input:** The image vector $\mathbf{x}$;

**Output:** The interpolation-compression directed optimal coefficient vector $\tilde{\mathbf{X}}$.

**1. Initialization:**

(1) Make transform on $\mathbf{x}$: $\mathbf{X}^{(0)} = \mathbf{X}^{(1)} = \mathbf{\Psi x}$;

(2) Initialize $t_0$, $t_1$, $L_1$, and $\lambda_1$.

**2. Iteration:**(stop until $\|\mathbf{X}^{(i)} - \mathbf{X}^{(i-1)}\|_2^2 \leq T$ or $i > Iter_{max}$, where $T$ is the pre-defined threshold and $Iter_{max}$ is the pre-defined maximum number of iteration.)

(1) $\alpha_i = (t_{i-1} - 1)/t_i$;

(2) $\mathbf{\Theta}^{(i)} = \mathbf{X}^{(i)} + \alpha_i(\mathbf{X}^{(i)} - \mathbf{X}^{(i-1)})$;

(3) $\mathbf{\Delta}^{(i)} = \mathbf{\Theta}^{(i)} - \frac{1}{L_i}(\mathbf{HD})^T\big((\mathbf{HD})\mathbf{\Theta}^{(i)} - \mathbf{x}\big)$;

(4) $\mathbf{X}^{(i+1)} = soft\big(\mathbf{\Delta}^{(i)}, \lambda_i/L_i\big)$
(where $soft$ stands for the soft thresholding function with threshold $\lambda_i/L_i$);

(5) Perform TDDC-directed operation on $\mathbf{X}^{(i+1)}$:
If $X_k^{(i+1)} \in \mathbf{X}_{\bar{\Re}}^{(i+1)}$, $X_k^{(i+1)} = 0$;

(6) Update parameters:
$t_{i+1} = 0.5(1 + \sqrt{1 + 4t_i^2})$;
$\lambda_{i+1} = max(0.75\lambda_i, \Lambda)$
(where $\Lambda$ is the pre-defined threshold);
$L_{i+1} = 1.5L_i$;

(7) $i = i + +$.

The solution to Eq. (14) may be effectively achieved by using the iterative shrinkage-thresholding algorithm (ISTA) [42], [43]. In our work, we aim at a TDDC-based coding and want to make the coefficients locating at $\bar{\Re}$ be zeros after the ICDF is carried out. To achieve this goal, we design a compression-dependent solution based on the fast iterative shrinkage-thresholding algorithm (FISTA) [44] and make the coefficients locating at $\bar{\Re}$ change to zeros in the process of solving Eq. (14). The details about how to implement the ICDF algorithm have been summarized in Algorithm 1, where all parameters and pre-defined thresholds are initialized by the corresponding default values in [44].

Similarly to the IDID-based coding, applying ICDF to image compression also needs to avoid producing the mismatched interpolation matrices at the encoder side and at the decoder side. In this work, we choose to use an interpolation with fixed coefficients to implement ICDF as well as the final reconstruction. To this end, we verify the effectiveness of different interpolation methods by separately integrating them in our proposed method to compress images. More details and results about such a verification will be presented in Section IV, where the bicubic interpolation performs best among all compared methods. Therefore, the bicubic interpolation is adopted in the ICDF algorithm to implement our proposed compression method.

To verify the convergence of ICDF, we apply it to the $16 \times 16$ macro-blocks of two popular grayscale images of Lena and Boat. The PSNR values of interpolated images versus the iteration number of ICDF are illustrated in Fig. 2.

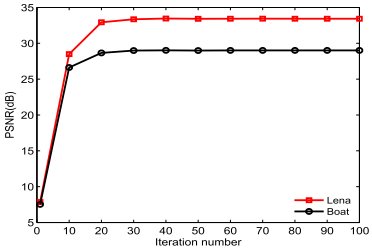



Fig. 2. Verification of the convergence for ICDF.

According to Fig. 2, the quality of the interpolated image tends to be stable when the iteration number exceeds 30, which not only reveals that the ICDF converges very fast but also indicates that the maximum number of iteration should be set to 30 when it is used in practice.

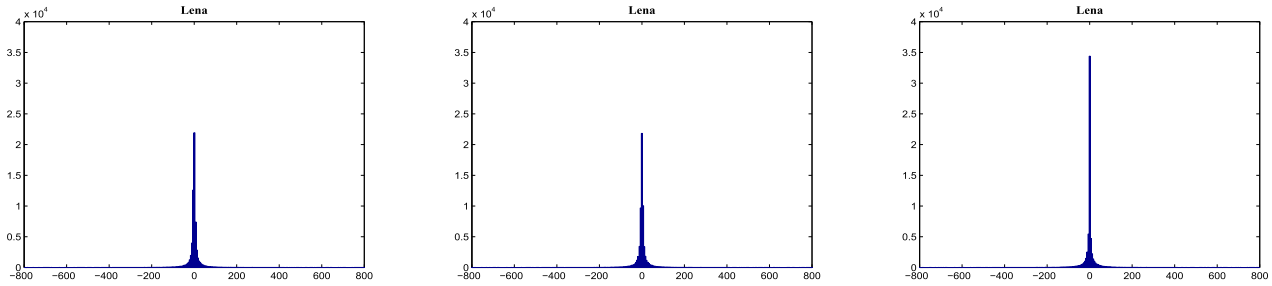
Moreover, the distributions of DCT coefficients locating at $\Re$ in all transformed macro-blocks for both test images are shown in Figs. 3 and 4. It is obvious that these coefficients concentrate much closer to zeros by performing ICDF on the macro-block. This also reveals that using ICDF for the downward conversion strengthens the sparsity of the transformed image signal, which offers a potential improvement on the coding efficiency.

Finally, both the IDID-based solution, i.e., Eq. (12), and the ICDF-based solution, are adopted in the TDDC-based coding to generate optimal transform coefficients for the compression. Then, we make a comparison between three coding methods: IDID-based coding, ICDF-based coding, and JPEG baseline coding. The rate-distortion (R-D) curves for these coding methods have been shown in Fig. 5. It can be seen from Fig. 5 that the ICDF-based coding outperforms the IDID-based method, especially achieving a lower bit-rate when used to compress one image with the same quality. However, the ICDF-based coding only works better than the JPEG coding at the low bit-rate. We will further improve it in the following subsection and also propose two corresponding solutions to make it more practical for the compression of grayscale and color images.

