This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2895840, IEEE Transactions on Circuits and Systems for Video Technology

Efficient Chroma Sub-Sampling and Luma Modification for Color Image Compression

Shuyuan Zhu, *Member, IEEE*, Chang Cui, Ruiqin Xiong, *Member, IEEE*, Yuanfang Guo, *Member, IEEE*, and Bing Zeng, *Fellow, IEEE*

***Abstract*—In color image compression, the chroma components are often sub-sampled before compression and up-sampled after compression. Although sub-sampling the chroma components saves bit-cost for compression, it often induces extra color distortions in the compressed images. In this paper, we propose two approaches to tackle this problem. Firstly, we propose a sub-sampling method in the transform domain and apply it to both the chroma components. Then, based on this sub-sampling, we propose a novel method to modify the luma component. In our proposed luma modification algorithm, the distortions occurred on two chroma components can be coupled together and utilized to modify the luma component. With our proposed chroma sub-sampling and luma modification algorithms, we can achieve a low RGB distortion in practical image coding. Experimental results demonstrate that our proposed methods offer more significant coding gains compared with the state-of-the-art methods for the compression of color images.**

***Index Terms*—Color image, compression, distortion, sub-sampling, modification**

I. Introduction

COLOR image compression typically happens in the YUV space, where the input RGB signal is converted into the YUV signal which is composed by one luma (Y) component and two chroma (U and V) components. The chroma component is usually more smooth than the luma component, and the latter one contains plenty of textures. After the color space conversion, the luma and chroma components are divided into image blocks. Then, the discrete cosine transform (DCT), quantization and entropy coding are accordingly performed on these blocks to achieve the compression.

In general, the RGB-to-YUV space conversion removes the redundancies between color channels [1], and the compression of the YUV signal can achieve a high compression ratio more easily. In practical YUV based image compression, the chroma sub-sampling is employed very frequently to reduce the image data-size. This sub-sampling saves a considerable bit-cost, and thus potentially improves the coding efficiency. Currently, the most popular chroma sub-sampling scheme down-samples two chroma components separately by a factor of 2 in both the horizonal and vertical directions in the pixel domain, leading to the YUV 4:2:0 format.

Unfortunately, sub-sampling of the chroma components usually induces extra distortions to the compressed RGB images. These distortions may be reduced by applying the interpolation dependent sub-sampling [2]–[5] to each individual chroma component. Note that these techniques [2]–[5] must be carried out in the YUV space. In practice, they cannot guarantee a low RGB distortion for the compressed color image. To solve this problem, the RGB distortion constrained chroma sub-sampling is proposed in [6], where sub-sampling of the chroma components is optimized to achieve a lower RGB distortion. The solution in [6] has demonstrated some promising results when it is employed in the 4:2:0 format coding.

Besides of the optimization of the sub-sampling operation for chroma components, the luma component may also be modified to achieve a high-quality color image compression [7]. In [7], the traditional spatial sub-sampling is performed on two chroma components and a low RGB distortion is guaranteed by modifying the luma component according to the sub-sampled chroma components. This method is named as the joint chroma sub-sampling and distortion minimization-based luma modification (JCSLM). Unfortunately, JCSLM has to be carried out before compression. When it is adopted in practical coding schemes, the compressed color image still suffers from certain compression distortion, which accordingly limits the coding efficiency.

In this paper, we firstly propose a transform domain sub-sampling scheme for both the chroma components. Our proposed method targets a low RGB distortion and yields a comparable sub-sampling performance to the YUV 4:2:0 format. Specifically, this proposed scheme converts each $N \times N$ chroma block into an $(N/2) \times (N/2)$ coefficient block, which accordingly induces a low bit-cost for compression. Then, based on this sub-sampling scheme, we propose a novel method to modify the luma component. When our proposed modification algorithm is adopted in practical image coding, the distortions, including both the sub-sampling and compression distortions, occurred on two chroma components can be coupled together and utilized to modify the luma component. At last, we can achieve the state-of-the-art performance, i.e., a low RGB distortion as well as a high coding efficiency, for the compression of color images.

