

High-Quality Color Image Compression by Quantization Crossing Color Spaces

Shuyuan Y. Zhu¹, Member, IEEE, Zhiying Y. He, Chen Chen¹, Member, IEEE, Shuaicheng C. Liu¹, Member, IEEE, Jiantao Zhou¹, Member, IEEE, Yuanfang Guo², Member, IEEE, and Bing Zeng¹, Fellow, IEEE

Abstract—Coding of a color image usually happens in the YCbCr space so that the rate-distortion optimization is conducted in this space. Due to the use of a non-unitary matrix in the RGB-to-YCbCr conversion, an optimal coding performance achieved in the YCbCr space does not guarantee an optimal quality in the RGB space, which would impact most display devices that need RGB signals as the inputs. In this paper, we first study the relationship between the coding distortions of the compressed RGB signals and the quantization errors occurred in the coded YCbCr signals. Then, we design a new quantization scheme crossing the RGB and YCbCr spaces to achieve a high-quality color image compression with the YCbCr 4:4:4 format. Although our proposed quantization takes place in the YCbCr space, it aims at reducing the coding distortion in the RGB space as much as possible. Experimental results demonstrate that our proposed method offers a significant quality gain over the existing block-based coding methods for various images.

Index Terms—Color image compression, RGB space, YCbCr space, error compensation.

I. INTRODUCTION

IN CLASSICAL block-based color image compression, a source RGB image is usually converted to a YCbCr image to remove the redundancies across three color channels. Then, some necessary compression operations, including the discrete cosine transform (DCT), quantization, and entropy coding, are performed on the luminance component (Y) and two chrominance components (Cb and Cr) in the YCbCr space to perform the compression. Since most display devices only

Manuscript received December 19, 2017; revised April 21, 2018; accepted May 23, 2018. Date of publication May 28, 2018; date of current version May 3, 2019. This work was supported in part by the National Key Basic Research Program of China under Grant 2015CB351804, in part by the National Natural Science Foundation of China under Grants 61672134, 61502079, and 61720106004, in part by the “111” Projects under Grant B17008, in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-17-006, and in part by the Macau Science and Technology Development Fund under Grant FDCT/022/2017/A1. This paper was recommended by Associate Editor M. Rabbani. (*Corresponding authors*: Yuanfang Guo; Bing Zeng.)

S. Y. Zhu, Z. Y. He, S. C. Liu, and B. Zeng are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: eezeng@uestc.edu.cn).

C. Chen is with Tencent Company Ltd., Shenzhen 518057, China.

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

Y. Guo is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the Science and Technology on Information Assurance Laboratory, Beijing 100072, China (e-mail: eeandyguo@connect.ust.hk).

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/TCSVT.2018.2841642

accept RGB signals as the inputs, the compressed YCbCr image needs to be converted back to the RGB space. Accordingly, the compression distortion should be measured in the RGB space to evaluate the quality of the compressed image. In practice, minimizing such a distortion requires a design of a high-efficient compression scheme for the RGB signals [1].

To achieve a high coding efficiency, the spatial down-sampling is often applied to the chrominance components before the compression, generating the YCbCr 4:2:0 color format. This format can improve the coding efficiency by saving considerable bit-costs, while an image interpolation is exploited to rebuild each chrominance component with an acceptable quality after the decompression. Although the 4:2:0 format coding offers a higher compression ratio and is adopted in most image and video coding standards [2]–[4], it may cause some extra distortion in the final reconstructed RGB image [5]. Such a distortion is resulted from the down-sampling and interpolation of the chrominance components and it degrades the compressed images, especially at the high bit-rate. To solve this problem, the YCbCr 4:4:4 coding format, i.e., without the down-sampling of chrominance components, is preferred to achieve a high-quality color image compression. This coding format, which maintains the color fidelity, has been adopted in H.265/HEVC range extension (HEVC RExt) for the screen content video coding [6], [7]. Unfortunately, even if the optimal 4:4:4 YCbCr coding is utilized to compress the color image, the quality of the output RGB image will still be degraded by two extra distortions due to the RGB-to-YCbCr conversion.

One of the distortions is caused by the rounding error [8] during the RGB-to-YCbCr conversion, and it has been proved in [9] that this rounding error limits the performance of color image compression. To avoid this error, people proposed to compress RGB images without color conversion. In practice, however, directly coding a RGB image obtains a higher compression efficiency than the conversion-based method only at the high bit-rate [10]. In most cases, the color conversion-based compression still performs better than the RGB space compression. Another attempt to solve the rounding problem is to exploit the reversible color transform (RCT) [11], [12] to implement the color-space conversion where the floating number based conversion is replaced by an integer based one.

The other distortion comes from the RGB-to-YCbCr conversion itself. In the classical image coding, the input image is first divided into image blocks. Then, DCT, scalar

quantization, and entropy coding are performed sequentially on each block. To control the distortion and obtain a higher compression efficiency, a number of advanced techniques have been developed and used in practice in the past few years. In addition to those techniques to reduce the coding artifacts, such as pre/post-filtering [13], deblocking [14], [15], and denosing [16]–[18], there are several techniques focusing on the transform, quantization, and entropy coding steps in the coding procedure. For example, the directional discrete cosine transforms (DDCT) [19], [20] and some novel unitary transforms [21], [22] have been proposed to improve the performance of the transform used in block-based image coding; some advanced quantizers [23]–[25] are designed to reduce the quantization distortion; and a more efficient arithmetic coding [26] is developed to improve the entropy coding of JPEG. In general, all of these advanced techniques are proposed to achieve a higher coding efficiency for luminance and chrominance components in the YCbCr space. However, an optimal YCbCr coding cannot guarantee an optimal performance in the RGB space due to the use of a non-unitary matrix in the RGB-to-YCbCr conversion. A potential solution to this problem is to design a unitary matrix for the color space conversion. Unfortunately, it is very difficult to design such a matrix that also offers a higher compression ratio compared with the traditional RGB-to-YCbCr conversion matrix. An alternative solution is to compress the color image in another space, such as the ICtCp space [27] (which demonstrates promising results when compressing the high dynamic range images), rather than the YCbCr space.

Based on the traditional color image coding, especially the JPEG coding, we propose a novel method to reduce the distortion in the RGB space by controlling the quantization error in the YCbCr space. To achieve this goal, we first perform an analysis on the relationship between the coding distortions of the compressed RGB signal and the quantization errors occurred on the coded YCbCr signal. Then, we develop a sum-of-square-error (SSE) directed quantization (SSEDQ) to reduce the distortion of the compressed RGB signal. The proposed SSEDQ, which can be integrated with other techniques to further improve the coding efficiency, is implemented by performing an error compensation on the transform coefficients in the YCbCr space.

It is important to notice that this work is different from a previous work [28]. Firstly, the YCbCr 4:2:0 format is considered in [28], whereas we focus on the YCbCr 4:4:4 format in this work so as to achieve a high-quality compression. Secondly, the error compensation algorithm proposed in [28] is only applied to the Y component - the interpolation and compression errors of two chrominance components are compensated to the Y component. In this work, we design the error compensation algorithm in an iterative way so that the Cb distortion is firstly compensated to the Cr component and then the distortions of both chrominance components are jointly compensated to the Y component. Thirdly, we propose an advanced quantization strategy based on the adaptive down-sampling of the Cr component in this work so that we may achieve a higher coding efficiency as well as a higher color fidelity for the 4:4:4 format coding.

The rest of this paper is organized as follows. A review about the compression distortion in color image coding is presented in Section II. The SSEDQ algorithm is proposed in Section III. The experimental results are provided in Section IV and some conclusions are finally drawn in Section V.

II. COMPRESSION DISTORTION IN COLOR IMAGE COMPRESSION

A. Compression Distortion in the YCbCr Space

In the YCbCr space, the quantization, which is performed on the DCT coefficients, causes the degradation on the compressed image. To minimize the degradation, the Lloyd-Max quantizer [29] is theoretically considered as the optimal quantizer according to the minimum mean-square-error principle. However, it is seldom adopted in practice because of its signal-dependent nature.

In a practical lossy image coding, the uniform scalar quantization is often employed to quantize the transform coefficients. It has been demonstrated in [30] that such a quantizer offers an excellent performance as compared to the optimal quantizer while possessing a much lower complexity. For example, in the classical JPEG coding [2], the rounding-based uniform quantization is used to quantize each individual transform coefficient as

$$\hat{X}_k = \text{round}(X_k/S_k) \times S_k \quad (1)$$

where X_k is the transform coefficient, \hat{X}_k denotes the quantized coefficient, S_k represents the quantization step-size that is determined according to the quantization table (defined in JPEG) and $\text{round}(\cdot)$ stands for rounding the corresponding value to its nearest integer.

The uniform quantizer adopted in the YCbCr image coding leads to an optimal quantization as well as an optimal compression in the YCbCr space. However, when the quantization errors are transferred from the YCbCr space to the RGB space, it becomes more complicated to maintain the optimal performance. Therefore, it is necessary to study the relationship between the reconstruction error of the RGB signal and the quantization error of the YCbCr signal.

B. Compression Distortion in the RGB Space

First, the RGB input signal needs to be converted into the YCbCr signal to remove the redundancies across three color channels. According to ITU Rec. BT. 709 [31], the color conversion from the RGB space to the YCbCr space is represented as

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1146 & -0.3854 & 0.5000 \\ 0.5000 & -0.4542 & -0.0458 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \Lambda \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2)$$

After that, some necessary coding operations, such as DCT, quantization, and entropy coding, are performed on the full-size luminance and chrominance components separately. Here, the compression efficiency is evaluated by the bit-rate and PSNR in the YCbCr space. After the compression, the compressed YCbCr signal needs to be converted back to the RGB

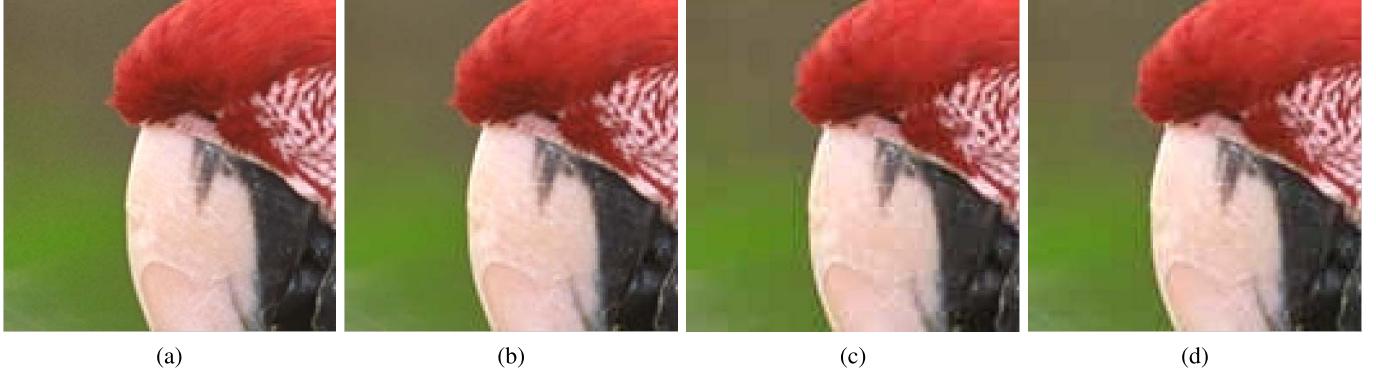


Fig. 1. RGB-YCbCr cross-space distortion for compressed image portion of *Parrots*. (a) Original image portion. (b) Full-resolution RGB coding (2.15 bpp) with RGB-space PSNR = 37.11 dB. (c) 4:4:4 YCbCr coding (0.78 bpp) with YCbCr-space PSNR = 37.11 dB and RGB-space PSNR = 32.43 dB. (d) Our proposed ASSEDQ-based method (0.78 bpp) with YCbCr-space PSNR = 37.85 dB and RGB-space PSNR = 33.62 dB. Please zoom the image portions to perceive the differences more clearly.

space for display. Therefore, the evaluation of the compression distortion in the RGB space, which is directly related to the human visual system (HVS), is very important [32], [33].