### *B. Error-Compensated Scalar Quantization*

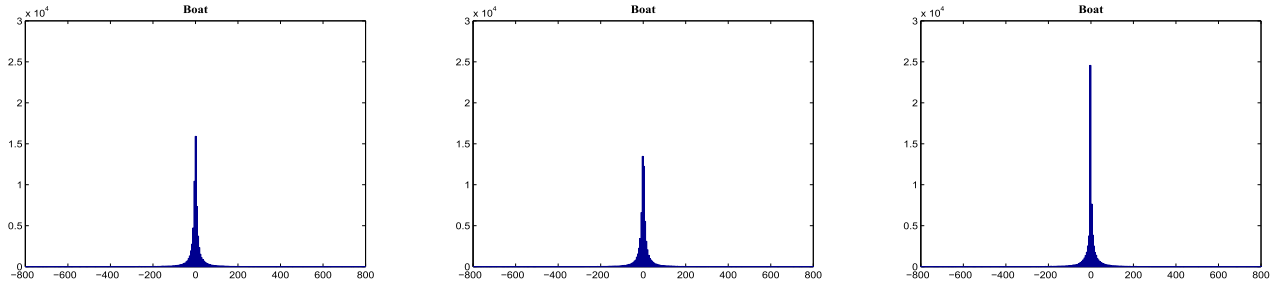
The application of ICDF in the TDDC-based coding aims at a better interpolation and a lower compression cost. However, when the compression happens, the interpolation efficiency as well as the coding efficiency will be limited by the distortion occurring on those filtered pixels (denoted as $\tilde{\mathbf{x}}_{\mathbf{\Omega}}$) that will be used for interpolation. To solve this problem, we purpose to reduce the sum of square error (SSE) distortion of $\tilde{\mathbf{x}}_{\mathbf{\Omega}}$ as much as possible via controlling the quantization error of the transformed macro-block based on an error-compensated scalar quantization (ECSQ).

Due to the use of a unitary transform in the compression, the SSE distortion of all pixels and the SSE distortion of all transform coefficients are exactly identical. In this work, we only focus on the SSE distortion occurring on $\tilde{\mathbf{x}}_{\mathbf{\Omega}}$ rather than the whole filtered block. First of all, the coefficient vector obtained by using ICDF may be represented as $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_{\Re}^T \quad \tilde{\mathbf{X}}_{\bar{\Re}}^T]^T$, where $\tilde{\mathbf{X}}_{\Re}$ is a column-vector composed by $K$




(a) (b) (c)

Fig. 3. Distribution of DCT coefficients at $\in \Re$ for "Lena": (a) without filtering, (b) with IDID-based filtering, and (c) with ICDF.




(a) (b) (c)

Fig. 4. Distribution of DCT coefficients at $\in \Re$ for "Boat": (a) without filtering, (b) with IDID-based filtering, and (c) with ICDF.

coefficients locating at $\Re$ in a coefficient block after our proposed filtering and $\tilde{\mathbf{X}}_{\bar{\Re}}$ is composed by the other $(N^2 - K)$ coefficients. Since $\tilde{\mathbf{X}}_{\bar{\Re}} = \mathbf{0}$, according to Eq. (9), we may get $\tilde{\mathbf{x}}_{\Omega} = \mathbf{D}_{\Re}\tilde{\mathbf{X}}_{\Re}$. after compression, the SSE distortion of $\tilde{\mathbf{x}}_{\Omega}$ may be calculated as

$$\begin{aligned} \|\tilde{\mathbf{x}}_{\Omega} - \hat{\mathbf{x}}_{\Omega}\|_2^2 &= (\tilde{\mathbf{X}}_{\Re} - \hat{\mathbf{X}}_{\Re})^T \mathbf{D}_{\Re}{}^T \mathbf{D}_{\Re}(\tilde{\mathbf{X}}_{\Re} - \hat{\mathbf{X}}_{\Re}) \\ &= (\tilde{\mathbf{X}}_{\Re} - \hat{\mathbf{X}}_{\Re})^T \mathbf{W}(\tilde{\mathbf{X}}_{\Re} - \hat{\mathbf{X}}_{\Re}) \\ &= \Delta\tilde{\mathbf{X}}^T \mathbf{W} \Delta\tilde{\mathbf{X}} \end{aligned} \tag{15}$$

where $\mathbf{W} = \mathbf{D}_{\Re}{}^T\mathbf{D}_{\Re}$, $\hat{\mathbf{x}}_{\Omega}$ is the compressed vector for $\tilde{\mathbf{x}}_{\Omega}$, $\hat{\mathbf{X}}_{\Re}$ is the quantized vector for $\tilde{\mathbf{X}}_{\Re}$, and $\Delta\tilde{\mathbf{X}}$ consists of the quantization errors of $K$ transform coefficients locating at $\Re$. According to Eq. (15), it is noticed that $\mathbf{W}$ is not an identity matrix. Therefore, the SSE distortion of $K$ specific pixels locating at $\Omega$ in a filtered block and the SSE distortion of $K$ coefficients in the transformed block are not identical. More specifically, the SSE distortion of $\tilde{\mathbf{x}}_{\Omega}$ (denoted as $SSE_{\Omega}$) and the quantization error of $\tilde{\mathbf{X}}_{\Re}$ is related through a complicated weighting process, where the weighting coefficients are determined according to $\mathbf{W}$. On the other hand, a smaller $SSE_{\Omega}$ may be possibly achieved by controlling the quantization errors of $\tilde{\mathbf{X}}_{\Re}$.

Based on the Cholesky factorization, $\mathbf{W}$ may be decomposed as

$$\mathbf{W} = \Phi^T \Phi \tag{16}$$

where $\Phi$ is a $K \times K$ upper triangular matrix

$$\Phi = \begin{bmatrix} \varphi_{0,0} & \varphi_{0,1} & \cdots & \varphi_{0,K-1} \\ 0 & \varphi_{1,1} & \cdots & \varphi_{1,K-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \varphi_{K-1,K-1} \end{bmatrix}. \tag{17}$$

By substituting Eqs. (16) and (17) into Eq. (15), we get

$$\begin{aligned} SSE_{\Omega} &= \Big(\sum_{i=0}^{K-1} \varphi_{0,i}\Delta\tilde{X}_i\Big)^2 + \Big(\sum_{i=1}^{K-1} \varphi_{1,i}\Delta\tilde{X}_i\Big)^2 \\ &\quad + \cdots + \Big(\varphi_{K-1,K-1}\Delta\tilde{X}_{K-1}\Big)^2 \\ &= \sum_{k=0}^{K-1} E_k \end{aligned} \tag{18}$$

where $\Delta\tilde{X}_i \in \Delta\tilde{\mathbf{X}}$, and

$$E_k = \Big(\sum_{i=k}^{K-1} \varphi_{k,i}\Delta\tilde{X}_i\Big)^2. \tag{19}$$

It is found from Eq. (18) that many cross-terms are involved to determine $SSE_{\Omega}$ so that the uniform quantization as used in the JPEG baseline coding will no longer be the best strategy to get a minimal $SSE_{\Omega}$. However, $SSE_{\Omega}$ cannot be minimized by simply taking partial derivatives on $\Delta\tilde{X}_i$ because the quantization is a non-linear operation and is always required to implement the data compression.