II. RGB Color Distortion

The RGB-to-YUV space conversion is implemented by performing a transform on the red (R), green (G) and blue (B) pixels $\{x_{R_i}, x_{G_i}, x_{B_i}\}$ to produce the corresponding Y, U and V pixels $\{x_{Y_i}, x_{U_i}, x_{V_i}\}$ as

Manuscript received November 1, 2018; revised January 14, 2019; accepted January 23, 2019. This work was supported in part by the National Natural Science Foundation of China (61672134, 61772041, 61802391, and 61720106004), in part by the Fundamental Research Funds for Central Universities of China under Grant ZYGX2016J038, and in part by the "111" Projects (B17008). *(Corresponding authors: Yuanfang Guo; Bing Zeng.)*

S. Y. Zhu, C. Cui, and B. Zeng are with School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China.

R. Q. Xiong is with the Institute of Digital Media, School of Electronic Engineering and Computer Science, Peking University, Beijing, China.

Y. F. Guo is with the Laboratory of Intelligent Recognition and Image Processing, School of Computer Science and Engineering, Beihang University, Beijing, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2895840, IEEE Transactions on Circuits and Systems for Video Technology

$$\begin{bmatrix} x_{Y_i} \\ x_{U_i} \\ x_{V_i} \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_{R_i} \\ x_{G_i} \\ x_{B_i} \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}, \tag{1}$$

where $\mathbf{A}$ is the color transform matrix.

If the degradation, such as the sub-sampling or the compression, happens to the Y, U and V pixels, the reconstructed R, G and B pixels $\{\hat{x}_{R_i}, \hat{x}_{G_i}, \hat{x}_{B_i}\}$ should be determined by the degraded Y, U and V pixels $\{\hat{x}_{Y_i}, \hat{x}_{U_i}, \hat{x}_{V_i}\}$ based on the inverse transform of (1). Then, the color distortion measured on $\{x_{R_i}, x_{G_i}, x_{B_i}\}$ is calculated as

$$\begin{aligned} d_{RGB_i} &= \begin{bmatrix} x_{R_i} - \hat{x}_{R_i} \\ x_{G_i} - \hat{x}_{G_i} \\ x_{B_i} - \hat{x}_{B_i} \end{bmatrix}^T \begin{bmatrix} x_{R_i} - \hat{x}_{R_i} \\ x_{G_i} - \hat{x}_{G_i} \\ x_{B_i} - \hat{x}_{B_i} \end{bmatrix} \\ &= \begin{bmatrix} x_{Y_i} - \hat{x}_{Y_i} \\ x_{U_i} - \hat{x}_{U_i} \\ x_{V_i} - \hat{x}_{V_i} \end{bmatrix}^T (\mathbf{A}^{-1})^T (\mathbf{A}^{-1}) \begin{bmatrix} x_{Y_i} - \hat{x}_{Y_i} \\ x_{U_i} - \hat{x}_{U_i} \\ x_{V_i} - \hat{x}_{V_i} \end{bmatrix}. \end{aligned} \tag{2}$$

Let us use $\mathbf{x}_Y = [x_{Y_i}]_{K\times 1}$, $\mathbf{x}_U = [x_{U_i}]_{K\times 1}$ and $\mathbf{x}_V = [x_{V_i}]_{K\times 1}$ to represent the K-point column-vectors for the Y, U and V components, respectively. When the degradation happens, with the degraded luma and chroma components, i.e., $\hat{\mathbf{x}}_Y = [\hat{x}_{Y_i}]_{K\times 1}$, $\hat{\mathbf{x}}_U = [\hat{x}_{U_i}]_{K\times 1}$ and $\hat{\mathbf{x}}_V = [\hat{x}_{V_i}]_{K\times 1}$, we calculate the distortions of the Y, U and V components as $\mathbf{e}_Y = \mathbf{x}_Y - \hat{\mathbf{x}}_Y$, $\mathbf{e}_U = \mathbf{x}_U - \hat{\mathbf{x}}_U$ and $\mathbf{e}_V = \mathbf{x}_V - \hat{\mathbf{x}}_V$, respectively. Then, based on $\mathbf{e}_Y$, $\mathbf{e}_U$ and $\mathbf{e}_V$, the YUV error vector can be constructed as $\mathbf{e}_{YUV} = [\mathbf{e}_Y^T\ \mathbf{e}_U^T\ \mathbf{e}_V^T]^T$. Similarly, in the RGB space, the corresponding RGB error vector $\mathbf{e}_{RGB}$ can be obtained.