For a better illustration of the pros and cons of different coding approaches, three compressed image portions of the test image *Parrots*, produced by different coding methods, are shown in Fig. 1. Specifically, Fig. 1(b) shows the compression result obtained by running the 4:4:4 RGB format coding and the distortion is also measured in the RGB space. This coding approach maintains the color fidelity well as supported by the best perceptual quality, but it is seldom used in practice due to its high bit-cost. On the other hand, Fig. 1(c) presents the result obtained by carrying out the 4:4:4 YCbCr coding, where the distortion is measured in both YCbCr and RGB spaces. After the compression, if the compression distortion for the result in Fig. 1(c) is measured in the YCbCr space, the same objective score will be obtained as Fig. 1(b) does in the RGB space. However, Fig. 1(b) offers a more pleasant perceptual quality while obvious block-artifacts appear in Fig. 1(c). Meanwhile, if we measure the compression distortion for the result in Fig. 1(c) in the RGB space, it is found that the PSNR drops significantly after the YCbCr-to-RGB conversion. Finally, Fig. 1(d) presents the result obtained by using our proposed ASSEDQ (advanced SSE-DIRECTED QUANTIZATION) based 4:4:4 YCbCr coding, where the distortion is measured in both YCbCr space and RGB space. By reducing the distortion in the RGB space instead of the YCbCr space, we apply our proposed ASSEDQ in the traditional 4:4:4 YCbCr coding and obtain an obvious quality gain, both objectively and subjectively.

According to this example, we may conclude that measuring the compression distortion in the YCbCr space is not sufficient enough to guide the design of a high-efficient color image compression. On the other hand, although the direct compression in the RGB space achieves a lower color distortion, the resulted bit-rate is rather high. Based on these observations, we believe that both high quality and high efficiency for the color image compression could be achieved by controlling the RGB-space distortion when performing the YCbCr-space coding.

TABLE I
MAIN NOTATIONS AND CORRESPONDING EXPLANATIONS

	Main notation	Explanation
YCbCr space	$\Delta \mathbf{x}_Y, \Delta \mathbf{x}_{Cb}, \Delta \mathbf{x}_{Cr}$	N -point coding errors for Y, Cb and Cr pixels
	$\Delta \mathbf{X}_Y, \Delta \mathbf{X}_{Cb}, \Delta \mathbf{X}_{Cr}$	N -point quantization errors for Y, Cb and Cr coefficients
	$\Delta x_{Y_k}, \Delta x_{Cb_k}, \Delta x_{Cr_k}$	Coding errors for individual Y, Cb and Cr pixels
	$\Delta X_{Y_k}, \Delta X_{Cb_k}, \Delta X_{Cr_k}$	Quantization errors for individual Y, Cb and Cr coefficients
	$X_{Y_k}, X_{Cb_k}, X_{Cr_k}$	Y, Cb and Cr coefficients
	$\delta_{Y_k}, \delta_{Cb_k}$	Compensation terms for Y and Cb coefficients
	$\tilde{X}_{Y_k}, \tilde{X}_{Cb_k}$	Renewed Y and Cb coefficients
	$\Delta \tilde{X}_{Y_k}, \Delta \tilde{X}_{Cb_k}$	Renewed quantization errors for Y and Cb coefficients
	$E_{YCbCr}^{(k)}$	SSE distortion for each group of $\{X_{Y_k}, X_{Cb_k}, X_{Cr_k}\}$
RGB space	$\tilde{E}_{YCbCr}^{(k)}$	SSE distortion for each group of $\{\tilde{X}_{Y_k}, \tilde{X}_{Cb_k}, \tilde{X}_{Cr_k}\}$
	$\Delta \mathbf{x}_R, \Delta \mathbf{x}_G, \Delta \mathbf{x}_B$	N -point coding errors for R, G and B pixels
	$\Delta x_{R_k}, \Delta x_{G_k}, \Delta x_{B_k}$	Coding errors for individual R, G and B pixels
	E_{RGB}	SSE distortion in the RGB space
	\tilde{E}_{RGB}	Updated SSE distortion in the RGB space

III. PROPOSED SSE-DIRECTED QUANTIZATION

In this section, we first analyze the relationship between the RGB-space distortion and the YCrCr-space distortion. Then, based on our analysis, we design a new quantization method for color image coding. To facilitate the analysis, we list all notations used in our paper in Table I.

A. Cross-Space Distortion

For simplicity, only the 1-D case is considered in our analysis, whereas a 2-D block can be mapped into the 1-D case by concatenating all columns. Let \mathbf{x}_Y be a vector composed by N luminance (Y) pixels. Then, the transform is performed

on \mathbf{x}_Y to get the transform coefficient vector $\mathbf{X}_Y = \mathbf{C}\mathbf{x}_Y$. After that, the quantization is performed on all elements of \mathbf{X}_Y to obtain the quantized coefficient vector $\hat{\mathbf{X}}_Y$. Finally, the inverse transform is applied to reconstruct all luminance pixels $\hat{\mathbf{x}}_Y$. With $\Delta\mathbf{x}_Y = \mathbf{x}_Y - \hat{\mathbf{x}}_Y$ and $\Delta\mathbf{X}_Y = \mathbf{X}_Y - \hat{\mathbf{X}}_Y$ being the coding errors in the pixel domain and the quantization errors in the transform domain, respectively, we can obtain $\Delta\mathbf{x}_Y = \mathbf{C}^{-1}\Delta\mathbf{X}_Y$.

Similarly, for two chrominance components \mathbf{x}_{Cb} and \mathbf{x}_{Cr} , the relationship between the coding errors in the pixel domain and the quantization errors in the transform domain can be formulated as $\Delta\mathbf{x}_{Cb} = \mathbf{C}^{-1}\Delta\mathbf{X}_{Cb}$ and $\Delta\mathbf{x}_{Cr} = \mathbf{C}^{-1}\Delta\mathbf{X}_{Cr}$, respectively. Then, we use $\Delta\mathbf{x}_Y$, $\Delta\mathbf{x}_{Cb}$, and $\Delta\mathbf{x}_{Cr}$ to compose an error vector in the pixel domain as $\Delta\mathbf{x}_{YCbCr} = [\Delta\mathbf{x}_Y^T \quad \Delta\mathbf{x}_{Cb}^T \quad \Delta\mathbf{x}_{Cr}^T]^T$. In the same way, we use $\Delta\mathbf{X}_Y$, $\Delta\mathbf{X}_{Cb}$, and $\Delta\mathbf{X}_{Cr}$ to compose an error vector in the transform domain as $\Delta\mathbf{X}_{YCbCr} = [\Delta\mathbf{X}_Y^T \quad \Delta\mathbf{X}_{Cb}^T \quad \Delta\mathbf{X}_{Cr}^T]^T$. Meanwhile, we compose an inverse transform matrix \mathbf{H} based on \mathbf{C}^{-1} as

$$\mathbf{H} = \begin{bmatrix} \mathbf{C}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}^{-1} \end{bmatrix}. \quad (3)$$

As a result, the coding error of all luminance and chrominance pixels is calculated as

$$\Delta\mathbf{x}_{YCbCr} = \mathbf{H}\Delta\mathbf{X}_{YCbCr}. \quad (4)$$

After the compression, the compressed YCbCr signal is converted back to the RGB space. Let Δx_{R_k} , Δx_{G_k} and Δx_{B_k} denote the distortion on the red (R), green (G), and blue (B) components of each color pixel, respectively. Meanwhile, we use Δx_{Y_k} , Δx_{Cb_k} and Δx_{Cr_k} to represent the distortions on the Y, Cb, and Cr components of each pixel, respectively. Based on the inverse conversion of (2), we can obtain

$$\begin{bmatrix} \Delta x_{R_k} \\ \Delta x_{G_k} \\ \Delta x_{B_k} \end{bmatrix} = \Lambda^{-1} \begin{bmatrix} \Delta x_{Y_k} \\ \Delta x_{Cb_k} \\ \Delta x_{Cr_k} \end{bmatrix}. \quad (5)$$

Next, we compose N -point distortion errors as $\Delta\mathbf{x}_R = [\Delta x_{R_0}, \dots, \Delta x_{R_{N-1}}]^T$, $\Delta\mathbf{x}_G = [\Delta x_{G_0}, \dots, \Delta x_{G_{N-1}}]^T$, and $\Delta\mathbf{x}_B = [\Delta x_{B_0}, \dots, \Delta x_{B_{N-1}}]^T$. The overall coding error in the RGB space can be represented as $\Delta\mathbf{x}_{RGB} = [\Delta\mathbf{x}_R^T \quad \Delta\mathbf{x}_G^T \quad \Delta\mathbf{x}_B^T]^T$. Based on (5), the elements in Λ^{-1} can be used to form a $3N \times 3N$ matrix \mathbf{A} which makes

$$\Delta\mathbf{x}_{RGB} = \mathbf{A}\Delta\mathbf{x}_{YCbCr}. \quad (6)$$

Moreover, \mathbf{A} can be represented as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{00} & \mathbf{A}_{01} & \mathbf{A}_{02} \\ \mathbf{A}_{10} & \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{20} & \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \quad (7)$$

where each $\mathbf{A}_{m,n}$ ($m, n = 0, 1, 2$) is an $N \times N$ diagonal matrix composed as

$$\mathbf{A}_{m,n} = \begin{bmatrix} a_{m,n} & 0 & 0 & 0 \\ 0 & a_{m,n} & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & a_{m,n} \end{bmatrix} \quad (8)$$

and its element $a_{m,n}$ is determined according to Λ^{-1} .