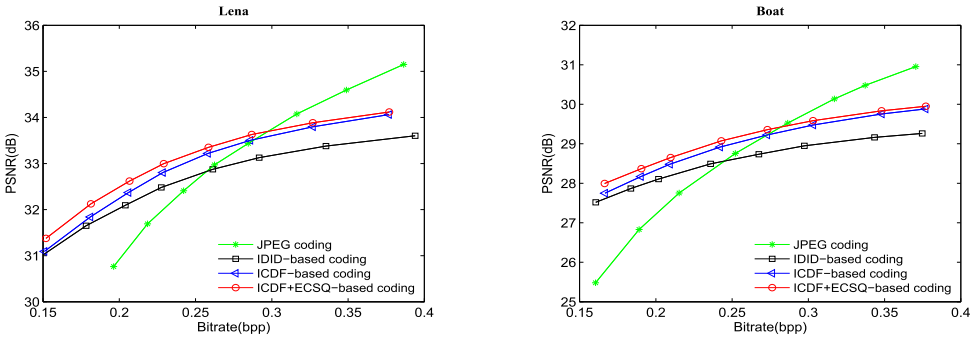



Fig. 5. Verifications of the effectiveness for ICDF and ECSQ.

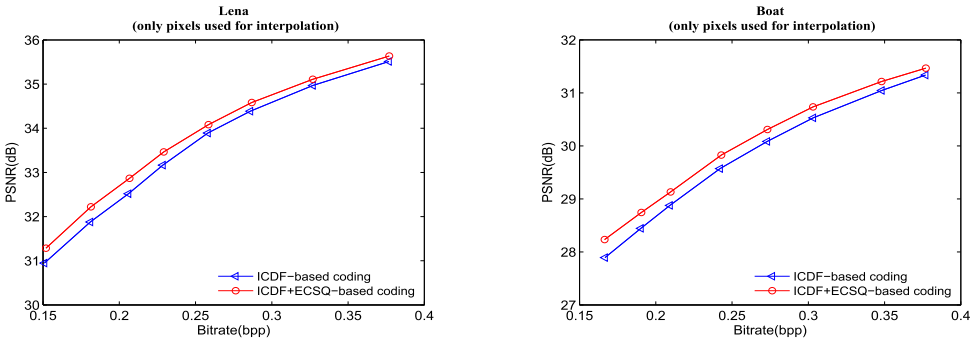



Fig. 6. Verification of the effectiveness for ECSQ (only pixels used for interpolation are included).

In fact, the quantization error can be positive and negative in a random fashion. When two quantization errors have different signs, the resulting cross-term would contribute a reduction to $SSE_{\Omega}$. Different combinations of such cross-terms will lead to different distortion and the minimal one may be found by trying an exhausted-search on all possible distortion combinations. However, such a search strategy becomes impossible in practical applications and it is thus necessary to find a more efficient way to reduce the distortion for $\tilde{\mathbf{x}}_{\Omega}$.

According to Eq. (18), it also reveals that a reduced $SSE_{\Omega}$ may be achieved by minimizing each $E_k$ separately. When $k < K-1$, let's use $\Delta_k$ to represent $(1/\varphi_{k,k}) \sum_{i=k+1}^{K-1} \varphi_{k,i} \Delta \tilde{X}_i$. As a result, $E_k$ may be further represented as

$$\begin{aligned} E_k &= \varphi_{k,k}^2 \left(\Delta \tilde{X}_k + \Delta_k\right)^2 \\ &= \varphi_{k,k}^2 \left(\left(\tilde{X}_k - Q(\tilde{X}_k)\right) + \Delta_k\right)^2 \end{aligned} \tag{20}$$

where $\tilde{X}_k \in \tilde{\mathbf{X}}_{\Re}$, $Q(\cdot)$ stands for the quantization carried out on the transform coefficient and $\Delta \tilde{X}_k = (\tilde{X}_k - Q(\tilde{X}_k))$. Thus, no mater how $\tilde{X}_k$'s are quantized, the quantization errors produced on them will contribute a compensation term $\Delta_k$ to $\tilde{X}_k$. Based on Eq. (20), as long as this compensation term is added into $\tilde{X}_k$, the quantization on $\bar{X}_k = \tilde{X}_k + \Delta_k$ will yield an item of $\left(\bar{X}_k - Q(\bar{X}_k)\right)^2 = \left(\Delta \bar{X}_k\right)^2$, which would be smaller (statistically) than $\left(\Delta \tilde{X}_k + \Delta_k\right)^2$. This finally leads to a new quantization algorithm, i.e., the proposed ECSQ, and it has been summarized in Algorithm 2. After ECSQ is performed on each $\tilde{X}_k$ $(0 \le k < K-1)$, we may get a smaller SSE distortion $SSE_{\Omega}$ through combining all $E_k$ together.

We verify the effectiveness of the proposed ECSQ algorithm by applying it to the test images of Lena and Boat. More

Fig. 7. Test images. From left to right and top to bottom: *Lena*, *Man*, *Goldhill*, *Boat*, *Barbara*, *Baboon*, *House*, *Airplane*, *Flowers*, *Peppers*, *Parrot* and *Butterfly*.

specifically, it is performed on each $16 \times 16$ macro-block to make a further compression after the ICDF-based filtering. Firstly, the overall R-D performance for the whole image is presented in Fig. 5. Then, the R-D performance evaluated based on all pixels used for interpolation is shown in Fig. 6. It is found from Fig. 6 that the reconstruction quality of those pixels involved in interpolation has been improved after ECSQ is carried out. As a result, an overall quality gain has been achieved accordingly, as proved by the better R-D performance offered by ECSQ in Fig. 5.

### *C. Our CDTDDC-Based Coding*

The proposed ICDF is used to implement the downward conversion in the transform domain for an efficient interpolation as well as a low bit-cost. Meanwhile, the proposed ECSQ makes a low compression distortion for those pixels used for interpolation. Coupling them together, we build up our CDTDDC-based coding for the compression of image signals. The framework for the CDTDDC-based coding has been shown in Fig. 1. According to Fig. 1, it is found

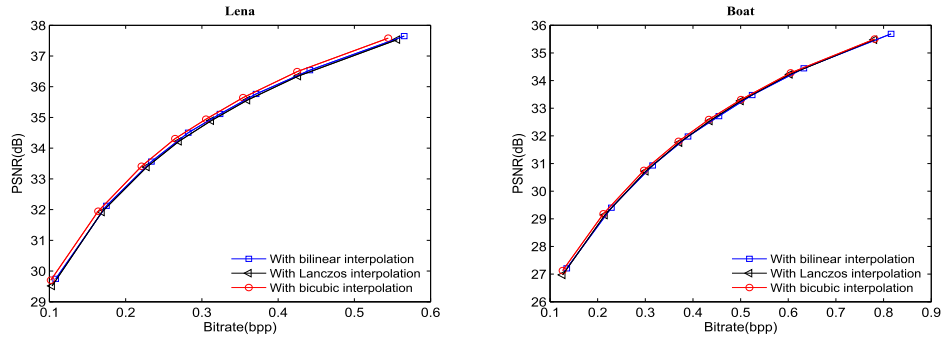



Fig. 8. Comparison of using different interpolations in our proposed method for grayscale image coding.