By assuming that $\lambda_{m,n} (m, n = 0, 1, 2)$ is the element of $\mathbf{A}^{-1}$, we can exploit $\lambda_{m,n}$ to form a color conversion matrix $\mathbf{\Lambda}$ such that

$$\mathbf{e}_{RGB} = \mathbf{\Lambda}\mathbf{e}_{YUV}. \tag{3}$$

Specifically, $\mathbf{\Lambda}$ is formed as

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_{00} & \mathbf{\Lambda}_{01} & \mathbf{\Lambda}_{02} \\ \mathbf{\Lambda}_{10} & \mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{20} & \mathbf{\Lambda}_{21} & \mathbf{\Lambda}_{22} \end{bmatrix},$$

where each $\mathbf{\Lambda}_{m,n}$ is the $K \times K$ diagonal matrix and $\mathbf{\Lambda}_{m,n} = diag(\lambda_{m,n}, \cdots, \lambda_{m,n})$. Based on (3), the RGB distortion can be calculated as

$$\begin{aligned} d_{RGB} &= \mathbf{e}_{RGB}^T \mathbf{e}_{RGB} \\ &= \mathbf{e}_{YUV}^T (\mathbf{\Lambda}^T \mathbf{\Lambda}) \mathbf{e}_{YUV}. \end{aligned} \tag{4}$$

III. Upward-Conversion Guided Transform Domain Sub-Sampling

In the sub-sampling based image coding, sub-sampling is usually carried out in the pixel domain to reduce the resolution of the input image. Besides, sub-sampling can also be performed in the transform domain by dropping some of the high-frequency coefficients from the transformed block [8], [9]. Compared with the pixel domain sub-sampling based coding scheme, image interpolation is not required in the transform domain sub-sampling based scheme to reconstruct the image signal, which yields a lower computational complexity. In this work, we propose the upward-conversion guided transform domain sub-sampling for the compression of color images.

Based on the Kronecker product, the 2-D transform of an $N \times N$ image block can be implemented in a 1-D manner as $\mathbf{X} = (\mathbf{C} \otimes \mathbf{C})\mathbf{x}$, where $\mathbf{x}$ is the $N \times N$ block in vector form, $\mathbf{X}$ contains N^2 transform coefficients, $\otimes$ denotes the Kronecker product and $\mathbf{C}$ is the $N \times N$ transform matrix. Meanwhile, if an $N^2 \times N^2$ matrix $\mathbf{D}$ is defined as $\mathbf{D} = \mathbf{C}^{-1} \otimes \mathbf{C}^{-1}$, the inverse transform can be obtained via $\mathbf{x} = \mathbf{D}\mathbf{X}$.

If we employ the first M coefficients of $\mathbf{X}$ to form a low-frequency vector $\mathbf{X}_L$ and utilize the other $(N^2 - M)$ coefficients to form a high-frequency vector $\mathbf{X}_H$, $\mathbf{X}$ can be partitioned as $\mathbf{X} = [\mathbf{X}_L^T\ \mathbf{X}_H^T]^T$. Also, we can partition $\mathbf{D}$ as $\mathbf{D} = [\mathbf{D}_L\ \mathbf{D}_H]$ such that

$$\mathbf{x} = [\mathbf{D}_L\ \mathbf{D}_H] \begin{bmatrix} \mathbf{X}_L \\ \mathbf{X}_H \end{bmatrix}, \tag{5}$$

where $\mathbf{D}_L$ is an $N^2 \times M$ matrix and $\mathbf{D}_H$ is an $N^2 \times (N^2 - M)$ matrix. If we remove $\mathbf{X}_H$ and only reserve $\mathbf{X}_L$ for further processing, the transform domain sub-sampling for $\mathbf{X}$ is accordingly achieved. With the sub-sampled $\mathbf{X}$, i.e., $\mathbf{X}_L$, $\mathbf{x}$ can be rebuilt as $\hat{\mathbf{x}} = \mathbf{D}_L\mathbf{X}_L$. However, due to the removal of the high-frequency components, rebuilding $\mathbf{x}$ directly from $\mathbf{X}_L$ tends to induce a high distortion. When the compression is performed on $\mathbf{X}_L$, the reconstruction of $\mathbf{x}$ from the compressed $\mathbf{X}_L$ will suffer from a more serious distortion.