With (6) and (7), we can calculate the SSE distortion of all pixels in the RGB space (denoted as E_{RGB}) as

$$\begin{aligned} E_{RGB} &= \Delta\mathbf{x}_{RGB}^T \Delta\mathbf{x}_{RGB} \\ &= \Delta\mathbf{x}_{YCbCr}^T (\mathbf{A}^T \mathbf{A}) \Delta\mathbf{x}_{YCbCr}. \end{aligned} \quad (9)$$

By substituting (4) into (9), we obtain

$$E_{RGB} = \Delta\mathbf{X}_{YCbCr}^T (\mathbf{AH})^T (\mathbf{AH}) \Delta\mathbf{X}_{YCbCr}. \quad (10)$$

By defining $\mathbf{W} = (\mathbf{AH})^T (\mathbf{AH})$, we can rewrite (10) as

$$E_{RGB} = \Delta\mathbf{X}_{YCbCr}^T \mathbf{W} \Delta\mathbf{X}_{YCbCr}. \quad (11)$$

In particular, \mathbf{W} is a $3N \times 3N$ symmetric matrix and can be represented as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{00} & \mathbf{W}_{01} & \mathbf{W}_{02} \\ \mathbf{W}_{10} & \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{20} & \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix} \quad (12)$$

where each $\mathbf{W}_{m,n}$ ($m, n = 0, 1, 2$) is an $N \times N$ diagonal matrix.

The composition of \mathbf{W} makes it not an identity matrix. Therefore, based on (11), it is found that the SSE distortion of the RGB pixels and the quantization errors of the luminance and chrominance transform coefficients are related via a rather complicated weighting process. Therefore, even if an optimal compression is achieved in the YCbCr space, it still cannot guarantee an optimal performance in the RGB space.

Let ΔX_{Y_k} , ΔX_{Cb_k} and ΔX_{Cr_k} denote the elements of $\Delta\mathbf{X}_Y$, $\Delta\mathbf{X}_{Cb}$ and $\Delta\mathbf{X}_{Cr}$, respectively. Based on (12), we can rewrite (11) as

$$\begin{aligned} E_{RGB} &= \sum_{k=0}^{N-1} \left(\alpha_k^{(0)} (\Delta X_{Y_k})^2 + \alpha_k^{(1)} (\Delta X_{Cb_k})^2 \right. \\ &\quad \left. + \alpha_k^{(2)} (\Delta X_{Cr_k})^2 \right) + \sum_{k=0}^{N-1} \left(\beta_k^{(0)} \Delta X_{Y_k} \Delta X_{Cb_k} \right. \\ &\quad \left. + \beta_k^{(1)} \Delta X_{Y_k} \Delta X_{Cr_k} + \beta_k^{(2)} \Delta X_{Cb_k} \Delta X_{Cr_k} \right) \end{aligned} \quad (13)$$

where $\alpha_k^{(j)}$ and $\beta_k^{(j)}$ ($j = 0, 1, 2$) are determined by the corresponding elements of \mathbf{W} ($\alpha_k^{(j)} > 0$ and $\beta_k^{(j)} > 0$ for all k and j).

In the traditional YCbCr coding, the quantization errors in the transform domain and the coding errors in the pixel domain have the same energy, thanks to using the unitary transform matrix, i.e., DCT, in compression. However, in the coding scenario considered here, (13) reveals two different results: (a) the first sum in (13) implies that three individual quantization errors will be weighted by some coefficients that may not be equal to 1 and (b) the second sum in (13) implies that there exist three cross- and weighted-terms. In general, quantization errors can be positive or negative in a random fashion. When two quantization errors have different signs, the resulted cross-term would contribute a reduction to the coding distortion.

Let X_k represent a single transform coefficient in the YCbCr space and $X_k \in \{X_{Y_k}, X_{Cb_k}, X_{Cr_k}\}$, where X_{Y_k} , X_{Cb_k} and X_{Cr_k} denote the Y, Cb and Cr transform coefficients,

respectively. For each X_k , we may simply use both flooring-based and ceiling-based quantizers to quantize it as

$$\hat{X}_k = \begin{cases} Q_{S_k}^{(floor)}(X_k) = \text{floor}(X_k/S_k) \times S_k \\ Q_{S_k}^{(ceil)}(X_k) = \text{ceil}(X_k/S_k) \times S_k \end{cases} \quad (14)$$

With (14), there exist two corresponding quantization results for each X_k and it brings 2^{3N} combinations to quantize $[X_Y^T \ X_{Cb}^T \ X_{Cr}^T]^T$. Different combinations will lead to different RGB distortions and the minimal one can be found by trying an exhausted-search on all 2^{3N} possible combinations. Obviously, this search strategy becomes impossible in practical applications, and therefore it is necessary to find a more efficient solution to reduce RGB distortion.

B. SSE-Directed Quantization

Based on (13), we propose to jointly control ΔX_{Y_k} , ΔX_{Cb_k} , and ΔX_{Cr_k} , as well as $\Delta X_{Y_k} \Delta X_{Cb_k}$, $\Delta X_{Y_k} \Delta X_{Cr_k}$, and $\Delta X_{Cb_k} \Delta X_{Cr_k}$, so as to reduce E_{RGB} as much as possible.

According to the Cholesky decomposition [34], the symmetric matrix \mathbf{W} can be factorized as $\mathbf{W} = \mathbf{D}^T \mathbf{D}$, where \mathbf{D} is a $3N \times 3N$ upper triangular matrix

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^0 & \mathbf{D}^{(1)} & \mathbf{D}^{(2)} \\ \mathbf{0} & \mathbf{D}^{(3)} & \mathbf{D}^{(4)} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}^{(5)} \end{bmatrix} \quad (15)$$

and each $\mathbf{D}^{(n)}$ is an $N \times N$ diagonal matrix

$$\mathbf{D}^{(n)} = \begin{bmatrix} d_{0,0}^{(n)} & & & \\ & \ddots & & \\ & & d_{N-1,N-1}^{(n)} & \end{bmatrix}. \quad (16)$$

Then, (11) can be rewritten as

$$E_{RGB} = \Delta \mathbf{X}_{YCbCr}^T (\mathbf{D}^T \mathbf{D}) \Delta \mathbf{X}_{YCbCr}. \quad (17)$$

If $E_{YCbCr}^{(k)}$ is used to represent the SSE distortion of each independent $\{X_{Y_k}, X_{Cb_k}, X_{Cr_k}\}$, we can obtain

$$E_{RGB} = \sum_{k=0}^{N-1} E_{YCbCr}^{(k)} \quad (18)$$

where

$$\begin{aligned} E_{YCbCr}^{(k)} &= (d_{k,k}^{(0)} \Delta X_{Y_k} + d_{k,k}^{(1)} \Delta X_{Cb_k} + d_{k,k}^{(2)} \Delta X_{Cr_k})^2 \\ &\quad + (d_{k,k}^{(3)} \Delta X_{Cb_k} + d_{k,k}^{(4)} \Delta X_{Cr_k})^2 \\ &\quad + (d_{k,k}^{(5)} \Delta X_{Cr_k})^2. \end{aligned} \quad (19)$$

It is seen from (18) that the overall SSE distortion in the RGB space is determined by the linear combination of $E_{YCbCr}^{(k)}$. Moreover, the minimum E_{RGB} can be achieved by separately minimizing each $E_{YCbCr}^{(k)}$. However, (19) cannot be minimized by simply taking partial derivatives on each ΔX_{Y_k} , ΔX_{Cb_k} , and ΔX_{Cr_k} .

Based on (19), we can rewrite the second item as

$$\begin{aligned} &(d_{k,k}^{(3)} \Delta X_{Cb_k} + d_{k,k}^{(4)} \Delta X_{Cr_k})^2 \\ &= (d_{k,k}^{(3)})^2 (\Delta X_{Cb_k} + (d_{k,k}^{(4)} / d_{k,k}^{(3)}) \Delta X_{Cr_k})^2 \\ &= (d_{k,k}^{(3)})^2 (\Delta X_{Cb_k} + \delta_{Cb_k})^2 \\ &= (d_{k,k}^{(3)})^2 (X_{Cb_k} - Q(X_{Cb_k}) + \delta_{Cb_k})^2 \\ &= (d_{k,k}^{(3)})^2 ((X_{Cb_k} + \delta_{Cb_k}) - Q(X_{Cb_k}))^2 \end{aligned} \quad (20)$$

where $Q(\cdot)$ stands for the quantization carried out on the transform coefficient and $\delta_{Cb_k} = (d_{k,k}^{(4)} / d_{k,k}^{(3)}) \Delta X_{Cr_k}$. Thus, no matter how X_{Cr_k} is quantized, the corresponding quantization error produced on it will contribute a compensation term δ_{Cb_k} to X_{Cb_k} . As long as this compensation term is added into X_{Cb_k} , the quantization on $\tilde{X}_{Cb_k} = X_{Cb_k} + \delta_{Cb_k}$ will guarantee to yield an item of $(\tilde{X}_{Cb_k} - Q(\tilde{X}_{Cb_k}))^2 = (\Delta \tilde{X}_{Cb_k})^2$, which is smaller (statistically) than $(\Delta X_{Cb_k} + \delta_{Cb_k})^2$.