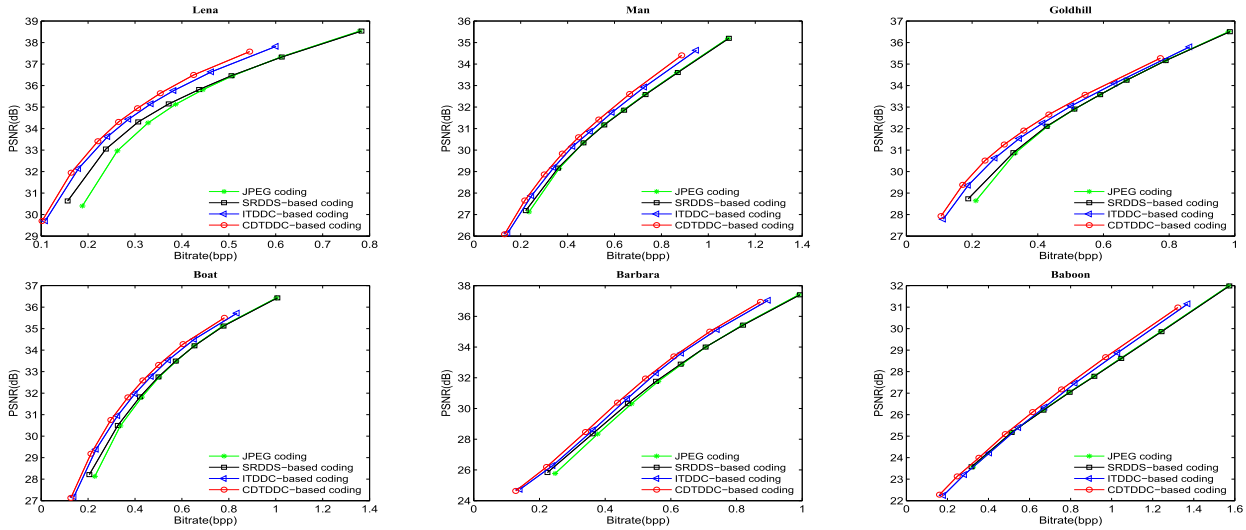



Fig. 9. R-D performances for grayscale image coding by using different methods.

**Algorithm 2**

**1. Initialization:**

(1) Quantize $\tilde{X}_{K-1}$: $\hat{X}_{K-1} = Q(\tilde{X}_{K-1})$;

(2) Calculate the quantization error:

$\Delta\tilde{X}_{K-1} = \tilde{X}_{K-1} - \hat{X}_{K-1}$.

**2. Error compensation:**

*For* $k = (K-2) : -1 : 0$

(1) Calculate the compensation amount:

$\Delta_k = (1/\varphi_{k,k}) \sum_{i=k+1}^{K-1} \varphi_{k,i} \Delta\tilde{X}_i$;

(2) Perform the compensation on $\tilde{X}_k$: $\bar{X}_k = \tilde{X}_k + \Delta_k$;

(3) Quantize $\bar{X}_k$: $\hat{X}_k = Q(\bar{X}_k)$;

(4) Calculate the quantization error:

$\Delta\tilde{X}_k = \tilde{X}_k - \hat{X}_k$.

*End for*

that most additional operations are carried out only at the encoder side, and we will make a detailed discussion about the computational complexity in next section. In our work, the ICDF algorithm is performed on each $16 \times 16$ macro-block if it is needed. We will further verify the compression efficiency of our proposed CDTDDC-based coding through applying it to compress both grayscale and color images.

## IV. Experimental Results

We verify the effectiveness of our CDTDDC-based coding by using it to compress grayscale and color images, respectively. Meanwhile, the comparison between our CDTDDC-based coding and some existing compression methods is also presented. To make a fair comparison, all compared methods are integrated into the block-based coding scheme, i.e., the JPEG coding, instead of the JPEG2000 coding. Some popular images with the resolution of $512 \times 512$, including six grayscale images and six color images as shown in Fig. 7, are used in our simulation. A number of simulations are carried out and more discussions about the experimental results are presented in this section.

### A. Application in Grayscale Image Coding

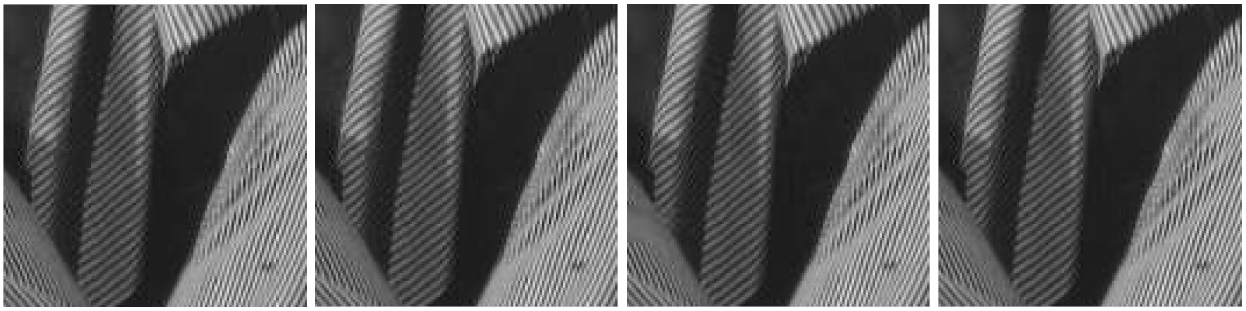
According to the preliminary experimental results shown in Fig. 5, using image interpolation in the CDTDDC-based coding limits the coding efficiency for the compression of grayscale images at the high bit-rate. To solve this problem, we propose to make the CDTDDC-based coding and the JPEG baseline coding work competitively as two coding modes for the compression of each macro-block. More importantly, the used coding mode for each macro-block is determined according to a simple but efficient rate-distortion-

(a) (b) (c) (d)

Fig. 10. Image portions of "Lena" (0.33 bpp): (a) JPEG coding (32.44 dB), (b) SRDDS-based coding (32.75 dB), (c) ITDDC-based coding (33.10 dB), and (d) CDTDDC-based coding (33.24 dB).

(a) (b) (c) (d)

Fig. 11. Image portions of "Boat" (0.50 bpp): (a) JPEG coding (31.49 dB), (b) SRDDS-based coding (31.49 dB), (c) ITDDC-based coding (32.25 dB), and (d) CDTDDC-based coding (32.47 dB).



(a) (b) (c) (d)

Fig. 12. Image portions of "Barbara" (0.71 bpp): (a) JPEG coding (31.49 dB), (b) SRDDS-based coding (31.49 dB), (c) ITDDC-based coding (32.22 dB), and (d) CDTDDC-based coding (32.62 dB).

optimization (RDO) based criterion, which is composed by the product of MSE and bit-count of the macro-block. Meanwhile, it needs 1 overhead bit to represent the mode information for each macro-block.

To conduct the simulations, the JPEG baseline coding, SRDDS-based coding [22], interpolation-directed TDDC (ITDDC) based coding [45] and our CDTDDC-based method are all applied to six grayscale images, as shown in Fig. 7, to implement the compression.

Firstly, we make a comparison of using different interpolation methods in our proposed CDTDDC-based compression. In this work, three interpolation methods, i.e., bicubic, bilinear and Lanczos interpolations, are integrated in the ICDF algorithm at the encoder side and used to implement the image interpolation at the decoder side. Some comparison results are shown in Fig. 8 and these results demonstrate that using the bicubic interpolation achieves the best compression performance. Therefore, it will be adopted in our proposed CDTDDC for the grayscale image coding.