To achieve a high-quality reconstruction, we propose to preserve the high-frequency information as much as possible in the remaining low-frequency coefficients. Specifically, we produce $\mathbf{X}_L$ with the minimal reconstruction distortion as

$$\tilde{\mathbf{X}}_L = \arg\min_{\mathbf{X}_L} \|\mathbf{x} - \mathbf{D}_L\mathbf{X}_L\|_2^2. \tag{6}$$

Then, the optimal $\mathbf{X}_L$ is generated as

$$\tilde{\mathbf{X}}_L = (\mathbf{D}_L^T\mathbf{D}_L)^{-1}(\mathbf{D}_L^T\mathbf{x}). \tag{7}$$

According to (7), a sub-sampled coefficient vector with M coefficients is produced and these coefficients will be further exploited to reconstruct $\mathbf{x}$, leading to our proposed upward-conversion guided transform domain sub-sampling (UGTS).

The proposed UGTS can be applied to both the chroma components for color image compression. Specifically, we specify $M = N^2/4$ to yield a comparable sub-sampling performance to the YUV 4:2:0 format. With UGTS, the high-frequency information of an $N \times N$ chroma block will be preserved in the $N^2/4$ remaining coefficients as much as possible. These coefficients are further compressed by quantization and entropy coding, and a considerable bit-cost can be saved accordingly.

IV. Our Proposed Luma Modification Method

Our proposed UGTS algorithm can certainly be applied to the chroma components to reduce the color distortion caused by sub-samplings. Meanwhile, the color distortion can also be reduced by modifying the luma component. In this work, we propose a novel method based on UGTS to modify the luma

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2895840, IEEE Transactions on Circuits and Systems for Video Technology

component for a low RGB color distortion. When this luma modification method is applied to practical image coding, both the sub-sampling and the compression distortions of the two chroma components can be coupled together and utilized to modify the luma component. Then, a low RGB distortion can be accordingly achieved in practical image coding.

A. Decomposition of RGB Distortion

If we define $\mathbf{W} = \mathbf{\Lambda}^T\mathbf{\Lambda}$, (4) can be rewritten as

$$d_{RGB} = \mathbf{e}_{YUV}^T \mathbf{W} \mathbf{e}_{YUV}, \tag{8}$$

where $\mathbf{W}$ is a $3N^2 \times 3N^2$ symmetric matrix. To make a much clearer analysis of the composition of d_{RGB}, we perform the Cholesky factorization on $\mathbf{W}$ to factorize it as $\mathbf{W} = \mathbf{V}^T\mathbf{V}$, where $\mathbf{V}$ is a $3N^2 \times 3N^2$ upper triangular matrix, i.e.,

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_0 & \mathbf{V}_1 & \mathbf{V}_2 \\ \mathbf{0} & \mathbf{V}_3 & \mathbf{V}_4 \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_5 \end{bmatrix}, \tag{9}$$

and each sub-matrix $\mathbf{V}_i = diag(\gamma_i, \cdots, \gamma_i)\{i = 0, \cdots, 5\}$ is an $N^2 \times N^2$ diagonal matrix. With the $N^2 \times N^2$ identity matrix $\mathbf{I}$, each $\mathbf{V}_i$ can be represented by $\mathbf{V}_i = \gamma_i \mathbf{I}$.

Based on (8) and (9), d_{RGB} can be decomposed as

$$d_{RGB} = \sum_{k=0}^{2} d_k, \tag{10}$$

where

$$\begin{aligned} d_0 =& (\mathbf{V}_0\mathbf{e}_Y + \mathbf{V}_1\mathbf{e}_U + \mathbf{V}_2\mathbf{e}_V)^T(\mathbf{V}_0\mathbf{e}_Y \\ &+ \mathbf{V}_1\mathbf{e}_U + \mathbf{V}_2\mathbf{e}_V) \\ =& \|\mathbf{V}_0\mathbf{e}_Y + \mathbf{V}_1\mathbf{e}_U + \mathbf{V}_2\mathbf{e}_V\|_2^2 \\ =& \|\gamma_0\mathbf{e}_Y + \gamma_1\mathbf{e}_U + \gamma_2\mathbf{e}_V\|_2^2, \end{aligned} \tag{11}$$

$$\begin{aligned} d_1 =& (\mathbf{V}_3\mathbf{e}_U + \mathbf{V}_4\mathbf{e}_V)^T(\mathbf{V}_3\mathbf{e}_U + \mathbf{V}_4\mathbf{e}_V) \\ =& \left\|\gamma_3\mathbf{e}_U + \gamma_4\mathbf{e}_V\right\|_2^2, \end{aligned} \tag{12}$$

and

$$\begin{aligned} d_2 =& (\mathbf{V}_5\mathbf{e}_V)^T(\mathbf{V}_5\mathbf{e}_V) \\ =& \|\gamma_5\mathbf{e}_V\|_2^2. \end{aligned} \tag{13}$$

According to the decomposition of d_{RGB} above, we can minimize d_{RGB} by minimizing each d_i independently. Moreover, this minimization provides a potential solution to simultaneously modify the luma component and sub-sample the chroma components to achieve a low RGB distortion.