After the above compensation, the first item in (19) may be rewritten as

$$\begin{aligned} &(d_{k,k}^{(0)} \Delta X_{Y_k} + d_{k,k}^{(1)} \Delta \tilde{X}_{Cb_k} + d_{k,k}^{(2)} \Delta X_{Cr_k})^2 \\ &= (d_{k,k}^{(0)})^2 (\Delta X_{Y_k} + (1/d_{k,k}^{(0)}) (d_{k,k}^{(1)} \Delta \tilde{X}_{Cb_k} + d_{k,k}^{(2)} \Delta X_{Cr_k}))^2 \\ &= (d_{k,k}^{(0)})^2 (\Delta X_{Y_k} + \delta_{Y_k})^2 \\ &= (d_{k,k}^{(0)})^2 ((X_{Y_k} - Q(X_{Y_k})) + \delta_{Y_k})^2 \\ &= (d_{k,k}^{(0)})^2 ((X_{Y_k} + \delta_{Y_k}) - Q(X_{Y_k}))^2 \end{aligned} \quad (21)$$

where the compensation term for X_{Y_k} is calculated as $\delta_{Y_k} = (d_{k,k}^{(1)} \Delta \tilde{X}_{Cb_k} + d_{k,k}^{(2)} \Delta X_{Cr_k}) / d_{k,k}^{(0)}$. Then, the normal quantization performed on $X_{Y_k} = X_{Y_k} + \delta_{Y_k}$ will yield an item of $(\tilde{X}_{Y_k} - Q(\tilde{X}_{Y_k}))^2 = (\Delta \tilde{X}_{Y_k})^2$, which is also smaller (statistically) than $(\Delta X_{Y_k} + \delta_{Y_k})^2$. Finally, the new SSE distortion is calculated as

$$\tilde{E}_{YCbCr}^{(k)} = (d_{k,k}^{(0)} \Delta \tilde{X}_{Y_k})^2 + (d_{k,k}^{(3)} \Delta \tilde{X}_{Cb_k})^2 + (d_{k,k}^{(5)} \Delta X_{Cr_k})^2. \quad (22)$$

After the error compensation performed on each $\{X_{Y_k}, X_{Cb_k}\}$, a smaller SSE distortion \tilde{E}_{RGB} can be achieved by combining all $\tilde{E}_{YCbCr}^{(k)}$ together as

$$\begin{aligned} \tilde{E}_{RGB} &= \sum_{k=0}^{N-1} ((d_{k,k}^{(0)})^2 (\Delta \tilde{X}_{Y_k})^2 + (d_{k,k}^{(3)})^2 (\Delta \tilde{X}_{Cb_k})^2 \\ &\quad + (d_{k,k}^{(5)})^2 (\Delta X_{Cr_k})^2). \end{aligned} \quad (23)$$

Obviously, (23) builds up a simple relationship between the coding distortion of the compressed RGB signal and the quantization error occurred in the compressed YCbCr signal. Specifically, the distortion in the RGB space is controlled by the weighted quantization errors in the YCbCr space. As a result, a further reduction on the RGB distortion can be achieved with an advanced design of $d_{k,k}^{(0)}$, $d_{k,k}^{(3)}$, and $d_{k,k}^{(5)}$, which can also be used to compose a more efficient color conversion matrix. We will focus on this in our future work.

The proposed SSEDQ is summarized in Algorithm 1. In 2-D case, each $N \times N$ YCbCr image block is first converted into a column-vector via concatenating all columns and the 2-D

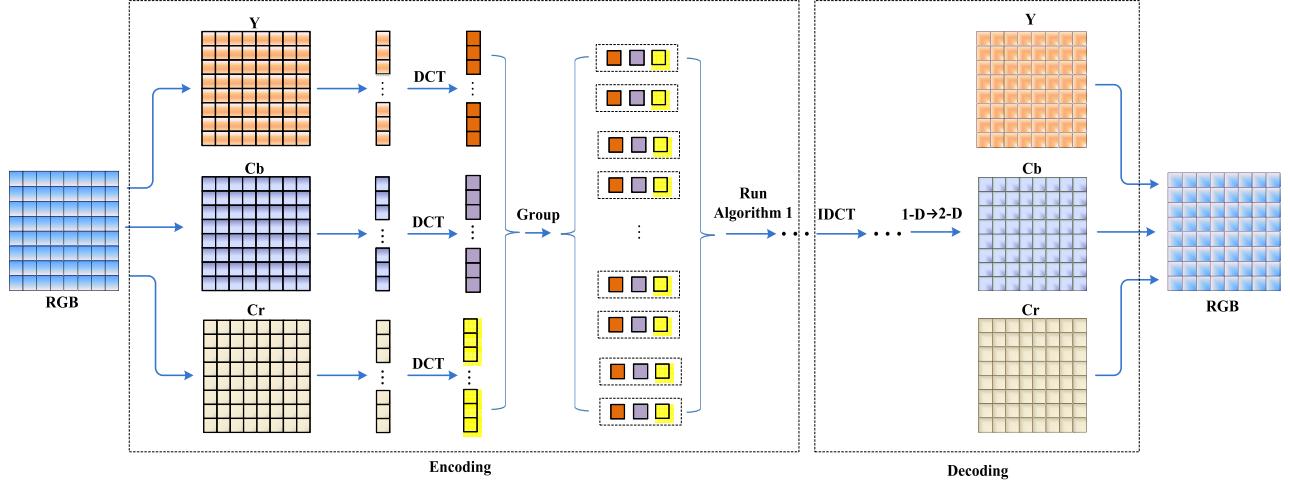


Fig. 2. Framework of the proposed SSEDQ-based color image compression.

Algorithm 1 SSE-Directed Quantization

Loop $k = 0 : N-1$

Step-1: Calculate the quantization error for X_{Cr_k} :
 $\Delta X_{Cr_k} = X_{Cr_k} - Q(X_{Cr_k})$;
Calculate the compensation term for X_{Cb_k} :
 $\delta_{Cb_k} = (d_{k,k}^{(4)} / d_{k,k}^{(3)}) \Delta X_{Cr_k}$;
Step-2: Renew X_{Cb_k} : $\tilde{X}_{Cb_k} = X_{Cb_k} + \delta_{Cb_k}$;
Quantize \tilde{X}_{Cb_k} : $Q(\tilde{X}_{Cb_k})$;
Calculate the quantization error for X_{Cb_k} :
 $\Delta \tilde{X}_{Cb_k} = X_{Cb_k} - Q(\tilde{X}_{Cb_k})$;
Calculate the compensation term for Y_k :
 $\delta_{Y_k} = (d_{k,k}^{(1)} \Delta \tilde{X}_{Cb_k} + d_{k,k}^{(2)} \Delta X_{Cr_k}) / d_{k,k}^{(0)}$;
Step-3: Renew X_{Y_k} : $\tilde{X}_{Y_k} = X_{Y_k} + \delta_{Y_k}$;
Quantize \tilde{X}_{Y_k} : $Q(\tilde{X}_{Y_k})$.

End Loop

separable transform is converted into a 1-D transform via the Kronecker-product. Then, the proposed SSEDQ algorithm can be carried out. The framework of our SSEDQ-based color image compression is shown in Fig. 2 and an example of processing a 4×4 image block taken from the test image Lena is presented in Fig. 3 to show how SSEDQ works, where the quantization is performed as $Q(X_k) = \text{round}(X_k/10) \times 10$. Note that all extra computations to implement SSEDQ are required only at the encoder side, whereas the decoder does not need any extra computations.

C. Advanced SSE-Directed Quantization

In the above error compensation procedure, two compensation terms, i.e., δ_{Cb_k} and δ_{Y_k} , are added to X_{Cb_k} and X_{Y_k} , respectively. Once the compensation happens, a big compensation term has two-fold impact: 1) a smaller SSE distortion could be achieved on one hand and 2) a bigger bit-count would be yielded on the other hand. To effectively control the increment of the bit-count in the SSEDQ-based coding, the Cr component is sub-sampled before the compression and the image interpolation is used to reconstruct the

complete Cr component when the decompression is performed. Meanwhile, the resolutions of both Y and Cb components are kept unchanged. With the down-sampling and interpolation performed on the Cr component, we develop an advanced SSEDQ-based coding method to compress color images.

According to Algorithm 1, we calculate δ_{Cb_k} based on the quantization error of X_{Cr_k} . In fact, any distortion occurred on X_{Cr_k} can be used to calculate δ_{Cb_k} . Let \mathbf{B}_{Cr} represent the macro-block of the Cr component and \mathbf{b}_{Cr} denote the sub-sampled block for \mathbf{B}_{Cr} . After the compression on \mathbf{b}_{Cr} , we can obtain the compressed block $\hat{\mathbf{b}}_{Cr}$ and interpolate it to compose a full-size macro-block, denoted as $\hat{\mathbf{B}}_{Cr}$. In this work, the Cr component is sub-sampled along both the horizontal and vertical directions by a factor of 2 and the bicubic interpolation is used to reconstruct the complete Cr component.

After the above down-sampling based compression on the Cr component, it is found that both the interpolation error and the quantization error occurred on \mathbf{B}_{Cr} should be included in our proposed error compensation algorithm to calculate δ_{Cb_k} . To this end, $\hat{\mathbf{B}}_{Cr}$ will be firstly divided into several sub-blocks, and then the 2-D DCT is performed on each sub-block to obtain the degraded Cr coefficients. After that, each degraded Cr coefficient is used to replace the corresponding $Q(X_{Cr_k})$ in Algorithm 1, to calculate the compensation term for both the Y and Cb components.

Due to the use of image interpolation in the compression of Cr component, the overall compression efficiency for the RGB image will be limited, especially at the high bit-rate. To solve this problem, the above modified SSEDQ-based coding will collaborate with the previous one as two candidate coding modes for each RGB macro-block, which leads to an advanced SSEDQ (ASSEDQ) based coding. In this coding scheme, a simple but efficient rate distortion optimization (RDO) procedure is used to determine which coding mode is selected. Specifically, these two coding modes are competing with each other and the winner will be selected to compress the current macro-block according to the product of mean-square-error (MSE) and the bit-count of it. Besides, the ASSEDQ-based coding needs 1 overhead bit to represent the mode

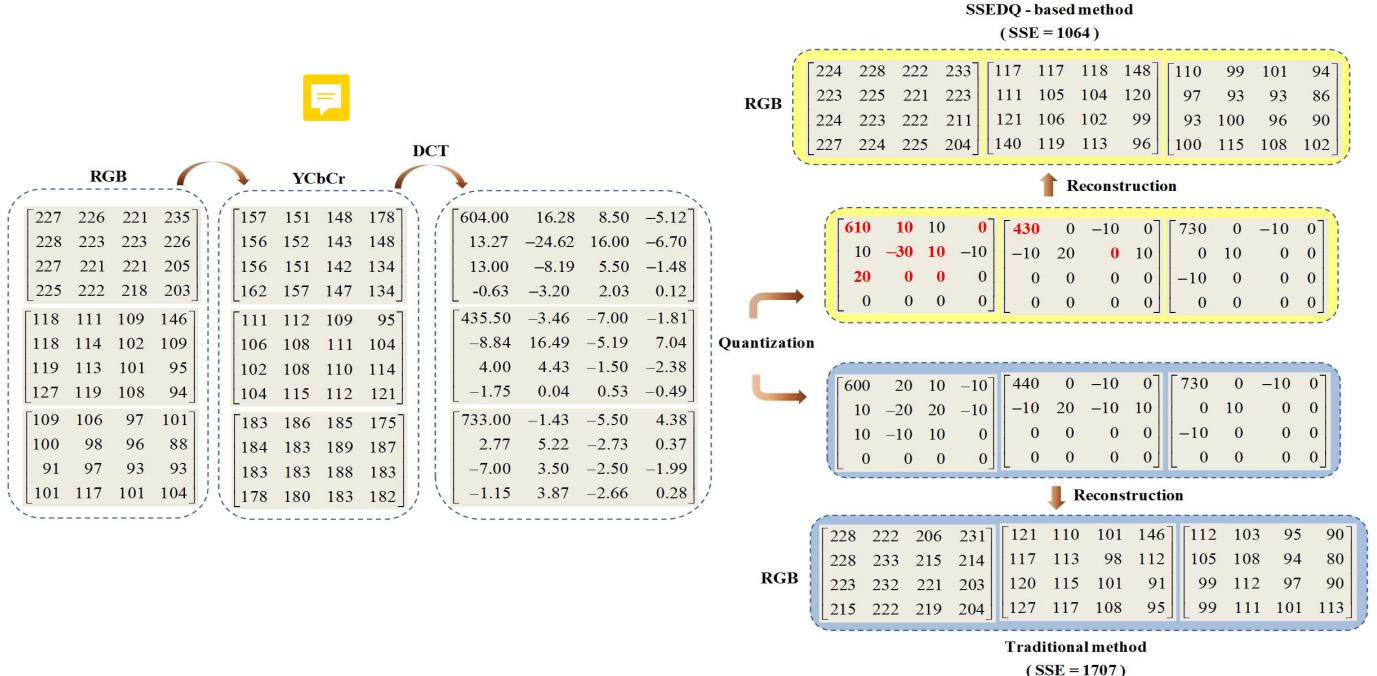


Fig. 3. Example of a 4×4 image block picked up from the test image Lena.