Then, the R-D performances of all compared methods are shown in Fig. 9. It is seen from Fig. 9 that our proposed method outperforms the other compression methods at all bit-rates while the SRDDS-based coding method only performs better than the JPEG baseline coding at the low bit-rate.

Meanwhile, we present some visual results to make a more comprehensive and clearer comparison between different methods. Some compressed image portions are shown

(a) (b) (c)

Fig. 13. Three components of "House": (a) Y-component, (b) Cb-component, and (c) Cr-component.

in Figs. 10-12. It is very clear from these visual results that the compressed images with less artifacts are obtained by using our CDTDDC-based method and the apparent visual quality gains have been achieved.

### *B. Application in Color Image Coding*

The color image compression usually happens in the YCbCr space where the input RGB image is converted into the YCbCr image by using a linear transform [46]. The compression operations, including DCT, scalar quantization and entropy coding, are performed on both the luminance component (Y) and two chrominance components (Cb and Cr) to implement the compression. The luminance component is always filled with plenty of textures, while less textures exist in the chrominance component. An example to compare such a difference has been illustrated in Fig. 13, where the luminance component and two chrominance components of the test image House (as shown in Fig. 7) have been presented after the RGB-to-YCbCr conversion. Apparently, the chrominance component is much smoother than the luminance component, which also makes the interpolation more effective for the chrominance component to compose a full-resolution image from its sub-sampled counterpart. As a result, the chrominance components are always sub-sampled uniformly before compression and the image interpolation is then used to reconstruct a full-resolution chrominance image after compression, leading to the 4:2:0 chroma format coding.

Similar to the 4:2:0 chroma format coding, when we apply our CDTDDC-based coding to compress the converted YCbCr image, performing it on the chrominance components should be more effective than on the luminance component. This has been verified by the experimental results shown in Fig. 14, where the R-D performances of the JPEG coding and the CDTDDC-based coding for the compression of each single component of test image House are all presented. Moreover, IDID [21] is also applied to each chrominance component to implement the spatial down-sampling for the compression. It is also seen from Fig. 14 that using our proposed CDTDDC-based method to compress chrominance components works more efficiently than using the IDID-based method.

Then, we design two CDTDDC-based color image coding methods by performing both ICDF and ECSQ on different combinations of luminance and chrominance components. The first one is implemented by applying them to both luminance component and chrominance components, and the second one is achieved by just performing them on the chrominance components. The comparison results for these two methods are presented in Fig. 15. Compared with the JPEG baseline coding, using the second coding strategy offers a consistent coding gain over all bit-rates, while implementing the compression by using the first method only gets the performance improvement at the low bit-rate. Based on the above discussions and the experimental results shown in Figs. 14 and 15, we propose to build up the CDTDDC-based coding for color images by applying both ICDF and ECSQ only to the chrominance components after the RGB-to-YCbCr conversion.

Although our CDTDDC-based coding is performed on the chrominance components to implement the data-size reduction, it is quite different from the traditional 4:2:0 chroma format coding. The 4:2:0 chroma format coding gets a reduction of data-size in the pixel domain, while our CDTDDC-based method achieves the same size reduction in the transform domain, offering a new solution to reduce the data-size for chrominance components. Moreover, the CDTDDC-based coding proposes a novel way to reduce the compression distortion, which potentially leads to a better reconstruction.

Before we make a further comparison of different compression methods, we firstly conduct a comparison of using different interpolation methods to implement the CDTDDC-based coding for color images. The bicubic, bilinear and Lanczos interpolations are still integrated in the ICDF algorithm and also used to interpolate chrominance images. With these interpolation methods, some comparison results obtained by using the CDTDDC-based coding to compress test images House and Flowers (as shown in Fig. 7) have been presented in Fig. 16. The results shown in Fig. 16 demonstrate again that using the bicubic interpolation in our proposed method gets the best compression performance, making it adopted in our proposed CDTDDC-based coding for color images.

Several coding methods are used to compress color images in this work, including our CDTDDC-based coding, the JPEG baseline coding and the adaptive color-space transform (ACT) based coding [47]. Meanwhile, IDID [21] is applied to both chrominance components to form the 4:2:0 format and it accordingly leads to the IDID-based coding method.

We make a comparison of these methods by conducting a number of simulations. Moreover, the adaptive color-space transform proposed in [47] is also adopted in the CDTDDC-based coding, generating the ACT+CDTDDC-based compression for a further comparison. All compared methods are applied to six color images as shown in Fig. 7. Particularly, the 4:2:0 coding format is adopted in the last three methods.

The R-D performances of all compared methods are shown in Fig. 17 in which the average PSNR of three color channels is adopted. It is found from Fig. 17 that our CDTDDC-based coding method always outperforms the JPEG coding and the IDID-based coding although it cannot exceed the ACT-based method for most test images. The RGB-to-YCbCr conversion is optimally designed in the ACT-based method, which makes it get a better compression performance. After we integrate such an optimal conversion in our CDTDDC-based coding, the generated ACT+CDTDDC-based compression achieves the highest compression efficiency, and it significantly improves the coding performance

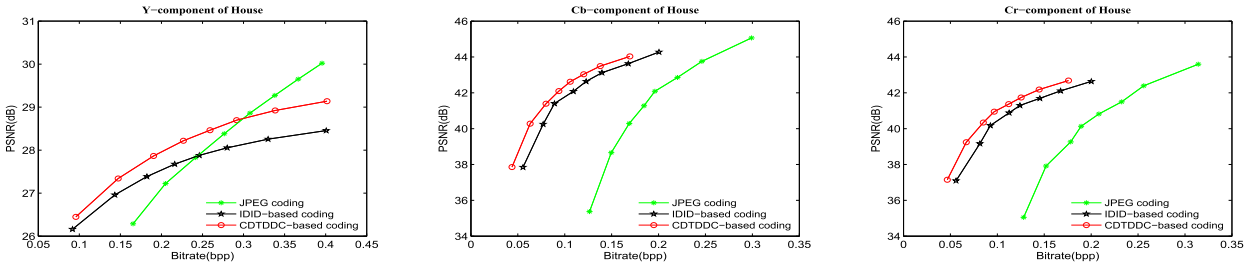



Fig. 14. Verification of CDTDDC-based coding for different components of "House": (a) Y-component, (b) Cb-component, and (c) Cr-component.

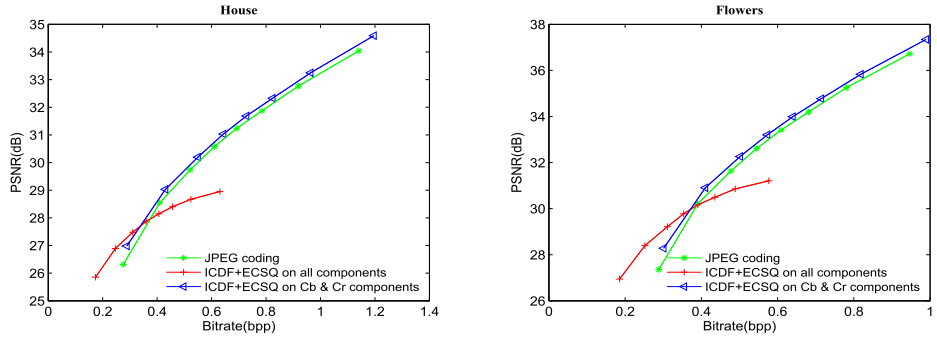



Fig. 15. Comparison of applying ICDF and ECSQ to different components in color image coding.