B. Luma Modification with Transform Domain Sub-Sampling of the Chroma Components

For convenience, we focus on the 1-D case to describe how to achieve the minimal d_{RGB} by jointly modifying the luma component and sub-sampling the chroma components. Let us use $\mathbf{x}_Y$, $\mathbf{x}_U$ and $\mathbf{x}_V$ to represent the N^2-point pixel vectors converted from the $N \times N$ Y, U and V blocks, respectively. Let $\mathbf{X}_{U_L}$ and $\mathbf{X}_{V_L}$ be the $(N^2/4)$-point vectors converted from the $(N/2) \times (N/2)$ sub-sampled U and V coefficient blocks, respectively. In our proposed method, the optimal modified luma pixels and the optimal sub-sampled chroma coefficients will be produced to minimize d_{RGB}.

Based on the decomposition of d_{RGB}, i.e., (10)−(13), both the optimal sub-sampled chroma coefficients and the luma component can be determined in an alternating iterative manner as follows.

According to our proposed UGTS algorithm, when the transform domain sub-sampling is applied to the V component, the optimal $\mathbf{X}_{V_L}$, which minimizes d_2, is determined by

$$\tilde{\mathbf{X}}_{V_L} = (\mathbf{D}_L^T\mathbf{D}_L)^{-1}(\mathbf{D}_L^T\mathbf{x}_V). \tag{14}$$

With $\tilde{\mathbf{X}}_{V_L}$, $\mathbf{e}_V$ is calculated via $\mathbf{e}_V = \mathbf{x}_V - \mathbf{D}_L\tilde{\mathbf{X}}_{V_L}$.

If the transform domain sub-sampling is also applied to the U component, $\mathbf{e}_U$ should be calculated by $\mathbf{e}_U = \mathbf{x}_U - \mathbf{D}_L\mathbf{X}_{U_L}$ and (12) can be rewritten as

$$d_1 = \left\|\gamma_3(\mathbf{x}_U - \mathbf{D}_L\mathbf{X}_{U_L}) + \gamma_4\mathbf{e}_V\right\|_2^2 \quad . \tag{15}$$

According to (15), the optimal $\mathbf{X}_{U_L}$, which minimizes d_1, is determined via

$$\begin{aligned} \tilde{\mathbf{X}}_{U_L} &= (\mathbf{D}_L^T\mathbf{D}_L)^{-1}\mathbf{D}_L^T(\mathbf{x}_U + \frac{\gamma_4}{\gamma_3}\mathbf{e}_V) \\ &= (\mathbf{D}_L^T\mathbf{D}_L)^{-1}\mathbf{D}_L^T(\mathbf{x}_U + \frac{\gamma_4}{\gamma_3}(\mathbf{x}_V - \mathbf{D}_L\tilde{\mathbf{X}}_{V_L})). \end{aligned} \tag{16}$$

As can be concluded from (16), the optimal sub-sampled coefficients of the U component are determined not only by the original pixels of the U component, but also by the distortions of the V component.

Then, with the optimal $\tilde{\mathbf{X}}_{U_L}$ and $\tilde{\mathbf{X}}_{V_L}$, d_0 is computed by

$$d_0 = \gamma_0^2\left\|\mathbf{e}_Y + \frac{1}{\gamma_0}(\gamma_1\mathbf{e}_U + \gamma_2\mathbf{e}_V)\right\|_2^2. \tag{17}$$

If the luma component is modified, d_0 should be determined by the modified $\mathbf{x}_Y$, which is denoted as $\tilde{\mathbf{x}}_Y$, as

$$d_0 = \gamma_0^2\left\|(\mathbf{x}_Y - \tilde{\mathbf{x}}_Y) + \frac{1}{\gamma_0}(\gamma_1\mathbf{e}_U + \gamma_2\mathbf{e}_V)\right\|_2^2 \tag{18}$$