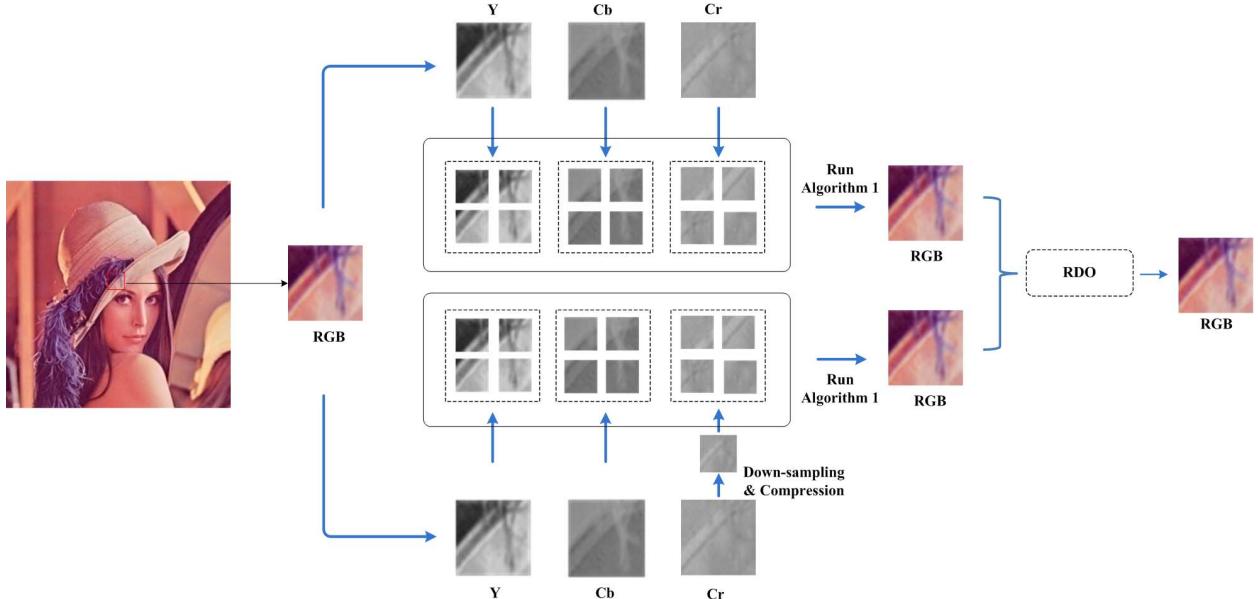


Fig. 4. Advanced SSEDQ-based color image compression.

information for each macro-block. When the decoding is carried out, the interpolation is required to reconstruct the Cr macro-block. The framework of our proposed ASSEDQ-based color image compression is shown in Fig. 4.

IV. EXPERIMENTAL RESULTS

We verify the effectiveness of our proposed SSE-directed quantization by integrating it into the JPEG coding (the most popular image compression scheme [35]), the RCT-based coding [36] and the ICtCp-based coding [27], respectively, to compress color images. The simulations are performed on some classical images, including three popular images (Lena,

Airplane, and Splash), three natural images (Caps, Parrots, and Butterfly), and three screen content images, as shown in Fig. 5. Here, the image block size is fixed at 8×8 and the macro-block size is specified to 16×16 for the compression.

A. Effectiveness of the Error Compensation

According to (20) and (21), the error compensation happened on the Y and Cb components aims at reducing E_{RGB} . To verify the effectiveness of this compensation process, we calculate the number of Y or Cb coefficients that are altered by the error compensation for all test images. Specifically, we integrate SSEDQ in JPEG and the percentages of the



Fig. 5. Test images: Lena, Airplane, Splash, Caps, Parrots, Butterfly, Screenimage_1, Screenimage_2 and Screenimage_3.

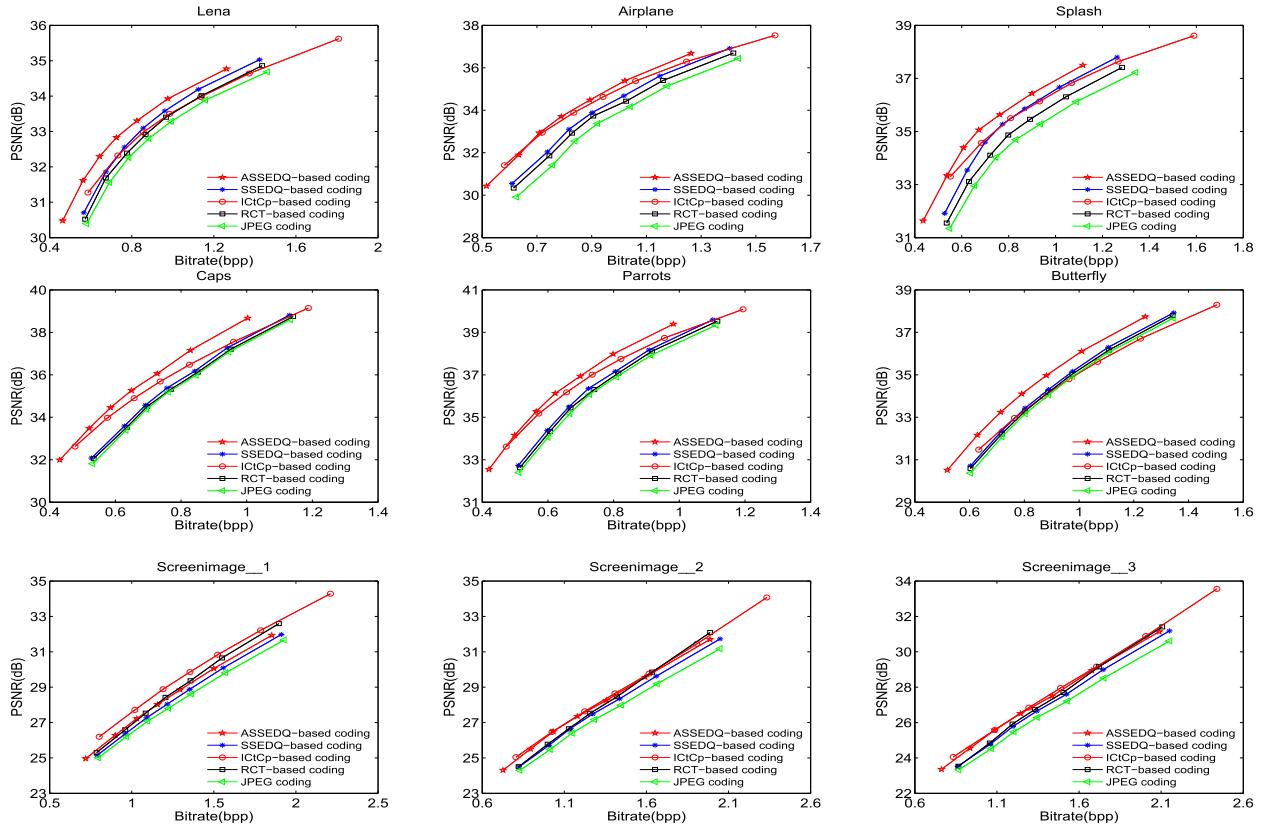


Fig. 6. R-D performances for test images when applying SSEDQ and ASSEDQ to the JPEG coding.

altered Y and Cb coefficients are presented in Table II, where two quantization quality factors (QF) (50 and 70) are tested. It can be seen from Table II that the number of altered Y coefficients is larger as compared to the Cb coefficients. This is due to the fact that the distortions of both Cb and Cr coefficients are employed to reduce the distortions of the Y coefficients, whereas only the quantization errors of the Cr coefficients are used to reduce the distortions of the Cb coefficients. Meanwhile, we observe that more Y and Cb coefficients have been changed by the error compensation when QF increases. Besides, one can find that, for the same QF, more Y and Cb coefficients have been altered in the screen content images. In summary, although the percentages of altered Y and Cb coefficients are rather small (as indicated by Table II), once a change happens, it will cause significant changes to the de-quantized coefficients and the accumulation of all of these changes potentially makes a considerable reduction for the overall SSE distortion.