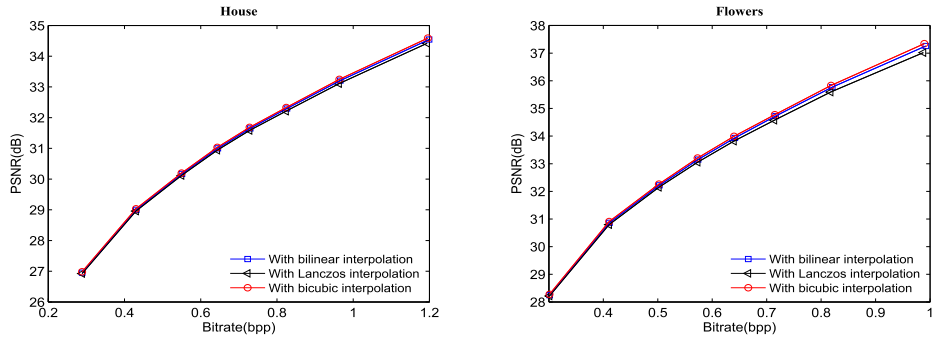



Fig. 16. Comparison of using different interpolations in our proposed method for color image coding.

for test images Airplane, Peppers and Parrot, as proved by up to 1 dB gain over the ACT-based coding. This further verifies the effectiveness of our CDTDDC-based coding scheme.

In addition, some visual results are presented in Figs. 18-20 to make a subjective comparison between various compression methods. Firstly, according to the results shown in Fig. 18, although the JPEG coding, IDID-based coding and CDTDDC-based coding offer similar objective results, the compressed image portion by using the CDTDDC-based method shows less blocking artifacts, especially in the wall of house. On the other hand, the compression performances of the ACT-based method and the ACT-CDTDDC-based method are also very close if evaluated by the PSNR values, but the latter one produces a smoother edge along the roof of house. Secondly, it is seen from Figs. 19-20 that the CDTDDC-based coding offers a better visual quality than the JPEG coding and the IDID-based coding, even than the ACT-based coding, when it is used to compress image Peppers. Moreover, the ACT-CDTDDC-based method provides the best visual quality over all compared coding methods.

### *C. Discussion About Computational Complexity*

In our proposed CDTDDC-based coding, the extra computation mainly results from running the FISTA algorithm to implement the ICDF, and it is important to note that this additional computation only occurs at the encoder side. Meanwhile, the ECSQ algorithm also contributes some additional calculations to implementing the encoding of image signals. Moreover, when our proposed method is applied to the grayscale image coding, the RDO-based mode determination occurring at the encoder side also leads to the increment of computational complexity. On the other hand, the image interpolation is needed at the decoder side for the grayscale image coding, while it does not need any additional operations at the decoder side for the color image coding.

Compared with the traditional JPEG coding, the increased running time spent by using some compared methods is shown in Tables I-II, for grayscale image coding and color image coding, respectively. In both coding scenarios, the quantization quality factors (QFs) are all set to 50. Due to just adopting a new color transform in the traditional JPEG coding, the

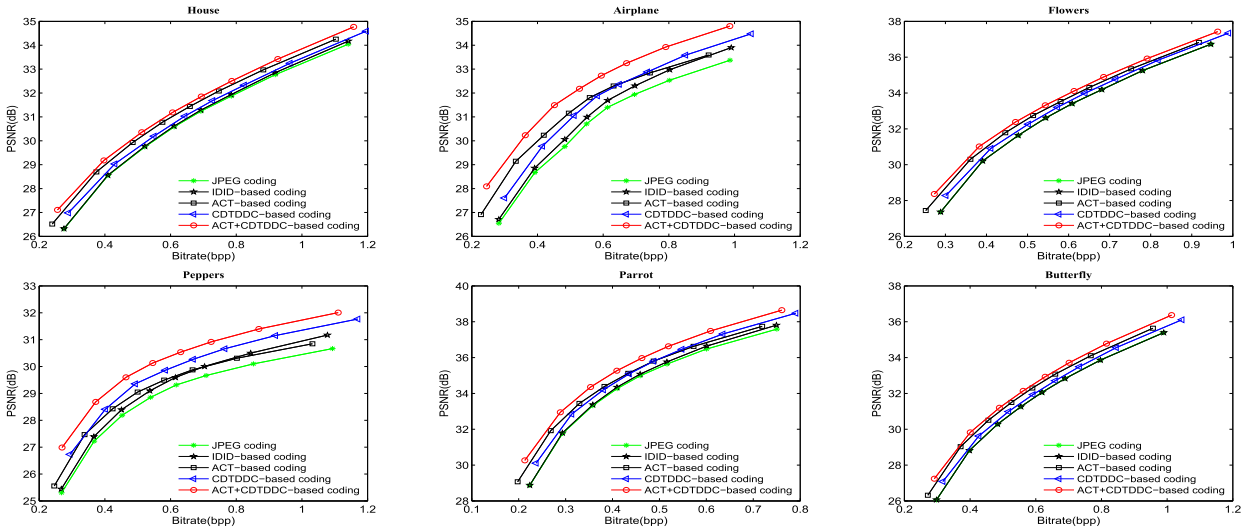



Fig. 17. R-D performances for color image coding by using different methods.

TABLE I
TIME INCREMENT (%) FOR GRAYSCALE IMAGE CODING

| | SRDDS-based coding | ITDDC-based coding | CDTDDC-based coding |
|---|---|---|---|
| Lena | 68.62 | 197.82 | 475.21 |
| Man | 66.41 | 211.23 | 502.07 |
| Goldhill | 65.41 | 199.73 | 490.54 |
| Boat | 67.26 | 202.55 | 491.71 |
| Barbara | 69.59 | 211.82 | 493.76 |
| Baboon | 64.75 | 207.81 | 483.91 |
| **Average** | **67.01** | **205.16** | **489.53** |

TABLE II
TIME INCREMENT (%) FOR COLOR IMAGE CODING

| | IDID-based coding | CDTDDC-based coding |
|---|---|---|
| House | 4.41 | 426.08 |
| Airplane | 4.69 | 420.56 |
| Flowers | 4.43 | 424.02 |
| Peppers | 4.62 | 434.10 |
| Parrot | 4.31 | 421.36 |
| Butterfly | 4.59 | 423.98 |
| **Average** | **4.51** | **425.02** |

ACT-based coding does not cause any additional computations and the evaluation for its complexity is not included in this work. Similarly, the complexity evaluation for the ACT+CDTDDC-based method is not included either, and one can refer the evaluation on the CDTDDC-based coding.