According to (18), if the minimal d_0 is desired, we can modify $\mathbf{x}_Y$ as

$$\begin{aligned} \tilde{\mathbf{x}}_Y =& \mathbf{x}_Y + \frac{1}{\gamma_0}(\gamma_1\mathbf{e}_U + \gamma_2\mathbf{e}_V) \\ =& \mathbf{x}_Y + \frac{\gamma_1}{\gamma_0}(\mathbf{x}_U - \mathbf{D}_L\tilde{\mathbf{X}}_{U_L}) + \frac{\gamma_2}{\gamma_0}(\mathbf{x}_V - \mathbf{D}_L\tilde{\mathbf{X}}_{V_L}). \end{aligned} \tag{19}$$

Based on the above modification, $d_0 = 0$ can be achieved accordingly. At last, with the minimal d_0, d_1 and d_2, a minimal d_{RGB} can be achieved.

After obtaining the sub-sampled chroma coefficients, i.e., $\{\tilde{\mathbf{X}}_{U_L}, \tilde{\mathbf{X}}_{V_L}\}$, and the modified luma component, i.e., $\tilde{\mathbf{x}}_Y$, we can further process them to achieve the compression.

C. Compression-Dependent Luma Modification

In practical image coding, if the compression is performed on both the chroma components, we can also implement our proposed luma modification based on the compressed U and V components, and thus propose the compression-dependent luma modification algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2895840, IEEE Transactions on Circuits and Systems for Video Technology

Let $\check{\mathbf{X}}_{V_L}$ denote the compressed $\tilde{\mathbf{X}}_{V_L}$. When the compression is applied to the sub-sampled V component, the overall distortion of the V component, including both the sub-sampling and compression distortions, is calculated as $\mathbf{e}_V = \mathbf{x}_V - \mathbf{D}_L\check{\mathbf{X}}_{V_L}$. Based on $\mathbf{e}_V$, the optimal $\mathbf{X}_{U_L}$, i.e., $\tilde{\mathbf{X}}_{U_L}$, can be computed according to (16).

Similarly, with the compressed $\tilde{\mathbf{X}}_{U_L}$, which is denoted as $\check{\mathbf{X}}_{U_L}$, the overall distortion of the U component can be calculated as $\mathbf{e}_U = \mathbf{x}_U - \mathbf{D}_L\check{\mathbf{X}}_{U_L}$. Based on $\mathbf{e}_U$ and $\mathbf{e}_V$, the modified luma component $\tilde{\mathbf{x}}_Y$ can be obtained according to (19). Then, we apply the compression to $\tilde{\mathbf{x}}_Y$ and generate the compressed $\tilde{\mathbf{x}}_Y$, which is denoted as $\check{\mathbf{x}}_Y$. At last, d_0 is calculated by

$$d_0 = \gamma_0^2\left\|\tilde{\mathbf{x}}_Y - \check{\mathbf{x}}_Y\right\|_2^2. \tag{20}$$

Comparing (20) with (11), after our proposed luma modification, d_0 is only determined by the compression distortion of $\tilde{\mathbf{x}}_Y$. Note that this distortion is highly related to the quantization adopted in compression rather than the distortions occurred on the two chroma components. Therefore, a smaller d_{RGB} can be achieved for practical image coding.

V. Experimental Results

The conversion matrix employed in [7] is adopted in this work to implement the RGB-to-YUV space conversion, and $N = 16$ is specified for our proposed UGTS and luma modification algorithms. All the simulations are conducted not only on three popular image data sets, including the Kodak data set, the IMAX data set and the SCI data set, which are utilized in [7], but also on eight classical test images, including Lena, Airplane, Baboon, Barbara, Peppers, Splash, Tiffany and Monarch, as shown in Fig. 1. The experiments are performed on Matlab R2016b supported by the Intel Core i7-7500U CPU at 2.90 GHz with 8G memory and the Microsoft Windows 10 operating system. The experimental results are presented in this section, where the color PSNR (CPSNR) [7], derived from the color-mean-square-error (CMSE) over three color channels, is utilized as the distortion metric. Specifically, CPSNR is calculated via

$$CPSNR = 10\log_{10}\frac{255^2}{CMSE}. \tag{21}$$

A. Comparisons of Different Sub-Sampling Methods and Luma Modification Methods

Firstly, we compare our proposed UGTS algorithm with the traditional direct spatial down-sampling (DSDS) and the interpolation dependent image down-sampling (IDID) [4] by performing them on the U and V components of the converted YUV images. When UGTS is carried out, the full-resolution reconstruction is implemented based on (7). If DSDS or IDID is carried out, the bicubic interpolation is utilized to rebuild the full-resolution components. After the YUV-to-RGB space conversion, the average CPSNR result over all the test images for each image data set is calculated to evaluate the performance of the sub-sampling method. The average CPSNR results for the three sub-sampling methods are presented in Table I. As can be observed from Table I, our proposed UGTS offers the best reconstruction results compared with the DSDS-based and IDID-based methods. These results also indicate that our proposed sub-sampling method can well preserve the necessary information in the sub-sampled images to recover high-quality images.