B. Effectiveness of the SSE-Directed Quantization

We integrate both SSEDQ and ASSEDQ into three classical compression methods, including the JPEG baseline coding,

TABLE II
PERCENTAGE (%) OF ALTERED COEFFICIENTS IN SSEDQ-BASED CODING

	Y component		Cb component	
	QF=50	QF=70	QF=50	QF=70
Lena	3.21	5.09	0.27	0.40
Airplane	3.32	4.73	0.27	0.38
Splash	3.40	5.16	0.22	0.33
Caps	1.61	2.34	0.15	0.19
Parrots	1.89	2.74	0.20	0.26
Butterfly	2.69	3.73	0.29	0.34
Screenimage_1	2.83	4.29	0.38	0.54
Screenimage_2	3.96	5.38	0.45	0.75
Screenimage_3	4.74	6.20	0.50	0.77
Average	3.07	4.41	0.30	0.44

the RCT-based coding and the ICICp-based coding, respectively. Then, we build up six new compression methods: the SSEDQ-based, the ASSEDQ-based, the SSEDQ+RCT-based, the ASSEDQ+RCT-based, the SSEDQ+ICICp-based, and the ASSEDQ+RCT-based coding methods. All compression methods are performed on the test images to verify the effectiveness of SSEDQ and ASSEDQ. The rate-distortion (R-D)

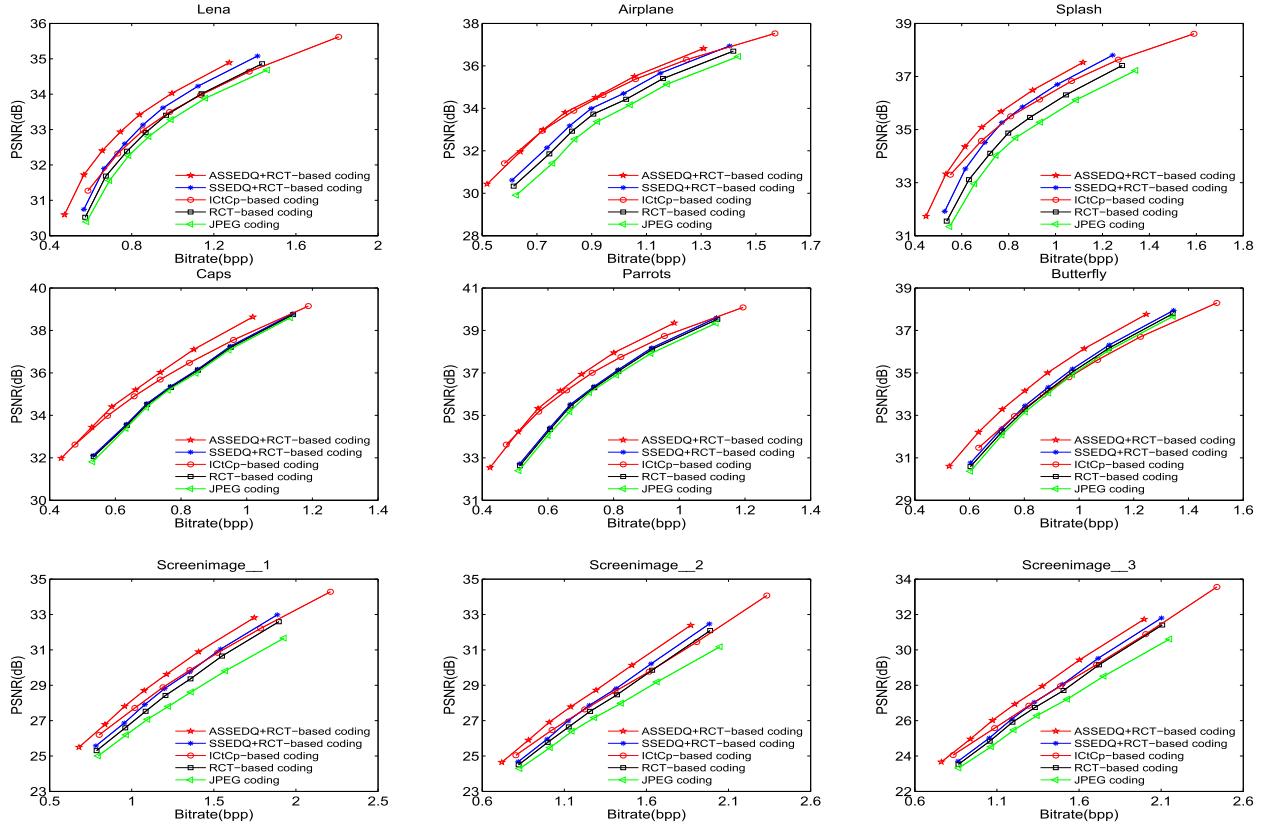


Fig. 7. R-D performances for test images when applying SSEDQ and ASSEDQ to the RCT-based coding.

TABLE III
BD-RATE SAVINGS (%) COMPARED WITH JPEG

	RCT-based	ICICp-based	JND-based	SSEDQ+RCT	SSEDQ+ICICp	ASSEDQ+RCT	ASSEDQ+ICICp
Lena	-4.35	-7.07	-15.32	-9.83	-10.49	-18.91	-19.46
Airplane	-5.40	-16.36	-17.10	-9.48	-19.97	-17.82	-26.03
Splash	-5.74	-16.36	-22.86	-13.88	-22.11	-22.19	-31.12
Caps	-1.17	-9.74	-19.03	-1.77	-10.53	-14.48	-20.04
Parrots	-1.91	-11.12	-16.60	-2.79	-12.05	-14.89	-21.01
Butterfly	-1.48	-0.56	-14.22	-3.13	-1.51	-11.11	-10.69
Screenimage_1	-8.24	-15.22	-19.29	-12.79	-16.15	-21.30	-18.87
Screenimage_2	-6.78	-10.79	-17.63	-10.72	-11.76	-18.38	-14.56
Screenimage_3	-6.65	-11.05	-17.38	-9.88	-11.99	-17.11	-14.66
Average	-4.63	-10.92	-17.71	-8.25	-12.95	-17.16	-19.61

results for this verification are presented in Figs. 6–8, where the PSNR over three color channels is adopted as the distortion metric.

One can see from Figs. 6–8 that integrating ASSEDQ in the classical compression methods have significantly improved their coding performances for all test images. In our proposed ASSEDQ algorithm, the spatial down-sampling and error compensation are used to control the bit-rate and the overall distortion, respectively, and the final RDO procedure determines the most efficient coding mode. As a result, a significant performance gain has been achieved.

On the other hand, applying SSEDQ to three classical methods introduces more gains on images Lena, Airplane, Splash and three screen content images than Caps, Parrots and Butterfly. This phenomenon exists because larger percentages

of altered Y and Cb coefficients are yielded in the former six images - referring to Table II.

C. Comparisons With Various Methods

According to the results shown in Figs. 6–8, one can find that the RCT-based coding outperforms JPEG for all test images. However, the ICICp-based coding cannot guarantee a high compression efficiency all the time, as proved by the limited R-D performances at the high bit-rate for most natural images. Besides, it is seen from Fig. 6 that the SSEDQ-based and ASSEDQ-based coding methods outperform the RCT-based method for all natural images. Then, the ASSEDQ-based coding also performs better than the ICICp-based method for these images. On the other hand, both proposed coding methods offer competitive performances compared

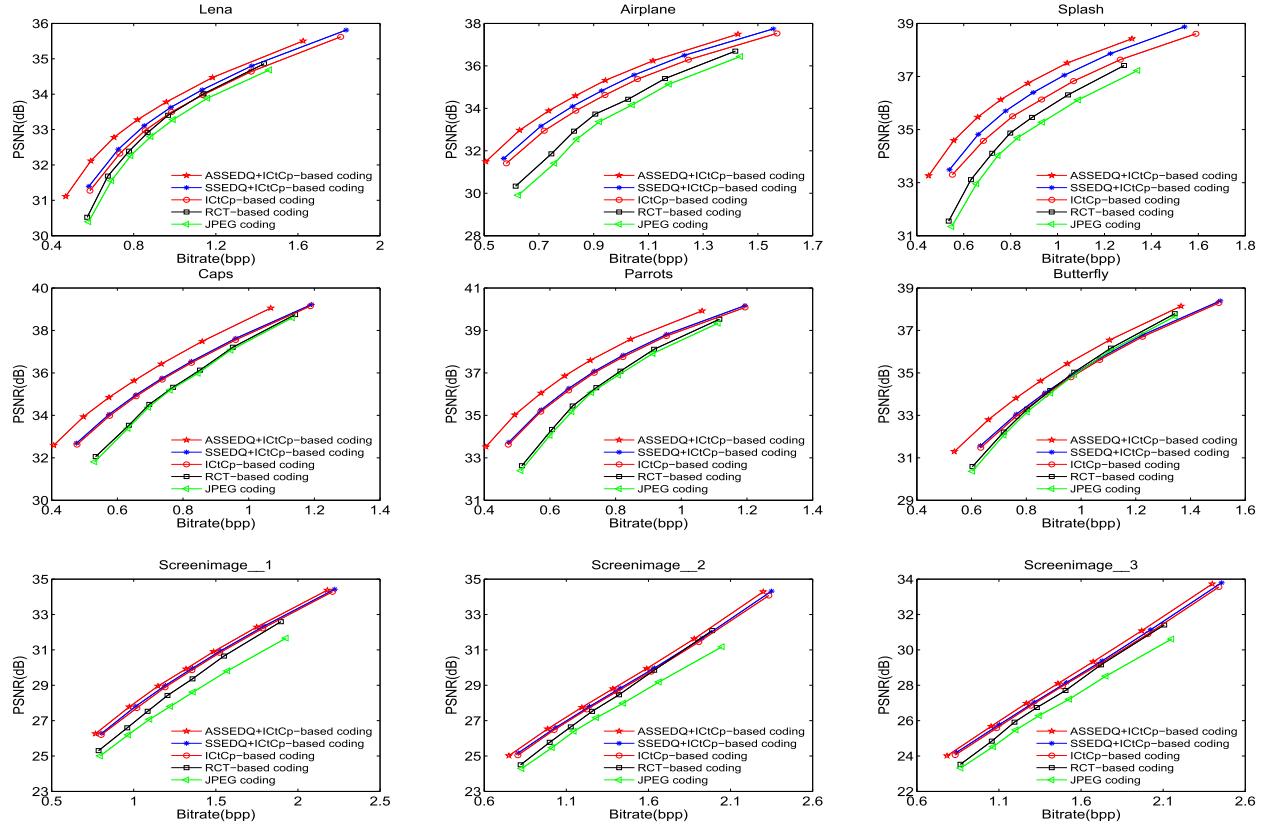


Fig. 8. R-D performances for test images when applying SSEDQ and ASSEDQ to the ICtCp-based coding.

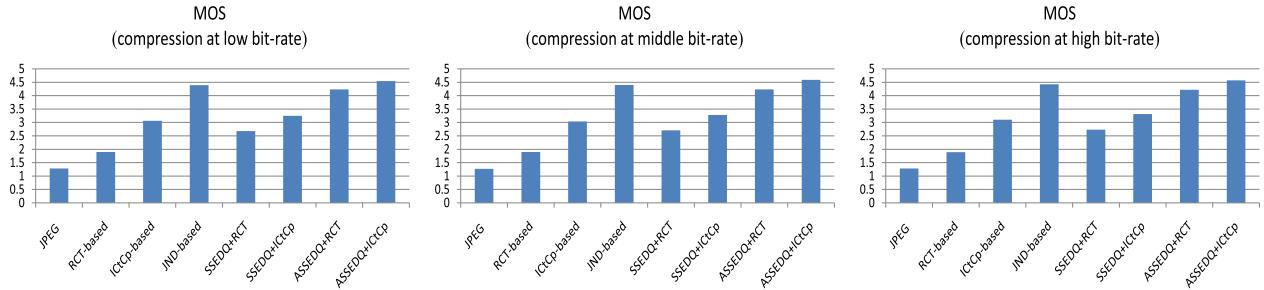


Fig. 9. Averaged MOS for different compression methods.

with the RCT-based and ICtCp-based methods for the last two screen images. Note that the SSEDQ-based and ASSEDQ-based methods are employing the JPEG baseline coding as the basis. After we integrate SSEDQ and ASSEDQ into the RCT-based and ICtCp-based coding methods, the significant gains have been achieved, as demonstrated in Figs. 7 and 8. Moreover, according to the results shown in Figs. 6–8, it is found that integrating SSEDQ and ASSEDQ in either the RCT-based coding or the ICtCp-based coding can achieves a more consistent gain (on average) than integrating them in JPEG.