According to the experimental results presented in Tables I-II, it is found that using our proposed method to compress images costs much more time than using the other coding methods. In fact, running the FISTA algorithm at the encoder side contributes near 300% time increment to the grayscale image coding and near 350% time increment to the color image coding. However, using such a solution in our proposed coding will not cause any extra workload at the decoder side.

Based on the above discussions, we believe that our proposed CDTDDC-based coding is useful for the practical application due to its higher compression efficiency, although its encoding procedure is complicated. More specifically, we may use the proposed ICDF and ECSQ algorithms to encode image signals with a powerful server, and just utilize a simple JPEG-based decoder to conduct the decoding for each client. This kind of applications has been very popular in today's cloud computing.

### *D. Further Discussion About the CDTDDC-Based Coding*

It is found that both our proposed grayscale image coding and color image coding have outperformed the state-of-the-art methods. Moreover, applying the CDTDDC-based method to compress color images works more efficiently than applying it to compress grayscale images. In our proposed method, we design two different solutions, one for grayscale image and another for color image, to compress the image signals. These two solutions are developed based on the corresponding characteristics of the grayscale and color images, thus leading to different compression gains.

More specifically, the grayscale image is normally filled with various textures, which makes recovering a full-resolution image from its sub-sampled counterpart quite difficult. Any interpolation method, even coupled with an adaptive down-sampling, still cannot rebuild a high-quality image from a sub-sampled one. Such a limitation also impacts the compression efficiency of our proposed CDTDDC-based coding when used to compress grayscale images. We adopt two coding modes to improve the coding efficiency for this scenario. Although such an operation makes our proposed grayscale image coding outperform the state-of-the-art methods, it still cannot work so efficiently as the proposed CDTDDC-based color image coding does.

Based on the simulation results shown in Fig. 14, it is found using the CDTDDC-based coding to compress chrominance components performs much better than using it to compress luminance component. Compressing the luminance component with CDTDDC only achieves the coding gain at the low bit-rate. Moreover, the R-D results presented in Fig. 15 also demonstrate that applying the CDTDDC-based coding to compress all components cannot guarantee a high overall compression efficiency. Therefore, when such a coding method

(a) (b) (c) (d) (e)

Fig. 18. Image portions of "House" (0.52 bpp): (a) JPEG coding (28.45 dB), (b) IDID-based coding (28.44 dB), (c) CDTDDC-based coding (28.47 dB), (d) ACT-based coding (28.89 dB), and (e) ACT+CDTDDC-based coding (28.94 dB).

(a) (b) (c) (d) (e)

Fig. 19. Image portions of "Airplane" (0.80 bpp): (a) JPEG coding (31.68 dB), (b) IDID-based coding (31.88 dB), (c) CDTDDC-based coding (32.26 dB), (d) ACT-based coding (32.37 dB), and (e) ACT+CDTDDC-based coding (33.05 dB).

(a) (b) (c) (d) (e)

Fig. 20. Image portions of "Peppers" (0.62 bpp): (a) JPEG coding (28.39 dB), (b) IDID-based coding (28.52 dB), (c) CDTDDC-based coding (29.17 dB), (d) ACT-based coding (28.79 dB), and (e) ACT+CDTDDC-based coding (29.46 dB).

is used to compress color images, it is just performed on both chrominance components and the luminance component is still compressed by the JPEG coding. Compressing color images in this way not only guarantees a high coding efficiency, but also makes the CDTDDC-based color image coding fully comparable to the traditional 4:2:0 chroma format coding. Benefited from the excellent compression of chrominance components, our proposed method achieves much more gains over the state-of-the-art methods, which makes it more suitable for practical applications.

## V. Concluding Remarks

In this paper, we have proposed a new TDDC-based coding method to compress image signals. In our way, each $16 \times 16$ macro-block is converted into an $8 \times 8$ small-sized coefficient block by using our proposed ICDF algorithm, which guarantees a better interpolation for the reconstruction of a compressed macro-block as well as a lower compression cost. After that, the small-sized coefficient block is compressed with our proposed ECSQ algorithm to reduce the compression distortion of some specific pixels collected to implement the interpolation for the whole macro-block. Based on the ICDF and ECSQ algorithms, we have proposed a CDTDDC-based coding to compress both grayscale and color images. When our CDTDDC-based coding is used to compress color images, it not only offers a new way to reduce the data-size of the chrominance components but also makes a high compression efficiency. Experimental results show that using our CDTDDC-based coding to compress images has achieved a significant gain over the existing methods, especially for the compression of color images.

## References

[1] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, Apr. 1991.

[2] *Information Technology—JPEG 2000 Image Coding System: Core Coding System*, document ISO/IEC 15444-1 and ITU-T Rec. T.800, 2000.

[3] B. Zeng and J. 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.

[4] S. Muramatsu, D. Han, T. Kobayashi, and H. Kikuchi, "Directional lapped orthogonal transform: Theory and design," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2434–2448, May 2012.

[5] 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.

[6] E. H. Yang, C. Sun, and J. Meng, "Quantization table design revisited for image/video coding," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4799–4811, Nov. 2014.

[7] T. Richter, "JPEG on STEROIDS: Common optimization techniques for JPEG image compression," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 61–65.

[8] G. Lakhani, "Modifying JPEG binary arithmetic codec for exploiting inter/intra-block and DCT coefficient sign redundancies," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1326–1339, Apr. 2013.

[9] D. Liu, X. Y. Sun, F. Wu, S. Li, and Y. Q. Zhang, "Image compression with edge-based inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1273–1287, Oct. 2007.

[10] Z. Xiong, X. Sun, and F. Wu, "Block-based image compression with parameter-assistant inpainting," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1651–1657, Jun. 2010.

[11] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multiresolution graph Fourier transform for compression of piecewise smooth images," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 419–433, Jan. 2015.

[12] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven graph transform for image compression," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 2631–2635.

[13] Y. P. Sun, X. M. Tao, Y. Li, L. Dong, and J. Lu, "HEMS: Hierarchical exemplar-based matching-synthesis for object-aware image reconstruction," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 171–181, Feb. 2016.

[14] J. Xiao, R. Hu, L. Liao, Y. Chen, Z. Wang, and Z. Xiong, "Knowledge-based coding of objects for multisource surveillance video data," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1691–1706, Sep. 2016.

[15] X. F. Zhang, S. Wang, K. Gu, W. Lin, S. Ma, and W. Gao, "Just-noticeable difference-based perceptual optimization for JPEG compression," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 96–100, Jan. 2017.

[16] X. D. Song, X. Peng, J. Xu, G. Shi, and F. Wu, "Distributed compressive sensing for cloud-based wireless image transmission," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1351–1364, Jun. 2017.

[17] B. Zeng and A. N. Venetsanopoulos, "A JPEG-based interpolative image coding scheme," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1993, pp. 393–396.

[18] W. Lin and L. Dong, "Adaptive downsampling to improve image compression at low bit rates," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2513–2521, Sep. 2006.

[19] T. Frajka and K. Zeger, "Downsampling dependent upsampling of images," *Signal Process. Image Commun.*, vol. 19, no. 3, pp. 257–265, Mar. 2004.