TABLE I
Average CPSNR (dB) Results of Reconstructed Images

	DSDS	IDID	UGTS	JCSLM	TS-LM
IMAX	38.97	40.13	40.56	39.68	**41.26**
Kodak	45.93	47.89	48.07	46.52	**48.63**
SCI	33.77	35.51	35.75	34.22	**36.20**
Classical	35.33	37.46	38.05	35.95	**38.69**
Average	38.50	40.25	40.61	39.09	**41.20**

Then, we compare our proposed transform domain sub-sampling based luma modification algorithm (TS-LM) with JCSLM [7]. The corresponding CPSNR results are also p-resented in Table I. One can see from Table I that our proposed TS-LM achieves a much better reconstruction quality than JCSLM. Besides, TS-LM also outperforms our UGTS algorithm. These results demonstrate that our proposed TS-LM can well preserve the necessary information in the sub-sampled image and effectively modify the luma component to achieve low RGB distortions.

B. Compressing Color Images with Different Methods

In this experiment, we integrate our proposed UGTS algorithm and luma modification algorithm into JPEG [10] to construct the corresponding compression methods for color images. Specifically, UGTS is applied to both the chroma components and then the sub-sampled components are compressed accordingly. This method is denoted as the UGTS-based coding. Meanwhile, we utilize our proposed TS-LM algorithm to construct the TS-LM-based coding scheme. Besides, we design the compression-dependent TS-LM (CDTS-LM), as mentioned in Section IV-C, and employ it to form the CDTS-LM-based coding.

We also apply IDID to both the chroma components to implement the IDID-based coding and utilize the JCSLM algorithm to implement the JCSLM-based coding as the baseline methods. Note that all the coding methods are implemented based on JPEG to guarantee fair comparisons.

Compared with the traditional JPEG coding, the average BD-rate [11] saving over all the test images for each image da-ta set is employed to evaluate the rate-distortion performance for each compression method, where CPSNR is adopted as the distortion metric. The average BD-rate savings for the five compression methods are presented in Table II, in which the compression results of three QF sets, $\{20, 30, 40, 50\}$, $\{40, 50, 60, 70\}$ and $\{60, 70, 80, 90\}$, are introduced to evaluate the performances of different coding scenarios. According to the results presented in Table II, our UGTS-based coding has outperformed the JCSLM-based coding and IDID-based coding in all the coding scenarios. After our proposed luma modification algorithm is carried out, our TS-LM-based coding and CDTS-LM-based coding offer more significant performance gains compared with our UGTS-based coding.

For subjective comparisons, some compressed image portions of image Peppers are presented in Fig. 2. As can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2895840, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 1. Eight classical test images: Lena, Airplane, Baboon, Barbara, Peppers, Splash, Tiffany, and Monarch.

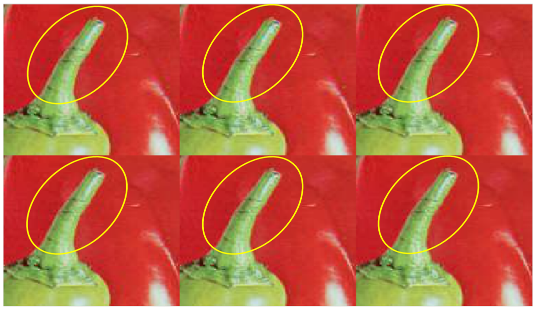

Fig. 2. Compressed image partons for Peppers: From left to right and from top to bottom: JPEG, JCSLM, IDID, UGTDS, TS-LM, and CDTS-LM.