Next, the BD-rate [37] savings of various compression methods are presented in Table III, where PSNR is adopted as the distortion metric, and two SSEDQ-integrated and two ASSEDQ-integrated methods are included in this comparison. Moreover, the just-noticeable difference (JND) based method [25] is also included to make a comparison. It is

found from Table III that the ASSEDQ+RCT-based method and the ASSEDQ+ICtCp-based method perform better than other methods for most test images.

To conduct a more comprehensive evaluation on the performances of different methods, the perceptual distortion specified by the mean opinion score (MOS) [38] is adopted in this work. The absolute category rating scale is used to evaluate the distortion, where 5 ITU labels (bad, poor, fair, good and excellent) are mapped to 5 numbers between 1 and 5. Specifically, when we compress the same image with different methods, “Excellent” means the best visual quality for a resulted image and “Bad” indicates the worst visual quality for a compressed image. For fair comparisons, we invite 25 participants to evaluate the visual qualities of the resulted images compressed at the same bit-rate but with different methods. The evaluation results for three coding scenarios, i.e., coding with the low, middle and high bit-rates,

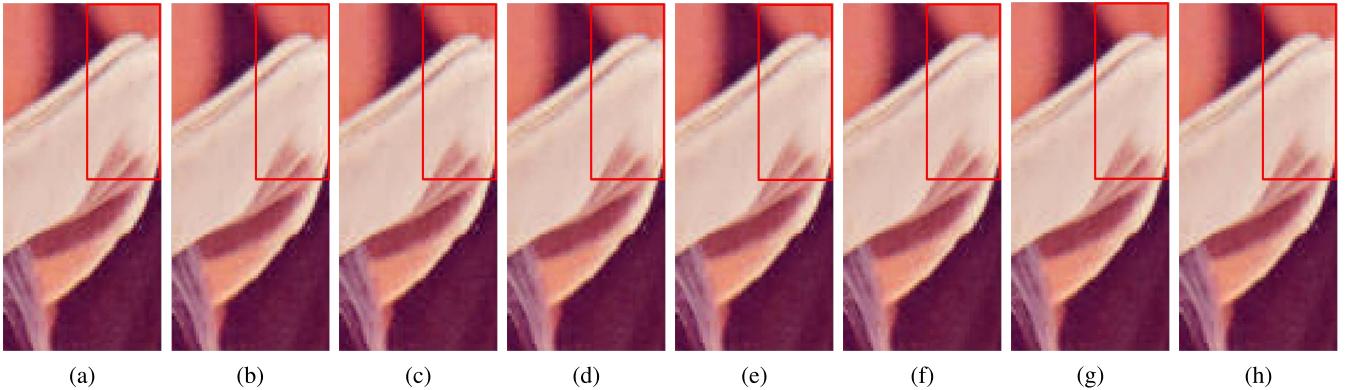


Fig. 10. Image portions of *Lena* (1.15 bpp): (a) JPEG coding, (b) RCT-based coding, (c) ICtCp-based coding, (d) JND-based coding, (e) SSEDQ+RCT-based coding, (f) SSEDQ+ICtCp-based coding, (g) ASSEDQ+RCT-based coding, and (h) ASSEDQ+ICtCp-based coding.

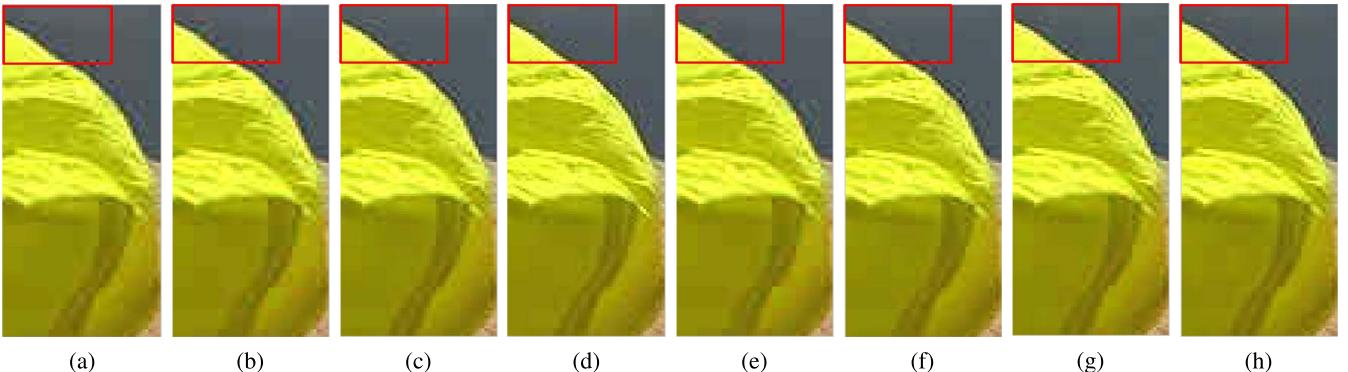


Fig. 11. Image portions of *Caps* (0.60 bpp): (a) JPEG coding, (b) RCT-based coding, (c) ICtCp-based coding, (d) JND-based coding, (e) SSEDQ+RCT-based coding, (f) SSEDQ+ICtCp-based coding, (g) ASSEDQ+RCT-based coding, and (h) ASSEDQ+ICtCp-based coding.

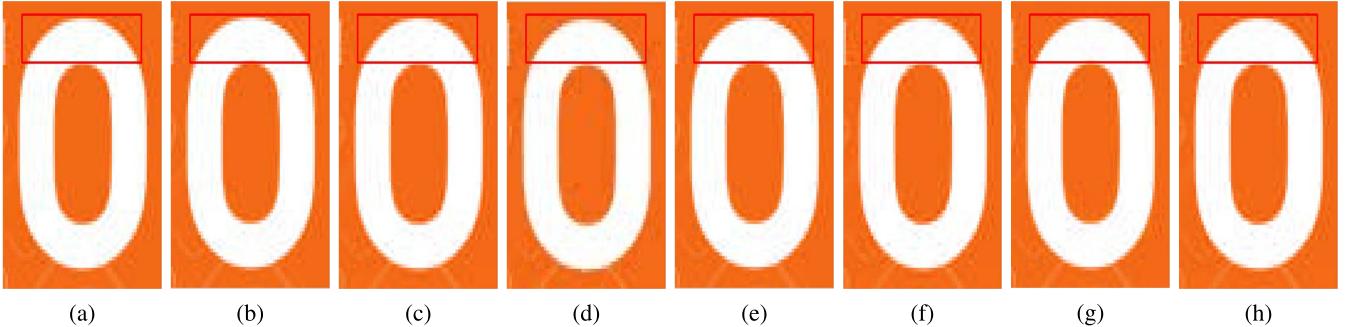


Fig. 12. Image portions of *Screenimage_1* (1.71 bpp): (a) JPEG coding, (b) RCT-based coding, (c) ICtCp-based coding, (d) JND-based coding, (e) SSEDQ+RCT-based coding, (f) SSEDQ+ICtCp-based coding, (g) ASSEDQ+RCT-based coding, and (h) ASSEDQ+ICtCp-based coding.

respectively, are presented in Fig. 9. One can see from Fig. 9 that the compression images obtained by using the ASSEDQ-integrated methods and the JND-based method are more favored by the invited viewers in all coding scenarios. Moreover, the ASSEDQ+ICtCp-based method gets the highest MOS value compared with other methods.

We further perform a perceptual comparison for various methods by presenting some portions of the compressed images in Figs. 10–12. According to these results, when applied to the RCT-based and the ICtCp-based coding schemes, the SSE-directed algorithm, especially ASSEDQ, can improve the perceptual qualities of the compressed images by removing some unpleasant artifacts.

D. Discussion About the Computational Complexity

In Algorithm 1, $d_{k,k}^{(4)}/d_{k,k}^{(3)}$, $d_{k,k}^{(1)}/d_{k,k}^{(0)}$, and $d_{k,k}^{(2)}/d_{k,k}^{(0)}$ are pre-computed and thus fixed for the given transform matrix, which means that they do not contribute any additional computations during the coding process. The extra computations only come from the calculation of all compensation terms and adding them to Cb and Y components. Specifically, according to the first and second steps of Algorithm 1, we need 2 additions and 1 multiplication to implement the compensation for one Cr component. Then, we need 3 additions and 2 multiplications to make the compensation on one Y component. Therefore, our analysis reveals that these extra computations include $5N^2$ additions and $3N^2$ multiplications for an $N \times N$ color block.

TABLE IV
TIME INCREMENT (%) COMPARED WITH JPEG CODING

	JND-based	SSEDQ-based	ASSEDQ-based
Lena	799.34	39.89	129.62
Airplane	489.22	36.69	135.20
Splash	667.15	33.82	128.72
Caps	663.30	38.63	120.07
Parrots	822.83	43.17	134.13
Butterfly	805.56	38.56	131.81
Screenimage_1	1127.04	35.88	140.76
Screenimage_2	949.03	39.08	135.94
Screenimage_3	933.58	38.62	123.75
Average	806.34	38.26	131.11

It is important to note that the extra computations to implement Algorithm 1 are only needed at the encoder side. Therefore, when SSEDQ is integrated in the traditional coding scheme, the decoder does not need any extra computations. On the other hand, if ASSEDQ is applied to the color image compression, the image interpolation performed on the Cr component introduces the additional calculations at both the encoder side and the decoder side. Meanwhile, the RDO-based mode determination occurred at the encoder side also contributes some extra calculations. In fact, it needs $8N^2 + (4N^2 - 1)$ additions and $4N^2$ multiplications to calculate the MSE distortion of each coding mode for an $2N \times 2N$ Cr macro-block. Moreover, 1 multiplication is required to produce the product of MSE and the bit-count for the final determination. In addition, the bicubic interpolation adopted in our work also causes the extra computations at both the encoder side and the decoder side. For an down-sized $N \times N$ Cr block, the complexity of the bicubic interpolation is $O(N^2)$. Based on the above discussions, we can conclude that the complexity of either SSEDQ or ASSEDQ is $O(N^2)$.

Compared with the JPEG coding, the increased running time spent by using our proposed SSEDQ-based, ASSEDQ-based and the JND-based methods is presented in Table IV, where QF is specified to 50. Due to just adopting the new color transforms in the JPEG coding, the RCT-based and ICtCp-based coding methods do not cause any additional computations and the evaluation for their complexities are not included in this work. According to the results presented in Table IV, running the JND-based method costs much more time than applying both SSEDQ and ASSEDQ to compress color images. On the other hand, the increased time caused by employing either the SSEDQ-based method or the ASSEDQ-based method to compress images is still acceptable, which makes both methods very useful in practical applications.