[20] X. Wu, X. Zhang, and X. Wang, "Low bit-rate image compression via adaptive down-sampling and constrained least squares upconversion," *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 552–561, Mar. 2009.

[21] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao, "Interpolation-dependent image downsampling," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3291–3296, Nov. 2011.

[22] L. Ma, S. Li, and K. N. Ngan, "Perceptual image compression via adaptive block- based super-resolution directed down-sampling," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 97–100.

[23] Y. B. Zhang, X. Ji, H. Wang, and Q. Dai, "Stereo interleaving video coding with content adaptive image subsampling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1097–1108, Jul. 2013.

[24] Z. Wang and S. Simon, "Low complexity pixel domain perceptual image compression via adaptive down-sampling," in *Proc. Data Compress. Conf.*, Apr. 2016, p. 636.

[25] A. M. Bruckstein, M. Elad, and R. Kimmel, "Down-scaling for better transform compression," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1132–1144, Sep. 2003.

[26] G. Georgis, G. Lentaris, and D. Reisis, "Reduced complexity superresolution for low-bitrate video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 2, pp. 332–345, Feb. 2016.

[27] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 4, pp. 461–474, Apr. 2001.

[28] Z. Wu, H. Yu, and C. Chen, "A new hybrid DCT-Wiener-based interpolation scheme for video intra frame up-sampling," *IEEE Signal Process. Lett.*, vol. 17, no. 10, pp. 827–830, Oct. 2010.

[29] K.-W. Hung and W.-C. Siu, "Hybrid DCT-Wiener-based interpolation via learnt Wiener filter," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2013, pp. 1419–1423.

[30] K.-W. Hung and W.-C. Siu, "Novel DCT-based image up-sampling using learning-based adaptive $k$-NN MMSE estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 12, pp. 2018–2033, Dec. 2014.

[31] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[32] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 887–896, Jun. 2008.

[33] X. M. Liu, D. Zhao, R. Xiong, S. Ma, W. Gao, and H. Sun, "Image interpolation via regularized local linear regression," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3455–3469, Dec. 2011.

[34] C. E. Duchon, "Lanczos filtering in one and two dimensions," *J. Appl. Meteorology*, vol. 18, no. 8, pp. 1016–1022, Aug. 1979.

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

[36] 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.

[37] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[38] T. H. Thai, R. Cogranne, and F. Retraint, "Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1980–1993, May 2014.

[39] J. Yang, G. Zhu, and Y. Shi, "Analyzing the effect of JPEG compression on local variance of image intensity," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2647–2656, Jun. 2016.

[40] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[41] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[42] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[43] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

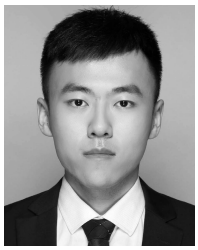
[44] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[45] S. Y. Zhu, J. Yu, M. Li, C. Chen, L. Zeng, and B. Zeng, "Interpolation-directed transform domain downward conversion for block-based image compression," in *Proc. Vis. Commun. Image Process.*, Nov. 2016, pp. 1–4.

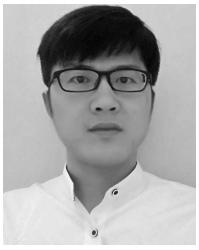
[46] *Parameter Values for High Definition Television Systems for Production and International Programme Exchange*, document ITU-R Rec. BT.709-5, Apr. 2002.

[47] L. Zhang *et al.*, "Adaptive color-space transform in HEVC screen content coding," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 446–459, Dec. 2016.


**Shuyuan Zhu** (S'08–A'09–M'13) received the Ph.D. degree from The Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010. From 2010 to 2012, he was with HKUST and the Hong Kong Applied Science and Technology Research Institute Company Ltd., respectively. In 2013, he joined the University of Electronic Science and Technology of China, where he is currently an Associate Professor with the School of Information and Communication Engineering. His research interests include image/video compression, image processing, and compressive sensing. He is a member of the IEEE CAS Society. He has over 60 research publications and received the Top 10% Paper Award at IEEE ICIP-2014 and the Top 10% Paper Award at VCIP-2016. He was the Special Session Chair of image super-resolution at IEEE DSP-2015. He served as a Committee Member for IEEE ICME-2014, VCIP-2016, and PCM-2017.



**Zhiying He** received the B.Eng. degree in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2015, where he is currently pursuing the M.Eng. degree with the School of Information and Communication Engineering. His research interests include image/video compression.



**Xiandong Meng** (S'16) received the M.Eng. degree in signal and information processing from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. From 2013 to 2014, he was a Research Assistant with the Institute of Image Processing, UESTC, where he was also a Visiting Student from 2016 to 2017. His current research interests include image/video processing, multimedia compression, machine learning, and computer vision. He received the Top 10% Paper Award at VCIP-2016.

**Jiantao Zhou** (M'11) received the B.Eng. degree from the Department of Electronic Engineering, Dalian University of Technology, in 2002, the M.Eng. degree from the Department of Radio Engineering, Southeast University, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, in 2009. He held various research positions at the University of Illinois at Urbana–Champaign, The Hong Kong University of Science and Technology, and McMaster University. He is currently an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. He holds four granted U.S. patents and two granted Chinese patents. His research interests include multimedia security and forensics, and multimedia signal processing. He has co-authored two papers that received the Best Paper Award at the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2016.

**Bing Zeng** (M'91–SM'13–F'16) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1991.

He was a Post-Doctoral Fellow at the University of Toronto from 1991 to 1992 and a Researcher with Concordia University from 1992 to 1993. Then, he joined The Hong Kong University of Science and Technology (HKUST). After 20 years of service at HKUST, he returned to UESTC in 2013, through China's 1000-Talent-Scheme, where he leads the Institute of Image Processing to work on image and video processing, 3D and multi-view video technology, and visual big data.

During his tenure at HKUST and UESTC, he graduated over 30 master's and Ph.D. students, received about 20 research grants, filed eight international patents, and published over 260 papers. Three representing works are as follows: one paper on fast block motion estimation, published in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) in 1994, has so far been SCI-cited over 1000 times (Google-cited over 2100 times) and currently stands at the 7th position among all papers published in this Transactions; one paper on smart padding for arbitrarily-shaped image blocks, published in IEEE TCSVT in 2001, leads to a patent that has been successfully licensed to companies; and one paper on directional discrete cosine transform, published in IEEE TCSVT in 2008, receives the 2011 IEEE CSVT Transactions Best Paper Award. He also received the best paper award at China-Com three times (2009 Xi'an, 2010 Beijing, and 2012 Kunming).

He was elected as an IEEE Fellow in 2016 for contributions to image and video coding. He was a General Co-Chair of IEEE VCIP-2016, Chengdu, China, in 2016. He serves as a General Co-Chair of PCM-2017. He received a second class Natural Science Award (the first recipient) from the Chinese Ministry of Education in 2014. He served as an Associate Editor for IEEE TCSVT for eight years and received the Best Associate Editor Award in 2011. He is currently on the Editorial Board of the *Journal of Visual Communication and Image Representation*.