TABLE II
AVERAGE BD-RATE REDUCTION FOR VARIOUS IMAGE SETS(%)

	QFs	JCSLM	IDID	UGTS	TS-LM	CDTS-LM
IMAX	20-50	-0.09	-0.24	-6.73	-7.31	**-12.38**
	40-70	-0.27	-0.49	-7.94	-8.92	**-14.33**
	60-90	-0.86	-1.63	-11.03	-12.61	**-17.04**
Kodak	20-50	-0.91	-0.37	-3.18	-3.30	**-4.89**
	40-70	-1.21	-0.69	-2.40	-2.59	**-4.11**
	60-90	-1.54	-1.56	-2.80	-3.10	**-4.51**
SCI	20-50	-1.31	-2.85	-4.51	-5.97	**-7.39**
	40-70	-1.97	-5.28	-7.43	-9.29	**-10.35**
	60-90	-2.97	-9.77	-12.79	-14.93	**-15.52**
Classical	20-50	-1.23	-4.03	-11.60	-13.09	**-17.56**
	40-70	-1.90	-6.28	-15.41	-17.73	**-22.53**
	60-90	-4.70	-13.98	-24.04	-27.45	**-32.19**
Average	20-50	-0.89	-1.87	-6.51	-7.42	**-10.56**
	40-70	-1.32	-3.19	-8.30	-9.63	**-12.83**
	60-90	-2.52	-6.74	-12.67	-14.52	**-17.32**

be observed from Fig. 2, our proposed methods, especially the CDTS-LM-based coding, generate the compressed color images with less artifacts.

C. Discussion about the Computational Complexity

When the proposed UGTS and luma modification algorithms are adopted in typical image coding scheme, they only introduce additional operations at the encoder side. Moreover, our proposed coding approaches do not require image interpolation to reconstruct the complete image at the decoder side. On the contrary, in the spatial sub-sampling based coding scheme, such as the traditional JPEG coding, the IDID-based coding and the JCSLM-based coding schemes, image interpolation is always required at the decoder side to reconstruct the final color image, which may induce a higher computational complexity.

Compared with the traditional JPEG coding, the increased or reduced executing time of the other five coding methods are presented in Table III, where QF is specified to 50. It can be seen from Table III that our proposed approaches can compress the color images with less time compared with the existing approaches. Due to the adoption of image interpolation at both the encoder and decoder sides, the JCSLM-based coding spends much more time on the compression of color images.

TABLE III
EXECUTING TIME OF DIFFERENT METHODS

	JCSLM	IDID	UGTS	TS-LM	CDTS-LM
IMAX	12.32%	3.46%	-3.81%	-2.17%	-1.69%
Kodak	8.52%	2.06%	-5.62%	-4.28%	-3.35%
SCI	11.36%	3.10%	-5.46%	-3.26%	-2.03%
Classical	17.51%	4.61%	-6.32%	-5.39%	-4.15%
Average	12.43%	3.31%	-5.30%	-3.78%	-2.81%

VI. CONCLUDING REMARKS

In this paper, we firstly propose a transform domain sub-sampling method and apply it to both the chroma components to yield a comparable sub-sampling performance to the YUV 4:2:0 format. Then, we propose a novel luma modification algorithm, which can simultaneously modify the luma component and sub-sample the chroma components based on our proposed transform domain sub-sampling method. In our proposed luma modification algorithm, the distortions occurred on the two chroma components can be coupled together and utilized to modify the luma component. Extensive experiments have demonstrated that our proposed methods achieve a significant coding gain over the state-of-the-art methods.

REFERENCES

[1] ITU-R, "Parameter values for high definition television systems for production and international programme exchange," *ITU-R Rec. BT.709-5*, Apr. 2002.

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

[3] X. L. Wu, X. J. Zhang, and X. H. 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, Nov. 2009.

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

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

[6] C.-H. Lin, K.-L. Chung, and C.-W. Yu, "Novel chroma subsampling strategy based on mathematical optimization for compressing mosaic videos with arbitrary RGB color filter arrays in H.264/AVC and HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26. no. 9, pp. 1712-1733, Sept. 2016.

[7] 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.

[8] N. Merhav and V. Bhaskaran,"Fast algorithm for DCT-domain image down-sampling and for inverse motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 468-476, Jun. 1997.

[9] 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.

[10] G. Hudson, A. Léger, B. Niss, and I. Sebestyén, "JPEG at 25: Still going strong," *IEEE MultiMedia*, vol. 24, no. 2, pp. 96-103, 2017.

[11] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEG-M33, 13th ITU-T Video Coding Experts Group (VCEG) Meeting, Austin, TX, USA, Apr. 2001.