E. Integration in H.265/HEVC

Our proposed SSEDQ algorithm may be integrated into any block-based coding schemes. This makes it suitable to practical applications. Meanwhile, it is flexible to perform the error compensation on any combination of luminance and chrominance components. For example, in H.265/HEVC, the compression strategy performed on Y component determines the compression procedure on Cb and Cr components.

TABLE V
BD-RATE REDUCTION (%) COMPARED WITH HEVC REXT

	QP 22-37	QP 27-42
Lena	-3.76	-3.56
Airplane	-3.33	-3.22
Splash	-4.44	-4.19
Caps	-4.60	-8.19
Parrots	-4.37	-6.27
Butterfly	-2.71	-4.21
Screenimage_1	-1.38	-0.44
Screenimage_2	-2.86	-3.45
Screenimage_3	-2.81	-3.84
Average	-3.36	-4.15

Therefore, we choose to keep the coding of Y component unchanged but only compensate the quantization errors on Cb and Cr components. To achieve this goal, a simple swapping of the corresponding columns of Λ^{-1} is needed to adapt to the components involved in SSEDQ. However, the ASSEDQ algorithm is not preferred to be integrated into the H.265/HEVC coding, because performing the spatial down-sampling on the Y component usually causes an unexpected overall distortion and the compression performance will be degraded.

We integrate our proposed SSEDQ into the HEVC RExt [6] and one simplification has been made in it, i.e., only the 8×8 block-size is enabled. The comparison result in terms of BD-rate reduction has been presented in Table V. One can see from Table V that up to 8.19% BD-rate reduction over HEVC RExt has been achieved by using the SSEDQ algorithm.

V. CONCLUSION

This work has identified a new error source in the coding of color images that is associated with the use of a non-unitary matrix during the RGB-to-YCbCr conversion. To achieve a high compression quality as well as a high compression efficiency, we have proposed an SSE-directed quantization algorithm and applied it to the coding of color images. The proposed new quantization algorithm is designed to minimize the coding distortion in the RGB space rather than in the YCbCr space. It is implemented via an error compensation to reduce the distortion resulted from either the quantization or the interpolation. Adopting the proposed quantization to the color image compression offers a remarkable performance improvement compared with the state-of-the-art methods.

REFERENCES

- [1] Y.-H. Kim, B. Choi, and J. Paik, "High-fidelity RGB video coding using adaptive inter-plane weighted prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1051–1056, Jul. 2009.
- [2] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.
- [3] 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.
- [4] G. J. Sullivan, J.-R. 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.

- [5] Z. Pan, H. Shen, Y. Lu, S. Li, and N. Yu, "A low-complexity screen compression scheme for interactive screen sharing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 6, pp. 949–960, Jun. 2013.
- [6] *High Efficiency Video Coding (HEVC) Range Extensions Text Specification: Draft 7*, document JCTVC-Q1005, ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Mar. 2014.
- [7] D. Flynn *et al.*, "Overview of the range extensions for the HEVC standard: Tools, profiles, and performance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 4–19, Jan. 2016.
- [8] G. Sullivan, *Approximate Theoretical Analysis of RGB to YCbCr to RGB Conversion Error*, document JVT-1017, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Sep. 2003.
- [9] W.-S. Kim, D.-S. Cho, and H.-M. Kim, "Interplane prediction for RGB video coding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004, pp. 785–788.
- [10] H. M. Kim, W. S. Kim, and D. S. Cho, "A new color transform for RGB coding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004, pp. 107–110.
- [11] H. S. Malvar, G. J. Sullivan, and S. Srinivasan, "Lifting-based reversible color transformations for image compression," *Proc. SPIE*, vol. 7073, pp. 707307-1–707307-10, Aug. 2008.
- [12] S.-C. Pei and J.-J. Ding, "Improved reversible integer-to-integer color transforms," in *Proc. IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 473–476.
- [13] W. Dai and T. D. Tran, "Regularity-constrained pre- and post-filtering for block DCT-based systems," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2568–2581, Oct. 2003.
- [14] G. Zhai, W. Zhang, X. Yang, W. Lin, and Y. Xu, "Efficient deblocking with coefficient regularization, shape-adaptive filtering, and quantization constraint," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 735–745, Aug. 2008.
- [15] S. B. Yoo, K. Choi, and J. B. Ra, "Post-processing for blocking artifact reduction based on inter-block correlation," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1536–1548, Oct. 2014.
- [16] H. Chang, M. K. Ng, and T. Zeng, "Reducing artifacts in JPEG decompression via a learned dictionary," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 718–728, Feb. 2014.
- [17] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, "Postprocessing of compressed images via sequential denoising," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3044–3058, Jul. 2016.
- [18] X. Zhang, W. Lin, R. Xiong, X. Liu, S. Ma, and W. Gao, "Low-rank decomposition-based restoration of compressed images via adaptive noise estimation," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4158–4171, Sep. 2016.
- [19] 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. 305–315, Mar. 2008.
- [20] J. Xu, F. Wu, J. Liang, and W. Zhang, "Directional lapped transforms for image coding," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 85–97, Jan. 2010.
- [21] S. Y. Zhu, S.-K. A. Yeung, and B. Zeng, "In search of ‘better-than-DCT’ unitary transforms for encoding of residual signals," *IEEE Signal Process. Lett.*, vol. 17, no. 11, pp. 961–964, Nov. 2010.
- [22] H. Chen and B. Zeng, "New transforms tightly bounded by DCT and KLT," *IEEE Signal Process. Lett.*, vol. 19, no. 6, pp. 344–347, Jun. 2012.
- [23] 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.
- [24] C. Sun and E.-H. Yang, "An efficient DCT-based image compression system based on Laplacian transparent composite model," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 886–900, Mar. 2015.
- [25] 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.
- [26] 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.
- [27] "White paper on ICtCp, version 7.2," Dolby, San Francisco, CA, USA, Tech. Rep., 2016.
- [28] S. Zhu, M. Li, C. Chen, S. Liu, and B. Zeng, "Cross-space distortion directed color image compression," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 525–538, Mar. 2018.
- [29] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. TIT-28, no. 2, pp. 129–137, Mar. 1982.
- [30] T. Berger, "Optimum quantizers and permutation codes," *IEEE Trans. Inf. Theory*, vol. TIT-18, no. 6, pp. 759–765, Nov. 1972.
- [31] *Parameter Values for High Definition Television Systems for Production and International Programme Exchange*, document Rec. BT. 709-5, ITU-R, Apr. 2002.
- [32] X. M. Zhang and B. A. Wandell, "Color image fidelity metrics evaluated using image distortion maps," *Signal Process.*, vol. 20, no. 3, pp. 201–214, Nov. 1998.
- [33] K.-L. Chung, T.-C. Hsu, and C.-C. Huang, "Joint chroma subsampling and distortion-minimization-based luma modification for RGB color images with application," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4626–4638, Oct. 2017.
- [34] G. A. F. Seber, *A Matrix Handbook for Statisticians*. Hoboken, NJ, USA: Wiley, 2008.
- [35] G. Hudson, A. Léger, B. Niss, and I. Sebestyén, "JPEG at 25: Still going strong," *IEEE Multimedia*, vol. 24, no. 2, pp. 96–103, Apr./Jun. 2017.
- [36] L. Zhang, J. Chen, J. Sole, M. Karczewicz, X. Xiu, and J.-Z. Xu, "Adaptive color-space transform for HEVC screen content coding," in *Proc. Data Compress. Conf.*, Apr. 2015, pp. 233–242.
- [37] G. Bjøntegaard, *Calculation Average PSNR Differences Between RD Curves*, document VCEG-M33, 13th ITU-T Video Coding Experts Group (VCEG) Meeting, Austin, TX, USA, Apr. 2001.
- [38] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, document Rec. BT.500, ITU-R, Geneva, Switzerland, 2002.



Shuyuan Y. 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 Hong Kong Applied Science and Technology Research Institute Company Limited, 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. He has over 60 research publications. His research interests include image/video compression, image processing, and compressive sensing. He served as the committee member for IEEE ICME-2014, VCIP-2016, and PCM-2017. He is a member of IEEE CAS society. He was a recipient of 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.



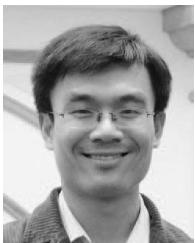
Zhiying Y. He received the B.Eng. degree in electronic engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2015, where he is currently pursuing the M.Eng. degree with the School of Information and Communication Engineering. His research interest is image/video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



Chen Chen (S'15–M'17) received the B.Eng. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology, Hong Kong, China, in 2017. He is currently a Senior Researcher with Tencent Company Ltd., Shenzhen, China. His research interests focus on video compression, multimedia processing, computer vision, and artificial intelligence. He was a recipient of the Top 10% paper award at VCIP-2016.



Shuaicheng C. Liu (M'15) received the B.Eng. degree from Sichuan University, Chengdu, China, in 2008, and the M.Eng. and Ph.D. degrees from the National University of Singapore, Singapore, in 2010 and 2014, respectively. In 2014, he joined University of Electronic Science and Technology of China, where he is currently an Associate Professor with the School of Electronic Engineering, Institute of Image Processing. His research interests include computer vision and computer graphics.



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 with 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 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 Tampere University of Technology, Tampere, Finland, in 1991.

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

During his tenure at HKUST and UESTC, he graduated more than 30 master's and Ph.D. students, received about 20 research grants, filed 8 international patents, and published over 260 papers. Three representing works are as follows: one paper on fast block motion estimation, published in *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY* (TCSVT) in 1994, has so far been SCI-cited over 1000 times (Google-cited over 2200 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, led to a patent that has been successfully licensed to companies; and one paper on directional discrete cosine transform, published in *IEEE TCSVT* in 2008, which received 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 received a Second Class Natural Science Award (the first recipient) from the Chinese Ministry of Education in 2014. He was a General Co-Chair of the IEEE VCIP-2016, held in Chengdu, China, in 2016. He is currently on the Editorial Board of *Journal of Visual Communication and Image Representation* and serves as the General Co-Chair of PCM-2017. He served as an Associate Editor for *IEEE TCSVT* for eight years and received the Best Associate Editor Award in 2011. He was elected as an IEEE Fellow for contributions to image and video coding in 2016.



Yuanfang Guo (S'11-M'15) received the B.Eng. and Ph.D. degrees from The Hong Kong University of Science and Technology, Hong Kong, China, in 2009 and 2015, respectively. He is currently an Assistant Professor with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His research interests include image/video watermarking, data hiding, forensics, compression, restoration, and enhancement.